

数据驱动的自动化码头 AGV 分布鲁棒优化分配方法

李明泽, 曾庆成, 李兴春

(大连海事大学 航运经济与管理学院, 大连 116026)

摘要 自动引导车 (automated guided vehicle, AGV) 是决定自动化集装箱码头作业效率的关键一环. AGV 经常因拥堵、冲突、天气等因素造成其行驶时间不确定, 从而导致岸桥和场桥等待, 降低码头作业效率. 本文利用历史行驶数据, 引入基于 Wasserstein 距离的分布鲁棒方法, 构建数据驱动模型控制岸桥和场桥的等待风险, 决策作业成本最小的 AGV 任务分配方案. 为了求解模型, 首先使用条件风险价值近似分布鲁棒机会约束, 并通过对偶等方法得到其数学闭式解, 转化为可解形式. 其次, 针对自动化集装箱码头中不同规模问题, 分别设计精确的分支切割算法和启发式算法以及相应的加速策略. 实验结果表明, 分布鲁棒优化方法能够有效刻画行驶时间的不确定性. 与样本平均近似法相比, 本文提出的模型及求解方法可以将岸桥和场桥等待风险降低 60%, 有效提高分配方案的鲁棒性.

关键词 自动化集装箱码头; AGV 任务分配; 不确定行驶时间; 分布鲁棒优化; Wasserstein 距离

Data-driven distributionally robust optimization for AGV scheduling problem in automated container terminals

LI Mingze, ZENG Qingcheng, LI Xingchun

(School of Maritime Economics and Management, Dalian Maritime University, Dalian 116026, China)

Abstract Automated guided vehicles (AGVs) are a key factor in determining the operational efficiency of automated container terminals. AGVs often face travel time uncertainties due to congestion, conflicts, weather, and other factors, resulting in quay and yard crane delays that reduce operational efficiency. In this paper, we introduce a distributionally robust method based on Wasserstein distance using historical travel data. We construct a data-driven model to manage the risk of waiting at quay cranes and yard cranes and to determine the task allocation scheme for AGVs with minimal operating costs. To solve the model, we first use conditional value-at-risk to approximate the distributionally robust chance constraints. Its closed-form solution is obtained by duality theory and analytic transformation, which are then transformed into a practical form. Secondly, exact branch-and-cut algorithms, along with heuristic algorithms and corresponding

收稿日期: 2023-04-28

作者简介: 李明泽 (1995-), 男, 汉, 辽宁朝阳人, 博士研究生, 研究方向: 港口与物流管理, E-mail: limingze_dlmu@163.com; 曾庆成 (1978-), 男, 汉, 山东沂南人, 博士, 教授, 博士生导师, 研究方向: 港口与物流管理, E-mail: qzeng@dlmu.edu.cn; 李兴春 (1984-), 男, 汉, 山东肥城人, 博士研究生, 研究方向: 港口与物流管理, 港口新能源等, E-mail: li_xingchun@126.com.

基金项目: 国家重点研发计划 (2023YFE0113200); 国家自然科学基金青年科学基金 (72203029)

Foundation item: National Key R&D Program of China (2023YFE0113200); National Natural Science Foundation of China (72203029)

中文引用格式: 李明泽, 曾庆成, 李兴春. 数据驱动的自动化码头 AGV 分布鲁棒优化分配方法 [J]. 系统工程理论与实践, 2025, 45(4): 1375-1388.

英文引用格式: Li M Z, Zeng Q C, Li X C. Data-driven distributionally robust optimization for AGV scheduling problem in automated container terminals[J]. Systems Engineering — Theory & Practice, 2025, 45(4): 1375-1388.

acceleration strategies, are designed for problems of varying sizes in automated container terminals. Experimental results demonstrate that the distributionally robust optimization method effectively captures the uncertainties in AGV travel times. Compared to the sample average approximation, the model and solution proposed in this paper can reduce the risk of waiting at quay cranes and yard cranes by 60%, significantly enhancing the robustness of the allocation scheme.

Keywords automated container terminals; AGV scheduling; uncertainty travel time; distributionally robust optimization; Wasserstein distance

1 引言

相对于传统码头,自动化集装箱码头具有作业效率高、用工少和可靠性高等优势,对保障港口装卸效率以及全球供应链安全做出了巨大贡献。码头作业效率直接影响着船舶在港停留时间,港口运营商需要提高装卸效率,降低运营成本,巩固并提升区域集装箱枢纽港的地位。码头前方作业区主要由泊位作业系统、堆场作业系统以及链接二者的水平运输作业系统构成。自动引导车(automated guided vehicle, AGV)¹是水平运输系统的典型作业设备,负责在岸桥(quay crane, QC)与场桥(yard crane, YC)的缓存区之间运输集装箱,是制约整体作业效率的关键之一。为了最大化装卸作业效率,AGV需要准时甚至提前到达缓存区,避免QC和YC等待。但是受到可用数量、路网容量和作业成本的限制,码头无法无限投入AGV以保证作业效率。因此,如何合理调度AGV,提高整体作业效率,是码头面临的重要问题之一,已经成为当前研究热点。

AGV调度优化通常包含任务分配和路径规划两个阶段^[1]。在任务分配时,调度部门根据整体作业计划制定分配方案,合理分配集装箱,生成多条作业序列^[2]。而路径规划阶段则重点解决运行时可能出现的冲突、拥堵和死锁等情况,为多个AGV同时计算路网中的具体行驶轨迹^[3]。在码头实际决策时,两个阶段紧密联系、互相影响:一方面,任务分配结果决定了路径规划的起点与终点;另一方面,路径规划方案影响AGV实际行驶时间,是任务分配的前提。但是,AGV行驶过程受到多种不确定因素的影响,如实时情况、设备故障、天气、交通拥堵等,使得行驶时间难以准确预测。另外,随着人工智能等技术的进步,水平运输设备以及路径规划方法也在不断更新^[4,5],进一步影响着实际行驶状态。若任务分配时采用的行驶时间与实际存在显著差异,可能无法使调度方案达到最优,甚至引发连锁反应,导致码头调度系统停滞或崩溃^[6]。为了解决这一问题,码头调度部门需要在任务分配阶段考虑AGV实际行驶状态和行驶时间的不确定性,使调度方案保持一定冗余,不仅更加贴合码头实际情况,而且可以提高整体作业协调性^[7]。

然而,现有大部分关于任务分配的研究不考虑AGV行驶的不确定性,只是基于静态行驶时间直接求解调度模型。另外虽然已经有少数研究关注不确定因素对任务分配的影响,但由于码头作业环境复杂,理论模型^[1,7-10]在预测精度上依旧存在一定局限,参见文献综述。因此,本文基于分布鲁棒优化(distributionally robust optimization, DRO)建立了一个数据驱动的AGV任务分配模型,利用历史数据刻画不确定的行驶时间,避免了理论模型的过拟合问题,旨在最小化作业成本并提高码头整体调度的协同性。本文的主要贡献包括:1)基于码头作业方案的决策过程,建立水平运输系统的调度逻辑框架,提高了任务分配方法对不同设备的适用性;2)基于Wasserstein距离生成行驶时间的模糊集,使用分布鲁棒机会约束控制不确定行驶时间对QC和YC的影响,加强设备之间的协同性;3)引入条件风险价值(conditional value at risk, CVaR)近似分布鲁棒机会约束,通过数学方法获得CVaR的闭式解,并提出了针对小规模问题的精确算法、大规模问题的启发式算法以及几种加速策略,提升模型求解质量;4)通过大量实验验证了本文方法,证明与样本平均近似法(sample average approximation, SAA)^[11]相比,DRO方法将QC和YC的等待风险降低60%以上。

¹自动化集装箱码头中存在多种水平运输设备,功能略有差异,但分配原则相同,本文以AGV为代表进行说明。

2 文献综述

本文旨在 AGV 任务分配阶段考虑不确定的行驶时间, 提高分配方案的鲁棒性. 针对任务分配问题, 国内外学者已经进行了大量研究, 为本文打下了基础. 现有文献常常考虑充电需求^[12,13]、双循环策略^[14,15]和多载功能^[16,17]等实际情况, 以最小化作业时间^[12-15,17]或作业成本^[16]为目标, 建立数学规划模型^[12-17]并设计启发式算法求解^[12-17]. 例如, 李林蔓^[13]提出了一种启发式规则的换电式 AGV 分配方法, 并设计蚁群算法求解. Zhang 等^[14,15]提出了一种装卸同步双循环的调度模型, 分别设计带有惩罚函数的混合粒子群算法^[14]和遗传算法^[15]生成 AGV 作业顺序. 如何在任务分配问题中考虑行驶时间的不确定性是本文研究的重点, 已经有小部分文献关注这一问题, 使用仿真系统^[1]、模拟实验^[7]、模糊理论^[8,9]或拥堵公式^[10]刻画不确定的行驶时间, 描述实际运行状态对分配过程的影响. 例如, Yang^[1]建立仿真模型, 基于动态时间估计 AGV 调度算法以减少路网拥堵. 徐卉^[8]和 Jian^[9]采用模糊函数刻画 AGV 行驶时间, 并分别结合 QC 调度^[8]和堆场堆存位置问题^[9]联合调度, 提高作业协同性. 可以发现, 大部分已有任务分配的研究主要关注作业效率和成本, 并未考虑 AGV 行驶状态和行驶时间的不确定性^[12-17]. 虽然有小部分文献基于理论模型刻画了某一类运行状态对行驶时间的影响, 但难以包含所有运行情况, 使预测的行驶时间存在偏差^[6]. 这种偏差令分配方案无法达到最优, 甚至影响码头整体作业效率.

同时, 通过梳理现有文献发现, 还有两类方法可以提高任务分配方案的鲁棒性, 达到降低或捕捉行驶不确定性的作用: 第一类文献综合任务分配和路径规划问题^[18,19], 指派 AGV 的同时计算无冲突路径. 如 Yue 和 Fan^[19]以及 Wang 和 Zeng^[18]均构建了混合整数规划模型同时求解任务分配和路径规划问题, 分别考虑单向路径^[19]或双向路径^[18]建立两阶段模型^[19]或整体模型^[18], 设计动态启发式算法^[19]或分支定界算法^[18]求解. 这类文献在指派 AGV 的同时计算行驶路径, 降低实际行驶过程的不确定性. 但是依然依赖理论模型刻画行驶状态, 同时受限于具体码头, 对新的路径规划算法和场景适应性较低. 第二类文献采用在线调度思想, 通过实时捕捉 AGV 运行数据, 可以避免行驶不确定性的影响. 这类文献通常使用深度学习等方法动态完成 AGV 任务指派, 如 Gong 等^[20]和 Chen 等^[21]基于深度强化学习方法, 用马尔科夫决策过程刻画 AGV 分配过程, 并动态调整调度策略. 贾石岩等^[22]则采用偏好函数和深度信念网络实时完成任务指派. 这类方法通常具有较强的实时性, 可以及时对不确定性做出反馈. 但由于数学解释性不足而且较为短视, 可能无法获得全局最优解. 同时在自动化集装箱码头中, AGV 的作业任务可以提前确定, 实时调度的优势并不明显.

