

学校代码：10255

学 号：2130865

中图法分类号：

集装箱码头泊位与岸桥联合在线调度研究

Study of online berth and quay crane integrated allocation in
container terminals

学科专业：管理科学与工程

作者姓名：乔龙亮

指导教师：郑斐峰

答辩日期：2016 年 1 月 4 日

东华大学学位论文原创性声明

本人郑重声明：我恪守学术道德，崇尚严谨学风。所呈交的学位论文，是本人在导师的指导下，独立进行研究工作所取得的成果。除文中已明确注明和引用的内容外，本论文不包含任何其他个人或集体已经发表或撰写过的作品及成果的内容。论文为本人亲自撰写，我对所写的内容负责，并完全意识到本声明的法律结果由本人承担。

学位论文作者签名：

日期： 年 月 日

东华大学学位论文版权使用授权书

学位论文作者完全了解学校有关保留、使用学位论文的规定，同意学校保留并向国家有关部门或机构送交论文的复印件和电子版，允许论文被查阅或借阅。本人授权东华大学可以将本学位论文的全部或部分内容编入有关数据库进行检索，可以采用影印、缩印或扫描等复制手段保存和汇编本学位论文。

保密 ☐，在 ____ 年解密后适用本版权书。

本学位论文属于

不保密 ☐。

学位论文作者签名：

指导教师签名：

日期： 年 月 日

日期： 年 月 日

东华大学

硕士学位论文答辩委员会成员名单

姓名	职称	职务	工作单位	备注
林杰	教授	答辩委员会主席	同济大学	
宋福根	教授	答辩委员会委员	东华大学	
姚卫新	教授	答辩委员会委员	东华大学	
范宏	教授	答辩委员会委员	东华大学	
周力	副教授	答辩委员会委员	东华大学	
周蓉	讲师	答辩委员会秘书	东华大学	

集装箱码头泊位与岸桥联合在线调度研究

摘要

为了应对集装箱码头准班船舶准班率的下降,码头管理者需要对实时到达的集装箱船舶进行排班和调度,如何对这些需求进行合理调度决策,从而力求服务时间最短是集装箱码头的核心目标之一。针对集装箱船舶实时到达的上述情形,运用在线理论工具,对泊位与岸桥的联合调度进行问题刻画,构建相应的在线调度模型。突破了现有问题现有研究设计复杂启发式算法的主导思路,设计在线调度策略,并运用竞争比指标对其执行效果进行度量。

考虑了由3个相邻泊位组成的混合型泊位布局,论文对有预知与没有预知能力的两种情形,均进行了深入剖析。针对加班船舶请求在零时刻逐个抵港且需要及时决策的Over-list在线情形展开深入研究;决策内容包括靠泊位置安排以及服务岸桥的分配,目标为最小化最大完工时间即最后一艘船舶服务完成时间的优化目标。基于对后续请求个数的不同预知能力分别建立了调度模型。对于有些情形,给出了最优的在线调度策略;对于另一些情形,也给出了竞争比上下界相近的在线竞争策略。论文的主要工作与创新点:

- (1) 研究了无预知信息下的集装箱码头泊位和岸桥联合调度在线模型。针对给定的混合型泊位,对于配置了4、5、6个岸桥的三种情形,分别设计了具有最优竞争比 $5/4$ 、 $4/3$ 和 $3/2$ 的在线调度策略。最优在线策略意味着该策略的竞争比与问题的竞争比下界相等。
- (2) 对预知后续 $k=1$ 个请求的泊位与岸桥联合调度在线情形展开探究。类似地对于岸桥数量为4、5或6个的三种情形,分别设计了具有最优竞争比为 $(1+\sqrt{2})/2$ 、 $5/4$ 和 $4/3$ 的联合调度在线策略;结合没有预知信息情形的相应结论,可知,有限的预知能力可以有效的改进在线策略的竞争性能。
- (3) 进一步预知后续 $k \geq 2$ 个请求的码头泊位与岸桥联合调度在线模型。首先,对于 $k \geq 2$ 的一般情形,对于4个岸桥的情形,证明其竞争比下界值随着 k 的增加而趋近于1;对于5、6个岸桥数量的两种情形,证明了在线策略竞争比下界分别为 $25/21$ 和 $9/7$ 。其次,对于 $k=2$ 的特定情形,对3种岸桥配置情况,分别设计了竞争比为 $9/8$ 、 $6/5$ 、 $9/7$ 的在线调度策略。其中对于6个岸桥的情形,预知未来2个请求即可实现最优的在线调度效果。

关键词: 调度决策; 集装箱码头; 联合调度; 在线策略; 竞争比

Abstract

In response to the decline punctuality rate of real-time ships in the container terminal, managers need to schedule and dispatch those real-time vessels, one of the key problems in improving service efficiency of terminal resources is to make and fulfill a flexible and reliable service schedule for arrive vessels. In view of the above situation of real-time ships in the container terminal, use competitive analysis to analyze the performance of online algorithms for the model of berth and quay crane integrated allocation in container terminals, and to construct the corresponding online scheduling model. This article broke the existing research which design of complex heuristics ideas of the current problem, and designs online scheduling strategy, and use the competitive ratio to measure the performance of its execution effect.

We consider a hybrid berth which consists of three adjacent small berths, this paper has carried on the thorough analysis to the kinds of situations, which are both predictable and unpredictable. For the real-time requests are to be released over list at zero time which arriving one by one. On the release of each request, a scheduler has to decide immediately its service resource combination, the special berth allocation and QCs. The objective is to minimize the makespan or the completion time of the last completed request. The scheduling model is established by the different look-ahead ability for the subsequent requests. For some cases, the optimal online scheduling strategies are given; for the other situations, the online competitive strategy is also given, which are similar to the upper and lower bounds of the competitive ratios. The main work and innovative points are as follows:

- (1) In the first part, we study an online over-list model of berth and quay crane integrated allocation in container terminals without look-ahead, aiming at minimizing the maximum completion time of container vessel. We focus on the scenario with small numbers of berths and cranes. For the setting with three berths, we prove matching upper and lower bounds of $5/4$, $4/3$ and $3/2$ for the cases with 4, 5 and 6 cranes, respectively. We can show that the algorithms we proposed are optimal in the sense that their competitive ratios match the lower bounds.
- (2) In the second part, we present an online over-list model of berth and quay crane integrated allocation in the scenario with a 1-lookahead ability. Given that an online algorithm can only foresee the next one vessel request on the release of each request, we propose three online algorithms which are optimally $(\sqrt{2}+1)/2$, $5/4$ and $4/3$ -competitive for the three cases with 4, 5 and 6 cranes respectively.

Combined with the corresponding conclusions without look-ahead, and thus reveal the positive effect of the look-ahead function.

- (3) In the last part, we study an online over-list model of berth and quay crane integrated allocation in the scenario with k -lookahead ability such that the next $k \geq 2$ requests are foreseen on the release of any request. First, for the case of with k -lookahead ability, one main result is a lower bound is close to 1 of competitive ratio for deterministic online strategies with the increase of look-ahead ability for 4 cranes, we further give lower bounds of $25/21$ and $9/7$ respectively for 5 and 6 cranes. Second, We propose a $9/8$, $6/5$ and $9/7$ -competitive algorithm for the case with 4, 5 and 6 cranes with 2-lookahead ability, respectively. For the case with 6 cranes, the above results reveal that any larger look-ahead ability cannot further improve the competitive performance of online strategies.

Keywords: Scheduling; Container terminal; integrated allocation; Online algorithm; Competitive ratio

目 录

摘 要	I
Abstract	II
目 录	IV
主要符号对照表.....	VI
第 1 章 绪论.....	1
1.1 研究背景及意义.....	1
1.1.1 研究背景.....	1
1.1.2 研究意义.....	2
1.2 基本概念.....	3
1.2.1 集装箱码头调度相关概念.....	3
1.2.2 在线问题及其竞争策略的基本概念.....	4
1.3 集装箱码头调度的国内外研究综述.....	5
1.3.1 泊位分配问题研究.....	6
1.3.2 泊位-岸桥联合调度 (BAP-QCAP) 问题研究.....	6
1.4 研究方法、思路与内容.....	9
1.4.1 研究方法.....	9
1.4.2 研究思路.....	10
1.4.3 研究内容.....	11
1.5 技术路线.....	11
1.6 工作的创新点.....	13
第 2 章 无预知信息下泊位与岸桥联合在线调度.....	14
2.1 问题描述与模型刻画.....	14
2.2 四个岸桥的调度情形.....	16
2.2.1 竞争比下界分析.....	16
2.2.2 在线策略 GR 设计及分析.....	17
2.3 五个岸桥的调度情形.....	18
2.3.1 竞争比下界分析.....	18
2.3.2 在线策略 LRSS 设计及分析.....	19
2.4 六个岸桥的调度情形.....	24
2.4.1 竞争比下界分析.....	24
2.4.2 在线策略 QBT 设计与分析.....	25
2.5 本章小结.....	26
第 3 章 预知 $k=1$ 个请求下泊位与岸桥联合在线调度.....	27
3.1 问题描述与模型刻画.....	27

3.2 四个岸桥的调度情形.....	28
3.2.1 竞争比下界分析.....	28
3.2.2 策略 LLWL 的设计及竞争比分析.....	29
3.2.3 策略 LSWL 的设计及竞争比分析.....	30
3.3 五个岸桥的调度情形.....	31
3.3.1 竞争比下界分析.....	31
3.3.2 $\Delta \geq 3$ 的情形.....	32
3.3.3 $2 \leq \Delta < 3$ 的情形.....	35
3.4 六个岸桥的调度情形.....	39
3.4.1 竞争比下界分析.....	39
3.4.2 策略 GRTWL 及其竞争分析.....	40
3.4.3 数值分析.....	42
3.5 本章小结.....	43
第 4 章 预知 $k \geq 2$ 个请求下泊位与岸桥联合在线调度.....	44
4.1 问题描述与模型刻画.....	44
4.2 四个岸桥的调度情形.....	44
4.2.1 竞争比下界分析.....	45
4.2.2 在线策略 GRWL ₂ 的设计.....	46
4.2.3 策略竞争分析.....	47
4.3 五个岸桥的调度情形.....	48
4.3.1 竞争比下界分析.....	48
4.3.2 在线策略 GSRWL ₂ 设计与竞争分析.....	49
4.3.3 在线策略 GLRWL ₂ 设计与竞争分析.....	53
4.4 六个岸桥的调度情形.....	55
4.4.1 竞争比下界分析.....	55
4.4.2 BLA 策略描述.....	56
4.4.3 BLA 竞争性能分析.....	57
4.5 本章小结.....	60
第 5 章 结论与展望.....	62
5.1 主要研究结论.....	62
5.2 未来研究展望.....	62
参考文献.....	64
附录.....	68
攻读硕士学位期间取得的研究成果.....	73
致谢.....	74

主要符号对照表

下面列出模型中主要使用的变量符号。

$C_{\max}(\sigma)$	在线策略的目标函数值——最大完工时间；
$C^*(\sigma)$	离线策略的目标函数值——最大完工时间；
$t_{i,1}$	释放 r_i 时，至少一个泊位完成此前分配至该泊位的所有请求的最早完工时间；
$t_{i,2}, t_{i,3}$	释放 r_i 时，2 个或 3 个连续泊位完成此前分配至各自泊位的所有请求的最早完工时间；
$C_{i,j} (1 \leq j \leq 3)$	释放 r_i 时， b_j 泊位完成此前所分配的所有请求的最早完工时间；
s_i, e_i	请求 r_i 的服务开始时间和结束时间；
$T_a = [t_1, t_2)$	在线策略分配某个大请求 r_i 导致的某一泊位服务该大请求前出现可利用空闲时间段 T_a ，满足 $t_2 = s_i \geq t_1 + 1/2$ ；初始化 $T_a = [0, 0)$ ；
n_q	调度系统中 3 个小泊位所配置的岸桥数量；
$T_w = [e_k, s_i)$	在线策略分配某个大请求 r_i 导致的某一泊位服务该大请求前出现无法利用的空闲时间段 T_w ；

第 1 章 绪论

1.1 研究背景及意义

1.1.1 研究背景

集装箱运输是以集装箱作为载体的一种现代化货物运输方式，它具有“安全、迅速、简便、价廉”的优点；而船舶运输工具一直以来在集装箱运输行业中发挥不可替代的重要作用。由于其装运空间的有利条件，使得包括铁路、公路以及航空在内的其他运输途径难以与水路船舶运输展开市场竞争。我国的集装箱运输行业起始于上个世纪 70 年代，并在近十几年中取得蓬勃发展，尤其是集装箱港口更是呈加速发展态势。在 2012 年集装箱吞吐量世界排名前十位的港口中，我国港口就占了 6 个，2013 年更是占据了 7 个名额。2007-2012 年，我国港口集装箱吞吐量年复合增长率为 9.17%。2012 年全国港口完成集装箱吞吐量 17747 万 TEU（标准箱），比上年增长 8.43%；2013 年 1-9 月，我国港口集装箱吞吐量为 1.41 亿 TEU，同比增长 7.5%；目前已初步建成环渤海、长江三角洲、东南沿海、珠江三角洲和西南沿海 5 个规模化、集约化、现代化的港口群体。在这些港口群体中，多数港口的现有设计通过能力都小于实际吞吐量需求。2011 年全国港口集装箱设计通行能力约为 1.42 亿 TEU，而实际完成量为 1.64 亿 TEU，总体利用率达 115%，而青岛港更是高达 170%。在未来市场需求不断上升的大好趋势下，一部分港口进行二期、三期码头建设以扩充产能，而其他许多港口目前只能维持码头资源透支运作的状态。因此，如何针对当前的有限资源条件，通过优化调度方案进一步提升港口资源利用率及通行能力，以缓解供需矛盾，这是一个重要而迫切的任务。

作为集装箱码头衔接海、陆运输的转换点，泊位与岸桥资源是制约集装箱码头规模与效率的最重要资源，也是码头其他作业的输入端，提升这两项资源的利用率对于码头作业效率提高的重要性不言而喻；此外，这两者也是相互制约的，泊位的分配决定了岸桥的分配，而岸桥的数量又决定了泊位的占用时间，影响后续泊位计划，因此，同时对这两种资源进行调度优化是集装箱码头重要课题之一。由于码头调度效率主要取决于水岸泊位、岸桥等关键资源的整体效率，因此，本文拟针对泊位与岸桥的联合调度问题展开深入系统的研究，为码头资源调度实践方案的改进提供优化思路和理论支持。

从宏观经济的角度来看，港口物流在我国的经济发展中发挥着日益重要的作用，上海正在为建设成为世界航运中心的目标而努力，在以外高桥港口为中心建立的中国（上海）自由贸易区试点，相继又开发了洋山深水港一期、二期、三期

和四期。因而本文所研究的集装箱码头资源联合调度问题是上海经济发展中的热点问题之一，极具现实意义。

其次，当服务请求发生的时候，服务资源的利用率成为影响企业盈利的一个主要因素。在码头实际作业中，船舶需求不可预见性的特点会使得现有的码头资源配置不合理。为了提高顾客的满意度和船舶的准班率，要求码头计划部门针对顾客提出的需求给出实时反馈和资源配置。因此，如何合理的配置仅有的码头资源，使得码头能够为更多的船舶提供服务将是在线理论研究中非常具有挑战性的领域之一。

就集装箱码头资源配置而言，可以把该问题看作更具普遍意义的资源调度问题，实施理论上和数学上的建模与决策。鉴于我国目前集装箱码头面临空间资源不足、设备资源分配不合理等各类问题，需以最少的资源投入实现资源效用最大化。

1.1.2 研究意义

集装箱码头泊位与岸桥联合在线调度优化问题是跨学科、交叉性、综合性的研究课题，对该问题的研究具有一定现实意义和理论意义。

1) 现实意义

集装箱运输是现代物流的主要形式，集装箱码头作为其中的关键节点，承载着衔接海、陆运输的重任，其核心竞争力取决于如何合理利用自身的资源条件，快速而有效地进行集装箱装卸操作。其中，泊位和岸桥资源，通常是集装箱码头的稀缺及瓶颈资源，其协同作业效率决定了集装箱运转量。因此，合理有效的泊位及岸桥资源调度，可以降低码头的运营成本，并提高客户的满意度。

另外，我国大部分集装箱码头生产自动化程度较低，在处理泊位与岸桥调度时还是以经验为主，调度系统的准确性、稳定性、可靠性还较低。研究集装箱码头泊位与岸桥联合调度优化对提高我国集装箱码头运营效率具有实际意义。

2) 理论意义

本项研究所提出的集装箱码头泊位与岸桥联合在线调度优化，为集装箱码头前沿生产组织优化提供了一种新工具，弥补了现有调度优化方法在在线调度问题上研究的不足，进一步完善了泊位与岸桥联合调度理论。

另外，泊位与岸桥调度问题是典型的大规模、多约束和多目标冲突下的复杂优化问题，属于 NP 难题。研究此类问题的传统研究方法—精确求解该问题所需时间会随着输入的规模呈指数增长；而本文主要应用理论计算机科学兴起的在线竞争分析理论进行研究，这是近年来动态调度领域的一个热点研究方向。

1.2 基本概念

1.2.1 集装箱码头调度相关概念

集装箱港口是水陆联运的枢纽，包括码头、堆场、仓库及中转站等设施。其中，码头是港口运营最为关键的一个环节。码头调度的效率在很大程度上影响船公司的船舶周转率、客户集装箱送交计划、码头运营成本等^[1]。随着国际上诸多集装箱港口吞吐量需求的逐年增长，如何最大程度地优化码头调度方案以提高运转效率，这不仅仅是港口行业实践的迫切需求，而且也已经引起了国际学术界的极大兴趣^[2]。

码头调度效率主要受影响于泊位、岸桥以及转运集卡等因素，其中，泊位与岸桥属于战略性资源，也是调度作业的瓶颈性资源。这里，泊位是指码头水域集装箱船靠泊的地方；岸桥是指码头陆域前沿相邻于泊位布设的集装箱起重机；集卡为集装箱卡车的简称。各种资源的使用布局如图 1-1 所示。

船舶到港后的一系列过程包括船舶到港、分配靠泊位置、分配特定的岸桥、装卸集装箱以及船舶离港。泊位分配问题（BAP）指的是已知码头的泊位布局和在计划范围内需要被服务的船舶，以及船只的一些参数，通过一些时间或者空间上的约束，为每个船舶分配泊位和靠泊时间；岸桥分配问题（QACP）指的是给定可行的泊位计划和可用的岸桥，船只的装货量，最大最小岸桥数等，将岸桥分配给船只以满足所有货物的运转需要^[3]。泊位和岸桥联合调度是指在一段时间内，码头管理者根据船舶的相关信息和调度方案，为每个到港进行装卸作业的船舶分配适当的靠泊位置和靠泊时间，并分配适当数量的岸桥负责装卸，使码头作业达到期望的服务水平。

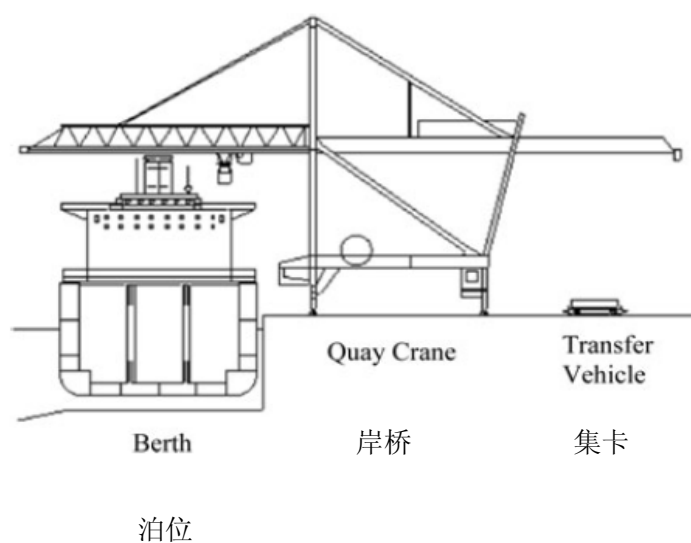


图 1-1：码头调度资源布局示意图

1.2.2 在线问题及其竞争策略的基本概念

本节将简单介绍在线问题及其竞争策略理论的一些基本概念,主要包括什么是在线问题以及什么是竞争比的概念,并举例说明,以此作为下文的理论基础。本质上讲,在线问题就是一类最优化问题,其目标就是最大化或最小化目标函数。

近年来在优化领域中兴起的在线问题(on-line problem)与竞争策略(Competitive Strategy)分析方法,为解决不完全信息下的决策问题提供了理论依据。把决策者只掌握当前信息,对未来既无确定性信息也无概率信息的条件下必须立即对当前状态做出决策的一类动态决策问题称为在线(on-line problem)问题。强调在变化因素的每个特例中都能给出解决方案,使之与相应离线(off-line)问题的最优方案差异总在一定比例范围内^[4]。在线问题及竞争策略理论最早应用在机器调度领域,其后慢慢发展到计算机领域和金融相关领域,直至今日,在经济管理和计算机科学领域中都有广泛研究。具有代表性的文献有:Kovalyov MY等^[5], Kolen AWJ等^[6]对在线排序问题进行了评述;也有大量文献对具有预知信息(look-ahead)的在线排序问题进行了讨论和研究^[7-8]。

通常,采用竞争比这一指标来度量在线策略的性能(Sleator等^[9])。这种方法以相应问题的离线(假设所有决策信息都完全已知)最优调度作为参照,用在线算法在最坏情况下的性能比来衡量该算法的好坏。考虑本文所研究的问题,对于任意请求输入序列 σ ,用 $C_{\max}(\sigma)$ 表示某一在线策略 A 的最大完工时间, $C^*(\sigma)$ 表示离线最优调度方案对应的目标函数值,则称 A 具有竞争比 ρ ,或者 A 是 ρ -竞争的,其中, $\rho = \inf_r \left\{ r \mid \frac{C_{\max}(\sigma)}{C^*(\sigma)} \leq r \right\}$ 。从其定义可以看出,竞争比反映了由

该算法构造的调度在性能上偏离最优调度的最大程度,也即反映了该算法在最坏情况下的性能比。由于缺乏问题的全局整体信息,一般情况下,在线算法都不能达到离线最优,但从在线的角度可能已经是最好的算法了,这就涉及到问题竞争比下界的概念。

给定一个在线调度问题,如果存在一个值LB使得该问题的所有在线算法的竞争比都不小于LB,则称LB为该问题所对应的任意在线算法的竞争比下界,在不引起混淆的情况下,简称为问题的竞争比下界。用数学表达式则是:令 Ξ 表示当前已给出的在线策略集合 Ξ , $\underline{c} = \arg \inf_{A \in \Xi} C_A$ 表示策略 $A \in \Xi$ 的竞争比 \underline{c} ,则竞争比下界 \underline{c} 定义为: $\underline{c} = \arg \inf_{A \in \Xi} C_A$ 。

如果一个在线算法的竞争比正好等于相应问题的竞争比下界,则称该算法为相应问题的最优在线算法。最优在线算法是指从算法的竞争比角度而言该算法是最优的,并不意味着由该算法构造的调度是最优调度,仅当问题的竞争比下界为1的情况下,最优在线算法才是离线最优的,即其构造的调度就是最优调度。需要

注意的是，问题的竞争比下界是针对求解该问题的所有在线算法而言的，而算法的竞争比下界，指的是由该算法构造的调度在性能上偏离最优调度的最小程度。

为了更好的解释在线问题，以下通过一个例子来说明：比如我们日常生活中的等公交问题中。假如你要从 A 地去往 B 地，有两种方法，一是等公交，一是打车。一般的，公交行驶缓慢，整个过程需要 30 分钟，而且公交何时能够到来并不知道，但公交的价格便宜，成本很低，而出租车速度较快，全程需要 10 分钟，并且可以看作是能够马上出发，但价格较贵，成本相对较高。因此，从你准备出发时刻起，每一时刻你都需作出决策：到底是等公交，还是打车。其结果直接影响到你的全局解。此时，你的目标是使得你的时间成本和交通费用总体最小。我们可以得出一个决策：先等待公交，一定时间后，若公交还未到来，则选择打车。该问题还可以延伸：从 A 地前往 B 地的方式除了乘坐公交和打车以外，还可以选择步行，那么问题将更加复杂。

1.3 集装箱码头调度的国内外研究综述

集装箱码头通常依据水岸地理条件设计泊位布局。中小型码头或杂货码头中的集装箱泊位往往由于水岸凹凸不规则而呈现为离散或者混合型泊位布局，如上海宝山码头和深圳妈湾港码头的集装箱专用泊位；大型或特大型（深水）码头依照直线水岸呈连续型或离散型泊位布局，如上海港洋山港区码头、深圳港蛇口国际集装箱码头等。根据Imai等^[10]对泊位的分类描述，连续型泊位的整条直线水岸可任意靠泊且各船舶首尾相接并保持安全距离，每个离散型泊位在任一时刻只能靠泊一艘船且大船只能停靠大泊位，而每个混合型泊位在任一时刻要么靠泊一艘大船要么同时靠泊几艘小船。Bierwirth与Meisel^[4]给出了更完整的分类，包括泊位空间属性（连续、离散及混合型）、处理时间属性（静态与动态）、优化目标属性（总逗留时间、总等待时间等）。在静态调度中，所有船舶决策前已抵港就绪或者抵港时间不迟于码头调度计划要求的靠泊时间，而在动态调度中，船舶具有不同抵港时间且只能在其抵港后安排靠泊。船舶在码头装卸的主要步骤是选择靠泊合适泊位并分配岸桥，再由集卡配合岸桥进行目标箱的装卸。由于选择靠泊位置及时间是一项重要决策内容，因而许多文献针对泊位分配（BAP: Berth Allocation Problem）问题进行探究；由于在BAP决策时船舶逗留时间依赖于靠泊位置及岸桥分配数量，所以许多学者研BAP-QCAP 联合调度以期给出总体优化方案^[11-12]。下面对BAP问题、BAP-QCAP问题的国内外研究进行综述。

1.3.1 泊位分配问题研究

泊位分配 (BAP) 问题是指如何安排集装箱船至指定的泊位并明确装卸作业时间段, 以最大化码头泊位的利用率。不同的码头由于海港地理特征不同其泊位结构设计不尽相同。下面根据不同泊位类型分两个小点阐述。

(1) 连续型泊位BAP问题。绝大多数研究则是探讨了动态情形。Xu等^[13]对动态连续型BAP问题也假设了船舶到达时间和处理时间为随机变量, 并给出了结合运用分支定界法与模拟退火法的启发式算法。张晋东^[14]针对船公司计划变化、天气等不确定因素容易导致泊位分配动态调整的现象, 构建了相应的鲁棒优化模型展开分析。魏晓东和杨智应^[15]探究了岸桥在服务船舶期间允许随时移动调整的情形, 给出了最大化岸桥利用率的分配调整方案。

(2) 离散或混合型泊位BAP问题。Guan和Yang^[16]探讨了静态离散型泊位BAP问题, 运用模拟仿真方法探讨泊位分配的各种方法; Xu等^[17]考虑了船舶吃水深度和涨退潮因素, 将其刻画为具有处理集合约束的平行机调度 (Parallel Scheduling) 问题。对于动态离散型泊位BAP的研究, De Oliveira等^[18]分析了考虑离港时间约束的动态离散型泊位BAP问题, 设计了一种结合模拟退火算法的混合族搜索算法来求解此问题。Ting等^[19]以最小化总服务时间为目标构建了混合整数规划模型, 并设计了粒子群算法。刘杰^[20]探讨了动态离散型BAP分配问题, 针对最小化总逗留时间的优化目标设计了遗传算法进行求解, 并以广州港码头的运行数据进行了实例验证。Golias等^[21]考虑了船舶抵港时间与服务时间的不确定性, 将其假设为随机变量展开分析。对于动态混合型泊位BAP问题, Moorthy和Teo^[22]着重探讨了服务水平与调度成本之间的均衡, 其中假设船舶具有随机抵港时间。Umang等^[23]探讨了以最小化总服务时间为优化目标的混合整数规划模型。Cheong等^[24]运用多目标规划方法来探讨动态混合型BAP问题, 优化目标涉及船舶逗留时间 (makespan, 亦或船只停靠泊位的时长)、总等待时间、船舶实际服务优先序与计划偏离度等, 文中设计了一种多目标进化算法进行求解。

1.3.2 泊位-岸桥联合调度 (BAP-QCAP) 问题研究

在码头调度实践中, BAP、QCAP 这两个调度问题是密切关联的, 船舶逗留泊位的时长很大程度上决定于所分配的岸桥数量, 而岸桥数量又影响着岸桥分配与岸桥调度方案; 同时, 岸桥分配与调度效果又影响泊位分配计划能否如期履行。许多学者开展 BAP 问题与 QCAP 问题的联合调度研究, 以期给出整体优化调度方案。在码头调度实际作业过程中, 泊位分配问题和岸桥调度问题是紧密联系的两部分。岸桥并不是固定在某个位置不动的, 而是可以沿码头岸线作水平移动船舶的作业时间会随着作业岸桥的变动而发生变化。因此, 在研究泊位和岸桥调度问题时, 需要同时考虑泊位和岸桥, 进行泊位和岸桥联合调度研究。

(i) 静态 BAP-QCAP 问题

静态BAP-QCAP 问题是指所有船舶决策前均已抵港就绪或抵港时间能够迎合码头调度计划的靠泊时间的情形。静态BAP-QCAP的研究文献基本上探讨连续型泊位。Park 和Kim^[25]假设船舶服务时间依赖于岸桥分配数量,以最小化船舶实际靠泊点与最佳位置的偏离量、靠泊延迟时间,以及离港延迟时间等为优化目标构建了多目标规划。Theofanis等^[26]构建了一个混合整数规划模型,并给出了考虑船舶靠泊时间、靠泊位置及岸桥分配数量的一种最佳方案。Oguzd等^[27]引入平行机调度模型来刻画这一问题,其考察目标是最小化船舶的最大离港时间。Meisel和Bierwirth^[28]假设岸桥在服务过程中可以随时移动调整,每艘船的装卸任务量采用“岸桥个数 \times 小时数”的方式计量且船舶实际处理时间是靠泊位置与最大可分配岸桥数的非线性函数。Meisel和Bierwirth^[29]进一步结合了岸桥装卸调度环节提出了三阶段决策方法,阶段一估计岸桥产能,阶段二结合岸桥产能信息进行泊位和岸桥的联合分配决策,阶段三进行岸桥装卸调度决策。Hu等^[30]、高超锋和胡志华^[31]结合考虑船舶燃料消耗与排放以及船舶等待期间的排放成本因素,构建了非线性多目标混合整数规划模型。

(ii) 连续型泊位动态 BAP-QCAP 问题

Rodriguez-Molins等^[32]、乐美龙和刘秀玲^[33]、桂小娅等^[34]对船舶总等待时间、总逗留时间等优化目标构建混合整数规划并设计启发式算法。Turkogullari等^[35]对于包括船舶靠泊位置偏离值、等待时间和离港延迟时间的总成本优化目标构建整数线性规划模型并设计了可求解60艘船舶规模的精确算法。Yang等^[36]运用两阶段决策思想,采用两个内循环分别求解BAP和QCAP子问题。杨华龙和腾川川^[37]对于总逗留时间、岸桥利用率的双优化目标,引入装箱问题的分析思路设计了一种“挤压”算法。Zhang等^[38]假设在船舶装卸过程中允许少量岸桥移动调整,并给出了次梯度优化求解算法。曾庆成等^[39]着重探讨了调度过程中各种不确定干扰因素对调度方案的影响情况以及应对策略,建立了两阶段干扰管理决策模型。彭丽姣和韩晓龙^[40]建立了以最小化延迟完工任务量、偏离最佳靠泊位置和岸桥移动的三种惩罚费用之和为目标的整数规划模型,并进行了算例分析。该文将泊位长度划分与时间划分均进行了一定程度的离散化假设,并假定岸桥可以在船舶之间进行实时移动以提高其利用率。有一些学者采用滚动时域(Rolling-horizon)方法进行探讨。Chang等^[41]提出了通过滚动调度策略模型和目标规划解决集装箱码头泊位与岸桥动态调度问题,采用并行遗传算法来解决所提出的模型,旨在减少停泊位置偏差、总惩罚成本与码头起重机的能源消耗;并探讨了每个滚动时域跨度为3天但每次决策只前移半天的一种调度方法。Raa等^[42]结合了船舶优先级、最佳靠泊位置及处理时间等因素,运用混合启发式算法来优化每个滚动时域的长

度。Xiao和Hu^[43]在设计滚动时域求解算法时将已抵港的船舶根据服务状态划分成三类进行分析。

(iii) 离散或混合型泊位动态BAP-QCAP问题

对于离散型泊位的情形, Giallombardo等^[44]、Blazewicz等^[45]、Zhen等^[46]和Chen等^[47]对于最大离港时间、最大延误时间等多种优化目标构建混合整数规划模型, 并设计了禁忌搜索等启发式算法或者精确求解算法。杨春霞和王诺^[48]研究了最小化船舶在港时间与码头调度成本的双目标模型, 并用多目标遗传算法进行求解。Imai等^[49]探讨了每艘船舶对岸桥分配数量有特定要求而且装卸时间取决于靠泊位置的情形。Liang等^[50]分析了岸桥允许随时移动调整且船舶装卸时间取决于岸桥数量的情形, 对多目标模型设计了混合遗传算法, 针对4个相邻泊位和7个岸桥的资源组合进行了算例验证。Vacca等^[51]假设船舶服务时间取决于所分配岸桥数量, 对所构建模型设计了分枝定价精确算法。孙彬等^[52]考虑了存在不确定因素的情形, 并设计了具有系统鲁棒性的调度策略, 其优化目标为最小化所有船舶总延误时间。文中给出了基于某种协商机制的岸桥实时调度模型, 通过仿真实验验证了所设计策略的有效性。对于混合型泊位动态BAP-QCAP问题的研究文献非常少, Lokuge与Alahakoon^[53]运用多代理系统的方法对这一模型进行了刻画与分析。

以上是对集装箱港口码头作业中的泊位分配、泊位-岸桥联合调度两类问题已有研究的分类综述。根据上述文献可以总结归纳出如下几个特征:

1) 虽然泊位与岸桥的联合调度研究近年来涌现出了许多成果, 但对于需求到达时间具有很强不确定的情形, 目前相关成果极少, 针对该情形下的联合调度问题有待于展开深入探讨。

2) 近年来较多文献研究的是船舶抵港时间影响决策的动态情形, 然而在动态决策中, 现有研究基本上假定所有船舶预报到达码头的时事先已知, 主要运用动态规划、整数规划或混合规划方法来刻画与分析问题。虽然有一些学者考虑到了执行过程中可能出现的各种干扰因素会使船舶计划到达时间与实际到达时间出现偏差, 但在模型中假设了干扰因素服从某种统计特征分布。实际码头作业当中, 这些干扰因素可能并不符合任何分布特征, 从而使船舶到达码头的时很大程度上具有难以预估的不确定性。针对这种需求具有很强动态特征且不满足统计特征的不确定情形如何建模研究, 现有的相关成果极少, 只有Zhang等^[54]利用在线理论工具进行了初步研究。

