

控制与决策

Control and Decision

面向云平台中心化集成调度的跨供应商订单分配模型

谢乃明, 吴乔, 郑绍祥

引用本文:

谢乃明, 吴乔, 郑绍祥. 面向云平台中心化集成调度的跨供应商订单分配模型[J]. *控制与决策*, 2020, 35(3): 667–676.

在线阅读 View online: <https://doi.org/10.13195/j.kzyjc.2018.0545>

您可能感兴趣的其他文章

Articles you may be interested in

基于系统动力学的大型客机供应链质量管控契约优化

Contracts and optimization to quality control in supply chain of large-scale passenger aircraft based on system dynamics

控制与决策. 2020, 35(1): 215–227 <https://doi.org/10.13195/j.kzyjc.2018.0649>

基于成本共担契约的生鲜供应链保鲜努力机制

Freshness efforts mechanism of fresh-keeping supply chain based on cost sharing contract

控制与决策. 2020, 35(1): 205–214 <https://doi.org/10.13195/j.kzyjc.2018.0612>

基于ACC支付模式的两层信用支付的EOQ模型

An EOQ model under two levels of trade credit with ACC payments

控制与决策. 2019, 34(8): 1769–1775 <https://doi.org/10.13195/j.kzyjc.2017.1772>

基于微分博弈的供应链合作减排和政府补贴策略

Reduce emissions in cooperation and government subsidies strategies in supply chain based on differential game

控制与决策. 2019, 34(8): 1733–1744 <https://doi.org/10.13195/j.kzyjc.2018.0043>

产出不确定环境下考虑供货承诺的定价与投入决策模型

Pricing and input decision models under yield uncertainty considering supply commitment

控制与决策. 2017, 32(9): 1664–1671 <https://doi.org/10.13195/j.kzyjc.2016.1000>

不对称信息下供应链赊销交易中的激励契约

Incentive contract in supply chain trade credit under asymmetric information

控制与决策. 2017, 32(3): 570–576 <https://doi.org/10.13195/j.kzyjc.2015.1543>

离散型帝国竞争算法在仓储订单调度中的应用

Application of discrete imperialist competitive algorithm in warehouse order scheduling

控制与决策. 2017, 32(1): 39–46 <https://doi.org/10.13195/j.kzyjc.2015.1230>

考虑损失规避型供应商的VMI供应链协调

VMI supply chain coordination with the loss-averse supplier

控制与决策. 2016, 31(5): 935–942 <https://doi.org/10.13195/j.kzyjc.2015.0385>

面向云平台中心化集成调度的跨供应商订单分配模型

谢乃明, 吴 乔[†], 郑绍祥

(南京航空航天大学 经济与管理学院, 南京 211106)

摘 要: 跨供应商订单分配可以充分发挥云平台优势, 有效整合利用各供应商的剩余资源. 鉴于此, 提出一种云平台主导的集成调度模式, 基于该模式和供应商的产能约束、运输约束、订单约束构建面向云平台中心化集成调度的跨供应商订单分配模型, 给出基于改进遗传算法的模型求解方法, 并通过算例演示所构建模型和算法的有效性. Matlab 实验结果表明, 订单能够在各供应商间得以科学分配, 实现各供应商闲置资源有效利用和成本最优.

关键词: 云平台; 中心化集成调度; 订单分配; 遗传算法

中图分类号: C93-03; TP301

文献标志码: A

Cross-vendor order allocation model for centralized integrated scheduling of cloud platforms

XIE Nai-ming, WU Qiao[†], ZHENG Shao-xiang

(College of Economic and Management, Nanjing University of Aeronautics and Astronautics, Nanjing 211106, China)

Abstract: Cross-supplier order allocation can give full play to the advantages of the cloud platform, and effectively integrate and utilize the remaining resources of each supplier. This paper presents an integrated scheduling model dominated by cloud platform. Based on this pattern and the supplier's capacity constraints, transportation constraints, and order constraints, a cross-vendor order allocation model for cloud platform centralized integration scheduling is constructed. The method for model solving based on the improved genetic algorithm is given. Finally, an example is given to illustrate the effectiveness of the proposed model and algorithm. The Matlab experiment results show that the orders can be distributed scientifically among various suppliers, and that the idle resources of each supplier can be effectively used and the cost can be optimized.

Keywords: cloud platform; centralized integration scheduling; order allocation; genetic algorithm

0 引 言

我国制造企业中普遍存在制造能力不均衡的现象, 导致资源闲置和资源瓶颈并存, 造成制造资源的浪费, 因此迫切需要一个可以进行制造资源服务及交易的平台, 有效平衡企业的制造能力. 与传统只售卖产品的电子商务平台不同, 云制造平台可以满足各类不同的定制化需求, 按订单组织生产和服务, 通过云将多个不同地理位置的制造单元和客户集成在同一平台, 随时随地访问供应链内的任何制造资源, 实现行业内资源的共享, 高效地利用现有供应商的剩余产能. 国内有多家高校和集团参与了云平台的开发, 上线了一系列云平台产品, 代表性产品有中国航天科工集团开发的航天云网、国家智能铸造中心的共享工业云和机械科学研究总院的中机云创等. 云调度是

云制造平台需要解决的一个核心难题, 充分利用云平台主导的集成调度优势, 可以明确各方任务需求, 实现云平台各参与主体的利益最优, 加快云平台在国内外的发展进程.

所谓云制造, 是一种基于网络的、面向服务的智慧化制造新模式, 它将各类制造资源和制造能力虚拟化、服务化, 构成服务云池, 进行统一集中的优化管理. 2010年, 李伯虎等^[1]首次提出云制造的定义和平台体系框架, 强调设计为服务(DAAS)、生产加工为服务(FAAS)等制造即服务模式, 之后, 研究人员多集中于对云平台系统的理论架构和关键技术的突破中, 出现了基于云平台的分布式制造和多代理商^[2]的研究, 以及在云制造背景下分布式制造的运行机理与框架改变、制造网络系统的集成^[3]. 同时, 由于云制

收稿日期: 2018-04-27; 修回日期: 2018-12-09.

基金项目: 国家自然科学基金项目(71671090, 71871117, 71801127, 51705250); 国家自然科学基金委与英国皇家学会合作交流项目(71811530338); 中央高校基本科研业务费专项资金项目(NP2018466); 江苏省青蓝工程中青年学术带头人计划.

[†]通讯作者. E-mail: Wuqiaonuaa@163.com.

造本身就是基于敏捷制造发展起来的一种更具有自适应性的体系^[4],在原有调度问题的基础上,研究了多代理商协同下的组合调度^[5]和高度动态环境中的实时调度优化模式^[6].进一步地,也有对如何从理论构架走向实施的思考,探讨如何搭建云平台. Ren等^[7]提出了针对中小企业云制造平台的原型和应用程序,李伯虎等^[8]构建了面向复杂产品研发的云制造平台, Wu等^[9]基于云制造系统的供应商绩效评价,探讨了如何将供应商内部制造和众包集成到系统中发挥其制造的可扩展性.