另一方面, 自动化集装箱码头中路径规划方法也在不断发展. 现有研究通常应用调度规则^[23,24]和车辆行为模式^[19,25]等方式规划多 AGV 行驶路径, 协调互相之间造成的冲突、死锁以及拥堵等情况. 例如, 张素云等^[25]提出一个多参数优化控制模型, 通过控制速度解决冲突问题. Li 等^[26]合理设置交通规则, 避免车辆运行时相撞、死锁和故障. 随着人工智能技术的发展, 强化学习^[27]和深度学习^[28]等前沿方法也被引入解决路径规划问题, 进一步增强了多 AGV 行驶方案的科学性. 例如, Zhou 等^[27]根据实时车辆状态, 提出一种 Q 学习方法寻找 AGV 最短时间的行驶路径, Hu 等^[28]设计了多智能体深度确定性策略梯度方法解决调度冲突情况. 由此可见, 深度学习等底层算法的发展推动路径规划方法不断创新, 为多 AGV 协同控制提供了更多思路. 同时随着自动驾驶技术的发展, 已有 AGV 搭配自动驾驶系统^[4], 进一步提高了行驶的灵活性, 也对任务分配方法提出了新的挑战.

为了应对新的发展趋势, 本文通过 DRO 方法刻画不确定的行驶时间, 旨在提高 AGV 任务分配方案的鲁棒性和作业协同性. DRO 方法基于统计学习的思想构建不确定性参数所在模糊集, 兼具数学模型的可解释性和统计方法的泛化性, 已经被广泛应用于电力系统^[29,30]、交通系统^[31,32]和库存管理^[33,34]等领域中. 这些研究通过 DRO 方法刻画发电能力^[29,30]、社会需求^[31,32]和消费需求^[33,34]等不确定性参数, 被证明能够在经济性和保守性之间实现均衡决策^[35]. 因此, 本文引入 DRO 方法刻画 AGV 行驶时间的不确定性, 可以有效弥补现有研究不足. 首先, 通过历史数据总结 AGV 的行驶规律, 能够有效避免理论模型的过拟合问题, 提高模型的泛化性. 其次, 历史数据能够作为纽带解耦任务分配和路径规划阶段, 使本文方法能够适用于不同的水平运输设备和路径规划算法, 增强方法的适用性. 最后, 本文设计

分支切割算法求解小规模问题得到精确解,并设计能够求解大规模问题的变邻域搜索-模拟退火混合启发式算法,保证求解的质量与速度.

3 问题描述与模型构建

3.1 问题描述

船舶靠泊前,码头管理部门首先根据实际情况制定集装箱配积载计划,从而生成码头整体作业计划.集装箱依次通过泊位系统、水平运输系统和堆场系统,并根据作业计划在船舶和堆场之间流转.码头前沿典型布局及作业流程如图1所示.对于卸船任务,QC将进口箱从船舶积载位置运送至缓存区并放置在AGV上,随后由AGV运送至相应堆场的缓存区,并由YC拾取后放置到指定堆存位置;对于装船任务,集装箱作业过程相反.QC和YC结构复杂,单箱作业时间较长,相比之下AGV更加灵活,这导致三者之间作业特征差异较大.同时大部分AGV不具有举升功能,需要与QC和YC精确对接.理想状态下AGV需要提前到达缓存区等待作业,避免设备之间相互等待.因此,水平运输系统的协同功能对保证整体效率和稳定运行至关重要.码头需要根据码头作业计划计算集装箱到达各缓存区的位置以及时间,生成最优分配方案.

另一方面,水平运输系统中多AGV在封闭路网中同时行驶会出现等待、拥堵、冲突和死锁等情况,甚至可能与外部集卡混行^[36].在计划阶段,码头管理部门难以精确刻画AGV实际运行中可能发生的各类情况,只能根据预测的行驶时间制定分配方案.而历史行驶数据可以综合反映AGV的运行情况,可以进一步作为纽带使任务分配方法适用于不同设备和路径规划算法.因此,本文在任务分配阶段基于数据驱动方法考虑不确定的行驶时间,能够提高任务分配方案的鲁棒性和适用性,更加符合码头实际作业要求.

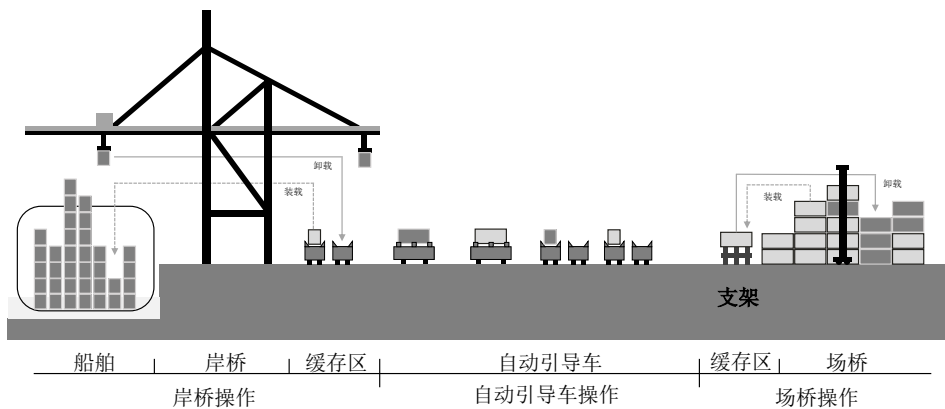


图1 自动化集装箱码头前沿操作流程

3.2 确定性模型

首先构建作业网络图, V 为图上节点表示集装箱作业任务, $V = (\{o\}, C, \{o'\})$, o 表示虚拟起始点、 o' 表示虚拟终点、 C 表示作业任务集合. A 表示图上作业任务节点之间的弧, $|A|$ 为总数, 共有三类: $A_o = \{(i, j) \mid i = o, j \in C \cup \{o'\}\}$ 表示由虚拟开始节点 o 到作业任务、 $A_c = \{(i, j) \mid i, j \in C, i \neq j\}$ 表示任务之间调度、 $A_{o'} = \{(i, j) \mid i \in C \cup \{o\}, j = o'\}$ 表示作业任务到虚拟结束节点 o' , 则存在 $A = A_o \cup A_c \cup A_{o'}$. 定义从节点 $i \in V$ 出发向外的弧的集合为 $\delta^+(i) = \{(i, j) \in A \mid i \in V, j \in V \setminus \{i\}\}$, 同理进入节点 $i \in V$ 的弧的集合为 $\delta^-(i) = \{(j, i) \in A \mid i \in V, j \in V \setminus \{i\}\}$, 附录A给出一个作业网络示意图.

d 为行驶距离参数, d_i 表示执行作业任务 $i \in C$ 的行驶距离, d_{ij} 表示作业任务 $i, j \in C$ 之间的距离. c 为成本参数, c_m 表示单次运行的固定成本, c_d 表示单位距离行驶成本. 假设行驶费用与行驶距离成正比, 图中 $(i, j) \in A$ 的运行成本 c_{ij} 为: 1) 从 o 出发的弧 $(i, j) \in A$, $c_{ij} = c_m + c_d d_{ij}$; 2) 其他弧

$(i, j) \in A_c \cup A_{o'}$, $c_{ij} = c_d d_{ij}$. s_i 表示作业任务 $i \in C$ 的行驶时间, t_{ij} 表示弧 $(i, j) \in A$ 的行驶时间. 决策变量为 x_{ij}^k 、 T_i^k , x_{ij}^k 表示若第 k 个 AGV 从任务 i 结束节点出发后到达任务 j 的开始节点, 则为 1, 否则为 0; T_i^k 表示第 k 个 AGV 到达任务 $i \in C$ 的时刻, 建立任务分配模型.

$$\min \sum_{k \in K} \sum_{(i,j) \in A} c_{ij} x_{ij}^k \tag{1}$$

$$\text{s.t.} \quad \sum_{k \in K} \sum_{(j,i) \in \delta^-(i)} x_{ji}^k = 1, \quad \forall i \in C, \tag{2}$$

$$\sum_{k \in K} \sum_{(i,j) \in \delta^+(i)} x_{ij}^k = 1, \quad \forall i \in C, \tag{3}$$

$$\sum_{(i,j) \in \delta^-(i)} x_{ij}^k = \sum_{(j,i) \in \delta^+(i)} x_{ji}^k, \quad \forall k \in K, \forall i \in C, \tag{4}$$

$$\sum_{(o,i) \in A} x_{oi}^k = \sum_{(i,o') \in A} x_{io'}^k = 1, \quad \forall k \in K, \forall i \in C, \tag{5}$$

$$T_i^k + s_i + t_{ij} - T_j^k \leq M(1 - x_{ij}^k), \quad \forall k \in K, \forall (i, j) \in A, \tag{6}$$

$$x_{ij}^k \in \{0, 1\}, \quad \forall k \in K, \forall (i, j) \in A, \tag{7}$$

其中, 式 (1) 表示最小化作业总成本; 式 (2)、(3) 保证作业任务只能被执行一次; 式 (4) 表示进出作业任务节点的弧流量守恒; 式 (5) 表示任务序列从起点开始至终点结束; 式 (6) 任务之间的时间关系, 其中 M 是一个较大的数; 式 (7) 表示 x_{ij}^k 为 0-1 变量.

3.3 数据驱动模型

3.3.1 等待时间函数

为考虑行驶时间不确定, \tilde{s}_i 表示作业任务 $i \in C$ 不确定的行驶时间, \tilde{t}_{ij} 表示弧上 $(i, j) \in A$ 不确定的行驶时间. 根据附录 B 通过码头作业计划确定作业时间窗 $[a_i, b_i]$, 并遵循以下规定: 如果早于 a_i 到达必须等待, 直到 a_i 才能开始作业; 晚于 b_i 到达会造成 QC 和 YC 等待, 可以继续执行, 但被认为质量较低. ξ_i 表示 QC 和 YC 的等待时间, 可知 $\xi_i = T_i - b_i$. 基于某一可行解 \mathbf{x} (x_{ij}^k 的矢量形式), 令 $C_j(\mathbf{x}) = \{o, 1, \dots, j-1, j\}$ 表示 \mathbf{x} 下从起点 o 至点 j 的作业序列上依次经过的任务集合, $A_{ij}(\mathbf{x}) = \{(0, 1), \dots, (j-1, j)\}$ 表示任务 i 到任务 j 之间的任务. 定理 1 表示任务 $j \in C$ 中 QC 或 YC 等待时间函数 ξ_j .