3) 泊位—岸桥调度问题是 NP 难题, 难以用精确算法求解。目前, 求解该类问题大多数文献采用启发式算法, 但启发式算法大多依赖由实际经验制定的启发式规则, 计算速度较快, 但算法设计往往依赖于具体问题, 普适性较差, 很容易陷入局部极小值, 难以保证得到全局最优的计算结果。

综上, 本论文将针对集装箱船舶到达具有动态性以及不完全信息调度决策的特点, 重点探讨BAP-QCAP 联合调度问题, 针对混合型泊位类型以及动态调度情形, 运用在线理论工具来构建BAP-QCAP联合实时调度的优化模型, 旨在于设计实用高效的调度算法, 为集装箱码头的动态调度方案设计做好铺垫。

1.4 研究方法、思路与内容

1.4.1 研究方法

在码头资源调度问题的模型描述形式上, 将借鉴使用 Bierwirth与Meisel^[4]给出的4 参数分类法: $\alpha|\beta|\gamma|\Gamma$ 。其中, α (=hybr) 参数表示泊位属性 (=混合型); β (=online) 参数表示船舶信息属性 (= 静态、在线到达); γ (=BAP、BAP-QCAP) 参数表示泊位、岸桥资源的组合属性 (=泊位分配模型、泊位与岸桥联合调度模型); Γ (= max(compl)) 参数表示目标属性 (=最大离港时间)。对于拟开展的项目内容, 相关研究方案与思路如下:

本文研究的理论出发点为集装箱码头泊位与岸桥联合调度中的优化问题以及在线问题及其竞争策略理论。应用在线问题及其竞争策略理论研究了集装箱码头泊位与岸桥实时调度过程中基于总完工时间最小化的优化问题。

与传统的组合优化问题和调度问题所使用的诸如线性规划、启发式算法和遗传算法等方法不同, 本文采用了在线问题及其竞争策略理论中的有关方法。竞争分析方法与以往解决此类问题方法的最大区别在于: 它在变化因素的每一个特例中都能给出一个方案, 使得这一方案所得到的解与最优方案给出的解总在一定的比例之内, 从而使在线问题的解始终保持在一个较优的状态。

本文应用最坏情形分析法来处理集装箱码头泊位与岸桥联合在线调度问题。具体来讲, 利用最坏情形最优的方法, 假设最坏情形存在, 努力使得最坏情形最好。主要应用了归纳和演绎的方法, 配合大量的公式和严格的逻辑证明, 详细阐述了相关理论和结果。下面简单地介绍最坏情形分析法。

最坏情形分析法 (Worst Case Analysis)。最坏情形分析法是证明占线策略竞争比的主要方法之一, 其核心思想如下。不妨考虑最小化完工时间的某个在线问题, 要证明某个在线策略 OLP 的竞争比, 只需证明在 OPT 针对 OLP 所设计的最坏输入序列中 OLP (On-line Player) 与 OPT 的总完工时间之比即可^[9]。用数学具体描述如下: 令 $C_{OLP}(\sigma)$ 和 $C_{OPT}(\sigma)$ 分别表示对于问题的某一个输入序列 σ , OLP 与 OPT 各自需要完成序列请求的总完工时间。如果存在任意其他输入序列 σ' , 都有以下不等式成立,

$$C_{OLP}(\sigma)/C_{OPT}(\sigma) \geq C_{OLP}(\sigma')/C_{OPT}(\sigma')$$

则称 σ 为该问题的最坏输入情形, 并将 $c = C_{OLP}(\sigma) / C_{OPT}(\sigma)$ 称为策略 OLP 的竞争比。

需要注意的是最坏情形分析法是将 OPT 与 OLP 两个策略在同一个输入序列中的总完工时间进行比较。当运用最坏情形法来证明 OLP 策略的竞争比时, 只需分析在其最坏情形下 OPT 与 OLP 的完工时间之比。所以, 由此引出的相关问题是, 如何寻求或者判定某一个输入序列就是针对 OLP 策略的最坏序列。

1.4.2 研究思路

本论文中集装箱码头岸桥与泊位的联合调度问题与并行机调度 (Parallel Machines Scheduling) 有着较高的相似度。我们对集装箱码头岸桥与泊位联合调度问题的分析也借鉴了近年来在并行机调度所研究的热点问题, 如在线和半在线排序问题。考察不完全信息下的动态 BAP-QCAP 模型。引入在线理论方法工具将问题刻画为有限预知能力下的资源实时调度模型, 借鉴并行机调度理论中考虑预知能力下的相关研究思路, 设计码头资源联合调度算法。

船舶抵达港口, 首先进行泊位的分配, 此时泊位作为并行机器为船舶提供靠泊服务。若船舶占用的是泊位上的多个单位距离, 可认为船舶在靠泊阶段需要占用多个并行机器。接下来, 给船舶分配岸桥, 此时岸桥作为并行机器为船舶进行装卸服务。船舶需要占用至少一台岸桥, 甚至四台岸桥同时作业。在这个并行机调度阶段中, 与泊位分配相类似, 每艘船舶同样需要占用一台以上的机器。同时由于岸桥位于码头岸线的同一轨道上, 不能交叉跨越, 这一阶段岸桥的分配受到第一阶段靠泊位置的影响, 可供选择的机器属于一个可行的岸桥子集。

对于不完全信息下的动态 BAP-QCAP 模型, 考虑的是每次决策时刻并不掌握所有未来抵港船舶信息的情形, 引入在线理论分析工具将问题刻画为有限预知能力下的并行机在线调度模型。该模型假设在决策零时刻可以预知后续 l ($l = 1, 2, \dots$) 艘船舶的抵港时间及需求情况。关于模型优化目标首先考虑 min-max(compl)。在码头资源在线调度问题中每个泊位的处理速度并不确定, 它受制于泊位的岸桥配置数量、船舶靠泊偏离值, 以及船舶长度。同时, 在线理论分析方法的优点在于通过分解原问题有效地降低计算时间复杂度, 且能保证求解效果控制在距离最优方案的一定范围内。本文对混合型泊位 BAP-QCAP 问题假设同一型号的船舶具有相同装卸需求量的情形, 对于各种岸桥配置数量均设计给出了具有最优竞争性能的在线调度算法。

在描述集装箱码头泊位与岸桥调度问题背景与研究意义的基础上, 主要运用在线问题与竞争策略理论和方法对各种特定泊位与岸桥联合调度情形进行剖

析,建立相应的在线调度模型,并主要利用最坏情形分析法对竞争比的上、下界进行分析证明。

1.4.3 研究内容

基于对国内多个集装箱码头的调研分析,结合码头泊位实际特征,刻画出相应的泊位分配模型、泊位-岸桥联合调度模型。首先,对于中小型集装箱码头或者杂货码头中的集装箱专用泊位,将其凝练为离散型或混合型泊位下的调度模型;其次,根据船舶抵港时间因素,码头工作方式实时滚动制定船舶装卸计划,则刻画为动态调度模型。具体研究内容包括:中小型码头离散或混合型泊位BAP-QCAP问题。

考察不完全信息下的动态BAP-QCAP模型。引入在线理论方法工具将问题刻画为有限预知能力下的资源实时调度模型,借鉴平行机调度理论中考虑预知能力下的相关研究思路,设计码头资源联合调度算法。

各章节具体内容安排如下:

第一章,绪论。首先概述集装箱码头泊位与岸桥联合调度问题的研究背景与意义,然后,介绍有关理论知识并综述国内外相关研究情况;简要阐述研究方法。最后,给出论文研究思路和内容框架。

第二章主要是分析无预知信息下泊位与岸桥联合在线调度。考虑1个混合型泊位布局且存在大小两种服务请求的情形。在分析问题中,假定有三个泊位,针对四个岸桥、五个岸桥、六个岸桥的情形,分别设计在线调度策略,并分析其竞争性能。本章分析结论同时也为后续各章的问题研究提供理论基础。

第三章提出了预知 $k=1$ 个请求下泊位与岸桥联合在线调度,在分配每艘船舶服务请求时预知后续一个请求的信息,要求在完成所有服务请求的分配后最小化最大完工时间。也探讨1个混合型泊位布局且存在两种大小船舶服务需求的情形。对于配置4个、5个、6个岸桥的三种情形,针对模型中预知信息的特征分别设计在线调度策略,并分析其竞争性能;并与第二章得出的结论做比较。

第四章在第三章的基础上,进一步分析预知 $k=2$ 个请求下泊位与岸桥联合在线调度模型,在分配每艘船舶服务请求时预知后续 k 个请求的信息。对于配置4个、5个、6个岸桥的三种情形,分别设计在线调度策略,并分析其竞争性能;并与前两章得出的结论做比较。

最后,第五章对论文的主要工作与创新点进行总结,总结本文研究与结论,然后给出有待于进一步研究的几个问题。

1.5 技术路线

本论文拟开展研究的技术路线如下图1-2所示。

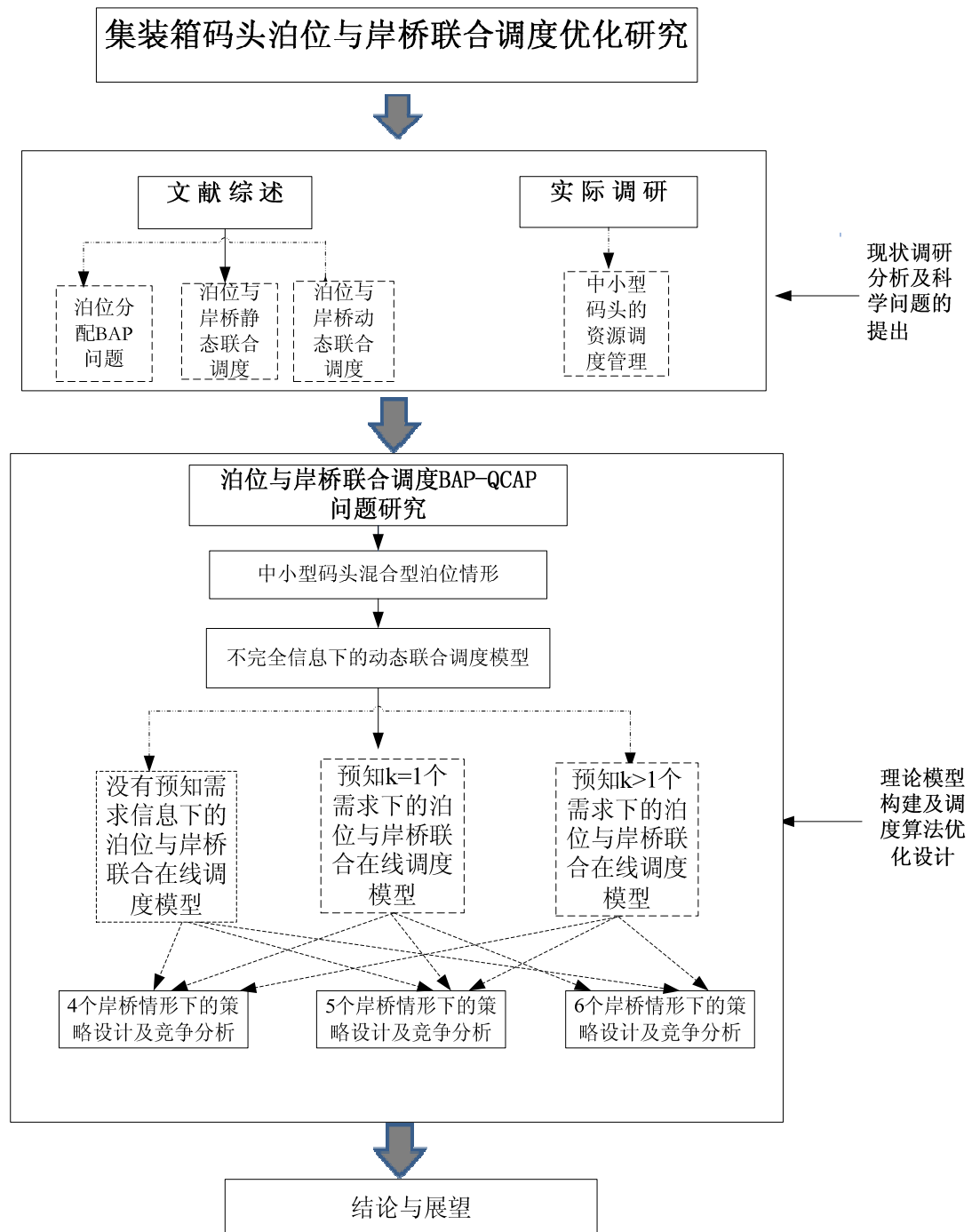


图1-2: 论文技术路线图

在调研与文献综述部分,首先将对国内包括上海洋山港码头、深圳蛇口港码头、妈湾港码头等集装箱码头开展充分的实践调研活动,掌握中小型港口不同类型码头的调度业务实际资料。同时,对已有的研究文献进行分类归纳与总结分析。在此基础上凝练出相应的关键资源调度优化问题。

在理论研究部分,着重对BAP-QCAP问题进行深入探讨,对于中小型码头的不同泊位结构,结合船舶抵港时间因素构建动态调度模型,采用在线理论竞争分析方法进行建模分析与算法设计。

1.6 工作的创新点

本文的创新点在于：

第一，本文所提出的在线策略具有很强的实际应用价值，因为在实际情形中对需求的快速反应有时往往比求得最优解更为迫切。

第二，将问题分为多个阶段多个子问题求解，算法的运行时间复杂性都很低，其次，对其解的质量度量是对其最坏情形下的解与最优解相比，因而算法的性能优劣一目了然。

第三，突破 BAP 与 BAP-QCAP 问题现有研究设计复杂启发式算法的主导思路，创新性地利用在线理论分析工具来刻画剖析不完全信息下的泊位与岸桥动态联合调度，构建具有前瞻能力的在线 BAP-QCAP 联合调度模型。所设计在线算法不仅无需完全的需求信息进行决策，而且具有明显低于精确算法及多数启发式算法的时间复杂性，还可通过与最优解的比较严格度量其性能优劣。

第2章 无预知信息下泊位与岸桥联合在线调度

由 1.3 节可知,假设所有的船舶请求事先完全已知条件下的 BAP-QCAP 已经有很深入的研究;本章针对不确定环境下动态抵港的船舶和加班船舶,着重探讨了无预知信息下泊位和岸桥联合在线调度问题,本章采用 Zhang 等^[54]的建模方法,运用在线理论来刻画码头泊位与岸桥的联合调度,并根据船舶动态到达特征构建相应的 over-list 在线模型。主要思路是提出泊位和岸桥联合调度的模型和算法,将依次介绍所研究的问题描述、所建立的在线模型以及模型的证明推理。

2.1 问题描述与模型刻画

本章研究的是无预知信息下泊位和岸桥联合在线调度问题,目的是探究如何减少不确定因素对码头作业的影响,从而提高集装箱码头资源的利用率。对于码头来说,泊位与岸桥都是非常稀缺的瓶颈资源,只有把泊位与岸桥合理地分配给每一艘抵港的船舶,才能提高客户的满意度以及降低码头运营成本。泊位与岸桥联合调度的核心问题在于确定每一艘到港船舶的靠泊时间、服务时间区段、靠泊位置和岸桥分配方案,从而使得所有到港船舶的服务(装卸)完工总时间最小化。

一般地,为了使数学模型便于理解,泊位与岸桥的联合调度方案可以表示为离散化的空间—时间二维坐标图,如图 2-1 所示。其中,横坐标轴表示船舶靠泊位置,图中 b_1, b_2, b_3 表示左、中、右 3 个离散且相邻的小泊位,其中, b_1, b_2 和 b_2, b_3 是连续泊位,可以停靠大船舶。纵坐标表示时间轴和靠泊时间。图中的每个矩形表示计划期内靠泊作业的船舶,矩形中船舶的字母代表该船在调度方案中的编号;矩形的高度代表该船舶的服务时长(作业时间);矩形的左、右边界表示该船实际靠泊位置 b_i 。因为在实际作业中,两艘船舶在时间上或空间上不会存在冲突,因此,两个矩形之间不会出现重叠部分;另外,在二维平面内,由于调度方案出现的空隙区域表示资源的浪费。

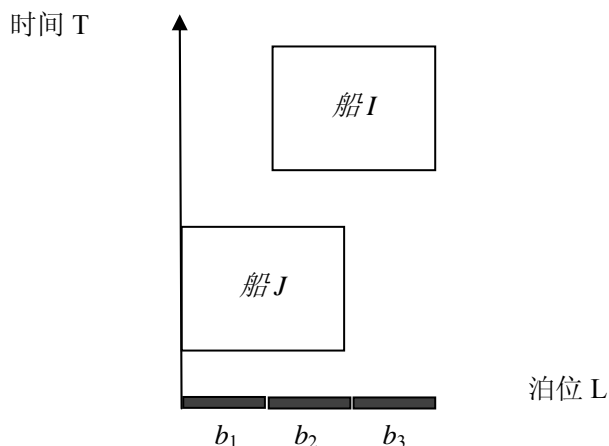


图 2-1: 离散化的泊位-时间二维坐标图

在不确定环境下,主干船舶和临时加班船实际到港时间和装卸箱量都存在不确定性,忽视随机因素的调度方案往往脱离码头实际而难以执行,给码头调度作业计划带来很大的困难。通过调研发现,在现实中因为临时加班船的大量存在,即使是大型深水码头如上海小洋山港的一二三期码头,其船期表也是每天动态地进行调整并安排新抵港的加班船,在其排期密度表中加班船数量甚至占据三分之一以上的比例。一般地,主干船舶提前多个工作日即可确定其准确抵港日期,码头在安排装卸之前要求其提早 24 小时抵港并在锚地等待;而加班船往往无法实现上述要求,常常提早一两天预报抵港需求,因而其服务请求的实时性特征十分明显。对于码头而言,在加班船需求信息提早预知时间量较小的前提下,仍然尽力接纳并服务加班船从而提升码头的资源利用率和吞吐量。当然,主干船舶的抵港日期也存在着一些不确定性,比如因恶劣天气因素导致抵港日期发生临时性的变更等。

对于这些船舶来说,作业员一般按先到先服务的原则,制定作业计划列表去执行已到达的船舶请求,当作业员对当前船舶进行资源的分配决策时,不在计划列表中的那些未到达船舶的信息是未知的。因此,我们在先到先服务分配模式(FCFS)的基础上讨论了泊位与岸桥在线联合调度问题,并分析了这种策略算法的性能。

基本假设:

本文考虑集装箱码头的1个混合型泊位,该泊位可以同时停靠1艘大船和1艘小船,或者同时停靠3艘小船。为了方便表述,将这一混合型泊位刻画为左、中、右3个离散且相邻的小泊位,从而小船(小请求)占用1个泊位而大船(大请求)需占用2个相邻泊位。在online-over-list在线模型下,服务请求序列 $\sigma = \{r_1, r_2, \dots, r_n\}$ 中的 n 个请求在零时刻逐一释放,每个请求 r_i 释放时必须马上将其分配至某个(或某两个)泊位和相应的1个(或多个)岸桥,以及服务的时间区段。

本文研究基于以下9条假设:

- (1) 所有岸桥共享一条平行于泊位的移动轨道,因而每个岸桥的左右相对位置始终保持不变;且每一台岸桥都有特定的泊位服务范围,并且相互之间不能跨越,从左至右的岸桥记为 q_1, q_2, q_3, \dots ;
- (2) 系统中只有大小两种请求(船舶),小请求任务量(装卸量)为1,记 $r_i = 1$;大请求任务量 $\Delta \geq 2$,记 $r_i = \Delta$;
- (3) 泊位轴被划分为3个离散的等额泊位段;系统从左至右三个离散泊位记为 b_1, b_2, b_3 ;
- (4) 在任一时刻,每个泊位、岸桥只能服务一个请求(只能被一艘船舶占用);
- (5) 小请求占用1个泊位且至多分配2个相邻的岸桥,大请求占用2个相邻泊位且至多分配4个相邻岸桥;

- (6) 若将请求 r_i 分配给 $m_i \geq 1$ 个岸桥, 则它的服务时间等于 r_i / m_i , 即每艘船舶的装卸时间=装卸量 (任务量) / 作业岸桥数量;
- (7) 泊位和岸桥一经分配就不再改变, 直到该船舶作业结束为止, 且服务过程中船舶不移动;
- (8) 忽略岸桥移动过程带来的时间和效率损失;
- (9) 本文不考虑甩港, 即每个请求必须被分配。

运用 Bierwirth 和 Meisel^[4] 的四元刻画方法, 记本章所研究的在线模型为 $hybr | online-over-list | BAP-QCAP | C_{\max}$, 其中, $hybr$ 代表混合型泊位; $online-over-list$ 是指请求逐个释放的在线模型; 在该模型下, 船舶需求信息是一个接一个逐次到达, 决策者需要在每个需求到达时刻就做出决策, 而决策一旦做出就不能改变的情形; 每个需求释放时需立即将其分配至具体泊位及配置相应的岸桥, 以及确定服务时间段。在无预知信息情形下, 只有在当前船舶到达时刻, 才能根据该请求 (船舶) 的相关信息对当前船舶进行调度, 而决策者对未来船舶的信息是无法预知的。BAP-QCAP 指泊位和岸桥的联合调度; C_{\max} 表示最大完工时间的目标函数。

接下来本章节针对整个序列服务请求总完工时间最小的目标, 重点讨论了 3 个相邻泊位, 4、5、6 个岸桥的情形。对于每个情形, 首先讨论了各自情形的竞争比下界值; 接着结合各自的下界值, 分别设计联合调度的在线策略, 并分析其竞争性能, 对于任意组合的请求序列, 在线策略都能使其完工时间与离线最优值的比值尽可能的小。

2.2 四个岸桥的调度情形

2.2.1 竞争比下界分析

下面针对 4 个岸桥的情形, 给出该情形竞争比下界的相关结论。

定理 2-1. 当 $n_q = 4$ 时, 问题 $hybr | online-over-list | BAP-QCAP | C_{\max}$, 不存在任一在线策略具有竞争比小于 $5/4$ 。

证明. 要证明该定理, 只需设计一个请求输入序列 σ , 使得对于任一在线策略 A 有不等式 $C_{\max}(\sigma) / C^*(\sigma) \geq 5/4$ 成立, 其中 $C_{\max}(\sigma)$ 和 $C^*(\sigma)$ 分别为策略 A 的最大完工时间、离线最优加工方案下的最大完工时间。

序列 σ 包含至多 2 个请求, 第一个请求 $r_1 = 1$ 。若策略 A 只调度 1 台岸桥服务该请求, 则序列 $\sigma = \{r_1\}$ 。 $C_{\max}(\sigma) \geq 1$, 最优调度方案将调度两台岸桥从 0 时刻开始服务该请求, 即 $C^*(\sigma) = 1/2$ 。因此, $C_{\max}(\sigma) / C^*(\sigma) \geq 2$ 。反之, 若策略 A 调度两台岸桥来服务 r_1 , 则 $r_2 = \Delta$ 且序列 $\sigma = \{r_1, r_2\}$ 。由于总共只有 4 台岸桥,

策略A要么调度2台岸桥最早从0时刻开始服务 r_2 ，要么调度4台岸桥最早从1/2时刻开始服务 r_2 ， $C_{\max}(\sigma) \geq \min\{\Delta/2, 1/2 + \Delta/4\}$ 。最优方案则是分别调度1、3台岸桥给请求 r_1 、 r_2 ，并从0时刻起不间断地服务两个请求， $C^*(\sigma) = \max\{\Delta/3, 1\}$ 。令 $\Delta=3$ ，则 $C_{\max}(\sigma)/C^*(\sigma) \geq 5/4$ 。定理得证。证毕

通过定理2-1可知，在线策略的竞争比至少为5/4。

2.2.2 在线策略GR设计及分析

在该模型中，在线策略GR（Greedy）将最左边的两个泊位 b_1, b_2 分别配置岸桥 q_1, q_2 和 q_3, q_4 ，而最右边的泊位 b_3 一直处于空闲状态。基于上述对岸桥与泊位两种资源的固定配置方案，在线策略GR只需刻画分配每个请求至哪一个或两个泊位以及服务的启动时间即可。下面给出在线策略GR的描述：

策略GR：

C1 当 $r_i = \Delta$ 时：令 $s_i = \max\{c_{i,1}, c_{i,2}\}$ ，把 r_i 分到左边的两个泊位。如果 $C_{i,1} \geq C_{i,2}$ ，则令 $T_a = [t_1, t_2) = [C_{i,2}, C_{i,1})$ ， $b_T = b_2$ ；如果 $C_{i,1} < C_{i,2}$ ，则令 $T_a = [t_1, t_2) = [C_{i,1}, C_{i,2})$ ，令 $b_T = b_1$ 。

C2 当 $r_i = 1$ 时：如果 $T_a \neq [0, 0)$ ，把 r_i 分到空闲泊位 b_T 上的空闲间隔 T_a ，再令 $T_a = [0, 0)$ ；当 $T_a = [0, 0)$ ，如果 $C_{i,1} \geq C_{i,2}$ ，令 $s_i = C_{i,2}$ ，把 r_i 分到泊位 b_2 ；如果 $C_{i,1} < C_{i,2}$ ，令 $s_i = C_{i,1}$ ，把 r_i 分到泊位 b_1 。

根据上述策略描述可知，泊位 b_3 一直处于空闲；当 $T_a = [t_1, t_2) \neq [0, 0)$ 时， $t_2 - t_1 = 1/2$ ，因为在泊位 b_1 和 b_2 都是配置两个岸桥，每个小请求需要1/2单位时间。在空闲间隔 T_a ，可用的岸桥资源为 $2 * (1/2) = 1$ 。接下来分析该策略的竞争性能。

定理2-2. $n_q = 4$ 时，问题 $hybr | online - over - list | BAP - QCAP | C_{\max}$ 问题，在线策略GR的竞争比为5/4。

证明. 对于任何输入序列为 $\sigma = \{r_1, r_2, \dots, r_n\}$ ，令 $C_{\max}(\sigma)$ 为策略GR的最大完工时间； $C^*(\sigma)$ 为离线最优加工方案下的最大完工时间。

Case 1. 最后一个请求是大请求。

Case 1.1. 当 $T_a = [0, 0)$ ，因为在 $[0, C_{\max}(\sigma))$ 没有岸桥资源的浪费，则有 $C_{\max}(\sigma) = C^*(\sigma) = C_{n,1} + \Delta/4$ 。竞争比显然为1。

Case 1.2. 当 $T_a = [t_1, t_2) \neq [0, 0)$ ，假设在 t_2 时刻后有 $k (\geq 1)$ 个大请求到达，则 $C_{\max}(\sigma) = t_2 + k\Delta/4$ ；如果 $t_2 = 1/2$ 和 $k = 1$ ，说明总共有两个请求， $r_1 = 1$ 和 $r_2 = \Delta$ ，

此时 $C^*(\sigma) \geq \min\{1/2 + \Delta/4, \max(1, \Delta/3)\}$, 则有 $C_{\max}(\sigma)/C^*(\sigma) \leq 5/4$ 。如果 $t_2 \geq 1$ 或 $k \geq 2$, $C_{\max}(\sigma) = t_2 + k\Delta/4 - 1/4$, 那么 $C_{\max}(\sigma)/C^*(\sigma) \leq 6/5$ 。

Case 2. 最后一个请求是小请求。显然 $T_a = [0, 0)$; 不然, 当 $r_i = 1$ 时: 如果 $T_a \neq [0, 0)$, 把 r_i 分到空闲泊位 b_T 上的空闲间隔 T_a 。

Case 2.1. 如果 $C_{n,1} = C_{n,2}$, 在这种情形下, $C_{\max}(\sigma) = C_{n,1} + 1/2$ 。如果请求只有1个或3个连续小请求, 则有 $C_{n,1} \leq 1/2$, $C_{\max}(\sigma) = C^*(\sigma)$; 如果请求只有两个, 那么 $r_1 = \Delta$, $C_{n,1} = \Delta/4$, 此时 $C^*(\sigma) \geq \max(1, \Delta/3)$, $C_{\max}(\sigma)/C^*(\sigma) \leq 5/4$; 如果 $C_{n,1} \geq 1$, $C^*(\sigma) \geq C_{n,1} + 1/4$, 则 $C_{\max}(\sigma)/C^*(\sigma) \leq 6/5$ 。

Case 2.2. 如果 $C_{n,1} < C_{n,2}$ (或 $C_{n,2} < C_{n,1}$), 根据策略分析则有 $C_{n,2} = C_{n,1} + 1/2$ (或 $C_{n,1} = C_{n,2} + 1/2$) 则有 $C_{\max}(\sigma) = C^*(\sigma)$ 。定理得证。证毕

通过对定理2-2的证明可知, 在线策略GR的竞争比为5/4; 结合定理2-1中, 该问题不存在在线策略使得竞争比小于5/4, 可知, GR策略是最优的。

2.3 五个岸桥的调度情形

对于5个岸桥的情形分析, 本节先讨论了该问题的竞争比下界, 并结合此下界值试着去设计相应最优的在线策略; 针对设计的 LRSS 策略, 并证明其是最优的。

2.3.1 竞争比下界分析

下面针对5个岸桥的情形, 给出竞争比下界的相关结论。

定理2-3. 当 $n_a = 5$ 时, 问题 $hybr | online - over - list | BAP - QCAP | C_{\max}$, 不存在在线策略的竞争比下界小于4/3。

证明. 类似于定理2-1的证明, 同样定义 $C_{\max}(\sigma)$ 和 $C^*(\sigma)$, 且只需设计一个请求输入序列 σ 使得对于任一在线策略 A 有 $C_{\max}(\sigma)/C^*(\sigma) \geq 4/3$ 成立。

序列 σ 包含至多2个请求, 第一个请求 $r_1 = 1$ 。若策略 A 只调度1台岸桥服务该请求, 则序列 $\sigma = \{r_1\}$ 。 $C_{\max}(\sigma) \geq 1$, 最优调度方案将调度两台岸桥从0时刻开始服务该请求, 即 $C^*(\sigma) = 1/2$ 。因此, $C_{\max}(\sigma)/C^*(\sigma) \geq 2$ 。反之, 若策略 A 调度两台岸桥来服务 r_1 , 则 $r_2 = \Delta$ 且序列 $\sigma = \{r_1, r_2\}$ 。由于总共只有5台岸桥, 策略 A 要么调度3台岸桥最早从0时刻开始服务 r_2 , 要么调度4台岸桥最早从1/2时刻开始服务 r_2 , $C_{\max}(\sigma) \geq \min\{\Delta/3, 1/2 + \Delta/4\}$ 。最优方案则是分别调度1、4台岸桥给请求 r_1 、 r_2 , 并从0时刻起不间断地服务两个请求, $C^*(\sigma) = \max\{\Delta/4, 1\}$ 。令 $\Delta = 6$, 则 $C_{\max}(\sigma)/C^*(\sigma) \geq 4/3$ 。定理得证。证毕

通过定理 2-3 可知, 在线策略的竞争比至少为 4/3。

2.3.2 在线策略 LRSS 设计及分析

接下来,针对无预知情形,具有5个岸桥的联合调度模型,设计在线策略LRSS该策略的主要思想:将最左边的两个泊位 b_1, b_2 分别配置岸桥 q_1, q_2 和 q_3, q_4 ,而最右边的泊位 b_3 配置最右边的1个岸桥 q_5 。因此,若某个大请求被分配至泊位 b_2, b_3 ,则将右侧3个岸桥 q_3, q_4, q_5 进行服务。策略的基本思想是尽可能地把大请求分到左边的两个泊位,并安排四个岸桥去服务它;但以下两种情形除外:1) $r_i = \Delta$; 2) $r_i = 1$ 和 $r_2 = \Delta$ 。基于上述对联合资源的固定配置方案,在线策略只需刻画分配每个请求至哪一个或两个泊位以及服务的启动时间即可。下面给出在线策略LRSS (Large request of scheduling is special) 的描述:示意图如图2-2所示。

策略LRSS:

C1. $r_i = \Delta$ 。按如下三种情形进行调度,如果不是特殊情形的话,把 r_i 分到左边的两个泊位 b_1, b_2 。

C1.1. $t_{i,1} = t_{i,2} \leq t_{i,3}$, 如果 $t_{i,2} = 0$, 令 $s_i = 0$, 把 r_i 分到右边的两个泊位。如果 $C_{i,1} = t_{i,3}$ ($C_{i,3} = t_{i,3} > t_{i,1}$), 令 $s_i = t_{i,3}$ ($s_i = t_{i,1}$)。令 $T_a = [t_{i,1}, t_{i,3}]$, 如 $C_{i,1} = t_{i,3} = t_{i,1} + 1/2$, 令 $b_T = b_2$ 。

C1.2. $t_{i,1} < t_{i,2} = t_{i,3}$, 令 $s_i = t_{i,3}$, 如果 $t_{i,3} - C_{i,1} \geq 1/2$, 令 $T_a = [C_{i,1}, t_{i,3}]$, $b_T = b_1$ 。

C1.3. $t_{i,1} < t_{i,2} < t_{i,3}$, 如果 $C_{i,1} = t_{i,3}$, 则 $t_{i,3} - t_{i,2} = 1/2$ 令 $s_i = t_{i,3}$, $T_a = [t_{i,2}, t_{i,3}]$, $b_T = b_2$ 。如果 $C_{i,3} = t_{i,3}$, 令 $s_i = t_{i,2}$ 。

C2. 当 $r_i = 1$ 时: 如果 $T_a \neq [0, 0]$, 把 r_i 分到空闲泊位 b_T 上的空闲间隔 T_a , 如果 $t_2 - t_1 < 1$ ($t_2 - t_1 \geq 1$), 再令 $T_a = [0, 0]$ ($T_a = [t_1 + 1/2, t_2]$); 如果 $T_a = [0, 0]$, 按以下五种情形去调度。

C2.1. $t_{i,1} = t_{i,3}$ 时, 令 $s_i = t_{i,1}$, 把 r_i 分到左边的泊位 b_1 。

C2.2. $t_{i,1} = t_{i,2} < t_{i,3}$ 时, 令 $s_i = t_{i,1}$, 把 r_i 分到中间的泊位 b_2 。

C2.3. $C_{i,1} = t_{i,1} < t_{i,2} = t_{i,3}$ 时, 令 $s_i = t_{i,1}$, 把 r_i 分到最左边的泊位 b_1 。

C2.4. $C_{i,3} = t_{i,1} < t_{i,2} = t_{i,3}$ 时, 如果 $t_{i,2} - t_{i,1} < 1/2$, 令 $s_i = t_{i,3}$, 把 r_i 分到最左边的泊位 b_1 。如果 $t_{i,2} - t_{i,1} \geq 1/2$, 令 $s_i = t_{i,1}$, 把 r_i 分到最右边的泊位 b_3 。

