國立臺灣大學電機資訊學院電機工程學系

碩士論文

Department of Electrical Engineering

College of Electrical Engineering and Computer Science

National Taiwan University

Master Thesis

即時車輛共乘問題在三點循環模型的競爭分析

Competitive Analysis of Online Car-sharing Problems With Three Locations

許孟翔

Meng-Hsiang Hsu

指導教授: 陳和麟 博士

Advisor: Ho-Lin Chen, Ph.D.

中華民國 112 年 7 月

July, 2023

doi:10.6342/NTU202302667

國立臺灣大學碩士學位論文
口試委員會審定書
MASTER'S THESIS ACCEPTANCE CERTIFICATE
NATIONAL TAIWAN UNIVERSITY

即時車輛共乘問題在三點循環模型的競爭分析

Competitive analysis of online car-sharing problem with three locations

本論文係_____許孟翔_______(姓名)_____R10921094_______（學號）在國立臺灣大學電機工程學系完成之碩士學位論文，於民國_112__年__7__月_26_日承下列考試委員審查通過及口試及格，特此證明。

The undersigned, appointed by the Department of Electrical Engineering on 26 (date) 07 (month) 2023 (year) have examined a Master's thesis entitled above presented by_____Hsu, Meng-Hsiang______ (name)_________R10921094___________ (student ID) candidate and hereby certify that it is worthy of acceptance.

口試委員 Oral examination committee:

陳和麟

（指導教授 Advisor）

系主任 Director: 李建模

doi:10.6342/NTU202302667

Acknowledgements

一個半月的時間，只能宅在家裡寫論文的我躺在床上，有無數個瞬間腦海飄過已經通過論文、口試地獄，在寫謝辭的畫面，但在目前的當下又遲遲無法下筆，顯得格外矛盾。

這種矛盾感來自我的指導教授陳和麟老師，老師無論在課堂、研究、時間安排上都給予最大的彈性空間，並信任每位學生都能做到最好。這點對我來說非常重要，因為我對於時間安排都有自己的想法，在這兩年和老師研究、上課的相處過程中，老師的細心指導都能讓我對於原本問題的看法有更深刻的理解，也從不對學生的問題感到不耐煩，挑選研究主題更是尊重我的喜好。在全心投入研究的一個多月中，與老師每週定時的 1-1 個人指導並得到肯定，是我人生求學階段最快樂的時光。對比之下，之後的論文寫作雖然很痛苦，過程中改了好幾個版本，也時時念念想快點結束一切，但最後寫謝辭前還是忍不住在電腦前短暫回憶了我的碩士生涯，非常謝謝老師。

謝謝我的父母在經濟上給于支柱，我不太懂情感、壓力的方面如何宣洩，有時候會無意間把情緒垃圾倒給你們。在這兩年，因為台中台北的距離因素，鮮少回家，感謝你們在各方面的包容。

謝謝怡堯在這兩年間的學校瑣事、Latex 寫作上幫了許多，跟你討論演算法和打比賽也是碩士期間的歡樂時光。

doi:10.6342/NTU202302667

謝謝芃廷幫我錄影，因為我早九起不來，睡得飽很重要。

謝謝泰宇提供在口試前給我 Beamer 的模板，我不用花費大把力氣在投影片的製作。

謝謝實驗室的學弟妹們在日常生活的陪伴，以及在口試當天的幫忙。

謝謝 ChatGPT 老師，在 Latex 語法、英文字句修飾方面幫了我很多。

謝謝嘉佑，應你要求寫上來的，但我想不到感謝理由。

doi:10.6342/NTU202302667

中文摘要

在汽車共乘問題中，我們旨在有效地分配有限的車輛來滿足城市中許多服務站的顧客。顧客可以在線上來指定他們的接送時間、接送地點和下車地點。演算法會以即時的方式來決定接受或拒絕請求。我們的目標是最大化滿足顧客需求的數量。

在之前的研究中，羅等人提出了一個新的車輛共乘模型，在這個模型中，每個請求都指定了預訂時間（提交請求的時間）、接送時間和接送地點。梁等人延續前面的研究，同時引入了一個新的模型稱為 kS2L-S，該模型具有 k 個服務器和兩個地點。在這個模型中，演算法可以同時看到所有在相同預訂時間的請求，並同時做出決策。梁等人證明了他們提出的貪婪平衡演算法（GBA）是最佳的。

在我們的研究中，我們想知道 GBA 中貪婪和平衡的想法是否仍然適用於延伸的 kS2L-S 模型。我們將 kS2L-S 模型延伸為三個地點的循環單向模型，稱為 kS3L-S。我們的目標是在時間內最大化滿足（接受）請求的數量。我們證明了 kS2L-S 中貪婪和平衡的想法仍然適用於 kS3L-S，並提出了相對應的貪婪平衡演算法 GBA-3L。在競爭分析中，我們展示了 kS3L-S 的下界是 $\frac{2k}{k+\lfloor k/3 \rfloor}$，而 GBA-3L 的 competitive ratio 也是 $\frac{2k}{k+\lfloor k/3 \rfloor}$。因此，我們證明了演算法 GBA-3L 是最佳的。

關鍵字：車輛共乘問題、競爭分析、線上演算法

doi:10.6342/NTU202302667

doi:10.6342/NTU202302667

Abstract

In the car-sharing problem, we aim to efficiently assign the limited cars to customers across many service stations in a city. Customers can make online requests specifying their pick-up time, pick-up location and drop-off location. The decision to accept or reject a request is made in online fashion. Our goal is to maximize the number of fulfilled requests from customers.

Luo et al. proposed a new car-sharing model in which each request specifies its booking time (request submission time), pick-up time, and pick-up location. Liang et al. followed the previous car-sharing formulation and then introduced a new simultaneous model called kS2L-S, featuring k servers and two locations. In this model, the scheduler (algorithm) can view all requests at the same booking time and make decisions together simultaneously. Liang et al. demonstrated that the greedy balanced algorithm (GBA) they proposed is optimal.

In our work, we wonder whether the key concepts of greediness and balance in GBA

doi:10.6342/NTU202302667

are still applicable to the extended work of kS2L-S. We extend kS2L-S to a circular one-way model with three locations, named kS3L-S. Our goal is to maximize the number of fulfilled (accepted) requests over time as much as possible. We demonstrate that the key concepts of greediness and balance in kS2L-S are still applicable to kS3L-S and present the extended greedy balanced algorithm named GBA-3L. We present that the lower bound of the competitive ratio for kS3L-S is $\frac{2k}{k+\lfloor k/3 \rfloor}$ and the the competitive ratio of GBA-3L is also $\frac{2k}{k+\lfloor k/3 \rfloor}$. Therefore, we show that the bound is tight and our algorithm is optimal.

Keywords: Car-sharing, Competitive analysis, On-line scheduling

doi:10.6342/NTU202302667

Contents

doi:10.6342/NTU202302667

doi:10.6342/NTU202302667

List of Figures

doi:10.6342/NTU202302667

doi:10.6342/NTU202302667

List of Tables

doi:10.6342/NTU202302667

doi:10.6342/NTU202302667

Chapter 1 Introduction

1.1 Motivation

Our work is focused on the online car-sharing problem. In the car-sharing problem, we aim to efficiently assign the limited cars to customers (as well as other resources such as bikes or shuttle-buses) across many service stations in a city. Customers can make online requests specifying their pick-up time, pick-up location and drop-off location. The decision to accept or reject a request is made in online fashion. Our goal is to maximize the number of fulfilled requests from customers. Car-sharing models are applicable to various transportation systems, including bike systems (such as U-bike in Taipei and Citi bike in New York), bus systems and subway systems in cities, with the purpose of reducing congestion during peak hours. This is particularly important for big cities. As a result, many car-sharing models have been proposed in recent years to efficiently allocate these limited resources."

When I entered National Taiwan University (NTU), I noticed the imbalanced distribution of U-Bikes across the campus, which often bothered me. The lack of available U-Bikes near the campus frequently caused me to be late for my 9 a.m. classes. Additionally, if there were no U-Bikes near my department buildings after my 5 p.m. class, I had to walk through the campus to the MRT station, which took me about 15 minutes. For me,

doi:10.6342/NTU202302667

the car-sharing problem is not just a research topic but also a part of my daily life.

1.2 Prior works

Let R be a sequence of requests that are released over time and L be the set of locations. We denote the i-th request by $r_i = (\tilde{t_i}, t_i, p_i, \dot{p_i})$ which is specified by the booking time or release time $\tilde{t_i}$ (request submission time), the start time t_i, the pick-up location $p_i \in L$ and the drop-off location $\dot{p_i} \in L$. The traveling time from one location to another location is denoted by t and the time interval $t_i - \tilde{t_i}$ is called the booking interval. If r_i is accepted, a server must be allocated to the customer at time t_i at location p_i and returned by the customer at time $t_i + t$ at $\dot{p_i}$. Every server can only fulfill one customer request at a time. Serving one customer can yield a profit of r. Additionally, empty moves are allowed with a cost of c.

Luo et al. [7] studied the problem with a server traveling between two locations, considering both fixed booking time and variable booking time. It should be noted that variable booking time refers to the concept where the (i-1)-th request, which has a start time of t_{i-1}, can occur after the i-th request with a start time of t_i. Luo et al [8] extended the previous problem to two locations and two servers, known as 2S2L. They showed the lower bound of the competitive ratio when $0 \leq c \leq r$ and the upper bound of the competitive ratio when $0 \leq c < r$. Luo et al. [9] extended the previous problem 2S2L to the problem involving two locations and k servers for both fixed booking time (kS2L-F) and variable booking time (kS2L-V). They showed that the competitive ratios in both kS2L-F and kS2L-V are at least 1.5 for all k and proposed a 1.5-competitive algorithm for $k = 3i$. They also showed that the competitive ratio is at least 5/3 for all k and proposed a

doi:10.6342/NTU202302667

5/3-competitive algorithm for $k = 5i$ for kS2L-V. Luo et al. [10] studied on the car-sharing problem within a star network configuration involving k servers. They specifically studied on two variants of travel time: one with a fixed unit travel time and another with a variable travel time.

Liang et al. [6] extended the work on kS2L-F for cases when $k \neq 3i$. They showed that the competitive ratio for kS2L-F is at least $\frac{2k}{k+\lfloor k/3 \rfloor}$ and at most $\frac{2k}{k+\lfloor k/3 \rfloor}$ for all values of k. Additionally, when randomization is allowed, they showed that the competitive ratio for kS2L-F is at least 1.5 and at most 1.5. Furthermore, Liang et al. [6] introduced a new simultaneous decision model called kS2L-S, which involves k servers and two locations. They proposed a greedy balanced algorithm (GBA), which is a $\frac{2k}{k+\lfloor k/2 \rfloor}$-competitve algorithm, and showed that the competitive ratio for kS2L-S is at least $\frac{2k}{k+\lfloor k/2 \rfloor}$. Similarly, when randomization is allowed, they showed that the competitive ratio for kS2L-S is at least 4/3 and at most 4/3.

In comparison with the online models, Berbeglia et al. [3] showed that when all customer requests for car bookings are known in advance (offline car-sharing model), the problem of maximizing the number of accepted requests can be solved in polynomial time. In addition, they considered a specific variant of the problem where two locations are involved, and each customer requests two rides in opposite directions. The scheduler is then required to either accept both rides or reject both rides simultaneously. They proved that this variant is NP-hard and APX-hard.

Another closely related area of research is the online dial-a-ride problem (OLDARP). In OLDARP, requests for transportation between specific points in a metric space arrive incrementally over time. The problem has various variations, each with different objec-

doi:10.6342/NTU202302667

tives such as minimizing the total makespan [1, 2] or the maximum flow time [5]. Christman et al. [4] focused on a variation of OLDARP where each request is associated with a revenue. The objective is to efficiently serve the requests within a given time limit while maximizing the overall revenue. Another variant of OLDARP was investigated by Yi et al. [13], where each request is associated with a deadline. The aim is to devise an online server motion plan that fulfills the maximum number of requests within their respective deadlines.

In practice, there are many simulation studies on the car-sharing problem. Repoux et al. [12] redesigned a new car relocation strategy to address the issue of imbalanced car distribution. Perboli et al. [11] introduced Monte Carlo simulation to compare four different business models. This work may be the first comprehensive analysis from a business model viewpoint.

1.3 Our results

In the real world, many circular transportation systems are designed for easier navigation, increased resilience, and a more balanced passenger load when compared to linear lines. For example, Santander Cycles in London, Central Area Transit in Launceston, and the Yamanote Line in Tokyo Metro are two-way circular systems. However, there is a need to develop a one-way circular system with fewer locations (two or three locations). From the provider's perspective, one-way system can reduce transportation costs, while from the passenger's perspective, even the farthest destination is acceptable. For example, certain school shuttle bus systems like those between NTU and Academia Sinica or the Marguerite shuttle in Stanford campus.

doi:10.6342/NTU202302667

We are basically following the circular one-way simultaneous decision model proposed by Liang et al. Since some systems (subway systems) are designed to prioritize passenger volume and are relatively less concerned about costs, which aligns with our concerns, and some systems (bike systems) have costs much lower than profit. In the extension work, we would like to study the empty move cost with zero. Furthermore, we want to know if the key concepts of greediness and balance in GBA are still applicable to the extension work of kS2L-S. We naturally extend kS2L-S to the circular one-way model with three locations. This model is named kS3L-S, and the scheduler can make decisions on the accumulated requests until a certain time. It is equivalent to the scheduler always making decisions at the discrete time intervals. For kS3L-S, We propose a new algorithm following the similar ideas in kS2L-S, called GBA-3L.

We summarize the prior works that are closely related to our own work below. Table 1.1 summarizes 2S2L results with the continuous start time and different empty move costs $0 \leq c \leq y$. Table 1.2 summarizes kS2L and kS3L results with the discrete start time, i.e. $t_i = vt$ for $v \in \mathbb{N}$, and empty move cost 0. It should be noted that D is for deterministic and R is for randomized in algorithm type.

