

基于云边协作的工业互联网排产方法：以钢铁热轧生产为例

丁婧伊¹, 金嘉晖^{1*}, 杨丰赫², 熊润群¹, 单 冯¹, 东 方¹

(1. 东南大学计算机科学与工程学院, 江苏南京 211189; 2. 东南大学网络空间安全学院, 江苏南京 211189)

摘 要: 随着工业互联网的蓬勃发展, 工业生产需要满足用户的个性化需求. 由于个性化产品规格多样种类繁多, 一个高效的智能排产方法对企业的生产制造尤为重要. 从部署模式来看, 现有的智能排产系统可分为企业本地部署和云排产服务两类. 本地排产系统的计算与存储资源相对有限, 难以满足精确排产算法的需求; 而云排产系统需要大量工业核心排产数据的支撑并按需计费, 计算存储与网络传输的开销使排产服务成本较高. 此外, 工业核心数据上传至云可能存在数据泄露的风险. 针对以上问题, 本文以钢铁热轧生产为例, 将边缘计算技术引入智能排产, 提出了一种云边协作的工业互联网排产框架(Production Scheduling based on Edge-Cloud-Collaboration, PSECC), 本框架在边缘端预处理原始工业数据, 保证核心生产数据保留在企业端; 在云端进行算法求解, 通过部署通用型求解算法又为框架赋予了可扩展性. 本文基于PSECC框架设计实现了针对钢铁热轧排产任务的云边分解方法, 实验证明本文提出的云边协作排产方法与常规求解器的性能相似, 但是可以避免工业核心数据上云, 且云端求解器的选择更加灵活. 在性能方面, 云排产的总排产时间是PSECC的1.4~3.7倍, 其中网络传输时间是10~15倍.

关键词: 云边协作; 工业互联网; 钢铁热轧; 作业车间调度; 个性化定制; 智能排产

基金项目: 国家重点研发计划(No.2021YFB2900100); 国家自然科学基金(No.62232004, No.62072099, No.61632008)

中图分类号: TP393

文献标识码: A

文章编号: 0372-2112(2024)09-2988-12

电子学报 URL: <http://www.ejournal.org.cn>

DOI: 10.12263/DZXB.20221018

Industrial Internet Scheduling Method Based on Cloud-Edge Collaboration: A Case Study of Steel Hot Rolling

DING Jing-yi¹, JIN Jia-hui^{1*}, YANG Feng-he², XIONG Run-qun¹, SHAN Feng¹, DONG Fang¹

(1. School of Computer Science and Engineering, Southeast University, Nanjing, Jiangsu 211189, China;

2. School of Cyber Science and Engineering, Southeast University, Nanjing, Jiangsu 211189, China)

Abstract: With the rapid development of the industrial Internet, industrial production needs to satisfy personalized user requirements. Due to the wide variety of personalized product specifications, an efficient and intelligent scheduling method is particularly important for manufacturing enterprises. From the perspective of deployment mode, existing intelligent scheduling systems can be divided into two categories: enterprise on-premises deployment and cloud scheduling services. The computing and storage resources of the local scheduling system are relatively limited, making it difficult to meet the needs of accurate scheduling algorithms. In contrast, cloud scheduling systems require the support of a large amount of industrial core scheduling data and charge on demand. The overhead of computing, storage, and network transmission makes scheduling service costs high. Additionally, uploading core industrial data to the cloud may carry the risk of data leakage. To address these issues, this paper takes the hot rolling production of iron and steel as an example, introduces edge computing technology into intelligent production scheduling, and proposes a cloud-edge collaborative industrial internet production scheduling framework (PSECC). The framework preprocesses the original industrial data at the edge to ensure that core production data is kept at the enterprise end, while the algorithm is solved in the cloud. The framework is also extended by deploying a general-purpose algorithm. Based on the PSECC framework, we designed and realized a cloud-edge decomposition method for hot rolling production scheduling tasks in steel. Experiments show that the performance of the cloud-edge

collaborative production scheduling method proposed in this paper is similar to that of the conventional solver, but it can avoid uploading industrial core data to the cloud, and the choice of cloud solver is more flexible. In terms of performance, the total scheduling time of cloud scheduling is 1.4 to 3.7 times that of PSECC, and the network transmission time is 10 to 15 times..

Key words: cloud-edge collaboration; industrial Internet; hot steel rolling; job shop scheduling; personalized customization; intelligent production scheduling

Foundation Item(s): National Key R&D Program of China (No.2021YFB2900100); National Natural Science Foundation of China (No.62232004, No.62072099, No.61632008)

1 引言

工业互联网是实现新一轮工业革命的核心,其发展推动着制造业,特别是个性化定制生产的转型升级,个性化定制是按照用户的需求,在有效时间内提供个性化、定制化的产品,具有多品种、小批量、高时效性、紧急程度多样的特点,并广泛存在于机械设计制造、电子产品制造、钢铁生产等各种工业制造领域. 工业生产计划(即排产)通过寻找一组操作序列,在满足约束条件下优化最大完工时间或其他性能指标. 由于实际的工业生产场景通常具有多个生产指标和约束条件,制定精细的生产计划可以帮助企业解决个性化定制问题^[1].

本文以钢铁热轧为例讨论实际生产调度问题. 热轧阶段的生产流程如图1所示,首先将原始板坯送入加

热炉,将其加热到适当的变形温度,然后送入热轧机通过粗轧和精轧逐渐对板坯施加压力,使其尺寸规格逐渐符合订单的要求. 热轧排产系统接收板坯的长度、宽度、出炉温度等属性,再根据工艺要求求解最优的板坯加工顺序. 钢铁热轧过程可以描述为以下问题:每个板坯相当于一个作业,分别通过加热炉、热轧机两个阶段的操作,每个阶段有若干台可并行处理的机器,所有板坯在流水线上顺序处理. 钢铁热轧通常需要轧制一批长度、宽度、出炉温度各不相同的板坯,由于轧制不同属性的板坯需要调整生产系统(Manufacturing Execution System, MES)参数并带来一定开销,所以热轧生产调度成为钢材生产中最核心的步骤之一. 钢铁热轧排产的目标是找到一组板坯的生产序列,使各相邻板坯的属性过渡相对平滑,以提升生产效率、减少等待时间和减轻设备磨损.

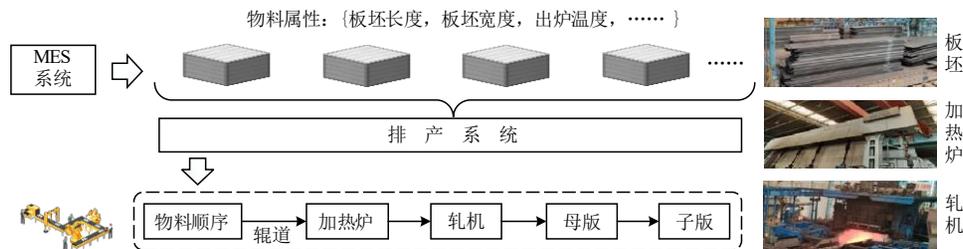


图1 钢铁轧制工艺排产流程

钢铁热轧排产可看作流水车间任务调度问题^[2],现有的相关排产系统可分为本地排产系统^[3]和云排产系统^[4]两类. 本地排产系统将算法部署在企业本地,由企业负责系统的部署、维护与升级,相比于人工极大地减少了排产时间,更加高效合理地制定生产计划. 然而排产问题属于组合优化问题,精确的排产算法通常具有较高的时间和空间复杂度^[5]. 而企业本地既缺少高效的算法求解库又面临高昂的计算成本问题,无法满足精确求解需求. 云排产系统上传排产任务到云平台进行求解,更高效地获得优质的排产结果,例如华为云商店(marketplace.huaweicloud.com)为众多中小型企业提供排产服务. 与本地排产相比具有更加强大的计算能力和更加丰富的求解算法. 然而,由于每个工厂各自的

生产流程和需求不同,云排产系统通常根据不同的排产需求设计不同的求解方案和应用. 例如,在钢铁制造行业中,不同车间的产品线流程以及属性约束、优化目标有所区别,因此存在不同的建模和求解方法,导致在云端重复构建排产应用^[6]. 另外,云排产需要将企业具体的生产数据直接上云,这存在核心排产数据泄露的风险. 在传输数据量方面,除了物料数据需要上云,包括生产设备实时数据、物料库实时信息等关联模块的数据也同时需要上云,若涉及到采样视频数据则其传输量将以GB为单位计数,这需要较长的执行时间和大量的传输数据,导致排产服务的成本较高.