C2.5. $t_{i,1} < t_{i,2} < t_{i,3}$ 时, 令 $s_i = t_{i,2}$, 把 r_i 分到中间的泊位 b_2 。

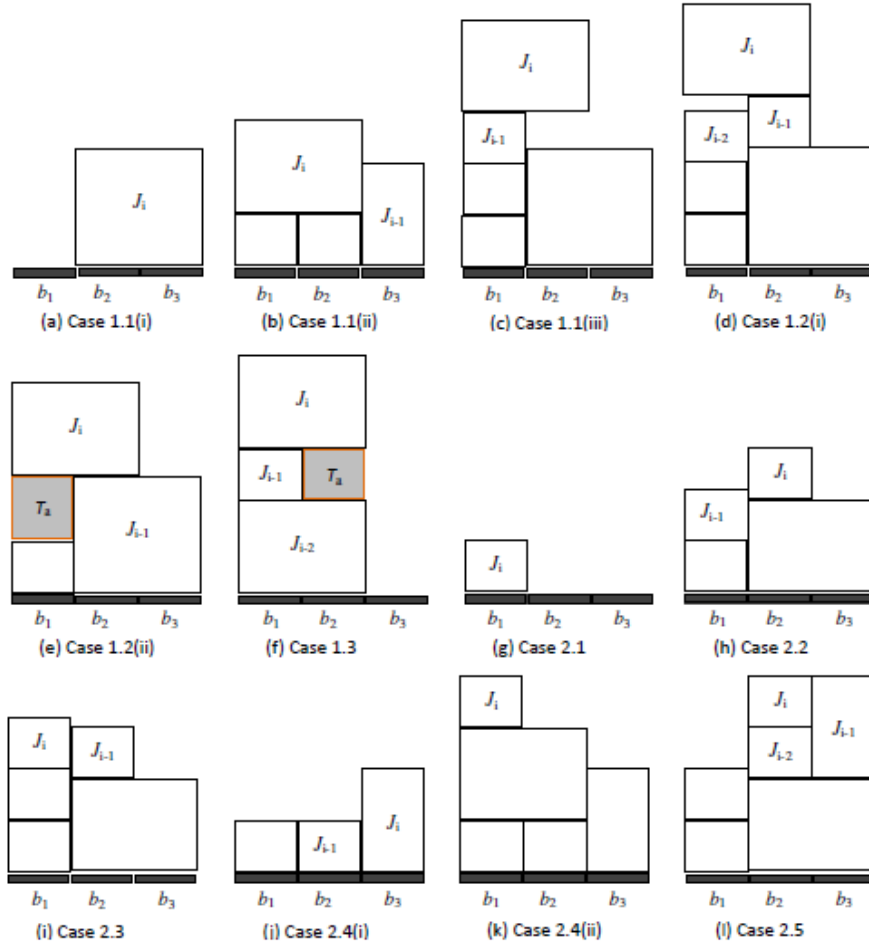


图2-2: 策略设计示意图。

根据上述策略描述可知, r_i 按C1.1或C1.2或C1.3去调度, 大请求 r_i 可能会产生 T_a 或 T_w 。 r_i 如果按C2去调度, 则至少存在一个 T_a 。对 T_w 而言, 在C1.1中, 如果在 $t_{i,1}$ 时刻, 有3个岸桥处于空闲, 那么 $0 < t_{i,3} - t_{i,1} < 1/2$; 在泊位 b_2 会出现 $T_w = [t_{i,1}, t_{i,3})$ 。在C1.2中, 如果 $0 < t_{i,3} - C_{i,1} < 1/2$; 在泊位 b_1 会出现 $T_w = [C_{i,3}, t_{i,3})$ 。在C1.3中, 如果 $t_{i,3} = C_{i,3}$; 在泊位 b_1 会出现 $T_w = [t_{i,1}, t_{i,2})$ 。

推理2-1. 在序列 $\sigma = \{r_1, r_2, \dots, r_n\}$ 中, 至多出现一个 T_w , 且它是首个被分配到左边两个泊位的大请求造成的。

证明. 由策略描述可知, T_w 时间段的存在, 肯定是分配到左边两个泊位的大船造成的, T_w 要么在泊位 b_1 , 要么在泊位 b_2 。假定有两个连续的大请求 r_u 和 r_v , 其中 $e_u \leq s_v$ 和 $i \leq u < v$ 。通过策略的描述, 两个大请求都被分到左边的两个泊位。泊位 b_1 和 b_2 都配置两个岸桥, 在这两个泊位上, 小请求的服务时间为 $1/2$ 。如果在时间段 $[e_u, s_v)$, r_u 和 r_v 之间有偶数个小请求分配到泊位 b_1 和 b_2 上, 那么在此时间段不会出现 T_w 。如果在时间段 $[e_u, s_v)$, r_u 和 r_v 之间有奇数个小请求分配到泊

位 b_1 和 b_2 上,会出现 T_w 。因此, T_w 出现在第一个被分到左边两个泊位的大请求之前。推理得证。证毕

引理 2-1. 如果序列 $\sigma = \{r_1, r_2, \dots, r_n\}$ 中出现 T_w , 那么 $r_1 = \Delta$ 或者 $r_2 = \Delta$ 且 $\Delta \neq 3k/2 (k \geq 2)$ 。

证明. 首先,如果 $r_1 = r_2 = 1$, 根据策略描述,序列中第一个释放的大请求 r_i 会被安排到左边的两个泊位。由 $C_{i,1} = C_{i,2}$ 或 $|C_{i,1} - C_{i,2}| = 1/2$ 可知, r_i 可能会造成 T_a ,但不会产生 T_w 。根据推理1可知,在这种情况下,序列不会出现 T_w 。其次,假设在 $r_1 = \Delta$ 或者 $r_2 = \Delta$ 且 $\Delta = 3k/2 (k \geq 2)$ 情形下,序列 $\sigma = \{r_1, r_2, \dots, r_n\}$ 中会出现 T_w ;那么 r_1 会被分配到右边的两个泊位,在服务 r_1 的时间段 $[0, \Delta/3) = [0, k/2)$ 内,在泊位 b_1 上,刚好可以服务 k 个小请求;序列中的其他大请求仅可能产生 T_a 。与假设矛盾,引理成立。证毕

通过对 T_w 时间段的定义,它的时间跨度严格小于 $1/2$,在 T_w 中,存在两个岸桥资源的浪费;因此,在浪费间隔 T_w 内,浪费的岸桥资源小于 $2 * (1/2) = 1$ 。定义 $T_w (T_a)$ 的不存在为 $T_w = \phi (T_a = [0, 0))$ 和 $T_w (T_a)$ 存在为 $T_w \neq \phi (T_a \neq [0, 0))$ 。

通过对 T_a 和 T_w 的分析和定义,可得出引理2-2的结论。

引理 2-2. 如果在 $\sigma = \{r_1, r_2, \dots, r_n\}$ 中, $r_1 = r_2 = 1$ 且 $n \geq 4$, 那么则有 $C_{\max}(\sigma) / C^*(\sigma) \leq 4/3$ 。

证明. 根据引理的条件 $r_1 = r_2 = 1$,由引理1可知,在序列 $\sigma = \{r_1, r_2, \dots, r_n\}$ 中, T_w 是不存在的。如果 r_n 按C1.1或C1.2或去C1.3调度,则 $r_n = \Delta$ 。如果 r_n 按C2或C2.1或C2.2或C2.3或C2.4或C2.5去调度,则 $r_n = 1$ 。下文对 r_n 的不同调度情形去分析。

Case 1. r_n 按策略设计中情形C1.1或C1.2或C1.3调度去调度。按 T_a 是否存在分如下两种情形。假设最后有 $k (1 \leq k \leq n-2)$ 个大请求到达,则 r_{n-k} 是小请求;如果 $T_a = [0, 0)$,那么左边两个泊位的四个岸桥在时间段 $[0, C_{\max})$ 一直处于忙碌状态,所以有 $C_{\max}(\sigma) / C^*(\sigma) \leq 5/4$ 。如果 $T_a = [t_1, t_2) \neq [0, 0)$,根据引理2-1的证明,必然存在条件 $t_2 - t_1 = 1/2$,根据策略,可推出 $C_{n-k,3} \geq 1$;又 $C_{\max}(\sigma) = t_{n-k,3} + k\Delta/4$,故有 $C^*(\sigma) \geq t_{n-k,3} + k\Delta/5 + (t_{n-k,3} - C_{n-k,3})/5 - 1/5 \geq 4t_{n-k,3}/5 + k\Delta/5 = 4C_{\max}(\sigma)/5$ 。

Case 2. r_n 按C2或C2.3或C2.5去调度,在这种情况下,由条件 $r_1 = r_2 = 1$,可知左边两个泊位的四个岸桥在时间段 $[0, C_{\max})$ 一直处于忙碌状态,所以显然有 $C_{\max}(\sigma) / C^*(\sigma) \leq 5/4$ 。

Case 3. r_n 按策略设计中情形C2.1调度,那么则有 $t_{n,2} = C_{n,3} \geq 1$,而在线策略目标值 $C_{\max}(\sigma) = t_{n,2} + 1/2$,而 $C^*(\sigma) \geq t_{n,2} + 1/5$;故有 $C_{\max}(\sigma) / C^*(\sigma) \leq 5/4$ 。

Case 4. r_n 按策略设计中情形C 2.2去调度, 由条件 $C_{\max}(\sigma) = e_n$ 知 $T_a = [0, 0)$; 如果序列中没有大请求, 则 $C_{\max}(\sigma) = C^*(\sigma)$; 如果序列中至少有一个大请求, 那么 $t_{n,2} \geq 1/2 + \Delta/4 \geq 1$, 剩下的证明同Case 3。

Case 5. r_n 按策略设计中情形C2.4去调度, 由条件 $r_1 = r_2 = 1$ 知, 在这种情形下, $C_{n,1} = C_{n,2} = t_{n,2}$ 。

Case 5.1. $t_{n,2} - t_{n,1} \geq 1/2$, r_n 被分到泊位 b_3 。如果序列中没有大请求, 则 $C_{\max}(\sigma) = C^*(\sigma)$; 如果请求序列中至少有一个大请求, 那么则有 $t_{n,1} = C_{n,3} \geq 1$, 如果 $C_{\max} = t_{n,2}$, 离线最优 $C^*(\sigma) > C_{\max}(\sigma) - (C_{\max}(\sigma) - (1 + t_{n,1}))/5 > 4C_{\max}(\sigma)/5$ 。如果 $C_{\max} = t_{n,1} + 1$, 离线最优 $C^*(\sigma) \geq t_{n,2} + 1/5 - (t_{n,2} - t_{n,1})/5 \geq 4t_{n,2}/5 + 2/5$; 故有 $C_{\max}(\sigma)/C^*(\sigma) \leq 4/3$ 。

Case 5.2. $t_{n,2} - t_{n,1} < 1/2$, r_n 被分到泊位 b_1 。在这种情况下, 序列中至少有一个大请求, 则可推知 $t_{n,1} > 1$, 在线策略的目标值 $C_{\max}(\sigma) = t_{n,1} + 1/2$, 离线最优 $C^*(\sigma) \geq t_{n,2} + 1/5 - (t_{n,2} - t_{n,1})/5 \geq 4t_{n,2}/5 + 2/5 \geq 4C_{\max}(\sigma)/5$ 。引理得证。证毕

引理2-3. 如果在 $\sigma = \{r_1, r_2, \dots, r_n\}$ 中, $r_1 = \Delta$ 或者 $r_2 = \Delta$ 且 $\Delta = 3k/2 (k \geq 2)$ 和 $n \geq 4$, 那么 $C_{\max}(\sigma)/C^*(\sigma) \leq 4/3$ 。

证明. 由条件 $r_1 = \Delta$ 或者 $r_2 = \Delta$, 如果序列中至少存在两个大请求, 那么 $T_w \neq \emptyset$ 或 $T_a \neq [0, 0)$ 。下面我们按 $T_w = \emptyset$ 或 $T_a = [0, 0)$ 是否存在, 分如下四种情形进行讨论。

Case 1. $T_w = \emptyset$ 和 $T_a = [t_1, t_2) \neq [0, 0)$ 。在这种情况下, 可以断定 T_a 是在泊位 b_1 , 它是被安排到右边2个泊位的第一个大请求造成的, 那么则有 $t_2 = \Delta/3$, $t_1 \geq 0$ 。由条件 $T_a \neq [0, 0)$, 那么序列中最后完成的请求为大请求, $t_{n,1} = C_{n,3} = t_2$, 所以有 $C^*(\sigma) \geq C_{\max}(\sigma) + (C_{\max}(\sigma) - t_{n,1})/5 - (t_2 - t_1)/5 \geq 4C_{\max}(\sigma)/5$ 。

Case 2. $T_w = \emptyset$ 和 $T_a = [0, 0)$ 。这种情形下, 说明序列中只有一个大请求, 所以有 $C_{\max}(\sigma) \geq \Delta/3$; 而 $C^*(\sigma) \geq (C_{\max}(\sigma) - \Delta/3) + \Delta/4 = C_{\max}(\sigma) - \Delta/12$; 两者比值至多为4/3。

Case 3. $T_w \neq \emptyset$ 和 $T_a = [t_1, t_2) \neq [0, 0)$ 。在这种情况下, 说明最后一个请求是大请求, 认为 $r_i (i \leq n)$ 是大请求, 并造成了 T_a 的出现; 而令 $r_j (j < i)$ 是另外一个大请求, 并造成 T_w 的出现; 那么可得出 $t_{n,1} = C_{n,3} \geq \Delta/3 + 1 \geq 5/3$, 由于 $T_w \neq \emptyset$ 和 $T_a = [t_1, t_2) \neq [0, 0)$ 同时出现, $C_{\max}(\sigma) \geq \Delta/3 + \Delta/4 + 1/2 + \Delta/4 = 5\Delta/6 + 1/2 \geq 13/6$, 注意到 $t_2 - t_1 = 1/2$, 在 T_a 中, 浪费的资源能力为1。所以离线最优 $C^*(\sigma) \geq C_{\max}(\sigma) - (C_{\max}(\sigma) - t_{n,1})/5 - 1/5 - 1/5 \geq 4C_{\max}(\sigma)/5 - 1/15$, 在线策略的目标值与离线最优的比值不超过13/10。

Case 4. $T_w \neq \emptyset$ 和 $T_a = [0, 0)$ 。由条件 $T_w \neq \emptyset$ 知, 在序列中, 至少存在2个大请求, 我们按序列中 r_n 是大请求还是小请求分两种情形展开讨论。

Case 4.1. $r_n = \Delta$ ，在序列中，一定是另一个大请求被分到右边的两个泊位，且其完工时间为 $\Delta/3$ ，如果 r_n 的开始时间满足 $s_n = \Delta/3$ ，那么 $C_{\max}(\sigma) \geq \Delta/3 + \Delta/4 = 7\Delta/12$ ，而 $C^*(\sigma) \geq \Delta/4 + \Delta/4 = \Delta/2 = 6C_{\max}(\sigma)/7$ ；如果 $s_n > \Delta/3 (\geq 2/3)$ ，它表明要么在 $[0, s_n)$ 之间在泊位 b_1 上至少服务完两个小请求，要么是另一个大请求造成了 T_w 的产生；可推 $s_n \geq 1$ ， $C_{\max}(\sigma) = e_n \geq 1 + \Delta/4 \geq 3/2$ ，而 $C_{n,3} \geq \Delta/3 \geq 2/3$ ， $C^*(\sigma) \geq C_{\max}(\sigma) - (C_{\max}(\sigma) - C_{n,3})/5 - 1/5 \geq 4C_{\max}(\sigma)/5 - 1/15$ ，所以两者的比值不超过 $45/34$ 。

Case 4.2. $r_n = 1$ ，如果 r_n 按 C2 去调度，把 r_n 分到 T_a ，证明过程与 Case 4.1 类似，其中 $C_{\max}(\sigma) \geq \Delta/3 + \Delta/4 + 1/2 + \Delta/4 \geq 13/6$ ，两者比值不超过 $13/10$ 。

Case 4.2.1. 如果 r_n 按 C2.1 去调度，根据条件 $T_w \neq \emptyset$ 和情形 $t_{n,1} = t_{n,3}$ ，可推知 $t_{n,3} \geq 1 + \Delta/3 \geq 5/3$ ， $C_{\max}(\sigma) = t_{n,3} + 1/2$ ，而 $C^*(\sigma) > t_{n,3} + 1/5 - 1/5 = t_{n,3}$ ；故两者比值不超过 $13/10$ 。

Case 4.2.2. 如果 r_n 按 C2.2 去调度，在这种情况下，策略目标值 $C_{\max}(\sigma) = t_{n,2} + 1/2$ 。如果 $t_{n,2} = \Delta/4 + \Delta/3 \geq 7/6$ ，则有 $C^*(\sigma) \geq \Delta/4 + \Delta/4 + 1/2$ ；若 $t_{n,2} = \Delta/4 + \Delta/3 + 1/2 \geq 5/3$ ，那么 $C^*(\sigma) > t_{n,2} + 1/5 - 1/5 = t_{n,2}$ 。在这两种情形下，比值都不超过 $13/10$ 。

Case 4.2.3. 如果 r_n 按 C2.3 去调度，在这种情形下，可知 $t_{n,1} + 1/2 = t_{n,2}$ ，而 $t_{n,2} \geq \Delta/4 + \Delta/3 + 1/2 \geq 5/3$ 。由 C2.3 的条件 $C_{n,1} = t_{n,1} < C_{n,3}$ 知， $t_{n,2} - C_{n,3} < 1/2$ ， $C_{\max}(\sigma) = t_{n,2}$ ，和 $C^*(\sigma) > t_{n,2} - (t_{n,2} - C_{n,3})/5 - 1/5 > t_{n,2} - 3/10$ ；比值不超过 $50/41$ 。

Case 4.2.4. 如果 r_n 按 C2.4 去调度，那么在序列中，至少存在两个小请求和两个大请求。

如果 $t_{n,2} - t_{n,1} \geq 1/2$ ，则把 r_n 分到泊位 b_3 。在这种情况下，可知 $C_{n,1} = C_{n,2} = t_{n,2}$ 。如果策略目标值 $C_{\max}(\sigma) = t_{n,2}$ ，加上条件 $C_{n,3} > 0$ ，可推得离线最优 $C^*(\sigma) \geq C_{\max}(\sigma) - (C_{\max}(\sigma) - (C_{n,3} + 1)/5) - 1/5 > 4C_{\max}(\sigma)/5$ 。另外，如果 $C_{\max}(\sigma) = t_{n,1} + 1 \leq t_{n,2} + 1/2$ ，那么如果 $t_{n,1} = \Delta/3$ ，而 $C^*(\sigma) \geq \min\{2\Delta/3, 2\}$ ；当 $t_{n,1} \geq 1 + \Delta/3 \geq 5/3$ ， $C^*(\sigma) > t_{n,2} - (1 - (t_{n,2} - t_{n,1}))/5 - 1/5 = 4t_{n,2}/5 + t_{n,1}/5 \geq 4t_{n,2}/5 + 1/3$ ，所以比值均有 $C_{\max}(\sigma)/C^*(\sigma) \leq 25/19$ 。

如果 $t_{n,2} - t_{n,1} < 1/2$ ，把 r_n 分到泊位 b_1 。在这种情形下， $C_{n,1} = C_{n,2} = t_{n,2}$ 。由条件 $t_{n,2} - t_{n,1} < 1/2$ 知， $t_{n,1} \geq 1 + \Delta/3 \geq 5/3$ ， $C_{\max}(\sigma) = t_{n,2} + 1/2$ ，而离线最优 $C^*(\sigma) > t_{n,2} - (t_{n,2} - t_{n,1})/5 + 1/5 - 1/5 \geq 4t_{n,3}/5 + 1/3$ 。根据已知条件 $t_{n,2} > t_{n,1} \geq 5/3$ ，故比值不超过 $13/10$ 。

Case 4.2.5. 如果 r_n 按 C2.5 去调度，在这种情形下，有 $C_{\max}(\sigma) = t_{n,2} + 1/2 = t_{n,3}$ ， $t_{n,1} = C_{n,3}$ 。因为 $t_{n,3} - t_{n,1} = 1/2 + t_{n,2} - t_{n,1} < 1$ ，故 $t_{n,3} \geq \Delta/3 + \Delta/4 + 1/2 \geq 5/3$ ，而 $C^*(\sigma) > t_{n,3} - (t_{n,3} - t_{n,1})/5 - 1/5 > t_{n,3} - 2/5$ ；故两者比值 $C_{\max}(\sigma)/C^*(\sigma) < 25/19$ 。

综上, 对以上各种情形, 均有 $C_{\max}(\sigma)/C^*(\sigma) \leq 4/3$ 。引理得证。证毕
通过对引理2-1、2-2、2-3的证明可得出下面的定理2-4。

定理2-4. 当 $n_a = 5$ 时, 针对 $hybr | online - over - list | BAP - QCAP | C_{\max}$, 在线策略 LRSS 的竞争比为 $4/3$ 。

证明. 在序列 $\sigma = \{r_1, r_2, \dots, r_n\}$ ($n \geq 1$) 中, 如果序列请求长度(总个数) $1 \leq n \leq 3$, 对小请求和大请求的任意组合, 很容易去验证 $C_{\max}(\sigma)/C^*(\sigma) \leq 4/3$; 在剩下的证明中, 我们认为 $n \geq 4$, 和 $t_{n,3} \geq t_{n,2} > 0$ 。对于 $\Delta \neq 3k/2$ ($k \geq 2$) 情形, 根据引理 2-2 和引理 2-3, 定理成立。对于 $\Delta = 3k/2$ ($k \geq 2$) 的情形, 引理 2-1 可知, 在序列 $\sigma = \{r_1, r_2, \dots, r_n\}$ 中, 无论前两个请求是否有大请求, 都不会出现 T_w 。剩下证明分析的过程如同之前分析的情形 $\Delta \neq 3k/2$ ($k \geq 2$), 两者比值同样也是 $4/3$ 。定理得证。证毕

根据定理2-4和定理2-3可知, 在线策略 LRSS 是具有最优性能的。

2.4 六个岸桥的调度情形

2.4.1 竞争比下界分析

下面针对6个岸桥的情形, 给出竞争比下界的相关结论。

定理2-5. 当 $n_a = 6$ 时, 对问题 $hybr | online - over - list, LD = 1 | BAP - QCAP | C_{\max}$, 不存在在线策略的竞争比下界小于 $3/2$ 。

证明. 类似于定理2-1的证明, 同样定义 $C_{\max}(\sigma)$ 和 $C^*(\sigma)$, 且只需设计一个请求输入序列 σ 使得对于任一在线策略 A 有 $C_{\max}(\sigma)/C^*(\sigma) \geq 3/2$ 成立。

序列 σ 包含至多 3 个请求且前两个请求均为小请求, 即 $r_1 = r_2 = 1$ 。若策略 A 将两个请求分配至同一个泊位由两台岸桥服务, 则序列 $\sigma = \{r_1, r_2\}$ 。 $C_{\max}(\sigma) \geq 1$, 最优调度方案将是分配两个请求至不同泊位并分别由两台岸桥从 0 时刻开始服务, $C^*(\sigma) = 1/2$ 。因此, $C_{\max}(\sigma)/C^*(\sigma) \geq 2$ 。反之, 若策略 A 将两个请求分配至不同泊位, 则 $r_3 = \Delta$ 且序列 $\sigma = \{r_1, r_2, r_3\}$ 。因为 r_3 需要占用两个相邻的泊位, 无论策略 A 如何分配第 3 个请求, 均有 $C_{\max}(\sigma) \geq 1/2 + \Delta/4$ 成立。最优方案则是分配 r_1, r_2 至最左或最右边的同一泊位而把 r_3 分配至另两个相邻泊位, 并从 0 时刻起不间断地服务这些请求, $C^*(\sigma) = \max\{\Delta/4, 1\}$ 。令 $\Delta = 4$, 则 $C_{\max}(\sigma)/C^*(\sigma) \geq 3/2$ 。定理得证。证毕

根据定理2-5可知, 所有在线策略的竞争比不小于 $3/2$ 。下面根据此结论去设计在线策略 QBT, 并分析其算法性能。

2.4.2 在线策略 QBT 设计与分析

针对无预知的情形,设计在线策略QBT (quay of each berth is two)。该策略的主要思想是:将最左边的两个泊位 b_1, b_2 分别配置岸桥 q_1, q_2 和 q_3, q_4 ,而最右边的泊位 b_3 配置最右边的两个岸桥 q_5, q_6 。下面给出在线策略QBT具体描述。

策略 QBT:

C1 当 $r_i = \Delta$ 。

C1.1. $C_{i,3} = t_{i,2}$ 时, 令 $s_i = t_{i,2}$, 把 r_i 分到右边的两个泊位。 $C_{i,1} = t_{i,2}$, 令 $s_i = t_{i,2}$; 把 r_i 分到左边的两个泊位。

C1.2. 如果 $C_{i,3} = t_{i,1} < C_{i,1} < t_{i,2}$, 令 $s_i = t_{i,2}$, 把 r_i 分到右边的两个泊位; 令 $T_a = [t_{i,1}, t_{i,2})$, $b_T = b_3$ 。如果 $C_{i,1} = t_{i,1} < C_{i,3} < t_{i,2}$, 令 $T_a = [t_{i,1}, t_{i,2})$, 令 $b_T = b_1$ 。

C2 当 $r_i = 1$ 时: 如果 $T_a \neq [0, 0)$, 把 r_i 分到空闲泊位 b_T 上的空闲间隔 T_a , 再令 $T_a = [0, 0)$; 当 $T_a = [0, 0)$, 按以下三种情形去调度。

C2.1. $t_{i,1} = t_{i,3}$ 时, 令 $s_i = t_{i,1}$, 把 r_i 分到左边的泊位 b_1 。

C2.2. $t_{i,1} = t_{i,2} < t_{i,3}$ 时, 令 $s_i = t_{i,1}$, 把 r_i 分到中间的泊位 b_2 。

C2.3. $t_{i,1} < t_{i,2} = t_{i,3}$ 时, 令 $s_i = t_{i,1}$, 如果 $t_{i,1} = C_{i,1}$, 把 r_i 分到最左边的泊位 b_1 ; 如果 $t_{i,1} = C_{i,3}$, 把 r_i 分到最右边的泊位 b_3 。

根据上述策略描述可知,每个泊位都是配置两个岸桥,因而是同质的;每个大请求需占两个相邻的泊位,占用 $\Delta/4$ 单位时间,而小请求可以停靠于任何一个泊位上,需要 $1/2$ 的单位时间。通过策略C1.2的描述,在泊位 b_1 或 b_3 上,大请求 r_i 导致的某一泊位服务该大请求前出现可利用空闲时间段 $T_a = [t_1, t_2)$;我们可推知 $t_2 = 1/2 + t_1$; 不然如果当 $t_2 < 1/2 + t_1$ 时,随着 r_{i-1} 的释放,其完工时间 $t_{i,2} = t_2$,而 $C_{i-1,2} < \min(C_{i-1,1}, C_{i-1,3})$,按C2.1或C2.2或C2.3去调度是不可能出现这种情形的。具体的推理过程见下文引理2-4。

引理2-4. 如果大请求 r_i 造成 T_a 时间段的出现,那么至少在 r_i 之前存在一个大请求。

证明. (反证法) 假设如果前 $i-1$ 个请求都是小请求, r_i 产生了 T_a 时间段。随着 r_i 的释放, r_{i-1} 通过C2.1或C2.2或C2.3去调度,我们有 $C_{i,1} = C_{i,2}$ 或 $C_{i,2} = C_{i,3}$; 因此 r_i 会按C1.1去调度,不会产生 T_a 时间段。因此引理成立。 引理得证, 证毕

定理2-6. 当 $n_q = 6$ 时, 对于问 $hybr | online - over - list, LD = 1 | BAP - QCAP | C_{\max}$, 策略QBT具有竞争比 $3/2$ 。

证明. 给定任何输入序列 $\sigma = \{r_1, r_2, \dots, r_n\}$, 令 $C_{\max}(\sigma)$ 为在线策略A的最大完工时间, $C^*(\sigma)$ 为离线最优策略OPT对应的目标函数值。如果 r_n 按C1.1或C1.2去调

度, 则 $r_n = \Delta$ 。如果 r_n 按C2或C2.1或C2.2或C2.3去调度, 则 $r_n = 1$ 。由引理2-4可知, 在序列 $\sigma = \{r_1, r_2, \dots, r_n\}$ 中, T_w 是不存在的。 r_n 可按以下5种情形去调度。

Case 1. r_n 按策略设计中情形C1.1去调度。按 T_a 是否存在分如下两种情形。

Case 1.1. 当 $T_a = [t_1, t_2) \neq [0, 0)$, 假设最后有 $k (\geq 1)$ 个大请求到达, 则 r_{n-k} 是小请求; 那么 $t_j = t_{n-k, j} (j=1, 2)$ 。由引理2可知, $\Delta \geq 2$, $t_1 \geq \Delta/4 \geq 1/2$ 。我们通过先前的分析可知, $t_2 - t_1 = 1/2$, 因此 $C_{\max} = t_1 + k\Delta/4 + 1/2$; 而离线最优 $C^*(\sigma) = t_1 + k\Delta/6 + 1/6$ 。又因为条件 $t_1 \geq 1/2$, 所以得出 $C_{\max}(\sigma)/C^*(\sigma) \leq 3/2$ 。

Case 1.2. $T_a = [0, 0)$; 在这种情况下, 策略目标值 $C_{\max}(\sigma) = t_{n,2} + \Delta/4$, 而 $C^*(\sigma) \geq t_{n,2} + \Delta/6 - 2(t_{n,2} - t_{n,1})/6 \geq 2t_{n,3} + \Delta/6 \geq 2C_{\max}(\sigma)/3$ 。

Case 2. r_n 按C1.2去调度, 在这种情形下, 会产生一个 T_a 。剩下的证明过程类似Case 1.1; 当 $k=1$ 时, $C_{\max}(\sigma)/C^*(\sigma) < 3/2$ 。

Case 3. r_n 按策略设计中情形C2去调度, 在这种情形下, $T_a = [0, 0)$; 我们可推知 r_{n-1} 是第一个大请求, 不然 r_{n-1} 会按策略设计中情形C2去调度。因为 T_a 的存在, 所以有 $t_{n,1} > 0$ 。在线策略目标值 $C_{\max}(\sigma) = t_{n-1,2} + \Delta/4 = t_{n,2}$, 而离线最优 $C^*(\sigma) \geq t_{n,2} - 2(t_{n,2} - t_{n,1})/6 > 2t_{n,2}/3 = 2C_{\max}(\sigma)/3$ 。

Case 4. r_n 按策略设计中情形C2.1或C2.2去调度, 则有 $C_{\max}(\sigma) = t_{n,2} + 1/2$ 。如果 $t_{n,2} = 0$, 那么则 $C_{\max}(\sigma) = C^*(\sigma)$ 。如果 $t_{n,2} \geq 1/2$, $C^*(\sigma) \geq t_{n,2} + 1/6 \geq 2C_{\max}(\sigma)/3$ 。

Case 5. r_n 按策略设计中情形C2.3去调度, 不失一般性假定 $t_{n,1} = C_{n,1}$; 对于 $t_{n,1} = C_{n,3}$ 的情形, 分析过程相同。

Case 5.1. $t_{n,2} = C_{n,3}$, 在这种情况下, 如果 $t_{n,2} - t_{n,1} < 1/2$, 因为在 r_n 之前至少有一个大请求到达, 故有 $C_{\max}(\sigma) = t_{n,1} + 1/2 < t_{n,2} + 1/2$; 如果 $t_{n,2} = \Delta/4$, 则可推出 $C_{\max}(\sigma) = C^*(\sigma)$; 如果 $t_{n,2} \geq \Delta/4 + 1/2 \geq 1$, 而离线最优 $C^*(\sigma) > t_{n,2}$, 因此 $C_{\max}(\sigma)/C^*(\sigma) < 3/2$; 此外, 如果 $t_{n,2} - t_{n,1} \geq 1/2$, 则有 $C_{\max}(\sigma) = t_{n,2}$, 而离线最优 $C^*(\sigma) \geq t_{n,2} - 2(t_{n,2} - (t_{n,1} + 1))/6 > 2t_{n,2}/3 = 2C_{\max}(\sigma)/3$ 。

Case 5.2. $C_{n,3} < t_{n,2}$ 。在这种情况下, $t_{n,2} - t_{n,1} = 1/2$, $C_{\max}(\sigma) = t_{n,1} + 1/2 = t_{n,2}$, 因为 $t_{n,1} > 0$, 离线最优 $C^*(\sigma) \geq t_{n,1} + 2/6 \geq 2C_{\max}(\sigma)/3$ 。定理得证, 证毕

根据定理2-6可知, 在线策略QBT满足最坏情形下任意输入序列目标值与最离线最优的比值。

2.5 本章小结

本章提出了无预知信息下集装箱码头泊位与岸桥联合在线调度问题。考虑了1个混合型泊位布局且存在大小两种服务请求的情形; 着重分析了4、5、6个岸桥的情形, 分别设计了相应的在线调度策略, 并证明了所有在线策略的竞争性能都是最优的。

第3章 预知 $k=1$ 个请求下泊位与岸桥联合在线调度

第二章已分析了无预知信息下的泊位与岸桥联合调度模型,并证明所设计在线策略的良好性能;在码头调度实际中,通常每天制定次日的调度计划,制定计划时基于当天及次日抵港船舶的信息,而非仅仅考虑所安排船舶本身的信息,因而本章构建有限预知信息下的泊位与岸桥联合调度在线模型,并探讨预知后续一艘船舶需求信息的特殊情形,该研究将有助于设计一般情形下即预知多艘船舶时的调度计划。

因此,本章主要研究了预知 $k=1$ 个请求下的泊位与岸桥联合调度在线模型,同样讨论了3个相邻泊位,4、5、6个岸桥的情形。首先,讨论了各自情形的竞争比下界,然后对于每个情形,分别设计出预知 $k=1$ 个请求下的联合调度的在线策略,接着分析其竞争性能;并对其中2个在线策略进行数值仿真来验证所设计算法的有效性。通过与第二章相关结论相比,本章结论表明即使十分有限的预知能力($k=1$)也可以有效改进联合调度策略的竞争性能。

3.1 问题描述与模型刻画

本章采用Zhang等^[54]的建模方法,并运用在线理论来刻画码头泊位与岸桥的联合调度,根据船舶动态到达特征构建相应的over-list在线模型。在现实当中,各艘船舶逐天动态抵港,码头往往是提前一两天才能准确获知即将抵港的船舶信息;因而所设计调度方案要求能够应对这种动态特征,调度计划往往呈现滚动操作特征。具有有限预知信息的调度模型在生产、管理调度等领域已有相关研究成果。苏兵,兰小毅^[55]从在线与竞争策略的角度讨论了决策者在车辆到达一交叉口时可以提前获取后一交叉口的关联路段是否堵塞及堵塞恢复时间情形下的路径选择问题。郑斐峰和徐寅峰^[56]探讨了有限预知信息的集装箱搬卸占线问题,在服务当前请求时预先知道后续一部分请求的信息,并运用最坏情形分析方法研究了所提出的贪婪移位策略的竞争性能。Pan等^[57]在Zheng等^[58]的基础上讨论了预知后续1个请求信息下的集装箱码头泊位与岸桥联合在线调度问题,针对岸桥总数为4或5个的两种情形,分别设计了具有最优竞争比 $(1+\sqrt{2})/2$ 、 $4/3$ 的联合调度在线策略。

