考虑策略性等待的并行制造系统云订单动态接收优化

侯林辉[1]，孙晓晨[1]，李勇建[2†]

(1. 天津大学 管理与经济学部，天津 300072；2. 南开大学 商学院，天津 300071)

摘　要: 云制造环境下, 制造商面临随机、动态、异质的在线订单流, 常在即时响应当前订单与保留产能以等待未来潜在高价值订单之间进行动态权衡, 以实现长期利润最大化. 当前针对随机订单接收问题的研究多基于确定性假设或短视规则, 难以有效量化等待的机会价值, 且现有精确动态规划方法受限于“维数灾难”. 研究具有策略性等待的多机并行制造系统随机动态订单接收问题, 提出一种特征驱动的近似动态规划方法 (FD-ADP), 将“策略性等待”建模为状态依赖的主动决策机制, 通过“资源-负载-时序”的特征压缩与价值导向的等待准则, 在保持策略可解释性的同时高效逼近最优解. 数值实验表明, FD-ADP 相较于精确动态规划平均利润损失率仅为 4.85%, 大幅降低计算时间, 且利润表现优于其他对比算法. 研究进一步揭示, 策略性等待机制在不同订单结构和需求模式下均保持较好的适应性. 所提出方法为云制造中应对随机订单冲击的动态订单接收提供了兼具计算效率、可解释性与近似最优性的解决方案.

关键词: 策略性等待；近似动态规划；订单接收；云制造；并行制造系统；随机订单到达

中图分类号: TP301　　**文献标志码:** A

DOI: 10.13195/j.kzyjc.2026.0531

引用格式: 侯林辉, 孙晓晨, 李勇建. 考虑策略性等待的并行制造系统云订单动态接收优化 [J]. 控制与决策.

Optimization of dynamic order acceptance in parallel manufacturing systems with strategic waiting in cloud manufacturing

HOU Lin-hui[1]，*SUN Xiao-chen*[1]，*LI Yong-jian*[2†]

(1. College of Management and Economics，Tianjin University，Tianjin 300072，China；2. Business School，Nankai University，Tianjin 300071，China)

Abstract: Online order acceptance in cloud manufacturing faces heterogeneous and stochastic order streams with real-time arrivals. Manufacturers must dynamically trade off accepting current orders and reserving capacity for potential high-value orders to maximize long-term profit. This paper focuses on the perspective of manufacturers in cloud manufacturing, studying the problem of dynamic order acceptance with strategic waiting in a multi-machine parallel manufacturing system under random order arrivals. Existing studies largely rely on deterministic assumptions or myopic rules that fail to quantify the value of waiting; exact dynamic programming further suffers from the curse of dimensionality in large-scale settings. We propose a feature-driven approximate dynamic programming (FD-ADP) approach that models strategic waiting through state-dependent value approximation using compressed resource-load-time features. Numerical results show that FD-ADP limits the average profit gap to 4.85%, while outperforming exact dynamic programming in efficiency and other approximate methods in profit. Strategic waiting remains robust across order and demand settings and depends on system load, time, and arrival rates. FD-ADP thus offers an efficient, interpretable, near-optimal solution.

Keywords: strategic waiting；approximate dynamic programming；order acceptance；cloud manufacturing；parallel manufacturing systems；stochastic order arrivals

0　引　言

随着云计算与工业互联网技术的发展, 云制造为制造能力的服务化提供了新的组织与运行模式[1]. 订单需求在云平台在线持续发布与更新, 制造商通过开放订单池获取生产订单. 例如, 服装制造云平台东纺云工厂通过订单滚动更新机制及周度定向推送策略, 为入驻制造商构建了充沛且需求稳定的订单池. 与以长期合同或计划订单为主的传统制造不同,

收稿日期: 2026-06-03；**录用日期:** 2026-09-10.
基金项目: 国家自然科学基金重点项目 (72132007).
†通信作者. E-mail: liyongjian@nankai.edu.cn.

云平台订单由不同客户动态发起, 难以提前预知其属性与价值, 使制造商面临的订单流具有随机性、属性异质性及高度可选择性的特征. 制造商须依据实时系统状态, 对新到达订单做出即时且不可逆的接收或拒绝决策.