近年来,云边协作技术的出现很大程度上弥补了本地排产和云排产的上述缺陷^[7]. 云边协作是一种以云计算

为主要算力、边缘计算进行预处理与部分计算的模式,通过给予网络边缘设备一定的计算能力^[8]和存储能力^[9],将时延敏感的数据处理与分析任务下放至网络边缘处进行计算,以降低时延、提升服务质量^[10].这种云边协作模式在医疗系统^[11]、多媒体视频处理^[12]、能源管理^[13]等各领域都有了一定的成果.虽然近几年出现了很多基于云边协作模式的系统架构和相关应用,但是现有工作中针对工业互联网的云边协作排产方法的研究较少^[14].

云边协作排产框架构建的主要难点在于如何设计新的排产算法,即如何分配云端和边缘端的计算任务,以实现快速响应和精确分析.现有的排产算法由于在不同的生产场景下约束条件不同且较为复杂,领域专家需要根据实际场景设计特定的算法来求解.而现有的基于边缘计算的排产方法大多数将排产算法作为一个整体部署在云端,原始数据在边缘通过预处理后,交给云平台的特定应用求解,此方法由于在不同的场景下在云端构建特定应用而缺乏可复用性;另一种排产方法是将原始数据上传到云后,云平台将任务卸载到各个边缘端求解,云平台会收到排产相关的各类核心参数,从而存在核心数据泄露的风险.与现有研究工作不同,本文关注如何分解排产算法,将排产任务可分为两阶段:(1)在边缘端预处理原始工业数据,保证核心生产数据保留在企业端的需求;(2)在云端进行算法求解,通过部署通用型求解算法为框架赋予可扩展性,在工序类似的不同生产场景下都可以复用同一套云端排产服务.本文以钢铁热轧排产为例,针对云边协作场景下的车间调度问题开展研究,提出了一种面向云边协作的新型排产框架,并将该框架用于钢铁热轧排产过程.具体而言,本框架设计了一种面向钢铁热轧排产的云边任务分解方法,实现了排产模型,并讨论了排产模型在实际场景下的有效性.

本文的主要工作包括:(1)提出了一种云边协作排产框架,通过构建边缘预处理模块与云端通用求解模块,实现了排产模型的解耦合;(2)设计了针对钢铁热轧排产场景下的云边协作方法,利用云边协作排产框架,分解相应的排产算法,并优化中间结果的计算过程,降低网络传输开销、提升排产效率;(3)设计实现了基于车间调度问题的云边协作排产模型,并采用真实的钢铁集团真实数据集,对排产任务执行流程以及结果性能指标进行测试,验证了该模型在实际场景下的有效性.

2 相关工作

边缘计算通过赋予网络边缘设备一定的计算与存储能力,在边缘端执行计算与联网任务^[15,16].而边缘计算的体系结构在智慧城市^[17]、智慧能源^[18]、智慧农业与工业互联网^[19]等基于物联网(Internet of Things, IoT)的

场景中发挥了十分重要的作用,得以最大程度地减少延迟、改进隐私保护并减少带宽与相关成本.由于终端设备产生的大量数据无需全部上传至云端计算,而是首先在网络边缘进行过滤与预处理,因此,与云计算模式相比,其服务更快、响应时间更短^[20],即使在断网的情况下也可以本地执行计算任务.

目前在边缘计算与工业互联网的结合领域已经有相当多的研究与应用,比如:结合边缘计算构建工业设备监控系统^[21,22],将其用于在实际生产中集中监控和管理众多设备、基于边缘计算评估设备能耗,构建系统级电池模型^[23]、基于边缘计算的智能计算框架^[24],将不同流量准确发送给边缘服务器与云计算中心,以实现计算卸载优化、面向工业互联网的实时需求的基于边缘计算的混合计算框架^[25],以及基于边缘计算构建工业机器人系统^[26]等.

在基于云边协作的排产方法方面,有一些工作提出了用于解决资源分配、质量监测等与作业车间调度问题相关的云边协作生产调度框架.在传统的方法中,所有的任务都传输到云端计算,传输时延无法满足生产调度的实时性要求,为此 Afrin Mahbuba 等人^[27]提出了一种基于云边协作的框架来处理智能工厂中机器人工作流的资源分配问题,以克服纯云端框架执行时延敏感任务时的局限.类似的, Li Xiaomin 等人^[25]为满足车间调度的实时性要求,提出了一种在边缘端执行计算资源调度的两阶段算法的云边端混合框架. Feng Yixiong 等人^[28]关注汽车轴承生产场景下的产品质量管控问题,提出了一种基于云边协作的轴类产品控制体系,在边缘接入设备和传感器并处理边缘数据,在云平台实现多源异构数据的存储和管理. Yuanjun Laili 等人^[7]提出了面向云辅助边缘计算和边缘辅助云计算的两种云边协作模式的任务调度模型,任务分配的主要思路是根据任务和相关终端设备的通信开销,将任务分配给云服务器或边缘服务器.

虽然目前与车间调度问题相关的排产方法已有大量的相关工作,但是现有的大多数方法分别处理算法模型和云边协作框架,模型往往完整部署在云端或者边缘端而没有针对框架进行合理分解,导致不同业务场景下的车间调度问题要分别在云端或边缘端建模,缺乏可复用性.如何设计有效的云边协作策略从而实现云端和边缘端的任务分解,以适应工业互联网场景下的排产任务,成为亟需解决的问题.此外,以往工作往往聚焦关于质量监测、资源和任务调度方面,对于一个具体的排产场景如何分解模型缺乏充分的探讨^[29].本文以钢铁热轧排产为例,研究序列依赖的流水车间任务调度问题,提出了一种云边任务分解方法:在边缘端生成统一的标准化中间结果,在云端实现高精度求

解. 通过中间结果的设计, 将工业原始数据特征与通用型求解方法分离开, 从而避免将排产数据上传到云端所导致的企业核心数据泄露的风险.

3 问题建模与排产框架

本文分析钢铁热轧生产中的生产瓶颈, 并根据具体的板坯属性和生产开销构建两阶段的钢铁热轧排产模型. 第一阶段将满足工艺要求的程度建模为板坯属性模型, 第二阶段基于综合属性差异值矩阵进行模型求解. 基于排产模型设计云边协作的排产框架 PSECC (Production Scheduling based on Edge-Cloud-Collaboration), 在边缘服务器对排产数据做预处理, 在云端使用通用算法进行求解, 由此在保证算法求解性能的同时, 避免了企业核心数据上云. 本框架可以应用到不同场景的作业车间调度问题 (Job Shop Scheduling, JSP) 中, 面对新的场景, 只需要调整边缘个性化预处理模块 (如 $\delta_k()$ 函数) 就能够较好地提取适用于当前场景的工艺约束条件, 并生成统一形式的中间结果数据上传至云端进行求解, 无需修改云端的通用求解模块.

3.1 背景描述

本文以某钢铁厂的宽厚板厂为例对钢铁热轧排产问题进行分析. 在实际的钢铁生产流程中, 一组原料钢坯需要按照个性化的生产订单进行轧制, 生产的钢板并需满足钢板长度、宽度、出炉温度、完成时间等约束条件. 钢铁轧制过程包括“炼铁—炼钢—连铸—热轧”等工序, 前两道操作将原料转化为液态铁水, 连铸将液态钢转化为连铸坯 (通常存储在板坯库), 热轧将连铸坯打造成订单需要的钢板, 最后经过矫正、探伤、切割等成为成品板坯出售. 宽厚板热轧生产调度关键在于编制合理的“辊期计划”, 辊期计划是根据轧辊的使用周期编排的板坯轧制序列, 辊期计划的生产结果直接决定了产品质量和生产效益, 合理的辊期计划可以提高生产效率、降低能耗和成本、提高企业的竞争力, 而不合理则会导致生产组织不利和整个热轧生产线物流不平衡. 以加热炉加热和热轧机轧制两道工序为例:

(1) 在加热炉加热部分, 加热炉将连铸坯加热到均匀的、适合轧制的温度. 当板坯加热到理想的温度时, 按照先进先出的顺序退出加热炉并以相同顺序送往热轧机. 在实际生产工艺要求中, 每块板坯必须达到其出炉温度才能进入下一道工序, 并且硬性要求相邻两块板坯之间的出炉温度偏差在 40°C 之内. 不同的出炉温度会影响能源利用效率, 如果相邻两块板坯的出炉温度差异过大, 会给机器带来较大的用电损耗, 增加额外的生产开销^[30,31]. 因此, 相邻板坯的出炉温度在同一区间内是一个重要的条件.