Problem	**Booking interval**	**Empty move cost**	**Algorithm type**	**Lower bound**	**Upper bound**	**Reference**
2S2L	Fixed	$c = y$	D	-	1	MFCS' 18 [8]
2S2L	Fixed	0	D	2	2	MFCS' 18 [8]
2S2L	Fixed	$0 < c < y$	D	2	2	MFCS' 18 [8]

Table 1.1: Summary of 2S2L

Our contribution is that we follow the key concept of GBA in kS2L-S and propose the extension version GBA-3L in kS3L-S. Furthermore, we address the challenge for kS3L-S in the induction proof where one location has a significantly high number of servers, while the other two locations have significantly fewer servers. It will result in allocating all the

doi:10.6342/NTU202302667

Problem	Booking interval	Algorithm type	Lower bound	Upper bound	Reference
kS2L-F	Fixed	D	1.5	$1.5(k = 3i, i \in \mathbb{N})$	ISAAC' 18 [9]
kS2L-F	Fixed	D	$\frac{2k}{k+\lfloor k/3 \rfloor}$	$\frac{2k}{k+\lfloor k/3 \rfloor}$	TCS' 22 [6]
kS2L-F	Fixed	R	1.5	1.5	TCS' 22 [6]
kS2L-V	Variant	D	1.5	$1.5(k = 3i, i \in \mathbb{N})$	ISAAC' 18 [9]
kS2L-V	Variant	D	$5/3$	$5/3(k = 5i, i \in \mathbb{N})$	ISAAC' 18 [9]
kS2L-S	Fixed	D	$\frac{2k}{k+\lfloor k/2 \rfloor}$	$\frac{2k}{k+\lfloor k/2 \rfloor}$	TCS' 22 [6]
kS2L-S	Fixed	R	$4/3$	$4/3$	TCS' 22 [6]
kS3L-S	Fixed	D	$\frac{2k}{k+\lfloor k/3 \rfloor}$	$\frac{2k}{k+\lfloor k/3 \rfloor}$	This paper

Table 1.2: Summary of kS2L and kS3L

free servers to the location with the fewest servers, while the number of servers at the second fewest location remains low. In contrast, in kS2L-S, GBA can always allocate all the free servers to the location with the fewest local servers at each stage. Finally, we prove that the lower bound of the competitive ratio for kS3L-S is $\frac{2k}{k+\lfloor k/3 \rfloor}$, and we demonstrate that the competitive ratio of GBA-3L is also $\frac{2k}{k+\lfloor k/3 \rfloor}$. The proof illustrate our algorithm is optimal.

In chapter 2, we will formulate the problem kS3L-S. In chapter 3, we will present two examples to elaborate the key concept of GBA-3L. Furthermore, we will demonstrate that GBA-3L is a $\frac{2k}{k+\lfloor k/3 \rfloor}$-competitive algorithm by induction hypothesis. In chapter 4, we make the conclusion of this paper.

doi:10.6342/NTU202302667

Chapter 2 Preliminaries and Problem formulation for our work

In this chapter, we will formulate the problem kS3L-S. We are basically following the previous work kS2L-S [6], extending two locations to three locations.

Let R be a sequence of requests that are released over time and L be the set of locations. We denote the i-th request by $r_i = (\tilde{t_i}, t_i, p_i, \dot{p_i})$ which is specified by the booking time or release time $\tilde{t_i}$ (request submission time), the start time t_i, the pick-up location $p_i \in L$ and the drop-off location $\dot{p_i} \in L$. The traveling time from one location to another location is denoted by t and the time interval between the submission of a request (booking time) and the pick-up time is called the booking interval. If r_i is accepted, a server must be allocated to the customer at time t_i at location p_i and returned by the customer at time $t_i + t$ at $\dot{p_i}$. Each server can only serve one customer at a time. Our goal is to maximize the total fulfilled requests given a sequence of requests.

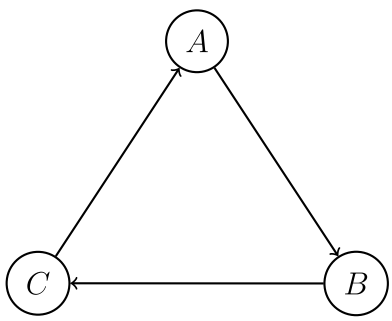

Figure 2.1: kS3L-S model

doi:10.6342/NTU202302667

For our work, as shown in Figure 2.1, kS3L-S is a one-way circular model with three locations and k servers. The pick-up location can be one of A, B, or C, with the corresponding drop-off locations being B, C, and A, respectively. Furthermore, the empty move for one server takes time t.

Moreover, kS3L-S is also a simultaneous decision model. In this model, assuming that the current time is vt, the scheduler can observe all requests exactly with the same start time at $t_i = (v+1)t$ and make decisions together simultaneously at the current time. We can take advantage of this feature to reallocate idle servers (or free servers), which are not serving customers at time t, to handle the future requests $t_i = (v+1)t$ in advance. For example, there are k free servers at time 0, the scheduler can observe the requests with the start time t at three locations and move the free servers in advance.

In the next chapter, we present the discrete time as different stages.

doi:10.6342/NTU202302667

Chapter 3 Deterministic algorithm

3.1 Greedy balanced algorithm with three locations

In this chapter, we will present GBA-3L for kS3L-F and elaborate on the key concepts of GBA-3L. Since GBA-3L needs to be compared to an offline optimal algorithm, we denote the offline optimal solution as OPT-3L. The following table summarizes the symbols and their descriptions used in this paper.

Symbols	**Descriptions**
k	The number of total servers.
$I_{\overrightarrow{AB}_n}$	The number of the requests from location A to B for the stage n.
$I_{\overrightarrow{BC}_n}$	The number of the requests from location B to C for the stage n.
$I_{\overrightarrow{CA}_n}$	The number of the requests from location C to A for the stage n.
$G_{\overrightarrow{AB}_n}$	The number of the requests from location A to B accepted by GBA-3L for the stage n.
$G_{\overrightarrow{BC}_n}$	The number of the requests from location B to C accepted by GBA-3L for the stage n.
$G_{\overrightarrow{CA}_n}$	The number of the requests from location C to A accepted by GBA-3L for the stage n.
A_n	The number of initial servers at location A for the stage n before GBA-3L allocates any free server to location A.

doi:10.6342/NTU202302667

B_n	The number of initial servers at location B for the stage n before GBA-3L allocates any free server to location B.
C_n	The number of initial servers at location C for the stage n before GBA-3L allocates any free server to location C.
$G_{\vec{f}_n}$	The number of servers not used by GBA-3L for the stage n.
$O_{\overrightarrow{AB}_n}$	The number of the requests from location A to B accepted by OPT-3L for the stage n.
$O_{\overrightarrow{BC}_n}$	The number of the requests from location B to C accepted by OPT-3L for the stage n.
$O_{\overrightarrow{CA}_n}$	The number of the requests from location C to A accepted by OPT-3L for the stage n.
$O_{\vec{f}_n}$	The number of servers not used by OPT-3L for the stage n.

Table 3.1: Symbols and Corresponding Descriptions

In the above definition, we know that $A_n = G_{\overrightarrow{CA}_{n-1}}$. Additionally, it should be noted that $G_{\overrightarrow{AB}_0}$, $G_{\overrightarrow{BC}_0}$, $G_{\overrightarrow{CA}_0}$ and $G_{\vec{f}_0}$ are 0, 0, 0 and k at the stage 0, respectively.

The key concept of GBA-3L is to fulfill as many requests as possible first, and then balance every accepted requests at three locations as much as possible. Suppose initially there are k = 100 servers, i.e. $G_{\vec{f}_0} = 100$. Consider two examples, one is ($I_{\overrightarrow{AB}_1}$, $I_{\overrightarrow{BC}_1}$, $I_{\overrightarrow{CA}_1}$) = (40, 30, 20) and the other is ($I_{\overrightarrow{AB}_1}$, $I_{\overrightarrow{BC}_1}$, $I_{\overrightarrow{CA}_1}$) = (60, 60, 20) at stage 1. How will GBA-3L allocate these servers? In the first example, GBA-3L will allocate ($G_{\overrightarrow{AB}_1}$, $G_{\overrightarrow{BC}_1}$, $G_{\overrightarrow{CA}_1}$) = (40, 30, 20). This is because we need to fulfill requests first, in this case, all requests can be fulfilled. In the second example, GBA-3L will allocate ($G_{\overrightarrow{AB}_1}$, $G_{\overrightarrow{BC}_1}$, $G_{\overrightarrow{CA}_1}$) = (40, 40, 20). Unfortunately, we cannot fulfill all requests this time. The maximum number of fulfilled requests at stage 1 is k, but there are many allocation methods that result in a total of k fulfilled requests. Therefore, we need to balance the fulfilled re-

doi:10.6342/NTU202302667

quests at the three locations as much as possible. Algorithm 1 will elaborate on the greedy balanced idea.

Algorithm 1 Greedy Balanced Algorithm with three locations (GBA-3L)

Input: An instance I, which has n stages of the requests at three locations;
Output: The total fulfilled requests for the instance I.

$G_n \longleftarrow 0$
$G_{\overrightarrow{AB}_0} \longleftarrow 0$
$G_{\overrightarrow{BC}_0} \longleftarrow 0$
$G_{\overrightarrow{CA}_0} \longleftarrow 0$
$G_{\overrightarrow{f}_0} \longleftarrow k$
for i **from** 1 **to** n **do**
    $G_{\overrightarrow{AB}_i} \longleftarrow min\{I_{\overrightarrow{AB}_i}, G_{\overrightarrow{CA}_{i-1}}\}$
    $G_{\overrightarrow{BC}_i} \longleftarrow min\{I_{\overrightarrow{BC}_i}, G_{\overrightarrow{AB}_{i-1}}\}$
    $G_{\overrightarrow{CA}_i} \longleftarrow min\{I_{\overrightarrow{CA}_i}, G_{\overrightarrow{BC}_{i-1}}\}$
    while $G_{\overrightarrow{f}_{i-1}} > 0$ **do**
        if All requests at stage i are fulfilled **then**
            break
        end if
        Let S_L be the set of locations where requests are not completely fulfilled.
        Add one server to the location with the fewest assigned servers in S_L. If multiple locations have the same minimum number of assigned servers, add a free server to the location with the highest initial number of local servers, or break the tie arbitrarily.
        $G_{\overrightarrow{f}_{i-1}}$ minus 1.
    end while
    $G_{\overrightarrow{f}_i} \longleftarrow k - G_{\overrightarrow{AB}_i} - G_{\overrightarrow{BC}_i} - G_{\overrightarrow{CA}_i}$
    $G_n \longleftarrow G_n + G_{\overrightarrow{AB}_i} + G_{\overrightarrow{BC}_i} + G_{\overrightarrow{CA}_i}$
end for
return G_n;

Theorem 1. *GBA-3L is a $1/\delta$-competitive algorithm for kS3L-S for any $k \geq 2$, where $\delta = \frac{k+\lfloor k/3 \rfloor}{2k}$.*

To prove $\delta = \frac{k+\lfloor k/3 \rfloor}{2k}$ in GBA-3L, we need to introduce the following definitions and lemmas.

Proof.

Definition 1. *Given any instance I, G_n and O_n denote the total fulfilled requests of GBA-3L and OPT-3L for the first n stages of I.*

doi:10.6342/NTU202302667

$$G_n = \sum_{i=1}^{n}(G_{\overrightarrow{AB}_i} + G_{\overrightarrow{BC}_i} + G_{\overrightarrow{CA}_i}) \tag{3.1}$$

$$O_n = \sum_{i=1}^{n}(O_{\overrightarrow{AB}_i} + O_{\overrightarrow{BC}_i} + O_{\overrightarrow{CA}_i}) \tag{3.2}$$

Definition 2. *Given any instance I, let $I_{n,AB}$ be the requests of I for the first n stages combined with the k requests from A to B and k requests from B to C at stage n + 1.*

Definition 3. *Given any instance I and any optimal solution OPT-3L of the instance I. Let $G_{n,AB} = G_n + G_{\overrightarrow{CA}_n} + G_{\overrightarrow{AB}_n} + G_{\overrightarrow{f}_n}$ and $O_{n,AB} = O_n + O_{\overrightarrow{CA}_n} + O_{\overrightarrow{AB}_n} + O_{\overrightarrow{f}_n}$.*

Intuitively, in the above definition, $G_{n,AB}$ and $O_{n,AB}$ denote the total number of fulfilled requests for the input instance $I_{n,AB}$ in two different solutions. $G_{n,AB}$ denotes the total number of fulfilled requests when the GBA-3L algorithm receives instance $I_{n,AB}$. $O_{n,AB}$ denotes the total number of fulfilled requests when the solution of $I_{n,AB}$ follows OPT-3L (an optimal solution of I) in the first n stages and try to fulfill as many requests as possible in stage $n + 1$.

Furthermore, notice that equations 3.3 and 3.4 holds because $G_{\overrightarrow{AB}_n} + G_{\overrightarrow{BC}_n} + G_{\overrightarrow{CA}_n} + G_{\overrightarrow{f}_n} = k$, and $O_{\overrightarrow{AB}_n} + O_{\overrightarrow{BC}_n} + O_{\overrightarrow{CA}_n} + O_{\overrightarrow{f}_n} = k$.

$$G_{n,AB} = G_{n-1} + G_{\overrightarrow{CA}_n} + G_{\overrightarrow{AB}_n} + k \tag{3.3}$$

$$O_{n,AB} = O_{n-1} + O_{\overrightarrow{CA}_n} + O_{\overrightarrow{AB}_n} + k \tag{3.4}$$

Here, $G_{n,BC}, O_{n,BC}, G_{n,CA}, G_{n,CA}$ are defined symmetrically.

Definition 4. *Given any instance I, let $I_{n,A}$ be the requests of I for the first n stages combined with the k requests from A to B at stage n + 1.*

doi:10.6342/NTU202302667

Definition 5. *Given any instance I and any optimal solution OPT-3L of the instance I. Let $G_{n,A} = G_n + G_{\overrightarrow{CA}_n} + G_{\overrightarrow{f}_n}$ and $O_{n,A} = O_n + O_{\overrightarrow{CA}_n} + O_{\overrightarrow{f}_n}$.*

Intuitively, in the above definition, $G_{n,A}$ and $O_{n,A}$ denote the total number of fulfilled requests for the input instance $I_{n,A}$ in two different solutions. $G_{n,A}$ denotes the total number of fulfilled requests when the GBA-3L algorithm receives instance $I_{n,A}$. $O_{n,A}$ denotes the total number of fulfilled requests when the solution of $I_{n,A}$ follows OPT-3L (an optimal solution of I) in the first n stages and try to fulfill as many requests as possible in stage $n+1$.

Furthermore, notice that equations 3.5 and 3.6 holds because $G_{\overrightarrow{AB}_n} + G_{\overrightarrow{BC}_n} + G_{\overrightarrow{CA}_n} + G_{\overrightarrow{f}_n} = k$, and $O_{\overrightarrow{AB}_n} + O_{\overrightarrow{BC}_n} + O_{\overrightarrow{CA}_n} + O_{\overrightarrow{f}_n} = k$.

$$G_{n,A} = G_n + G_{\overrightarrow{CA}_n} + G_{\overrightarrow{f}_n} = G_{n-1} + G_{\overrightarrow{CA}_n} + k \tag{3.5}$$

$$O_{n,A} = O_n + O_{\overrightarrow{CA}_n} + O_{\overrightarrow{f}_n} = O_{n-1} + O_{\overrightarrow{CA}_n} + k \tag{3.6}$$

Here, $G_{n,B}, O_{n,B}, G_{n,C}, G_{n,C}$ are defined symmetrically.

We will prove that lemma 2 to lemma 7 hold for $n \geq 0$ by induction in section 3.1.1 and section 3.1.2. For convenience, we denote the set of lemma 2 to lemma 7 as S(n). Then, in section 3.1.3, we will demonstrate that lemma 1 holds.

Lemma 1. *Given any instance of I, the inequality $G_n \geq \delta O_n$ holds for all stages n, where $\delta = \frac{k+\lfloor k/3 \rfloor}{2k}$.*

Lemma 1 is an equivalent statement of theorem 1.