本章后续对4、5个岸桥的讨论分析和文献Pan等^[57]的不同之处在于在线策略(调度方案)的设计不同,这表明在线问题的竞争比一般是最坏情形下的比值,而对于某个在线问题,可能有多种在线策略。

本章预知 $k=1$ 个请求下的模型假设基本相同于第二章无预知情形下的模型假设,所不同的一条假设在于:

- (1) 当请求 $r_i (1 \leq i < n)$ 释放时可以预知下一请求 r_{i+1} 的信息，而 r_n 释放时得知序列长度即 n 值。即在预知 $k=1$ 个请求的情境下，在线策略在 r_n 释放时，即可获知请求序列长度即 n 的取值。

运用 Bierwirth 和 Meisel^[4] 的四参数描述法，将所本章中所构建模型标记为 $hybr|online-over-list, LD=1|BAP-QCAP|C_{\max}$ ，其中， $hybr$ 代表混合型泊位， $online-over-list$ 表示请求逐个释放的在线模型， $LD=1$ 表示在线策略可以预知后续1个请求，而 C_{\max} 表示最大完工时间的目标函数。

下面将对4、5、6个岸桥的情形分别展开讨论。

3.2 四个岸桥的调度情形

对于4个岸桥的情形，下面先证明了当预知能力 $k=1$ 情形下不存在确定性在线策略的竞争比小于 $(1+\sqrt{2})/2$ ；接着针对该下界值设计出具有最优性能的在线策略。

3.2.1 竞争比下界分析

下面对于预知 $k=1$ 个请求的一般情形，证明任意确定性在线策略不可能具有竞争比小于 $(1+\sqrt{2})/2$ 。

定理3-1. 当 $n_q = 4$ ，针对 $hybr|online-over-list, LD=1|BAP-QCAP|C_{\max}$ ，问题的竞争比下界为 $(1+\sqrt{2})/2$ 。

证明. 为了证明其成立，由敌手设计一个输入序列，使其情形最坏。输入序列为 $\sigma = \{r_1, r_2, r_3\} = \{1, \Delta, 1\}$ ，令 $C_{\max}(\sigma)$ 为在线策略 A 的完成时间； $C^*(\sigma)$ 为离线OPT完成时间。如果 r_1 和 r_2 后无其他请求到达，分给 r_1 两个岸桥，则 $C_{\max}(\sigma) \geq 1/2 + \Delta/4$ ，而最优OPT策略 r_1 可能分给 r_1 一个岸桥，分给 r_2 三个岸桥，则 $C^*(\sigma) = \max(\Delta/3, 1)$ ，则竞争比 $C_{\max}(\sigma)/C^*(\sigma) \geq (1/2 + \Delta/4)/[\max(1, \Delta/3)]$ 。如果 A 策略分给 r_1 一个岸桥和 r_2 三个岸桥，那么第三个请求是小请求且已知后续无请求到达， $C_{\max}(\sigma) \geq 1/2 + \Delta/3$ 。OPT策略分给 r_2 四个岸桥，分给 r_1 和 r_3 两个岸桥，那么 $C^*(\sigma) = 1/2 + \Delta/4$ ，则竞争比 $C_{\max}(\sigma)/C^*(\sigma) \geq (1/2 + \Delta/3)/(1/2 + \Delta/4)$ 。如果 Δ 取 $\Delta = (6 + 12\sqrt{2})/7$ ，那么 $C_{\max}(\sigma)/C^*(\sigma) \geq (1+\sqrt{2})/2$ 。定理得证。证毕

由定理3-1知，该问题的所有在线策略竞争比不小于 $(1+\sqrt{2})/2$ ；由于大请求任务量取值的不同对联合调度的影响较大，接下来按大请求任务量 Δ 取值范围的不同分为两个小节去讨论。

3.2.2 策略 LLWL 的设计及竞争比分析

接下来, 针对预知一个 ($k=1$) 请求的特定情形, 当大请求任务量取值 $\Delta > (6+12\sqrt{2})/7$ 时, 设计在线策略 LLWL (Load is Large With Lookahead)。该策略的主要思想是: 在分配服务序列的最后1个请求之前, 泊位 b_1, b_2 分别配置岸桥 q_1, q_2 和 q_3, q_4 , 而将 b_3 泊位置于空闲状态; 对于最后1个请求则根据当前状态下的局部最优目标值进行泊位与岸桥的分配。

下面给出策略 LLWL 的具体描述。

策略 LLWL:

若 $r_i = r_n$, 由于预知没有后续请求, 因而对最后1个请求分配适当的泊位以及服务岸桥数量以使 C_{\max} 最小。否则, 若 $1 \leq i < n$, 由于预知后续请求 r_{i+1} , 分如下两种情形进行讨论:

Case1. 当 $r_i = \Delta$ 时: 令 $s_i = \max\{C_{i,1}, C_{i,2}\}$, 把 r_i 分到左边的两个泊位。如果 $C_{i,1} \geq C_{i,2}$, 令 $T_a = [t_1, t_2) = [C_{i,2}, C_{i,1})$, $b_T = b_2$; 如果 $C_{i,1} < C_{i,2}$, 令 $T_a = [t_1, t_2) = [C_{i,1}, C_{i,2})$, 令 $b_T = b_1$ 。

Case 2. 当 $r_i = 1$ 时: 如果 $T_a \neq [0, 0)$, 把 r_i 分到空闲泊位 b_T 上的空闲间隔 T_a , 再令 $T_a = [0, 0)$; 当 $T_a = [0, 0)$, 如果 $C_{i,1} \geq C_{i,2}$, 令 $s_i = C_{i,2}$, 把 r_i 分到泊位 b_2 ; 如果 $C_{i,1} < C_{i,2}$, 令 $s_i = C_{i,1}$, 把 r_i 分到泊位 b_1 。

由策略的描述可知, 泊位 b_3 一直处于空闲。在空闲间隔 T_a , 可用的岸桥能力为1。

定理3-2. 当 $n_q = 4$, 对于 $hybr | online - over - list, LD = 1 | BAP - QCAP | C_{\max}$ 问题, 在线策略 LLWL 的竞争比为 $(1 + \sqrt{2})/2$ 。

证明. 对于任何输入序列为 $\sigma = \{r_1, r_2, \dots, r_n\}$, 令 $C_{\max}(\sigma)$ 为在线策略 LLWL 的完成时间; $C^*(\sigma)$ 为离线 OPT 完成时间。

Case1. 最后一个请求是大请求。

Case1.1. 当 $T_a = [0, 0)$, 因为在 $[0, C_{\max}(\sigma))$ 没有岸桥资源的浪费, 则有 $C_{\max}(\sigma) = C^*(\sigma) = C_{n,1} + \Delta/4$; 竞争比显然为1。

Case1.2. 当 $T_a = [t_1, t_2) \neq [0, 0)$, 假设在 t_2 时刻后有 k 个大请求到达, 则 $C_{\max}(\sigma) = t_2 + \Delta/4$; 如果 $t_2 = 1/2$ 和 $k = 1$, 说明总共有两个请求, $r_1 = 1$ 和 $r_2 = \Delta$, 此时 $C^*(\sigma) \geq \min\{1/2 + \Delta/4, \max(1, \Delta/3)\}$, 则有 $C_{\max}(\sigma)/C^*(\sigma) \leq 1.2072$ 。如果 $t_2 \geq 1$ 或 $k \geq 2$, $C_{\max}(\sigma) = t_2 + k\Delta/4 - 1/4$, 那么 $C_{\max}(\sigma)/C^*(\sigma) \leq 6/5$ 。

Case 2. 最后一个请求是小请求。显然 $T_a = [0, 0)$; 不然, 当 $r_i = 1$ 时: 如果 $T_a \neq [0, 0)$, 把 r_i 分到空闲泊位 b_T 上的空闲间隔 T_a 。

Case2.1. 如果 $C_{n,1} = C_{n,2}$ ，在这种情形下， $C_{\max}(\sigma) = C_{n,1} + 1/2$ 。如果序列只有1个或3个连续小请求，则有 $C_{n,1} \leq 1/2$ ， $C_{\max}(\sigma) = C^*(\sigma)$ ；如果该序列中请求任务只有2个，那么 $r_1 = \Delta$ ， $C_{n,1} = \Delta/4$ ，此时 $C^*(\sigma) \geq \min\{1/2 + \Delta/4, \max(1, \Delta/3)\}$ ， $C_{\max}(\sigma)/C^*(\sigma) \leq (1 + \sqrt{2})/2$ ；如果 $C_{n,1} \geq 1$ ，而离线最优 $C^*(\sigma) \geq C_{n,1} + 1/4$ ，则有 $C_{\max}(\sigma)/C^*(\sigma) \leq 6/5$ 。

Case2.2. 如果 $C_{n,1} < C_{n,2}$ (或 $C_{n,2} < C_{n,1}$)，根据策略设计则有 $C_{n,2} = C_{n,1} + 1/2$ (或 $C_{n,1} = C_{n,2} + 1/2$)，则有 $C_{\max}(\sigma) = C^*(\sigma)$ 。定理得证。证毕

通过定理3-2的证明，我们可知当大请求任务量 $\Delta > (6 + 12\sqrt{2})/7$ 时，在线策略LLWL的性能是最优的。

3.2.3 策略 LSWL 的设计及竞争比分析

针对大请求任务量取值在 $2 \leq \Delta \leq (6 + 12\sqrt{2})/7$ 时，本章设计了在线策略 LSWL (Load is Small With Lookahead)。该策略的主要思想是：

若序列中除非存在两种特殊情形： $r_1 = 1, r_2 = \Delta$ 或 $r_1 = \Delta, r_2 = 1$ 时， b_3 泊位配置1个岸桥。否则，在分配服务序列的最后1个请求之前，泊位 b_1, b_2 分别配置岸桥 q_1, q_2 和 q_3, q_4 ，而将 b_3 泊位置于空闲状态；对于最后1个请求则根据当前状态下的局部最优目标值进行泊位与岸桥的分配。

下面给出策略LSWL的具体描述。

策略LSWL:

若 $r_i = r_n$ ，由于预知没有后续请求，因而对最后1个请求分配适当的泊位以及服务岸桥数量以使 C_{\max} 最小。否则，若 $1 \leq i < n$ ，由于预知后续请求 r_{i+1} ，分如下四种情形进行讨论：

Case1. 当 $r_i = \Delta$ ， $r_{i+1} = \Delta$ 时：令 $s_i = t_{i,3}$ ，把 r_i 分到左边的两个泊位。如果 $C_{i,1} \geq C_{i,2}$ 且 $C_{i,1} - C_{i,2} = 1/2$ ，令 $T_a = [t_1, t_2) = [C_{i,2}, C_{i,1})$ ， $b_T = b_2$ ；如果 $C_{i,1} < C_{i,2}$ 且 $C_{i,2} - C_{i,1} = 1/2$ ，令 $T_a = [t_1, t_2) = [C_{i,1}, C_{i,2})$ ，令 $b_T = b_1$ 。

Case 2. 当 $r_i = \Delta$ ， $r_{i+1} = 1$ 时：如果 $r_1 = \Delta, r_2 = 1$ ，把 r_i 分到左边的两个泊位，只使用三个岸桥。其他情形的调度和情形Case1相同。

Case 3. 当 $r_i = 1$ ， $r_{i+1} = 1$ 时：如果 $T_a \neq [0, 0)$ ，把 r_i 分到空闲泊位 b_T 上的空闲间隔 T_a ，再令 $T_a = [0, 0)$ ；当 $T_a = [0, 0)$ ，如果 $C_{i,1} > C_{i,2}$ ，令 $s_i = C_{i,2}$ ，把 r_i 分到泊位 b_2 ；如果 $C_{i,1} = C_{i,2}$ ，令 $s_i = t_{i,3}$ ，把 r_i 分到泊位 b_1 。

Case 4. 当 $r_i = 1$ ， $r_{i+1} = \Delta$ 时：如果 $T_a \neq [0, 0)$ ，把 r_i 分到空闲泊位 b_T 上的空闲间隔 T_a ，再令 $T_a = [0, 0)$ ；当 $T_a = [0, 0)$ ，如果特例 $r_1 = 1, r_2 = \Delta$ 出现，把 r_i 分到最右边的泊位 b_3 ，分配其一个岸桥；其他情形中如果 $C_{i,1} > C_{i,2}$ ，令 $s_i = C_{i,2}$ ，把 r_i 分到泊位 b_2 ；如果 $C_{i,1} = C_{i,2}$ ，令 $s_i = t_{i,3}$ ，把 r_i 分到泊位 b_1 。

定理3-3. 当 $n_q = 4$ ，针对 $hybr|online-over-list, LD=1|BAP-QCAP|C_{\max}$ 问题，在线策略LSWL的竞争比为 $(1+\sqrt{2})/2$ 。

证明. 对于任何输入序列为 $\sigma = \{r_1, r_2, \dots, r_n\}$ ，令 $C_{\max}(\sigma)$ 为在线策略LSWL的完成时间； $C^*(\sigma)$ 为离线OPT完成时间。在空闲间隔 T_a ，可用的岸桥能力为1。

Case1. 最后一个请求是大请求。

Case1.1. 当 $T_a = [0, 0)$ ，如果 $n = 1$ 或 $n = 2$ ，则有 $C_{\max}(\sigma) = C^*(\sigma)$ ；如果 $n \geq 3$ ， $C_{\max}(\sigma) = t_{n,3} + \Delta/4$ ，而 $C^*(\sigma) \geq t_{n,3} + \Delta/4 - \Delta/12$ ，所以有 $C_{\max}(\sigma)/C^*(\sigma) \leq 7/6$ 。

Case1.2. 当 $T_a = [t_1, t_2) \neq [0, 0)$ ，假设在 t_2 时刻后有 k 个大请求到达，则在线完工时间 $C_{\max}(\sigma) = t_2 + k\Delta/4$ ；因为 $t_2 \geq 3/2$ ，而离线最优 $C^*(\sigma) \geq t_2 + k\Delta/4 - \Delta/12$ ，那么 $C_{\max}(\sigma)/C^*(\sigma) \leq 1.16$ 。

Case 2. 最后一个请求是小请求。显然 $T_a = [0, 0)$ ；不然，当 $r_i = 1$ 时：如果 $T_a \neq [0, 0)$ ，把 r_i 分到空闲泊位 b_r 上的空闲间隔 T_a 。

Case 2.1. 如果 $C_{n,1} = C_{n,2}$ ，在这种情形下， $C_{\max}(\sigma) = C_{n,1} + 1/2$ 。如果请求只有1个或3个连续小请求，则有 $C_{n,1} \leq 1/2$ ， $C_{\max}(\sigma) = C^*(\sigma)$ ；如果请求只有两个，那么 $r_1 = \Delta$ ，此时 $C^*(\sigma) = C_{\max}(\sigma) = \{\max(1, \Delta/3)\}$ 。如果 $n = 3$ 且 $\{r_1, r_2\} = \{1, \Delta\}$ 或 $\{\Delta, 1\}$ ，则 $C_{\max}(\sigma) = 1/2 + \Delta/3$ ， $C^*(\sigma) = 1/2 + \Delta/4$ ，因此 $C_{\max}(\sigma)/C^*(\sigma) \leq (1+\sqrt{2})/2$ ；如果 $\{r_1, r_2\} \neq \{1, \Delta\}$ 或 $\{\Delta, 1\}$ 且 $C_{n,1} \geq 1$ ，则有 $C^*(\sigma) \geq C_{n,1} + 1/4$ ， $C_{\max}(\sigma)/C^*(\sigma) \leq 6/5$ 。

Case 2.2. 如果 $C_{n,2} < C_{n,1}$ ，根据策略分析则有 $C_{n,1} = t_{n,2} + 1/2$ ，如果 $n=2$ 或 $n=4$ 的四个小请求，则有 $C_{\max}(\sigma) = C^*(\sigma)$ 。如果 $n \geq 4$ 且 $\{r_1, r_2\} = \{1, \Delta\}$ 或 $\{\Delta, 1\}$ 或者 $n \geq 6$ ，则有 $C^*(\sigma) \geq C_{\max}(\sigma) - 1/4$ ，又因为 $C_{\max}(\sigma) \geq 3/2$ ，所 $C_{\max}(\sigma)/C^*(\sigma) \leq 6/5$ 。定理得证。 证毕

通过对定理3-3的证明可知，我们可知当大请求任务量 $2 \leq \Delta \leq (6+12\sqrt{2})/7$ 时，在线策略LSWL的竞争比为 $(1+\sqrt{2})/2$ 。结合定理3-1和定理3-2可知，该问题不存在在线策略使得竞争比小于 $(1+\sqrt{2})/2$ ，对于大请求任务量的不同取值，在线策略LSWL和策略LLWL能够有效应对任意的序列组合，使得其目标函数值与离线最优值的比值不超过 $(1+\sqrt{2})/2$ 。

3.3 五个岸桥的调度情形

对于5个岸桥的情形，下文先分析该问题竞争比的下界；接下来，结合此下界值设计相应的在线策略并通过分析证明指出其具有最优的竞争性能，即竞争比达到了下界值。

3.3.1 竞争比下界分析

对于预知后续一个请求的一般情形，下面通过构造特殊实例来证明任意在线策略不可能具有竞争比小于 $5/4$ 。

定理3-4. 当 $n_q = 5$ 时, 问题 $hybr | online - over - list, LD = 1 | BAP - QCAP | C_{\max}$ 的竞争比下界为 $5/4$ 。

证明. 要证明该定理, 只需设计一个请求输入序列 σ , 使得对于任一在线策略 A 满足不等式 $C_{\max}(\sigma)/C^*(\sigma) \geq 5/4$, 其中 $C_{\max}(\sigma)$ 和 $C^*(\sigma)$ 分别为策略 A 所产生服务序列的最大完工时间、离线最优方案下的目标函数值。序列 σ 包含至多3个请求且 $r_1 = r_2 = 1$ 。策略 A 在分配 r_1 时预知 $r_2 = 1$ 。若 A 分配一个岸桥服务 r_1 , 则 $\sigma = \{r_1, r_2\}$, 从而 $C_{\max}(\sigma) \geq 1$ 。最优方案将各分配2个岸桥分别服务两个请求, $C^*(\sigma) = 1/2$ 且 $C_{\max}(\sigma)/C^*(\sigma) \geq 2$ 。反之, 若策略 A 分配2个岸桥服务 r_1 , 则 $r_3 = \Delta$ 且 $\sigma = \{r_1, r_2, r_3\}$ 。因为 r_3 需占用两个相邻泊位, $C_{\max}(\sigma) \geq \min\{\Delta/3, 1/2 + \Delta/4\}$ 。最优方案将分配1个岸桥服务前2个请求, 分配4个岸桥服务 r_3 , $C^*(\sigma) = \max\{\Delta/4, 2\}$ 。令 $\Delta = 8$, 从而 $C_{\max}(\sigma)/C^*(\sigma) \geq 5/4$ 。定理得证。 证毕

根据定理 3-4, 任何在线策略在预知 1 个船舶请求下的竞争比不可能小于 $4/3$ 。

接下来, 针对预知后续 1 个请求特殊情形的竞争比下界值设计相应的在线策略。对于 5 个岸桥的联合调度模型, 在线策略将最左边的两个泊位 b_1, b_2 分别配置岸桥 q_1, q_2 和 q_3, q_4 , 而最右边的泊位 b_3 配置最右边的 1 个岸桥 q_5 。因此, 若某个大请求被分配至泊位 b_2, b_3 , 则将由右侧 3 个岸桥 q_3, q_4, q_5 进行服务。下面对大请求任务量 $\Delta \geq 3$ 和 $2 \leq \Delta < 3$ 的两种情形分别设计不同的调度策略。

3.3.2 $\Delta \geq 3$ 的情形

基于上述对岸桥与泊位两种资源的固定配置方案, 在线策略只需刻画分配每个请求至哪一个或哪两个泊位以及服务的启动时间即可。下面给出在线策略 GR_1SI (Greedy 1 With Simple description) 的具体描述:

策略 GR_1SI :

若 $r_i = r_n$, 由于预知没有后续请求, 因而分配该请求至某一泊位以最小化 C_{\max} 。若有多个泊位均可导致相同目标函数值, 则分配给下标最小的泊位; 否则, 若 $1 \leq i < n$, 由于预知后续请求 r_{i+1} , 分如下两种情形进行调度。

C1. $r_i = \Delta$ 。对于 $r_{i+1} = 1$ 或 Δ , 令启动时间 $s_i = t_{i,2}$, 把 r_i 分配至左边两个泊位 b_1, b_2 。

C2. $r_i = 1$ 。

C2.1. $C_{i,1} - C_{i,2} = 1/2$ 。令 $s_i = C_{i,2}$, 把 r_i 分配至 b_2 泊位。

根据上述策略中的C2.1、C2.2两种情形, 若 r_i ($1 \leq i < n$) 是小请求且被分配至 b_1 泊位, 则根据策略C2.2的条件, 下一请求 $r_{i+1} = 1$, 从而 r_{i+1} 满足C2.1条件并被分

配至 b_2 泊位。 r_{i+1} 分配之后 b_1, b_2 泊位同时从 e_{i+1} 时刻进入空闲状态, 因此, 任意一个大请求 $r_i = \Delta$ 释放时均有 $C_{i,1} = C_{i,2}$ 。上述分析表明, b_1, b_2 泊位及相应的岸桥资源在完成其所分配的所有任务之前不会出现空闲状态。

定理3-5. 当 $n_q = 5$, 对于问题 $hybr|online-over-list, LD=1|BAP-QCAP|C_{\max}$, 且当大请求任务量 $\Delta \geq 3$ 时, 在线策略 GR_1SI 具有竞争比为 $5/4$ 。

证明. 给定任一请求输入序列 $\sigma = \{r_1, r_2, \dots, r_n\}$, 令 $C_{\max}(\sigma)$ 为策略 GR_1SI 的最大完工时间, $C^*(\sigma)$ 为离线最优策略 OPT 对应的目标函数值。根据 GR_1SI 策略描述, 所有大请求均被分配至左边两个泊位 b_1, b_2 , 而且在完成最后一个大请求之前, 左边2个泊位与4个岸桥一直处于服务状态。下面根据最后一个请求 r_n 的分配结果分为两种情形。

Case 1. r_n 分配至 b_1 或 b_2 泊位。

Case 1.1. $r_n = \Delta$ 。如果 $C_{\max}(\sigma) = e_n$, 由于左边4个岸桥在 e_n 之前一直处于加工状态, 因此, $C^*(\sigma) \geq 4e_n/5 = 4C_{\max}(\sigma)/5$; 反之, 如果 $C_{\max}(\sigma) = C_{n,3} (> e_n)$, 根据策略C2.2情形, 每个小请求分配至 b_3 泊位的条件是其下一请求是大请求。因此, b_1, b_2 泊位至少分配了 $C_{n,3}$ 个大请求, 结合 $\Delta \geq 3$ 可得, 离线最优完工时间 $C^*(\sigma) \geq (C_{n,3}\Delta + C_{n,3})/5 \geq 4C_{\max}(\sigma)/5$ 。

Case 1.2. $r_n = 1$ 。由于 r_n 分配至 b_1 或 b_2 泊位, 其加工时间长度为 $1/2$ 。如果 $C_{\max}(\sigma) = C_{n,3} (> e_n)$, 则分析同情形Case1.1; 否则, 如果在线策略的完工时间 $C_{\max}(\sigma) = e_n = \min\{C_{n,1}, C_{n,2}\} + 1/2$, 当序列 σ 包含至多1个大请求时, 容易验证 $C_{\max}(\sigma)/C^*(\sigma) \leq 5/4$ 。当序列 σ 包含至少2个大请求时, 由于大请求均分配至泊位 b_1, b_2 , $\min\{C_{n,1}, C_{n,2}\} \geq 2\Delta/4 \geq 3/2$ 。同时, $C_{n,3} > \min\{C_{n,1}, C_{n,2}\} - 1/2 \geq 1$ (否则, r_n 将分配至 b_3 泊位), 因而 $C^*(\sigma) \geq (4\min\{C_{n,1}, C_{n,2}\} + 1 + C_{n,3})/5 > (4\min\{C_{n,1}, C_{n,2}\} + 2)/5 \geq 4C_{\max}(\sigma)/5$ 。

Case 2. r_n 分配至泊位 b_3 , 表明 $r_n = 1$ 。根据前一个请求 r_{n-1} 的大小分如下两种情形。

Case 2.1. $r_{n-1} = \Delta$, 表明 $C_{n,1} = C_{n,2}$ 。如果 $C_{\max}(\sigma) = C_{n,1}$, 分析同上述分析情形Case1.1中 $C_{\max}(\sigma) = e_n$ 的情形。反之, 如果 $C_{\max}(\sigma) = C_{n,3} + 1$, 则可推出 $C_{n,1} > C_{n,3} + 1/2 = C_{\max}(\sigma) - 1/2$, 否则 r_n 将分配至 b_1 泊位; 在该情形下若 $C_{n,3} = 0$, 则 $C_{\max}(\sigma) = 1$, 表明序列 σ 仅由1个小请求和1个大请求组成, 从而有 $C^*(\sigma) = C_{\max}(\sigma)$; 否则, 若 $C_{n,3} \geq 1$, 则 $C_{\max}(\sigma) \geq 2$, 从而 $C^*(\sigma) \geq (4C_{n,1} + C_{n,3} + 1)/5 > (4(C_{n,3} + 1/2) + 1 + 1)/5 = 4C_{\max}(\sigma)/5$ 。

Case 2.2. $r_{n-1} = 1$ 。如果 r_{n-1} 满足策略C2.1情形而分配至 b_2 泊位, 则 $C_{n,1} = C_{n,2}$ 。当 $C_{\max}(\sigma) = C_{n,1}$ 时, $C^*(\Delta) \geq 4C_{\max}(\Delta)/5$; 当 $C_{\max}(\Delta) = C_{n,3} + 1 < C_{n,1} + 1/2$ 时, 结合 $C_{n,3} + 1/2 < C_{n,1}$ 与 $C_{n,1} < C_{n,3} + 1$, 可以推出 $C_{n,3} \geq 1$ (否则 $1/2 < C_{n,1} = C_{n-1,1} < 1$,

$0 < C_{n-1,2}(=C_{n-1,1}-1/2) < 1/2$, 不等式矛盾), 从而得出 $C^*(\sigma) \geq (4C_{n,1} + C_{n,3} + 1)/5 \geq 4C_{\max}(\sigma)/5$; 如果 r_{n-1} 满足策略 C2.2 情形而分配至 b_1 泊位, 则 $C_{n,1} = C_{n,2} + 1/2$ 。由情形 (2) 的条件知 $C_{\max}(\sigma) = C_{n,1} > C_{n,3} + 1$ (否则, r_n 将分配至中间泊位 b_2)。 $C^*(\sigma) \geq (4C_{n,2} + 1 + C_{n,3} + 1)/5 \geq 4C_{n,1}/5 = 4C_{\max}(\sigma)/5$ 。

综上, 对于所有情形, $C_{\max}(\sigma)/C^*(\sigma) \leq 5/4$ 成立。定理得证。 证毕
通过对定理 3-5 的分析可知, GR₁SI 策略目标值与离线最优值比值不大于 5/4。

下面进一步通过数值仿真分析来验证在线策略的执行性能。针对策略 GR₁SI, 用 MATLAB 仿真实验; 用编程随机产生一些输入序列, 分析一下仿真数值中策略的目标值与最优值的平均比值 (通常比竞争比要好多)。

仿真环境如下: 仿真平台搭建在一台 CPU 为 3GHz, 内存为 4GB 的 PC 机上, 软件平台为 MATLAB R2014b。

仿真思路如下: 针对只存在大、小两种类型船舶的情形, 随机产生一些序列, 该序列只包含 1 和 Δ (正整数) 两种组合, 用 1 来表示小请求的任务量; Δ 来表示大请求的任务量; 根据该序列的随机情形, 应用所设计的 GR₁SI 策略, 可以得到一个策略的目标函数值; 根据这些已知的序列信息, 从而可以求出该序列最优完工时间的下界值, 两个目标函数值的比值得出仿真的结果。

为了减小不确定随机序列的影响, 让策略循环 100 次, 求出策略的目标值与离线最优下界值的平均比值; 为了更好的探讨策略的效果, 本节拟采用两种思路。

思路 1: 探讨随机任务的长度 n 的变化对策略性能的影响; 针对给定的大请求任务量 Δ 为 5, n 分别产生 50、100、150、200、250、300、350、400、450、500 这 10 组随机数组合进行仿真模拟; 对每个 n 的取值, 让策略算法随机运行 100 次, 并针对不同的比值取其平均比值; 算法程序设计的代码与步骤见附录 1, 得到数值如下表 3-1 所示。

表 3-1: 不同序列长度下的平均竞争比数值

序列长度	50	100	150	200	250	300	350	400	450	500
平均竞争比值	1.0559	1.0698	1.0487	1.0984	1.0628	1.0632	1.0649	1.0938	1.0850	1.0892

将表 3-1 对应的数据用折线图进行表示, 如图 3-1 所示。通过图 3-1 可知, 对于随机输入的服务序列, 在线策略目标函数值与离线最优值的平均比值在 [1.0487, 1.0984] 之间波动, 而理论竞争比为 1.25。这表明在线策略 GR₁SI 的平均执行性能对码头有一定的借鉴和指导作用。

思路 2: 针对特定的随机序列长度 (以 $n=150$ 为例), 对比分析大请求任务量 Δ 取值的不同对策略性能的影响情况。大请求任务量 Δ 的取值分别取 3、6、9、12、15、18、21、24、27、30, 这 10 组组合; 算法随机运行 100 次, 分别计算平均竞争比值, 得到数值如下表 3-2 所示。

表 3-2: 不同 Δ 取值下的平均竞争比数值

Δ 取值	3	6	9	12	15	18	21	24	27	30
平均竞争比	1.0812	1.0855	1.0677	1.0666	1.0487	1.0582	1.0586	1.0768	1.0623	1.0633

将表 3-2 对应的数据用折线图进行表示, 如图 3-2 所示。通过图 3-2 可知, 在线策略目标函数值与离线最优值的平均比值在 $[1.0487, 1.0855]$ 之间波动, 相比较于理论竞争比值 1.25, 随机产生序列的平均比值远小于最坏情形下的比值。

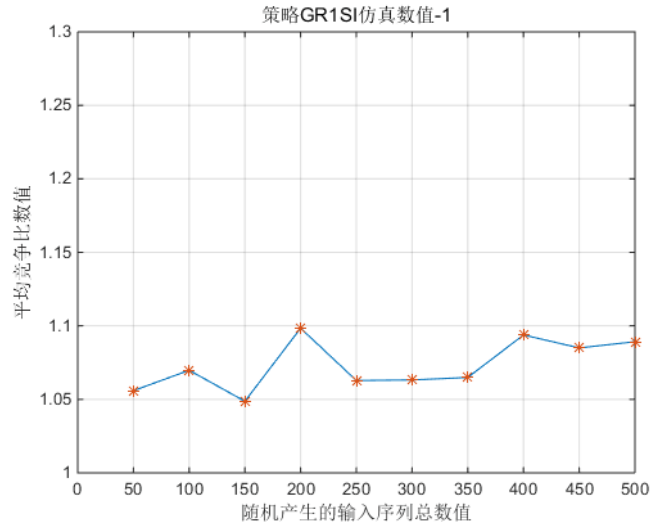


图 3-1: 输入序列长度与平均比值的关系

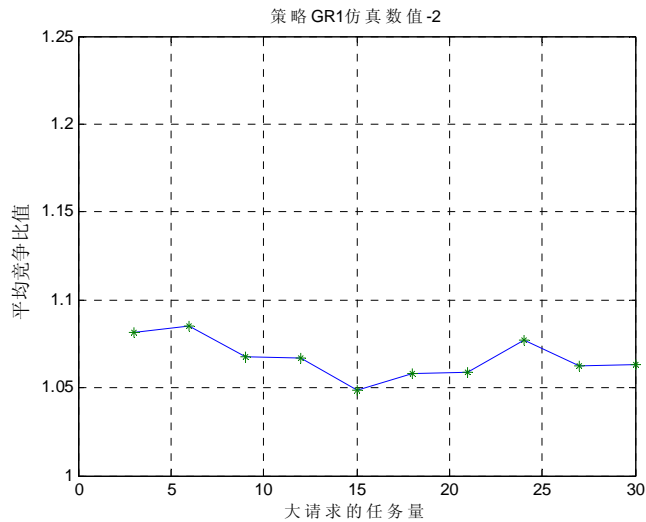


图 3-2: 大请求的任务量与平均比值的关系

3.3.3 $2 \leq \Delta < 3$ 的情形

对于 $2 \leq \Delta < 3$ 的情形, 将给出另一个在线策略 GR_2WL 。此时, 一个大请求 $r_i (= \Delta)$ 的分配可能导致 b_1 或 b_2 泊位及其配置的岸桥在服务该请求前产生一个空闲时间段 $[t_1, t_2) = [t_1, s_i)$, 相应的空闲泊位记为 b_{τ} 。若 $0 < t_2 - t_1 < 1/2$, 则记该时段

为 T_w ；若 $t_2 - t_1 \geq 1/2$ ，则记为 T_a 。在 T_a 时段内 b_T 泊位至少能够服务一个小请求；初始化 $T_a = T_w = [0, 0)$ 。下面给出策略 GR_2WL (Greedy 2 With Lookahead) 的具体描述。

策略 GR_2WL :

若 $r_i = r_n$ ，由于预知没有后续请求，分配 r_n 至某个泊位以使 C_{\max} 最小，若有多个泊位均导致相同的 C_{\max} ，则分配 r_n 至下标最小的泊位；否则，若 $i < n$ ，分两种情形调度。

C1. $r_i = \Delta$ 。若 $i=1$ 且 $r_2 = 1$ ，或者 $i=2$ 且 $r_1 = 1$ ，则令 $s_i = 0$ ，将 r_i 分配至 b_2, b_3 泊位；否则，令 $s_i = \max\{C_{i,1}, C_{i,2}\}$ ，将 r_i 分配至 b_1, b_2 泊位。若 $C_{i,1} - C_{i,2} = 1/2$ ，则令 $b_T = b_2$ 且 $T_a = [t_1, t_2) = [C_{i,2}, C_{i,1})$ ；若 $0 < |C_{i,1} - C_{i,2}| < 1/2$ ，则令 $T_w = [\min\{C_{i,1}, C_{i,2}\}, \max\{C_{i,1}, C_{i,2}\})$ 。