(2) 在热轧机轧制部分, 将加热后的板坯用轧辊轧

制, 使其达到预期的宽厚. 每个辊期计划分为预热阶段和主体阶段, 预热阶段的板坯宽度快速从窄到宽, 主体阶段则逐渐从宽到窄. 这是由于连续两块输入板坯的宽厚和出炉温度不同, 会给轧辊机器带来磨损, 这些磨损会减少轧辊的使用寿命, 并减少加工的板坯数量. 不合理的板坯轧制顺序不仅会影响轧辊的生产效率, 还会在板坯上留下痕迹, 降低产品质量.

3.2 问题建模

根据以上钢铁热轧生产过程, 本文的排产问题定义如下. 定义输入工业数据 $X = (\mathbf{x}_1, \mathbf{x}_2, \dots, \mathbf{x}_N)$ 表示集合中包含 N 块板坯. 每块板坯的断面类型、重量、轧制宽度、轧制厚度、交货期、在炉时间、出钢温度、板坯位置、是否热送热装等属性均已知. 排产结果应满足若干属性约束条件, 比如相邻板坯之间的宽厚跳跃过渡越小越好、相邻板坯需要有相似的出炉温度等. 定义属性约束集合为 A , 包含 $|A|$ 个属性约束条件, 参数向量 $\alpha = (\alpha_1, \alpha_2, \dots, \alpha_{|A|})$ 分别对应各项属性约束的权重, $\mathbf{x}_i = (x_i^1, x_i^2, \dots, x_i^{|A|})$ 表示板坯 i 包含 $|A|$ 个属性值, 其中 x_i^k 是板坯 i 的属性 k 的值, P 表示排产的辊期数量.

热轧计划调度问题是满足多个约束条件的多目标优化问题, 目标是找到使相邻板坯各属性间的差异最小的生产序列. 考虑到不同的属性约束条件之间可能存在冲突, 本文对各属性加权求和获取“综合属性差异值”, 目标为最小化相邻板坯间的综合属性差异, 由此将多目标优化问题转化为单目标优化问题. 该问题归类为流水车间调度问题的一种. 对于本文解决的钢铁热轧排产问题而言, 只关注一个辊期内的板坯生产情况, 每块板坯就是一个待加工的工件, 将 N 块板坯按照相同的加工顺序依次经过加热炉、粗轧机和精轧机, 经过不同的机器需要不同的时间, 但目标是建立差异最小的生产序列. 而由于实际钢铁生产场景采用流水线生产方式, 板坯之间的加工顺序一旦确定就不会改变, 因此将钢铁热轧排产看作流水车间调度问题的一个特例. 此问题具有一定的通用性, 可适用于宽厚板、中厚板等不同板材. 虽然板坯的属性不一样, 但是可以通用本模型求解. 具体而言, 本文为钢铁热轧排产问题建立优化模型如下:

$$f(X, \alpha) = \min_{y_{i,j,r}} \left(\sum_{r=1}^P \sum_{i=1}^N \sum_{j=1}^N \text{cost}_{i,j} \cdot y_{i,j,r} \right) \quad (1)$$

式(1)表示优化目标, 即最小化所有辊期计划相邻板坯间综合属性差异值的生产序列. $f(X, \alpha)$ 表示针对某工艺的排产任务, 输入原始工业数据 X 和参数向量 α 后, 将所有的板坯分配到 X 个辊期计划进行排产. 存在以下约束条件:

$$\text{cost}_{i,j} = \sum_{k \in A} \alpha_k \cdot \text{cost}_{i,j}^k + \sum_{k \in A} \alpha_k \alpha_k \cdot \text{cost}_i^k \quad (2)$$

$$\text{cost}_{i,j}^k = \delta_k(x_i^k, x_j^k) \quad (3)$$

$$\text{cost}_i^k = \delta_k(x_i^k) \quad (4)$$

$$y_{i,j,r} = \begin{cases} 1, & \text{辊期 } r \text{ 中, 板坯 } j \text{ 紧跟在 } i \text{ 后轧制} \\ 0, & \text{辊期 } r \text{ 中, 板坯 } j \text{ 不紧跟 } i \text{ 后轧制} \end{cases} \quad (5)$$

$$\sum_{r=1}^P \sum_{j=1}^N y_{i,j,r} = 1, \forall i \in V \quad (6)$$

$$\sum_{r=1}^P \sum_{j=1}^N y_{i,j,r} = 1, \forall j \in V \quad (7)$$

其中,式(2)表示任意两块板坯*i,j*之间的各项属性差异值, $\text{cost}_{i,j}^k$ 也称为板坯*i,j*之间的“综合属性差异值”,表现为相邻板坯之间各属性差异的加权和, α_k 表示当前属性*k*的权值;式(3)和(4)表示板坯*i,j*在属性*k*上的差异程度, $\delta_k()$ 表示用户自定义的惩罚函数, $\delta_k(x_i^k, x_j^k)$ 表示两块板坯之间的惩罚函数,既可以是板坯*i*和板坯*j*差的绝对值形式,也可以是用户自定义的惩罚表或惩罚函数, $\delta_k(x_i^k)$ 表示单块板坯的惩罚函数. 综合属性差异值 $\text{cost}_{i,j}^k$ 取决于板坯间各项属性差异值 $\text{cost}_{i,j}^k$ 和选择第*i*块板坯的代价值 cost_i^k , 属性完全相同的两块板坯最适合安排在相邻位置生产,综合属性差异越大则工艺满足程度越低;式(5)表示如果板坯*j*在板坯*i*后的相邻位置轧制,就会产生综合属性差异值并将其纳入最小化优化目标中进行计算,否则不会纳入计算;式(6)表示每个作业在生产计划中具有唯一性,需要让每个作业仅为除自身以外的任意作业的前驱节点. 式(7)表示为保证每个作业在生产计划中的唯一性,需要让其仅为除自身以外的任意作业的后继节点.

3.3 面向云边协作的两阶段排产算法

钢铁热轧排产问题通常可以使用启发式算法或者改进的元启发式算法求解^[30,32,33],但是这些求解方法难以分解为云边协同模式. 此外,现有排产模型的构建和求解都与企业信息相关,一般需要获得全部企业排产数据,因此存在数据泄露的风险. 本文的目标是设计适合云边协同的求解方法,边缘端根据不同企业的需求,生产与原始数据无关的中间结果(带权有向完全图),云端使用通用算法求解器求解. 云端求解算法与应用无关,不同的边缘建模,因此具有灵活性. 由于中间结果是一个抽象语义的图形式,每条边都是板坯属性差异和权重的加权和,在云端根据图模型通用求解,不涉及原始数据,保证了排产数据隐私安全.

为了满足云边协作的需求,本文将钢铁热轧排产问题转换为经典的车辆路径问题(Vehicle Routing Problem, VRP). 车辆路径问题是旅行商问题的扩展形式. 给定*N*个城市以及任意两个城市间的距离,一个车队有*P*辆货车,从一个仓库出发为各个城市供货. 已知每个城市都有自身固定的需求量,需要为车辆寻找一条

路线能够使得车辆访问各城市一次且仅访问一次,并且在保证各城市的需求量的前提下尽可能使得车辆所经过的路径最短. 由于车辆路径问题与作业车间调度问题可相互转换^[34],且在汽车零部件制造^[35]、焊装车间^[36]、药品制造^[37]等领域的排产求解都能利用作业车间调度问题建模,因此本求解方法可应用于多种排产领域. 为此,我们设计了两阶段算法,首先个性化预处理,将原始工业数据*X*转换为可应用于VRP模型的有向完全图*G*,然后使用通用方法求解模型,如算法1所示.