订货量分配问题的研究由来已久,从早期的多供应商条件下订货批量和订单分配问题研究^[10],到结合供应商选择问题确定各供应商最优订单数量^[11],再到近几年基于需求、价格不确定性的订货量分配、供应商选择^[12],研究已较深入.但与基于云制造的多供应商订单分配问题还是有本质区别,云制造要考虑的不仅是订单量的分配或者设备与订单的匹配,更多地应站在服务的层面,探讨服务的柔性和可组合性.所以应该充分考虑物流、组织生产的可能性、组合优化的利益主体的满意程度.目前将订货量分配与企业实际调度结合的论文十分有限,仅有的几篇如李占丞等^[13]考虑订货量分配决策对订单生产和交货的影响,建立了订货量分配与订单排产联合优化模型.

近几年关于云制造的研究涉及技术攻关、调度等多个方向,围绕制造任务需求和企业剩余制造能力的匹配以及云制造背景下的调度是其中的重要分支.赵道致等^[14]基于双边匹配理论,研究了考虑主体心理预期的云制造资源双边匹配机制;Liu等^[15]在考虑任务负载对云制造多任务调度的模型构建中,验证了优先调度负载大的任务可以有效减少总完成时间;Cao等^[16]在考虑TQCS(时间、质量、成本和服务)的基础上建立了服务选择和调度的模型,并利用蚁群算法进行了求解;王贞等^[17]在已有订单的基础上考虑了如何利用设备空闲时间对云任务进行作业车间调度.在云制造中心化调度中,订货量分配不仅要考虑订货量分配问题,还应该与企业的排产决策、运输策略相结合,将集成调度的思想渗入,追求总成本的最优化,这类研究目前较少见到.

现有的云制造调度研究中基于云平台主导的集成调度研究较少,而这些研究尚未考虑订单的可拆单情形.本文考虑将资源剩余工时的租用扩展到任务的可拆单情形,如何进行多任务的组合调度,使总成本最低.在这里设想一下可拆单情形对供应链双方的价值,假设供应商 M_1 有 A_1 、 A_2 、 A_3 、 A_4 四种剩余资

源,剩余能力极限分别为100、120、100、110,如果订单池中对资源类型 A_1 下单的订单需求工时均大于130,则对于供应商 M_1 而言,其剩余能力不足以承担平台内订单池中的任何一个需租用资源类型 A_1 的订单,但在可拆单情形下,供应商 M_1 的参与对供应链双方都是共赢的局面; M_1 可利用剩余能力增加收入,客户可在组合下单中降低采购成本,这充分体现了云平台的共享优势.

综上所述,随着智能制造的推广及云制造平台服务的不断发展与完善,面向云平台中心化的集成调度研究无疑会成为未来智能制造发展的重点,所以对该类问题的研究尤为必要.本文提出了云平台中心化运作模式,构建了考虑产能约束、运输约束、订单约束的云平台中心化集成调度的跨供应商订单分配模型,给出了基于改进遗传算法的模型求解算法,并将其结果与CPLEX^[18]的求解结果进行对比.实验结果表明,订单能够在各供应商间得以科学分配,实现了各供应商闲置资源有效利用和成本最优.

1 云平台服务运作模式研究

现阶段新兴的云制造平台主要分为4种类型:针对复杂产品的高端制造业设计能力服务平台;区域性的加工资源共享平台;制造公共服务支持平台;整机制造商、零部件供应商以及物流企业间的多方物流协同平台.本文背景主要基于区域性加工资源共享云平台,平台各方参与者如图1所示,分别为云平台计算中心、供应商和客户三方.供应商在云平台可实现剩余资源信息共享、询单、报价等;客户在云平台发布订单需求信息、下单等;云平台计算中心对云平台进行日常管理维护,实现精确搜索、主动匹配.

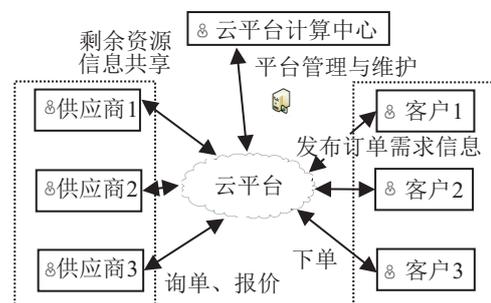


图1 云平台框架

在已有的云平台框架下,目前投入运营的平台其生产性服务模块共有两种典型运作模式:一种是“需求型”模式,如图2(a)所示,其开始与结束节点由客户控制,客户将订单信息(工件对象、需求数量)共享到平台中,由供应商报价投标,“航天云网”早期实行的就是这种模式;另一种是“服务型”模式,如图2(b)所

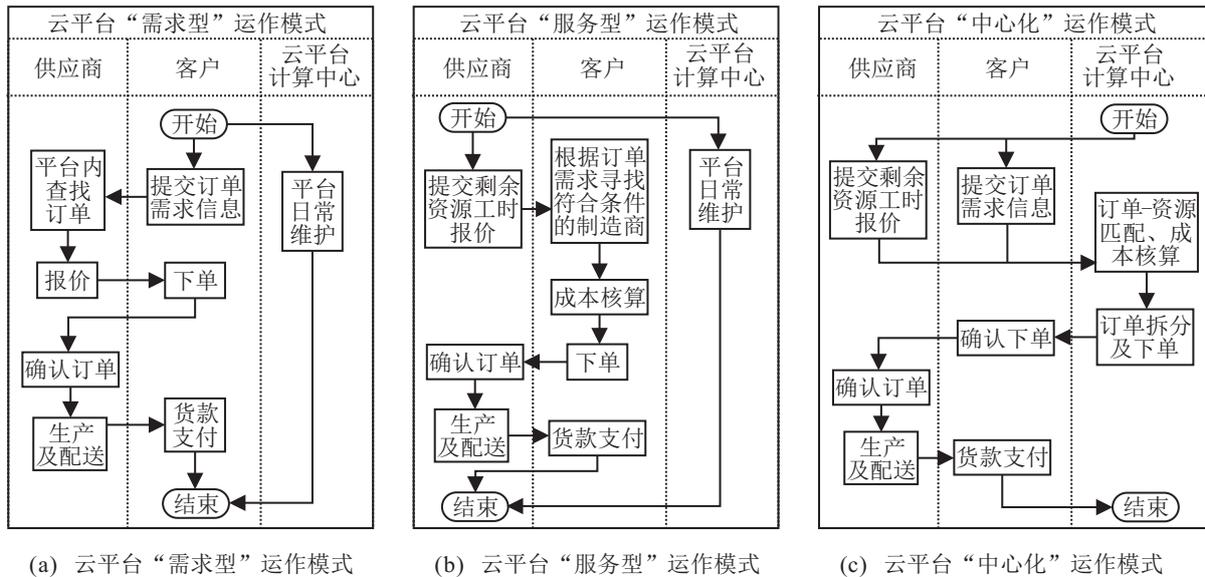


图2 云平台3种运作模式

示,其开始与结束节点由供应商控制,供应商将剩余生产能力,如设备资源工时、报价等共享到平台中,由客户对生产能力工时下单,采用这种模式的产品有“共享工业云”等。

在图2(a)和图2(b)两种运作模式中,云平台计算中心承担的主要任务是日常维护,主要的交易过程依赖于供应链双方对信息的“单向式”共享和分析,初级阶段没有发挥云平台后台的数据处理能力和计算优势,很难达到订单和剩余资源最优化匹配的目标.在此基础上,为了充分发挥云平台的调度中心优势,提出了如图2(c)所示的“中心化”运作模式.云平台掌握全流程的开始与结束,顾客和供应商同时提交订单和资源信息,利用云平台的计算能力进行供应商匹配、成本核算、订单拆分和下单等操作,尽可能地使供应链总成本最低.在图2(c)的“中心化”运作模式下,客户基于云平台寻求外部供应商资源,供应商以资源工时进行报价,假设其单位报价已包含单位时间内的物料、设备、人员投入等,对于客户而言,一个订单可以选择从一家供应商预订或多家组合预订,其成本由零部件供应商的报价(生产成本)、物流成本和库存成本所决定.一般来说,生产成本最高,其次是物流成本,库存成本.本文将讨论如何以云平台为调度中心进行订单分派、运输方式组织,使订单总成本最低,构建面向云平台中心化集成调度的跨供应商订单分配模型。