C2. $r_i = 1$ 。若 $T_a = [t_1, t_2) \neq [0, 0)$ ，则令 $s_i = t_1$ ，将 r_i 分配至泊位 b_T 并重新设置 $T_a = [0, 0)$ ；若 $r_{i+1} = \Delta$ 且 $\max\{C_{i,1}, C_{i,2}\} - C_{i,3} + \Delta/4 \geq 1$ ，则分配 r_i 至泊位 b_3 且令 $s_i = C_{i,3}$ ；否则，若以上两种情形均不满足，则不论 $r_{i+1} = \Delta$ 或 1，按以下两种情形进行调度：

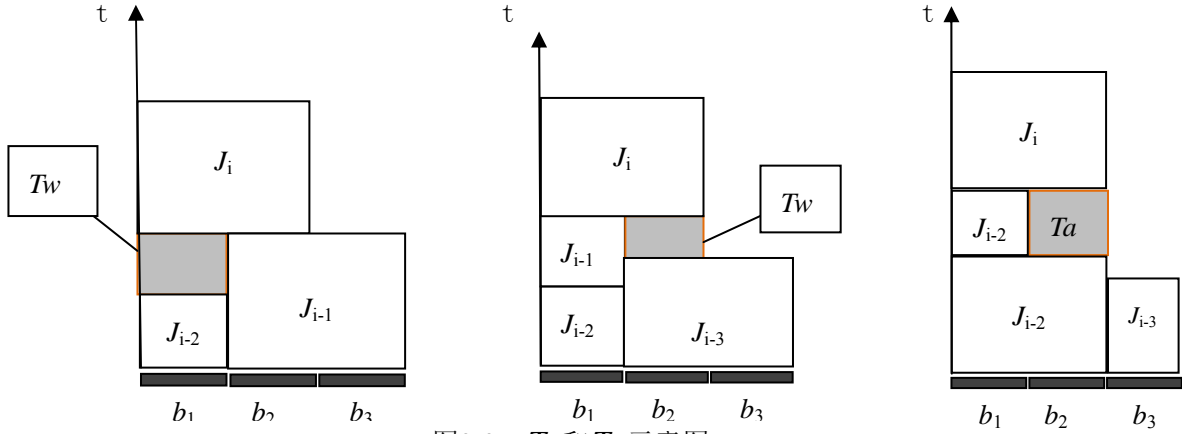
C2.1. $t_{i,1} = t_{i,3}$ 或 $t_{i,1} < t_{i,2} = t_{i,3}$ 。令 $s_i = C_{i,1}$ ，将 r_i 分配至泊位 b_1 。

给定任意请求输入序列 $\sigma = \{r_1, r_2, \dots, r_n\}$ ，根据上述策略中的 C1 情形，只有在 $r_1 = \Delta$ 且 $r_2 = 1$ 或者 $r_1 = 1$ 且 $r_2 = \Delta$ 的情形下，第 1 个大请求才被分配至右边两个泊位；否则，所有大请求均被分配至左边两个泊位。同时，只有当第 1 个大请求分配至右边两个泊位时，才可能由第 2 个大请求产生一个 T_w 时间段。对于左边两个泊位所分配的任意两个相邻大请求，它们之间若存在 1 个空闲时段，根据策略 C2.1 情形，必定是后一个大请求释放前给 b_1 泊位分配了一个小请求所致，因此该空闲泊位 b_T 必定是 b_2 ，对应空闲时段长度等于 $1/2$ 。综上， GR_2WL 策略所产生的加工序列中至多出现一个 T_w 时段。其中， T_w 和 T_a 如下图 3-3 所示。

下面的定理给出策略 GR_2WL 在 $2 \leq \Delta < 3$ 情形下的竞争性能。

定理3-6. 当 $n_q = 5$ ，对于问题 $hybr | online-over-list, LD=1 | BAP-QCAP | C_{\max}$ ，且当大请求任务量 $2 \leq \Delta < 3$ 时，策略 GR_2WL 具有竞争比为 $5/4$ 。

证明. 给定任一请求输入序列 $\sigma = \{r_1, r_2, \dots, r_n\}$ ，令 $C_{\max}(\sigma)$ 为策略 GR_2WL 的最大完工时间， $C^*(\sigma)$ 为离线最优方案对应的目标函数值。讨论以下3种情形：


 图3-3: T_a 和 T_w 示意图

Case 1. 序列 σ 中没有大请求。根据策略中的C2.1和C2.2情形可知, r_1, r_2, \dots, r_{n-1} 均分配至左边两个泊位即 $C_{n,3}=0$ 。若 r_{n-1} 分配至 b_1 泊位, 则 r_n 将分配至 b_2 泊位, b_1, b_2 泊位的完成时间相同, 从而 $C_{\max}(\sigma) = C_{n,1}$ 且 $C^*(\sigma) \geq 4C_{\max}(\sigma)/5$; 若 r_{n-1} 分配至 b_2 泊位且 r_n 分配至 b_3 泊位, 则分析同上; 若 r_{n-1} 分配至 b_2 泊位且 r_n 分配至 b_1 泊位, 因为 $C_{n,3}=0$, 所以 $C_{n,1}=1/2$ (否则 r_n 将分配至 b_3 泊位) 且 σ 中仅由3个小请求组成, 此时, $C^*(\sigma) = C_{\max}(\sigma) = 1$ 。

Case 2. 序列 σ 中有且仅有1个大请求。在该情形下 $T_w = [0, 0)$ 。

Case 2.1. $r_1 = r_2 = 1$ 。不妨设 $r_k = \Delta$ ($2 < k \leq n$)。

Case 2.1.1. $k=3 < n$ 。则 $s_3 = C_{2,1} = C_{2,2} = 1/2$ 。由于 $C_{n,1} \geq C_{n,2} \geq e_3 \geq 1$ 及 $C_{n,3}=0$, r_n 必定分配至 b_3 泊位。对 $C_{n,1} = C_{n,2}$ 与 $C_{n,1} = C_{n,2} + 1/2$ 两种情形, 均可得 $C^*(\sigma) \geq 4C_{n,1}/5 = 4C_{\max}(\sigma)/5$ 。

Case 2.1.2. $3 < k = n$ 。由于 $\min\{C_{n-1,1}, C_{n-1,2}\} \geq 1/2$, $C_{n-1,3}=0$ 且 $C_{n-1,1} + \Delta/4 \geq 1$, 表明 r_{n-1} 必定分配至 b_3 泊位即 $C_{n,3}=1$ 。如果 r_n 分配至 b_1, b_2 泊位, 不论分配后是否 $T_a \neq [0, 0)$, 均有 $C_{\max}(\sigma) = C_{n,1} + \Delta/4$ 且 $C^*(\sigma) \geq 4C_{\max}(\sigma)/5$; 如果 r_n 分配至 b_2, b_3 泊位, 则 $C_{n,1} = C_{n,2} + 1/2$, $C_{\max}(\sigma) = C_{n,2} + \Delta/3$, 且 $C^*(\sigma) \geq (4C_{n,2} + 1 + \Delta + C_{n,3})/5 > 4C_{\max}(\sigma)/5$ 。

Case 2.1.3. $3 < k < n$ 。类似于 (2.1.2) 情形的分析有 $C_{n,3}=1$ 。若 r_n 分配至 b_1 泊位, 则 $C_{n,1} = C_{n,2}$, $C_{\max}(\sigma) = C_{n,1} + 1/2$, $C^*(\sigma) \geq (4C_{n,1} + 1 + C_{n,3})/5 = 4C_{\max}(\sigma)/5$; 若 r_n 分配至 b_2 泊位, 则 $C_{\max}(\sigma) = C_{n,1}$, $C^*(\sigma) \geq (4C_{n,1} + C_{n,3})/5 > 4C_{\max}(\sigma)/5$; 若 r_n 分配至 b_3 泊位且 $C_{\max}(\sigma) = C_{n,1}$, 分析同上; 若 r_n 分配至 b_3 泊位且 $C_{\max}(\sigma) = C_{n,3} + 1 = 2$, 则 $C_{n,1} > C_{n,3} + 1/2 = 3/2$, $C^*(\sigma) \geq (4C_{n,1} + (C_{n,3} + 1))/5 \geq 8/5 = 4C_{\max}(\sigma)/5$ 。

Case 2.2. r_1, r_2 中有且仅有一个大请求。则该大请求必定分配至泊位 b_2, b_3 , 根据策略中的C2.1和C2.2情形描述, r_n 之前的所有小请求均分配至 b_1 或 b_2 泊位, 表明 $C_{n,3} = \Delta/3$ 。

Case 2.2.1. r_n 分配至 b_1 或 b_2 泊位。结合 $2/3 \leq \Delta/3 < 1$ 与 $C_{n,3} = \Delta/3$ 可得, $C_{n,1} \leq 3/2$ 且 $C_{n,2} \leq \Delta/3 + 1/2$ (否则, r_n 分配至 b_3 泊位), 从而序列 σ 由1个大请求和至多5个小请求组成即 $n \leq 6$ 。对于 $2 \leq n \leq 6$, 均可验证 $C_{\max}(\sigma)/C^*(\sigma) < 5/4$ 。

Case 2.2.2. r_n 分配至 b_3 泊位。结合Case2.2.1中情形的分析知 $n > 6$ 且有 $C_{\max}(\sigma) = \max\{C_{n,1}, C_{n,2}\} \geq C_{n,3} + 1$ 。由于 $|C_{n,1} - C_{n,2}| < 1/2$, $C^*(\sigma) \geq (2(C_{n,1} + C_{n,2}) + C_{n,3} + 1)/5 > (4C_{\max}(\sigma) - 1 + C_{n,3} + 1)/5 > 4C_{\max}(\sigma)/5$ 。

Case 3. σ 中至少包含2个大请求。

Case 3.1. $T_w = [0, 0)$ 。若 r_1, r_2 中有且仅有1个大请求, 则由 $T_w = [0, 0)$ 可推断, σ 中仅有2个大请求且 $r_n = \Delta$ 并分配至 b_2, b_3 泊位。 $C_{\max}(\sigma) = C_{n,2} + \Delta/3$, 由于 $C_{n,3} = \Delta/3$, 因而 $C^*(\sigma) > (4C_{n,2} + \Delta + C_{n,3})/5 = 4C_{\max}(\sigma)/5$ 。下面讨论 $r_1 = r_2 = 1$ 的情形。在该情形中, 序列 σ 中的所有大请求均分配至 b_1, b_2 泊位, 除非 $r_n = \Delta$ 。

Case 3.1.1. r_n 分配之后 $T_a \neq [0, 0)$ 。则在该 T_a 出现之后的所有请求均为大请求; 同时, 根据策略中的C2情形可以判定 $C_{n,3} \geq 1$ (否则, 若 $C_{n,3} = 0$, 则 T_a 期间 b_1 泊位上加工的小请求 r_i 由于满足 $r_{i+1} = \Delta$ 且 $\max\{C_{i,1}, C_{i,2}\} - C_{i,3} + \Delta/4 \geq 1$ 将被分配至 b_3 泊位)。由于 T_a 的空闲时长等于 $1/2$, 在该时段内两个空闲岸桥的总加工能力为 $2 \times (1/2) = 1$, 因此, $C^*(\sigma) \geq (4C_{\max}(\sigma) - 1 + C_{n,3})/5 \geq 4C_{\max}(\sigma)/5$ 。

Case 3.1.2. r_n 分配之后 $T_a = [0, 0)$ 。若 $r_n = \Delta$ 且分配至 b_1, b_2 泊位, 则策略完工时间 $C_{\max}(\sigma) = C_{n,1} + \Delta/4$ 且 $C^*(\sigma) \geq 4C_{\max}(\sigma)/5$; 若 $r_n = \Delta$ 且分配至 b_2, b_3 泊位, 则 $C_{\max}(\sigma) = C_{n,2} + \Delta/3$, 结合 $1 > \Delta/3$ 与 $C_{n,1} - C_{n,2} = 1/2$, $C^*(\sigma) \geq (4C_{n,2} + 2 \times (1/2) + \Delta)/5 > 4C_{\max}(\sigma)/5$; 若 $r_n = 1$ 且 r_n 分配至 b_1 泊位, 则可得出 $C_{\max}(\sigma) = C_{n,1} + 1/2$, $C_{n,3} \geq C_{n,1} - 1/2 \geq (1/2 + 2\Delta/4) - 1/2 \geq 1$, $C^*(\sigma) \geq (4C_{n,1} + 1 + C_{n,3})/5 \geq 4C_{\max}(\sigma)/5$; 若 $r_n = 1$ 且分配后有 $C_{\max}(\sigma) = C_{n,1}$, 则 $C^*(\sigma) \geq 4C_{\max}(\sigma)/5$; 若 $r_n = 1$ 且分配至 b_3 泊位后有 $C_{\max}(\sigma) = C_{n,3} + 1$, 则 $C_{n,2} > C_{n,3} + 1/2$ 。类似于Case2.1.3情形的分析有 $C_{n,3} > 1$, 从而 $C^*(\sigma) \geq (4C_{n,2} + C_{n,3} + 1)/5 > ((4C_{n,3} + 2) + 1 + 1)/5 = 4C_{\max}(\sigma)/5$ 。

Case 3.2. $T_w \neq [0, 0)$ 。则 r_1, r_2 中有且仅有一个大请求且其分配至泊位 b_2, b_3 。根据定义, T_w 只出现在 b_1 或 b_2 泊位。在 T_w 之前, b_1 泊位只被分配了小请求且加工长度均为 $1/2$, 而 b_2 泊位分配的第1个请求是加工长度为 $\Delta/3$ 的大请求, 其他均为小请求。因此, 若 T_w 在 b_1 或 b_2 泊位, 则其长度分别为 $\Delta/3 - 1/2$ 和 $1 - \Delta/3$ 。结合条件 $2 \leq \Delta < 3$, $\max\{\Delta/3 - 1/2, 1 - \Delta/3\} \leq \Delta/6$, 表明在 T_w 期间两个空闲岸桥的总加工能力至多为 $\Delta/3$ 。

Case 3.2.1. r_n 分配之后 $T_a \neq [0, 0)$ 。则在该 T_a 出现之后的所有请求均为大请求。类似于Case3.1.1情形的分析可知 $C_{n,3} \geq \Delta/3 + 1$, $C_{\max}(\sigma) = C_{n,1} + \Delta/4$ 。由于 T_w 与 T_a 时段内两个空闲岸桥的总加工能力分别不超过 $\Delta/3$ 和 1 , 因此, $C^*(\sigma) \geq (4C_{\max}(\sigma) - \Delta/3 - 1 + C_{n,3})/5 \geq 4C_{\max}(\sigma)/5$ 。

Case 3.2.2. r_n 分配之后 $T_a = [0, 0)$ 。已知 $C_{n,3} \geq \Delta/3$ 且 T_w 期间两个空闲岸桥总加工能力至多 $\Delta/3$ 。若 $r_n = \Delta$ 且分配至 b_1, b_2 泊位, 则 $C_{\max}(\sigma) = C_{n,1} + \Delta/4$ 且 $C^*(\sigma) \geq (4C_{\max}(\sigma) - \Delta/3 + C_{n,3})/5 \geq 4C_{\max}(\sigma)/5$; 若 $r_n = \Delta$ 且分配至 b_2, b_3 泊位, 则 $C_{\max}(\sigma) = C_{n,2} + \Delta/3$, 结合 $1 > \Delta/3$ 与 $C_{n,1} - C_{n,2} = 1/2$, $C^*(\sigma) > (4C_{n,2} - \Delta/3 + 2 \times 1/2 + \Delta + C_{n,3})/5 > 4C_{\max}(\sigma)/5$; 若 $r_n = 1$ 且分配至 b_1 泊位, 则策略完工时间 $C_{\max}(\sigma) = C_{n,1} + 1/2 = C_{n,2} + 1/2$, $C_{n,3} \geq C_{n,2} + 1/2 \geq (\Delta/3 + \Delta/4) + 1/2 \geq \Delta/3 + 1$, 从而 $C^*(\sigma) \geq (4C_{n,1} - \Delta/3 + 1 + C_{n,3})/5 \geq 4C_{\max}(\sigma)/5$; 若 $r_n = 1$ 且分配后有 $C_{\max}(\sigma) = C_{n,1}$, 则 $C_{n,3} \geq \Delta/3$ 且 $C^*(\sigma) \geq (4C_{n,1} - \Delta/3 + C_{n,3})/5 \geq 4C_{\max}(\sigma)/5$; 若 $r_n = 1$ 且分配至 b_3 泊位后有 $C_{\max}(\sigma) = C_{n,3} + 1$, 则 $C_{n,1} = C_{n,2} > C_{n,3} + 1/2$ 。当 $C_{n,3} = \Delta/3$ 时, 由于 $C_{\max}(\sigma) = \Delta/3 + 1$, 且 σ 中包含至少2个大请求和2个小请求, 结合离线最优 $C^*(\sigma) \geq \min\{2\Delta/3, 1 + \Delta/4\}$ 可得, $C_{\max}(\sigma)/C^*(\sigma) \leq 5/4$; 当 $C_{n,3} \geq \Delta/3 + 1$ 时, $C^*(\sigma) \geq (4C_{n,2} - \Delta/3 + C_{n,3} + 1)/5 \geq 4C_{\max}(\sigma)/5$ 。

综上, 对于任意序列 σ 中, 总有 $C_{\max}(\sigma)/C^*(\sigma) < 5/4$ 。定理得证。 证毕

对于 $n_q = 5$ 的调度模型, 可结合上述两种在线策略设计出综合性调度策略: 对于 $\Delta \geq 3$ 的情形采用 GR_1SI 策略, 而对于 $2 \leq \Delta < 3$ 的情形采用 GR_2WL 策略。由定理3-5和定理3-6可知, 该综合性调度策略具有竞争比 $5/4$ 。

结合定理2-3、定理3-4、3-5和3-6可知, 当预知后续1个请求时, 在线策略的竞争性能可以被显著改进。

3.4 六个岸桥的调度情形

对于6个岸桥的情形, 本节先给出此问题的竞争比下界。

3.4.1 竞争比下界分析

定理3-7. 当 $n_q = 6$ 时, 问题 $hybr | online - over - list, LD = 1 | BAP - QCAP | C_{\max}$ 的竞争比下界为 $4/3$ 。

证明. 类似于定理3-4的证明, 同样定义 $C_{\max}(\sigma)$ 和 $C^*(\sigma)$, 且只需设计一个请求输入序列 σ , 使得对于任一在线策略 A 有 $C_{\max}(\sigma)/C^*(\sigma) \geq 4/3$ 成立。序列 σ 包含至多4个请求且 $r_1 = r_2 = r_3 = 1$ 。若策略 A 将 r_1, r_2 分配至同一个泊位由2个岸桥服务, 则 $\sigma = \{r_1, r_2, r_3\}$ 。 $C_{\max}(\sigma) \geq 1$, 而最优方案将分配3个小请求至3个不同泊位即 $C^*(\sigma) = 1/2$, 从而 $C_{\max}(\sigma)/C^*(\sigma) \geq 2$ 。反之, 若 A 将 r_1, r_2 分配至不同泊位, 则 $r_4 = \Delta$ 且 $\sigma = \{r_1, r_2, r_3, r_4\}$ 。因为 r_4 需占用两个相邻泊位, 无论 A 如何分配请求 r_3 , 均有 $C_{\max}(\sigma) \geq 1/2 + \Delta/4$ 。最优方案则将前3个请求分配至同一个泊位而分配 r_4 至另两个相邻泊位, $C^*(\sigma) = \max\{\Delta/4, 3/2\}$ 。令 $\Delta = 6$, 则 $C_{\max}(\sigma)/C^*(\sigma) \geq 4/3$ 。定理得证。 证毕

通过对定理3-7的证明，不存在任一在线策略具有竞争比小于4/3。

3.4.2 策略 GRTWL 及其竞争分析

对于6个岸桥的调度情形，设计另一个在线策略GRTWL (Greedy of Two quays With Lookahead)。该策略对每个泊位分别配置2个岸桥，即泊位 b_1 配置岸桥 q_1, q_2 ，泊位 b_2 配置 q_3, q_4 ，而泊位 b_3 配置 q_5, q_6 。由于三个泊位配备相同的岸桥数量，因此每个泊位的处理能力相同。首先，给出策略GRTWL的具体描述。

策略GRTWL:

若 $r_i = r_n$ ，由于预知没有后续请求，因而分配该请求至三个泊位之一使其完成时间 e_n 最小；否则，若 $1 \leq i < n$ ，由于预知后续请求 r_{i+1} ，分如下三种情形进行调度。

C1. $r_i = \Delta$ 。令 $s_i = t_{i,2}$ 。若 $t_{i,2} = C_{i,1}$ ，则 r_i 分配至 b_1, b_2 泊位；否则， r_i 分配至 b_2, b_3 泊位。

C2. $r_i = 1$ 且 $r_{i+1} = 1$ 。令 $s_i = t_{i,1}$ 。若 $t_{i,1} = t_{i,3}$ 或者 $C_{i,1} = t_{i,1} < t_{i,2} = t_{i,3}$ ，则 r_i 分配至 b_1 泊位；若 $t_{i,1} = t_{i,2} < t_{i,3}$ ， r_i 分配至 b_2 泊位；若 $C_{i,3} = t_{i,1} < t_{i,2} = t_{i,3}$ ， r_i 分配至 b_3 泊位。

C3. $r_i = 1$ 且 $r_{i+1} = \Delta$ 。

C3.1. $t_{i,1} = t_{i,3}$ 。令 $s_i = t_{i,1}$ ，且 r_i 分配至 b_1 泊位。

C3.2. $t_{i,1} = t_{i,2} < t_{i,3}$ 。令 $s_i = t_{i,3}$ ，若 $t_{i,3} = C_{i,1}$ ， r_i 分配至 b_1 泊位；否则 r_i 分配至 b_3 泊位。

C3.3. $t_{i,1} < t_{i,2} = t_{i,3}$ 。令 $s_i = t_{i,1}$ ，若 $t_{i,1} = C_{i,1}$ ， r_i 分配至 b_1 泊位；否则 r_i 分配至 b_3 泊位。

对于策略GRTWL，若输入序列中存在至少2个大请求，则最大完工时间具有如下特性。

引理3-1. 当 $n_q = 6$ 时，针对 $hybr | online - over - list, LD = 1 | BAP - QCAP | C_{\max}$ 问题，如果序列 σ 中存在 $u (\geq 2)$ 个大请求和 $v = 3k, 3k+1, 3k+2 (k \in \mathbb{Z})$ 个小请求，则有 $C_{\max}(\sigma) \leq u\Delta/4 + v/6$ 。

证明. 考虑任一请求输入序列 $\sigma = \{r_1, r_2, \dots, r_n\}$ ，令 $C_{\max}(\sigma)$ 和 $C^*(\sigma)$ 分别表示在线策略GRTWL和离线最优方案对应的目标函数值。根据策略中的C3情形，若 $r_i = 1$ 且 $r_{i+1} = \Delta$ ，则 r_i 分配之后必定有两个相邻泊位其当前分配任务量相同，结合策略中的C1情形，大请求 r_{i+1} 所分配的两个泊位在 s_{i+1} 之前不会产生空闲时间段，即这两个泊位截至 $s_{i+1} + \Delta/4$ 时刻一直处于加工状态。注意每个大、小请求的服务时长各为 $\Delta/4$ 和 $1/2$ 。

根据引理假设, 序列 σ 中存在 $u \geq 2$ 个大请求。讨论如下三种情形:

Case1. 所有大请求位于序列 σ 的最后, 即 $r_{n-u+1} = \dots = r_n = \Delta$ 。则最后1个小请求 $r_{n-u} (= r_v)$ 释放时, 针对 $v = 3k, 3k+1, 3k+2$ 的三种情形, 分别满足策略中的C3.3, C3.1和C3.2条件, 因而分别有最后0, 1, 2个小请求的服务时间落在 u 个大请求的服务时间段内, 而此前的 $v, v-1, v-2$ 个小请求所需要的服务总时长等于 $3k/6 < v/6$, 因而 $C_{\max}(\sigma) \leq u\Delta/4 + v/6$ 。

Case 2. 序列 σ 中的第1个大请求之后只有1个小请求, 记为 r_l 。

Case 2.1. $v = 3k$ 。对 r_l 之前的 $v-1$ 个小请求的讨论类似于情形Case 1, 即小请求 $r_{n-u-1} (= r_{v-1})$ 释放时满足策略中的C3.2条件, 因此, 第1个大请求之前释放的最后2个小请求, 其服务时间将落在 u 个大请求的服务时间段内。此前 $v-3$ 个小请求的服务时长为 $3(k-1)/6 = v/6 - 1/2$ 。因此, 按策略设计的算法, r_l 启动时间 $s_l = (v/6 - 1/2) + 1 = v/6 + 1/2$, 结束时间 $e_l = v/6 + 1 \leq v/6 + u\Delta/4$; 最后1个大请求结束时间为 $(v/6 - 1/2) + u\Delta/4$, 从而 $C_{\max}(\sigma) \leq u\Delta/4 + v/6$ 。

Case 2.2. $v = 3k+1$ 或 $3k+2$ 。对 r_l 之前的 $v-1$ 个小请求的讨论类似于上一情形, 即小请求 $r_{n-u-1} (= r_{v-1})$ 释放时, 针对 $v = 3k+1, 3k+2$ 的两种情形分别满足策略中的C3.3和C3.1条件, 因此, 在第1个大请求之前释放的最后0, 1个小请求, 其服务时间落在 u 个大请求的服务时间段内。已知在 u 个大请求的服务时间段内第3个泊位至少可以完成2个小请求, 因此 r_l 的服务时间也落在 u 个大请求的服务时间段内, 从而共有1, 2个小请求的服务时间段处于 u 个大请求的服务时间段内; 其余的 $v-1, v-2$ 个小请求所需服务时长为 $3k/6 < v/6$, 从而 $C_{\max}(\sigma) \leq u\Delta/4 + v/6$ 。

Case 3. 序列 σ 中的第1个大请求之后至少有2个小请求。由于每个大请求服务时长为 $\Delta/4 \geq 1/2$, 因而在某两个相邻泊位服务 u 个大请求的时间段内, 另一个泊位至少可以完成最后这2个小请求。其余 $v-2 \leq 3k$ 个小请求的服务时长至多为 $3k/6 < v/6$ 。从而, $C_{\max}(\sigma) \leq u\Delta/4 + v/6$ 。

基于上述三种情形的讨论, 引理成立。

证毕

通过对引理3-1的证明, 可以得到下面的定理3-8。

定理3-8. 当 $n_q = 6$ 时, 对于问题 $hybr | online - over - list, LD = 1 | BAP - QCAP | C_{\max}$, GRTWL策略具有竞争比为4/3。

证明. 给定任一包含 $u \geq 0$ 个大请求和 $v \geq 0$ 个小请求的输入序列 σ , 对 $C_{\max}(\sigma)$ 和 $C^*(\sigma)$ 的定义同引理1。首先, 如果 $u = 0$ 或者1, 容易验证 $C_{\max}(\sigma)/C^*(\sigma) \leq 4/3$ 。

其次, 如果 $u \geq 2$, 由引理1可知, $C_{\max}(\sigma) \leq u\Delta/4 + v/6$ 。若 $u\Delta/4 \geq v/2$, 结合 $C^*(\sigma) \geq u\Delta/4$ 可得, $C_{\max}(\sigma)/C^*(\sigma) \leq 4/3$; 反之, 若 $u\Delta/4 \leq v/2$, 结合 $C^*(\sigma) \geq (u\Delta + v)/6$,

$$\frac{C_{\max}(\sigma)}{C^*(\sigma)} \leq \frac{u\Delta/4 + v/6}{u\Delta/6 + v/6} \leq \frac{u\Delta/4 + u\Delta/12}{u\Delta/6 + u\Delta/12} = \frac{4}{3}。$$

定理得证。证毕

结合定理2-5、3-7和3-8，当预知后续1个请求时，在线策略的竞争性能可以被显著改进。

3.4.3 数值分析

下面通过数值计算进一步验证在线策略GRTWL的执行性能。利用计算机随机产生一些输入序列，对每个序列计算策略GRTWL的运行结果与最优值的比值。算法程序设计的过程与步骤见附录2。具体的实验计算设计和参数设置如下：

首先，为了减小序列请求组合不确定性对目标值的影响，每一个参数配置下随机生成10个需求序列，计算10个序列下在线策略解与最优解的平均比值；其次，给定的大请求的任务量 $\Delta=5$ ，对比分析不同的序列长度对平均竞争比值的影响，分别取序列长度 $n=50、100、150、200、250、300、350、400、450、500$ 等10个值。对每个序列长度取值，计算平均比值；进而，针对特定的随机序列长度（设 $n=150$ ），对比分析大请求任务量 Δ 的不同取值对平均竞争比值的影响，分别取 $\Delta=2、4、6、8、10、12、14、16、18、20$ 等10个值。

首先，设定大请求任务量 $\Delta=5$ ，针对上述给出的10种序列长度，分别随机产生10组序列计算平均竞争比值，得到表3-3所示的结果。

表 3-3：不同序列长度下的平均竞争比值

随机数列 的取值	50	100	150	200	250	300	350	400	450	500
竞争比取 平均值	1.052 2	1.056 0	1.081 3	1.099 0	1.078 8	1.084 9	1.098 1	1.080 2	1.086 0	1.090 4

将表3-3对应的数据用折线图进行表示，如图3-4所示。由图3-4可知，在线策略目标函数值与离线最优值的平均比值在 $[1.0522, 1.0990]$ 之间波动，相比较于理论竞争比值 $4/3$ ，表明在线策略目标函数值与离线最优值的平均比值远小于最坏情形下的理论竞争比。

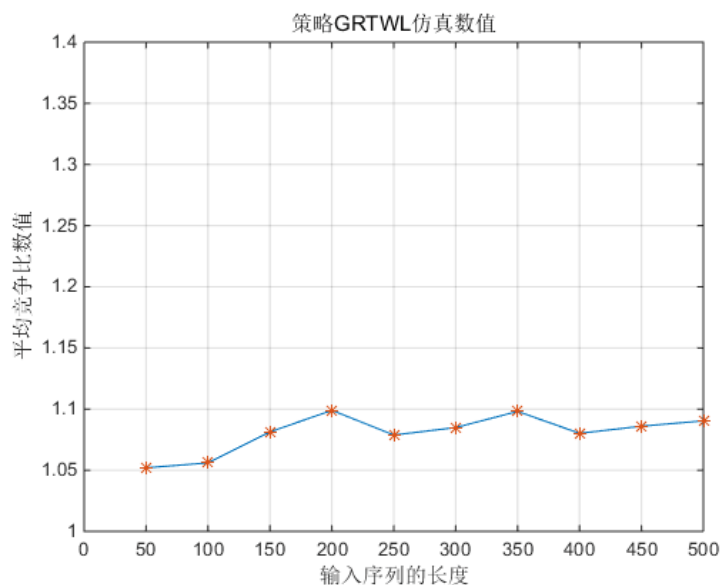


图 3-4: 输入序列总数与竞争比取平均值的关系

其次, 设定序列长度 $n=50$, 针对前面给出的 10 组 Δ 取值, 即 $\Delta=2、4、6、8、10、12、14、16、18、20$, 分别随机产生 10 组序列计算平均竞争比值, 得到表 3-4 所示的结果。

表 3-4: 不同 Δ 取值下的平均竞争比数值

大请求 Δ 的取值	2	4	6	8	10	12	14	16	18	20
竞争比的 平均值	1.104 5	1.088 4	1.068 7	1.061 6	1.040 8	1.063 0	1.067 2	1.038 0	1.037 2	1.035 5

将表 3-4 对应的数据用折线图进行表示, 如图 3-5 所示。

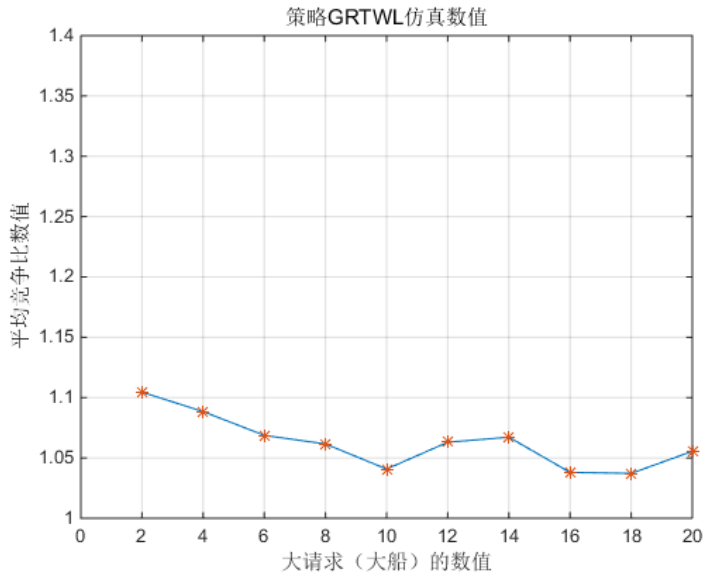


图 3-5: 大请求的数值（船舶的载重量）与竞争比取平均值的关系

通过图 3-5 可知, 对于随机输入的服务序列, 在线策略目标函数值与离线最优值的平均比值在 $[1.0372, 1.1045]$ 之间波动, 而理论竞争比为 $4/3$; 实际上, 随机产生序列请求的在线策略目标函数值与离线最优值的平均比值远小于最坏情形下的理论竞争比; 表明在线策略 GRTWL 对码头有一定的借鉴和指导作用。

3.5 本章小结

提出了有限预知信息下的集装箱码头泊位与岸桥联合调度在线模型, 探讨了 1 个混合型泊位布局且存在两种大小船舶服务需求的情形。对于配置 4、5 个、6 个岸桥的两种情形, 针对模型中预知信息的特征分别设计给出了具有最优竞争比 $(1+\sqrt{2})/2$ 、 $5/4$ 、 $4/3$ 的在线调度策略; 同时, 证明了在没有预知时的相关模型的竞争比下界分别 $5/4$ 、 $4/3$ 与 $3/2$, 该结论表明即使十分有限的预知能力也可以有效地改进在线调度策略的竞争性能。

第4章 预知 $k \geq 2$ 个请求下泊位与岸桥联合在线调度

本章在第三章预知后续一艘船舶需求的基础上,进一步探讨预知多艘船舶请求的联合调度计划。同样讨论了3个相邻泊位,4、5、6个岸桥的情形。对于每个情形,分别设计了预知 $k=2$ 请求下的联合调度的在线策略,并分析其竞争性能;同时也分析了预知 $k \geq 3$ 请求下的联合调度策略的竞争比下界,并比较了预知 $k=2$ 请求下的联合调度策略与预知较多请求情形下在线策略性能上的差距。

4.1 问题描述与模型刻画

现实中各艘船舶每天动态抵港,码头往往只提前一两天才能准确获知即将抵港的船舶信息,调度员通常是前一天决策第二天的详细调度计划,这种滚动调度伴随着一些干扰因素使其面临着因信息不完全而偏离最优方案的风险。本章采用 Zhang等^[54]的建模方法,运用在线理论来刻画码头泊位与岸桥的联合实时调度问题,根据船舶动态到达特征构建over-list在线调度模型。如前所述,码头调度作业中通常每天制定次日的调度计划,制定计划时考虑了当天及次日抵港的多艘船舶信息,因此,相应地构建预知多个船舶请求下的在线联合调度模型,并设计具有竞争性的调度策略。具有多个预知信息的在线模型在生产、管理调度领域已有相关研究。Mao和Kincaid^[59]、Mandelbaum和Shabtay^[60]分别研究了单机与平行机环境下,在线策略每次决策时预知后续 k 个请求的情形,研究表明预知能力可以有效改进在线策略的竞争性能。Wang^[61]研究了在线矩形填充问题,并讨论了 $k \geq 2$ 预知信息下确定性在线算法和随机在线算法。