算法1 两阶段排产算法

输入:板坯集合*X*,属性约束集合*A*,属性约束权重*α*,惩罚函数 δ_k

输出:有向完全图*G*

```

1://阶段一:个性化预处理
2://初始化每块板坯的属性约束值
3:FOR EACH i IN X DO
4:  FOR EACH k IN A DO
5:     $x_i^k \leftarrow$  板坯 i 在属性 k 上的值
6:  END FOR
7:END FOR
8://计算两块板坯之间的综合属性差异值
9:初始化 C 为大小为  $N \times N$  的二维矩阵,每个元素取值为 0
10:FOR EACH pair(i,j) IN X DO
11:  FOR EACH k IN A DO
12:    IF 属性 k 是相邻板坯的属性 THEN
13:       $\text{cost}_{i,j} \pm \alpha_k \cdot \delta_k(x_i^k, x_j^k)$ 
14:    ELSE IF 属性 k 是单块板坯的属性 THEN
15:       $\text{cost}_{i,j} \pm \alpha_k \cdot \delta_k(x_i^k)$ 
16:    ENF IF
17:  END FOR
18:   $C[i,j] \leftarrow \text{cost}_{i,j}$ 
19:END FOR
20:V  $\leftarrow$  nodes in X
21:E  $\leftarrow C$ 
22:构建 VRP 模型所用的带权有向完全图  $G = (V, E)$ 
23://阶段二:VRP 问题求解
24:Y = VRP(G)
25:RETURN Y

```

算法1中,在个性化预处理阶段,第1步是初始化工业输入数据,即对每块物料的所有属性值进行初始化. 例如,板坯的属性可能包括目标宽度、成品厚度、坯厚、出炉温度、实际重量、堆放位置等,而每个属性的重要程度和优先级不同,这一步需要设定属性的优先级权重,越重要的属性设定越高的权重. 第二步是针对每个板坯属性,确定相邻板坯的代价映射与计算,即相邻板坯对某一属性约束的惩罚情况,综合考虑所有属性约束代价之后求得该相邻板坯对的综合评价得分. 这

一步需要遍历所有有序板坯对 $\text{pair}(i,j)$ 和属性约束集合 A , 使用惩罚函数 $\delta_k()$ 计算相邻板坯对之间的综合属性差异值 $\text{cost}_{i,j}$. $\text{cost}_{i,j}$ 通过计算相邻板坯属性差异值和单块板坯属性差异值之和得到. 由此构建一个有向完全图 $G=(V,E)$, 点集 V 表示所有板坯的集合, 边集 E 的取值表示 $\text{cost}_{i,j}$.

在模型求解阶段, 输入一个带权有向完全图 $G=(V,E)$, 点集 $V=\{\text{Node}_1, \text{Node}_2, \dots, \text{Node}_N\}$ 表示板坯的集合, 每个点表示 VRP 中的一个城市, 边集 $E=\{\text{Pair}_{1,2}, \text{Pair}_{1,3}, \dots, \text{Pair}_{N,N-1}\}$ 表示两块相邻板坯顺序的集合. 每条边代表了城市间的运输成本, 由两块板坯之间的综合属性差异值(如成品厚度差异、目标宽度差值等)和前一个板坯自身的属性惩罚值(如是否热装、在炉时间等)之和表示, 连接同一点对的两条边取值往往不同. 每块板坯的重量设为 VRP 中每个城市的需求, 辊期计划的数量对应货车数量 P , 每个辊期计划最多能轧制的板坯总重量对应于每辆车的容量. 目标是在满足所有城市需求且不违反车辆容量限制的前提下, 最

小化车队的总成本. 此外, 当板坯生产的时间范围受限时, 可加入时间因素建模为带时间窗的车辆路径问题 (Vehicle Routing Problems with Time Windows, VRPTW) 等. 在实际钢铁生产中, 宽厚板厂中厚板、中宽带钢等产品的生产具有相似工序, 但钢材的约束函数和差异惩罚函数不同, 利用个性化预处理, 可根据产线需求和钢材的属性约束构建其属性差异惩罚值函数, 继而可对排产问题建模, 并在下一阶段基于 VRP 模型求解. 车辆路径问题的求解方法包括分支限界法、分支切割法、集合涵盖法等精确求法, 也有模拟退火法、禁忌搜索法、遗传算法等元启发式求解算法, 云端可以灵活选择不同算法.

3.4 云边协作的排产框架

基于两阶段排产算法, 本文设计云边协作的排产框架 PSECC, 如图 2 所示. 其中, $f(X, \alpha)$ 表示完整的排产任务, $g()$ 表示各个工厂在边缘端设立自己的预处理模块, 生成中间结果 G 并传输给云端通用求解器 $h()$ 求解, 且云端不知道所上传数据的具体语义信息.

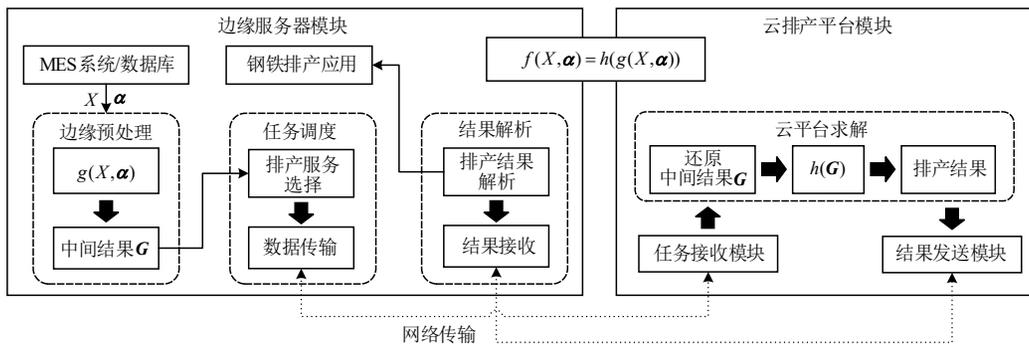


图2 云边协作的排产框架

变量的符号和含义如表 1 所示. PSECC 将完整的排产任务 $f(X, \alpha)$ 分解为边缘与云两个模块, 在边缘服务器设计合理的个性化预处理模块, 并构建能够直接与云平台交互的中间结果 G ; 在云排产平台设计合理的函数模块 $h(G)$, 根据接收到的中间结果 G 还原出所需数据, 并计算排产结果.

对于车间调度问题而言, 实现云边协作的关键在于如何将问题分解成以上模块, 在边缘预处理原始数

据, 构建适合的中间结果, 在云端设计排产求解模块, 计算满足优化目标的排产序列. 由此, 构建云边协作的排产框架范式如式(8)所示:

$$f(X, \alpha) = h(g(X, \alpha)) \quad (8)$$

其中原始数据 X 与参数向量 α 是整个排产任务的输入起点, X 描述了所有待处理作业的原始属性值, α 是对应于排产逻辑各项属性约束的权重系数, $g(X, \alpha)$ 表示由边缘服务器自定义的个性化预处理模块. 在边缘服务器处经过个性化预处理模块后, 将排产任务转化成为标准形式的中间结果 G , 其表示如式(9)所示:

$$G = g(X, \alpha) = \begin{pmatrix} \infty & \text{cost}_{1,2} & \text{cost}_{1,3} & \dots & \text{cost}_{1,N} \\ \text{cost}_{2,1} & \infty & \text{cost}_{2,3} & \dots & \text{cost}_{2,N} \\ \text{cost}_{3,1} & \text{cost}_{3,2} & \infty & \dots & \text{cost}_{3,N} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ \text{cost}_{N,1} & \text{cost}_{N,2} & \text{cost}_{N,3} & \dots & \infty \end{pmatrix} \quad (9)$$

表 1 云边协作的排产模型符号表

变量符号	含义
X	表示工业原始数据
α	表示用于提取工业信息的参数向量
G	表示边缘服务器构造的中间结果
$f(X, \alpha)$	表示针对某工艺的排产任务
$g(X, \alpha)$	表示边缘个性化预处理模块
$h(G)$	表示云排产平台求解模块

式(9)可以进一步转换为图3所示的有向完全图,将任意作业视为节点,连接节点的有向边表示两块相邻板坯的顺序,任意两个节点之间用权值不同的双向边连接,表示任意两块板坯之间的综合属性差异值.在此基础上可以设置连接到所有作业的超级源点 s 与超级汇点 d .构造出的中间结果 \mathbf{G} 是不带有语义信息.