制造商进行上述决策时常陷入一个关键权衡: 是立即使用空闲产能承接当前订单池中的订单, 还是主动保留部分产能以等待未来潜在的高价值订单. 本文将这种基于机会价值评估的主动产能保留行为称为策略性等待. 以 Craftcloud 等增材制造云服务平台为例, 制造商可在需求稳定的标准件任务与需求较少但具有定制溢价的个性化定制任务之间进行选择. 标准订单有助于维持设备利用率和短期现金流, 定制订单虽随机到达且加工复杂, 却能为制造商带来定制溢价. 因此, 制造商需要在限时决策的情形下, 实时决定是否接收当前订单池中的订单, 或保留产能以等待未来潜在的定制订单, 使订单接收问题从以往的阶段性计划问题转变为具有严格时效要求的在线决策问题. 本文研究云制造背景下考虑策略性等待的动态订单接收问题: 在随机异质订单流和多机资源约束下, 如何通过状态依赖的订单接收与策略性等待实现长期期望利润最大化.

订单接收研究主要关注有限产能下的订单筛选与利润优化, 相关研究已从静态或确定性环境下的整数规划和滚动时域方法[2], 拓展至考虑风险传导、交货违约和产能配置的决策模型[3,4]. 随着云制造的发展, 研究进一步涉及跨供应商订单分配[5]、订单接收—排单协同[6]、服务组合优化[7-8] 及采用启发式算法或强化学习等方法改善随机在线订单接收中的决策与调度性能[9-10]. 然而, 现有研究多侧重订单到达后的资源配置、可行调度或即时收益, 且常基于静态订单集、单机或既定调度任务, 尚未充分量化多机动态状态下主动保留产能的机会价值.

策略性等待与收益管理中的产能保护和容量预留思想密切相关[11-12], 这些研究为将等待理解为主动产能管理行为提供了理论依据. 但服务系统中的产能通常较为同质, 而离散制造产能表现为随设备占用和计划期推进动态演化的“机器-时间”资源. 已有延迟调度研究主要关注完工时间或拖期[13], 近似动态规划研究则表明保留未来决策弹性有助于改善长期绩效[14], 但现有研究尚未将策略性等待建模为状态依赖的主动控制行为, 缺乏一个能够动态评估其价值、并据此做出序贯决策的方法, 这正是本文旨在解决的核心问题.

为处理此类序贯决策问题, 随机订单接收问题常被建模为马尔可夫决策过程[15], 但精确动态规划在多机、长周期条件下面临“维数灾难”. 为克服这一瓶颈, 近似动态规划通过状态压缩、价值近似和自适应探索, 提供了有效的求解途径[16-17], 并已应用于资源约束下的作业调度优化[18] 和订单的动态定价问题[19]. 然而, 现有方法尚缺少面向在线订单接收中策略性等待的系统建模及状态特征设计. 为此, 本文提出特征驱动的近似动态规划 (Feature-Driven Approximate Dynamic Programming, FD-ADP) 方法, 从资源可用性、负载分布和时序演进三个维度进行特征压缩, 动态评估保留产能的机会价值, 并据此形成状态依赖的策略性等待决策.

表 1 从五个维度归纳相关研究. 现有研究已在订单接收与调度、云制造资源匹配、产能保留和 ADP 随机资源调度等方面具有一定基础, 但尚未将云制造平台中的随机异质订单到达、多机动态产能状态与策略性等待决策纳入统一序贯决策框架. 本文以云制造环境下制造商长期期望利润最大化为目标, 构建了考虑策略性等待行为的在线并行机动态订单接收随机优化模型, 并提出一种特征驱动的近似动态规划方法.

表 1 结构化文献评述表

研究类别	代表文献	随机订单到达	多机资源约束	等待/产能保留	动态规划/ADP求解	云制造应用
订单接收与调度	Gourgand等[2]; 于超等[3]; Wang等[4]; Abedi等[20]; Li和Ventura[21]	×	√	×	√	×
云制造资源匹配与订单调度	Wu等[22]; 丁祥海等[6]; 张晓冬等[7]; 谢乃明等[5]	√	√	×	×	√
在线订单接收与实时决策	Zheng等[9]; Arredondo等[10]	√	×	×	√	×
产能保留与收益管理	Zhang等[11]; Chen等[12]	√	×	√	√	×
ADP与动态随机资源调度	Powell[16]; Satic等[18]; 温广辉等[17]; Bertsekas[23]; Bertsekas[24]	√	√	×	√	×
本文	本文	√	√	√	√	√