2 面向云平台中心化集成调度的跨供应商订单分配模型构建

本文针对多任务订单需求的情形,以多供应商、

单一客户情形为例,如图3所示.客户提交订单需求,供应商将剩余资源分享到云平台的资源池中,通过云平台进行匹配.例如订单 B_1 需求的资源类型为 A_1 ,需求工时数为 t ,其在云平台资源池中寻求承接的供应商,可拆单分配,只需 $a + b + c$ 的总工时数满足工时的需求,在满足其他现实约束(订单约束、设备能力约束、运输约束等)的情况下寻求使总成本最小的组合方式。

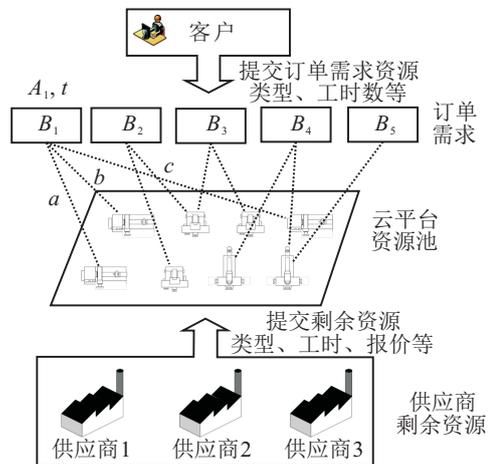


图3 云平台的服务匹配过程

2.1 参数定义

2.1.1 平台使用者及其服务

平台内有 $M_j (j = 1, 2, \dots, n)$ 个供应商,对于每一供应商 M_j ,其所提供的资源类型为 $A_k (k = 1, 2, \dots, m)$,其中 m 表示 M_j 所提供资源类型的最大数量. M_j 对于资源类型 A_k 的剩余能力极限为 $L_{j,k}$, M_j 对每一 A_k 单位工时的报价为 $P_{j,k}$,供应商 M_j 到客户的距离为 d_j 。

2.1.2 任务需求

平台内有需要寻求外部资源的多个任务订单 $B_i (i = 1, 2, \dots, z)$, 其中 z 代表订单的总数. 每一订单 B_i 可由一个五元组表示 $|A_k, t_i, H_i, \text{Cons}_i, R_i|$, A_k 为 B_i 所需租用的资源类型, t_i 为资源类型所对应的租用工时, H_i 为 B_i 所占用的运输车箱容量, Cons_i 为 B_i 的规定完工周期(从发布订单到货物送达入库), R_i 为 B_i 的优先级. 优先级由规定的完工周期 Cons_i 和资源类型的租用工时 t_i 的差值决定, 越紧急的订单优先级越高.

2.1.3 决策变量

1) 拆单决策矩阵为

$$C = \begin{bmatrix} \alpha_{1,1} & \alpha_{1,2} & \dots & \alpha_{1,z} \\ \alpha_{2,1} & \alpha_{2,2} & \dots & \alpha_{2,z} \\ \vdots & \vdots & \ddots & \vdots \\ \alpha_{n,1} & \alpha_{n,2} & \dots & \alpha_{n,z} \end{bmatrix}.$$

其中

$$\sum_{j=1}^n \alpha_{j,i} = 1; \alpha_{j,i} \in \begin{cases} [\alpha_0, 1], \text{ order;} \\ \{0\}, \text{ otherwise;} \end{cases}$$

$\alpha_{j,i}$ 为供应商 M_j 在订单 B_i 中的分配比例. 设备租用时间过低, 供应商不易组织生产, 在进行订单分配的过程中保证订单的分配比例要大于 α_0 , 否则不接单.

2) 订单 B_i 在供应商 M_j 的设备资源上的实际开工时间为 $O_{i,j}$. 任意设备 A_k 都对应一个订单集合 $K_B = [K_1, K_2, \dots, K_b]$, K_B 的订单总数为 b 个.

3) 供应商 M_j 配送时间矩阵为

$$T = \begin{bmatrix} T_{1,1} & T_{1,2} & \dots & T_{1,r} \\ T_{2,1} & T_{2,2} & \dots & T_{2,r} \\ \vdots & \vdots & \ddots & \vdots \\ T_{n,1} & T_{n,2} & \dots & T_{n,r} \end{bmatrix}.$$

其中: $T_{j,c} (c = 1, 2, \dots, r)$ 为供应商 M_j 第 c 次配送的时点, r 为 M_j 配送的总次数. $T_{j,c}$ 时点 M_j 内的库存量为 $q_{j,c}$ ($q_{j,c}$ 为整数).

2.2 模型构建

2.2.1 背景和假设

1) 现有国内外工业云平台的运作中, 对于不涉及复杂产品的订单, 为减少生产周期, 降低运输成本, 原材料一般由承接方(供应商)提供, 故本文假设报价已包含原材料成本, 设备、人员投入.

2) 不同供应商在提供同类型的服务时能力一般会有所差异, 需要考虑效率系数, 效率系数的计算可参考文献[16]的计算形式. 如果平台内有4家公司提供某一资源类型的服务, 生产能力分别为每天4件、3

件、5件、4件, 则平均效率为每天4件, 效率系数分别为 $1.0(4/4)$ 、 $0.75(3/4)$ 、 $1.25(5/4)$ 、 $1.0(4/4)$. 为简化计算, 本文假设各供应商的服务效率差距忽略不计.

3) 订单承接方会向客户收取延迟取货的费用, 即库存费用, 设 M_j 的单位时间库存费用为 S_j .

4) 在各设备类型上生产的各订单之间的换型时间、准备时间需根据各公司实际测算, 本文暂不考虑.

5) 假设运输由供应商向客户单向配送, 单个供应商对客户的运输仅由一辆运输车承担, 配送时点为 $T_{j,c}$. 在云平台中仅单程运费由客户承担, 由于物流运输可外包, 返程一般不空载. 设单位距离运费为 V_j , 标准运输车辆的总容量为 D , 单位距离的物流时间为 l .

6) 在云平台上进行工时租借的多是多品种、小批量订单, 这种生产模式下运输成本一般远大于库存成本, 所以本文的运输规则为: 在供应商生产库存数量 $q_{j,c}$ 接近但不超过运输车总容量 D 的情况下尽可能早地运输货物; 排好的运输批若某两批容量和仍小于 D , 则需要进一步更新, 重新合并组合, 运输时点为合并后最晚完工订单的完工时点. 例如, 运输车总容量 D 为 10, 按规则排好的订单运输批各容量为 10、5、8、5, 将其更新后应该为 10、8、10, 易验证这时运输和库存的组合成本最低.