Lemma 2. *Given any instance I and any optimal solution OPT-3L of the instance I, the*

doi:10.6342/NTU202302667

inequality $G_{n,A} \geq \delta O_{n,A}$ holds for all stages n, where $\delta = \frac{k+\lfloor k/3 \rfloor}{2k}$.

Lemma 3. *Given any instance I and any optimal solution OPT-3L of the instance I, the inequality $G_{n,B} \geq \delta O_{n,B}$ holds for all stages n, where $\delta = \frac{k+\lfloor k/3 \rfloor}{2k}$.*

Lemma 4. *Given any instance I and any optimal solution OPT-3L of the instance I, the inequality $G_{n,C} \geq \delta O_{n,C}$ holds for all stages n, where $\delta = \frac{k+\lfloor k/3 \rfloor}{2k}$.*

Lemma 5. *Given any instance I and any optimal solution OPT-3L of the instance I, the inequality $G_{n,AB} \geq \delta O_{n,AB}$ holds for all stages n, where $\delta = \frac{k+\lfloor k/3 \rfloor}{2k}$.*

Lemma 6. *Given any instance I and any optimal solution OPT-3L of the instance I, the inequality $G_{n,BC} \geq \delta O_{n,BC}$ holds for all stages n, where $\delta = \frac{k+\lfloor k/3 \rfloor}{2k}$.*

Lemma 7. *Given any instance I and any optimal solution OPT-3L of the instance I, the inequality $G_{n,CA} \geq \delta O_{n,CA}$ holds for all stages n, where $\delta = \frac{k+\lfloor k/3 \rfloor}{2k}$.*

Suppose that the induction hypothesis S(j) holds when $0 \leq j \leq n-1$. We will prove that S(n) still holds in section 3.1.1 and section 3.1.2.

For the base case, we have $G_0 = O_0 = G_{0,A} = O_{0,A} = G_{0,B} = O_{0,B} = G_{0,C} = O_{0,C} = G_{0,AB} = O_{0,AB} = G_{0,BC} = O_{0,BC} = G_{0,CA} = O_{0,CA} = 0$ and $G_{\overrightarrow{f}_0} = O_{\overrightarrow{f}_0} = k$. When $n = 0$, it is obvious that lemma 2 to lemma 7 hold since $\delta \leq 1$.

But before proving S(n) in section 3.1.1 and section 3.1.2, we need lemma 8 to lemma 14, which will be used repeatedly.

Lemma 8. $O_{\overrightarrow{AB}_i} \leq I_{\overrightarrow{AB}_i}$, $O_{\overrightarrow{AB}_i} \leq O_{\overrightarrow{CA}_{i-1}} + O_{\overrightarrow{f}_{i-1}}$

Proof. The OPT-3L choice $O_{\overrightarrow{AB}_i}$ cannot exceed the requests $I_{\overrightarrow{AB}_i}$. $O_{\overrightarrow{CA}_{i-1}} + O_{\overrightarrow{f}_{i-1}}$ is the maximum available servers at location A for next stage. Thus, $O_{\overrightarrow{AB}_i}$ cannot exceed

doi:10.6342/NTU202302667

$O_{\overrightarrow{CA}_{i-1}} + O_{\overrightarrow{f}_{i-1}}$. □

The proofs of lemma 9 and lemma 10 are similar and will be omitted.

Lemma 9. $O_{\overrightarrow{BC}_i} \leq I_{\overrightarrow{BC}_i}$, $O_{\overrightarrow{BC}_i} \leq O_{\overrightarrow{AB}_{i-1}} + O_{\overrightarrow{f}_{i-1}}$

Lemma 10. $O_{\overrightarrow{CA}_i} \leq I_{\overrightarrow{CA}_i}$, $O_{\overrightarrow{CA}_i} \leq O_{\overrightarrow{BC}_{i-1}} + O_{\overrightarrow{f}_{i-1}}$

Lemma 11. $O_{\overrightarrow{AB}_i} + O_{\overrightarrow{BC}_i} \leq O_{\overrightarrow{CA}_{i-1}} + O_{\overrightarrow{AB}_{i-1}} + O_{\overrightarrow{f}_{i-1}}$

Proof. $O_{\overrightarrow{CA}_{i-1}} + O_{\overrightarrow{AB}_{i-1}} + O_{\overrightarrow{f}_{i-1}}$ is the maximum available servers at location A and B for next stage. Thus, $O_{\overrightarrow{AB}_i} + O_{\overrightarrow{BC}_i}$ cannot exceed $O_{\overrightarrow{CA}_{i-1}} + O_{\overrightarrow{AB}_{i-1}} + O_{\overrightarrow{f}_{i-1}}$. □

The proofs of lemma 12 and lemma 13 are similar and will be omitted.

Lemma 12. $O_{\overrightarrow{BC}_i} + O_{\overrightarrow{CA}_i} \leq O_{\overrightarrow{AB}_{i-1}} + O_{\overrightarrow{BC}_{i-1}} + O_{\overrightarrow{f}_{i-1}}$

Lemma 13. $O_{\overrightarrow{CA}_i} + O_{\overrightarrow{AB}_i} \leq O_{\overrightarrow{BC}_{i-1}} + O_{\overrightarrow{CA}_{i-1}} + O_{\overrightarrow{f}_{i-1}}$

Lemma 14. $O_{\overrightarrow{AB}_i} + O_{\overrightarrow{BC}_i} + O_{\overrightarrow{CA}_i} \leq O_{\overrightarrow{CA}_{i-1}} + O_{\overrightarrow{AB}_{i-1}} + O_{\overrightarrow{BC}_{i-1}} + O_{\overrightarrow{f}_{i-1}} = k$

Now, we start to prove S(j) will hold when $j = n$. Without loss of generality, we assume that $A_n \geq B_n \geq C_n$ at stage n. For readability, let's denote $G_{\overrightarrow{f}_{n-1}}$ as F_n. We will prove that S(n) holds in the following section 3.1.1 and section 3.1.2.

3.1.1 The initial servers can fulfill its requests at location C.

In this section, we will list all the cases when the initial local servers can fulfill the requests at location C, i.e., $C_n \geq I_{\overrightarrow{CA}_n}$, and prove that $S(n)$ holds based on the previous induction step $S(n-1)$ for all cases. It should be noted that $C_n \leq \lfloor \frac{k}{3} \rfloor$.

doi:10.6342/NTU202302667

1. $A_n \geq I_{\overrightarrow{AB}_n}$, $B_n \geq I_{\overrightarrow{BC}_n}$.

$(G_{\overrightarrow{AB}_n}, G_{\overrightarrow{BC}_n}, G_{\overrightarrow{CA}_n}) = (I_{\overrightarrow{AB}_n}, I_{\overrightarrow{BC}_n}, I_{\overrightarrow{CA}_n})$

In this case, the initial servers can fulfill the requests at location A, B and C.

2. $A_n \geq I_{\overrightarrow{AB}_n}$, $B_n < I_{\overrightarrow{BC}_n}$.

2.1. $B_n + F_n \geq I_{\overrightarrow{BC}_n}$.

$(G_{\overrightarrow{AB}_n}, G_{\overrightarrow{BC}_n}, G_{\overrightarrow{CA}_n}) = (I_{\overrightarrow{AB}_n}, I_{\overrightarrow{BC}_n}, I_{\overrightarrow{CA}_n})$

In this case, the initial servers can fulfill the requests at locations A and C. In the GBA-3L while loop, the free servers are sufficient to fulfill the requests at location B.

2.2. $B_n + F_n < I_{\overrightarrow{BC}_n}$.

$(G_{\overrightarrow{AB}_n}, G_{\overrightarrow{BC}_n}, G_{\overrightarrow{CA}_n}) = (I_{\overrightarrow{AB}_n}, B_n + F_n, I_{\overrightarrow{CA}_n})$

In this case, the initial servers can fulfill the requests at locations A and C. In the GBA-3L while loop, all free servers will be allocated to location B.

3. $A_n < I_{\overrightarrow{AB}_n}$, $B_n \geq I_{\overrightarrow{BC}_n}$.

3.1. $A_n + F_n \geq I_{\overrightarrow{AB}_n}$.

$(G_{\overrightarrow{AB}_n}, G_{\overrightarrow{BC}_n}, G_{\overrightarrow{CA}_n}) = (I_{\overrightarrow{AB}_n}, I_{\overrightarrow{BC}_n}, I_{\overrightarrow{CA}_n})$

In this case, the initial servers can fulfill the requests at locations B and C. In the GBA-3L while loop, the available free servers are sufficient to fulfill the requests at location A.

3.2. $A_n + F_n < I_{\overrightarrow{AB}_n}$.

$(G_{\overrightarrow{AB}_n}, G_{\overrightarrow{BC}_n}, G_{\overrightarrow{CA}_n}) = (A_n + F_n, I_{\overrightarrow{BC}_n}, I_{\overrightarrow{CA}_n})$

doi:10.6342/NTU202302667

In this case, the initial servers can fulfill the requests at locations B and C. In the GBA-3L while loop, all available free servers will be allocated to location A.

4. $A_n < I_{\overrightarrow{AB}_n}, B_n < I_{\overrightarrow{BC}_n}$.

4.1. $A_n + B_n + F_n \geq I_{\overrightarrow{AB}_n} + I_{\overrightarrow{BC}_n}$.

$(G_{\overrightarrow{AB}_n}, G_{\overrightarrow{BC}_n}, G_{\overrightarrow{CA}_n}) = (I_{\overrightarrow{AB}_n}, I_{\overrightarrow{BC}_n}, I_{\overrightarrow{CA}_n})$

In this case, the initial servers can fulfill the requests at location C. In the GBA-3L while loop, the available free servers are sufficient to fulfill the requests at locations A and B.

4.2. $A_n + B_n + F_n < I_{\overrightarrow{AB}_n} + I_{\overrightarrow{BC}_n}, A_n \geq B_n + F_n$. It should be noted that $A_n \geq \lfloor \frac{k}{3} \rfloor$.

4.2.1. $I_{\overrightarrow{BC}_n} \leq B_n + F_n$.

$(G_{\overrightarrow{AB}_n}, G_{\overrightarrow{BC}_n}, G_{\overrightarrow{CA}_n}) = (k - I_{\overrightarrow{BC}_n} - I_{\overrightarrow{CA}_n}, I_{\overrightarrow{BC}_n}, I_{\overrightarrow{CA}_n})$

In this case, the initial servers can fulfill the requests at location C. In the GBA-3L while loop, the requests at location B will be fulfilled first. Afterwards, the remaining available free servers will be allocated to location A.

4.2.2. $I_{\overrightarrow{BC}_n} > B_n + F_n$.

$(G_{\overrightarrow{AB}_n}, G_{\overrightarrow{BC}_n}, G_{\overrightarrow{CA}_n}) = (A_n, B_n + F_n, I_{\overrightarrow{CA}_n})$

In this case, the initial servers can fulfill the requests at location C. In the GBA-3L while loop, due to the condition $A_n \geq B_n + F_n$, all available free servers will be allocated to location B.

4.3. $A_n + B_n + F_n < I_{\overrightarrow{AB}_n} + I_{\overrightarrow{BC}_n}, A_n < B_n + F_n$. It should be noted that

doi:10.6342/NTU202302667

$\left\lceil \frac{k-C_n}{2} \right\rceil, \left\lfloor \frac{k-C_n}{2} \right\rfloor \geq \left\lfloor \frac{k}{3} \right\rfloor$.

4.3.1. $I_{\overrightarrow{AB}_n} > \left\lceil \frac{k-C_n}{2} \right\rceil, I_{\overrightarrow{BC}_n} > \left\lfloor \frac{k-C_n}{2} \right\rfloor$.

$(G_{\overrightarrow{AB}_n}, G_{\overrightarrow{BC}_n}, G_{\overrightarrow{CA}_n}) = (\left\lceil \frac{k-C_n}{2} \right\rceil, \left\lfloor \frac{k-C_n}{2} \right\rfloor, I_{\overrightarrow{CA}_n})$

In this case, the initial servers can fulfill the requests at location C. In the GBA-3L while loop, all free servers will be allocated to A and B, with the server numbers at these two locations being approximately equal.

4.3.2. $I_{\overrightarrow{AB}_n} > \left\lceil \frac{k-C_n}{2} \right\rceil, I_{\overrightarrow{BC}_n} \leq \left\lfloor \frac{k-C_n}{2} \right\rfloor$.

$(G_{\overrightarrow{AB}_n}, G_{\overrightarrow{BC}_n}, G_{\overrightarrow{CA}_n}) = (k - C_n - I_{\overrightarrow{BC}_n}, I_{\overrightarrow{BC}_n}, I_{\overrightarrow{CA}_n})$

In this case, the initial servers can fulfill the requests at location C. In the GBA-3L while loop, the requests at location B will be fulfilled first, and then the remaining available free servers will be allocated to location A. It should be noted that $k - C_n - I_{\overrightarrow{BC}_n} \geq \left\lfloor \frac{k}{3} \right\rfloor$.

4.3.3. $I_{\overrightarrow{AB}_n} \leq \left\lceil \frac{k-C_n}{2} \right\rceil, I_{\overrightarrow{BC}_n} > \left\lfloor \frac{k-C_n}{2} \right\rfloor$.

$(G_{\overrightarrow{AB}_n}, G_{\overrightarrow{BC}_n}, G_{\overrightarrow{CA}_n}) = (I_{\overrightarrow{AB}_n}, k - C_n - I_{\overrightarrow{AB}_n}, I_{\overrightarrow{CA}_n})$

In this case, the initial servers can fulfill the requests at location C. In the GBA-3L while loop, the requests at location A will be fulfilled first, and then the remaining available free servers will be allocated to location B. It should be noted that $k - C_n - I_{\overrightarrow{AB}_n} \geq \left\lfloor \frac{k}{3} \right\rfloor$.

4.3.4. $I_{\overrightarrow{AB}_n} \leq \left\lceil \frac{k-C_n}{2} \right\rceil, I_{\overrightarrow{BC}_n} \leq \left\lfloor \frac{k-C_n}{2} \right\rfloor$. It will not happen.

Here, we will prove that $S(n)$ holds from the previous induction step $S(n-1)$ for all the cases mentioned above.

Observe that $G_{\overrightarrow{AB}_n}$ is equal to one of the three: $\{I_{\overrightarrow{AB}_n}, G_{\overrightarrow{CA}_{n-1}} + G_{\overrightarrow{f}_{n-1}}, \geq \left\lfloor \frac{k}{3} \right\rfloor\}$, $G_{\overrightarrow{BC}_n}$ is equal to one of the three: $\{I_{\overrightarrow{BC}_n}, G_{\overrightarrow{AB}_{n-1}} + G_{\overrightarrow{f}_{n-1}}, \geq \left\lfloor \frac{k}{3} \right\rfloor\}$ and $G_{\overrightarrow{CA}_n}$ is equal to

doi:10.6342/NTU202302667

$I_{\overrightarrow{CA}_n}$. Due to $\lfloor \frac{k-C_n}{2} \rfloor \geq \lfloor \frac{k}{3} \rfloor$, if not all requests at location A are fulfilled, $G_{\overrightarrow{AB}_n}$ must be greater than or equal to $\lfloor \frac{k}{3} \rfloor$, and if not all requests at location B are fulfilled, $G_{\overrightarrow{BC}_n}$ must be greater than or equal to $\lfloor \frac{k}{3} \rfloor$.