本文的主要贡献如下: 第一, 将云制造订单接收刻画为异质订单随机到达下的在线序贯决策问题, 构建包含多机资源约束与决策不可逆性的随机动态规划模型, 弥补了现有研究中确定性假设和短视规则对云制造开放订单池的动态订单特征刻画不足的问题. 第二, 将策略性等待建模为状态依赖的主动决

策行为, 通过策略对比与扩展实验量化其长期收益, 揭示主动保留产能的经济价值及适用条件, 拓展了传统订单接收研究中设备空闲即浪费的研究视角[4,10]. 第三, 构建融合资源、负载和时序信息的状态特征及价值导向等待准则, 提出兼顾求解效率与策略可解释性的特征驱动近似动态规划方法. 数值实验表明, 该方法相较精确动态规划的平均利润损失率仅为 4.85%, 且利润表现优于其他对比算法.

1 问题描述与建模

考虑一个拥M台并行制造设备的云制造商, 设备编号$m \in \{1,2,\ldots,M\}$. 将一个有限的生产计划期分为T个时期, 每个时期$t \in \{1,2,\ldots,T\}$足够短, 以确保每台设备在每个时期最多只能被分配一个订单.

云平台订单池中订单持续发布与更新, 令$\mathcal{O}_t$为在t时期订单池中的订单集合. 我们将制造商在云平台订单池中需要处理的订单分为两类: 一类是标准订单, 其需求稳定、工艺流程成熟且高度标准化, 具有加工周期短、收益透明但相对较低的特点; 另一类是定制订单, 其需求随机, 需根据客户的个性化要求进行设计、工艺适配或跨组织协同制造, 往往工艺结构更复杂、加工周期更长, 但因其带来的定制化溢价, 定制订单的期望利润率较高. 令在t时期标准订单集合为$\mathcal{J}_t$、定制订单集合为$\mathcal{C}_t$, 则$\mathcal{O}_t = \mathcal{J}_t \cup \mathcal{C}_t$.

集合$\mathcal{O}_t$中的任意订单o的收益、加工时长、单位时间加工成本、资源占用系数及协调成本等订单属性均在到达时刻t动态生成. 具体而言, 若在t时刻订单o为标准订单 ($o \in \mathcal{J}_t$), 其收益为$r^J_{t,o}$, 单位时间加工成本为$k^J_{t,o}$, 加工时长为$p^J_{t,o}$. 若在t时刻订单o为定制订单 ($o \in \mathcal{C}_t$), 其收益为$r^C_{t,o}$, 单位时间加工成本为$k^C_{t,o}$, 加工时长为$p^C_{t,o}$. 定制订单因其个性化需求, 通常涉及更复杂的设计沟通、工艺适配及生产准备等, 其单位时间加工成本与加工时长高于标准订单, 且两类订单的加工时长均为单位时间的整数倍. 因定制订单的特殊工艺要求、复杂结构等特性, 其单位加工时间内对设备能耗、工具磨损等设备资源的消耗强度比标准订单更高, 设标准订单的资源占用系数为基准值$\beta^J_{t,o} = 1$, 定制订单资源占用系数$\beta^C_{t,o} > 1$. 此外, 定制订单通常需要借助平台设计、仿真等服务或与平台上的合作商协作生产, 会产生单位时间协调成本$\omega^C_{t,o} > 0$, 标准订单无额外协调成本, 即$\omega^J_{t,o} = 0$. 每个时期订单池中订单随机到达, 且订单的收益、单位时间加工成本等订单属性均在其所属订单类型对应的取值区间内随机生成 (具体参数分布见第 3 节数值实验设置).

在上面的情景下, 云制造商的决策问题为: 面对持续更新的订单池$\mathcal{O}_t$, 每个周期需决定从$\mathcal{O}_t$中接收哪些订单, 并为其指定生产设备, 使T个时期的总期望利润达到最大. 问题的核心运营困境在于: 持续选择标准订单可避免设备闲置, 但可能因产能占用错失定制订单; 选择等待定制订单虽有机会获取高收益, 却需承担设备空闲的短期收益损失风险. 制造商需在随机订单流和多机环境下, 量化等待定制订单的机会价值, 动态权衡即时利润与未来期望价值, 并据此做出序贯决策.