本章预知 $k \geq 2$ 个请求下的模型假设基本相同于第二章无预知情形下的模型假设,所不同的一条假设在于:

- (1) 当请求 r_i ($1 \leq i \leq n-k+1$) 释放时可预知后续 k 个请求 r_{i+1}, \dots, r_{i+k} 的信息,且当 r_{n-k+1} 释放时即可获知服务请求序列 σ 的长度即 n 的取值。

运用 Bierwirth 和 Meisel^[4] 的四参数描述法,将上述模型标记为 $hybr|online-over-list, LD=k|BAP-QCAP|C_{\max}$ 。其中 $hybr$ 表示混合型泊位, $online-over-list$ 表示请求逐个释放的在线模型, $LD=k$ 表示在线策略每个时刻预知后续 k (≥ 2) 个请求,以及 C_{\max} 表示目标函数为最大完工时间。

本章接下来分别对4、5、6个岸桥的情形分别进行分析探讨。

4.2 四个岸桥的调度情形

下面分析具有4个岸桥的联合调度模型;先通过特殊序列构造法得出预知 $k \geq 2$ 个请求下的竞争比下界,针对预知2个请求的特殊情形,结合其竞争比下界值设计出具有最优性能的在线策略。

4.2.1. 竞争比下界分析

下面分别讨论预知2个请求、预知 $k \geq 3$ 个请求的两种情形，分别给出竞争比下界的相关结论。

定理4-1. 对于问题 $hybr | online-over-list, LD=2 | BAP-QCAP | C_{\max}$ ，当预知 $k=2$ 个请求时，不存在任一在线策略具有竞争比小于 $9/8$ 。

证明. 要证明该定理，只需设计一个请求输入序列 σ ，使得对于任一在线策略 A 有不等式 $C_{\max}(\sigma)/C^*(\sigma) \geq 9/8$ 成立，其中 $C_{\max}(\sigma)$ 和 $C^*(\sigma)$ 分别为策略 A 的最大完工时间、离线最优调度策略 OPT 对应的目标函数值。令大请求的任务量 $\Delta=2$ 。

序列 σ 包含至多4个请求，前3个请求 $r_1=1, r_2=\Delta, r_3=\Delta$ 。若策略 A 安排2个岸桥服务请求 r_1 ，则序列 $\sigma=\{r_1, r_2, r_3\}$ 。 $C_{\max}(\sigma) \geq 1/2 + \Delta/2$ ， OPT 将安排1个岸桥服务请求 r_1 ，另安排3个岸桥服务请求 r_2, r_3 ，即 $C^*(\sigma) = \max\{1, 2\Delta/3\}$ 。因此， $C_{\max}(\sigma)/C^*(\sigma) \geq 9/8$ 。反之，若策略 A 安排1个岸桥服务 r_1 ，则释放最后一个请求 $r_4=1$ 且序列 $\sigma=\{r_1, r_2, r_3, r_4\}$ 。 $C_{\max}(\sigma) \geq \min\{\max(2, 2\Delta/3), \Delta/4 + 1/2 + \max(1, \Delta/3)\}$ 。 OPT 最优方案将分别安排2个岸桥服务请求 r_1, r_4 ；安排4个岸桥服务请求 r_3 和 r_2 ，即 $C^*(\sigma) = 1/2 + \Delta/2$ 。此时，同样分析可得 $C_{\max}(\sigma)/C^*(\sigma) \geq 9/8$ 。定理得证。 证毕

通过定理4-1的证明，当预知 $k=2$ 个请求时，在线策略竞争比下界为 $9/8$ 。

定理4-2. 对于问题 $hybr | online-over-list, LD=k | BAP-QCAP | C_{\max}$ ，当预知未来请求的个数 $k(\geq 3)$ 为奇数时，不存在任一在线策略具有竞争比小于 $(3\sqrt{k^2+k/2+1/2+4k+1})/(4\sqrt{k^2+k/2+1/2+3k-1})$ 。

证明. 类似于定理4-1的证明，只需设计一个请求输入序列 σ ，使得对于任一在线策略 A 有不等式 $C_{\max}(\sigma)/C^*(\sigma) \geq (3\sqrt{k^2+k/2+1/2+4k+1})/(4\sqrt{k^2+k/2+1/2+3k-1})$ 成立，其中 $C_{\max}(\sigma)$ 和 $C^*(\sigma)$ 的定义同定理1。令大请求的任务量 $\Delta = (9k-3+\sqrt{k^2+k/2+1/2})/7$ 。

序列 σ 包含至多 $k+2$ 个请求， $r_1=\Delta$ 且其他请求的长度均为1。若策略 A 将 r_1 分配由4个岸桥服务，则序列 $\sigma=\{r_1, r_2, r_3, \dots, r_{k+1}\}$ ， $C_{\max}(\Delta) \geq \Delta/4 + (k-1)/4 + 1/2$ 。根据 Δ 的取值，最优策略 OPT 将分配3个岸桥服务 r_1 且分配另一个岸桥服务所有其他请求， $C^*(\sigma) = \Delta/3$ 。因此， $C_{\max}(\sigma)/C^*(\sigma) \geq (\Delta/4 + 1/4 + k/4)/(\Delta/3)$ 。

反之，若 A 分配3个岸桥服务 r_1 ，则序列 $\sigma=\{r_1, r_2, r_3, \dots, r_{k+1}, r_{k+2}\}$ ， $C_{\max}(\sigma) \geq \Delta/3 + 1/2$ 。 OPT 将分配4个岸桥服务 r_1 ，分配2个岸桥服务其余任一请求， $C^*(\sigma) = \Delta/4 + 1/4 + k/4$ ， $C_{\max}(\sigma)/C^*(\sigma) \geq (\Delta/3 + 1/2)/(\Delta/4 + k/4 + 1/4)$ 。

根据 Δ 的取值, $C_{\max}(\sigma)/C^*(\sigma) \geq \left(3\sqrt{k^2+k/2+1/2+4k+1}\right)/(4\sqrt{k^2+k/2+1/2+3k-1})$ 。定理得证。 证毕

定理4-3. 对于问题 $hybr|online-over-list, LD=k|BAP-QCAP|C_{\max}$, 当预知未来请求的个数 $k(\geq 3)$ 为偶数时, 不存在任一在线策略具有竞争比小于 $\left(3(\sqrt{4k^2+8k+9/4})/2+4k+23/4\right)/(2\sqrt{4k^2+8k+9/4}+3k+3)$ 。

证明. 证明思路同定理4-2, 即只需设计一个请求输入序列 σ , 使得对于任一在线策略 A 有 $C_{\max}(\sigma)/C^*(\sigma) \geq \left(3(\sqrt{4k^2+8k+9/4})/2+4k+23/4\right)/(2\sqrt{4k^2+8k+9/4}+3k+3)$ 。令大请求的任务量 $\Delta = (9k+9+6\sqrt{4k^2+8k+9/4})/7$ 。

序列 σ 包含至多 $k+2$ 个请求, $r_1 = \Delta$ 且其他请求长度均为1。若策略 A 分配4个岸桥服务 r_1 , 则序列 $\sigma = \{r_1, r_2, r_3, \dots, r_{k+1}, r_{k+2}\}$ 。 $C_{\max}(\sigma) \geq \Delta/4 + k/4 + 1/2$ 。最优策略 OPT 将分配3个岸桥服务 r_1 , 分配1个岸桥服务剩余的请求, $C^*(\sigma) = \Delta/3$, 因而, $C_{\max}(\sigma)/C^*(\sigma) \geq (\Delta/4 + k/4 + 1/2)/(\Delta/3)$ 。

反之, 若 A 分配3个岸桥服务 r_1 , 则 $\sigma = \{r_1, r_2, r_3, \dots, r_{k+1}\}$, 且 $C_{\max}(\sigma) \geq \Delta/3$ 。OPT 最优方案将分配4个岸桥服务 r_1 , 分配2个岸桥服务任一其他请求, $C^*(\sigma) = \Delta/4 + k/4$ 且有 $C_{\max}(\sigma)/C^*(\sigma) \geq (\Delta/3)/(\Delta/4 + k/4)$ 。根据 Δ 的取值, $C_{\max}(\sigma)/C^*(\sigma) \geq \left(3(\sqrt{4k^2+8k+9/4})/2+4k+23/4\right)/(2\sqrt{4k^2+8k+9/4}+3k+3)$ 。定理得证。 证毕

下表是 k 的不同取值下竞争比的下界值。

表 4-1: 不同预知能力下的竞争比下界值

k 取值	3	4	5	6	7	8	9	49	50
竞争比下界	1.079	1.052	1.049	1.036	1.035	1.028	1.027	1.005	1.005

由上表可以看出, 当预知能力 $k(\geq 3)$ 时, 该问题的竞争比下界值随着 k 的增大而减少, 最终趋近于1。

4.2.2 在线策略 GRWL₂ 的设计

接下来, 针对预知两个 ($k=2$) 请求的特定情形, 设计在线策略 GRWL₂ (Greedy With 2-Lookahead)。该策略的主要思想是: 在分配服务序列的最后2个请求之前, 泊位 b_1, b_2 分别配置岸桥 q_1, q_2 和 q_3, q_4 , 而将 b_3 泊位置于空闲状态; 对于最后两个请求则根据当前状态下的局部最优目标值进行泊位与岸桥的分配。下面给出策略 GRWL₂ 的具体描述。

策略 GRW

若 $r_i = r_{n-1}$ ，由于预知 r_n 且已知没有其他后续请求，因而对最后2个请求分配适当的泊位以及服务岸桥数量以使 C_{\max} 最小。对于 $1 \leq i \leq n-2$ 的请求，由于预知后续2个请求 r_{i+1}, r_{i+2} ，分如下两种情形进行讨论：

C1. $r_i = \Delta$ 。令 $s_i = \max\{c_{i,1}, c_{i,2}\}$ ，把 r_i 分到最左边的两个泊位。如果 $C_{i,1} > C_{i,2}$ ，令 $T_a = [t_1, t_2) = [C_{i,2}, C_{i,1})$ 且 $b_T = b_2$ 。

C2. $r_i = 1$ 。若 $T_a \neq [0, 0)$ ，分配 r_i 至空闲泊位 b_T 上的空闲时段 T_a ，重置 $T_a = [0, 0)$ ；反之，若 $T_a = [0, 0)$ ，则如果 $C_{i,1} > C_{i,2}$ ，令 $s_i = C_{i,2}$ 并把 r_i 分配至泊位 b_2 ；如果 $C_{i,1} = C_{i,2}$ ，令 $s_i = C_{i,1}$ 并把 r_i 分配至 b_1 泊位。

4.2.3 策略竞争分析

根据上述策略描述可知，因为泊位 b_1 和 b_2 均配置2个岸桥，每个小请求服务时间均为 $1/2$ 。当空闲时段 $T_a = [t_1, t_2) \neq [0, 0)$ 时，有 $t_2 - t_1 = 1/2$ ，即在时段 T_a 中岸桥资源服务能力为 $2 * (1/2) = 1$ 。

定理4-4. 对于 $hybr | online - over - list | BAP - QCAP | C_{\max}$ 问题，策略GRWL₂的竞争比为 $9/8$ 。

证明. 对于任何输入序列为 $\sigma = \{r_1, r_2, \dots, r_n\}$ ，令 $C_{\max}(\sigma)$ 为策略GRWL₂的目标函数值； $C^*(\sigma)$ 为最优调度方案的目标值。针对最后一个请求 r_n 的大小分两种情况讨论如下：

Case 1. $r_n = \Delta$ 。

Case 1.1. $T_a = [0, 0)$ 。若 $C_{n,1} = C_{n,2}$ 且 $C_{n,3} = 0$ ，因为在 $[0, C_{\max}(\sigma))$ 期间岸桥均不空闲，在这种情形下 $C_{\max}(\sigma) = C^*(\sigma) = C_{n,1} + \Delta/4$ ；若 $C_{n,1} > C_{n,2}$ （或 $C_{n,1} = C_{n,2} < C_{n,3}$ ），表明 $r_{n-1} = 1$ ， $C_{n-1,1} = C_{n-1,2}$ ，且 r_{n-1} 被分配至 b_1 （或 b_3 ）泊位。同理， $C_{\max}(\sigma) = C^*(\sigma)$ 成立。

Case 1.2. $T_a = [t_1, t_2) \neq [0, 0)$ 。假设在 t_2 时刻之后共有 $u \geq 1$ 个大请求到达，则 $C_{\max}(\sigma) = t_2 + u\Delta/4$ 。如果 $t_2 = 1/2$ 且 $u = 2$ ，表明序列 σ 中只有3个请求， $r_1 = 1$ ， $r_2 = \Delta$ 且 $r_3 = \Delta$ 。此时，策略目标完成时间 $C_{\max}(\sigma) = \Delta/2 + 1/2$ ；结合 $C^*(\sigma) \geq \min\{1/2 + \Delta/2, \max(1, 2\Delta/3)\}$ ，可得 $C_{\max}(\sigma)/C^*(\sigma) \leq 9/8$ 。如果 $t_2 = 1/2$ 且 $u = 3$ ，或者 $t_2 = 1$ 且 $u = 2$ ，容易验证比值不大于 $9/8$ 。如果 $t_2 = 1/2$ 且 $u \geq 4$ ，或者 $t_2 \geq 1$ 且 $u \geq 3$ ，则 $t_2 + u\Delta/4 \geq 9/4$ ，则 $C_{\max}(\sigma) = t_2 + u\Delta/4$ ， $C^*(\sigma) = t_2 + u\Delta/4 - 1/4$ 。因此， $C_{\max}(\sigma)/C^*(\sigma) \leq 9/8$ 。

Case 2. $r_n = 1$ 。此时，当 r_n 分配之后有 $T_a = [0, 0)$ 。

Case 2.1. $C_{n,1} = C_{n,2}$ 。若 r_n 分配之前 $T_a = [t_1, t_2) \neq [0, 0)$ ，则 $C_{\max}(\sigma) = C^*(\sigma) = C_{n,1}$ ；反之，若 r_n 分配之前 $T_a = [0, 0)$ ，表明序列 σ 在 r_n 之前释放偶数个小请求。此时，讨论以下两种子情形：

Case 2.1.1. r_{n-1} 由 3 个岸桥服务。此时, $C_{\max}(\sigma) = C^*(\sigma) = C_{n,1}$ (或 $C_{\max}(\sigma) = C^*(\sigma) = C_{n,1} - \Delta/3 + 1$, 当 $2 \leq \Delta < 3$)。

Case 2.1.2. r_{n-1} 由 4 个岸桥服务。此时, $C_{\max}(\sigma) = C_{n,1} + 1/2$ 且 $C^*(\sigma) \geq C_{n,1} + 1/4$ 。当 $C_{n,1} > 7/4$ 时, $C_{\max}(\sigma)/C^*(\sigma) \leq 9/8$; 当 $C_{n,1} \leq 7/4$ 时, 表明 r_n 前至多包含 3 个大请求或者 6 个小请求。对于满足 $C_{n,1} \leq 7/4$ 的不同请求组合均可验证, $C_{\max}(\sigma)/C^*(\sigma) \leq 9/8$ 成立。

Case 2.2. $C_{n,2} < C_{n,1}$ 。此时, 在 r_n 分配之前 $T_a = [0, 0]$ 且 $C_{n,1} = C_{n,2} + 1/2$ 。对于该情形有 $C_{\max}(\sigma) = C^*(\sigma)$ 。

综上, 对于任意情形, 均有 $C_{\max}(\sigma)/C^*(\sigma) \leq 9/8$ 。定理得证。 证毕

由定理4-4可知, 在预知 $k=2$ 请求的特殊情形下, 在线策略 GRWL₂ 的竞争比为 $9/8$; 等于定理4-1中的竞争比下界值, 从而该策略是最优的。

4.3 五个岸桥的调度情形

针对 5 个岸桥的情形, 同上节, 先分析预知 $k \geq 2$ 个请求下的竞争比下界, 然后针对预知 2 个请求的特殊情形, 结合其竞争比下界数值设计了具有最优性能的在线策略。

4.3.1 竞争比下界分析

下面分别讨论预知 2 个请求、预知 $k \geq 3$ 个请求的两种情形, 分别给出竞争比下界的相关结论。

定理4-5. 当 $n_q = 5$ 时, 问题 $hybr|online-over-list, LD=2|BAP-QCAP|C_{\max}$, 预知 $k=2$ 个请求时, 不存在任一在线策略具有竞争比小于 $6/5$ 。

证明. 类似于定理4-1的证明, 同样定义 $C_{\max}(\sigma)$ 和 $C^*(\sigma)$, 且只需设计一个请求输入序列 σ , 使得对于任一在线策略 A 有 $C_{\max}(\sigma)/C^*(\sigma) \geq 6/5$ 成立。序列 σ 包含至多 4 个请求且 $r_1 = \Delta$ 和 $r_2 = r_3 = r_4 = 1$ 。令 $\Delta = 3$ 。

策略 A 在分配 r_1 时预知 $r_2 = r_3 = 1$ 。若 A 分配 4 个岸桥服务 r_1 , 则 $\sigma = \{r_1, r_2, r_3\}$, 从而 $C_{\max}(\sigma) \geq \Delta/4 + 1/2 = 5/4$ 。最优方案将分配 3 个岸桥服务请求 r_1 , 那么 $C^*(\sigma) = 1$ 且 $C_{\max}(\sigma)/C^*(\sigma) \geq 5/4$; 反之, 若策略 A 分配 3 个岸桥服务 r_1 , 则 $r_4 = 1$ 且 $\sigma = \{r_1, r_2, r_3, r_4\}$ 。从而 $C_{\max}(\sigma) \geq \Delta/3 + 1/2 = 3/2$ 。最优方案则是分配 4 个岸桥服务请求 r_1 , $C^*(\sigma) = \Delta/4 + 1/2 = 5/4$ 。从而 $C_{\max}(\sigma)/C^*(\sigma) \geq 6/5$ 。定理得证。 证毕

通过定理4-5的证明, 当预知 $k=2$ 个请求时, 在线策略竞争比下界为 $6/5$ 。

定理4-6. 当 $n_q = 5$ 时, 问题 $hybr|online-over-list, LD=k|BAP-QCAP|C_{\max}$, 预知 $k \geq 3$ 个请求时, 不存在任一在线策略具有竞争比小于 $25/21$ 。

证明. 反证法。假设存在一个在线策略 A 具有竞争比 $25/21 - \varepsilon$ ，其中 ε 为任意小的正实数。只需设计一个请求输入序列 σ ，使得 $C_{\max}(\sigma)/C^*(\sigma) > 25/21 - \varepsilon$ ，其中 $C_{\max}(\sigma)$ 和 $C^*(\sigma)$ 分别为策略 A 的分配方案和最优分配方案所对应的最大完工时间。序列 σ 包含至多 $m+1$ 个请求且 $r_1 = r_2 = \dots = r_m = 1$ ，其中 $m = 5w$ (w 为自然数)；令 $\Delta = 4m = 20w$ 。

当策略 A 分配请求 r_{m-k} 之后，令 $C_{m-k,i^*} = \max\{C_{m-k,1}, C_{m-k,2}, C_{m-k,3}\}$ 。若 $C_{m-k,i^*} \geq 25w/21$ ，则 $\sigma = \{r_1, r_2, \dots, r_m\}$ 。在最优分配方案中每个泊位完成 w 个请求，因而 $C_{\max}(\sigma)/C^*(\sigma) \geq (25w/21)/(w) = 25/21$ ；反之，若 $C_{m-k,i^*} < 25w/21$ ，即 b_{i^*} 泊位当前累积的请求分配数量严格小于 $25w/21$ ，则 $r_{m+1} = \Delta$ 且 $\sigma = \{r_1, r_2, \dots, r_{m+1}\}$ ，此时，讨论如下两种情形。

Case 1. $i^* = 2$ 。即 r_{m-k} 分配之后中间泊位 b_2 的当前完工时间最大，则 b_2 至少已分配了 $2(m-k)/5$ 个小请求，表明 $C_{m-k,i^*} \geq (m-k)/5$ 。结合大请求 $\Delta = 4m = 20w$ ， $C_{\max}(\sigma) \geq C_{m-k,i^*} + \Delta/4 \geq 9w - k/5$ 。最优分配方案则是将最后一个大请求分配至两个相邻泊位而将所有小请求分配至另一个泊位， $C^*(\sigma) = \max\{\Delta/4, m\} = 5w$ 。当 $w \geq 21k/320$ 时， $C_{\max}(\sigma)/C^*(\sigma) \geq (9w - k/5)/(5w) \geq 25/21$ 。

Case 3. $i^* = 3$ ，已知分配完 r_{m-k} 之后 b_3 泊位上当前累积的请求数量严格小于 $25w/21$ ，则 b_2, b_3 泊位中请求分配量较多的泊位至少已分配 $(m-k - 25/21w)/2$ 个小请求，表明 $C_{\max}(\sigma) \geq \lceil (m-k - 25/21w)/2 \rceil/2 + \Delta/4$ ，最优方案则将前 m 个请求分配至同一个泊位而分配 r_{m+1} 至另两个相邻泊位， $C^*(\sigma) = \max\{\Delta/4, m\}$ 。令 $\Delta = 4m = 20w$ ，则 $C_{\max}(\sigma)/C^*(\sigma) \geq \{5w + \lceil (5w - 25w/21 - k)/2 \rceil/2\}/5w$ 。当 $w > k/(5\varepsilon)$ 时， $C_{\max}(\sigma)/C^*(\sigma) > 25/21 - \varepsilon$ 。

综上，不存在竞争比为 $25/21 - \varepsilon$ 的在线策略。定理得证。

证毕

下面将对预知 $k = 2$ 个请求的特殊情形，结合其竞争比下界值，设计相应的在线策略。对于5个岸桥的联合调度模型，在线策略将最左边的两个泊位 b_1, b_2 分别配置岸桥 q_1, q_2 和 q_3, q_4 ，而最右边的泊位 b_3 配置最右边的1个岸桥 q_5 。因此，若某个大请求被分配至泊位 b_2, b_3 ，则将由右侧3个岸桥 q_3, q_4, q_5 进行服务。由于该情形的复杂性，下文将对大请求任务量 $2 \leq \Delta < 10/3$ 、 $\Delta \geq 10/3$ 两种情形分别设计在线策略。

4.3.2 在线策略 GSRWL₂ 设计与竞争分析

基于上述对岸桥与泊位两种资源的固定配置方案，在线策略只需刻画分配每个请求至哪一个或哪两个泊位以及服务的启动时间即可。对于大请求任务量数值在 $2 \leq \Delta < 10/3$ 之间时，下面给出在线策略 GSRWL₂ (Greedy for Small Request With 2-Lookahead) 的具体描述：

策略GSRWL₂:

若 $r_i = r_{n-1}$ 或 r_n ，由于已预知最后的请求，因而分配该请求和后续请求至某个泊位以使 C_{\max} 最小，若有多个泊位均导致相同的 C_{\max} ，则分配 r_n 至下标最小的泊位；否则，若 $1 \leq i \leq n-2$ ，由于预知后续2个请求 r_{i+1}, r_{i+2} ，分如下五种情形进行调度。

C1. $r_i = \Delta$ 。令启动时间 $s_i = t_{i,2}$ 。若 $r_{i+1}, r_{i+2} = 1$ 且 $t_{i,1} = t_{i,3} = 0$ ，则把 r_i 分配至 b_2, b_3 泊位。否则令 $s_i = \max\{C_{i,1}, C_{i,2}\}$ ，把 r_i 分配至左边两个泊位 b_1, b_2 。若 $C_{i,1} - C_{i,2} = 1/2$ ，则令 $b_T = b_2$ 且 $T_a = [t_1, t_2) = [C_{i,2}, C_{i,1})$ ；若 $0 < |C_{i,1} - C_{i,2}| < 1/2$ ，则令 $T_w = [\min\{C_{i,1}, C_{i,2}\}, \max\{C_{i,1}, C_{i,2}\})$ 。

C2. $r_i = 1$ 。若 $T_a = [t_1, t_2) \neq [0, 0)$ ，则令 $s_i = t_i$ ，将 r_i 分配至泊位 b_T 并重新设 $T_a = [0, 0)$ ；

C2.1. $r_i = 1$ 且 $r_{i+1} = 1$ 。

C2.11. $t_{i,1} = t_{i,3}$ 。令 $s_i = t_{i,1}$ 。若 $r_{i+2} = 1$ ，把 r_i 分配至 b_3 泊位；若 $r_{i+2} = \Delta$ ，把 r_i 分配至 b_1 泊位。

C2.12. $t_{i,1} = t_{i,2} < t_{i,3}$ 。令 $s_i = t_{i,1}$ ，把 r_i 分配至 b_2 泊位。

C2.13. $t_{i,1} < t_{i,2} \leq t_{i,3}$ 。令 $s_i = t_{i,1}$ ，把 r_i 分配至 $t_{i,1}$ 所对应的泊位。

C2.2. $r_i = 1$ 且 $r_{i+1} = \Delta$ 。

C2.21. $t_{i,1} = t_{i,2} \leq t_{i,3} = C_{i,3}$ 。若 $r_{i+2} = 1$ ，令 $s_i = t_{i,1}$ ，把 r_i 分配至 b_1 泊位；若 $r_{i+2} = \Delta$ ，令 $s_i = t_{i,3}$ ，把 r_i 分配至 b_3 泊位。

C2.22. $t_{i,1} < t_{i,2} < t_{i,3} = C_{i,3}$ 或 $t_{i,1} < t_{i,2} = t_{i,3}$ 。令 $s_i = t_{i,1}$ ，把 r_i 分配至 $t_{i,1}$ 所对应的泊位。

C2.23. $t_{i,1} = t_{i,2} < t_{i,3} = C_{i,1}$ 。令 $s_i = t_{i,1}$ ，若 $t_{i,3} - t_{i,1} = 1/2$ ，把 r_i 分配至 b_2 泊位；否则，把 r_i 分配至 b_3 泊位。

给定任意请求输入序列 $\sigma = \{r_1, r_2, \dots, r_n\}$ ，由策略设计的思想，大请求会被分配至右边两个泊位的情形只有下面两个情况：

(i). 根据上述策略中的C1情形，只有在 $r_1 = \Delta$ 和 $r_2, r_3 = 1$ 的情形下，第1个大请求才被分配至右边两个泊位。

(ii). 最后一个大请求到达时，若此大请求是最后两个请求之一时，把其分至右边两个相邻泊位以使 C_{\max} 最小。否则，其他所有大请求均分配至左边两个泊位。

同时，只有当第1个大请求分配至右边两个泊位时，才可能由第2个大请求产生一个 T_w 时间段。 T_a 的产生是由大请求造成的，按策略设计的思想，序列的最后， $T_a = [0, 0)$ ；不然按策略C2.21和C2.22知， T_a 的产生是因为预知到未来至少有

一个小请求到达。若所有大请求分配至左边两个泊位，根据GSRWL₂策略描述，那么在完成最后一个大请求之前，左边2个泊位与4个岸桥一直处于服务状态且 b_1, b_2 泊位及相应的岸桥资源在完成其所分配的所有任务之前不会出现空闲状态。

引理4-1. 如果在输入序列 $\sigma = \{r_1, r_2, \dots, r_n\}$ 中, $r_1 = \Delta$, $r_2 = r_3 = 1$ 且 $\Delta \neq 3k/2 (k \geq 2)$ 和 $n \geq 4$, 那么 $C_{\max}(\sigma)/C^*(\sigma) \leq 6/5$ 。

证明. 由条件 $r_1 = \Delta$ 且 $r_2 = r_3 = 1$, 如果序列中至少存在两个大请求, 那么可能存在 $T_w \neq \phi$ 。下面我们按 $T_w = \phi$ 是否存在, 分如下两种情形分析。其中定义 $C_{\max}(\sigma)$ 和 $C^*(\sigma)$ 分别为策略GSRWL₂的分配方案和最优分配方案所对应的最大完工时间。

Case 1. $T_w = \phi$ 。在这种情形下, 说明序列中只有一个大请求或序列中仅有两个大请求且 r_{n-1}, r_n 中有且仅有一个大请求。

Case1.1. 序列中只有一个大请求 $r_1 = \Delta$, 在这种情形下, 当 $C_{\max}(\sigma) \geq \Delta/2$, 那么离线最优 $C^*(\sigma) \geq (C_{\max}(\sigma) - \Delta/3) + \Delta/4 = C_{\max}(\sigma) - \Delta/12 \geq 5C_{\max}(\sigma)/6$; 当 $C_{\max}(\sigma) < \Delta/2$, 容易验证 $C_{\max}(\sigma)/C^*(\sigma) \leq 6/5$ 成立。

Case1.2. 序列中仅有2个大请求且 $r_{n-1} = \Delta$ 或 $r_n = \Delta$ 并分配至 b_2, b_3 泊位, $C_{n,2} \geq \Delta/3$ 。

Case1.2.1. 当 $C_{\max}(\sigma) = C_{n,2} + \Delta/3$, 而离线最优 $C^*(\sigma) > (2C_{n,2} + 3C_{n,1} + \Delta)/5$; 因 $C_{n,1} + \Delta/4 > \Delta/3 + C_{n,2}$, 故 $C^*(\sigma) > (2C_{n,2} + 3(C_{n,2} + \Delta/12) + \Delta)/5 > 5C_{\max}(\sigma)/6$ 。

Case1.2.2. 当 $C_{\max}(\sigma) = C_{n,1} + 1/2 > 2\Delta/3$, 而 $C^*(\sigma) \geq (2C_{n,1} + 3C_{n,2} + 1)/5 = (5C_{n,1} + 1)/5$, 若 $C_{n,1} \geq 13/10$, 从而 $C_{\max}(\sigma)/C^*(\sigma) \leq 6/5$; 如果 $2\Delta/3 - 1/2 < C_{n,1} < 13/10$ 时, 表明在 b_1 泊位上已分配2个小请求, 容易验证定理成立。

Case 2. $T_w \neq \phi$ 。由条件 $T_w \neq \phi$ 知, 在序列中, 至少存在两个大请求。根据定义, T_w 只出现在 b_1 或 b_2 泊位。在 T_w 之前, b_1 泊位只被分配了小请求且加工长度均为 $1/2$, 而 b_2 泊位分配的第1个请求是加工长度为 $\Delta/3$ 的大请求, 其他均为小请求。因此, 若 T_w 在 b_1 或 b_2 泊位, 若 $2 \leq \Delta < 3$, 则其长度分别为 $\Delta/3 - 1/2$ 和 $1 - \Delta/3$; 若 $3 \leq \Delta < 10/3$, 则其长度分别为 $\Delta/3 - 1$ 和 $3/2 - \Delta/3$ 。结合 $2 \leq \Delta < 3$, $\max\{\Delta/3 - 1/2, 1 - \Delta/3\} \leq \Delta/6$; 结合 $3 \leq \Delta < 10/3$, $\max\{\Delta/3 - 1, 3/2 - \Delta/3\} \leq 1/2 \leq \Delta/6$ 。表明在 T_w 期间两个空闲岸桥的总加工能力至多为 $\min\{\Delta/3, 1\}$ 。

Case 2.1. 已知 T_w 期间两个空闲岸桥总加工能力至多为 $\min\{\Delta/3, 1\}$ 。

Case2.1.1. 若 $r_n = \Delta$ 且分配至 b_1, b_2 泊位。

当 $C_{\max}(\sigma) = \max\{C_{n,1}, C_{n,2}\} + \Delta/4$, $C^*(\sigma) \geq (4\max(C_{n,1}, C_{n,2}) + \Delta - 1 + C_{n,3})/5$ 成立; 又因为存在条件 $C_{n,3} + 1 > \max\{C_{n,1}, C_{n,2}\}$, 若 $\max\{C_{n,1}, C_{n,2}\} \geq 12/5 + \Delta/20$, 显然有 $C_{\max}(\sigma)/C^*(\sigma) < 6/5$; 若 $\max\{C_{n,1}, C_{n,2}\} < 12/5 + \Delta/20$, 表明在 b_2 泊位上至多分配1个小请求, 易验证定理成立。

如果当 $C_{\max}(\sigma) = C_{n,3} > \Delta/3 + \Delta/4$ 。表明 $r_{n-1} = 1, r_n = \Delta$ 且有 $3 < \Delta \leq 10/3$ ，否则不会出现此情形。从而 $C^*(\sigma) \geq (4C_{n,1} + \Delta - \min(1, \Delta/3) + C_{n,3})/5 \geq (4(C_{n,3} - 1) + \Delta - 1 + C_{n,3})/5$ ；若有 $C_{n,3} \geq 6 - 6\Delta/5$ ，则有 $C_{\max}(\sigma)/C^*(\sigma) < 6/5$ ；若有不等式 $C_{n,3} < 6 - 6\Delta/5$ ，表明在 b_3 泊位上至多分配2个小请求，易验证定理成立。

Case2.1.2. $r_n = \Delta$ 且分配至 b_2, b_3 泊位。

如果在线完工时间 $C_{\max}(\sigma) = C_{n,2} + \Delta/3$ ，其中 $C_{n,2} \geq \Delta/3 + \Delta/4$ 。在此情形下，有 $C_{n,1} \geq C_{n,2} + 1/2$ ， $C_{n,3} > C_{n,2} - 1/2$ ，不然与题设矛盾。而离线最优完工时间 $C^*(\sigma) \geq (4C_{n,2} - 1 + 1 + \Delta + C_{n,3})/5 \geq (4C_{n,2} + \Delta + C_{n,2} - 1/2)/5 = (5C_{n,2} + \Delta - 1/2)/5$ 。如果当 $C_{n,2} \geq 7\Delta/15 + 3/5$ ，则有 $C_{\max}(\sigma)/C^*(\sigma) \leq 6/5$ ；若 $C_{n,2} < 7\Delta/15 + 3/5$ ，表明在 b_3 泊位上至多分配1个小请求，容易验证定理成立。

如果有 $C_{\max}(\sigma) = C_{n,1} > \Delta/3 + \Delta/4 + \Delta/3$ ，表明大请求的任务量 $2 \leq \Delta < 3$ 。因为有不等式 $C_{n,2} + \Delta/3 > C_{n,1} - 1/2$ 和策略C2.25的设计思想；而离线最优完工时间 $C^*(\sigma) \geq (2C_{n,1} + 2C_{n,2} + \Delta - \Delta/3 + C_{n,3})/5 \geq (2C_{n,1} + 3(C_{n,1} - 1/2 - \Delta/3) + \Delta - \Delta/3)/5$ ；当 $C_{n,1} \geq 9/5 + 2\Delta/5$ ， $C_{\max}(\sigma)/C^*(\sigma) < 6/5$ ；若 $C_{n,1} < 9/5 + 2\Delta/5$ ，表明在 b_1 泊位上至多分配3个小请求，易验证定理成立。