2.2.2 目标函数和约束

1) 生产成本: 对于每一 M_j 而言, 其订单 B_i 实际的服务时间应为 $\text{TR}_{i,j} = t_i \times \alpha_{j,i}$, 则有

$$\text{PC} = \sum_{i=1}^z \sum_{j=1}^n \text{TR}_{i,j} \times P_{j,k}, \quad k = 1, 2, \dots, m. \quad (1)$$

2) 运输成本

$$\text{DC} = \sum_{j=1}^n V_j \times d_j \times r. \quad (2)$$

3) 库存成本

$$\text{IC} = \sum_{i=1}^z \sum_{j=1}^n (T_{j,c} - (O_{i,j} + \text{TR}_{i,j})) \times S_j. \quad (3)$$

客户总成本最低的线性规划模型 P 为

$$\min \text{TC} = \text{PC} + \text{DC} + \text{IC}. \quad (4)$$

$$\text{s.t.} \quad \sum_{j=1}^n \alpha_{j,i} = 1, \quad i = 1, 2, \dots, z; \quad (5)$$

$$\alpha_{j,i} \in \begin{cases} [\alpha_0, 1], \text{ order;} \\ \{0\}, \text{ otherwise;} \end{cases} \quad (6)$$

$$T_{j,c} + l \times d_j \leq \text{Cons}_i; \quad (7)$$

$$O_{i+1,j} - O_{i,j} - \text{TR}_{i,j} = 0, \quad i \in K_B; \quad (8)$$

$$\sum_{i=1}^b \text{TR}_{i,j} \leq L_{j,k},$$

$$k = 1, 2, \dots, m, j = 1, 2, \dots, n; \quad (9)$$

$$\sum_{i=1}^z [H_i \times \alpha_{j,i}] \leq rD; \quad (10)$$

$$T_{j,c} \geq O_{i,j} + TR_{i,j}; \quad (11)$$

$$q_{j,c} \leq D, c = 1, 2, \dots, r; \quad (12)$$

$$T_{j,c+1} - T_{j,c} - 2 \times l \times d_j \geq 0. \quad (13)$$

该模型的目标为确定各供应商分配到的订单比例、订单的开完工时间及各供应商的运输批和运输时点,使得生产、运输和库存的总成本最低.式(1)~(4)为目标函数,式(4)表示订单的总成本最低.式(5)~(7)为订单约束:式(5)表示订单需要的租用工时数应全部分配给平台内承接的供应商;式(6)表示进行订单分配的过程中,如果接单则订单分配比例一定要大于等于 α_0 ,否则不接单;式(7)表示交货期需满足订单要求的完工周期.式(8)和(9)为设备能力约束:式(8)表示对于每一服务提供商 M_j ,其 A_k 设备资源完成上一订单即开始下一订单的生产,不考虑换型等待时间;式(9)表示对于任意设备 A_k 所分配的订单工时需小于等于设备的剩余能力极限.式(10)~(13)为运输约束:式(10)保证每一 M_j 在 r 次内能运输完所有订单产成的工件, $[H_i \times \alpha_{j,i}]$ 表示对每一 M_j 所分到的订单 B_i 而言,其运输车占容需向上取整;式(11)表示订单的运输时间要晚于订单的完工时间,分配到同一供应商的同一订单不能拆单运输;式(12)表示对于任一 M_j ,任意运输时点的库存量水平都要低于货物的一次最大运量;式(13)表示两次运输时间的差值要大于等于运输车一次运输的往返时间.

模型的计算复杂性可由以下特殊情形 P' 简单给出:1)库存成本为零;2)生产成本为零;3)只有一个供应商;4) $\alpha_{j,i} = 1$.

条件1)和条件2)将问题简化为只考虑最小化运输成本,条件3)使问题等价于只考虑最小化运送次数 r ,条件4)限制了订单任务的不可分割性.将问题进一步简化为经典的装箱问题:目标为最小化箱子的数量(运送次数);物品数量:订单数 z ,物品的尺寸 H_i ,箱子的容积 D .装箱问题是著名的NP-hard问题,故 P' 是NP-hard的,而 P' 是 P 的特例, P 的计算复杂性高于 P' ,因此 P 是NP-hard的.

3 基于改进遗传算法的算法构建

3.1 改进遗传算法求解框架

上文描述的云平台中心化模式下的跨供应商订单分配模型决策变量较多,已被证明为NP-hard问题,传统的计算方式并不适用于求解该类大规模问题,所

以本文采用仿生智能算法求解这类有约束最优化问题的近似解.目前在同类问题中使用最多的是粒子群算法和遗传算法,由于遗传算法适合离散型问题,有较强的全局收敛能力,收敛速度快,近几年被广泛运用于求解订单分配及调度问题.但遗传算法本身具有易陷入过早收敛的缺点,所以本文对原始的遗传算法进行改进,以避免遗传算法的求解结果陷入“早熟”.具体算法求解流程如图4所示.

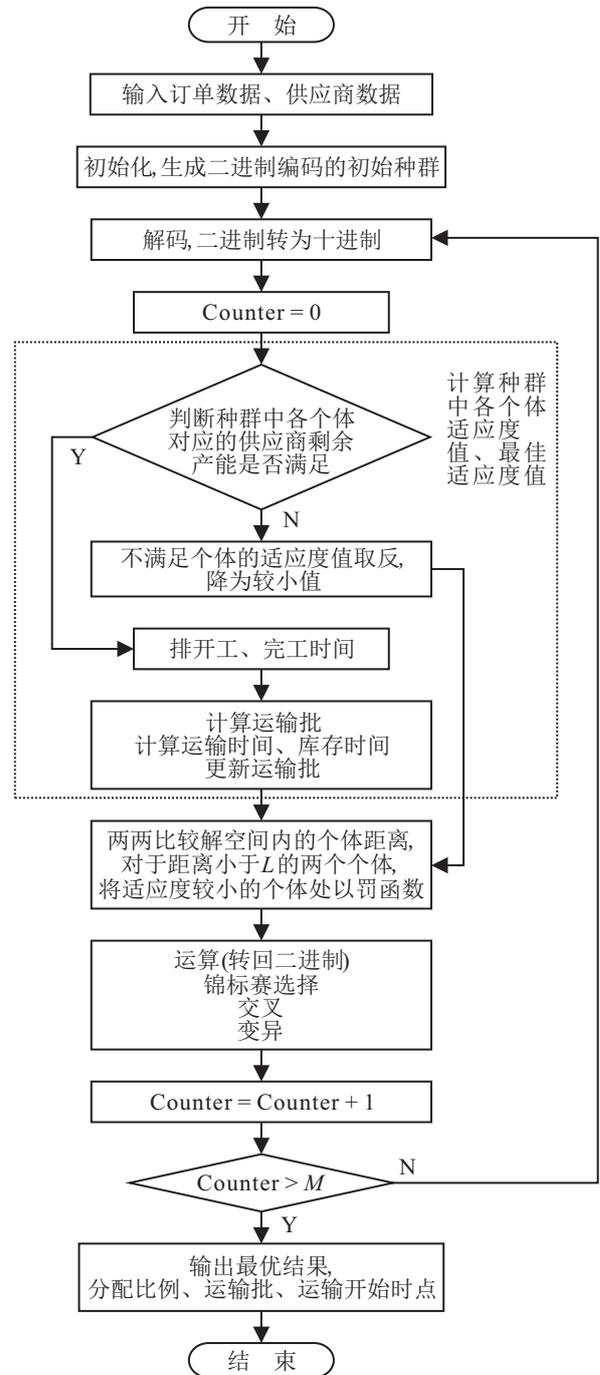


图4 求解算法流程

3.2 关键算法步骤描述

1) 个体编码及初始化.