进而, 结合实际背景, 对该问题做如下假设: 云制造环境下订单池中订单需求充足, 在每个决策时刻动态更新, 所有订单必须在到达时立即接收或拒绝, 不允许暂存; 不考虑订单提前期, 每个决策时刻到达并接收的订单将被立刻安排生产; 每台机器在同一时间至多处理一个订单, 且订单处理过程不可中断; 设备具备通用性, 可处理任意类型订单, 且订单间切换无额外成本; 动态订单接收过程中需满足资源约束与时序可行性, 且不允许订单延期或积压.

为方便, 将本文所使用的相关符号及定义列于表 2.

表 2 符号定义表

符号	定义
M	设备数量
m	设备编号, $m \in \{1,2,\ldots,M\}$
T	计划期长度
t	离散决策时刻, $t \in \{1,2,\ldots,T\}$
$\mathcal{J}_t \cup \mathcal{C}_t$	t时刻订单池中的标准订单集合、定制订单集合
$\mathcal{O}_t$	t时刻订单池中的订单集合, $\mathcal{O}_t = \mathcal{J}_t \cup \mathcal{C}_t$
$r^J_{t,o}, r^C_{t,o}$	t时刻标准订单、定制订单的收益
$p^J_{t,o}, p^C_{t,o}$	t时刻标准订单、定制订单的生产时长
$k^J_{t,o}, k^C_{t,o}$	t时刻标准订单、定制订单的单位时间加工成本
$\beta^J_{t,o}, \beta^C_{t,o}$	t时刻标准订单、定制订单的单位时间资源占用系数
$\omega^J_{t,o}, \omega^C_{t,o}$	t时刻标准订单、定制订单的单位时间协调成本
$\tau_{t,m}$	设备m在t时刻的预计最早可用时间
τ_t	t时刻的系统状态向量, $\tau_t = (\tau_{t,1},\ldots,\tau_{t,M})$
$V(\tau_t,\pi)$	从 t时刻状态τ_t至计划期末采取策略π的期望总利润
$R(\tau_t,x_t)$	t时刻状态为τ_t时采取行动x_t后所获得的即时利润
$x_{t,o,m}$	t时刻设备m接收第o个订单, 则取值为1, 否则为0
x_t	t时刻的动作矩阵, $x_t = [x_{t,o,m}]_{\lvert\mathcal{O}_t\rvert\times M}$

针对上面的决策问题, 本文构建一个基于随机动态规划的数学模型进行求解和分析. 针对动态规划模型构建所需的状态、决策、状态转移过程和目标函数等要素, 下面分别描述.

状态变量由设备状态定义, 记为$\tau_t=(\tau_{t,1},\ldots,\tau_{t,M})$. τ_t表示设备状态, $\tau_{t,m}$表示设备m在时刻t开始时预计可用的最早时间.

决策变量定义为$x_{t,o,m}\in\{0,1\}$, 表示是否在t时刻接收订单o并分配给机器m, 若接收则取值为 1, 否则为 0. 动作矩阵记为$x_t=[x_{t,o,m}]_{|\mathcal{O}_t|\times M}$. 在时刻$t$, 制造商接收的订单总数不得超过空闲机器数, 即$\sum\limits_{o=1}^{|\mathcal{O}_t|}\sum\limits_{m=1}^{M}x_{t,o,m}\leqslant\sum\limits_{m=1}^{M}\mathbb{I}\{\tau_{t,m}=t\}$. 同时, 对任一订单$o$, 在任一时刻$t$至多只能被分配给一台设备, 即$\sum\limits_{m=1}^{M}x_{t,o,m}\leqslant 1$; 对于任意一台设备$m$, 在任一时刻$t$至多只能接收一个新订单, 即$\sum\limits_{o=1}^{|\mathcal{O}_t|}x_{t,o,m}\leqslant 1$.