Here, we start to prove lemma 2 to lemma 4. In order to demonstrate that lemma 2 to lemma 4 hold in $S(n)$ given $S(n-1)$ for all cases, we will first prove that lemma 3 is true for all cases. The proofs of lemma 2 and lemma 4 are similar and will be omitted.

Proof. When $G_{\overrightarrow{AB}_n} = I_{\overrightarrow{AB}_n}$,

$$\begin{aligned} G_{n,B} &= G_{n-1} + G_{\overrightarrow{AB}_n} + k \\ &\geq \delta O_{n-1} + I_{\overrightarrow{AB}_n} + k \\ &\geq \delta O_{n-1} + \delta O_{\overrightarrow{AB}_n} + \delta k \\ &= \delta(O_{n-1} + O_{\overrightarrow{AB}_n} + k) \\ &= \delta O_{n,B} \end{aligned} \tag{3.7}$$

In Equation 3.7, the first inequality follows from the fact that $G_{\overrightarrow{AB}_n} = I_{\overrightarrow{AB}_n}$, and lemma 1 in the inductive hypothesis $S(n-1)$. The second inequality follows from lemma 2 in the inductive hypothesis $S(n-1)$ and the fact that $\delta \leq 1$.

doi:10.6342/NTU202302667

When $G_{\overrightarrow{AB}_n} \geq \lfloor \frac{k}{3} \rfloor$,

$$
\begin{aligned}
G_{n,B} &= G_{n-1} + G_{\overrightarrow{AB}_n} + k \\
&\geq \delta O_{n-1} + \left\lfloor \frac{k}{3} \right\rfloor + k \\
&= \delta O_{n-1} + \delta(k+k) \\
&\geq \delta(O_{n-1} + O_{\overrightarrow{AB}_n} + k) \\
&= \delta O_{n,B}
\end{aligned}
\tag{3.8}
$$

Assuming that $\delta = \frac{k + \lfloor k/3 \rfloor}{2k}$. In equation 3.8, the first inequality follows from the fact that $G_{\overrightarrow{AB}_n} \geq \lfloor \frac{k}{3} \rfloor$, and lemma 1 in the inductive hypothesis $S(n-1)$. The second inequality follows from the fact that $k \geq O_{\overrightarrow{AB}_n}$.

When $G_{\overrightarrow{AB}_n} = G_{\overrightarrow{CA}_{n-1}} + G_{\overrightarrow{f}_{n-1}}$,

$$
\begin{aligned}
G_{n,B} &= G_{n-1} + G_{\overrightarrow{AB}_n} + k \\
&\geq (G_{n-1} + G_{\overrightarrow{CA}_{n-1}} + G_{\overrightarrow{f}_{n-1}}) + \delta k \\
&\geq \delta(O_{n-1} + O_{\overrightarrow{CA}_{n-1}} + O_{\overrightarrow{f}_{n-1}}) + \delta k \\
&= \delta(O_{n-1} + O_{\overrightarrow{CA}_{n-1}} + O_{\overrightarrow{f}_{n-1}} + k) \\
&= \delta O_{n,B}
\end{aligned}
\tag{3.9}
$$

In equation 3.9, the first inequality follows from the fact that $G_{\overrightarrow{AB}_n} = G_{\overrightarrow{CA}_{n-1}} + G_{\overrightarrow{f}_{n-1}}$ and the fact that $\delta \leq 1$. The second inequality follows from lemma 2 in the inductive hypothesis $S(n-1)$. □

Next, we will proceed to prove lemma 5 to lemma 7. In order to demonstrate that lemma 5 to lemma 7 hold in $S(n)$ given $S(n-1)$ for all cases, we will symmetrically

doi:10.6342/NTU202302667

prove that lemma 6 is true for all cases. The proofs of lemma 5 and lemma 7 are similar and will be omitted.

Proof. When $G_{\overrightarrow{AB}_n} = I_{\overrightarrow{AB}_n}$ and $G_{\overrightarrow{BC}_n} = I_{\overrightarrow{BC}_n}$,

$$\begin{aligned} G_{n,BC} &= G_{n-1} + G_{\overrightarrow{AB}_n} + G_{\overrightarrow{BC}_n} + k \\ &\geq \delta O_{n-1} + I_{\overrightarrow{AB}_n} + I_{\overrightarrow{BC}_n} + k \\ &\geq \delta O_{n-1} + \delta O_{\overrightarrow{AB}_n} + \delta O_{\overrightarrow{BC}_n} + \delta k \\ &= \delta(O_{n-1} + O_{\overrightarrow{AB}_n} + O_{\overrightarrow{BC}_n} + k) \\ &= \delta O_{n,BC} \end{aligned} \tag{3.10}$$

In equation 3.10, the first inequality follows from the fact that $G_{\overrightarrow{AB}_n} = I_{\overrightarrow{AB}_n}, G_{\overrightarrow{BC}_n} = I_{\overrightarrow{BC}_n}$ and lemma 1 in the inductive hypothesis $S(n-1)$. The second inequality follows from lemma 8, lemma 9 in the inductive hypothesis S(n - 1) and the fact that $\delta \leq 1$.

When $G_{\overrightarrow{AB}_n} = I_{\overrightarrow{AB}_n}$ and $G_{\overrightarrow{BC}_n} = G_{\overrightarrow{AB}_{n-1}} + G_{\overrightarrow{f}_{n-1}}$,

$$\begin{aligned} G_{n,BC} &= G_{n-1} + G_{\overrightarrow{AB}_n} + G_{\overrightarrow{BC}_n} + k \\ &\geq (G_{n-1} + G_{\overrightarrow{AB}_{n-1}} + G_{\overrightarrow{f}_{n-1}}) + I_{\overrightarrow{AB}_n} + \delta k \\ &\geq \delta(O_{n-1} + O_{\overrightarrow{AB}_{n-1}} + O_{\overrightarrow{f}_{n-1}}) + \delta O_{\overrightarrow{AB}_n} + \delta k \\ &\geq \delta(O_{n-1} + O_{\overrightarrow{BC}_n}) + \delta O_{\overrightarrow{AB}_n} + \delta k \\ &= \delta(O_{n-1} + O_{\overrightarrow{AB}_n} + O_{\overrightarrow{BC}_n} + k) \\ &= \delta O_{n,BC} \end{aligned} \tag{3.11}$$

In equation 3.11, the first inequality follows from the fact that $G_{\overrightarrow{AB}_n} = I_{\overrightarrow{AB}_n}, G_{\overrightarrow{BC}_n} = G_{\overrightarrow{AB}_{n-1}} + G_{\overrightarrow{f}_{n-1}}$ and $\delta \leq 1$. The second inequality follows from lemma 3 and lemma 8 in the inductive hypothesis $S(n-1)$. The third inequality follows from lemma 9 in the

doi:10.6342/NTU202302667

inductive hypothesis $S(n-1)$.

When $G_{\overrightarrow{AB}_n} \geq \lfloor \frac{k}{3} \rfloor$,

$$\begin{aligned} G_{n,BC} &= G_{n-1} + G_{\overrightarrow{AB}_n} + k \\ &\geq \delta O_{n-1} + \left\lfloor \frac{k}{3} \right\rfloor + k \\ &= \delta O_{n-1} + \delta(k+k) \\ &\geq \delta(O_{n-1} + O_{\overrightarrow{AB}_n} + O_{\overrightarrow{BC}_n} + k) \\ &= \delta O_{n,BC} \end{aligned} \tag{3.12}$$

Assuming that $\delta = \frac{k+\lfloor k/3 \rfloor}{2k}$. In equation 3.12, the first inequality follows from the fact that $G_{\overrightarrow{AB}_n} \geq \lfloor \frac{k}{3} \rfloor$ and lemma 1 in the inductive hypothesis S(n - 1). The second inequality follows from the fact that $k \geq O_{\overrightarrow{AB}_n} + O_{\overrightarrow{BC}_n}$. □

Finally, given the inductive hypothesis $S(n-1)$, we prove that $S(n)$ holds when the initial servers can fulfill the requests at location C.

3.1.2 The initial servers cannot fulfill its requests at location C.

In this section, we will list all cases when the initial local servers cannot fulfill the requests at location C, i.e., $C_n < I_{\overrightarrow{CA}_n}$, and prove that $S(n)$ holds based on the previous induction step $S(n-1)$ for all cases.

1. $A_n \geq I_{\overrightarrow{AB}_n}$, $B_n \geq I_{\overrightarrow{BC}_n}$.

   1.1. $C_n + F_n \geq I_{\overrightarrow{CA}_n}$.

   $(G_{\overrightarrow{AB}_n}, G_{\overrightarrow{BC}_n}, G_{\overrightarrow{CA}_n}) = (I_{\overrightarrow{AB}_n}, I_{\overrightarrow{BC}_n}, I_{\overrightarrow{CA}_n})$

doi:10.6342/NTU202302667

In this case, the initial servers can fulfill its requests at location A and B. In the GBA-3L while loop, the free servers are sufficient to fulfill the requests at location C.

1.2. $C_n + F_n < I_{\overrightarrow{CA}_n}$.

$(G_{\overrightarrow{AB}_n}, G_{\overrightarrow{BC}_n}, G_{\overrightarrow{CA}_n}) = (I_{\overrightarrow{AB}_n}, I_{\overrightarrow{BC}_n}, C_n + F_n)$

In this case, the initial servers can fulfill its requests at location A and B. In the GBA-3L while loop, all free servers will be allocated to location C.

2. $A_n \geq I_{\overrightarrow{AB}_n}, B_n < I_{\overrightarrow{BC}_n}$.

2.1. $B_n + C_n + F_n \geq I_{\overrightarrow{BC}_n} + I_{\overrightarrow{CA}_n}$.

$(G_{\overrightarrow{AB}_n}, G_{\overrightarrow{BC}_n}, G_{\overrightarrow{CA}_n}) = (I_{\overrightarrow{AB}_n}, I_{\overrightarrow{BC}_n}, I_{\overrightarrow{CA}_n})$

In this case, the initial servers can fulfill its requests at location A. In the GBA-3L while loop, the free servers are sufficient to fulfill the requests at location B and C.

2.2. $B_n + C_n + F_n < I_{\overrightarrow{BC}_n} + I_{\overrightarrow{CA}_n}, B_n \geq C_n + F_n$. It should be noted that $C_n + F_n \leq \lfloor \frac{k}{3} \rfloor$.

2.2.1. $I_{\overrightarrow{CA}_n} \geq C_n + F_n$.

$(G_{\overrightarrow{AB}_n}, G_{\overrightarrow{BC}_n}, G_{\overrightarrow{CA}_n}) = (I_{\overrightarrow{AB}_n}, B_n, C_n + F_n)$

In this case, the initial servers can fulfill the requests at location A. In the GBA-3L while loop, due to $B_n \geq C_n + F_n$, all free servers will be allocated to location C. It should be noted that $G_{\overrightarrow{f}_{n,exclB}} \leq \lfloor \frac{k}{3} \rfloor$.

2.2.2. $I_{\overrightarrow{CA}_n} < C_n + F_n$.

$(G_{\overrightarrow{AB}_n}, G_{\overrightarrow{BC}_n}, G_{\overrightarrow{CA}_n}) = (I_{\overrightarrow{AB}_n}, k - A_n - I_{\overrightarrow{CA}_n}, I_{\overrightarrow{CA}_n})$

In this case, the initial servers can fulfill the requests at location A. In the

doi:10.6342/NTU202302667

GBA-3L while loop, the requests at location C will be fulfilled first, and then the remaining available free servers will be allocated to location B. It should be noted that $G_{\overrightarrow{f}_{n,exclB}} \leq \lfloor \frac{k}{3} \rfloor$.

2.3. $B_n + C_n + F_n < I_{\overrightarrow{BC}_n} + I_{\overrightarrow{CA}_n}, B_n < C_n + F_n$.

2.3.1. $I_{\overrightarrow{BC}_n} > \lceil \frac{k-A_n}{2} \rceil, I_{\overrightarrow{CA}_n} > \lfloor \frac{k-A_n}{2} \rfloor$.

$(G_{\overrightarrow{AB}_n}, G_{\overrightarrow{BC}_n}, G_{\overrightarrow{CA}_n}) = (I_{\overrightarrow{AB}_n}, \lceil \frac{k-A_n}{2} \rceil, \lfloor \frac{k-A_n}{2} \rfloor)$

In this case, initial servers can fulfill requests at location A. In the GBA-3L while loop, all free servers will be allocated to location B and C, with the server numbers at these two locations being approximately equal. Furthermore, we can split it into two cases as follows.

If $A_n > \lfloor \frac{k}{3} \rfloor$, this implies $\lceil \frac{k-A_n}{2} \rceil$ and $\lfloor \frac{k-A_n}{2} \rfloor$ are both less than or equal to $\lfloor \frac{k}{3} \rfloor$, which further implies that $G_{\overrightarrow{f}_{n,exclB}}$ and $G_{\overrightarrow{f}_{n,exclC}}$ are both less than or equal to $\lfloor \frac{k}{3} \rfloor$.

If $A_n \leq \lfloor \frac{k}{3} \rfloor$, this implies $\lceil \frac{k-A_n}{2} \rceil$ and $\lfloor \frac{k-A_n}{2} \rfloor$ are both greater than or equal to $\lfloor \frac{k}{3} \rfloor$.

2.3.2. $I_{\overrightarrow{BC}_n} > \lceil \frac{k-A_n}{2} \rceil, I_{\overrightarrow{CA}_n} \leq \lfloor \frac{k-A_n}{2} \rfloor$.

$(G_{\overrightarrow{AB}_n}, G_{\overrightarrow{BC}_n}, G_{\overrightarrow{CA}_n}) = (I_{\overrightarrow{AB}_n}, k - A_n - I_{\overrightarrow{CA}_n}, I_{\overrightarrow{CA}_n})$

In this case, initial servers can fulfill requests at location A. In the GBA-3L while loop, the requests at location C will be fulfilled first, then the remaining free servers will be allocated to location B. Furthermore, we can split it into two cases as follows.

If $k - A_n - I_{\overrightarrow{CA}_n} \geq \lfloor \frac{k}{3} \rfloor$.

If $k - A_n - I_{\overrightarrow{CA}_n} < \lfloor \frac{k}{3} \rfloor$, this implies $I_{\overrightarrow{CA}_n} < \lfloor \frac{k}{3} \rfloor$, which further implies $G_{\overrightarrow{f}_{n,exclB}} < \lfloor \frac{k}{3} \rfloor$.

doi:10.6342/NTU202302667

2.3.3. $I_{\overrightarrow{BC}_n} \leq \left\lceil \frac{k-A_n}{2} \right\rceil, I_{\overrightarrow{CA}_n} > \left\lfloor \frac{k-A_n}{2} \right\rfloor$.