定理 1 假设 \mathbf{x} 、行驶时间已知, 任务 $j \in C$ 的 QC 或 YC 等待时间函数如下:

$$\xi_j(\mathbf{x}) = \max_{i \in C_j(\mathbf{x})} \left\{ a_i + \sum_{(i,k) \in A_{ij}(\mathbf{x})} (s_i + t_{ik}) \right\} - b_j. \tag{8}$$

证明: 详见附录 C.

3.3.2 分布鲁棒机会约束

根据定理 1 得到 ξ_i 也具有不确定性, 用 $\tilde{\xi}_i$ 表示. 为了刻画 $\tilde{\xi}_i$, 本文引入 DRO 方法, 基于 Wasserstein 距离的模糊集 $\mathcal{F}(\theta)$ ^[37], 通过样本数据刻画行驶时间的真实分布. 假设 $\mathcal{F}(\theta)$ 已经包含了接近经验分布的所有离散和连续概率分布, 并针对其中“最差”分布 \mathbb{P} 进行优化. 同时结合 Laporte 等^[38] 提出的机会约束得到数据驱动的分布鲁棒机会约束 (10) 控制 QC 和 YC 的延误风险, 其中 $1 - \varepsilon$ 表示目标服务水平 (level of service, LoS), $\tilde{\xi}$ 与情景 ω 相关. 为了简化说明, 除去上标 k , 得到数据驱动的分布鲁棒机会约束模型如下.

$$\min \sum_{(i,j) \in A} c_{ij} x_{ij} \tag{9}$$

$$\text{s.t. } \inf_{\mathbb{P} \in \mathcal{F}(\theta)} \mathbb{P}(\tilde{\xi}_i(\mathbf{x}, \omega) \leq 0) \geq 1 - \varepsilon, \forall \omega \in \Omega, i \in C, \quad (10)$$

$$\mathbf{x} \in \mathcal{X}, \quad (11)$$

其中, 式 (10) 为分布鲁棒机会约束, 采用 1 型 Wasserstein 距离模糊集; 式 (11) 表示可行解 \mathbf{x} 属于可行解集 \mathcal{X} , 具体如 (12), 第 4 项表示最大车辆数限制, 第 5 项表示去除子回路约束, A_S 表示集合 S 中任务 $i, j \in S$ 之间的弧.

$$\mathcal{X} = \left\{ \mathbf{x} \in \{0, 1\}^{|A|} \left| \begin{array}{l} \sum_{(i,j) \in \delta^+(i)} x_{ij} = 1, \forall i \in C, \\ \sum_{(i,j) \in \delta^-(i)} x_{ij} = 1, \forall i \in C, \\ \sum_{(i,j) \in \delta^+(i)} x_{ij} = \sum_{(i,j) \in \delta^-(i)} x_{ij}, \forall i \in C, \\ \sum_{(i,j) \in \delta^+(o')} x_{ij} \leq K, \\ \sum_{(i,j) \in A_S} x_{ij} \leq |S| - 1, \forall S \subset C, |S| \geq 2 \end{array} \right. \right\}. \quad (12)$$

4 模型转化与算法设计

4.1 模型转化

分布鲁棒机会约束的存在使模型无法直接求解, 本文引入 CVaR 对其近似. CVaR 被 Ben 和 Teboulle^[39] 提出可以作为等价风险测度后, 被认为是机会约束问题的“最佳”凸保守近似^[40]. 同时, CVaR 的约束优化问题可以表示基于 Wasserstein 距离模糊集的分布鲁棒机会约束^[41]. 因此, 本文分布鲁棒机会约束转化为式 (13), 其中 β 为 CVaR 的辅助决策变量.

$$\sup_{\mathbb{P} \in \mathcal{F}(\theta)} \inf_{\beta \in \mathbb{R}} \left(\beta + \frac{1}{\varepsilon} \mathbb{E}_{\mathbb{P}}[(\tilde{\xi}_i(\mathbf{x}, \omega) - \beta)^+] \right) \leq 0. \quad (13)$$

引入辅助决策变量 α_i 后对其线性化, 可以视为一个优化问题.

$$\text{CVaR}_{\varepsilon}^{\theta}[\tilde{\xi}_i(\mathbf{x}, \omega)] = \min \alpha_i \quad (14)$$

$$\text{s.t. } \beta + \frac{1}{\varepsilon} \sup_{\mathbb{P} \in \mathcal{F}(\theta)} \mathbb{E}_{\mathbb{P}}[(\tilde{\xi}_i(\mathbf{x}, \omega) - \beta)^+] \leq \alpha_i, \forall \mathbf{x} \in \mathcal{X}, \omega \in \Omega, i \in C. \quad (15)$$

该优化问题为一个难以解决的半无限规划问题. 为求解模型, 令 $\Omega = \{1, 2, \dots, |\Omega|\}$ 表示所有情景集合, 可转化如下, 证明见附录 D.

$$\text{CVaR}_{\varepsilon}^{\theta}[\tilde{\xi}_i(\mathbf{x}, \omega)] = \min \alpha_i \quad (16)$$

$$\text{s.t. } \beta + \frac{\theta}{\varepsilon} + \frac{1}{\varepsilon|\Omega|} \sum_{\omega \in \Omega} (\tilde{\xi}_i(\mathbf{x}, \omega) - \beta)^+ \leq \alpha_i, \forall \mathbf{x} \in \mathcal{X}, \omega \in \Omega, i \in C. \quad (17)$$

虽然该模型已经能够求解. 但为进一步加快求解速度, 提出定理 2 得到 CVaR 的闭式解.

定理 2 将 $\omega \in \Omega$ 情景下任务 $i \in C$ 的等待时间 $\hat{\xi}_i(\mathbf{x}, \omega)$ 从大到小排序后, 计算如下:

$$\text{CVaR}_{\varepsilon}^{\theta}[\tilde{\xi}_i(\mathbf{x}, \omega)] = \sum_{\omega=1}^{\lfloor \varepsilon|\Omega| \rfloor} \frac{\hat{\xi}_i(\mathbf{x}, \omega)}{\varepsilon|\Omega|} + \left(1 - \frac{\lfloor \varepsilon|\Omega| \rfloor}{\varepsilon|\Omega|} \right) \hat{\xi}_i(\mathbf{x}, \lfloor \varepsilon|\Omega| \rfloor + 1) + \frac{\theta}{\varepsilon}. \quad (18)$$

证明: 详见附录 E.

4.2 分支切割算法

分支切割算法属于精确算法, 能够将机会约束转化为割平面加入模型中, 流程详见附录 F. 下面对关键步骤进行说明.

4.2.1 割平面

基于 CVaR 的闭式解, 设计割平面 (19). 如果 CVaR 大于 0, 表示当前解下该任务不可行, 将式 (19) 加入松弛后的模型中.

$$\sum_{(k,l) \in A_{oj}(\mathbf{x})} (x_{kl} - 1) + 1 \leq 0. \quad (19)$$

同时, 求解过程中可行解集 \mathcal{X} 中子回路约束也通过割平面的方式进行添加.

4.2.2 加速策略

弧消除: 对网络图中的弧进行预处理, 可加速算法求解, 规则如下:

$$\text{If: } \text{CVaR}_\varepsilon^\theta(\max\{a_i + \tilde{s}_{ij} + \tilde{t}_{ij}, a_j\} - b_j) > 0, \forall j \in C, (i, j) \in A,$$

$$\text{Then: } x_{ij} = 0. \quad (20)$$

$$\text{If: } \text{CVaR}_\varepsilon^\theta(\max\{\max\{a_i + \tilde{s}_{ik} + \tilde{t}_{ik}, a_k\} + \tilde{s}_{kj} + \tilde{t}_{kj}, a_j\} - b_j) > 0, \forall j \in C, (i, k), (k, j) \in A,$$

$$\text{Then: } x_{ik} + x_{kj} \leq 1. \quad (21)$$

储存结构: 计算过程中, 缓存中间结果并与剪枝策略配合以避免重复计算.

热启动: 良好的初始解有利于加快分支切割算法的求解, 算法流程详见附录 G.

剪枝策略: 当求解 CVaR 时, 一旦存在 CVaR 大于 0, 可以提前终止计算, 避免重复计算.

4.3 启发式算法

大规模问题基于变邻域搜索算法 (variable neighborhood search, VNS) 和模拟退火算法机制设计了一种混合式启发式算法. 算法流程详见附录 H, 下面说明关键步骤.

4.3.1 适应度函数

部分算子允许生成违反机会约束的解, 根据式 (22) 计算解的适应度, 其中 η 表示惩罚系数.

$$\sum_{(i,j) \in A} c_{ij}x_{ij} + \eta \text{CVaR}_\varepsilon^\theta[\tilde{\xi}_i(\mathbf{x}, \omega)]. \quad (22)$$

4.3.2 搜索算子

根据研究问题的特性设计本文算子, 各算子均在不同任务序列之间搜索, 具体如下:

- 1) 随机交换两条任务序列之间连续 h 个任务;
- 2) 随机选择某一任务, 将其插入其他任务序列中;
- 3) 随机选取两个任务序列, 在每个序列中各选取一个任务进行交换;
- 4) 随机选取两条任务序列, 取一个交换位置, 在该位置后交换两路径剩余任务;
- 5) 随机选取一部分任务, 将其从任务序列中删除后, 重新插入.

4.3.3 接受准则

采用模拟退火算法中的 Metropolis 准则式 (23) 对当前解的质量进行判断, 若新解适应度小于当前最优解适应度, 则接受新解作为新的解; 否则, 以一定的概率接受新解, 其中 T 表示当前温度, fit 表示解的适应度.

$$P = \begin{cases} 1, & \text{fit}_{\text{new}} < \text{fit}_{\text{old}}, \\ \exp^{-\left(\frac{\text{fit}_{\text{new}} - \text{fit}_{\text{old}}}{T}\right)}, & \text{others.} \end{cases} \quad (23)$$

5 算例分析

实验在配备了 AMD Ryzen 7 6800H 3.20 GHZ 和 32 G RAM 的电脑上运行. 算法使用 Python 3.9 编写, 求解器使用 Gurobi 9.5.2.

5.1 数据说明

基于青岛前湾自动化码头生成码头前沿路网信息等实验所需数据, 水平运输设备选为升降式自动导引车 (lift-automated guided vehicle, L-AGV). YC 的缓存区容量为 3, 最大允许延误时间为 30 s. L-AGV 的平均运行速度可以综合反映其运行过程中的载重量、天气、转弯次数、拥堵、冲突等因素的影响. 实验中 L-AGV 最大行驶速度为 6 m/s, 最小速度为 2 m/s, 得到 50、100、200、500、1000 和 5000 等不同规模样本数据.