拆单决策矩阵是由 $\alpha_{j,i}$ (小数)组成的矩阵,用实

数编码方式在运算时不易交叉和补位,所以本文求解算法的编码采用二进制编码.染色体个体由 $10 \times n \times z$ (n 为供应商总数, z 为订单总数)列0,1编码组成.

由于分配比例均为 $[0, 1]$ 范围内的小数,在编码时将小数转为整数以减少编码位数,如0.134(保留3位有效数)可转换为134,因此用10位长度的0,1编码可以表示一个订单分配给一个供应商的分配结果.每 $10 \times n$ 列编码代表一个订单的分配结果,一个订单必须全部分配给平台内的 n 个供应商,所以每 $10 \times n$ 列编码在解码后的和应约等于1000.一条染色体个体由 z 组 $10 \times n$ 二进制编码组成,初始化随机生成NIND条染色体.对不符合订单分配要求 α_0 的染色体个体中的每10列置0,随后对每条染色体进行归一化处理,以满足每 $10 \times n$ 列解码和为1000的要求.

2) 适应度计算.

适应度在计算过程中参考了基于改进小生境遗传算法求解多峰函数优化问题的算法,在计算种群各个体适应度时加入罚函数,具体步骤如下.

step 1: 将二进制编码转化为十进制,恢复小数表示,排出种群内各个体的订单工时分配结果.

step 2: 排开工时间.根据订单 $B_1 \sim B_z$ 的优先级对供应商各设备类型的租用时间进行排序.

step 3: 计算每个供应商的合理运输时间.运输规则是当 $T_{j,c} \sim T_{j,c+1}$ 时间段 M_j 内的库存量 $q_{j,c}$ 接近但不超过一次运输容量 D 时,在时间段内最后一个订单的完工时间点组织一次运输,对于两结束时点相同的订单,应在满足运输容量 D 的情况下优先排占容大的订单;如果运输车还未返回,则时间点需延迟至运输车返回时点,记录每个供应商的每次运输时点和运输的订单号;对于运输批可合并的情况要进行更新操作,更新规则参考第2.2.1节假设6).

step 4: 根据运输时间和订单的完工时间记录订单的库存时间,计算各供应商的总成本.

step 5: 计算种群中各条染色体个体的总成本.由于遗传算法通常进化的方向是适应度函数不断增大的方向,需要将总成本取反 $\text{fitness} = K/\text{TC}$, K 为一个较大的数.过程中需要对供应商产能进行核算,如果接单超过供应商剩余的可用产能,则将适应度值再次降低 $\text{fitness} = 1/\text{fitness}$.

step 6: 借鉴小生境遗传算法对适应度值进行更新,比较解空间内两两个体的距离,如个体0111与个体1010的距离为

$$\sqrt{(0-1)^2 + (1-0)^2 + (1-1)^2 + (1-0)^2} = \sqrt{3},$$

当两个体距离小于界限 L 时,降低适应度值较低个体

的适应度.

3) 选择操作.

由于对适应度值进行了更新操作,在比选方法过程发现选择操作中用传统的轮盘赌的方法效果不佳,而锦标赛算法的收敛速度更快、过程更稳定,本文选用锦标赛算法进行选择.具体步骤如下.

step 1: 确定每次从种群中选择的个体数量或占种群总数的百分比 per .

step 2: 在随机选择的小种群中由每个个体的适应度值选择其中适应度值最高的个体进入子代种群.

step 3: 重复Step 2,筛选出个体构成新一代种群.

4) 交叉操作.

交叉操作通过个体间基因片段的重组和调整以获得新的染色体,增加个体的多样性,从而推动种群的进化.具体步骤如下.

step 1: 将种群中所有个体随机打乱,按顺序从种群中选择 $1 \sim \text{NIND} \cdot \text{pc}$ (交叉概率)个个体进行交叉.

step 2: 每次选取两个染色体,随机选取两个交叉位置,将两位置区间段内的编码进行交换.

step 3: 调整操作.对交叉后的新个体进行调整,置0和归一化操作(具体可参照编码时的调整规则).

5) 变异操作.

变异算子从种群中随机选择 $\text{NIND} \cdot \text{pm}$ 个个体进行变异,随机选取两个变异位置 pos_1 和 pos_2 ,将个体中 pos_1 和 pos_2 位置的编码由‘0’变‘1’,由‘1’变‘0’,再进行调整操作.

该遗传算法的时间复杂度为 $O(nz \ln z)$.具体分析如下:在对染色体个体进行编码时,染色体的长度是订单数量 z 和供应商数目 n 的整数倍,且种群数目为常数,因此在染色体编码中其时间复杂度为 $O(nz)$.适应度的计算分为6个步骤,解码二进制编码时间复杂度为 $O(nz)$,根据订单优先级对 n 个供应商各设备类型的租用时间进行排序,其时间复杂度最高为 $O(nz \ln z)$.计算 n 个供应商运输时间、记录各个订单完工时间和计算各供应商总成本的时间复杂度均为 $O(nz)$,评估各染色体适应度的时间复杂度为 $O(n)$,调整个体适应度时间复杂度为 $O(1)$.使用锦标赛算法进行选择操作以及交叉和变异操作的时间复杂度为 $O(1)$.综上,该遗传算法时间复杂度为 $O(nz \ln z)$.

4 案例分析

4.1 数值仿真

本节就以上模型及算法设计了算例,采用Matlab R2015a进行仿真,以验证模型及算法在解决该类云

制造平台中的订单拆单及调度问题的有效性. 云制造平台内现有一个客户发布了12个订单需求, 其订单基本信息如表1所示.

表1 订单信息

| B_i | A_k | t_i/h | H_i | $Cons_i/d$ | R_i |
|----------|-------|---------|-------|------------|-------|
| B_1 | A_1 | 136 | 10 | 22 | 10 |
| B_2 | A_2 | 112 | 8 | 15 | 5 |
| B_3 | A_3 | 150 | 12 | 15 | 2 |
| B_4 | A_4 | 160 | 14 | 21 | 9 |
| B_5 | A_2 | 120 | 12 | 18 | 7 |
| B_6 | A_1 | 130 | 9 | 23 | 11 |
| B_7 | A_4 | 85 | 11 | 16 | 6 |
| B_8 | A_3 | 110 | 14 | 10 | 1 |
| B_9 | A_2 | 120 | 12 | 14 | 3 |
| B_{10} | A_4 | 130 | 16 | 20 | 8 |
| B_{11} | A_1 | 140 | 12 | 15 | 4 |
| B_{12} | A_3 | 100 | 8 | 22 | 12 |

每个订单需求对应一个五元组: 设备类型、租用工时、运输车占容、规定完工周期以及订单的优先级. 本算例中订单的优先级根据订单的紧急程度设置, 一般为规定完工周期和租用工时的差值, 差值越小优先级越高. 如订单 B_1 , 设备租用类型为 A_1 , 租用工时为136h, 运输车占容为10, 规定完工周期为22d, 在所有订单中优先级为10. 平台内现有8家供应商, 有资源类型 $A_1 \sim A_4$ 的剩余资源, 剩余资源信息如表2所示.