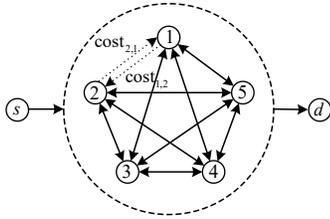


图3 排产中间结果对应的图结构(即中间结果 \mathbf{G})

中间结果 \mathbf{G} 是可直接用于排产算法求解且不直接包含板坯真实属性信息的标准形式数据,可表示为所有 $\text{cost}_{i,j}$ 组成的矩阵,矩阵中的每个元素表示任意两块板坯之间的综合属性差异值.云边双方通过中间结果 \mathbf{G} 进行信息传递,保证了原始工业数据安全.云排产平台仅通过标准的中间结果 \mathbf{G} 即可进行通用型的求解,如式(10)所示.

$$f = h(\mathbf{G}) \quad (10)$$

中间结果 \mathbf{G} 的构建是云边协作排产框架的核心,它直接负担着云边通信的任务,其每个元素都由原始数据 X 和参数向量 α 构建,而相关数据都保存在边缘服务器,难以将中间结果解析回原始数据,从而保证了原始工业数据安全.为了构建中间结果 \mathbf{G} ,在边缘个性化预处理模块 $g(X, \alpha)$ 提取属性信息,并且将其转化为 $N \times N$ 维的代价值矩阵,矩阵中第 i 行 j 列的非对角线元素由 $\text{cost}_{i,j}$ 计算得到.然而随着作业数量 N 的增加,中间结果 \mathbf{G} 的规模也会以平方规模增长,增加云边之间的数据传输开销.对中间结果 \mathbf{G} 进行数据压缩可以优化数据传输开销.在钢铁热轧排产中, $\delta_k()$ 往往通过差值求解,表现为板坯属性真实数值差的绝对值,矩阵 \mathbf{G} 中的非对角线元素 $\text{cost}_{i,j}$ 表示为作业 i, j 之间各项属性之差的绝对值加权,如式(11)所示:

$$\text{cost}_{i,j} = \sum_{m=1}^{|A|} \alpha_m \cdot |x_i^m - x_j^m| \quad (11)$$

此时,可以首先将每个作业的属性根据参数向量 α 单独提取出来后,再进行代价值的计算,令 $\tilde{X} = (\alpha_1 \cdot x_i^1, \alpha_2 \cdot x_i^2, \dots, \alpha_{|A|} \cdot x_i^{|A|})$ 表示作业 i 的综合属性值,函数 $\text{abs}()$ 表示对于向量中的每一个元素做取其绝对值的操作,由此可得式(12).各属性与参数向量 α 相乘,与原始值产生了差异,一定程度上保护了数据隐私.

$$\text{cost}_{i,j} = \text{abs}(\tilde{X}_i - \tilde{X}_j) \cdot \begin{pmatrix} 1 \\ 1 \\ \vdots \\ 1 \end{pmatrix} \quad (12)$$

用一维向量 $(\tilde{X}_1, \tilde{X}_2, \dots, \tilde{X}_N)$ 构建出中间结果 \mathbf{G} , \mathbf{G} 中的非对角线元素 $\text{cost}_{i,j}$ 可以由式(12)计算得到.至此可实现将 $N \times N$ 的矩阵数据转化为 $1 \times N$ 的向量,进一步减少了工业上传数据,云排产平台仅根据任意 \tilde{X}_i, \tilde{X}_j 的值就可计算代价值,并将该向量还原为 $N \times N$ 的矩阵数据,由此实现中间结果的传输优化.

当属性差异使用差值求解时可使用式(12)计算,但是更通用的情况下需要根据属性特征构造惩罚函数,此时可以采用矩阵奇异值分解(Singular Value Decomposition, SVD)算法,将中间结果矩阵 \mathbf{G} 分解为若干矩阵后,通过保留部分奇异值进行有损压缩,将其上传至云平台后恢复出近似代价值的矩阵进行排产算法求解.虽然矩阵奇异值分解存在有损压缩从而损失部分精度,但是其带来的误差较小,且SVD算法处理中间结果矩阵还能进一步保护用户数据安全^[38].

4 实验与分析

在本节中,使用现实世界中某钢铁集团在工业排产时所使用的真实数据集来验证PSECC排产框架的有效性,主要关注以下几点研究问题:

问题1, PSECC的排产结果是否有效?

问题2, 和其他云边协作排产相比是否高效?

问题3, PSECC排产框架的执行性能如何?

4.1 实验设置

本实验使用某钢铁集团所提供的2019年6月份的真实板坯数据,从中取样并构成数据量为300、500、800、1 100、1 500、1 700、2 000块板坯的七个对比数据集.本文使用开源优化算法包Google Optimization(OR-Tools)求解寻路问题.本实验使用边缘服务器的操作系统为Windows 10, CPU型号为11th Gen Intel(R) Core(TM) i7-11800H @ 2.30 GHz, 内存大小为16 GB, 硬盘为SSD 512 GB, 分别选择华为云数据中心的部分节点和东南大学本地节点充当远程云排产平台.华为云弹性云服务器的配置为2核4 GB, 操作系统为Ubuntu 20.04 server 64 bit, 带宽为2 Mbps, 实际测试的入网速度平均值为2.61 MB/s, 出网速度平均值为269.3 KB/s.使用本地局域网服务器, 测试其实际入网速度为平均值为10.22 MB/s, 出网速度平均值为5.96 MB/s.

4.2 PSECC的排产结果分析(问题1)

钢铁热轧排产中根据轧制目标的不同可分为多个子任务, 比如宽厚板坯、中厚板坯、窄厚板坯、宽薄板坯等生产任务, 每种生产任务的属性约束条件及其权重

存在差异。例如,当轧制宽薄板坯时,约束条件包括目标宽度约束、板坯厚度约束、成品厚度约束、出炉温度共同区间约束、同钢种优先约束等,权值最高的约束为板坯厚度约束和同钢种优先约束。当轧制宽厚板坯和中厚板坯时,则无需考虑板坯厚度约束和成品厚度约束,且权重最高的是目标宽度约束。不同约束需求的生产任务都可以使用上述模型和PSECC框架求解,下面以轧制宽厚板坯为例进行测试分析。

为验证PSECC框架求解钢铁热轧排产问题的合理性,本文使用线性规划求解器Lingo直接求解式(1),并与PSECC框架在云端采用基于贪心的启发式求解(PSECC-Heuristic, PSECC-H)、禁忌搜索算法求解(PSECC-Tabu Search, PSECC-TS)、模拟退火算法求解(PSECC-Simulated Annealing, PSECC-SA)的结果做比较。这四种方法的时间上限均设置为120 s,得到属性约束指标的对比结果如图4所示。考虑到中间结果转换为统一的图结构,云端的通用求解器可以选择基于贪心的启发式求解算法或者禁忌搜索、模拟退火等多种元启发式算法。其中启发式求解方法是基于以下思路:在每一块当前板坯中都选取以当前板坯为起点的代价最小的板坯对所对应的下一跳板坯,直到生产计划的重量达到辊期额定重量为止。

我们从六个属性约束条件的满足情况角度分析排产结果:(1)成品厚度平均变化量是统计相邻板坯间成品厚度的平均跳变量的衡量指标,要求相邻板坯的过渡尽可能平滑,变化范围尽可能小,以保证板坯的表面质量满足工艺要求;(2)目标宽度平均变化量是统计相邻板坯间目标宽度平均跳变量的衡量指标,要求相邻两块板坯之间的宽度变化应尽量小于500 mm,最大不超过800 mm,且宽度变化越小越好;(3)平均翻堆次数是指在取板坯时平均搬动该板坯上方的板坯次数,减少翻堆次数可以减少工作人员在板坯库的操作时间,避免翻堆造成的时间浪费,从而提高生产劳动效率;(4)出炉温度保证率是指相邻两款板坯之间的出炉温度差值在允许的偏差范围内所占的比率,该比率尽可能靠近100%;(5)热装率是指在一个辊期计划中,成功热装的板坯数量占整个辊期板坯总数量的比例。热装率高的生产计划能够极大程度地减少待温时间,从而减少消耗的热力资源;(6)同钢种平均分布次数是任意钢种在排产计划中平均出现的次数。该约束值越小则说明同钢种分布的越集中,工艺满足程度越高。上述属性约束指标的计算公式如表2所示。