5.2 小规模算例

小规模算例基于作业线的原则对任务分配. 设定 $\theta = 0.5$, $\varepsilon = 0.3$, 任务数量取 20 至 100, $|\Omega|$ 取 50 至 5000, 结果如表 1 所示.

由实验结果可知, 小规模案例下分支切割算法可以在较短的时间内求得精确解, 求解时间随着历史行驶时间数量的增加而增大. 另外, 由于模型中考虑了固定使用成本, 可最小化 L-AGV 使用数量.

表 1 小规模算例求解结果

序号	规模 (数量)		AGVs	目标值	计算时间 (不同样本数量)				
	任务箱	QCs/YCs			50	100	500	1000	5000
1	20	1/1	2	1628.2	0.47	0.23	0.82	1.64	11.33
2		1/2	2	1624.4	0.18	0.21	0.84	1.64	11.35
3	30	1/1	2	1886.2	0.42	0.6	2.59	5.20	33.59
4		1/2	2	1878.6	0.38	0.59	2.58	5.27	35.86
5	40	1/1	2	2144.2	0.74	1.28	5.68	11.28	63.89
6		1/2	2	2136.6	0.76	1.59	5.74	11.42	65.29
7	50	1/1	2	2402.2	1.41	2.40	10.74	21.33	113.70
8		1/2	2	2394.6	1.50	2.56	11.55	23.62	149.73
9	60	1/1	2	2660.2	2.34	4.14	18.45	36.87	191.81
10		1/2	2	2652.6	2.37	4.20	18.61	37.33	191.63
11	80	1/1	2	3176.2	5.67	9.58	42.98	87.17	432.94
12		1/2	2	3168.6	5.51	9.64	43.50	86.19	447.47
13	100	1/1	2	3692.2	10.37	18.53	83.40	165.26	843.31
14		1/2	2	3684.6	10.45	18.73	84.65	167.27	846.57

5.3 大规模算例

基于作业面原则分配可以最大化作业效率, 但也增加了问题的复杂度. 对于大规模案例, 基于作业面的分配方式, 分别用启发式算法和分支切割算法求解. 测试算例中任务数量为 30 至 200, QC 和 YC 配置分为 1/2、2/4、3/6 三种情况. 设定 $\theta = 0.5$, $\varepsilon = 0.3$, $|\Omega| = 200$, 求解时间限制为 3600 s. 启发式算法中, 设定 $T^k = 3000$, $T^f = 1$, $\delta = 0.6$, 邻域搜索算子总数 $H = 6$, 其中四个基于算子 1), 设定其交换的任务数量为 h 为 2 或 3, 并且执行 h 次; 其余两个为随机选取算子 2) ~ 算子 4) 中的两个算子, 并分别执行 1 次和 2 次. 局部搜索算子集合 \mathbb{L} 为算子 2) ~ 算子 4). 每次搜索结束使用算子 5) 随机扰动, 令任务数量为 $n = \min(50\% * |C|, 10\% * |C| + n)$, 其中 $|C|$ 表示任务总数. 为了提高算子执行后解的可行性, 随机选择第一个任务序列中的任务或插入位置后, 通过时间窗选择第二个任务序列中的任务或插入位置, 再进行搜索. 实验结果如表 2 所示.

由实验结果可知, 分支切割算法在短时间内找到了 16 个大规模实例中的 10 个实例的解, 平均求解精度在 96% 以上. 启发式算法可以在牺牲较小求解精度的前提下, 较快获得问题的次优解. 与分支切割算法相比平均结果平均误差不足为 2%. 部分算例由于求解精度的问题, 结果优于分支切割算法.

表 2 大规模算例求解结果

序号	任务箱/QCs/YCs	启发式算法			分支切割算法				Gap 2
		AGVs	目标值 1	时间	AGVs	目标值 2	Gap 1	时间	
1	30/2/4	4	2983.2	10.66	4	2975.6	18.00%	3600.00	0.25%
2	50/2/4	4	3514.4	19.98	4	3484.0	15.40%	3600.00	0.87%
3	80/2/4	3	3758.9	84.25	3	3713.3	0.20%	117.51	1.01%
4	100/2/4	3	4305.3	125.56	3	4229.3	0.18%	240.46	1.50%
5	120/1/2	2	4200.6	256.56	2	4204.4	0.09%	355.96	-0.09%
6	120/2/4	3	4878.3	177.79	3	4749.1	0.24%	392.81	0.93%
7	120/3/6	4	5461.0	138.56	4	5274.8	0.00%	399.73	3.41%
8	150/1/2	2	4974.6	434.64	2	4978.4	0.08%	699.08	-0.08%
9	150/2/4	3	5675.1	313.12	3	5724.5	3.46%	3600.00	-0.60%
10	150/3/6	4	6257.8	236.95	4	6064.0	0.31%	774.21	3.04%
11	180/1/2	2	5752.4	833.10	2	5752.4	0.00%	1228.78	0.00%
12	180/2/4	4	7016.6	252.12	4	6834.2	7.75%	3600.00	2.60%
13	180/3/6	4	7191.4	350.43	4	6838.0	0.22%	1693.31	4.86%
14	200/1/2	2	6268.4	1050.52	2	6268.4	0.00%	1674.50	0.00%
15	200/2/4	4	7566.8	317.77	4	7350.2	7.20%	3600.00	2.86%
16	200/3/6	4	7802.4	404.49	4	7528.8	10.80%	3600.00	3.51%

注: Gap 1 表示 Gurobi 提供的求解结果与最优下界的差距, Gap 2 = (目标值 1 - 目标值 2)/目标值 1.

5.4 与 SAA 对比

SAA 是一种处理随机参数的方法^[11]. 为了验证 DRO 方法对减少 QC 和 YC 等待风险的效果. 基于表 2 中的案例 5、6、7、14、15、16, 设定 $\theta = 0.5$, $\varepsilon = 0.3$, $|\Omega| = 200$, 与基于 SAA 的确定性模型方案对比, 采用 $2 * |\Omega|$ 个数据进行样本外数据对比测试, 结果如表 3 所示. Gap 表示 SAA 方法与 DRO 方法之间的目标值差距.

表 3 与 SAA 方案对比结果

序号	任务箱/QCs/YCs	方法	方案表现						Gap
			目标值	最大概率	概率和	最大期望	期望和	延迟数量	
1	30/2/4	SAA	2430.9	0.39	0.47	2.69	2.86	4	22.41%
2		DRO	2975.6	0.09	0.17	0.10	0.27	4	
3	80/2/4	SAA	3705.7	0.31	1.53	3.51	15.31	9	0.21%
4		DRO	3713.3	0.11	0.22	0.21	0.34	2	
5	150/2/4	SAA	5511.7	0.39	2.77	4.60	23.72	14	3.86%
6		DRO	5724.5	0.23	1.64	2.06	9.30	17	
7	200/1/2	SAA	6268.4	0.62	3.19	6.80	25.35	15	0.00%
8		DRO	6268.4	0.23	0.75	0.85	1.67	6	
9	200/2/4	SAA	6816.9	0.89	5.33	12.29	57.88	20	7.82%
10		DRO	7350.2	0.20	0.39	0.74	1.32	3	
11	200/3/6	SAA	6816.9	0.62	4.40	6.80	39.67	19	10.44%
12		DRO	7528.8	0.23	2.11	3.69	16.45	15	

注: Gap = (目标值_{DRO} - 目标值_{SAA})/目标值_{SAA}.

由实验结果可知. 本文方法比 SAA 方法平均降低 QC 和 YC 等待风险达 60% 以上, 大大增加 AGV 与 QC 和 YC 之间的作业协同性. 虽然平均造成作业成本的升高约 7%. 但 QC 和 YC 单位时间作业成本较高, 降低其等待时间可以有效降低总体作业水平, 保证装卸作业效率.

5.5 加速策略

下面基于表 2 中 3、4、6、8、9、10 案例对加速策略进行实验. 设定 $\theta = 0.5$, $\varepsilon = 0.1$, $|\Omega| = 200$. 表 4 显示了不同加速技术对本文算法的影响.

表 4 不同加速技术的影响

加速方法	项目	任务规模					
		80/2/4	100/2/4	120/2/4	150/1/2	150/2/4	150/3/6
分支切割	目标值 0	3861.5	4388.9	4958.1	4978.4	5770.1	6326.2
	时间	105.3	206.7	359.3	704.4	716.1	754.3
(无) 弧消除	目标值 1	3861.5	4388.9	4958.1	4978.4	5861.3	6493.4
	时间	3600.0	3600.0	3600.0	3600.0	3600.0	3600.0
(无) 存储结构	目标值 2	3861.5	4388.9	4958.1	4978.4	5796.7	6364.2
	时间	263.15	620.19	1929.72	3481.01	3600.00	3600.00
(无) 热启动	目标值 3	3717.1	4766.4	-	4974.6	-	6582.1
	时间	119.91	3600	-	912	-	3600
(无) 剪枝策略	目标值 4	3861.5	4388.9	4958.1	4978.4	5770.1	6326.2
	时间	126.51	238.49	388.36	724.4	763.32	2655.02
	Gap 1 (%)	0.00	0.00	0.00	0.00	1.58	2.64
	Gap 2 (%)	0.00	0.00	0.00	0.00	0.46	0.6
	Gap 3 (%)	-3.74	8.60	-	-0.08	-	4.05
	Gap 4 (%)	0.00	0.00	0.00	0.00	0.00	0.00

注: $\text{Gap } n = (\text{目标值 } n - \text{目标值 } 0) / \text{目标值 } 0$.

实验结果可知, 四种加速策略均可以在不同程度上进行加速. 不考虑弧消除策略时所有案例均无法在规定时间内获得最优解, 但求解结果良好. 证明弧消除策略可以有效缩减解空间, 加快算法收敛. 不使用储存结构时, 大部分案例求解时间有所延长, 小部分案例甚至无法完成求解. 剪枝策略有一定的加速效果, 可与存储结构搭配使用. 热启动策略可能会影响算法的求解质量. 基于初始解可以很快就达到求解精度; 而不使用该策略时, 从初始结果开始搜索解空间, 有一定概率找到更优的解, 但规定时间内可能无法完成求解.

5.6 敏感性分析

5.6.1 目标服务水平

基于表 2 中的案例 4, 令 $\theta = 0.5$, $|\Omega| = 200$, 基于 $2 * |\Omega|$ 个样本外数据评估不同 LoS 水平下任务分配方案的质量. 设定 LoS 从 0.5 变至 0.9, 表 5 显示了不同 LoS 水平下的实验结果. 结果表明, 随着 LoS 的增大, QC 和 YC 的等待风险倾向越低, 体现了模型的有效性.