模型中其他参数设置: 单位距离运费 $V_j = 5$ 元/km, 单位时间库存费用 $S_j = 5$ 元/h, 运输标准车辆容量 $D = 10$, 平均速度 $1/l = 90$ km/h. 改进遗传算法部分参数设置(合理性已测试): 种群大小 $NIND = 200$, 遗传代数 $MAXGEN = 350$, $K = 10\,000\,000$, 交叉概率 $pc = 0.8$, 变异概率 $pm = 0.3$, 适应度更新个体距离界限 $L = 5$, 选择操作选择比率 $per = 0.6$.

4.2 结果分析

将模型得到的各项参数输入由Matlab编写的算法程序中, 计算得到各供应商的订单分配比例 $\alpha_{j,i}$ 如表3所示, 对应租用工时取整后的结果如表4所示. 各

表2 供应商剩余资源信息

| M_j | d_j/km | A_k | $L_{j,k}/h$ | $P_{j,k}/\text{¥}$ |
|-------|----------|-------|-------------|--------------------|
| M_1 | 183 | A_1 | 100 | 230 |
| | | A_2 | 120 | 180 |
| | | A_3 | 100 | 160 |
| | | A_4 | 110 | 210 |
| M_2 | 202 | A_1 | 100 | 240 |
| | | A_2 | 160 | 200 |
| | | A_3 | 180 | 190 |
| | | A_4 | 165 | 230 |
| M_3 | 217 | A_1 | 90 | 210 |
| | | A_2 | 150 | 190 |
| | | A_3 | 105 | 180 |
| | | A_4 | 140 | 200 |
| M_4 | 240 | A_1 | 90 | 220 |
| | | A_2 | 100 | 170 |
| | | A_3 | 100 | 170 |
| | | A_4 | 150 | 190 |
| M_5 | 125 | A_1 | 130 | 240 |
| | | A_2 | 90 | 150 |
| | | A_3 | 105 | 150 |
| | | A_4 | 160 | 210 |
| M_6 | 330 | A_1 | 120 | 220 |
| | | A_2 | 110 | 170 |
| | | A_3 | 110 | 190 |
| | | A_4 | 100 | 200 |
| M_7 | 261 | A_1 | 130 | 230 |
| | | A_2 | 100 | 160 |
| | | A_3 | 180 | 170 |
| | | A_4 | 145 | 190 |
| M_8 | 129 | A_1 | 160 | 210 |
| | | A_2 | 90 | 180 |
| | | A_3 | 100 | 180 |
| | | A_4 | 120 | 200 |

供应商承接订单排序按优先级顺排, 开完工时间、运输批及运输时点如表5所示. 结果的适应度值约为33.1, 总成本约为302115元. 由计算结果可知, 算例将订单和供应商的剩余资源做了有效的匹配. 为了进一步讨论本文的应用价值, 对比了算例订单拆分前后成本和资源利用率的情况.

表3 各供应商分配到的订单比例 $\alpha_{j,i}$

| | B_1 | B_2 | B_3 | B_4 | B_5 | B_6 | B_7 | B_8 | B_9 | B_{10} | B_{11} | B_{12} |
|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|----------|----------|----------|
| M_1 | — | — | — | — | — | — | — | 0.241 | 0.306 | — | — | 0.676 |
| M_2 | — | — | 0.218 | — | — | — | — | — | — | — | — | — |
| M_3 | — | — | 0.165 | — | — | 0.244 | 0.43 | 0.243 | — | — | 0.358 | — |
| M_4 | — | 0.187 | — | 0.667 | — | 0.199 | — | — | 0.349 | — | — | — |
| M_5 | 0.205 | 0.312 | 0.256 | — | 0.303 | 0.154 | — | 0.29 | 0.125 | — | — | 0.324 |
| M_6 | — | 0.139 | — | — | — | 0.403 | — | — | 0.22 | 0.206 | 0.357 | — |
| M_7 | — | 0.362 | 0.151 | 0.167 | 0.383 | — | 0.231 | 0.227 | — | 0.45 | — | — |
| M_8 | 0.795 | — | 0.21 | 0.166 | 0.315 | — | 0.339 | — | — | 0.343 | 0.284 | — |

表4 各供应商对应的租用工时 $TR_{i,j} / h$

| | B_1 | B_2 | B_3 | B_4 | B_5 | B_6 | B_7 | B_8 | B_9 | B_{10} | B_{11} | B_{12} |
|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|----------|----------|----------|
| M_1 | — | — | — | — | — | — | — | 27 | 37 | — | — | 68 |
| M_2 | — | — | 33 | — | — | — | — | — | — | — | — | — |
| M_3 | — | — | 25 | — | — | 32 | 37 | 27 | — | — | 51 | — |
| M_4 | — | 21 | — | 107 | — | 26 | — | — | 42 | — | — | — |
| M_5 | 28 | 35 | 39 | — | 37 | 21 | — | 32 | 15 | — | — | 33 |
| M_6 | — | 16 | — | — | — | 53 | — | — | 27 | 27 | 50 | — |
| M_7 | — | 41 | 23 | 27 | 46 | — | 20 | 25 | — | 59 | — | — |
| M_8 | 109 | — | 32 | 27 | 38 | — | 29 | — | — | 45 | 40 | — |