$(G_{\overrightarrow{AB}_n}, G_{\overrightarrow{BC}_n}, G_{\overrightarrow{CA}_n}) = (I_{\overrightarrow{AB}_n}, I_{\overrightarrow{BC}_n}, k - A_n - I_{\overrightarrow{BC}_n})$

In this case, the initial servers can fulfill the requests at location A. In the GBA-3L while loop, the requests at location B will be fulfilled first, and any remaining free servers will be allocated to location C.

If $k - A_n - I_{\overrightarrow{BC}_n} \geq \left\lfloor \frac{k}{3} \right\rfloor$.

If $k - A_n - I_{\overrightarrow{BC}_n} < \left\lfloor \frac{k}{3} \right\rfloor$, this implies $I_{\overrightarrow{BC}_n} < \left\lfloor \frac{k}{3} \right\rfloor$, which further implies $G_{\overrightarrow{f}_{n,exclC}} < \left\lfloor \frac{k}{3} \right\rfloor$.

2.3.4. $I_{\overrightarrow{BC}_n} \leq \left\lceil \frac{k-A_n}{2} \right\rceil, I_{\overrightarrow{CA}_n} \leq \left\lfloor \frac{k-A_n}{2} \right\rfloor$. It will not happen.

3. $A_n < I_{\overrightarrow{AB}_n}, B_n \geq I_{\overrightarrow{BC}_n}$.

3.1. $A_n + C_n + F_n \geq I_{\overrightarrow{AB}_n} + I_{\overrightarrow{CA}_n}$.

$(G_{\overrightarrow{AB}_n}, G_{\overrightarrow{BC}_n}, G_{\overrightarrow{CA}_n}) = (I_{\overrightarrow{AB}_n}, I_{\overrightarrow{BC}_n}, I_{\overrightarrow{CA}_n})$

In this case, the initial servers can fulfill the requests at location B. In the GBA-3L while loop, there are sufficient free servers to fulfill the requests at locations A and C.

3.2. $A_n + C_n + F_n < I_{\overrightarrow{AB}_n} + I_{\overrightarrow{CA}_n}$, $A_n \geq C_n + F_n$. It should be noted that $A_n \geq \left\lfloor \frac{k}{3} \right\rfloor$.

3.2.1. $I_{\overrightarrow{CA}_n} \geq C_n + F_n$.

$(G_{\overrightarrow{AB}_n}, G_{\overrightarrow{BC}_n}, G_{\overrightarrow{CA}_n}) = (A_n, I_{\overrightarrow{BC}_n}, C_n + F_n)$

In this case, the initial servers can fulfill the requests at location B. In the GBA-3L while loop, due to the condition $A_n \geq C_n + F_n$, all free servers will be allocated to location C.

3.2.2. $I_{\overrightarrow{CA}_n} < C_n + F_n$.

$(G_{\overrightarrow{AB}_n}, G_{\overrightarrow{BC}_n}, G_{\overrightarrow{CA}_n}) = (k - I_{\overrightarrow{CA}_n} - B_n, I_{\overrightarrow{BC}_n}, I_{\overrightarrow{CA}_n})$

doi:10.6342/NTU202302667

In this case, the initial servers can fulfill the requests at location B. In the GBA-3L while loop, the requests at location C will be fulfilled first, and any remaining free servers will be allocated to location A. It should be noted that $k - I_{\overrightarrow{CA}_n} - B_n \geq \left\lfloor \frac{k}{3} \right\rfloor$.

3.3. $A_n + C_n + F_n < I_{\overrightarrow{AB}_n} + I_{\overrightarrow{CA}_n}$, $A_n < C_n + F_n$. It should be note that $B_n \leq \left\lfloor \frac{k}{3} \right\rfloor$.

3.3.1. $I_{\overrightarrow{AB}_n} > \left\lceil \frac{k-B_n}{2} \right\rceil$, $I_{\overrightarrow{CA}_n} > \left\lfloor \frac{k-B_n}{2} \right\rfloor$.

$(G_{\overrightarrow{AB}_n}, G_{\overrightarrow{BC}_n}, G_{\overrightarrow{CA}_n}) = (\left\lceil \frac{k-B_n}{2} \right\rceil, I_{\overrightarrow{BC}_n}, \left\lfloor \frac{k-B_n}{2} \right\rfloor)$

In this case, the initial servers can fulfill the requests at location B. In the GBA-3L while loop, the free servers will be allocated to locations A and C, with the server numbers at these two locations being approximately equal. It should be note that $\left\lceil \frac{k-B_n}{2} \right\rceil$ and $\left\lfloor \frac{k-B_n}{2} \right\rfloor$ are both greater than or equal to $\left\lfloor \frac{k}{3} \right\rfloor$.

3.3.2. $I_{\overrightarrow{AB}_n} > \left\lceil \frac{k-B_n}{2} \right\rceil$, $I_{\overrightarrow{CA}_n} \leq \left\lfloor \frac{k-B_n}{2} \right\rfloor$.

$(G_{\overrightarrow{AB}_n}, G_{\overrightarrow{BC}_n}, G_{\overrightarrow{CA}_n}) = (k - B_n - I_{\overrightarrow{CA}_n}, I_{\overrightarrow{BC}_n}, I_{\overrightarrow{CA}_n})$

In this case, the initial servers can fulfill the requests at location B. In the GBA-3L while loop, the requests at location C will be fulfilled first, and any remaining free servers will be allocated to location A. It should be noted that $k - B_n - I_{\overrightarrow{CA}_n} \geq \left\lfloor \frac{k}{3} \right\rfloor$.

3.3.3. $I_{\overrightarrow{AB}_n} \leq \left\lceil \frac{k-B_n}{2} \right\rceil$, $I_{\overrightarrow{CA}_n} > \left\lfloor \frac{k-B_n}{2} \right\rfloor$.

$(G_{\overrightarrow{AB}_n}, G_{\overrightarrow{BC}_n}, G_{\overrightarrow{CA}_n}) = (I_{\overrightarrow{AB}_n}, I_{\overrightarrow{BC}_n}, k - B_n - I_{\overrightarrow{AB}_n})$

In this case, the initial servers can fulfill the requests at location B. In the GBA-3L while loop, the requests at location A will be fulfilled first, and any remaining free servers will be allocated to location C. It should be noted that $k - B_n - I_{\overrightarrow{AB}_n} \geq \left\lfloor \frac{k}{3} \right\rfloor$.

doi:10.6342/NTU202302667

3.3.4. $I_{\overrightarrow{AB}_n} \leq \left\lceil \frac{k-B_n}{2} \right\rceil, I_{\overrightarrow{CA}_n} \leq \left\lfloor \frac{k-B_n}{2} \right\rfloor$. It will not happen.

4. $A_n < I_{\overrightarrow{AB}_n}, B_n < I_{\overrightarrow{BC}_n}$.

4.1. $A_n + B_n + C_n + F_n \geq I_{\overrightarrow{AB}_n} + I_{\overrightarrow{BC}_n} + I_{\overrightarrow{CA}_n}$.

$(G_{\overrightarrow{AB}_n}, G_{\overrightarrow{BC}_n}, G_{\overrightarrow{CA}_n}) = (I_{\overrightarrow{AB}_n}, I_{\overrightarrow{BC}_n}, I_{\overrightarrow{CA}_n})$

In this case, the free servers can fulfill the requests at locations A, B, and C in the GBA-3L while loop.

4.2. $A_n + B_n + C_n + F_n < I_{\overrightarrow{AB}_n} + I_{\overrightarrow{BC}_n} + I_{\overrightarrow{CA}_n}, B_n \geq C_n + F_n$. It should be noted that $A_n \geq \left\lfloor \frac{k}{3} \right\rfloor, C_n + F_n \leq \left\lfloor \frac{k}{3} \right\rfloor$.

4.2.1. $I_{\overrightarrow{CA}_n} \geq C_n + F_n$.

$(G_{\overrightarrow{AB}_n}, G_{\overrightarrow{BC}_n}, G_{\overrightarrow{CA}_n}) = (A_n, B_n, C_n + F_n)$

In this case, due to the condition $B_n \geq C_n + F_n$, all free servers will be allocated to location C in the GBA-3L while loop. It should be noted that $G_{\overrightarrow{f}_{n,exclB}} \leq \left\lfloor \frac{k}{3} \right\rfloor$.

4.2.2. $I_{\overrightarrow{CA}_n} < C_n + F_n, A_n \geq B_n + (G_{\overrightarrow{f}_{n-1}} - I_{\overrightarrow{CA}_n} + C_n)$.

4.2.2.1. $I_{\overrightarrow{BC}_n} > B_n + (G_{\overrightarrow{f}_{n-1}} - I_{\overrightarrow{CA}_n} + C_n)$.

$(G_{\overrightarrow{AB}_n}, G_{\overrightarrow{BC}_n}, G_{\overrightarrow{CA}_n}) = (A_n, k - A_n - I_{\overrightarrow{CA}_n}, I_{\overrightarrow{CA}_n})$

In this case, the requests at location C will be fulfilled first. Because the initial servers at location A are greater than or equal to the remaining free servers and the initial servers at location B. After fulfilling the requests at location C, GBA-3L will assign all the remaining servers to location B. It should be noted that $G_{\overrightarrow{f}_{n,exclB}} \leq \left\lfloor \frac{k}{3} \right\rfloor$.

4.2.2.2. $I_{\overrightarrow{BC}_n} \leq B_n + (G_{\overrightarrow{f}_{n-1}} - I_{\overrightarrow{CA}_n} + C_n)$.

$(G_{\overrightarrow{AB}_n}, G_{\overrightarrow{BC}_n}, G_{\overrightarrow{CA}_n}) = (k - I_{\overrightarrow{BC}_n} - I_{\overrightarrow{CA}_n}, I_{\overrightarrow{BC}_n}, I_{\overrightarrow{CA}_n})$

doi:10.6342/NTU202302667

In this case, the requests at location C will be fulfilled first, followed by the requests at location B. Any remaining servers will then be allocated to location A. It should be noted that $k - I_{\overrightarrow{BC}_n} - I_{\overrightarrow{CA}_n} \geq \left\lfloor \frac{k}{3} \right\rfloor$.

4.2.3. $I_{\overrightarrow{CA}_n} < C_n + F_n, A_n < B_n + (G_{\overrightarrow{f}_{n-1}} - I_{\overrightarrow{CA}_n} + C_n)$.

4.2.3.1. $I_{\overrightarrow{AB}_n} > \left\lceil \frac{k - I_{\overrightarrow{CA}_n}}{2} \right\rceil, I_{\overrightarrow{BC}_n} > \left\lfloor \frac{k - I_{\overrightarrow{CA}_n}}{2} \right\rfloor$.

$(G_{\overrightarrow{AB}_n}, G_{\overrightarrow{BC}_n}, G_{\overrightarrow{CA}_n}) = (\left\lceil \frac{k - I_{\overrightarrow{CA}_n}}{2} \right\rceil, \left\lfloor \frac{k - I_{\overrightarrow{CA}_n}}{2} \right\rfloor, I_{\overrightarrow{CA}_n})$

In this case, the requests at location C will be fulfilled first, then GBA-3L will assign the remaining free servers to location A and B, with the server numbers at these two locations being approximately equal. It should be noted that $\left\lceil \frac{k - I_{\overrightarrow{CA}_n}}{2} \right\rceil, \left\lfloor \frac{k - I_{\overrightarrow{CA}_n}}{2} \right\rfloor \geq \left\lfloor \frac{k}{3} \right\rfloor$.

4.2.3.2. $I_{\overrightarrow{AB}_n} > \left\lceil \frac{k - I_{\overrightarrow{CA}_n}}{2} \right\rceil, I_{\overrightarrow{BC}_n} \leq \left\lfloor \frac{k - I_{\overrightarrow{CA}_n}}{2} \right\rfloor$.

$(G_{\overrightarrow{AB}_n}, G_{\overrightarrow{BC}_n}, G_{\overrightarrow{CA}_n}) = (k - I_{\overrightarrow{BC}_n} - I_{\overrightarrow{CA}_n}, I_{\overrightarrow{BC}_n}, I_{\overrightarrow{CA}_n})$

In this case, the requests at location C will be fulfilled first, followed by the requests at location B. Any remaining servers will then be allocated to location A. It should be noted that $k - I_{\overrightarrow{BC}_n} - I_{\overrightarrow{CA}_n} \geq \left\lfloor \frac{k}{3} \right\rfloor$.

4.2.3.3. $I_{\overrightarrow{AB}_n} \leq \left\lceil \frac{k - I_{\overrightarrow{CA}_n}}{2} \right\rceil, I_{\overrightarrow{BC}_n} > \left\lfloor \frac{k - I_{\overrightarrow{CA}_n}}{2} \right\rfloor$.

$(G_{\overrightarrow{AB}_n}, G_{\overrightarrow{BC}_n}, G_{\overrightarrow{CA}_n}) = (I_{\overrightarrow{AB}_n}, k - I_{\overrightarrow{AB}_n} - I_{\overrightarrow{CA}_n}, I_{\overrightarrow{CA}_n})$

In this case, the requests at location C will be fulfilled first, followed by the requests at location A. Any remaining servers will then be allocated to location B. It should be noted that $k - I_{\overrightarrow{AB}_n} - I_{\overrightarrow{CA}_n} \geq \left\lfloor \frac{k}{3} \right\rfloor$.

4.2.3.4. $I_{\overrightarrow{AB}_n} \leq \left\lceil \frac{k - I_{\overrightarrow{CA}_n}}{2} \right\rceil, I_{\overrightarrow{BC}_n} \leq \left\lfloor \frac{k - I_{\overrightarrow{CA}_n}}{2} \right\rfloor$. It will not happen.

4.3. $A_n + B_n + C_n + F_n < I_{\overrightarrow{AB}_n} + I_{\overrightarrow{BC}_n} + I_{\overrightarrow{CA}_n}, B_n < C_n + F_n$. It should be noted that $B_n \leq \left\lfloor \frac{k}{3} \right\rfloor$.

4.3.1. $A_n > \left\lfloor \frac{k}{3} \right\rfloor, I_{\overrightarrow{BC}_n} > \left\lceil \frac{k - A_n}{2} \right\rceil, I_{\overrightarrow{CA}_n} > \left\lfloor \frac{k - A_n}{2} \right\rfloor$.

doi:10.6342/NTU202302667

$(G_{\overrightarrow{AB}_n}, G_{\overrightarrow{BC}_n}, G_{\overrightarrow{CA}_n}) = (A_n, \lceil \frac{k-A_n}{2} \rceil, \lfloor \frac{k-A_n}{2} \rfloor)$

In this case, both the number of assigned servers at location B and the number of assigned servers at location C after the GBA-3L while loop will not exceed the number of initial servers at location A, and no free servers can be assigned to location A. Furthermore, we can split it into two cases as follows.

If $\lceil \frac{k-A_n}{2} \rceil > \lfloor \frac{k}{3} \rfloor$, this implies $\lfloor \frac{k-A_n}{2} \rfloor \geq \lfloor \frac{k}{3} \rfloor$.

If $\lceil \frac{k-A_n}{2} \rceil \leq \lfloor \frac{k}{3} \rfloor$, this implies $\lfloor \frac{k-A_n}{2} \rfloor \leq \lfloor \frac{k}{3} \rfloor$, which further implies $G_{\vec{f}_{n,exclB}}, G_{\vec{f}_{n,exclC}} \leq \lfloor \frac{k}{3} \rfloor$.