5.6.2 Wasserstein 距离

模糊集 $\mathcal{F}(\theta)$ 中 θ 表示真实分布与样本空间之间的距离, 控制对经验分布的拟合程度. 当 $\theta = 0$ 时, 模糊集 $\mathcal{F}(\theta)$ 将直接退化为样本空间. θ 越大, $\mathcal{F}(\theta)$ 越发散, 真实分布与样本的间隔越远. 令 $|\Omega| = 200$, $\varepsilon = 0.3$, 用 400 个样本外数据评估所得的任务分配方案的质量. 选取表 2 中的案例 4, θ 取 0 至 3. 表 6 显示了不同 θ 下的实验结果.

尽管模型固有的随机性使趋势并不明显, 但由于 θ 表示实际分布与样本数据之间的“距离”, 倾向于随着 θ 的增大, 任务分配方案鲁棒性越强, 目标值也随之增加.

6 结论

AGV 之间的冲突、拥堵等各类情况使其在路网中的行驶时间产生波动, 导致 QC 和 YC 等待, 降低装卸效率. 本文研究了考虑行驶时间不确定的 AGV 任务分配问题. 根据历史行驶时间, 在控制 QC

表 5 Los 敏感性分析结果

序号	任务箱/QCs/YCs	服务水平	数据	表现				
				最大概率	概率和	最大期望	期望和	延迟数量
1	100/2/4	0.50	样本内	0.20	1.23	2.12	12.28	11
2			样本外	0.39	2.05	3.51	17.60	11
3		0.55	样本内	0.12	0.37	0.60	0.99	4
4			样本外	0.23	0.54	0.85	1.30	4
5		0.60	样本内	0.12	0.37	0.60	0.99	4
6			样本外	0.23	0.54	0.85	1.30	4
7		0.65	样本内	0.12	0.36	0.60	1.06	6
8			样本外	0.23	0.53	0.85	1.37	6
9		0.70	样本内	0.12	0.66	0.92	4.18	7
10			样本外	0.24	1.02	1.12	5.34	7
11		0.75	样本内	0.11	0.36	0.46	0.84	4
12			样本外	0.14	0.44	0.57	1.01	4
13		0.80	样本内	0.09	0.29	0.17	0.42	5
14			样本外	0.11	0.35	0.21	0.49	5
15		0.85	样本内	0.03	0.07	0.08	0.16	3
16			样本外	0.04	0.08	0.09	0.17	3
17		0.90	样本内	0.00	0.00	0.00	0.00	0
18			样本外	0.00	0.00	0.00	0.00	0

表 6 Wasserstein 距离 θ 敏感性分析结果

序号	规模	θ	数据	目标值	表现				
					最大概率	概率和	最大期望	期望和	延迟数量
1	100/2/4	0.0	样本内	4229.3	0.12	0.37	0.60	0.99	4
2			样本外		0.23	0.54	0.85	1.30	4
3		0.1	样本内	4229.3	0.12	0.42	0.60	1.07	6
4			样本外		0.23	0.59	0.85	1.38	6
5		0.2	样本内	4229.3	0.12	0.44	0.60	1.15	7
6			样本外		0.23	0.62	0.85	1.47	7
7		0.5	样本内	4229.3	0.12	0.66	0.92	4.18	7
8			样本外		0.24	1.02	1.12	5.34	7
9		1.0	样本内	4236.9	0.09	0.26	0.17	0.38	3
10			样本外		0.11	0.31	0.21	0.45	3
11		1.5	样本内	4236.9	0.09	0.18	0.17	0.29	2
12			样本外		0.11	0.22	0.21	0.34	2
13		3.0	样本内	4240.7	0.00	0.00	0.00	0.00	0
14			样本外		0.00	0.00	0.00	0.00	0

和 YC 等待风险的前提下, 得到最小作业成本的任务分配方案.

如何刻画行驶时间的不确定性是本文主要挑战之一. 本文基于数据驱动的思想, 引入 DRO 方法建立刻画不确定的行驶时间, 通过机会约束控制 QC 和 YC 的等待风险, 建立鲁棒机会约束的任务分配模型. 为了求解该模型, 首先使用 CVaR 近似鲁棒机会约束, 并通过偶理论等数学方法得到 CVaR 的数学闭式解. 随后针对小规模问题设计了一个精确的分支切割算法, 针对大规模问题设计了一个变邻域搜索-模拟退火混合启发式算法进行求解. 大量计算实验表明, 本文提出的模型可以有效利用行驶的经验数据刻画 AGV 任务分配问题, 提出的方法能够在合理的时间内找到较优的解, 几种加速策略均可以有效加速算法求解. 与 SAA 方法相比, 基于数据驱动的 DRO 方法能够降低岸桥和场桥等待风险

60% 以上, 有效提高任务分配方案的鲁棒性。

未来的研究方向可以讨论模糊集由其他距离度量定义的问题, 比如基于矩的模糊集、KL 散度等。另一个研究方向是采用机器学习等其他方法预测不确定的行驶时间。

参考文献

- [1] Yang Z M, Li C H, Zhao Q C. Dynamic time estimation based AGV dispatching algorithm in automated container terminal[C]// 2018 37th Chinese Control Conference (CCC). New York: IEEE, 2018: 7868–7873.
- [2] 张亚琦, 杨斌, 胡志华, 等. 自动化码头 AGV 充电与作业的集成调度研究 [J]. 计算机工程与应用, 2017, 53(18): 257–262.
Zhang Y Q, Yang B, Hu Z H, et al. Research of AGV charging and job integrated scheduling at automated container terminal[J]. Computer Engineering and Applications, 2017, 53(18): 257–262.
- [3] 王志刚, 胡伟新. 考虑 AGV 路径冲突的自动化集装箱码头装卸设备的协同调度 [J]. 工业工程与管理, 2023, 28(6): 80–92.
Wang Z G, Hu W X. Cooperative scheduling of loading and unloading equipment in automated container terminal considering AGV path conflict[J]. Industrial Engineering and Management, 2023, 28(6): 80–92.
- [4] 王里, 王超, 安利峰, 等. 港口 ART 自动驾驶系统设计和应用 [J]. 港口科技, 2022(8): 5–10.
Wang L, Wang C, An L F, et al. Design and application of port ART autonomous driving system[J]. Science & Technology of Ports, 2022(8): 5–10.
- [5] 冷典典, 杜鹏, 陈建廷, 等. 面向自动化集装箱码头的 AGV 行驶时间估计 [J]. 计算机科学, 2022, 49(9): 208–214.
Leng D D, Du P, Chen J T, et al. Automated container terminal oriented travel time estimation of AGV[J]. Computer Science, 2022, 49(9): 208–214.
- [6] Angeloudis P, Bell M G H. An uncertainty-aware AGV assignment algorithm for automated container terminals[J]. Transportation Research Part E: Logistics and Transportation Review, 2010, 46(3): 354–366.
- [7] 马越汇, 胡志华. 不确定环境下自动化集装箱码头 AGV 调度与配置问题 [J]. 广西大学学报 (自然科学版), 2016, 41(2): 589–597.
Ma Y H, Hu Z H. The dispatching and scheduling problems of AGVs at automated container terminals under uncertainty conditions[J]. Journal of Guangxi University (Nat Sci Ed), 2016, 41(2): 589–597.
- [8] 徐卉. 考虑不确定因素的自动化集装箱码头岸桥与 AGV 联合调度优化 [D]. 大连: 大连海事大学, 2023.
Xu H. Optimization of combined dispatching of quay cranes and AGVs for automated container terminal considering uncertainties[D]. Dalian: Dalian Maritime University, 2023.
- [9] Jian W T, Zhu J S, Zeng Q C. An optimization model of integrated AGVs scheduling and container storage problems for automated container terminal considering uncertainty[J]. Symmetry, 2021, 13(10). doi: 10.3390/sym13101904.
- [10] 朱禹仲. 考虑拥堵的自动化码头 ALV 调度与堆存位置分配优化模型 [D]. 大连: 大连海事大学, 2019.
Zhu Y Z. Optimization model of ALV task dispatching and container location considering congestion at automated terminal[D]. Dalian: Dalian Maritime University, 2019.
- [11] 冯春, 陈木泉, 蒋雪. 随机充电需求下城市电动汽车充电站选址优化 [J]. 计算机仿真, 2022, 39(11): 193–198.
Feng C, Chen M Q, Jiang X. Location optimization of urban electric vehicle charging station under random charging demand[J]. Computer Simulation, 2022, 39(11): 193–198.
- [12] Sun B, Zhai G, Li S, et al. Multi-resource collaborative scheduling problem of automated terminal considering the AGV charging effect under COVID-19[J]. Ocean & Coastal Management, 2023, 232: 106422.
- [13] 李林蔓, 李雨青, 王孟雅, 等. 基于启发式规则的自动化码头换电式 AGV 调度优化方法 [J]. 运筹与管理, 2023, 32(10): 9–15.
Li L M, Li Y Q, Wang M Y, et al. Optimization method for power-changing AGV scheduling of automatic terminal based on heuristic rules[J]. Operations Research and Management Science, 2023, 32(10): 9–15.
- [14] Zhang H, Qi L, Luan W, et al. Double-cycling AGV scheduling considering uncertain crane operational time at container terminals[J]. Applied Sciences, 2022, 12(10). doi: 10.3390/app/12104820.
- [15] Zhang H, Qi L, Ma H. A method for AGV double-cycling scheduling at automated container terminals[M]// Intelligent Information Processing XI, 2022: 495–507.