状态转移过程由当前状态及所采取的决策共同驱动. 每个决策时刻t, 系统根据当前状态τ_t从订单池中选择订单并分配给设备, 且所接收订单必须在计划期内完成; 若不存在满足该条件的订单, 则停止接收. 随后, 系统依据新接收订单的生产时长更新设备状态, 得到下一时刻的系统状态τ_{t+1}.

$$150\tau_{t+1,m}=\begin{cases}\tau_{t,m}+\max(\sum\limits_{o=1}^{|\mathcal{O}_t|}(p_{t,o}^{J}\mathbb{I}\{o\in\mathcal{J}_t\}+\\ \qquad p_{t,o}^{C}\mathbb{I}\{o\in\mathcal{C}_t\})x_{t,o,m},1),\\ \qquad\tau_{t,m}=t,\ \forall m,\\ \tau_{t,m},\tau_{t,m}>t,\ \forall m,\end{cases}\tag{1}$$

其中, $\mathbb{I}\{.\}$为指示函数, 当括号中内容为真时取 1, 否则为 0. $\mathbb{I}\{o\in\mathcal{J}_t\}=1$和$\mathbb{I}\{o\in\mathcal{C}_t\}=1$分别表示订单$o$为标准订单和定制订单. 当设备$m$在时刻$t$空闲$(\tau_{t,m}=t)$时, 若接收新订单, 其下一时刻预计最早可用时间按订单处理时长顺延; 若未接收订单, 则下一时刻预计最早可用时间为当前预计最早可用时间t加 1. 当设备m在时刻t尚未空闲, 即$\tau_{t,m}>t$, 则其下一时刻预计最早可用时间保持不变.

目标函数为制造商在整个计划期T内获得的长期期望总利润. 给定初始状态$\tau_1=(0,\ldots,0)$和策略$\pi=(x_1,x_2,\ldots,x_T)$, 制造商整个计划期T内的期望总利润为$V(\tau_1,\pi)=\mathbb{E}^{\pi}[\sum\limits_{t=1}^{T}\gamma^{t-1}R(\tau_t,x_t)]$. 其中, γ为折扣因子, 用于权衡即时利润与未来长期价值. $R(\tau_t,x_t)$代表在时刻t状态为τ_t时采取行动x_t后所获得的即时利润, 是时刻t累加被接收订单$(x_{t,o,m}=1)$的净利润, 计算公式如下:

$$\begin{aligned}R(\tau_t,x_t)=&\sum_{o=1}^{|\mathcal{O}_t|}\sum_{m=1}^{M}(r_{t,o}^{C}-(\omega_{t,o}^{C}+\beta_{t,o}^{C}k_{t,o}^{C})\\&p_{t,o}^{C})\mathbb{I}\{o\in\mathcal{C}_t\}x_{t,o,m}+\sum_{o=1}^{|\mathcal{O}_t|}\sum_{m=1}^{M}(r_{t,o}^{J}-\\&\beta_{t,o}^{J}k_{t,o}^{J}p_{t,o}^{J})\mathbb{I}\{o\in\mathcal{J}_t\}x_{t,o,m}\end{aligned}\tag{2}$$

公式 (2) 中的两项分别对应定制订单与标准订单的净利润, 均由基础收益减去生产过程中的综合成本构成. 综合成本包括由加工时长、单位时间加工成本及资源占用系数共同决定的设备资源消耗成本, 以及定制订单特有的额外协调成本. $T+1$时刻计划期结束, 对于所有的系统状态τ_{T+1}, 状态价值都为 0.

最优化问题: 给定初始状态τ_1, 找到一个最优策略π^*, 使得整个计划期T的期望总利润最大, 即:

$$V^*(\tau_1,\pi^*)=\max_{\pi}\mathbb{E}^{\pi}[\sum_{t=1}^{T}\gamma^{t-1}(R(\tau_t,x_t))],\tag{3}$$

并且满足如下条件:

$$\sum_{o=1}^{|\mathcal{O}_t|}\sum_{m=1}^{M}x_{t,o,m}\leqslant\sum_{m=1}^{M}\mathbb{I}\{\tau_{t,m}=t\},\ \forall t,\tag{4}$$

$$\sum_{o=1}^{|\mathcal{O}_t|}x_{t,o,m}\leqslant 1,\ \forall t,m,\tag{5}$$

$$\sum_{m=1}^{M}x_{t,o,m}\leqslant 1,\ \forall t,o\in\mathcal{O}_t,\tag{6}$$

$$\begin{aligned}&t+\sum_{o=1}^{|\mathcal{O}_t|}(p_{t,o}^{J}\mathbb{I}\{o\in\mathcal{J}_t\}+p_{t,o}^{C}\mathbb{I}\{o\in\mathcal{C}_t\})x_{t,o,m}\leqslant\\&\quad T,\ \forall t,m,\end{aligned}\tag{7}$$