图4表示四种算法在不同数据量的情况下各属性约束的满足程度。从成品厚度平均变化量和目标宽度平均变化量这两个数值尽可能低的指标来看,Lingo计算出成品厚度平均变化量在22~34 mm之间,另外三种

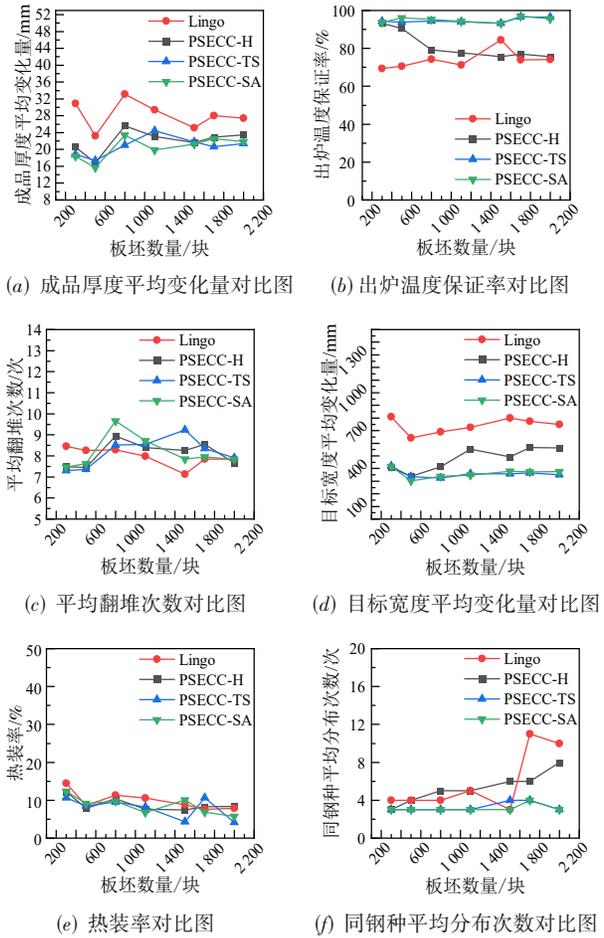


图4 各属性指标对比图

表2 钢铁热轧排产属性约束指标

指标	计算公式
成品厚度平均变化量	$\sqrt{\frac{\sum_{i=1}^{rows-1} (\text{thick}_i - \text{thick}_{i+1})^2}{rows - 1}}$
目标宽度平均变化量	$\sqrt{\frac{\sum_{i=1}^{rows-1} (\text{width}_i - \text{width}_{i+1})^2}{rows - 1}}$
平均翻堆次数	$\frac{\sum_{i=1}^{rows} \text{turning_num}_i}{rows}$
出炉温度保证率	$\frac{\sum_{i=1}^{rows} \text{temp}_i - \text{temp}_{i+1} }{rows - 1}$
热装率	$\frac{\sum_{i=1}^{rows} \text{heat_true}_i}{rows}$
同钢种平均分布次数	$\frac{\text{spatial_type_num}}{\text{total_type_num}}$

算法的平均变化量在14~26 mm之间。这是因为Lingo计算时时间较短、迭代次数较少而表现较差,而PSECC使用VRP(Vehicle Routing Problem)通用模型,使用启

发式和元启发式算法能够在较快时间得到一个较优解. 目标宽度平均变化量同样要求数值尽可能低, Lingo和PSECC-H的结果不满足要求,分别是600~800 mm和300~600 mm,只有PSECC-TS和PSECC-SA的结果满足低于500 mm的质量要求. 这是因为虽然启发式算法可以快速找到一个解,但是无法保证最优性,此时在云端可选择更合适的元启发式算法求解而无需构建特定的应用. 另一方面,平均翻堆次数越少则时间损耗越低且热装率越高则热力损耗越低,使用VRP模型和通用求解器求解的结果在两种属性约束指标上的表现差异并不明显,但是VRP模型可采用多种求解方法,所以更加灵活. PSECC-TS和PSECC-SA的出炉温度保证率在90%~100%之间,而Lingo和PSECC-H的总体保证率在70%~90%之间. 当采用启发式算法无法满足指标时,可灵活采用元启发式算法来保证在较短时间内得到较优结果,且保持各属性约束之间的平衡性. 同钢种平均分布次数越低越好,在300~1 500块板坯的数据量上使用VRP模型和通用求解器求解的结果类似,但Lingo因迭代次数短而存在误差,在1 700块以上时表现较差. 表现最好的是元启发式求解器,在实际排产时可根据指标的优先级灵活选用合适的求解器.

综上所述,PSECC在相同时间限制内与Lingo求解结果基本保持在同一个可接受的取值区间内,性能也和Lingo等求解器类似. 但是VRP求解可以采用多种求解器求解,在实际生产中可灵活采用合适的求解方法. 又由于VRP是通用模型,可以用任何适用于该模型的求解器求解,所以我们的框架不会受限于云服务提供商的约束,云服务平台无需为排产问题提供特定应用.

4.3 和其他云边协作方法的比较(问题2)

在实际排产场景中首先考虑将排产模型和求解器都放在本地服务器,数据无需传输,但是企业在每个排产场景下都要给本地服务器购买单独的排产服务. 而在工业互联网环境下,由于成本敏感的原因,相比于一次性购买复杂的排产软件,企业更希望把数据放在云端,让云端辅助排产. 因此本文只考虑和求解器在云上的方法做比较,而不考虑纯本地的方法.

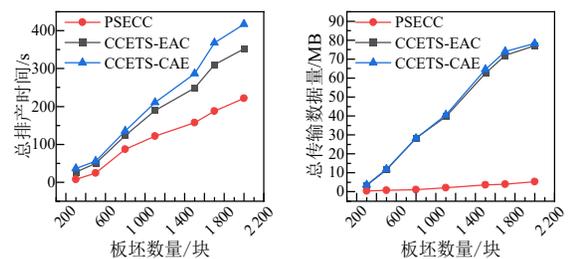
现有的求解器在云端的文章主要分为两类. 一类是云平台中心把数据卸载到边缘,边缘部署排产模型做决策^[7,27,25]. 此方法需要将所有的数据都上传到云,存在传输开销和数据隐私安全问题. 另一类是数据预处理在边缘,数据预处理后的排产模型放在云端^[7,28]. 此方法将原始数据留在边缘,但云端根据具体的排产场景部署不同的应用,因此缺乏可复用性. 不同云边协同排产模式的特征详见表3,云卸载数据到边缘和边缘上传任务到云这两种模式都面向完整的排产任务,需要在云端或边缘端部署特定的应用程序,因此这两种

方法均缺乏一定的可复用性. 对于核心数据是否上云,在云卸载数据到边缘的模式中,所有的排产数据都在云端,所以存在数据隐私安全问题;在边缘上传任务到云模式中,边缘端预处理数据,相比而言能够保护企业数据. 本文提出的PSECC排产框架设计了基于VRP的两阶段排产模型,两阶段的计算任务合理分解到云端和边缘端,第一阶段根据原始工业数据计算中间结果上云,保证核心数据留在企业本地,第二阶段灵活使用VRP求解算法,确保此框架的通用性和灵活性.