- [16] 霍凯歌, 张亚琦, 胡志华. 自动化集装箱码头多载 AGV 调度问题研究 [J]. 大连理工大学学报, 2016, 56(3): 244–251.
Huo K G, Zhang Y Q, Hu Z H. Research on multi-AGV scheduling problem in automated container terminals[J]. Journal of Dalian University of Technology, 2016, 56(3): 244–251.
- [17] 霍凯歌, 胡志华. 基于遗传算法的自动化集装箱码头多载 AGV 调度 [J]. 上海海事大学学报, 2016, 37(3): 7–12.
Huo K G, Hu Z H. Multi-AGV scheduling in automated container terminals based on genetic algorithm[J]. Journal of Shanghai Maritime University, 2016, 37(3): 7–12.
- [18] Wang Z H, Zeng Q C. A branch-and-bound approach for AGV dispatching and routing problems in automated container terminals[J]. Computers & Industrial Engineering, 2022, 166: 107968.
- [19] Yue L, Fan H. Dynamic scheduling and path planning of automated guided vehicles in automatic container terminal[J]. IEEE/CAA Journal of Automatica Sinica, 2022, 9(11): 2005–2019.
- [20] Gong L, Huang Z, Xiang X, et al. Real-time AGV scheduling optimisation method with deep reinforcement learning for energy-efficiency in the container terminal yard[J]. International Journal of Production Research, 2024, 62: 7722–7742.
- [21] Chen C, Hu Z H, Wang L. Scheduling of AGVs in automated container terminal based on the deep deterministic policy gradient (DDPG) using the convolutional neural network (CNN)[J]. Journal of Marine Science and Engineering, 2021, 9(12): 1439.
- [22] 贾石岩, 王泽浩, 张旭, 等. 基于偏好函数与深度信念网络的自动化码头 AGV 实时调度算法 [J]. 中国航海, 2020, 43(1): 121–127.
Jia S Y, Wang Z H, Zhang X, et al. Real-time scheduling algorithm for AGV in automated container terminals based on preference function and deep belief net[J]. Navigation of China, 2020, 43(1): 121–127.
- [23] Yang M, Bian Y, Ma L, et al. Research on traffic control algorithm based on multi-AGV path planning[C]// 2021 IEEE International Conference on Systems, Man, and Cybernetics (SMC), 2021: 697–702.
- [24] Cao Y, Yang A, Liu Y, et al. AGV dispatching and bidirectional conflict-free routing problem in automated container terminal[J]. Computers & Industrial Engineering, 2023, 184: 109611.
- [25] 张素云, 杨勇生, 梁承姬, 等. 自动化码头多 AGV 路径冲突的优化控制研究 [J]. 交通运输系统工程与信息, 2017, 17(2): 83–89.
Zhang S Y, Yang Y S, Liang C J, et al. Optimal control of multiple AGV path conflict in automated terminals[J]. Journal of Transportation Systems Engineering and Information Technology, 2017, 17(2): 83–89.
- [26] Li Q, Pogromsky A, Adriaansen T, et al. A control of collision and deadlock avoidance for automated guided vehicles with a fault-tolerance capability[J]. International Journal of Advanced Robotic Systems, 2016, 13(2): 64.
- [27] Zhou P, Lin L, Kim K H. Anisotropic Q-learning and waiting estimation based real-time routing for automated guided vehicles at container terminals[J]. Journal of Heuristics, 2023, 29(2): 207–228.
- [28] Hu H, Yang X, Xiao S, et al. Anti-conflict AGV path planning in automated container terminals based on multi-agent reinforcement learning[J]. International Journal of Production Research, 2023, 61(1): 65–80.
- [29] 樊伟, 范英, 谭忠富, 等. 基于多层利益共享的虚拟电厂参与电碳市场分布鲁棒优化模型 [J]. 系统工程理论与实践, 2024, 44(2): 661–683.
Fan W, Fan Y, Tan Z F, et al. Distributionally robust optimization model for virtual power plant participation in electricity carbon market based on multi-layer benefit sharing[J]. Systems Engineering — Theory & Practice, 2024, 44(2): 661–683.
- [30] 徐婉迪, 罗俊, 范薇薇. 不确定环境下并网型光储微电网的容量规划 [J]. 系统工程理论与实践, 2022, 42(4): 981–1000.
Xu W D, Luo J, Fan W W. Capacity planning of grid-connected PV-and-storage microgrid under uncertainty[J]. Systems Engineering — Theory & Practice, 2022, 42(4): 981–1000.
- [31] 王浩, 甘蜜, 魏力飞, 等. 面向山区铁路工程物流基地选址的分布鲁棒优化模型 [J]. 铁道学报, 2024, 46(1): 103–112.
Wang H, Gan M, Wei L F, et al. Distributionally robust optimization model for logistics facility location in mountainous railway projects[J]. Journal of the China Railway Society, 2024, 46(1): 103–112.
- [32] 代文强, 陈琳, 章满月. 道路容量不确定情形下可靠应急疏散路径规划问题 [J]. 系统工程理论与实践, 2022, 42(9): 2486–2495.

- Dai W Q, Chen L, Zhang X Y. Reliable emergency evacuation route planning under road capacity uncertainty[J]. *Systems Engineering — Theory & Practice*, 2022, 42(9): 2486–2495.
- [33] 黎俊, 王曙明. 基于截断需求的分布鲁棒报童问题 [J]. *系统工程理论与实践*, 2022, 42(5): 1260–1276.
Li J, Wang S M. Distributionally robust newsvendor problem with censored demand[J]. *Systems Engineering — Theory & Practice*, 2022, 42(5): 1260–1276.
- [34] 金小丹, 周泓. 需求分布未知下的供应链融资及鲁棒订购策略 [J]. *系统工程理论与实践*, 2021, 41(5): 1263–1271.
Jin X D, Zhou H. Joint supply chain finance and robust order decision with unknown distribution of demand[J]. *Systems Engineering — Theory & Practice*, 2021, 41(5): 1263–1271.
- [35] 竺如洁, 韦化, 白晓清. 多源动态最优潮流的分布鲁棒优化方法 [J]. *中国电机工程学报*, 2020, 40(11): 3489–3498.
Zhu R J, Wei H, Bai X Q. Distributionally robust optimization of multi-energy dynamic optimal power flow[J]. *Proceedings of the CSEE*, 2020, 40(11): 3489–3498.
- [36] Wang Z H, Zeng Q V, Li X C, et al. A branch-and-price heuristic algorithm for the ART and external truck scheduling problem in an automated container terminal with a parallel layout[J]. *Transportation Research Part E: Logistics and Transportation Review*, 2024, 184: 103464.
- [37] Esfahani P M, Kuhn D. Data-driven distributionally robust optimization using the Wasserstein metric: Performance guarantees and tractable reformulations[J]. *Mathematical Programming*, 2018, 171(1–2): 115–166.
- [38] Laporte G, Louveaux F, Mercure H. The vehicle routing problem with stochastic travel times[J]. *Transportation Science*, 1992, 26(3): 161–170.
- [39] Ben-Tal A, Teboulle M, Ben-Tal A, et al. Expected utility, penalty functions, and duality in stochastic nonlinear programming[J]. *Management Science*, 1986, 32(11): 1445–1466.
- [40] Nemirovski A, Shapiro A. Convex approximations of chance constrained programs[J]. *SIAM Journal on Optimization*, 2007, 17(4): 969–996.
- [41] Xie W J. On distributionally robust chance constrained programs with Wasserstein distance[J]. *Mathematical Programming*, 2021, 186(1): 115–155.

附录

A 作业网络图

图 A1 为一个简单的作业网络图示例, 其中包括 6 个作业任务节点和 2 个虚拟节点, 存在三个卸船任务和三个装船任务.

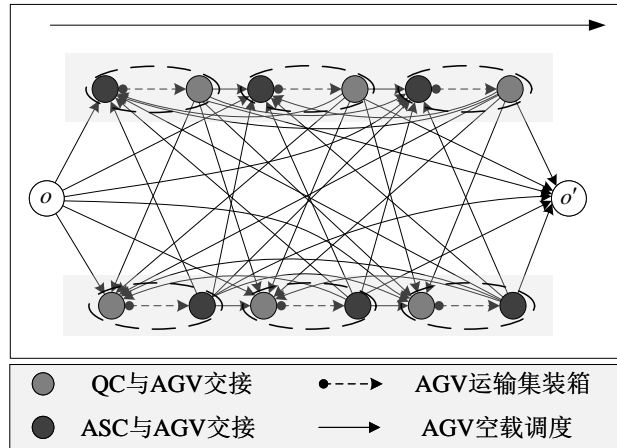


图 A1 作业网络图示例

B 集装箱装卸任务的时间窗生成方法

根据 QC 和 YC 作业计划, 以下两种方法确定 AGV 作业任务时间窗.

1) 需要精确对接: 如果水平运输设备与 QC 或 YC 存在精确耦合关系, 需提前到达等待作业. 时间窗 $[a_i, b_i]$ 下界 a_i 为集装箱作业任务到达缓存区的时间; 时间窗上界 $b_i = a_i + \xi_{\max}$, ξ_{\max} 为允许延误的最大时间.

2) 无需精确对接: 如果水平运输设备与 QC 或 YC 无耦合关系, 无需提前到达. 但由于缓存区容量有限, 水平运输设备需尽快到达, 否则影响作业效率. 因此根据所在缓存区数量, 应用服务规则生成任务的时间窗 $[a_i, b_i]$.

QC 与 (L-)AGV 交接集装箱, 两者需要精密耦合, 可能出现互相等待. 而 ALV 具有抓取功能, 无需精密耦合. 对于 YC 来说, 缓存区如有支架, 可暂时存放集装箱, (L-)AGV、ALV 均能够自主的将集装箱放置在支架上或者拾取集装箱, 无需精密耦合, 避免互相等待. 对于 AGV, 如支架具有举升功能, 能够达到同样效果. 表 B1 说明了各种情境下对应的的时间窗生成方法.

表 B1 不同设备之间对接的时间窗生成方法

对接设备	缓存区	AGV		L-AGV	ALV
		可举升	不可举升		
QC	路网	方法 2	方法 1	方法 1	方法 2
YC	AGV 支架		方法 2	方法 2	方法 2

图 B1 给出了一个简单的时间窗示例, 其中假设水平运输设备为 L-AGV, 堆场测缓存区容量为 3.

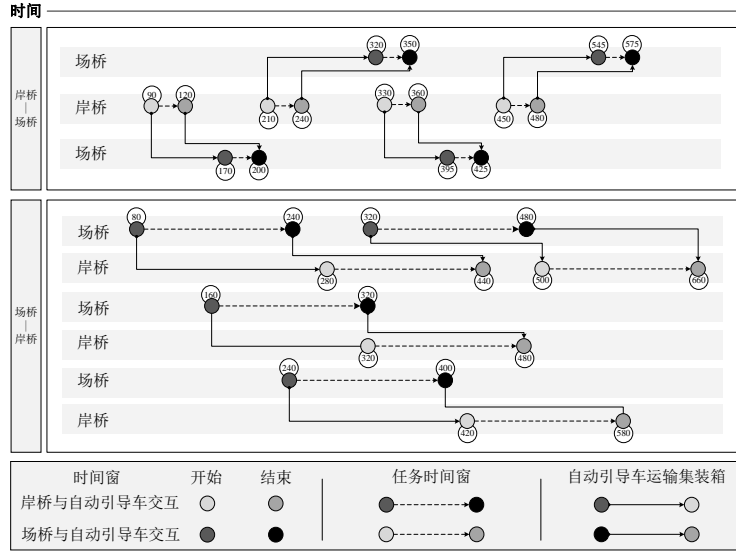


图 B1 任务时间窗示例

C 等待时间函数证明

假设 \mathbf{x} , s_i 和 t_{ij} 已知, 得到开始时间为 $T_j = \max\{T_i + s_i + t_{ij}, a_i\}$. 根据 \mathbf{x} 可以得到多条从起点 o 到终点 o' 的任务序列. 根据 $(o, 1, \dots, i-1, i, i+1, \dots, o')$, 计算到达时间为:

$$\begin{aligned} T_o &= a_o = 0, \\ T_1 &= \max\{T_o + t_{o1}, a_1\}, \\ T_2 &= \max\{T_1 + s_1 + t_{12}, a_2\}, \\ &\dots \\ T_j &= \max\{T_{j-1} + s_{j-1} + t_{(j-1)j}, a_j\}. \end{aligned}$$