表5 供应商对应订单开完工时间、运输批及运输时点

| M_j | A_k | 承接订单排序 | 开工-完工时间 | 运输批 / 占容 / 运输时点 |
|-------|-------|--------------------|---------------------|--------------------------------|
| M_1 | A_1 | — | — | — |
| | A_2 | B_9 | 0-37 | $(B_8, B_9) / 8 / 37$ |
| | A_3 | B_8, B_{12} | 0-27, 27-95 | $(B_{12}) / 6 / 95$ |
| | A_4 | — | — | — |
| M_2 | A_1 | — | — | — |
| | A_2 | — | — | — |
| | A_3 | B_3 | 0-33 | $(B_3) / 3 / 33$ |
| | A_4 | — | — | — |
| M_3 | A_1 | B_{11}, B_6 | 0-51, 51-83 | — |
| | A_2 | — | — | $(B_8, B_7) / 9 / 37$ |
| | A_3 | B_8, B_3 | 0-27, 27-52 | $(B_{11}, B_3, B_6) / 10 / 83$ |
| | A_4 | B_7 | 0-37 | — |
| M_4 | A_1 | B_6 | 0-26 | — |
| | A_2 | B_9, B_2 | 0-42, 42-63 | $(B_6, B_9, B_2) / 9 / 63$ |
| | A_3 | — | — | $(B_4) / 10 / 107$ |
| | A_4 | B_4 | 0-107 | — |
| M_5 | A_1 | B_1, B_6 | 0-28, 28-49 | — |
| | A_2 | B_9, B_2, B_5 | 0-15, 15-50, 50-87 | $(B_9, B_1, B_8) / 10 / 32$ |
| | A_3 | B_8, B_3, B_{12} | 0-32, 32-71, 71-104 | $(B_6, B_2, B_3) / 9 / 71$ |
| | A_4 | — | — | $(B_5, B_{12}) / 7 / 104$ |
| M_6 | A_1 | B_{11}, B_6 | 0-50, 50-103 | — |
| | A_2 | B_9, B_2 | 0-27, 27-43 | $(B_9, B_{10}, B_2) / 9 / 43$ |
| | A_3 | — | — | $(B_{11}, B_6) / 9 / 103$ |
| | A_4 | B_{10} | 0-27 | — |
| M_7 | A_1 | — | — | — |
| | A_2 | B_2, B_5 | 0-41, 41-87 | $(B_7, B_8, B_2) / 10 / 41$ |
| | A_3 | B_8, B_3 | 0-25, 25-48 | $(B_3, B_{10}) / 10 / 79$ |
| | A_4 | B_7, B_{10}, B_4 | 0-20, 20-79, 79-106 | $(B_5, B_4) / 8 / 106$ |
| M_8 | A_1 | B_{11}, B_1 | 0-40, 40-149 | $(B_7, B_3) / 7 / 32$ |
| | A_2 | B_5 | 0-38 | $(B_5, B_{11}) / 8 / 40$ |
| | A_3 | B_3 | 0-32 | $(B_{10}, B_4) / 9 / 101$ |
| | A_4 | B_7, B_{10}, B_4 | 0-29, 29-74, 74-101 | $(B_1) / 8 / 149$ |

4.2.1 订单拆分前后成本分析

拆单后的优化结果与不拆单情形下的最优结果对比如表6所示(单位:元).可以看出:算例在拆单后完全满足订单生产需求,生产成本PC为280920元;而在不拆单情形下,供应商无法满足客户所有订单的需求,部分订单无法派单,在 B_1 订单未派单状况下,总生产成本PC为263550元,假设 B_1 订单在行业内最低生产价格210元状况下组织生产,总生产成本PC为292110元,在不考虑订单超期惩罚的前提下保守估计生产成本节约11190元.

表6 整单派单和拆单派单的成本对比

| 派单模式 | 生产成本 PC | 运输+库存成本 DC + IC | 总成本 TC | 总成本 节约 |
|------|------------|--------------------|-----------|-----------|
| 整单派单 | 292110 | 20456 | 312556 | — |
| 拆单派单 | 280920 | 21410 | 302330 | 10226 |

订单拆分后运输与库存成本的和为21410元,在不拆单情形下,运输与库存成本的和为20456元(订单 B_1 的运输成本预计为1200元左右).可以发现,两种派单模式的运输与库存成本的和差距不大,拆单模式并不会因为参与分配的供应商增多而增加运输和

库存成本.相反,当算例订单量增大时,通过多订单的组合可以使运输车在运输过程中尽可能多地占容,从而使运输成本大幅下降.

综合来看,总成本节约近10226元.更加有利的是,通过云平台中心化拆单,客户订单被全部满足,降低了企业的超期惩罚,保证了企业的交货期.

4.2.2 订单拆分前后设备剩余能力利用率分析

订单拆分可以使得有剩余资源能力的供应商尽可能地参与到云平台的分配中,如表7所示.拆单后所有供应商都从客户处获得了一定比例的订单,而在整单派单中 M_6 未获得订单.同时,拆单后除 M_2 外,所有供应商剩余能力的利用率均超过30%;而在整单派单中,仅有 M_2 、 M_5 、 M_7 三家的剩余能力利用率高于30%.

表7 整单派单和拆单派单的剩余能力利用率对比 %

| M_j | 整单派单剩余能力利用率 | 拆单后剩余能力利用率 | M_j | 整单派单剩余能力利用率 | 拆单后剩余能力利用率 |
|-------|-------------|------------|-------|-------------|------------|
| M_1 | 27.91 | 30.7 | M_5 | 53.60 | 49.48 |
| M_2 | 38.02 | 5.45 | M_6 | 0 | 39.32 |
| M_3 | 24.74 | 35.46 | M_7 | 73.90 | 43.42 |
| M_4 | 19.32 | 44.55 | M_8 | 29.80 | 68.09 |

在云平台的开放模式中, M_2 会意识到自身优势的缺乏,由于定价高于平台内其他供应商,自身又不具有距离优势,使得 M_2 的订单分配显著低于其他供应商.在云平台中心化模式中,供应商为了使闲置资源得到更高的利用,会在信息不断共享的过程中使得各自的报价趋于合理,在这种情况下,各供应商的剩余能力利用率会更加均衡.

4.3 算法性能对比

为了证明改进的遗传算法在模型求解中的有效性,本文将改进遗传算法IGA、CPLEX求解器与遗传算法GA的求解结果进行对比.IGA利用小生境的思想处理较差个体,使较优个体在选择操作中更具优势,保证了种群的多样性.IGA与GA的适应度收敛情况如图5所示,改进遗传算法IGA的效果要优于遗传算法GA.本文算例中,IGA的最优适应度值约为33.1,总成本约为302115元,生产成本约为280920元;利用CPLEX求得的最优解其生产成本为274030元,总成本为293834元,其成本差距小于3%,因此本文算法是有效的.由于本文问题为NP-hard问题,CPLEX在求解中、大规模问题上不具有优势,CPLEX不适用于订单量较大的情况,IGA的性能可以较好地满足云平台中跨供应商订单调度分配的需求.

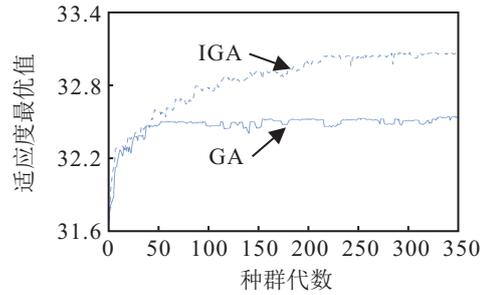


图5 改进前后算法各代适应度最优值收敛曲线对比

本文设计了不同问题规模($n/m/z$)下的问题集,由s-1到s-4共4个问题组成,对每一不同规模算例利用上述改进遗传算法(IGA)和遗传算法(GA)独立求解10次,最优解为10次测试中最优适应度最高的值,最差解为10次测试中最优适应度最低的值,平均值为10次测试中各次求得最优适应度的平均值,达优率为各次求解中所得的解与问题最优解相对偏差小于1%的比率,对比结果如表8所示.由表8可见,IGA的稳定性明显强于GA,在多次测试中的平均值和达优率更高,GA偶有最优解和达优率较高的情况,但其最优解和同样条件下其他解的差值较大,不具有一般性,同时其值与IGA相差很小.因此,经过测试,IGA的算法效果显著优于GA.