表3 云边协同排产模式的不同特征

云边协同排产模式	任务分解	核心数据	求解算法
云卸载数据到边缘 ^[7,27,25]	不可分解	上传	固定
边缘上传任务到云 ^[7,28]	不可分解	不上传	固定
PSECC	可分解	不上传	灵活

为验证云边协作排产框架的有效性,本节参考Yuanjun Laili等人^[7]提出的云边协作框架,将VRP两阶段模型分别部署在边缘端和云端,实现云把数据卸载到边缘(Collaborative Cloud-Edge Task Scheduling—Cloud-Assisted Edge, CCETS-CAE)和排产模型放在云端求解(Collaborative Cloud-Edge Task Scheduling—Edge-Assisted Cloud, CCETS-EAC)这两类云边协作模式,从总排产时间和传输数据量两方面比较PSECC与CCETS-EAC和CCETS-CAE的性能差异. 由图5可知,PSECC的总排产时间随数据量增加而稳定增长,但由于无需传输额外的板坯属性信息,其传输数据量仅仅是CCETS-EAC与CCETS-CAE的5%~10%左右. 从总的执行时间上看,数据传输量的减少导致数据传输时间降低,PSECC的总运行时间始终比CCETS-EAC和CCETS-CAE少30%~50%. 而CCETS-CAE相比于CCETS-EAC的总排产时间增加了8%~25%,这是由于边缘端的计算能力相比云端较差,其处理排产任务的时间稍长,但两者的总传输数据量相差不大,且都随着板坯数量的增加而大幅提升. 相比于上述两种方法,本文提出的基于VRP两阶段排产模型的云边协作框架有效降低了传输数据量和排产时间,减少云边通信开销.

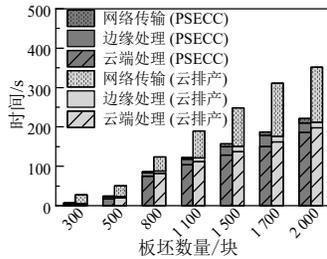


(a) 总排产时间对比图 (b) 总传输数据量对比图
图5 云边协作框架在排产时间和传输数据量方面的对比图

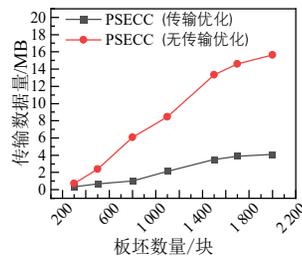
4.4 云边协作排产框架的有效性(问题3)

为验证本文提出的云边协作排产框架的性能,本节实现了基于云边协作框架 PSECC 的排产模型,比较排产流程中各个阶段所耗费的时间.云边协作排产模型在 7 个大小不同的数据集下的边缘处理时间、云端处理时间和网络传输时间如图 6(a)所示,其中总传输时间随着板坯数量的增多而稳定提升,当板坯数量达到 2 000 块时,排产的时间花销为 221.6 s.从时间占比来看,云端处理时间占总时间的 70%~90%,边缘处理时间占总时间的 10%~20%,网络传输时间仅占比 3%~10%.云端处理时间的占比最大,是因为在云端使用 VRP 求解器求解需要一定的迭代次数,消耗较多的时间得到较优的结果.而在边缘端主要做数据预处理操作,相对来说只需要较短的时间.本文的网络传输时间在 1~12 s 之间,这是由于云边协作模式主要传输数据量较小的中间结果 G ,而云边传输时间往往受到网络带宽的影响,本文使用华为云 2 MB 带宽的云服务器,实际的入网速度平均值为 2.61 MB/s,出网速度平均值为 269.3 KB/s,实验表明可以满足网络传输所需要的传输速率.

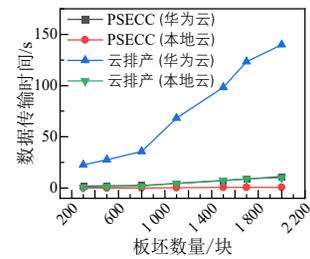
本文还测试了云排产模式下的排产性能,将排产逻辑和排产模型放到云平台进行排产,边缘端仅负责数据预处理和数据传输,其结果如图 6(a)所示.云排产的总排产时间是 PSECC 的 1.4~3.7 倍,网络传输时间是 PSECC 的 10~15 倍,这是总排产时间较长的主要原因.



(a) 排产时间对比图



(b) 传输数据量对比图



(c) 数据传输时间对比图

图 6 排产时间和传输数据量比较结果

5 结论

本文首先介绍了排产模式的发展现状以及在现有的排产模式下存在的问题,随后分析了云边协作模式对解决这些问题的有效性,提出一个基于云边协作的车间调度排产方法.本文的主要工作是提出了一个云边协作排产框架,并针对钢铁热轧排产应用,设计了云边任务分解方法,提出了中间结果的传输优化机制,实现了面向钢铁热轧排产的云边协作原型模型.本文的未来工作将研究云边协同排产模型的通用性,将模型应用于更多排产场景.此外,还将围绕云边协作框架开展任务调度、资源分配、传输优化等方面的研究工作,

分析可知,云边协作排产将大量的工业原始数据转换为图形式的中间结果 G ,相比于云排产传输大规模工业原始数据,其传输数据量大大减少.云排产平台采用按时间和数据量收费的模式,因此本云端协作排产框架可以在一定程度上降低排产的费用和成本,且可以在一定程度上避免数据泄露的风险.

中间结果 G 的大小对云边数据传输的时间和数据量也有一定影响,本文讨论了云边协作排产模型在传输优化和无传输优化两种情况下的云边数据传输量情况.传输优化指的是差值求解板坯属性差异值时,采用一维向量构建中间结果 G .如图 6(b)所示,在无传输优化情况下总的数据传输量在 0.7~16 MB 之间,是有传输优化情况下云边数据传输量的 2~6 倍,使用传输优化方法可一定程度减少中间结果传输所需要的数据量,节省传输开销.本文分别将 VRP 云端求解器放置于华为云服务器和东南大学本地云服务器,计算云排产和云边协作排产的数据传输时间如图 6(c)所示.由于实际网络传输速率的影响,基于华为云的云排产模型数据传输时间是本地云传输时间的 14~40 倍.而基于华为云的 PSECC 模型数据传输时间是本地云传输时间的 13~40 倍,但是由于云边协作排产传输的中间结果 G 数据量较小,因此本文提出的 PSECC 模型在传输时间上不会因为云边传输速率的降低而过多增长,从而保证运行效率.

以进一步优化云边排产模型的整体执行性能.

参考文献

- [1] 罗军舟,何源,张兰,等.云端融合的工业互联网体系结构及关键技术[J].中国科学:信息科学,2020,50(2):195-220.
LUO J Z, HE Y, ZHANG L, et al. The architecture and key technologies for an industrial Internet with synergy between the cloud and clients[J]. Scientia Sinica (Informationis), 2020, 50(2): 195-220. (in Chinese)
- [2] ARISHA A, YOUNG P, Baradie M E. Job shop scheduling problem: An overview[C]//International Conference for Flex-