$$|\mathcal{J}_t|\geqslant M,\ \forall t,\tag{8}$$

其中, 公式 (4)、(5)、(6) 分别表示确定决策变量取值时要满足的产能约束、订单唯一性约束和设备容量约束. 公式 (7) 表示系统不允许订单延期或跨计划期积压, 接收的订单需要在计划期内完成. 公式 (8) 表示标准订单的数量不少于制造商的设备数量, 以满足订单池中订单需求充足的假设.

2 特征驱动的近似动态规划方法

理论上, 目标函数 (3) 可以用 Bellman 方程求解:

$$V_t(\tau_t)=\max_{x_t}[R(\tau_t,x_t)+\gamma\mathbb{E}[V_{t+1}(\tau_{t+1})]],\ \forall t,\tau_t,\tag{9}$$

$V_t(\tau_t)$表示在状态为τ_t时, 从t时刻到整个计划期T结束的最优期望总利润, 边界方程为$V_{T+1}(\tau_{T+1})=0$. 但在实践中, 它面临“维数灾难”问题. “维数灾难”意味着状态的数量和计算需求随着状态变量的

数量呈现指数级增长[16]. 现有研究已充分讨论了传统精确动态规划 (Dynamic Programming, DP) 方法的局限性, 复杂制造调度通常需要借助近似评估以降低计算开销[25-26]. DP 方法在求解本文的问题时, 面临$O(Z^M T^{M+1})$的计算复杂度, 其中$Z = \max_{t\in T}|\mathcal{O}_t|$为订单池中最大订单数量. 随着设备数或计划期增加, DP 的计算复杂度迅速膨胀, 难以求解大规模问题实例. 为克服此瓶颈, 本节提出一种特征驱动的近似动态规划方法, 简称 FD-ADP, 包含状态特征压缩与特征驱动的策略性等待机制两部分.

2.1 状态特征压缩

高维状态空间是制约精确动态规划应用于多机动态订单接收问题的核心瓶颈. 在实际动态决策过程中, 往往不需要掌握每台设备的精确状态, 而是依赖于若干反映系统整体能力的关键统计量[18]. 为了既解决维数灾难, 又保证策略可解释性, 本文提出遵循“资源可用性-负载分布-时序演进”的特征驱动状态压缩方法, 在每次迭代中, 从原始系统状态向量τ_t中提取与“策略性等待”决策相关的四个指标, 构成特征向量$\phi(\tau_t^n)$. 设计逻辑如下:

资源可用性维度用空闲设备数表示, 即$\sum_{m=1}^{M}\mathbb{I}\{\tau_{t,m}^n \leqslant t\}$. $\tau_{t,m}^n$为设备m第n次迭代中t时刻的设备状态. 该指标刻画了即时产能冗余并确定了可行决策空间. 仅当存在空闲设备时, 制造商才需权衡即时接单与策略性等待. 负载分布维度用最大剩余忙碌时间和平均剩余忙碌时间表示, 即$\max_m(\tau_{t,m}^n - t)^+$和$\frac{1}{M}\sum_{m=1}^{M}(\tau_{t,m}^n - t)^+$, 用于刻画设备资源的释放节奏、长期占用情况及整体负载. 负载过高或设备长期占用时, 未来定制订单可能因无法在计划期内完成而被拒绝. 该维度用于识别过度饱和状态, 抑制不合理的等待行为. 时序演进维度用时间进度比表示, 即$\frac{t}{T}$. 该指标刻画了计划期的时间紧迫度, 反映未来可用时间的变化, 使等待策略能够随计划期推进自适应调整.

$$\phi(\tau_t^n) = [\sum_{m=1}^{M}\mathbb{I}\{\tau_{t,m}^n \leqslant t\}, \max_m(\tau_{t,m}^n - t)^+, \frac{1}{M}\sum_{m=1}^{M}(\tau_{t,m}^n - t)^+, \frac{t}{T}], \tag{10}$$

第n次迭代中t时刻的特征函数$\phi(\tau_t^n)$见公式 (10). 上述四个特征在等待价值判断中具有互补作用. 空闲设备数给出当前可用于接单或保留的产能边界. 最大剩余忙碌时间反映局部设备被长期占用的极端程度, 平均剩余忙碌时间刻画系统整体负载. 时间进度比则为相同负载状态赋予不同的剩余决策窗口. 该特征组合的目的并非完整重构原始设备状态, 而是保留与未来产能释放、订单可完成性及等待机会成本相关的信息, 为后续状态聚合提供具有实际含义的低维表示. 该方法将随机器数量增长的高维状态映射至四维特征空间, 缓解“维数灾难”[16], 并以具有物理意义的特征保留关键信息, 保证策略可解释性.