表8 两种算法求解不同规模问题的计算结果

| 编号 | $n/m/z$ | 最优解 | | 最差解 | | 平均值 | | 达优率 / % | |
|-----|---------|--------|--------|--------|--------|--------|--------|---------|----|
| | | IGA | GA | IGA | GA | IGA | GA | IGA | GA |
| s-1 | 4/4/8 | 688.35 | 688.47 | 658.01 | 635.92 | 669.65 | 652.30 | 20 | 10 |
| s-2 | 8/4/12 | 33.23 | 33.27 | 32.85 | 32.56 | 33.04 | 32.88 | 90 | 30 |
| s-3 | 4/5/25 | 98.25 | 96.23 | 95.42 | 93.83 | 96.77 | 95.1 | 40 | 50 |
| s-4 | 10/4/40 | 35.42 | 33.56 | 34.34 | 32.1 | 35.16 | 32.93 | 80 | 20 |

5 结论

本文提出了一种云平台主导的集成调度模式,基于该模式,构建了面向云平台中心化集成调度的跨供应商订单分配模型,给出了基于改进遗传算法的模型求解方法,最后通过算例演示了所构建模型和算法的有效性.主要贡献总结如下:

1) 分析和构建了一种中心化云平台调度模式,以云平台为主导,实现利用云的数据处理能力和计算优势使订单和闲置资源得到最优化的匹配,使云平台不仅有“信息分享”功能,还有“控制计算”优势;

2) 构建了跨供应商订单分配的模型与算法,将订单分配扩展到可拆单的情形,使云平台的参与主体更多元,可实现供应链双方的利益最大化,弥补了过往研究中将订单分配与生产排程、物流运输相割裂的问题;

3) 基于CPLEX、GA与IGA的算例仿真对比验证了模型及算法的有效性,详细分析了订单拆分前后成本和剩余能力利用率的数据,为云平台的运作完善与推广提供了实际应用的指导。

本文对各供应商服务能力做了简化,未能考虑相同订单不同供应商的服务能力差异.同时,单一订单对应多种服务类型的组合情形,以及组合中各服务进行跨企业的订单分配与调度等都将是未来可进一步研究的方向。

参考文献(References)

- [1] 李伯虎,张霖,王时龙,等.云制造——面向服务的网络化制造新模式[J].计算机集成制造系统,2010,16(1): 1-7.
(Li B H, Zhang L, Wang S L, et al. Cloud manufacturing: A new service—Oriented networked manufacturing model[J]. Computer Integrated Manufacturing Systems, 2010, 16(1): 1-7.)
- [2] Mishra N, Singh A, Kumari S, et al. Cloud-based multi-agent architecture for effective planning and scheduling of distributed manufacturing[J]. Int J of Production Research, 2016, 54(23): 7115-7128.
- [3] Helo P, Suorsa M, Hao Y, et al. Toward a cloud-based manufacturing execution system for distributed manufacturing[J]. Computers in Industry, 2014, 65(4): 646-656.
- [4] 齐二石,李天博,刘亮,等.云制造理论、技术及相关应用研究综述[J].工业工程与管理,2015,20(1): 8-14.
(Qi E S, Li T B, Liu L, et al. The review of cloud manufacturing theory, technologies and application states[J]. Industrial Engineering & Management, 2015, 20(1): 8-14.)
- [5] Jiang H, Yi J, Chen S, et al. A multi-objective algorithm for task scheduling and resource allocation in cloud-based disassembly[J]. J of Manufacturing Systems, 2016, 41: 239-255.
- [6] Zhang Y, Wang J, Liu S, et al. Game theory based realtime shop floor scheduling strategy and method for cloud manufacturing[J]. Int J of Intelligent Systems, 2017, 32(4): 437-463.
- [7] Ren L, Zhang L, Tao F, et al. Cloud manufacturing: From concept to practice[J]. Enterprise Information Systems, 2015, 9(2): 186-209.
- [8] 李伯虎,张霖,任磊,等.云制造典型特征、关键技术与应用[J].计算机集成制造系统,2012,18(7): 1345-1356.
(Li B H, Zhang L, Ren L, et al. Typical characteristics, technologies and applications of cloud manufacturing[J]. Computer Integrated Manufacturing Systems, 2012, 18(7): 1345-1356.)
- [9] Wu D, Rosen D W, Schaefer D. Scalability planning for cloud-based manufacturing systems[J]. J of Manufacturing Science and Engineering, 2015, 137(4): 040911-1-13.
- [10] 程海芳,张子刚.多供应商条件下集成供应商订货模型研究[J].工业工程与管理,2004,9(4): 27-30.
(Cheng H F, Zhang Z G. Integrated supplier's ordering model under multiple-supplier[J]. Industrial Engineering & Management, 2004, 9(4): 27-30.)
- [11] Fazlollahtabar H, Mahdavi I, Ashoori M T, et al. A multi-objective decision-making process of supplier selection and order allocation for multi-period scheduling in an electronic market[J]. Int J of Advanced Manufacturing Technology, 2011, 52(9/10/11/12): 1039-1052.
- [12] 徐辉,侯建明.需求不确定条件下的制造商订单分配模型[J].中国管理科学,2016,24(3): 80-88.
(Xu H, Hou J M. Manufacturer's order allocation model under demand uncertainty[J]. Chinese J of Management Science, 2016, 24(3): 80-88.)
- [13] 李占丞,刘晓冰,薄洪光.面向生产调度的订货量分配问题研究[J].工业工程与管理,2016,21(2): 32-40.
(Li Z C, Liu X B, Bao H G. Research on production scheduling oriented order quantity allocation problem[J]. Industrial Engineering & Management, 2016, 21(2): 32-40.)
- [14] 赵道致,李锐.考虑主体心理预期的云制造资源双边匹配机制[J].控制与决策,2017,32(5): 871-878.
(Zhao D Z, Li R. Two-sided matching mechanism with agents' expectation for cloud manufacturing resource[J]. Control and Decision, 2017, 32(5): 871-878.)
- [15] Liu Y, Xu X, Zhang L, et al. Workload-based multi-task scheduling in cloud manufacturing[J]. Robotics and Computer-Integrated Manufacturing, 2016, 45(C): 3-20.
- [16] Cao Y, Wang S, Kang L, et al. A TQCS-based service selection and scheduling strategy in cloud manufacturing[J]. Int J of Advanced Manufacturing Technology, 2016, 82(1/2/3/4): 1-17.
- [17] 王贞,张纪会,齐元青.具有空闲时间的云制造作业车间调度方法[J].控制与决策,2017,32(5): 811-816.
(Wang Z, Zhang J H, Qi Y Q. Job shop scheduling method with idle time in cloud manufacturing[J]. Control and Decision, 2017, 32(5): 811-816.)
- [18] Feng H, Xia T, Da W, et al. Concurrent design of cell formation and scheduling with consideration of duplicate machines and alternative process routings[J]. J of Intelligent Manufacturing, 2019, 30(1): 275-289.

作者简介

谢乃明(1981—),男,教授,博士,从事灰色系统、生产运作管理等研究, E-mail: xienaiming@nuaa.edu.cn;

吴乔(1995—),女,硕士生,从事生产计划与控制的研究, E-mail: Wuqiaonuaa@163.com;

郑绍祥(1995—),男,硕士生,从事生产计划与控制的研究, E-mail: zhengshaoxiang@nuaa.edu.cn.

(责任编辑:郑晓蕾)