因此, 通过递归计算得到该条任务序列上每个任务 $i \in C$ 服务开始时间为:

$$T_j = \max_{i \in C_j(\mathbf{x})} \left\{ a_i + \sum_{(i,k) \in A_{ij}(\mathbf{x})} (s_i + t_{ik}) \right\}.$$

根据延迟时间的定义 $\xi_i = T_i - b_i$, 得到:

$$\xi_j = \max_{i \in C_j(\mathbf{x})} \left\{ a_i + \sum_{(i,k) \in A_{ij}(\mathbf{x})} (s_i + t_{ik}) \right\} - b_j.$$

D CVaR 约束转化

不同情景 ω 下, 任务执行时间 \mathbf{s} 和任务之间的调度时间 \mathbf{t} 不同, 延误时间 $\tilde{\xi}_i(\mathbf{x}, \omega)$ 主要受到两者影响. 为了方便表述, 将 $\tilde{\xi}_i(\mathbf{x}, \omega)$ 记为 $\tilde{\xi}_j(\mathbf{x}, \tilde{\mathbf{s}}, \tilde{\mathbf{t}})$, 其中 \mathbf{s} 和 \mathbf{t} 分别表示任务执行时间和作业时间的矢量形式.

第一步: 根据 1 型 Wasserstein 距离模糊集的定义, $\sup_{\mathbb{P} \in \mathcal{F}(\theta)} \mathbb{E}_{\mathbb{P}}[(\tilde{\xi}_j(\mathbf{x}, \tilde{\mathbf{s}}, \tilde{\mathbf{t}}) - \beta)^+]$ 应写为如下形式, 其中 \otimes 表示 Kronecker 运算.

$$\begin{aligned} Z_1 &= \sup \mathbb{E}_{\mathbb{P}}[(\tilde{\xi}_j(\mathbf{x}, \tilde{\mathbf{s}}, \tilde{\mathbf{t}}) - \beta)^+] \\ \text{s.t. } &\mathbb{E}_{\mathbb{P}}[\|(\tilde{\mathbf{t}} + \tilde{\mathbf{s}} \otimes \mathbf{1}_{|V|}^{\top}) - (\mathbf{t} + \mathbf{s} \otimes \mathbf{1}_{|V|}^{\top})\|_1] \leq \theta, \\ &[(\tilde{\mathbf{t}} + \tilde{\mathbf{s}} \otimes \mathbf{1}_{|V|}^{\top}), (\mathbf{t} + \mathbf{s} \otimes \mathbf{1}_{|V|}^{\top})] \sim \bar{\mathbb{P}}, \end{aligned}$$

$$\begin{aligned}
& \rightarrow \begin{cases} \pi_\omega \geq \sup_{(\mathbf{t}+\mathbf{s} \otimes \mathbf{1}_{|V|}^\top) \in U} [\mathbf{v}_{ij}(\mathbf{x})^\top (\mathbf{t} + \mathbf{s} \otimes \mathbf{1}_{|V|}^\top) - \max_{\|\mathbf{q}_{\omega i}\|_\infty \leq r} \mathbf{q}_{\omega i}^\top (\mathbf{t} + \mathbf{s} \otimes \mathbf{1}_{|V|}^\top - (\hat{\mathbf{t}}_\omega + \hat{\mathbf{s}}_\omega \otimes \mathbf{1}_{|V|}^\top))] + \\ a_i - b_j - \beta, \forall \omega \in \Omega, i \in C_j(\mathbf{x}), \\ \pi_\omega \geq \sup_{(\mathbf{t}+\mathbf{s} \otimes \mathbf{1}_{|V|}^\top) \in U} [-\max_{\|\mathbf{q}_{\omega 0}\|_\infty \leq r} \mathbf{q}_{\omega 0}^\top (\mathbf{t} + \mathbf{s} \otimes \mathbf{1}_{|V|}^\top - (\hat{\mathbf{t}}_\omega + \hat{\mathbf{s}}_\omega \otimes \mathbf{1}_{|V|}^\top))], \forall \omega \in \Omega. \end{cases} \\
& \rightarrow \begin{cases} \pi_\omega \geq \min_{\|\mathbf{q}_{\omega i}\|_\infty \leq r} \sup_{(\mathbf{t}+\mathbf{s} \otimes \mathbf{1}_{|V|}^\top) \in U} [\mathbf{v}_{ij}(\mathbf{x})^\top (\mathbf{t} + \mathbf{s} \otimes \mathbf{1}_{|V|}^\top) - \mathbf{q}_{\omega i}^\top (\mathbf{t} + \mathbf{s} \otimes \mathbf{1}_{|V|}^\top - (\hat{\mathbf{t}}_\omega + \hat{\mathbf{s}}_\omega \otimes \mathbf{1}_{|V|}^\top))] + \\ a_i - b_j - \beta, \forall \omega \in \Omega, i \in C_j(\mathbf{x}), \\ \pi_\omega \geq \min_{\|\mathbf{q}_{\omega 0}\|_\infty \leq r} \sup_{(\mathbf{t}+\mathbf{s} \otimes \mathbf{1}_{|V|}^\top) \in U} [-\mathbf{q}_{\omega 0}^\top (\mathbf{t} + \mathbf{s} \otimes \mathbf{1}_{|V|}^\top - (\hat{\mathbf{t}}_\omega + \hat{\mathbf{s}}_\omega \otimes \mathbf{1}_{|V|}^\top))], \forall \omega \in \Omega. \end{cases} \\
& \rightarrow \begin{cases} \pi_\omega \geq \mathbf{q}_{\omega i}^\top (\hat{\mathbf{t}}_\omega + \hat{\mathbf{s}}_\omega \otimes \mathbf{1}_{|V|}^\top) + \sup_{\mathbf{t}+\mathbf{s} \otimes \mathbf{1}_{|V|}^\top \geq \underline{\mathbf{t}+\mathbf{s} \otimes \mathbf{1}_{|V|}^\top}} [(\mathbf{v}_{ij}(\mathbf{x}) - \mathbf{q}_{\omega i})^\top (\mathbf{t} + \mathbf{s} \otimes \mathbf{1}_{|V|}^\top)] + \\ a_i - b_j - \beta, \forall \omega \in \Omega, i \in C_j(\mathbf{x}), \\ \pi_\omega \geq \mathbf{q}_{\omega 0}^\top (\hat{\mathbf{t}}_\omega + \hat{\mathbf{s}}_\omega \otimes \mathbf{1}_{|V|}^\top) + \sup_{\mathbf{t}+\mathbf{s} \otimes \mathbf{1}_{|V|}^\top \geq \underline{\mathbf{t}+\mathbf{s} \otimes \mathbf{1}_{|V|}^\top}} [-\mathbf{q}_{\omega 0}^\top (\mathbf{t} + \mathbf{s} \otimes \mathbf{1}_{|V|}^\top)], \forall \omega \in \Omega, \\ \|\mathbf{q}_{\omega i}\|_\infty \leq r, \forall \omega \in \Omega, i \in C_j(\mathbf{x}) \cup 0. \end{cases} \\
& \rightarrow \begin{cases} \pi_\omega \geq \mathbf{q}_{\omega i}^\top (\hat{\mathbf{t}}_\omega + \hat{\mathbf{s}}_\omega \otimes \mathbf{1}_{|V|}^\top) + \inf_{-\mathbf{p}_{\omega i} = \mathbf{v}_{ij}(\mathbf{x}) - \mathbf{q}_{\omega i}, \mathbf{p}_{\omega i} \geq 0} -\mathbf{p}_{\omega i}^\top (\mathbf{t} + \mathbf{s} \otimes \mathbf{1}_{|V|}^\top) + a_i - b_j - \beta, \\ \forall \omega \in \Omega, i \in C_j(\mathbf{x}), \\ \pi_\omega \geq \mathbf{q}_{\omega 0}^\top (\hat{\mathbf{t}}_\omega + \hat{\mathbf{s}}_\omega \otimes \mathbf{1}_{|V|}^\top) + \inf_{\mathbf{p}_{\omega 0} = \mathbf{q}_{\omega 0}, \mathbf{p}_{\omega 0} \geq 0} -\mathbf{p}_{\omega 0}^\top (\mathbf{t} + \mathbf{s} \otimes \mathbf{1}_{|V|}^\top), \forall \omega \in \Omega, \\ \|\mathbf{q}_{\omega i}\|_\infty \leq r, \forall \omega \in \Omega, i \in C_j(\mathbf{x}) \cup 0. \end{cases} \\
& \rightarrow \begin{cases} \pi_\omega \geq \mathbf{q}_{\omega i}^\top (\hat{\mathbf{t}}_\omega + \hat{\mathbf{s}}_\omega \otimes \mathbf{1}_{|V|}^\top) - \mathbf{p}_{\omega i}^\top (\mathbf{t} + \mathbf{s} \otimes \mathbf{1}_{|V|}^\top) + a_i - b_j - \beta, \forall \omega \in \Omega, i \in C_j(\mathbf{x}), \\ \pi_\omega \geq \mathbf{q}_{\omega 0}^\top (\hat{\mathbf{t}}_\omega + \hat{\mathbf{s}}_\omega \otimes \mathbf{1}_{|V|}^\top) - \mathbf{p}_{\omega 0}^\top (\mathbf{t} + \mathbf{s} \otimes \mathbf{1}_{|V|}^\top), \forall \omega \in \Omega, \\ -\mathbf{p}_{\omega i} = \mathbf{v}_{ij}(\mathbf{x}) - \mathbf{q}_{\omega i}, \forall \omega \in \Omega, i \in C_j(\mathbf{x}), \\ \mathbf{p}_{\omega 0} = \mathbf{q}_{\omega 0}, \forall \omega \in \Omega, \\ \mathbf{p}_{\omega i} \geq 0, \forall \omega \in \Omega, i \in C_j(\mathbf{x}) \cup 0, \\ \|\mathbf{q}_{\omega i}\|_\infty \leq r, \forall \omega \in \Omega, i \in C_j(\mathbf{x}) \cup 0. \end{cases} \\
& \rightarrow \begin{cases} \pi_\omega \geq (\hat{\mathbf{t}}_\omega + \hat{\mathbf{s}}_\omega \otimes \mathbf{1}_{|V|}^\top - \underline{\mathbf{t} + \mathbf{s} \otimes \mathbf{1}_{|V|}^\top})^\top \mathbf{p}_{\omega i} + \mathbf{v}_{ij}(\mathbf{x})^\top (\hat{\mathbf{t}}_\omega + \hat{\mathbf{s}}_\omega \otimes \mathbf{1}_{|V|}^\top) + a_i - b_j - \beta, \\ \forall \omega \in \Omega, i \in C_j(\mathbf{x}), \\ \pi_\omega \geq (\hat{\mathbf{t}}_\omega + \hat{\mathbf{s}}_\omega \otimes \mathbf{1}_{|V|}^\top - \underline{\mathbf{t} + \mathbf{s} \otimes \mathbf{1}_{|V|}^\top})^\top \mathbf{p}_{\omega 0}, \forall \omega \in \Omega, \\ r \geq \|\mathbf{p}_{\omega i} + \mathbf{v}_{ij}(\mathbf{x})\|_\infty, \forall \omega \in \Omega, i \in C_j(\mathbf{x}), \\ r \geq \|\mathbf{p}_{\omega 0}\|_\infty, \forall \omega \in \Omega, \\ \mathbf{p}_{\omega i} \geq 0, \forall \omega \in \Omega, i \in C_j(\mathbf{x}) \cup 0. \end{cases}
\end{aligned}$$