4.3.2. $A_n > \lfloor \frac{k}{3} \rfloor, I_{\overrightarrow{BC}_n} > \lceil \frac{k-A_n}{2} \rceil, I_{\overrightarrow{CA}_n} \leq \lfloor \frac{k-A_n}{2} \rfloor$.

4.3.2.1. $A_n \geq B_n+(G_{\vec{f}_{n-1}}-I_{\overrightarrow{CA}_n}+C_n), B_n+(G_{\vec{f}_{n-1}}-I_{\overrightarrow{CA}_n}+C_n) \geq I_{\overrightarrow{BC}_n}$.

$(G_{\overrightarrow{AB}_n}, G_{\overrightarrow{BC}_n}, G_{\overrightarrow{CA}_n}) = (k - I_{\overrightarrow{BC}_n} - I_{\overrightarrow{CA}_n}, I_{\overrightarrow{BC}_n}, I_{\overrightarrow{CA}_n})$

In this case, the requests at location C will be fulfilled first in the GBA-3L while loop. Then, due to the condition $A_n \geq B_n+(G_{\vec{f}_{n-1}} - I_{\overrightarrow{CA}_n} + C_n)$, GBA-3L will allocate free servers to location B until fulfilling $I_{\overrightarrow{BC}_n}$. Finally, the remaining servers after fulfilling $I_{\overrightarrow{BC}_n}$ and $I_{\overrightarrow{CA}_n}$ will be assigned to location A. It is noted that $k - I_{\overrightarrow{BC}_n} - I_{\overrightarrow{CA}_n} \geq \lfloor \frac{k}{3} \rfloor$.

4.3.2.2. $A_n \geq B_n+(G_{\vec{f}_{n-1}}-I_{\overrightarrow{CA}_n}+C_n), B_n+(G_{\vec{f}_{n-1}}-I_{\overrightarrow{CA}_n}+C_n) < I_{\overrightarrow{BC}_n}$.

$(G_{\overrightarrow{AB}_n}, G_{\overrightarrow{BC}_n}, G_{\overrightarrow{CA}_n}) = (A_n, k - A_n - I_{\overrightarrow{CA}_n}, I_{\overrightarrow{CA}_n})$

In this case, the requests at location C will be fulfilled first in the GBA-3L while loop. Then, due to the condition $A_n \geq B_n+(G_{\vec{f}_{n-1}} - I_{\overrightarrow{CA}_n} + C_n)$, all remaining free servers after fulfilling $I_{\overrightarrow{CA}_n}$ will be allocated to location B. Furthermore, we can split it into two cases as

doi:10.6342/NTU202302667

follows.

If $k - A_n - I_{\overrightarrow{CA}_n} \geq \left\lfloor \frac{k}{3} \right\rfloor$.

If $k - A_n - I_{\overrightarrow{CA}_n} < \left\lfloor \frac{k}{3} \right\rfloor$, this implies $I_{\overrightarrow{CA}_n} < \left\lfloor \frac{k}{3} \right\rfloor$, which further implies $G_{\overrightarrow{f}_{n,exclB}} < \left\lfloor \frac{k}{3} \right\rfloor$.

4.3.2.3. $A_n < B_n + (G_{\overrightarrow{f}_{n-1}} - I_{\overrightarrow{CA}_n} + C_n), I_{\overrightarrow{AB}_n} > \left\lceil \frac{k - I_{\overrightarrow{CA}_n}}{2} \right\rceil, I_{\overrightarrow{BC}_n} > \left\lfloor \frac{k - I_{\overrightarrow{CA}_n}}{2} \right\rfloor$.

$(G_{\overrightarrow{AB}_n}, G_{\overrightarrow{BC}_n}, G_{\overrightarrow{CA}_n}) = (\left\lceil \frac{k - I_{\overrightarrow{CA}_n}}{2} \right\rceil, \left\lfloor \frac{k - I_{\overrightarrow{CA}_n}}{2} \right\rfloor, I_{\overrightarrow{CA}_n})$

In this case, the requests at location C will be fulfilled first in the GBA-3L while loop. Then, due to the condition $A_n < B_n + (G_{\overrightarrow{f}_{n-1}} - I_{\overrightarrow{CA}_n} + C_n)$, the remaining free servers after fulfilling $I_{\overrightarrow{CA}_n}$ will be allocated to both location A and B until the number of servers at the two locations is approximately equal. It should be noted that $\left\lceil \frac{k - I_{\overrightarrow{CA}_n}}{2} \right\rceil, \left\lfloor \frac{k - I_{\overrightarrow{CA}_n}}{2} \right\rfloor \geq \left\lfloor \frac{k}{3} \right\rfloor$.

4.3.2.4. $A_n < B_n + (G_{\overrightarrow{f}_{n-1}} - I_{\overrightarrow{CA}_n} + C_n), I_{\overrightarrow{AB}_n} > \left\lceil \frac{k - I_{\overrightarrow{CA}_n}}{2} \right\rceil, I_{\overrightarrow{BC}_n} \leq \left\lfloor \frac{k - I_{\overrightarrow{CA}_n}}{2} \right\rfloor$.

$(G_{\overrightarrow{AB}_n}, G_{\overrightarrow{BC}_n}, G_{\overrightarrow{CA}_n}) = (k - I_{\overrightarrow{BC}_n} - I_{\overrightarrow{CA}_n}, I_{\overrightarrow{BC}_n}, I_{\overrightarrow{CA}_n})$

In this case, the requests at location C will be fulfilled first in the GBA-3L while loop, followed by the requests at location B. The remaining free servers after fulfilling $I_{\overrightarrow{BC}_n}$ and $I_{\overrightarrow{CA}_n}$ will be allocated to location A. It should be noted that $k - I_{\overrightarrow{BC}_n} - I_{\overrightarrow{CA}_n} > \left\lfloor \frac{k}{3} \right\rfloor$.

4.3.2.5. $A_n < B_n + (G_{\overrightarrow{f}_{n-1}} - I_{\overrightarrow{CA}_n} + C_n), I_{\overrightarrow{AB}_n} \leq \left\lceil \frac{k - I_{\overrightarrow{CA}_n}}{2} \right\rceil, I_{\overrightarrow{BC}_n} > \left\lfloor \frac{k - I_{\overrightarrow{CA}_n}}{2} \right\rfloor$.

$(G_{\overrightarrow{AB}_n}, G_{\overrightarrow{BC}_n}, G_{\overrightarrow{CA}_n}) = (I_{\overrightarrow{AB}_n}, k - I_{\overrightarrow{AB}_n} - I_{\overrightarrow{CA}_n}, I_{\overrightarrow{CA}_n})$

In this case, the requests at location C will be fulfilled first in the

doi:10.6342/NTU202302667

GBA-3L while loop, followed by the requests at location A. The remaining free servers after fulfilling $I_{\overrightarrow{AB}_n}$ and $I_{\overrightarrow{CA}_n}$ will be allocated to location B. It should be noted that $k - I_{\overrightarrow{AB}_n} - I_{\overrightarrow{CA}_n} \geq \left\lfloor \frac{k}{3} \right\rfloor$.

4.3.2.6. $A_n < B_n + (G_{\overrightarrow{f}_{n-1}} - I_{\overrightarrow{CA}_n} + C_n), I_{\overrightarrow{AB}_n} \leq \left\lceil \frac{k - I_{\overrightarrow{CA}_n}}{2} \right\rceil, I_{\overrightarrow{BC}_n} \leq \left\lfloor \frac{k - I_{\overrightarrow{CA}_n}}{2} \right\rfloor$. It will not happen.

4.3.3. $A_n > \left\lfloor \frac{k}{3} \right\rfloor, I_{\overrightarrow{BC}_n} \leq \left\lceil \frac{k - A_n}{2} \right\rceil, I_{\overrightarrow{CA}_n} > \left\lfloor \frac{k - A_n}{2} \right\rfloor$.

4.3.3.1. $A_n \geq C_n + (G_{\overrightarrow{f}_{n-1}} - I_{\overrightarrow{BC}_n} + B_n), C_n + (G_{\overrightarrow{f}_{n-1}} - I_{\overrightarrow{BC}_n} + B_n) \geq I_{\overrightarrow{CA}_n}$.

$(G_{\overrightarrow{AB}_n}, G_{\overrightarrow{BC}_n}, G_{\overrightarrow{CA}_n}) = (k - I_{\overrightarrow{BC}_n} - I_{\overrightarrow{CA}_n}, I_{\overrightarrow{BC}_n}, I_{\overrightarrow{CA}_n})$

In this case, the requests at location B will be fulfilled first in the GBA-3L while loop, followed by the requests at location C. Finally, the remaining servers after fulfilling $I_{\overrightarrow{BC}_n}$ and $I_{\overrightarrow{CA}_n}$ will be allocated to location A. It should be noted that $k - I_{\overrightarrow{BC}_n} - I_{\overrightarrow{CA}_n} > \left\lfloor \frac{k}{3} \right\rfloor$.

4.3.3.2. $A_n \geq C_n + (G_{\overrightarrow{f}_{n-1}} - I_{\overrightarrow{BC}_n} + B_n), C_n + (G_{\overrightarrow{f}_{n-1}} - I_{\overrightarrow{BC}_n} + B_n) < I_{\overrightarrow{CA}_n}$.

$(G_{\overrightarrow{AB}_n}, G_{\overrightarrow{BC}_n}, G_{\overrightarrow{CA}_n}) = (A_n, I_{\overrightarrow{BC}_n}, k - A_n - I_{\overrightarrow{BC}_n})$

In this case, the requests at location B will be fulfilled first in GBA-3L while loop. Then, due to the condition $A_n \geq B_n + (G_{\overrightarrow{f}_{n-1}} - I_{\overrightarrow{CA}_n} + C_n)$, all remaining free servers after fulfilling $I_{\overrightarrow{BC}_n}$ will be allocated to location C. Furthermore, we can split it into two cases as follows.

If $k - A_n - I_{\overrightarrow{BC}_n} \geq \left\lfloor \frac{k}{3} \right\rfloor$.

If $k - A_n - I_{\overrightarrow{BC}_n} < \left\lfloor \frac{k}{3} \right\rfloor$, this implies $I_{\overrightarrow{BC}_n} < \left\lfloor \frac{k}{3} \right\rfloor$, which further implies $G_{\overrightarrow{f}_{n,exclC}} < \left\lfloor \frac{k}{3} \right\rfloor$.

4.3.3.3. $A_n < C_n + (G_{\overrightarrow{f}_{n-1}} - I_{\overrightarrow{BC}_n} + B_n), I_{\overrightarrow{AB}_n} > \left\lceil \frac{k - I_{\overrightarrow{BC}_n}}{2} \right\rceil, I_{\overrightarrow{CA}_n} >$

doi:10.6342/NTU202302667

$\left\lfloor \frac{k-I_{\overrightarrow{BC}_n}}{2} \right\rfloor$.

$(G_{\overrightarrow{AB}_n}, G_{\overrightarrow{BC}_n}, G_{\overrightarrow{CA}_n}) = (\left\lceil \frac{k-I_{\overrightarrow{BC}_n}}{2} \right\rceil, I_{\overrightarrow{BC}_n}, \left\lfloor \frac{k-I_{\overrightarrow{BC}_n}}{2} \right\rfloor)$

In this case, the requests at location B will be fulfilled first in GBA-3L while loop. Then, due to the condition $A_n < C_n + (G_{\overrightarrow{f}_{n-1}} - I_{\overrightarrow{BC}_n} + B_n)$, the remaining free servers after fulfilling $I_{\overrightarrow{BC}_n}$ will be allocated to location A and C until the number of servers at the two locations is approximately equal. It should be noted that $\left\lceil \frac{k-I_{\overrightarrow{BC}_n}}{2} \right\rceil, \left\lfloor \frac{k-I_{\overrightarrow{BC}_n}}{2} \right\rfloor \geq \left\lfloor \frac{k}{3} \right\rfloor$.

4.3.3.4. $A_n < C_n + (G_{\overrightarrow{f}_{n-1}} - I_{\overrightarrow{BC}_n} + B_n), I_{\overrightarrow{AB}_n} > \left\lceil \frac{k-I_{\overrightarrow{BC}_n}}{2} \right\rceil, I_{\overrightarrow{CA}_n} \leq \left\lfloor \frac{k-I_{\overrightarrow{BC}_n}}{2} \right\rfloor$.

$(G_{\overrightarrow{AB}_n}, G_{\overrightarrow{BC}_n}, G_{\overrightarrow{CA}_n}) = (k - I_{\overrightarrow{BC}_n} - I_{\overrightarrow{CA}_n}, I_{\overrightarrow{BC}_n}, I_{\overrightarrow{CA}_n})$

In this case, the requests at location B will be fulfilled first in the GBA-3L while loop, followed by the requests at location C. The remaining free servers after fulfilling $I_{\overrightarrow{BC}_n}$ and $I_{\overrightarrow{CA}_n}$ will be allocated to location A. It should be noted that $k - I_{\overrightarrow{BC}_n} - I_{\overrightarrow{CA}_n} > \left\lfloor \frac{k}{3} \right\rfloor$.

4.3.3.5. $A_n < C_n + (G_{\overrightarrow{f}_{n-1}} - I_{\overrightarrow{BC}_n} + B_n), I_{\overrightarrow{AB}_n} \leq \left\lceil \frac{k-I_{\overrightarrow{BC}_n}}{2} \right\rceil, I_{\overrightarrow{CA}_n} > \left\lfloor \frac{k-I_{\overrightarrow{BC}_n}}{2} \right\rfloor$.

$(G_{\overrightarrow{AB}_n}, G_{\overrightarrow{BC}_n}, G_{\overrightarrow{CA}_n}) = (I_{\overrightarrow{AB}_n}, I_{\overrightarrow{BC}_n}, k - I_{\overrightarrow{AB}_n} - I_{\overrightarrow{BC}_n})$

In this case, the requests at location B will be fulfilled first in the GBA-3L while loop, followed by the requests at location A. The remaining free servers after fulfilling $I_{\overrightarrow{AB}_n}$ and $I_{\overrightarrow{BC}_n}$ will be allocated to location C. It should be noted that $k - I_{\overrightarrow{AB}_n} - I_{\overrightarrow{BC}_n} \geq \left\lfloor \frac{k}{3} \right\rfloor$.

4.3.3.6. $A_n < C_n + (G_{\overrightarrow{f}_{n-1}} - I_{\overrightarrow{BC}_n} + B_n), I_{\overrightarrow{AB}_n} \leq \left\lceil \frac{k-I_{\overrightarrow{BC}_n}}{2} \right\rceil, I_{\overrightarrow{CA}_n} \leq \left\lfloor \frac{k-I_{\overrightarrow{BC}_n}}{2} \right\rfloor$. It will not happen.

doi:10.6342/NTU202302667

4.3.4. $A_n > \left\lfloor \frac{k}{3} \right\rfloor, I_{\overrightarrow{BC}_n} \leq \left\lceil \frac{k-A_n}{2} \right\rceil, I_{\overrightarrow{CA}_n} \leq \left\lfloor \frac{k-A_n}{2} \right\rfloor$.