Case2.1.3. 若 $r_n = 1$ 且分配至 b_1 泊位。

如果当 $C_{\max}(\sigma) = C_{n,2} = C_{n,3}$ ，分析过程同上文的Case2.1.2。

如果 $C_{\max}(\sigma) = C_{n,3} > C_{n,2}$ ；在这种情形下，有 $C_{n,1} = C_{n,2} - 1/2$ ， $C_{n,2} > C_{n,3} - 1/2$ 。 $C^*(\sigma) \geq (2C_{n,1} + 1 + 2C_{n,2} - 1 + C_{n,3})/5 \geq (4C_{n,2} + C_{n,3} - 1)/5 = (4(C_{n,3} - 1/2) + C_{n,3} - 1)/5$ 。当 $C_{n,3} \geq 18/5$ ， $C_{\max}(\sigma)/C^*(\sigma) < 6/5$ ；若 $C_{n,3} < 18/5$ ，表明在 b_3 泊位上至多分配3个小请求，易验证定理成立。

当 $C_{\max}(\sigma) = C_{n,1} + 1/2$ 。那么则有 $C_{n,3} > C_{n,1} - 1/2$ ，而 $C^*(\sigma) \geq (4C_{n,1} - 1 + 1 + C_{n,3})/5 \geq (4C_{n,1} + C_{n,1} - 1/2)/5$ ；从而如果当 $C_{n,1} \geq 31/10$ ， $C_{\max}(\sigma)/C^*(\sigma) < 6/5$ ；若 $C_{n,1} < 31/10$ ，表明在 b_1 泊位上至多分配5个小请求，易验证定理成立。

Case2.1.4. 若 $r_n = 1$ 且分配至 b_2 泊位。则必有 $C_{n,1} = C_{n,2} + 1/2$ 。

如果 $C_{\max}(\sigma) = C_{n,3}$ 。说明 $C_{n,3} = C_{n,1} + 1/2 = C_{n,2} + 1$ 。剩下证明过程同Case2.1.3。

如果 $C_{\max}(\sigma) = C_{n,2} + 1/2 = C_{n,1}$ 。剩下证明过程同Case1.1。

Case2.1.5. 若 $r_n = 1$ 且分配至 b_3 泊位。

如果当 $C_{\max}(\sigma) = C_{n,3} + 1$ ，则 $C_{n,1} = C_{n,2} > C_{n,3} + 1/2$ ；而离线最优完工时间 $C^*(\sigma) \geq (4C_{n,2} - \Delta/3 + C_{n,3} + 1)/5 = (4(C_{n,3} + 1/2) - \min(\Delta/3, 1) + C_{n,3} + 1)/5 = C_{n,3} + 2/5$ ；从而如果当 $C_{n,3} \geq 13/5$ ， $C_{\max}(\sigma)/C^*(\sigma) < 6/5$ ；若 $C_{n,3} < 13/5$ ，表明在 b_3 泊位上至多分配2个小请求，易验证定理成立。

如果当 $C_{\max}(\sigma) = C_{n,1} = C_{n,2}$ ，剩下证明过程同Case1.1。

综上，对于任意情形，均有 $C_{\max}(\sigma)/C^*(\sigma) \leq 6/5$ 。引理得证。证毕

通过对引理4-1的证明，可以推出下文定理4-7。

定理4-7. 对于问题 $hybr | online - over - list | BAP - QCAP | C_{\max}$ ，当 $n_q = 5$ ， $2 \leq \Delta < 10/3$ 时策略 $GSRWL_2$ 具有竞争比 $6/5$ 。

证明. 首先定义 $C_{\max}(\sigma)$ 和 $C^*(\sigma)$ 分别为策略 $GSRWL_2$ 的最大完工时间、离线最优调度策略 OPT 对应的目标函数值。在序列 $\sigma = \{r_1, r_2, \dots, r_n\}$ ($n \geq 1$) 中，如果请求总个数 $1 \leq n \leq 3$ ，对小请求和大请求的任意组合，很容易去验证 $C_{\max}(\sigma)/C^*(\sigma) \leq 6/5$ ；在剩下的证明中，我们仅需讨论 $n \geq 4$ 。对于 $r_1 = \Delta$ ($\Delta \neq 3k/2$ ($k \geq 2$)) 且 $r_2 = r_3 = 1$ 情形，根据引理4-1，定理成立。对于 $r_1 = \Delta$ ($\Delta = 3k/2$ ($k \geq 2$)) 且 $r_2 = r_3 = 1$ 或 $\{r_1, r_2, r_3\} \neq \{\Delta, 1, 1\}$ 的情形，从策略的分析可知，在序列 $\sigma = \{r_1, r_2, \dots, r_n\}$ 中，都不会出现 T_w ；余下情形证明分析的过程类似于之前分析的 $r_1 = \Delta$ ($\Delta \neq 3k/2$ ($k \geq 2$)) 且 $r_2 = r_3 = 1$ 情形，通过对引理4-1中 $T_w \neq [0, 0)$ 的讨论可知，此情形也成立，竞争比同样也是 $6/5$ 。定理得证，证毕

由定理4-7可知，当大请求任务量 $2 \leq \Delta < 10/3$ 时，在预知 $k=2$ 请求的特殊情形下，在线策略 $GSRWL_2$ 的竞争比为 $6/5$ ，恰好等于定理4-5中的竞争比下界值。

4.3.3 在线策略 $GLRWL_2$ 设计与竞争分析

针对大请求任务量 $\Delta \geq 10/3$ 的情形，下面给出在线策略 $GLRWL_2$ (Greedy for Large Request With 2-Lookahead) 的具体描述：首先，给出策略 $GLRWL_2$ 的具体描述。

策略 $GLRWL_2$:

若 $r_i = r_{n-2}$ ，由于预知最后2个请求，因而分配该请求和最后2个请求至三个泊位之一使序列完成时间 e_n 最小；否则，若 $1 \leq i < n-2$ ，由于预知后续2个请求 r_{i+1}, r_{i+2} ，分如下3种情形进行调度。

C1. $r_i = \Delta$ 。令启动时间 $s_i = t_{i,2}$ ，把 r_i 分配至左边两个泊位 b_1, b_2 ；若 $C_{i,1} - C_{i,2} = 1/2$ ，则令 $b_T = b_2$ 且 $T_a = [t_1, t_2) = [C_{i,2}, C_{i,1})$ 。

C2. $r_i = 1$ 且 $r_{i+1} = 1$ 。

C2.1. $t_{i,1} = t_{i,3}$ 。令 $s_i = t_{i,1}$ ，把 r_i 分配至 b_3 泊位。

C2.2. $t_{i,1} = t_{i,2} < t_{i,3} = C_{i,3}$ 。令 $s_i = t_{i,1}$ ，把 r_i 分配至 b_1 泊位。

C2.3. $t_{i,1} < t_{i,2} \leq t_{i,3}$ 。令 $s_i = t_{i,1}$ ，把 r_i 分配至 $t_{i,1}$ 所对应的泊位。

C3. $r_i = 1$ 且 $r_{i+1} = \Delta$ 。

C3.1. $t_{i,1} = t_{i,3}$ 。令 $s_i = t_{i,1}$ ，把 r_i 分配至 b_3 泊位。

C3.2. $t_{i,1} = t_{i,2} < t_{i,3}$ 。若 $r_{i+2} = \Delta$ ，令 $s_i = t_{i,3}$ ，把 r_i 分配至 b_3 泊位；若 $r_{i+2} = 1$ ，令 $s_i = t_{i,1}$ ，把 r_i 分配至 b_1 泊位。

C3.3. $t_{i,1} < t_{i,2} \leq t_{i,3}$ 。令 $s_i = t_{i,1}$ ，把 r_i 分配至 $t_{i,1}$ 所对应的泊位。

通过策略设计的描述，空闲时间段是不存在，在请求序列的调度计划中，也不会产生时间段的浪费。

下面将根据对策略的描述与算法思想，给出下面的定理4-8。

定理4-8. 针对问题 $hybr | online - over - list | BAP - QCAP | C_{\max}$ ，当 $n_q = 5$ 且 $\Delta \geq 10/3$ 时策略 $GLRWL_2$ 具有竞争比 $6/5$ 。

证明. 在下面分析过程中，分别考虑最后一个请求按策略中的何种子情形去调度。并定义 $C_{\max}(\sigma)$ 和 $C^*(\sigma)$ 分别为策略 $GLRWL_2$ 的最大完工时间、离线最优调度策略 OPT 对应的目标函数值。

Case 1. r_n 按策略设计中情形C1去调度，表明最后一个请求是大请求。

Case 1.1. 假设最后有 $k (1 \leq k \leq n-1)$ 个大请求被分到左边两个泊位，那么 r_{n-k} 则是小请求。

Case 1.1.1. 若 $C_{\max}(\sigma) = C_{\alpha+1,1} + k\Delta/4$ ，其中 r_α 定义为指泊位 b_1, b_2 上的最后一个小请求。而 $C^*(\sigma) \geq \max\{(4C_{\alpha+1,1} + k\Delta + C_{\alpha+1,3})/5, x\Delta/4\}$ ，其中 x 表示序列中总的大请求的个数；泊位 b_1, b_2 上的最后一个小请求 r_α 到达时，按策略设计的思想，只有按C2.2、C2.3 以及C3.3去调度， r_α 才有可能被分到 b_1, b_2 上，故有不等式 $C_{\alpha,3} \geq C_{\alpha,1} - 1/2$ 成立。下面根据序列中 r_{n-k+1} 请求之前分配的任务展开讨论。

当序列 r_{n-k+1} 请求之前全是小请求，也即所有大请求位于序列 σ 的最后即 $r_{n-k+1} = \dots = r_n = \Delta$ 。当最后1个小请求 $r_{n-k} (= r_p)$ 释放，对 $y = 5w, 5w+1, 5w+2$ 三种情形，分别满足在线策略 $GLRWL_2$ 中的C3.3，C3.1 和C3.2前的 $y, y-1, y-2$ 个小请求所需要的服务总时长等于 $5w/5 < y/5$ ，因而 $C_{\max}(\sigma) \leq x\Delta/4 + y/5$ 。针对 $y = 5w+3, 5w+4$ 的两种情形，分别满足策略中的C3.3和C3.2条件，因而此前的 y 个小请求所需要的服务总时长等于 $5w/5 + 1/2 < y/5$ ，因而 $C_{\max}(\sigma) \leq x\Delta/4 + y/5$ 。若 $x\Delta/4 \geq y$ ，结合 $C^*(\sigma) \geq x\Delta/4$ 可得， $C_{\max}(\sigma)/C^*(\sigma) \leq 6/5$ ；反之，若 $x\Delta/4 \leq y$ ，结合 $C^*(\sigma) \geq (x\Delta + y)/5$ ，有 $C_{\max}(\sigma)/C^*(\sigma) \leq (x\Delta/4 + y/5)/(x\Delta/5 + y/5) \leq (x\Delta/4 + y\Delta/20)/(x\Delta/5 + y\Delta/20) = 6/5$ 。

如果序列中 r_{n-k+1} 请求之前至少有一个大请求，那么则有 $x \geq 1+k$ ，从而得出 $x\Delta/4 \geq (1+k)\Delta/4 \geq 1/2 + k\Delta/4$ 。当 $(4C_{\alpha+1,1} + k\Delta + C_{\alpha+1,3})/5 \geq x\Delta/4$ 时，通过化简可得 $5C_{\alpha+1,1} \geq 5x\Delta/4 + 1/2 - k\Delta$ ，从而有 $C^*(\sigma) \geq (4C_{\alpha+1,1} + k\Delta + C_{\alpha+1,3})/5 \geq (4C_{\alpha+1,1} + k\Delta + C_{\alpha+1,1} - 1/2)/5 \geq 5C_{\max}(\sigma)/6$ 。否则，如果当 $(4C_{\alpha+1,1} + k\Delta + C_{\alpha+1,3})/5 < x\Delta/4$ ，即当 $5C_{\alpha+1,1} < 5x\Delta/4 + 1/2 - k\Delta$ 时，则有 $C^*(\sigma) \geq x\Delta/4 \geq 5C_{\max}(\sigma)/6$ 。

Case 1.1.2. 若 $C_{\max}(\sigma) = C_{n,3}$ ，表明 $k=1$ ，而 $C^*(\sigma) \geq (4C_{n,1} + \Delta + C_{n,3})/5$ 。总有 $C_{n,3} - C_{n,1} \leq \max\{1, 2 - \Delta/4\}$ 。对于 $10/3 \leq \Delta < 4$ 还是 $4 \leq \Delta$ ，都有 $C_{\max}(\sigma)/C^*(\sigma) \leq 6/5$ 。

Case 1.2. 若最后的请求 $r_n = \Delta$ 且分配至 b_2, b_3 泊位。按策略C3情形知， $r_{n-1} = 1$ 。

在此情形下，在线完工时间 $C_{\max}(\sigma) = C_{n,2} + \Delta/3$ 。由此可推出大请求的任务量 $\Delta < 6$ ，且有 $C_{n,1} \geq C_{n,2} + 1/2$ ， $C_{n,3} > C_{n,2} - 1/2$ ，否则与题设条件矛盾。而离线 $C^*(\sigma) \geq (2C_{n,1} + 2C_{n,2} + \Delta + C_{n,3})/5 \geq (4C_{n,2} + 1 + \Delta + C_{n,2} - 1/2)/5 = (5C_{n,2} + \Delta + 1/2)/5$ 。如果当 $C_{n,2} \geq 7\Delta/15 - 3/5$ ，则有 $C_{\max}(\sigma)/C^*(\sigma) < 6/5$ ；反之若 $C_{n,2} < 7\Delta/15 - 3/5$ ，表明在 b_2 泊位上，至多分配4个请求，容易验证定理成立。

Case 2. $r_n = 1$ 且有 $C_{\max}(\sigma) = C_{n,1} = C_{n,2}$ ，那么分析过程同Case1.1.1。

Case 3. $r_n = 1$ 且有 $C_{\max}(\sigma) = C_{n,3}$ ，若 $r_{n-1} = \Delta$ ，那么分析过程同Case1.1.2。若 $r_{n-1} = 1$ ，如果其被分配至空闲时间段 T_a ，分析过程同上；否则，其他情形中离线 $C^*(\sigma) \geq (2C_{n,1} + 2C_{n,2} + 1 + C_{n,3})/5 \geq (4C_{n,1} + C_{n,3})/5 \geq (4(C_{n,3} - 1/2) + C_{n,3})/5 = C_{n,3} - 2/5$ 。当 $C_{n,3} \geq 3 > 12/5$ ， $C_{\max}(\sigma)/C^*(\sigma) \leq 6/5$ ；而当 $C_{n,3} \leq 2 < 12/5$ ，表明泊位 b_3 仅分配2个小请求，容易验证定理成立。

Case 4. $r_n = 1$ 且有 $C_{\max}(\sigma) = C_{n,3} + 1$ ，那么则有 $C_{n,1} + 1/2 > C_{n,3} + 1$ ，否则不成立。而 $C^*(\sigma) \geq (4C_{n,1} + 1 + C_{n,3})/5 \geq (4(C_{n,3} + 1/2) + 1 + C_{n,3})/5 = C_{n,3} + 3/5$ ；当 $C_{n,3} \geq 2 > 7/5$ ， $C_{\max}(\sigma)/C^*(\sigma) \leq 6/5$ ；而 $C_{n,3} \leq 1 < 7/5$ ，表明泊位 b_3 仅分配1个小请求，容易验证定理成立。

Case 5. $r_n = 1$ 且有 $C_{\max}(\sigma) = C_{n,1} + 1/2$ ，根据策略设计，可推出 $C_{n,1} \leq C_{n,2}$ ，而离线最优 $C^*(\sigma) \geq (2C_{n,1} + 2C_{n,2} + 1 + C_{n,3})/5 \geq (4C_{n,1} + 1 + C_{n,1} - 1/2)/5 = C_{n,1} + 1/10$ 。当 $C_{n,1} \geq 19/10$ ， $C_{\max}(\sigma)/C^*(\sigma) \leq 6/5$ ；否则，而当 $C_{n,1} < 19/10$ ，表明 b_1 泊位上至多分配3个请求，容易验证定理成立。

Case 6. $r_n = 1$ 且有 $C_{\max}(\sigma) = C_{n,1}$ ，分析过程同上Case5情形。

综上，对于任意情形，均有 $C_{\max}(\sigma)/C^*(\sigma) \leq 6/5$ 。定理得证。 证毕

由定理4-8可知，当大请求任务量 $\Delta \geq 10/3$ 时，在预知 $k=2$ 请求的特殊情形下，在线策略 GLRWL_2 的竞争比为 $6/5$ ，不大于定理4-5中的竞争比下界值。对于 $n_q = 5$ 的调度模型，可结合上述两种在线策略设计出综合性调度策略：对于 $\Delta \geq 10/3$ 的情形采用 GLRWL_2 策略，而对于 $2 \leq \Delta < 10/3$ 的情形采用 GSRWL_2 策略。由定理4-7和定理4-8可知，该综合性调度策略具有竞争比 $6/5$ 。

4.4 六个岸桥的调度情形

4.4.1 竞争比下界分析

对于预知 $k \geq 2$ 个请求的一般情形，下面证明任意确定性在线策略不可能具有竞争比小于 $9/7$ 。

定理4-9. 当 $n_q = 6$ 时，问题 $\text{hybr} | \text{online-over-list}, LD = k | \text{BAP-QCAP} | C_{\max}$ 的竞争比下界为 $9/7$ 。

证明. 反证法。假设存在一个在线策略 A 具有竞争比 $9/7 - \varepsilon$ ，其中 ε 为任意小的正实数。只需设计一个请求输入序列 σ ，使得 $C_{\max}(\sigma)/C^*(\sigma) > 9/7 - \varepsilon$ ，其中 $C_{\max}(\sigma)$ 和 $C^*(\sigma)$ 分别为策略 A 的分配方案和最优分配方案所对应的最大完工时间。序列 σ 包含至多 $m+1$ 个请求且 $r_1 = r_2 = \dots = r_m = 1$ ，其中 $m = 3w$ (w 为自然数)；令 $\Delta = 2m = 6w$ 。当策略 A 分配请求 r_{m-k} 之后，令 $C_{m-k,i^*} = \max\{C_{m-k,1}, C_{m-k,2}, C_{m-k,3}\}$ 。若 $C_{m-k,i^*} \geq 9w/14$ ，则 $\sigma = \{r_1, r_2, \dots, r_m\}$ 。在最优分配方案中每个泊位完成 w 个请求，因而 $C_{\max}(\sigma)/C^*(\sigma) \geq (9w/14)/(w/2) = 9/7$ ；反之，若 $C_{m-k,i^*} < 9w/14$ ，即 b_{i^*} 泊位当前累积的请求分配数量严格小于 $9w/7$ ，则 $r_{m+1} = \Delta$ 且 $\sigma = \{r_1, r_2, \dots, r_{m+1}\}$ ，此时，讨论如下两种情形。

Case 1. $i^* = 2$ 。即 r_{m-k} 分配之后中间泊位 b_2 的当前完工时间最大，则 b_2 至少已分配了 $(m-k)/3$ 个小请求，表明 $C_{m-k,i^*} \geq (m-k)/6$ 。结合 $\Delta = 2m = 6w$ ， $C_{\max}(\sigma) \geq C_{m-k,i^*} + \Delta/4 \geq 2w - k/6$ 。最优分配方案则是将最后一个大请求分配至两个相邻泊位而将所有小请求分配至另一个泊位， $C^*(\sigma) = \max\{\Delta/4, m/2\} = 3w/2$ 。当 $w \geq 7k/3$ 时， $C_{\max}(\sigma)/C^*(\sigma) \geq (2w - k/6)/(3w/2) \geq 9/7$ 。

Case 2. $i^* = 1$ 或 $i^* = 3$ 。不妨令 $i^* = 1$ 。已知分配完 r_{m-k} 之后 b_1 泊位上当前累积的请求数量严格小于 $9w/7$ ，则 b_2, b_3 泊位中请求分配量较多的泊位至少已分配 $(m-k-9/7w)/2$ 个小请求，有 $C_{\max}(\sigma) > (m-k-9/7w)/4 + \Delta/4 \geq 27w/14 - k/4$ ，最优分配方案同Case1即 $C^*(\sigma) = 3w/2$ 。因此， $C_{\max}(\sigma)/C^*(\sigma) \geq 9/7 - k/(6w)$ 。当 $w > k/(6\varepsilon)$ 时， $C_{\max}(\sigma)/C^*(\sigma) > 9/7 - \varepsilon$ 。

综上，不存在竞争比为 $9/7 - \varepsilon$ 的在线策略。定理得证。

证毕

由定理4-9可知，任意确定性在线策略不可能具有竞争比小于 $9/7$ ；下文根据此数值设计预知后续2个请求的在线策略，并证明其是最优的。

4.4.2 BLA 策略描述

根据定理4-9，任何在线策略在预知 $k \geq 2$ 个请求下的竞争比不可能小于 $9/7$ 。考虑到问题的复杂性，接下来，下文针对 $k = 2$ 即预知后续2个请求的特殊情形提出Balance策略，记为BLA。该策略的主要思想是：当分配每个小请求时，只要当前最大完工时间不超过当前最优值的一定倍数范围，将其优先分配至最左侧 b_1 泊位；将所有大请求分配至右侧两个相邻泊位。在分配某个大请求 r_i 时，有可能造成最右侧泊位 b_3 上出现长度为 $1/2$ 的空闲时段 $[t_1, t_2) = [t_1, t_1 + 1/2)$ ，即 b_3 泊位服务 r_i 的启动时间为 t_2 ，而之前从 t_1 时刻已转入空闲状态。此时，该时段将被标记并将服务于后续释放的首个小请求。BLA策略具体描述如下。

BLA策略：

若 $r_i = r_{n-1}$ ，则 r_{n-1} 释放时即可预知 r_{n-1} 和 r_n 为序列 σ 的最后两个请求，因而分配这两个请求至合适的泊位以使 C_{\max} 最小。如果有多种最佳分配方案，则分配至下标最小的泊位；否则，若 $1 \leq i \leq n-2$ ，由于预知后续请求 r_{i+1}, r_{i+2} ，分如下两种情形进行调度：

C1. $r_i = \Delta$ 。令 $s_i = t_{i,2}$ 。分配 r_i 至 b_2, b_3 泊位。若 $C_{i,2} - C_{i,3} = 1/2$ ，则令

$$[t_1, t_2] = [C_{i,3}, C_{i,2}]。$$

C2. $r_i = 1$ 。若 $[t_1, t_2] \neq [0, 0)$ ，则令 $s_i = t_1$ ，将 r_i 分配至 b_3 泊位，并重置 $t_1 = t_2 = 0$ ；否则，分如下四种情形：

C2.1. $t_{i,1} = t_{i,3}$ 。令 $s_i = t_{i,1}$ ，并分配 r_i 至 b_1 泊位；

C2.2. $t_{i,1} = t_{i,2} < t_{i,3}$ 且 $t_{i,3} + 1/2 \leq 9 \left\{ (4t_{i,1} + 2t_{i,3} + 1 + r_{i+1} + r_{i+2}) / 6 \right\} / 7$ 。令 $s_i = t_{i,3}$ ，并分配 r_i 至 b_1 泊位；

C2.3. $t_{i,1} = t_{i,2} < t_{i,3}$ 且 $t_{i,3} + 1/2 > 9 \left\{ (4t_{i,1} + 2t_{i,3} + 1 + r_{i+1} + r_{i+2}) / 6 \right\} / 7$ 。令 $s_i = t_{i,1}$ ，并分配 r_i 至 b_2 泊位；

C2.4. $t_{i,1} < t_{i,2} \leq t_{i,3}$ 。令 $s_i = t_{i,1}$ ，若 $t_{i,1} = C_{i,1}$ ，分配 r_i 至 b_1 泊位，否则将其分配至 b_3 泊位。

在策略C2.2和C2.3情形中，不等式左边的 $t_{i,3}$ 是 r_i 释放时 b_1 泊位最后一个请求的完工时间，因而 $t_{i,3} + 1/2$ 表示如果 r_i 分配至 b_1 泊位的完工时间；不等式右侧括号内的 $4t_{i,1} + 2t_{i,3} + 1 + r_{i+1} + r_{i+2}$ 表示序列 σ 中前 $i+2$ 个请求的总任务量，而表达式 $(4t_{i,1} + 2t_{i,3} + 1 + r_{i+1} + r_{i+2}) / 6$ 表示离线最优策略服务前 $i+2$ 个请求的最大完工时间的下界。

4.4.3 BLA 竞争性能分析

对于任一请求输入序列 $\sigma = \{r_1, r_2, \dots, r_n\}$ ，令 $C_{\max}(\sigma)$ 和 $C^*(\sigma)$ 分别表示策略 BLA 和离线最优分配方案对应的最大完工时间。首先考虑 $C_{\max}(\sigma)$ 的如下特定情形： $r_n = \Delta$ 且 $C_{\max} = e_n = C_{n,2} + \Delta/4$ 。注意到当 b_2 泊位服务的始终是大请求时，有 $C_{\max}(\sigma) = C^*(\sigma)$ ；因此，下面针对 b_2 泊位至少服务1个小请求时的上述特定情形给出两个基本引理。不妨假设序列 σ 由 $u \geq 0$ 个大请求和 $v \geq 0$ 个小请求组成。

如图4-1所示，令 r_x 为 b_2 泊位上最后分配的小请求， r_Δ 为 b_2 泊位上紧邻 r_x 之后的大请求， r_y 为 r_Δ 之前释放且分配至 b_1 泊位的最后一个请求； r_i ($i = x, y$) 的启动与完工时间分别为 $s_i, e_i (= s_i + 1/2)$ ；令 v 为请求序列 σ 所包含小请求的数量， $v' (\leq v)$ 为 b_2, b_3 泊位所分配的小请求数量与 b_1 泊位截至 e_y 时刻所分配的小请求数量之和；令 w 为分配至 b_2 泊位的小请求数量。

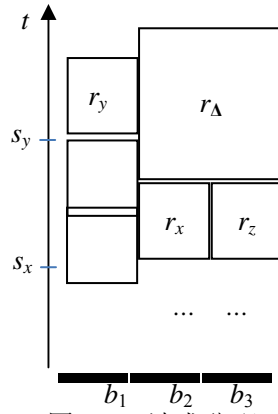


图 4-1: 请求分配示意图

引理4-2. 对于 $u \geq 1, v \geq 3$ 且 b_2, b_3 泊位上共分配偶数个小请求的情形，如果 b_2, b_3 泊位的最后完工请求为大请求且 $C_{\max}(\sigma)$ 等于该大请求的完工时间，则 $C_{\max}(\sigma) \leq u\Delta/4 + v/7$ 。

证明. 考虑任一请求输入序列 $\sigma = \{r_1, r_2, \dots, r_n\}$ ，结合引理条件即 b_2, b_3 泊位上共分配偶数个小请求，不妨令 r_z 为 b_3 泊位上最后分配的小请求，则 r_z 的启动与完工时间相同于 r_x 请求。讨论以下2种情形。

Case 1. r_y 在 r_x 之后释放。由于 r_y 在 r_Δ 之前释放，因而 r_z 必定紧邻在 r_x 与 r_y 之间释放，即紧邻 r_x 之后的两个请求为 r_z 与 r_y 。 r_x 分配至 b_2 泊位满足策略C2.3情形的不等式 $e_y > 9\{(4s_x + 2s_y + 1 + 1 + 1)/6\}/7 \geq 3v'/14$ 。令 $e_y = k/2$ ，其中 $k(\geq 1)$ 为截至 e_y 时刻 b_1 泊位服务的小请求数量。因此， $k/2 > 3v'/14$ 即 $k > 3v'/7$ ；同时，其余的 $v' - k$ 个小请求均分至 b_2, b_3 泊位。因而， $w = (v' - k)/2 < 2v'/7 \leq 2v/7$ 以及 $C_{\max}(\sigma) = u\Delta/4 + w/2 \leq u\Delta/4 + v/7$ 。

Case 2. r_y 紧邻在 r_x 之前释放。根据 r_y 的定义，紧邻 r_x 之后释放的请求 r_{x+1}, r_{x+2} 中至少有一个大请求。由于 r_x 分配至 b_2 泊位，结合策略C2.3情形的条件与 $\Delta \geq 2$ ，有以下不等式成立

$$e_y + 1/2 > 9\{(4s_x + 2e_y + 1 + 1 + \Delta)/6\}/7 \geq 3(4s_x + 2e_y + 4)/14 \quad (1)$$

Case 2.1. 序列 σ 在 r_x 之前释放的均为小请求。根据 w 的定义， $e_x = w/2$ 。 b_1 泊位截至 r_y 所分配的请求数量等于 $v' - 2w$ ，表明 $e_y = (v' - 2w)/2$ 。根据引理条件即 b_2, b_3 泊位上共分配偶数个小请求，因而 b_2, b_3 泊位共分配 $2w$ 个小请求，结合 v' 的定义可得， $v' = 2e_y + 2w$ ，再结合 $s_x = e_x - 1/2 = w/2 - 1/2$ ，由公式 (1) 可得， $v' > 7w/2 - 1/4$ 。由于 v' 与 w 均为正整数，因此上述不等式等价于 $v' \geq 7w/2$ 。因而， $w \leq 2v'/7 \leq 2v/7$ 以及 $C_{\max}(\sigma) = u\Delta/4 + w/2 \leq u\Delta/4 + v/7$ 。

Case 2.2. 序列 σ 在 r_x 之前释放至少1个大请求, 即 b_2, b_3 泊位在 s_x 之前至少服务1个大请求。此时, $s_x \geq (w/2 + \Delta/4) - 1/2 \geq w/2$, 结合 $v' = 2e_y + 2w$ 与公式(1)可得, $w < 2v'/7 \leq 2v/7$, 从而 $C_{\max}(\sigma) \leq u\Delta/4 + v/7$ 。

引理得证。

证毕

引理4-3. 对于 $u \geq 1, v \geq 3$ 且 b_2, b_3 泊位上共分配奇数个小请求的情形, 如果 b_2, b_3 泊位的最后完工请求为大请求且 $C_{\max}(\sigma)$ 等于该大请求的完工时间, 则 $C_{\max}(\sigma) \leq u\Delta/4 + v/7$ 。

证明. 结合引理条件即 b_2, b_3 泊位共分配奇数个小请求和策略C2.4情形条件可知, r_y 紧邻 r_x 之前释放且序列 σ 在 r_x 之后均为大请求。由于 r_x 被分配至 b_2 泊位, 它满足策略C2.3情形的不等式条件, 即

$$e_y + 1/2 > 9\{(4s_x + 2e_y + 1 + 2\Delta)/6\}/7 \geq 3(4s_x + 2e_y + 1)/14 + 6/7 \quad (2)$$

不妨令 $s_x + 1/2 = w/2 + n\Delta/4$, 其中 $n \geq 0$ 表示截至 s_x 时刻 b_2, b_3 泊位所完成的大请求数量, 则 $w \leq 2s_x + 1$; 令 $e_y = k/2$, 其中 $k \geq 1$ 表示为截至 e_y 时刻分配至 b_1 泊位的小请求数量。结合 v' 的定义可得, $v' = k + (2w - 1) \leq 2e_y + (4s_x + 1)$ 。因此, 公式(2)可转化为 $e_y + 1/2 > 3v'/14 + 6/7$, 即 $e_y = k/2 > 3v'/14 + 5/14$ 。结合条件 $v' = k + (2w - 1)$, 有 $k > 3w/2 + 1/2$ 。因为 k, w 均为正整数, 因而上述不等式等价于 $k \geq 3w/2 + 1$, 从而 $v \geq v' = k + 2w - 1 \geq (3w/2 + 1) + 2w - 1 = 7w/2$, 以及 $C_{\max}(\sigma) \leq u\Delta/4 + v/7$ 。引理得证。

证毕

通过对引理4-1和引理4-2的证明, 可得出定理4-10。

定理4-10. 对于问题 $hybr | online - over - list, LD = 2 | BAP - QCAP | C_{\max}$, 当 $n_q = 6$ 时, BLA策略具有竞争比9/7。

证明. 给定任一包含 $u \geq 0$ 个大请求和 $v \geq 0$ 个小请求的输入序列 σ , 考虑如下三种情形。

Case1. $r_n = 1$ 且 $C_{\max}(\sigma) = e_n$ 。不妨设 r_n 被分配至某个泊位 b_B ($B \in \{1, 2, 3\}$), 则 $e_n = C_{n,B} + 1/2$ 。根据策略对 r_n 的分配规则可知, 当 r_n 释放时必有 $C_{n,B} = \min\{C_{n,1}, C_{n,2}, C_{n,3}\}$, 从而 $C^*(\sigma) \geq (2C_{n,1} + 2C_{n,2} + 2C_{n,3} + 1)/6 \geq C_{n,B} + 1/6$ 。因此, 当 $C_{n,B} \geq 1$ 时, $C_{\max}(\sigma)/C^*(\sigma) = (C_{n,B} + 1/2)/(C_{n,B} + 1/6) \leq 9/7$; 当 $C_{n,B} < 1$ 时, 表明 r_n 释放时 b_B 泊位至多已分配1个请求, 此时容易验证 $C_{\max}(\sigma) = C^*(\sigma)$ 。

Case 2. $r_n = 1$ 或 Δ 且 $C_{\max}(\sigma) = C_{n,1}$ 。表明 r_n 被分配至 b_2 或 b_3 泊位, 且 b_1 泊位完工时间仍最迟。

Case 2.1. $C_{n,1} = e_w$ ($w \leq n - 2$)。此时, 小请求 r_w 将依据策略的C2.2情形分配至 b_1 泊位, 由不等式条件 $C_{\max}(\sigma) = e_w \leq 9\{(4t_{w,1} + 2t_{w,3} + 1 + r_{w+1} + r_{w+2})/6\}/7 \leq 9C^*(\sigma)/7$ 。

Case 2.2. $C_{n,1} = e_{n-1}$ 。

(i) 若 $r_n = 1$ 且 r_{n-1} 被分配至 b_2 泊位, 则 r_n 将分配至 b_3 泊位, 后续分析同Case2.1情形。

(ii) 若 $r_n = 1$ 且 r_{n-1} 被分配至 b_1 泊位, 结合策略描述可知, 当 r_{n-1} 释放时 $C_{n-1,2} = C_{n-1,3}$ 且 $C_{n-1,1} \leq C_{n-1,2}$, 进而得出 $C_{n-1,1} = C_{n-1,2} = C_{n-1,3}$; 否则, 如果 $C_{n-1,1} < C_{n-1,2}$, 则 r_n 分配至 b_2 泊位后 $e_n > e_{n-1}$, 这与 $C_{\max}(\sigma) = C_{n,1}$ 相矛盾。因此, $C^*(\sigma) \geq (4C_{n,1} + 2C_{n-1,3})/6 = (4C_{n,1} + 2(C_{n,1} - 1/2))/6 = C_{n,1} - 1/6$ 。当 $C_{n,1} \geq 3/4$, 有 $C_{\max}(\sigma)/C^*(\sigma) \leq 9/7$; 当 $C_{n,1} = 1/2$, 容易得出 $C_{\max}(\sigma) = C^*(\sigma)$ 。