- ible Automation and Intelligent Manufacturing (FAIM 01). Dublin: (Technological University Dublin), 2001: 682-693.
- [3] YUAN X, CHEN Y W, LIU B, et al. Advanced planning and scheduling system based on multi-resource closed-loop management[C]//2020 IEEE International Conference on Industrial Engineering and Engineering Management (IEEM). Piscataway: IEEE, 2020: 1291-1295.
- [4] WANG L C, CHEN C C, LIU J L, et al. Framework and deployment of a cloud-based advanced planning and scheduling system[J]. *Robotics and Computer-Integrated Manufacturing*, 2021, 70: 102088.
- [5] MANKOWSKI M, MOSHKOV M. *Dynamic Programming Multi-Objective Combinatorial Optimization*[M]. Cham: Springer International Publishing, 2021.
- [6] ÖZGÜR A, UYGUN Y, HÜTT M T. A review of planning and scheduling methods for hot rolling mills in steel production[J]. *Computers & Industrial Engineering*, 2021, 151: 106606.
- [7] LAILI Y J, GUO F Q, REN L, et al. Parallel scheduling of large-scale tasks for industrial cloud-edge collaboration[J]. *IEEE Internet of Things Journal*, 2023, 10(4): 3231-3242.
- [8] DONG L B, SATPUTE M N, SHAN J Y, et al. Computation offloading for mobile-edge computing with multi-user[C]//2019 IEEE 39th International Conference on Distributed Computing Systems (ICDCS). Piscataway: IEEE, 2019: 841-850.
- [9] LI C L, BAI J P, TANG J H. Joint optimization of data placement and scheduling for improving user experience in edge computing[J]. *Journal of Parallel and Distributed Computing*, 2019, 125: 93-105.
- [10] ALAM M G R, MUNIR M S, UDDIN M Z, et al. Edge-of-things computing framework for cost-effective provisioning of healthcare data[J]. *Journal of Parallel and Distributed Computing*, 2018, 123: 54-60.
- [11] DING S, LI L, LI Z M, et al. Smart electronic gastroscope system using a cloud-edge collaborative framework[J]. *Future Generation Computer Systems*, 2019, 100(C): 395-407.
- [12] LONG C C, CAO Y, JIANG T, et al. Edge computing framework for cooperative video processing in multimedia IoT systems[J]. *IEEE Transactions on Multimedia*, 2018, 20(5): 1126-1139.
- [13] RUAN L N, YAN Y, GUO S Y, et al. Priority-based residential energy management with collaborative edge and cloud computing[J]. *IEEE Transactions on Industrial Informatics*, 2020, 16(3): 1848-1857.
- [14] JIANG Z Q, YUAN S, MA J, et al. The evolution of production scheduling from Industry 3.0 through Industry 4.0[J]. *International Journal of Production Research*, 2022, 60(11): 3534-3554.
- [15] 熊小峰, 黄淳岚, 乐光学, 等. 边缘计算中基于综合信任评价的任务卸载策略[J]. *电子学报*, 2022, 50(9): 2134-2145.
- XIONG X F, HUANG C L, YUE G X, et al. Task offloading scheme based on comprehensive trust evaluation in edge computing[J]. *Acta Electronica Sinica*, 2022, 50(9): 2134-2145. (in Chinese)
- [16] 宋政育, 郝媛媛, 孙昕. 低轨卫星协作边缘计算任务迁移和资源分配算法[J]. *电子学报*, 2022, 50(3): 567-573.
- SONG Z Y, HAO Y Y, SUN X. Computation offloading and resource allocation algorithm for collaborative LEO satellite multi-access edge computing[J]. *Acta Electronica Sinica*, 2022, 50(3): 567-573. (in Chinese)
- [17] ALAMGIR HOSSAIN S K, RAHMAN M A, HOSSAIN M A. Edge computing framework for enabling situation awareness in IoT based smart city[J]. *Journal of Parallel and Distributed Computing*, 2018, 122: 226-237.
- [18] LIU Y, YANG C, JIANG L, et al. Intelligent edge computing for IoT-based energy management in smart cities[J]. *IEEE Network*, 2019, 33(2): 111-117.
- [19] KAUR K, GARG S, AUJLA G S, et al. Edge computing in the industrial Internet of Things environment: Software-defined-networks-based edge-cloud interplay[J]. *IEEE Communications Magazine*, 2018, 56(2): 44-51.
- [20] LIN J, YU W, ZHANG N, et al. A survey on Internet of Things: Architecture, enabling technologies, security and privacy, and applications[J]. *IEEE Internet of Things Journal*, 2017, 4(5): 1125-1142.
- [21] LAI C F, CHIEN W C, YANG L T, et al. LSTM and edge computing for big data feature recognition of industrial electrical equipment[J]. *IEEE Transactions on Industrial Informatics*, 2019, 15(4): 2469-2477.
- [22] UM C, LEE J, JEONG J. Industrial device monitoring and control system based on oneM2M for edge computing[C]//2018 IEEE Symposium Series on Computational Intelligence (SSCI). Piscataway: IEEE, 2018: 1528-1533.
- [23] SODHRO A H, PIRBHULAL S, DE ALBUQUERQUE V H C. Artificial intelligence-driven mechanism for edge computing-based industrial applications[J]. *IEEE Transactions on Industrial Informatics*, 2019, 15(7): 4235-4243.
- [24] SUN W, LIU J J, YUE Y L. AI-enhanced offloading in edge computing: When machine learning meets industrial IoT[J]. *IEEE Network: the Magazine of Global Internet-working*, 2019, 33(5): 68-74.

- [25] LI X M, WAN J F, DAI H N, et al. A hybrid computing solution and resource scheduling strategy for edge computing in smart manufacturing[J]. IEEE Transactions on Industrial Informatics, 2019, 15(7): 4225-4234.
- [26] CHEN Y D, FENG Q G, SHI W S. An industrial robot system based on edge computing: An early experience[C]//USENIX Workshop on Hot Topics in Edge Computing (HotEdge 18). Boston: USENIX Association, 2018.
- [27] AFRIN M, JIN J, RAHMAN A, et al. Multi-objective resource allocation for edge cloud based robotic workflow in smart factory[J]. Future Generation Computer Systems, 2019, 97(C): 119-130.
- [28] FENG Y X, YANG C, WANG T Y, et al. Quality control system of automobile bearing production based on edge cloud collaboration[C]//2020 International Conference on Advanced Mechatronic Systems (ICAMEchS). Piscataway: IEEE, 2020: 319-322.
- [29] 唐捷凯, 胡蓉, 钱斌, 等. 混合帝国竞争算法求解带多行程批量配送的多工厂集成调度问题[J]. 电子学报, 2022, 50(7): 1621-1630.
- TANG J K, HU R, QIAN B, et al. Hybrid imperialist competitive algorithm for solving multi-factory integrated scheduling problem with multi-trip batch delivery[J]. Acta Electronica Sinica, 2022, 50(7): 1621-1630. (in Chinese)
- [30] LIU L L, WAN X, GAO Z G, et al. Research on modelling and optimization of hot rolling scheduling[J]. Journal of Ambient Intelligence and Humanized Computing, 2019, 10(3): 1201-1216.
- [31] HU Z B, HE D F, SONG W, et al. Model and algorithm for planning hot-rolled batch processing under time-of-use electricity pricing[J]. Processes, 2020, 8(1): 42.
- [32] HU W Z, ZHENG Z, GAO X Q, et al. An improved method for the hot strip mill production scheduling problem[J]. International Journal of Production Research, 2019, 57(10): 3238-3254.
- [33] TANG L X, LIU J Y, RONG A Y, et al. A multiple traveling salesman problem model for hot rolling scheduling in Shanghai Baoshan Iron & Steel Complex[J]. European Journal of Operational Research, 2000, 124(2): 267-282.
- [34] BECK J C, PROSSER P, SELENSKY E. Vehicle routing and job shop scheduling: What's the difference? [C]//Proceedings of the Thirteenth International Conference on International Conference on Automated Planning and Scheduling. Trento: AAAI Press: 2003: 267-276.
- [35] LEUSIN M, FRAZZON E, URIONA MALDONADO M, et al. Solving the job-shop scheduling problem in the industry 4.0 era[J]. Technologies, 2018, 6(4): 107.
- [36] LI X Y, LU C, GAO L, et al. An effective multiobjective algorithm for energy-efficient scheduling in a real-life welding shop[J]. IEEE Transactions on Industrial Informatics, 2018, 14(12): 5400-5409.
- [37] DRISS I, MOUSS K N, LAGGOUN A. A new genetic algorithm for flexible job-shop scheduling problems[J]. Journal of Mechanical Science and Technology, 2015, 29(3): 1273-1281.
- [38] XU S T, ZHANG J, HAN D W, et al. Singular value decomposition based data distortion strategy for privacy protection [J]. Knowledge and Information Systems, 2006, 10(3): 383-397.

作者简介



丁婧伊 女, 2000年1月出生于内蒙古呼和浩特市. 现为东南大学计算机科学与工程学院硕士研究生. 主要研究方向为分布式计算、工业互联网.

E-mail: jingyi_ding@seu.edu.cn



金嘉晖 男, 1986年2月出生于浙江省温州市. 现为东南大学计算机科学与工程学院副教授、博士生导师. 主要研究方向为分布式数据处理、工业互联网、智能物联网.

E-mail: jjin@seu.edu.cn

杨丰赫 男, 1996年3月出生于浙江省丽水市. 2021年毕业于东南大学网络空间安全学院. 主要研究方向为工业互联网、云排产.

E-mail: fhyang@aa.seu.edu.cn

熊润群 男, 1982年9月出生于福建省龙岩市. 现为东南大学计算机科学与工程学院副教授. 主要研究方向涉及云计算、大数据、物联网、工业互联网.

E-mail: xiong@seu.edu.cn

单冯 男, 1985年5月出生于江苏省南通市. 2016年于东南大学获得博士学位, 现为东南大学计算机科学与工程学院副教授. 研究方向为算法设计与分析、物联网.

E-mail: shanfeng@seu.edu.cn

东方 男, 1982年5月出生于江苏省南京市. 现为东南大学计算机科学与工程学院教授、博士生导师. 主要研究方向为边缘计算、云计算与工业互联网.

E-mail: fdong@seu.edu.cn