第二步: 观察上述约束发现, 为了使目标函数最优, 可以令 $\mathbf{p}_{\omega i} = 0$, 前 4 项约束可以进一步减少为:

$$\rightarrow \begin{cases} \pi_\omega \geq a_i + \mathbf{v}_{ij}(\mathbf{x})^\top (\hat{\mathbf{t}}_\omega + \hat{\mathbf{s}}_\omega \otimes \mathbf{1}_{|V|}^\top) - b_j - \beta, \forall \omega \in \Omega, i \in C_j(\mathbf{x}), \\ \pi_\omega \geq 0, \forall \omega \in \Omega, \\ r \geq \|\mathbf{v}_{ij}(\mathbf{x})\|_\infty, \forall \omega \in \Omega, i \in C_j(\mathbf{x}), \\ r \geq 0, \forall \omega \in \Omega. \end{cases}$$

根据多项式范数的定义可知 $\|\mathbf{v}_{ij}(\mathbf{x})\|_\infty = |A_{ij}(\mathbf{x})|^0 = 1$. 同时根据 $\hat{\xi}_j(\mathbf{x}, \hat{\mathbf{s}}_\omega, \hat{\mathbf{t}}_\omega)$ 的计算公式, 模型

进一步变化为:

$$\begin{aligned} D_1 &= \inf \theta r + \frac{1}{|\Omega|} \sum_{\omega \in \Omega} \pi_\omega \\ \text{s.t. } \pi_\omega &\geq \hat{\xi}_j(\mathbf{x}, \hat{\mathbf{s}}_\omega, \hat{\mathbf{t}}_\omega) - \beta, \quad \forall \omega \in \Omega, \\ r &\geq 1, \quad \forall \omega \in \Omega. \end{aligned}$$

因此, 最优值为: $Z_1 = D_1 = \theta + \frac{1}{|\Omega|} \sum_{\omega \in \Omega} (\hat{\xi}_j(\mathbf{x}, \hat{\mathbf{s}}_\omega, \hat{\mathbf{t}}_\omega) - \beta)^+ = \theta + \frac{1}{|\Omega|} \sum_{\omega \in \Omega} (\hat{\xi}_i(\mathbf{x}, \omega) - \beta)^+$. 代入约束 (15) 中, 结果得证.

E CVaR 约束闭式解

基于 Wasserstein 距离模糊集 $\mathcal{F}(\theta)$ 的 CVaR 表达式如下, 并可进一步转化:

$$\begin{aligned} \text{CVaR}_\varepsilon^\theta(\tilde{\xi}(\mathbf{x}, \omega)) &= \inf_{\beta \in \mathbb{R}} \left(\beta + \frac{1}{\varepsilon} \sup_{\mathbb{P} \in \mathcal{F}(\theta)} \mathbb{E}_{\mathbb{P}}[(\tilde{\xi}_i(\mathbf{x}, \omega) - \beta)^+] \right) \\ &= \inf_{\beta \in \mathbb{R}} \left(\beta + \frac{\theta}{\varepsilon} + \frac{1}{\varepsilon|\Omega|} \sum_{\omega \in \Omega} [(\tilde{\xi}_i(\mathbf{x}, \omega) - \beta)^+] \right) \\ &= \inf_{\beta \in \mathbb{R}} \left(\beta + \frac{1}{\varepsilon|\Omega|} \sum_{\omega \in \Omega} [(\tilde{\xi}_i(\mathbf{x}, \omega) - \beta)^+] \right) + \frac{\theta}{\varepsilon} \\ &= \inf_{\beta \in \mathbb{R}} \left(\beta + \frac{1}{\varepsilon} \mathbb{E}_{\mathbb{P}}[(\tilde{\xi}_i(\mathbf{x}, \omega) - \beta)^+] \right) + \frac{\theta}{\varepsilon} \\ &= \text{CVaR}_\varepsilon[\tilde{\xi}_i(\mathbf{x}, \omega)] + \frac{\theta}{\varepsilon}. \end{aligned}$$

基于 CVaR 的定义公式得到其闭式解形式如下:

$$\text{CVaR}_\varepsilon[\tilde{\xi}_i(\mathbf{x}, \tilde{\mathbf{s}}, \tilde{\mathbf{t}})] = \sum_{\omega=1}^{\lfloor \varepsilon|\Omega| \rfloor} \frac{\hat{\xi}_i(\mathbf{x}, \omega)}{\varepsilon|\Omega|} + \left(1 - \frac{\lfloor \varepsilon|\Omega| \rfloor}{\varepsilon|\Omega|} \right) \hat{\xi}_i(\mathbf{x}, \lfloor \varepsilon|\Omega| \rfloor + 1),$$

其中定义 $\sum_{\omega=1}^0 \hat{\xi}_i(\mathbf{x}, \omega) = 0$ 和 $\hat{\xi}_i(\mathbf{x}, |\Omega| + 1) = 0$.

代入得到 $\text{CVaR}_\varepsilon^\theta(\tilde{\xi}_i(\mathbf{x}, \omega))$, 结果得证.

F 分支切割算法

算法 1: 分支切割算法

输入: 数据文件, QC 和 YC 的作业序列, 任务集合, 历史行驶时间, 算法参数

输出: AGV 作业序列 x^*

```
1 松弛子回路约束和分布鲁棒机会约束得到初始模型, 用分支定界算法或者求解器得到  $x$ ;  
2 if  $x$  中存在子回路 then  
3     根据子回路情况, 向初始模型中加入子回路约束;  
4     转到 1;  
5 else  
6     根据  $x$  情况生成作业序列 paths;  
7     foreach  $p \in \text{paths}$  do  
8         依次计算路径上任务的 CVaR;  
9         foreach  $i \in C$  do  
10            if CVaR > 0 then  
11                将约束 (19) 加入初始模型;  
12                转至 7;  
13            end  
14        end  
15    end  
16    if  $x$  不可行 then  
17        转到 1;  
18    else  
19         $x^* := x$ , 输出  $x^*$ ;  
20    end  
21 end
```

G 初始解算法

设置符号定义如下: n 表示当前任务位置, C' 表示全部任务中未完成的任务集合, k 表示当前任务序列, 初始值为 1, x 表示当前所有已有序列, x_k 表示第 k 个任务序列.

算法 2: 初始解算法

输入: 数据文件, QC 和 YC 的作业序列, 任务集合, 历史行驶时间, 算法参数

输出: AGVs 的作业序列 x^*

```

1  令  $x_k = [0]$ ,  $x = x \cup x_k$  得到当前未完成的任务集合  $C'$ 
2  foreach  $i \in C'$  do
3      令  $n$  为集合  $C'$  中时间窗上界最晚的任务;
4       $x_k = [0, n]$ ,  $C' = C'/n$ ;
5  end
6  if  $C' \neq \emptyset$  then
7      令  $n$  为集合  $C'$  中时间窗下界最早的任务;
8      foreach  $x_k \in x$  do
9          foreach  $i \in x_k$  do
10             将  $n$  插入到  $i$  后面, 生成  $x'_k$ ;
11             计算  $x'_k$  的 CVaR;
12             if  $\text{CVaR} \leq 0$  then
13                  $x_k = x'_k$ ,  $C' = C'/n$ ;
14                 break, 转到 8;
15             else
16                 将  $n$  插入  $x_k$  的尾部;
17                  $k = k + 1$ ;
18                 转到 1;
19             end
20         end
21     end
22 end
23  $x^* := x$ , 输出  $x^*$ .

```

H VNS 算法

T 表示当前温度, T^k 表示初始温度, T^f 表示终止温度, δ 表示降温系数, n 表示未改进的迭代次数, H 表示邻域搜索最大个数, \mathbb{L} 表示局部搜索算子集合.

算法 3: VNS 算法

输入: 数据文件, QC 和 YC 的作业序列, 任务集合, 历史行驶时间, 算法参数
 输出: AGVs 的作业序列 x^*

- 1 根据算法 G 生成初始解 x 作为当前解; 对初始解进行初始化, 设定最优解 x^* ;
- 2 初始化索引: $T := T^k$; $n := 0$, $h := 0$, $l := 0$;
- 3 **while** $T \leq T^k$ **do**
- 4 令 $n := n + 1$, $h := h + 1$;
- 5 **while** $h \leq H$ **do**
- 6 初始化索引: $k := 0$;
- 7 **while** $k \leq h$ **do**
- 8 对 x 使用第 h 个邻域结构得到新解 x' ;
- 9 令 $k := k + 1$;
- 10 打乱随机算子;
- 11 **foreach** $l \in \mathbb{L}$ **do**
- 12 对 x' 使用当前随机算子更新当前解 x'' ;
- 13 **if** x'' 和 x 符合 Metropolis 准则 **then**
- 14 令 $x := x''$;
- 15 **if** x 优于 x^* **then**
- 16 令 $x^* := x$, $h := 0$; $k := 0$, $n := 0$;
- 17 转到第 11 行;
- 18 **end**
- 19 **end**
- 20 **end**
- 21 令 $h := h + 1$;
- 22 **end**
- 23 **end**
- 24 对最优解 x^* 进行扰动, 得到 x' ;
- 25 **if** x' 和 x^* 符合 Metropolis 准则 **then**
- 26 令 $x := x'$;
- 27 **if** x 优于 x^* **then**
- 28 令 $x^* := x$; 初始化索引: $n := 0$;
- 29 **end**
- 30 **end**
- 31 $T := T * \delta$;
- 32 **end**