(iii) 若 $r_n = \Delta$, 结合 $C_{n,1} = e_{n-1}$ 和 $C_{\max}(\sigma) = C_{n,1}$, r_{n-1} 必为小请求且被分配至 b_1 泊位, 且 r_n 被分配至 b_2, b_3 泊位; 同时, $C_{n,2} = C_{n,3}$ 且 $C_{n,1} \leq C_{n,2} + \Delta/4 + 1/2$, 否则, r_{n-1} 分配至 b_3 或 b_2 泊位, 与Case2假设的 $C_{\max}(\sigma) = C_{n,1}$ 条件相矛盾。因此, $C^*(\sigma) \geq (2C_{n,1} + 4C_{n,2} + \Delta)/6 \geq C_{n,1} - 1/3$ 。当 $C_{n,1} \geq 3/2$ 时, $C_{\max}(\sigma)/C^*(\sigma) \leq 9/7$; 当 $C_{n,1} \leq 1$ 时, 即 r_{n-1} 释放时 b_1 泊位至多已分配1个小请求, 此时容易得出 $C_{\max}(\sigma) = C^*(\sigma)$ 。

Case 3. $r_n = 1$ 且 $C_{\max}(\sigma) = C_{n,2} = C_{n,3}$, 或者 $r_n = \Delta$ 且 $C_{\max}(\sigma) = e_n = C_{n,2} + \Delta/4$ 。若 $r_n = 1$, 则该请求被分配至 b_1 泊位且 $C_{n,1} + 1/2 < C_{n,2}$ 。根据该情形条件可知, 序列 σ 中的大请求个数 $u \geq 1$ 。当小请求个数 $v \leq 2$ 时, 由策略中的C2.1与C2.2情形可知, 所有小请求必定分配至 b_1 泊位, 从而 $C_{\max}(\sigma) = C^*(\sigma)$ 。当 $v \geq 3$ 时, 由引理1和引理2可得 $C_{\max}(\sigma) \leq u\Delta/4 + v/7$ 。因此, 若 $u\Delta/4 \geq v/2$, 结合 $C^*(\sigma) \geq u\Delta/4$, 有 $C_{\max}(\sigma)/C^*(\sigma) \leq 9/7$; 若 $u\Delta/4 \leq v/2$, 结合 $C^*(\sigma) \geq (u\Delta + v)/6$ 得, $C_{\max}(\sigma)/C^*(\sigma) \leq (u\Delta/4 + v/7)/(u\Delta/6 + v/6) \leq 9/7$ 成立。定理得证。 证毕

结合定理4-9和定理4-10可知, 在线策略BLA具有最优竞争比9/7。这一结论也表明, 有限的预知能力即 $k = 2$ 已经实现了最优的在线调度策略竞争性能, 因此, 更多的预知信息并不能改进在线联合调度的效果。上述结论也为码头调度实践中, 只需要根据当前少数几艘船舶的需求信息局部地制定联合调度方案提供理论上的依据。

4.5 本章小结

本文探讨集装箱码头前沿1个混合型泊位配置4、5、6个岸桥的联合调度在线模型, 针对资源动态调度过程中任意时刻可以预知 $k \geq 2$ 个后续请求的一般情形, 探讨了 $k \geq 2$ 的一般情况, 对于4个岸桥的情形, 当预知未来请求的个数 $k(\geq 3)$ 为奇数时, 竞争比下界为 $(3\sqrt{k^2 + k/2 + 1/2} + 4k + 1)/(4\sqrt{k^2 + k/2 + 1/2} + 3k - 1)$; 当预知未来请求的个数 $k(\geq 3)$ 为偶数时, 竞争比下界 $(3(\sqrt{4k^2 + 8k + 9/4})/2 + 4k + 23/4)/(2\sqrt{4k^2 + 8k + 9/4} + 3k + 3)$ 。对于5、6个岸桥数量的情形, 分别证明了任何确定性在线策略不可能具有竞争比小于25/21和9/7。

当 $k=2$ 预知能力时,对于配置了4、5、6个岸桥的3种情形分别设计了竞争比为 $9/8$ 、 $6/5$ 、 $9/7$ 的联合调度策略。研究结论发现对于4个岸桥的情形,在预知 $k \geq 3$ 请求下的联合调度的下界值随预知能力 k 的增大而越来越小,最终趋近于1;对于5个岸桥的情形, $k=2$ 情形下的竞争比与 $k \geq 3$ 情形下的竞争比下界两者相差 $1/125$,在码头调度中,为了便捷快速的进行调度计划与排班,可以借鉴 $k=2$ 请求下的联合调度的在线策略;对于6个岸桥, $k=2$ 的特定情形下的竞争比与竞争比下界相同,即预知 $k=2$ 请求下的联合调度已经实现了最优的在线调度策略竞争性能。

第5章 结论与展望

5.1 主要研究结论

论文主要讨论了集装箱码头加班船舶的在线调度问题，应用竞争分析方法，具体来说是用竞争比来衡量在线策略的效果；并对这类问题建立相应的在线调度模型，在模型性质剖析和求解算法设计方面，该模型下动态抵港的每艘船舶及其需求信息只有在邻近抵港时才被预见及利用；不同的预知信息量不但对算法的求解效果具有不同的作用，而且对算法的运算复杂性和调度管理成本也有影响。因此，本文的主要工作是结合混合型泊位布局情形下岸桥非交叉移动的服务特征，辨析不同的前瞻能力对上述各方面的作用，并设计最具竞争性的算法。

首先，研究了无预知信息下的集装箱码头泊位和岸桥联合调度在线模型。考虑1个混合型泊位布局且存在大小两种服务请求的情形；研究了请求序列最大完工时间最小化的目标函数；并着重分析了由3个相邻泊位组成的一个混合型泊位调度问题，针对加班船舶请求以Over-list在线形式到达的动态情形展开深入研究；对于配置了4、5、6个岸桥的三种情形，分别设计了具有最优竞争比为 $5/4$ 、 $4/3$ 和 $3/2$ 的在线调度策略。

其次，提出了预知后续 $k=1$ 个请求的集装箱码头泊位与岸桥联合调度在线模型，同样探讨了无预知信息下的泊位与岸桥配置条件。对于配置4个、5个、6个岸桥的三种情形，针对模型中预知信息的特征分别设计给出了具有最优竞争比 $(1+\sqrt{2})/2$ 、 $5/4$ 、 $4/3$ 的在线调度策略；同时，证明了在没有预知时相关模型的竞争比下界分别为 $5/4$ 、 $4/3$ 与 $3/2$ ，该结论也表明即使十分有限的预知能力也可以有效地改进在线调度策略的竞争性能。

再次，进一步探讨了预知后续 $k \geq 2$ 个请求的集装箱码头泊位与岸桥联合调度在线模型。深入研究了集装箱码头前沿1个混合型泊位配置4、5、6个岸桥的联合调度在线模型。首先，对于 $k \geq 2$ 的一般情况，对于4个岸桥的情形，证明了其竞争比下界值随着预知能力 k 的增大而减小，最终趋近于1；对于5和6个岸桥的情形，对于上述2种情形证明了任何确定性在线策略不可能具有竞争比小于 $25/21$ 和 $9/7$ 。其次，针对资源动态调度过程中任意时刻可以预知 $k=2$ 个后续请求的在线调度情形，分别设计了竞争比为 $9/8$ 、 $6/5$ 、 $9/7$ 的联合调度策略。

5.2 未来研究展望

对于本文所研究的问题，后续将进一步从以下几个方面进行研究。

1) 考虑每只船所需要的岸桥数约束的限制或每只船的大小类型较多时, 进一步设计新的有效策略以保证较好的竞争比。由于码头资源的稀缺加上岸桥不可交叉移动的特点, 每只船可供服务的岸桥数一般在一个范围之内, 服务的数量不同, 对码头作业效率的影响也是不同的。本文中, 只分析了两种大小不同的船舶, 而在码头实际调度中, 往往存在多种不同类型和载装卸量不同的船舶, 因而对多种船舶类型的分析, 更能对码头的实际作业提供实践上的理论支持。

2) 考察不完全信息下的动态BAP-QCAP模型, 结合连续型泊位特征, 构建具有预知能力下的在线调度模型, 对预知未来有限时间段内的船舶抵港信息与预知未来有限艘抵港船舶的两种前瞻模式, 辨析不同前瞻能力对求解方案及其效果的影响。对于大型码头来说, 泊位布局往往是连续型泊位特征, 其资源共享更具有一般性, 对其的分析可以探讨与混合型泊位布局的差异。

3) 对于一些中小型码头离散或混合型泊位, 针对泊位所能服务的船舶吨位级别受涨退潮因素影响的情况, 探讨涨退潮时间窗约束下的BAP-QCAP问题, 对相应的模型设计有效的联合调度算法。潮汐对码头的影晌主要表现在船舶的吃水深度, 当潮汐来临时, 泊位水位较深, 方便大型泊位的停靠, 而当潮汐退去时, 部分泊位往往只能服务小型船舶, 因此, 码头如何利用潮汐的影响, 来优化码头资源的调度, 是一个更加值得深入研究的问题。

参考文献

- [1] 杨春霞,王晓军. 集装箱码头前沿生产系统优化调度理论与方法[M]. 北京: 国防工业出版社, 2012.
- [2] Meisel F. Seaside operations planning in container terminals[M]. Berlin, Heidelberg: Physica-Verlag, 2009.
- [3] Bierwirth C, Meisel F, A survey of berth allocation and quay crane scheduling problems in container terminals[J]. European Journal of Operational Research, 2010, 202: 615-627.
- [4] Ben-David S, Borodin A. A new measure for the study of on-line algorithms[J]. Algorithmica, 1994, 11(1): 73-91.
- [5] Kovalyov MY, Ng CT, Cheng TC. Fixed interval scheduling: Models, applications, computational complexity and algorithms[J]. European Journal of Operational Research, 2007, 178(2): 331-342.
- [6] Kolen A WJ, Lenstra JK, Papadimitriou CH, et al. Interval scheduling: A survey[J]. Nava Research Logistics (NRL), 2007, 54(5): 530-543.
- [7] Mes M, Van Der Heijden M, Van Harten A. Comparison of agent-based scheduling to look-ahead heuristics for real-time transportation problems[J]. European Journal of Operational Research, 2007, 181(1): 59-75.
- [8] Mandelbaum M, Shabtay D. Scheduling unit length jobs on parallel machines with lookahead information[J]. Journal of Scheduling, 2011, 14(4): 335-350.
- [9] Sleator DD, Tarjan RE. Amortized efficiency of list update and paging rules[J]. Communications of the ACM, 1985, 28(2): 202-208.
- [10] Imai A, Sun X, Nishimura E, Papadimitriou S, Berth allocation in a container port: using a continuous location space approach[J]. Transportation Research, Part B, 2005, 39(3): 199-221.
- [11] Bierwirth C, Meisel F. A follow-up survey of berth allocation and quay crane scheduling problems in container terminals[J]. European Journal of Operational Research, 2015, 244: 675-689.
- [12] Carlo HJ, Vis I FA, Roodbergen K J. Seaside operations in container terminals: literature overview, trends, and research directions[J]. Flexible Services & Manufacturing Journal, 2015, 27(2): 412-430.
- [13] Xu Y, Chen Q, Quan X, Robust berth scheduling with uncertain vessel delay and handling time[J]. Annals of Operations Research, 2012, 192(1): 123-140.
- [14] 张晋东, 基于鲁棒优化的集装箱码头泊位分配问题研究[D].清华大学硕士学位论文, 2008.

- [15] 魏晓东, 杨智应, 基于桥吊迁移的集装箱码头连续泊位分配算法研究[J]. 计算机应用研究, 2013, 30(10): 2972-2976.
- [16] Guan Y, Yang K-h, Analysis of berth allocation and inspection operations in a container terminal[J]. Maritime Economics & Logistics, 2010, 12(4): 347-369.
- [17] Xu DS, Li CL, Leung JYT, Berth allocation with time-dependent physical limitations on vessels[J]. European Journal of Operational Research, 2012, 216(1): 47-56.
- [18] De Oliveira RM, Mauri GR, Lorena LAN, Clustering search for the berth allocation problem[J]. Expert Systems with Applications, 2012, 39(5): 5499-5505.
- [19] Ting CJ, Wu KC, Chou H, Particle swarm optimization algorithm for the berth allocation problem[J]. Expert Systems with Applications, 2014, 41(4): 1543-1550.
- [20] 刘杰, 集装箱码头泊位动态分配优化研究[D]. 华南理工大学硕士学位论文, 2012.
- [21] Golias M M. A bi-objective berth allocation formulation to account for vessel handling time uncertainty[J]. Maritime Economics & Logistics, 2011, 13(4): 419-441.
- [22] Moorthy R, Teo CP, Berth management in container terminal: the template design problem[J]. OR Spectrum, 2006, 28: 495-518.
- [23] Umang N, Bierlaire M, Vacca I, Exact and heuristic methods to solve the berth allocation problem in bulk ports[J]. Transportation Research Part E, 2013, 54: 14-31.
- [24] Cheong CY, Tan KC, Liu DK, Lin CJ, Multi-objective and prioritized berth allocation in container ports[J]. Annals of Operations Research, 2010, 180: 63-103.
- [25] Park YM, Kim KH, A scheduling method for berth and quay cranes[J]. OR Spectrum, 2003, 25(1): 1-23.
- [26] Theofanis S, Golias M, Boile M, Berth and quay crane scheduling: a formulation reflecting service deadlines and productivity agreements[C]. TRANSTEC2007. Czech Technical University, Prague, 2007, pp.124-140.
- [27] Oguz C, Blazewicz J, Cheng TCE, Machowiak M, Berth allocation as a moldable task scheduling problem[C]. In: Proceedings of Ninth International Workshop on Project Management and Scheduling, pp.201-205, 2014.
- [28] Meisel F, Bierwirth C, Heuristics for the integration of crane productivity in the berth allocation problem[J]. Transportation Research Part E, 2009, 45(1): 196-209.
- [29] Meisel F, Bierwirth C, A framework for integrated berth allocation and crane operations planning in seaport container terminals[J]. Transportation Science, 2013, 47(2): 131-147.
- [30] Hu QM, Hu ZH, Du YQ, Berth and quay-crane allocation problem considering fuel consumption and emissions from vessels[J]. Computers & Industrial Engineering, 2014, 70: 1-10.
- [31] 高超锋, 胡志华, 岸桥并行作业效率约束下泊位与岸桥集成分派[J]. 重庆交通大学学报(自然科学版), 2014, 33(3): 72-78.

- [32] Rodriguez-Molins M, Salido MA, Barber F, A GRASP-based metaheuristic for the berth allocation problem and the quay crane assignment problem by managing vessel cargo holds[J]. *Applied Intelligence*, 2014, 40(2): 273-290.
- [33] 乐美龙, 刘秀玲, 基于泊位偏好与岸桥干扰的泊位与岸桥分配[J]. *运筹与管理*, 2014 23(1): 90-100.
- [34] 桂小娅, 陆志强, 韩笑乐, 集装箱码头连续型泊位与岸桥集成调度[J]. *上海交通大学学报*, 2013, 47(2): 226-229.
- [35] Turkogullari YB, Taskin ZC, Aras N, Optimal berth allocation and time-invariant quay crane assignment in container terminals[J]. *European Journal of Operational Research*, 2014, 235(1): 88-101.
- [36] Yang CX, Wang XJ, Li ZF, An optimization approach for coupling problem of berth allocation and quay crane assignment in container terminal[J]. *Computers & Industrial Engineering*, 2012, 63(1): 243-253.
- [37] 杨华龙, 腾川川, 基于挤压算法的集装箱码头泊位与岸桥联合调度优化[J]. *大连海事大学学报*, 2014, 40(3): 8-12.
- [38] Zhang CR, Zheng L, Zhang ZH, Shi LY, Armstrong AJ, The allocation of berths and quay cranes by using a sub-gradient optimization technique[J]. *Computers & Industrial Engineering*, 2010, 58(1): 40-50.
- [39] 曾庆成, 胡祥培, 杨忠振, 集装箱码头泊位分配-装卸桥调度干扰管理模型[J]. *系统工程理论与实践*, 2010, 30(11): 2026-2035.
- [40] 彭丽姣, 韩晓龙, 连续泊位下泊位分配与岸桥动态调度的联合优化[J]. *水运工程*, 2013, 478(4): 83-88.
- [41] Chang DF, Jiang ZH, Yan W, He JL, Integrating berth allocation and quay crane assignments[J]. *Transportation Research Part E*, 2010, 46(6): 975-990.
- [42] Raa B, Dullaert W, Van Schaeren R, An enriched model for the integrated berth allocation and quay crane assignment problem[J]. *Expert Systems with Applications*, 2011, 38(11): 14136-14147.
- [43] Xiao L, Hu ZH, Berth allocation problem with quay crane assignment for container terminals based on rolling-horizon strategy[J]. *Mathematical Problems in Engineering*, 2014, 19(2): 281-297.
- [44] Giallombardo G, Moccia L, Salani M, Vacca I, Modeling and solving the tactical berth allocation problem[J]. *Transportation Research Part B*, 2010, 44(2): 232-245.
- [45] Blazewicz J, Cheng TCE, Machowiak M, Oguz C, Berth and quay crane allocation: a moldable task scheduling model[J]. *Journal of the Operational Research Society*, 2011, 62: 1189-1197.

- [46] Zhen L, Chew EP, Lee LH, An integrated model for berth template and yard template planning in transshipment hubs[J].Transportation Science, 2011, 45(4): 483-504.
- [47] Chen JH, Lee DH, Cao JX, A combinatorial benders' cuts algorithm for the quayside operation problem at container terminals[J]. Transportation Research Part E, 2012, 48(1):266-275.
- [48] 杨春霞, 王诺, 基于多目标遗传算法的集装箱码头泊位一岸桥分配问题研究[J].计算机应用研究, 2010, 27(5): 1720-1722.
- [49] Imai A, Chen HC, Nishimura E, Papadimitriou S, The simultaneous berth and quay crane allocation problem[J].Transportation Research Part E, 2008, 44(5): 900-920.
- [50] Liang CJ, Guo JQ, Yang Y, Multi-objective hybrid genetic algorithm for quay crane dynamic assignment in berth allocation planning[J]. Journal of Intelligent Manufacturing, 2011, 22(3): 471-479.
- [51] Vacca I, Salani M, Bierlaire M, An exact algorithm for the integrated planning of berth allocation and quay crane assignment[J]. Transportation Science, 2013, 47(2): 148-161.
- [52] 孙彬, 孙俊清, 陈秋双, 基于鲁棒反应式策略的泊位和岸桥联合调度[J]. 系统工程理论与实践, 2013, 33(4): 1076-1083.
- [53] Lokuge P, Alahakoon D, Improving the adaptability in automated vessel scheduling in container ports using intelligent software agents[J]. European Journal of Operational Research, 2007, 177(3): 1985-2015.
- [54] Zhang LL, Khammuang K, Wirth A, On-line scheduling with non-crossing constraints [J]. Operations Research Letters, 2008, 36(5): 579-583.
- [55] 苏兵, 兰小毅. 有限预知信息的可恢复加拿大旅行者问题[J]. 系统工程, 2009, 09: 102-107.
- [56] 郑斐峰, 徐寅峰. 有限预知信息的集装箱搬卸占线问题[J]. 系统工程理论方法应用, 2004, 05: 390-394.
- [57] Pan J, Xu Y. Online Integrated Allocation of Berths and Quay Cranes in Container Terminals with 1-Lookahead[C].COCOON 2015, LNCS 9198, pp. 402-416, 2015.
- [58] Zheng FF, Qiao LL, Liu M, An online model of berth and quay crane integrated allocation in container terminals[C]. COCOA 2015, LNCS 9486, pp. 721-730, 2015.
- [59] Mao W, Kincaid R K. A look-ahead heuristic for scheduling jobs with release dates on a single machine[J].Computers and Operations Research, 1994, 21: 1041-1050.
- [60] Mandelbaum M, Shabtay D. Scheduling unit length jobs on parallel machines with lookahead information[J].Journal of Scheduling, 2011, 14(4): 335-350.
- [61] Wang H, Chaudhary A, Chen D Z. New Algorithms for Online Rectangle Filling with k -Lookahead[J]. Journal of Combinatorial Optimization, 2008, 21(1): 385-394.

附录

1. 策略GRISI程序:

大请求任务量 Δ 取值为 5，随机序列长度为 150 为例。

```

q=0;
for gggg=1:100
seq=zeros(1,150);
r = fix(unifrnd(1,150));
test=5;
for i=r+1:length(seq)
    seq(i)=test;
end
for i=1:r
    seq(i)=1;
end
s=randperm(length(seq));
for i=1:length(seq)
    seq(i)=seq(s(i));
end
B=zeros(1,3);
B(1)=2;
B(2)=2;
B(3)=1;%岸桥数量
WorkTime=zeros(3,150);%
CompletionEndTime=zeros(3,150);
CoptEndTim=zeros(3,150);
%%%%%%%%%%%%%%算法的调度策略
for i=1:length(seq)
%%当前请求是小请求;
if seq(i)==1 && (sum(WorkTime(1,:))-sum(WorkTime(2,:)))==0.5
for j=1:length(WorkTime(2,:))
            if WorkTime(2,j)==0
                WorkTime(2,j)=seq(i)/B(2);
                break;
            end
        end
    end
elseif seq(i)==1 && (sum(WorkTime(1,:))-sum(WorkTime(2,:)))==0&&i~=150&&seq(i+1)==1
for j=1:length(WorkTime(1,:))
            if WorkTime(1,j)==0
                WorkTime(1,j)=seq(i)/B(1);
                break;
            end
        end
    end
elseif seq(i)==1 && (sum(WorkTime(1,:))-sum(WorkTime(2,:)))==0 && i~=150 &&seq(i+1)~=1
for j=1:length(WorkTime(3,:))
            if WorkTime(3,j)==0
                WorkTime(3,j)=seq(i)/B(3);
                break;
            end
        end
    end
end

```

```

        end
    end

    elseif seq(i)==1 && (sum(WorkTime(1,:))-sum(WorkTime(2,:)))~=1/2 && i==150
        && (min(sum(WorkTime(1,:)),sum(WorkTime(2,:)))-sum(WorkTime(3,:))>1/2
    for j=1:length(WorkTime(3,:))
        if WorkTime(3,j)==0
            WorkTime(3,j)=seq(i)/B(3);
            break;
        end
    end

elseif seq(i)==1 && (sum(WorkTime(1,:))-sum(WorkTime(2,:)))~=1/2
&& i==150 && min(sum(WorkTime(1,:)),sum(WorkTime(2,:)))-sum(WorkTime(3,:))<=1/2
    for j=1:length(WorkTime(2,:))
        if WorkTime(2,j)==0
            WorkTime(2,j)=seq(i)/B(2);
            break;
        end
    end

elseif seq(i)~=1
    for j=1:length(WorkTime(2,:))
        if WorkTime(2,j)==0
            WorkTime(2,j)=seq(i)/(B(2)+B(1));
            break;
        end
    end

    for j=1:length(WorkTime(1,:));
        if WorkTime(1,j)==0
            WorkTime(1,j)=seq(i)/(B(2)+B(1));
            break;
        end
    end

end
end
end
end
%在线完工时间;
CompletionEndTime=max([sum(WorkTime(1,:)),sum(WorkTime(2,:)),sum(WorkTime(3,:))])
%所有请求完工时间的下界;
CoptEndTime=sum(seq)/5
sumtmp=0;
for i=1:150;
    if seq(i)==5;
        sumtmp=sumtmp+seq(i);
    end
end
end
%所有大请求完工时间的下界;
BigTime=sumtmp/4
%竞争比的定义;
p=CompletionEndTime/max(BigTime,CoptEndTime)

```

```

q=q+p;
end
%竞争比取平均值;
avep=q/100

```

2. 策略GRTWL程序:

以大请求任务量 Δ 取值为 5，随机序列长度为 50 为例。

```

clc;clear;
q=0;
n=1;
while n<=100
seq=ones(1,50);
r = fix(unifrnd(1,50));
test=5;
for i=r+1:length(seq)
seq(i)=test;
end
for i=1:r
seq(i)=1;
end
s=randperm(length(seq));
for i=1:length(seq)
seq(i)=seq(s(i));
end
B=zeros(1,3);
B(1)=2;
B(2)=2;
B(3)=2;%岸桥的数量
WorkTime=zeros(3,50);
TimeTime=zeros(3,50);
CompletionEndTime=zeros(3,50);
CoptEndTime=zeros(3,50);
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
for i=1:length(seq)
%%%当前是小请求，下一个是小请求;
if seq(i)==1 &&TimeTime(1,i)==TimeTime(2,i) &&TimeTime(2,i)==TimeTime(3,i)
&& i~=50 &&seq(i+1)==1
WorkTime(1,i)=seq(i)/B(1);
TimeTime(3,i+1)=sum(WorkTime(1,:));
TimeTime(2,i+1)=TimeTime(1,i+1)==sum(WorkTime(2,:))=sum(WorkTime(3,:));

elseif seq(i)==1 &&TimeTime(1,i)==sum(WorkTime(1,:))
<TimeTime(2,i)==TimeTime(3,i) && i~=50 &&seq(i+1)==1
WorkTime(1,i)=seq(i)/B(1);
if TimeTime(2,i)-TimeTime(1,i)>=1/2;
TimeTime(1,i+1)=sum(WorkTime(1,:));
TimeTime(2,i+1)=TimeTime(2,i);
TimeTime(3,i+1)=TimeTime(2,i);
else
TimeTime(3,i+1)=sum(WorkTime(1,:));

```



```

        TimeTime(1,i+1)=TimeTime(2,i);
    end

elseif seq(i)==1 && TimeTime(1,i)==sum(WorkTime(3,:))
    <TimeTime(2,i)==TimeTime(3,i) && i~=50 &&seq(i+1)==1
    WorkTime(3,i)=seq(i)/B(3);
    if TimeTime(2,i)-TimeTime(1,i)>=1/2;
        TimeTime(1,i+1)=sum(WorkTime(3,:));
    else TimeTime(3,i+1)=sum(WorkTime(3,:));
    end

elseif seq(i)==1&&TimeTime(1,i)==TimeTime(2,i)<TimeTime(3,i)&&i~=50 &&seq(i+1)==1
    WorkTime(2,i)=seq(i)/B(2);
    if TimeTime(3,i)-TimeTime(2,i)>=1/2;
        TimeTime(2,i+1)=sum(WorkTime(2,:));
    else TimeTime(3,i+1)=sum(WorkTime(2,:));
    end
%%%当前是小请求，下一个是大请求
else if seq(i)==1&&TimeTime(1,i)==TimeTime(2,i)==TimeTime(3,i)&& i~=50&&seq(i+1)~=1
    WorkTime(1,i)=seq(i)/B(1);
    TimeTime(3,i+1)=sum(WorkTime(1,:));
elseif seq(i)==1 && TimeTime(1,i)==TimeTime(2,i)<TimeTime(3,i)
    ==sum(WorkTime(1,:)) && i~=50 &&seq(i+1)~=1
    WorkTime(1,i)=seq(i)/B(1);
    TimeTime(3,i+1)=sum(WorkTime(1,:));
elseif seq(i)==1 && TimeTime(1,i)==TimeTime(2,i)<TimeTime(3,i)~=sum(WorkTime(1,:))
    && i~=50 &&seq(i+1)~=1
    WorkTime(3,i)=seq(i)/B(3);
    TimeTime(3,i+1)=sum(WorkTime(3,:));
elseif seq(i)==1&&sum(WorkTime(1,:))==TimeTime(1,i)
    <TimeTime(2,i)==TimeTime(3,i) && i~=50 &&seq(i+1)~=1
    WorkTime(1,i)=seq(i)/B(1);
    if TimeTime(2,i)-TimeTime(1,i)>=1/2;
        TimeTime(1,i+1)=sum(WorkTime(1,:));
    else TimeTime(3,i+1)=sum(WorkTime(1,:));
    end

elseif seq(i)==1 &&sum(WorkTime(1,:)) ~=TimeTime(1,i)
    &&TimeTime(1,i)<TimeTime(2,i)==TimeTime(3,i) && i~=50 &&seq(i+1)~=1
    WorkTime(3,i)=seq(i)/B(3);
    if TimeTime(2,i)-TimeTime(1,i)>=1/2;
        TimeTime(1,i+1)=sum(WorkTime(3,:));
    else TimeTime(3,i+1)=sum(WorkTime(3,:));
    end
%%%%%%%%%%%%%% 当前是小请求，且是最后一个请求。
elseif seq(i)==1 &&i==50 && sum(WorkTime(1,:))
    ==min([sum(WorkTime(1,:)),sum(WorkTime(2,:)),sum(WorkTime(3,:))])
    WorkTime(1,i)=seq(i)/B(1);
elseif seq(i)==1&&i==50&&sum(WorkTime(2,:))=min([sum(WorkTime(1,:)),

```

```

sum(WorkTime(2,:),sum(WorkTime(3,:)))&&sum(WorkTime(2,:))<sum(WorkTime(1,:))
    WorkTime(2,i)=seq(i)/B(2);
elseif seq(i)==1 &&i==50 && sum(WorkTime(3,:))
    ==min([sum(WorkTime(1,:)),sum(WorkTime(2,:)),sum(WorkTime(3,:))])
    &&sum(WorkTime(3,:))<min([sum(WorkTime(1,:)),sum(WorkTime(2,:))])
        WorkTime(3,i)=seq(i)/B(3);
%%%%%% 当前是大请求，且不是最后一个请求；
elseif seq(i)~=1 && sum(WorkTime(1,:))==TimeTime(2,i)&&i~=50
    WorkTime(2,i)=seq(i)/(B(2)+B(1));
    WorkTime(1,i)=seq(i)/(B(2)+B(1));
    if TimeTime(3,i+1)-TimeTime(2,i)>=seq(i)/(B(2)+B(1));
        TimeTime(2,i+1)=TimeTime(1,i)+sum(WorkTime(2,:));
    else TimeTime(3,i+1)=sum(WorkTime(2,:));
end
elseif seq(i)~=1 && sum(WorkTime(1,:))==TimeTime(2,i)&&i==50
    WorkTime(2,i)=seq(i)/(B(2)+B(1));
    WorkTime(1,i)=seq(i)/(B(2)+B(1));
%%%%%% 当前是大请求，且是最后一个请求；
elseif seq(i)~=1 && sum(WorkTime(1,:))~=TimeTime(2,i)&&i~=50
    WorkTime(2,i)=seq(i)/(B(2)+B(3));
    WorkTime(3,i)=seq(i)/(B(2)+B(3));
    if TimeTime(3,i)-TimeTime(2,i)>=seq(i)/(B(2)+B(3));
        TimeTime(2,i+1)=TimeTime(1,i+1)+sum(WorkTime(2,:));
    else TimeTime(3,i+1)=sum(WorkTime(2,:));
end
elseif seq(i)~=1 && sum(WorkTime(1,:))~=TimeTime(2,i)&&i==50
    WorkTime(2,i)=seq(i)/(B(2)+B(3));
    WorkTime(3,i)=seq(i)/(B(2)+B(3));
end
end
%%%%%%在线完工时间；
CompletionEndTime=max([sum(WorkTime(1,:)),sum(WorkTime(2,:)),sum(WorkTime(3,:))])
%所有请求完工时间的下界；
CoptEndTime=sum(seq)/6
sumtmp=0;
for i=1:50;
    if seq(i)==5;
        sumtmp=sumtmp+seq(i);
    end
end
%所有大请求完工时间的下界；
BigTime=sumtmp/4
%竞争比的定义；
p=CompletionEndTime/max(CoptEndTime,BigTime)
q=q+p;
n=n+1;
end
%竞争比取平均值；
avep=q/100

```

攻读硕士学位期间取得的研究成果

- [1] 郑斐峰, 乔龙亮, 预知信息下集装箱码头资源联合调度在线策略研究[J]. 运筹与管理, 2015,已录用。
- [2] Zheng Feifeng, Qiao Longliang, Liu Ming, An online model of berth and quay crane integrated allocation in container terminals[C]. COCOA 2015, LNCS 9486, pp.721–730, 2015. (EI检索: 2016010174707)
- [3] 乔龙亮, 郑斐峰, 李英, 有限预知信息下集装箱码头泊位分配在线策略研究[C]. 已被“管理科学与工程学会2015年会暨第十三届中国管理科学与工程论坛”会议论文集收录。

致谢

光阴荏苒，岁月如梭。一转眼两年半的硕士学习生涯将要画上句号。本论文是在导师郑斐峰教授的悉心的指导下完成的，从论文的选题、结构的确定、论文内容的撰写到成稿的整个过程中，郑老师时时刻刻给予精心的指导和热情的帮助。郑老师首先给我一个明确的研究方向并鼓励我不断前进，我被他的科研精神和动力打动。这篇毕业论文不仅饱含自己两年多时间来的努力和辛勤的汗水，更是肩负着师友们殷切的期望。虽然仍有很多的不足之处，但欣慰的是我已尽自己最大的努力去完成，是它激励着我在学习上的不断探索。站在人生的又一个转折点上，过去的点点感恩之情油然而生。

从研一起，就一直跟着郑老师开始科研的生活，郑老师给予了我很多研究上的意见和建议，也让我切身体会了踏实和勤奋所发挥的重要作用。清楚的记得，对于一个粗糙的小论文草稿，郑老师都是一字一句细加斟酌的修改，可以说连一个标点符号都不放过，一遍又一遍修改到无法再修改、无懈可击为止，也正是这种耐心的教诲和潜移默化的熏陶，让我对科研产生了浓厚的兴趣和前进的动力。郑老师严谨的治学态度和踏实的工作作风为我树立了榜样，从恩师身上学习到的那种追求科研的无悔精神，是比知识更重要的、作为一位真正的学者所需的素质和品质。而生活中，郑老师也亦师亦友，正是与郑老师的无话不说，才让我少走了很多弯路，也渐渐学会了如何更好地为人处事和接人待物。在此，谨向可亲可敬的郑老师致以最诚挚的敬意！

同时也要感谢管理科学与工程专业的每一位老师给我的帮助；特别要感谢王素芬老师、杨东老师、王长军老师、贾永基老师。感谢703办公室的小伙伴们，感谢管理科学与工程专业的每一位同学，谢谢你们这两年多来给我的帮助与支持，大家一起度过的快乐时光，也将是我们学生时代最美好的记忆！

最后要特别感谢我的父母和亲人，是你们一直以来辛勤的劳作和默默的奉献让我有机会安心的求学，感谢你们一直以来无私的奉献和关怀，感谢你们对我生活上的照顾和学习上的支持！

还有许多帮助和关心过我的人，在此未能一一列举，珍惜我们相遇的缘分，感谢生活中有你们！