$(G_{\overrightarrow{AB}_n}, G_{\overrightarrow{BC}_n}, G_{\overrightarrow{CA}_n}) = (k - I_{\overrightarrow{BC}_n} - I_{\overrightarrow{CA}_n}, I_{\overrightarrow{BC}_n}, I_{\overrightarrow{CA}_n})$

In this case, the requests at location B and C will be fulfilled first in the GBA-3L while loop. Afterward, the remaining servers will be allocated to location A. It should be noted that $k - I_{\overrightarrow{BC}_n} - I_{\overrightarrow{CA}_n} > \left\lfloor \frac{k}{3} \right\rfloor$.

4.3.5. $A_n \leq \left\lfloor \frac{k}{3} \right\rfloor$.

4.3.5.1. $I_{\overrightarrow{AB}_n} \leq \left\lceil \frac{k}{3} \right\rceil, I_{\overrightarrow{BC}_n} \leq \left\lceil \frac{k}{3} \right\rceil, I_{\overrightarrow{CA}_n} \leq \left\lfloor \frac{k}{3} \right\rfloor$. It will not happen.

4.3.5.2. $I_{\overrightarrow{AB}_n} \leq \left\lceil \frac{k}{3} \right\rceil, I_{\overrightarrow{BC}_n} \leq \left\lceil \frac{k}{3} \right\rceil, I_{\overrightarrow{CA}_n} > \left\lfloor \frac{k}{3} \right\rfloor$.

$(G_{\overrightarrow{AB}_n}, G_{\overrightarrow{BC}_n}, G_{\overrightarrow{CA}_n}) = (I_{\overrightarrow{AB}_n}, I_{\overrightarrow{BC}_n}, k - I_{\overrightarrow{AB}_n} - I_{\overrightarrow{BC}_n})$

In this case, the requests at location A and B will be fulfilled first in the GBA-3L while loop, and then the remaining servers will be allocated to location C.

4.3.5.3. $I_{\overrightarrow{AB}_n} \leq \left\lceil \frac{k}{3} \right\rceil, I_{\overrightarrow{BC}_n} > \left\lceil \frac{k}{3} \right\rceil, I_{\overrightarrow{CA}_n} \leq \left\lfloor \frac{k}{3} \right\rfloor$.

$(G_{\overrightarrow{AB}_n}, G_{\overrightarrow{BC}_n}, G_{\overrightarrow{CA}_n}) = (I_{\overrightarrow{AB}_n}, k - I_{\overrightarrow{AB}_n} - I_{\overrightarrow{CA}_n}, I_{\overrightarrow{CA}_n})$

In this case, the requests at location A and C will be fulfilled first in the GBA-3L while loop, and then the remaining servers will be allocated to location B.

4.3.5.4. $I_{\overrightarrow{AB}_n} > \left\lceil \frac{k}{3} \right\rceil, I_{\overrightarrow{BC}_n} \leq \left\lceil \frac{k}{3} \right\rceil, I_{\overrightarrow{CA}_n} \leq \left\lfloor \frac{k}{3} \right\rfloor$.

$(G_{\overrightarrow{AB}_n}, G_{\overrightarrow{BC}_n}, G_{\overrightarrow{CA}_n}) = (k - I_{\overrightarrow{BC}_n} - I_{\overrightarrow{CA}_n}, I_{\overrightarrow{BC}_n}, I_{\overrightarrow{CA}_n})$

In this case, the requests at location B and C will be fulfilled first in the GBA-3L while loop, and then the remaining servers will be allocated to location A.

4.3.5.5. $I_{\overrightarrow{AB}_n} > \left\lceil \frac{k}{3} \right\rceil, I_{\overrightarrow{BC}_n} > \left\lceil \frac{k}{3} \right\rceil, I_{\overrightarrow{CA}_n} \leq \left\lfloor \frac{k}{3} \right\rfloor$.

$(G_{\overrightarrow{AB}_n}, G_{\overrightarrow{BC}_n}, G_{\overrightarrow{CA}_n}) = (\left\lceil \frac{k-I_{\overrightarrow{CA}_n}}{2} \right\rceil, \left\lfloor \frac{k-I_{\overrightarrow{CA}_n}}{2} \right\rfloor, I_{\overrightarrow{CA}_n})$

doi:10.6342/NTU202302667

In this case, the requests at location C will be fulfilled first in the GBA-3L while loop. Then, the remaining servers will be assigned to locations A and B until the number of servers at the two locations is approximately equal.

4.3.5.6. $I_{\overrightarrow{AB}_n} > \left\lceil \frac{k}{3} \right\rceil, I_{\overrightarrow{BC}_n} \leq \left\lceil \frac{k}{3} \right\rceil, I_{\overrightarrow{CA}_n} > \left\lfloor \frac{k}{3} \right\rfloor.$

$(G_{\overrightarrow{AB}_n}, G_{\overrightarrow{BC}_n}, G_{\overrightarrow{CA}_n}) = (\left\lceil \frac{k-I_{\overrightarrow{BC}_n}}{2} \right\rceil, I_{\overrightarrow{BC}_n}, \left\lfloor \frac{k-I_{\overrightarrow{BC}_n}}{2} \right\rfloor)$

In this case, the requests at location B will be fulfilled first in the GBA-3L while loop. Then, the remaining servers will be assigned to locations A and C until the number of servers at the two locations is approximately equal.

4.3.5.7. $I_{\overrightarrow{AB}_n} \leq \left\lceil \frac{k}{3} \right\rceil, I_{\overrightarrow{BC}_n} > \left\lceil \frac{k}{3} \right\rceil, I_{\overrightarrow{CA}_n} > \left\lfloor \frac{k}{3} \right\rfloor.$

$(G_{\overrightarrow{AB}_n}, G_{\overrightarrow{BC}_n}, G_{\overrightarrow{CA}_n}) = (I_{\overrightarrow{AB}_n}, \left\lceil \frac{k-I_{\overrightarrow{AB}_n}}{2} \right\rceil, \left\lfloor \frac{k-I_{\overrightarrow{AB}_n}}{2} \right\rfloor)$

In this case, the requests at location A will be fulfilled first in the GBA-3L while loop. Then, the remaining servers will be assigned to locations B and C until the number of servers at the two locations is approximately equal.

4.3.5.8. $I_{\overrightarrow{AB}_n} > \left\lceil \frac{k}{3} \right\rceil, I_{\overrightarrow{BC}_n} > \left\lceil \frac{k}{3} \right\rceil, I_{\overrightarrow{CA}_n} > \left\lfloor \frac{k}{3} \right\rfloor.$

$(G_{\overrightarrow{AB}_n}, G_{\overrightarrow{BC}_n}, G_{\overrightarrow{CA}_n}) = (\left\lceil \frac{k}{3} \right\rceil, \left\lceil \frac{k}{3} \right\rceil, \left\lfloor \frac{k}{3} \right\rfloor)$

In this case, all available free servers will be allocated to locations A, B, and C until the number of servers at each location is approximately equal.

Observe that $G_{\overrightarrow{BC}_n}$ in cases 2.2.1., 2.2.2., 2.3.1., 2.3.2., 4.2.1., 4.2.2.1., 4.3.1. and 4.3.2.2. are special, and $G_{\overrightarrow{CA}_n}$ in case 2.3.1., 2.3.3., 4.3.1., 4.3.3.2. are special. Thus, we need the following definition and lemmas.

doi:10.6342/NTU202302667

Definition 6. *Let $G_{\overrightarrow{f}_{n,exclA}}$, $G_{\overrightarrow{f}_{n,exclB}}$ and $G_{\overrightarrow{f}_{n,exclC}}$ denote the number of free servers remaining at stage n - 1, which are then not assigned to locations A, B, and C at stage n, respectively.*

Lemma 15. *If the request number is greater than or equal to the initial number of servers at location A at stage n, i.e., $I_{\overrightarrow{AB}_n} \geq A_n$, then $G_{n-1} + G_{\overrightarrow{AB}_n} + k \geq \delta(O_{n-1} + O_{\overrightarrow{AB}_n}) + (k - G_{\overrightarrow{f}_{n,exclA}})$.*

Proof.

$$\begin{aligned}
G_{n-1} + G_{\overrightarrow{AB}_n} + k &= G_{n-1} + G_{\overrightarrow{AB}_n} + G_{\overrightarrow{f}_{n,exclA}} + (k - G_{\overrightarrow{f}_{n,exclA}}) \\
&= G_{n-1} + G_{\overrightarrow{CA}_{n-1}} + G_{\overrightarrow{f}_{n-1}} + (k - G_{\overrightarrow{f}_{n,exclA}}) \\
&\geq \delta(O_{n-1} + O_{\overrightarrow{CA}_{n-1}} + O_{\overrightarrow{f}_{n-1}}) + (k - G_{\overrightarrow{f}_{n,exclA}}) \\
&\geq \delta(O_{n-1} + O_{\overrightarrow{AB}_n}) + (k - G_{\overrightarrow{f}_{n,exclA}})
\end{aligned} \tag{3.13}$$

□

The proofs of lemma 16 and lemma 17 are similar and will be omitted.

Lemma 16. *If the request number is greater than or equal to the initial number of servers at location B at stage n, i.e. $I_{\overrightarrow{BC}_n} \geq B_n$, then $G_{n-1} + G_{\overrightarrow{BC}_n} + k \geq \delta(O_{n-1} + O_{\overrightarrow{BC}_n}) + (k - G_{\overrightarrow{f}_{n,exclB}})$*

Lemma 17. *If the request number is greater than or equal to the initial number of servers at location C at stage n, i.e. $I_{\overrightarrow{CA}_n} \geq C_n$, then $G_{n-1} + G_{\overrightarrow{CA}_n} + k \geq \delta(O_{n-1} + O_{\overrightarrow{CA}_n}) + (k - G_{\overrightarrow{f}_{n,exclC}})$*

Here, we start to prove lemma 2 to lemma 4. In order to demonstrate that lemma 2 to lemma 4 hold in $S(n)$ given $S(n-1)$ for all cases, we will first prove that lemma 4 is

doi:10.6342/NTU202302667

true for all cases. The proofs of lemma 2 and lemma 3 are similar and will be omitted.

Proof. $G_{\overrightarrow{BC}_n}$ in cases 2.2.1., 2.2.2., 2.3.1., 2.3.2., 4.2.1., 4.2.2.1., 4.3.1. and 4.3.2.2. are special, we know that $G_{\overrightarrow{f}_{n,exclB}} \leq \lfloor \frac{k}{3} \rfloor$,

$$\begin{aligned} G_{n,C} &= G_{n-1} + G_{\overrightarrow{BC}_n} + k \\ &\geq (k - G_{\overrightarrow{f}_{n,exclB}}) + \delta(O_{n-1} + O_{\overrightarrow{BC}_n}) \\ &\geq \frac{2}{3}k + \delta(O_{n-1} + O_{\overrightarrow{BC}_n}) \\ &\geq \delta k + \delta(O_{n-1} + O_{\overrightarrow{BC}_n}) \\ &= \delta(O_{n-1} + O_{\overrightarrow{BC}_n} + k) \\ &= \delta O_{n,C} \end{aligned} \tag{3.14}$$

In equation 3.14, the first inequality follows from the lemma 16. The second and third inequalities can be easily understood.

The proofs of the other cases of $G_{\overrightarrow{BC}_n}$ in lemma 4 are the same as those discussed in section 3.1.1.

□

Next, we will proceed to prove lemma 5 to lemma 7. To demonstrate that lemma 5 to lemma 7 hold in $S(n)$ given $S(n-1)$ for all cases, we will symmetrically prove that lemma 6 is true for all cases. The proofs of lemma 5 and lemma 7 are similar and are will be omitted.

Proof. In cases 2.2.1., 2.2.2., 2.3.1., 2.3.2., we know that $G_{\overrightarrow{AB}_n} = I_{\overrightarrow{AB}_n}$ and $G_{\overrightarrow{f}_{n,exclB}} \leq$

doi:10.6342/NTU202302667

$\lfloor \frac{k}{3} \rfloor$,

$$
\begin{aligned}
G_{n,BC} &= G_{n-1} + G_{\overrightarrow{AB}_n} + G_{\overrightarrow{BC}_n} + k \\
&= (G_{n-1} + G_{\overrightarrow{BC}_n} + k) + G_{\overrightarrow{AB}_n} \\
&\geq (k - G_{\overrightarrow{f}_{n,exclB}}) + \delta(O_{n-1} + O_{\overrightarrow{BC}_n}) + I_{\overrightarrow{AB}_n} \\
&\geq \frac{2}{3}k + \delta(O_{n-1} + O_{\overrightarrow{BC}_n}) + \delta O_{\overrightarrow{AB}_n} \\
&\geq \delta k + \delta(O_{n-1} + O_{\overrightarrow{BC}_n}) + \delta O_{\overrightarrow{AB}_n} \\
&= \delta(O_{n-1} + O_{\overrightarrow{AB}_n} + O_{\overrightarrow{BC}_n} + k) \\
&= \delta O_{n,C}
\end{aligned}
\tag{3.15}
$$

In equation 3.15, the first inequality follows from the lemma 16 and the fact that $G_{\overrightarrow{AB}_n} = I_{\overrightarrow{AB}_n}$. The second inequality follows from lemma 8 and the fact $\delta \leq 1$. The proofs of the other cases of $G_{\overrightarrow{AB}_n} + G_{\overrightarrow{BC}_n}$ in lemma 6 are the same as those discussed in section 3.1.1.

□

Finally, given the inductive hypothesis $S(n-1)$, we prove that $S(n)$ holds when the initial servers cannot fulfill the requests at location C.

3.1.3 Proof of Lemma 1

In this section, we will prove that lemma 1 holds for all cases based on the previous induction step $S(n-1)$.

Proof. First, consider that no free server is left after the GBA-3L while loop.

1. If there is no location where initial servers can fulfill the requests, and all free servers are used up to fill three locations after the GBA-3L while loop, then G_n in equation 3.1 will be reduced to equation 3.16.

doi:10.6342/NTU202302667

$$G_n = G_{n-1} + G_{\overrightarrow{AB}_{n-1}} + G_{\overrightarrow{BC}_{n-1}} + G_{\overrightarrow{CA}_{n-1}} + G_{\overrightarrow{f}_{n-1}} \tag{3.16}$$

In this case, equation 3.17 shows the proof of lemma 1.

$$\begin{aligned}
G_n = G_{n-1} + G_{\overrightarrow{AB}_n} + G_{\overrightarrow{BC}_n} + G_{\overrightarrow{CA}_n} &= G_{n-1} + G_{\overrightarrow{AB}_{n-1}} + G_{\overrightarrow{BC}_{n-1}} + G_{\overrightarrow{CA}_{n-1}} + G_{\overrightarrow{f}_{n-1}} \\
&= G_{n-1} + k \\
&\geq G_{n-1} + O_{\overrightarrow{AB}_{n-1}} + O_{\overrightarrow{BC}_{n-1}} + O_{\overrightarrow{CA}_{n-1}} + O_{\overrightarrow{f}_{n-1}} \\
&\geq \delta O_{n-1} + \delta(O_{\overrightarrow{AB}_{n-1}} + O_{\overrightarrow{BC}_{n-1}} + O_{\overrightarrow{CA}_{n-1}} + O_{\overrightarrow{f}_{n-1}}) \\
&= \delta(O_{n-1} + O_{\overrightarrow{AB}_n} + O_{\overrightarrow{BC}_n} + O_{\overrightarrow{CA}_n}) \\
&= \delta O_n
\end{aligned} \tag{3.17}$$

In equation 3.17, the first inequality follows from lemma 14. The second inequality follows from lemma 1 and the fact that $\delta \leq 1$.

2. If there is only one location where initial servers can fulfill the requests, and all free servers are used up to fill the other two locations after the GBA-3L while loop. For example, if the initial servers at location A can fulfill the requests and all free servers are used up at location B and C after the GBA-3L while loop, then G_n in equation 3.1 will be reduced to equation 3.18. Equation 3.19 and equation 3.20 are reduced symmetrically.

$$G_n = G_{n-1} + I_{\overrightarrow{AB}_n} + G_{\overrightarrow{AB}_{n-1}} + G_{\overrightarrow{BC}_{n-1}} + G_{\overrightarrow{f}_{n-1}} \tag{3.18}$$

$$G_n = G_{n-1} + I_{\overrightarrow{BC}_n} + G_{\overrightarrow{BC}_{n-1}} + G_{\overrightarrow{CA}_{n-1}} + G_{\overrightarrow{f}_{n-1}} \tag{3.19}$$

$$G_n = G_{n-1} + I_{\overrightarrow{CA}_n} + G_{\overrightarrow{CA}_{n-1}} + G_{\overrightarrow{AB}_{n-1}} + G_{\overrightarrow{f}_{n-1}} \tag{3.20}$$

doi:10.6342/NTU202302667

In this case, equation 3.21 shows the proof of equation 3.18 in lemma 1. The proofs of equation 3.19 and equation 3.20 in lemma 1 are similar and will be omitted.

$$\begin{aligned}
G_n = G_{n-1} + G_{\overrightarrow{AB}_n} + G_{\overrightarrow{BC}_n} + G_{\overrightarrow{CA}_n} &= G_{n-1} + I_{\overrightarrow{AB}_n} + G_{\overrightarrow{AB}_{n-1}} + G_{\overrightarrow{BC}_{n-1}} + G_{\overrightarrow{f}_{n-1}} \\
&= (G_{n-1} + G_{\overrightarrow{AB}_{n-1}} + G_{\overrightarrow{BC}_{n-1}} + G_{\overrightarrow{f}_{n-1}}) + I_{\overrightarrow{AB}_n} \\
&\geq \delta(O_{n-1} + O_{\overrightarrow{AB}_{n-1}} + O_{\overrightarrow{BC}_{n-1}} + O_{\overrightarrow{f}_{n-1}}) + \delta I_{\overrightarrow{AB}_n} \\
&\geq \delta(O_{n-1} + O_{\overrightarrow{BC}_n} + O_{\overrightarrow{CA}_n}) + \delta O_{\overrightarrow{AB}_n} \\
&= \delta(O_{n-1} + O_{\overrightarrow{AB}_n} + O_{\overrightarrow{BC}_n} + O_{\overrightarrow{CA}_n}) \\
&= \delta O_n
\end{aligned} \tag{3.21}$$

In equation 3.21, the first inequality follows from lemma 6 and the fact that $\delta \leq 1$. The second inequality follows from lemma 8 and lemma 12.

3. If there are two locations where initial servers can fulfill the requests, and all free servers are used up to fill the last location after the GBA-3L while loop. For example, if the initial servers at location A and B can fulfill the requests and all free servers are used up at location C after the GBA-3L while loop, then G_n in equation 3.1 will be reduced to equation 3.22. Equation 3.23 and equation 3.24 are reduced symmetrically.

$$G_n = G_{n-1} + I_{\overrightarrow{AB}_n} + I_{\overrightarrow{BC}_n} + G_{\overrightarrow{BC}_{n-1}} + G_{\overrightarrow{f}_{n-1}} \tag{3.22}$$

$$G_n = G_{n-1} + I_{\overrightarrow{BC}_n} + I_{\overrightarrow{CA}_n} + G_{\overrightarrow{CA}_{n-1}} + G_{\overrightarrow{f}_{n-1}} \tag{3.23}$$

$$G_n = G_{n-1} + I_{\overrightarrow{CA}_n} + I_{\overrightarrow{AB}_n} + G_{\overrightarrow{AB}_{n-1}} + G_{\overrightarrow{f}_{n-1}} \tag{3.24}$$

In this case, equation 3.25 shows the proof of equation 3.22 in lemma 1. The proofs

doi:10.6342/NTU202302667

of equation 3.23 and equation 3.24 in lemma 1 are similar and will be omitted.

$$\begin{aligned}
G_n = G_{n-1} + G_{\overrightarrow{AB}_n} + G_{\overrightarrow{BC}_n} + G_{\overrightarrow{CA}_n} &= G_{n-1} + I_{\overrightarrow{AB}_n} + I_{\overrightarrow{BC}_n} + G_{\overrightarrow{BC}_{n-1}} + G_{\overrightarrow{f}_{n-1}} \\
&= (G_{n-1} + G_{\overrightarrow{BC}_{n-1}} + G_{\overrightarrow{f}_{n-1}}) + I_{\overrightarrow{AB}_n} + I_{\overrightarrow{BC}_n} \\
&\geq \delta(O_{n-1} + O_{\overrightarrow{BC}_{n-1}} + O_{\overrightarrow{f}_{n-1}}) + \delta(I_{\overrightarrow{AB}_n} + I_{\overrightarrow{BC}_n}) \\
&\geq \delta(O_{n-1} + O_{\overrightarrow{CA}_n}) + \delta(O_{\overrightarrow{AB}_n} + O_{\overrightarrow{BC}_n}) \\
&= \delta(O_{n-1} + O_{\overrightarrow{AB}_n} + O_{\overrightarrow{BC}_n} + O_{\overrightarrow{CA}_n}) \\
&= \delta O_n
\end{aligned} \tag{3.25}$$

In equation 3.25, the first inequality follows from lemma 4 and the fact that $\delta \leq 1$. The second inequality follows from lemma 8, lemma 9 and lemma 10.

4. If there are three locations where initial servers can fulfill the requests, and no free server is left after the GBA-3L while loop, then G_n in equation 3.1 will be reduced to equation 3.26.

$$G_n = G_{n-1} + I_{\overrightarrow{AB}_n} + I_{\overrightarrow{BC}_n} + I_{\overrightarrow{CA}_n} \tag{3.26}$$

In this case, equation 3.27 shows the proof of equation 3.26 in lemma 1.

$$\begin{aligned}
G_n = G_{n-1} + G_{\overrightarrow{AB}_n} + G_{\overrightarrow{BC}_n} + G_{\overrightarrow{CA}_n} &= G_{n-1} + I_{\overrightarrow{AB}_n} + I_{\overrightarrow{BC}_n} + I_{\overrightarrow{CA}_n} \\
&\geq \delta O_{n-1} + (I_{\overrightarrow{AB}_n} + I_{\overrightarrow{BC}_n} + I_{\overrightarrow{CA}_n}) \\
&\geq \delta O_{n-1} + \delta(O_{\overrightarrow{AB}_n} + O_{\overrightarrow{BC}_n} + O_{\overrightarrow{CA}_n}) \\
&= \delta(O_{n-1} + O_{\overrightarrow{AB}_n} + O_{\overrightarrow{BC}_n} + O_{\overrightarrow{CA}_n}) \\
&= \delta O_n
\end{aligned} \tag{3.27}$$

In equation 3.27, the first inequality follows from lemma 1. The second inequality

doi:10.6342/NTU202302667

follows from lemma 8, lemma 9 and lemma 10 and the fact that $\delta \leq 1$.

Second, consider that there are more than one free server left after the GBA-3L while loop. In this case, due to greedy property of GBA-3L, it means that the requests at all locations are fulfilled. The reduction of G_n in equation 3.1 is the same as equation 3.16, and the proof of equation 3.16 in lemma 1 is the same as equation 3.17. □

Given the inductive hypothesis $S(n-1)$, we prove that lemma 1 holds for all cases. Therefore, theorem 1 is proved. □

3.2 Tight bound analysis

Theorem 2. *No deterministic algorithms for the kS3L-F problem can achieve a competitive ratio less than* $\frac{2k}{k+\lfloor k/3 \rfloor}$.

Proof. Let A* be any deterministic algorithm. Consider the requests of an instance: $I_{\overrightarrow{AB}_1} = k$, $I_{\overrightarrow{BC}_1} = k$, and $I_{\overrightarrow{CA}_1} = k$ at stage 1. Suppose A* accepts $G_{\overrightarrow{AB}_1}$, $G_{\overrightarrow{BC}_1}$, and $G_{\overrightarrow{CA}_1}$ at stage 1. Without loss of generality, assume that $G_{\overrightarrow{AB}_1}$ is the minimum value, and the only requests for this instance at stage 2 are $I_{\overrightarrow{BC}_2} = k$. The fulfilled requests of A* are $G_{\overrightarrow{AB}_1} + G_{\overrightarrow{BC}_1} + G_{\overrightarrow{CA}_1}$ at stage 1 and at most $k - G_{\overrightarrow{AB}_1} - G_{\overrightarrow{BC}_1} - G_{\overrightarrow{CA}_1} + G_{\overrightarrow{AB}_1}$ at stage 2. Thus, the total fulfilled requests of A* will be at most $k + G_{\overrightarrow{AB}_1} \leq k + \lfloor k/3 \rfloor$. On the other hand, the total fulfilled requests of OPT-3L are $O_{\overrightarrow{AB}_1} + O_{\overrightarrow{BC}_2} = 2k$. Therefore, theorem 2 is proved and the bound is tight. □

doi:10.6342/NTU202302667

doi:10.6342/NTU202302667

Chapter 4 Conclusion and Future Work

In conclusion, we propose a new algorithm using similar idea named GBA-3L for kS3L-S. Furthermore, we address the problem for kS3L-S in the induction proof where one location has a significantly high number of servers, while the other two locations have significantly fewer servers. It will result in allocating all the free servers to the location with the fewest servers, while the number of servers at the second fewest location remains low. In contrast, in kS2L-S, GBA can always allocate all the free servers to the location with the fewest local servers at each stage. We believe that the problem encountered in the induction proof still exists for the future work. Lemmas 15 to 17 provide useful techniques for solving this kind of problem. Finally, we successfully demonstrate that the lower bound of the competitive ratio for kS3L-S is $\frac{2k}{k+\lfloor k/3 \rceil}$ and prove that GBA-3L achieves the same competitive ratio. The proof illustrate our algorithm is optimal.

In application, due to the discrete time property of the simultaneous model, it can be applied in certain circumstances. For instance, in a college setting where classes always start exactly at the hour, the simultaneous model can be used to schedule bikes at multiple stations. Furthermore, the intercity bus system in Taiwan is a real example where buses are scheduled on discrete time intervals.

doi:10.6342/NTU202302667

For the future work, one possible work is to extend the circular one-way simultaneous model from three locations to N locations, where $N > 3$. For the one-way circular model, we just allocate local servers within a single location for a single direction. In contrast, for the multiple-way model, we need to consider numerous combinations of server allocation for multiple ways within a single location. Another possible work is to study a multiple-way model with three locations, which is a problem we have yet to solve. In this model, each location has two directional requests. In addition to theoretical studies, one possible work is to practically analyze the particular request distributions given certain circumstances, using simulation or machine-learning techniques to uncover the distribution.

doi:10.6342/NTU202302667

References

[1] N. Ascheuer, S. O. Krumke, and J. 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 17–19, 2000 Proceedings 17, pages 639–650. Springer, 2000.

[2] A. Bjelde, J. Hackfeld, Y. Disser, C. Hansknecht, M. Lipmann, J. Meißner, M. Schlöter, K. Schewior, and L. Stougie. Tight bounds for online tsp on the line. ACM Transactions on Algorithms (TALG), 17(1):1–58, 2020.

[3] K. Böhmová, Y. Disser, M. Mihalák, and R. Šrámek. Scheduling transfers of resources over time: Towards car-sharing with flexible drop-offs. In LATIN 2016: Theoretical Informatics: 12th Latin American Symposium, Ensenada, Mexico, April 11-15, 2016, Proceedings 12, pages 220–234. Springer, 2016.

[4] A. Christman, W. Forcier, and A. Poudel. From theory to practice: maximizing revenues for on-line dial-a-ride. Journal of Combinatorial Optimization, 35:512–529, 2018.

[5] S. O. Krumke, W. E. de Paepe, D. Poensgen, M. Lipmann, A. Marchetti-Spaccamela, and L. Stougie. On minimizing the maximum flow time in the online dial-a-ride problem. In Approximation and Online Algorithms: Third International Workshop,

doi:10.6342/NTU202302667

WAOA 2005, Palma de Mallorca, Spain, October 6-7, 2005, Revised Papers 3, pages 258–269. Springer, 2006.

[6] Y.-C. Liang, K.-Y. Lai, H.-L. Chen, K. Iwama, and C.-S. Liao. Tight competitive analyses of online car-sharing problems. Theoretical Computer Science, 938:86–96, 2022.

[7] K. Luo, T. Erlebach, and Y. Xu. Car-sharing between two locations: Online scheduling with flexible advance bookings. In Computing and Combinatorics: 24th International Conference, COCOON 2018, Qing Dao, China, July 2-4, 2018, Proceedings 24, pages 242–254. Springer, 2018.

[8] K. Luo, T. Erlebach, and Y. Xu. Car-sharing between two locations: online scheduling with two servers. In 43rd International Symposium on Mathematical Foundations of Computer Science (MFCS 2018). Schloss Dagstuhl-Leibniz-Zentrum fuer Informatik, 2018.

[9] K. Luo, T. Erlebach, and Y. Xu. Online scheduling of car-sharing requests between two locations with many cars and flexible advance bookings. In 29th International Symposium on Algorithms and Computation (ISAAC 2018). Schloss Dagstuhl-Leibniz-Zentrum fuer Informatik, 2018.

[10] K. Luo, T. Erlebach, and Y. Xu. Car-sharing on a star network: on-line scheduling with k servers. In 36th International Symposium on Theoretical Aspects of Computer Science (STACS 2019). Schloss Dagstuhl-Leibniz-Zentrum fuer Informatik, 2019.

[11] G. Perboli, F. Ferrero, S. Musso, and A. Vesco. Business models and tariff simulation in car-sharing services. Transportation Research Part A: Policy and Practice, 115:32–48, 2018.

doi:10.6342/NTU202302667

[12] M. Repoux, B. Boyacı, and N. Geroliminis. Simulation and optimization of one-way car-sharing systems with variant relocation policies. In hEART 2014-3rd Symposium of the European Association for Research in Transportation, 2014.

[13] F. Yi and L. Tian. On the online dial-a-ride problem with time-windows. In International Conference on Algorithmic Applications in Management, pages 85–94. Springer, 2005.

doi:10.6342/NTU202302667