2.2 特征驱动的策略性等待机制

本节提出特征驱动的近似动态规划方法的核心——特征驱动的策略性等待机制, 包括四个关键步骤: 构建基于特征聚合的近似值函数; 生成组合动作并剪枝; 依据价值导向的等待准则权衡即时收益与未来机会, 并结合自适应探索做出动作决策; 通过自适应平滑更新特征近似值, 进行在线学习与策略优化. 具体实现如下:

2.2.1 基于特征聚合的值函数近似器

本文构建一个基于特征聚合的值函数近似器: 通过特征映射函数$\Phi(\tau_t^n)$将高维状态空间离散化为有限的特征集合, 每个$\Phi(\tau_t^n)$对应一个特征近似值$\hat{V}(\Phi(\tau_t^n))$. 在算法迭代初期, 所有特征近似值$\hat{V}(\Phi(\cdot))$初始化为 0; 随后, 通过时序差分学习机制与在线经验数据, 对每个特征近似值进行增量式更新, 逐步逼近其最优价值. 这种粗粒度状态聚合, 既避免了预设函数形式的局限性, 又能捕捉特征组合下的非线性价值关系, 通过牺牲部分状态分辨度来换取计算可行性和学习效率[23-24]. 该方法的核心思想是状态聚合: 将具有相同或相似特征向量的不同原始状态视为等价状态, 共享同一个值函数估计. 通过更新特征近似值, 可以在有限特征空间内刻画不同特征组合对应的状态价值差异, 在降低状态空间规模的同时保留主要决策信息.

2.2.2 组合动作生成与决策空间剪枝

以第n次迭代为例, 算法需在决策时刻从当前状态的可行动作中选择最优动作. 为压缩决策空间, 提出组合动作优化模型, 通过两步骤筛选候选动作:

第一步, 为每台空闲设备生成满足计划期约束的可选订单集, 形式如下:

$$A_{t,m}^n = \{o|o \in \mathcal{O}_t^n, t + p_{t,o}^J\mathbb{I}\{o \in \mathcal{J}_t^n\} + p_{t,o}^C\mathbb{I}\{o \in \mathcal{C}_t^n\} \leqslant T\} \cup \{\varnothing\},\ \forall m \in M_{t,idle}^n, \tag{11}$$

其中, $A_{t,m}^n$代表第n次迭代中t时刻设备m在满足计划期可行性约束下的设备可选订单集合. $M_{t,idle}^n = \{m \in \{1,\ldots,M\}|\tau_{t,m}^n \leqslant t\}$为第$n$次迭代中$t$时刻的

空闲设备集.

第二步, 对所有空闲设备的可选订单集取笛卡尔积, 生成组合动作集, 形式如下:

$$X^n_{t,comb} = \{a^n_t = (a^n_{t,m})_{m\in M^n_{t,idle}} \in \prod_{m\in M^n_{t,idle}} A^n_{t,m} | \Psi(a^n_t) = 1\}, \tag{12}$$

$$\Psi(a^n_t) = \mathbb{I}\{(\forall m \in M^n_{t,idle}, \sum_{o\in\mathcal{O}^n_t} \mathbb{I}\{a^n_{t,m} = o\} \leqslant 1) \wedge (\forall o \in \mathcal{O}^n_t, \sum_{m\in M^n_{t,idle}} \mathbb{I}\{a^n_{t,m} = o\} \leqslant 1)\}, \tag{13}$$

其中, $X^n_{t,comb}$表示第n次迭代中t时刻所有空闲设备的有效组合动作集合, $\prod_{m\in M^n_{t,idle}} A^n_{t,m}$表示对所有空闲设备的可选订单集$A^n_{t,m}$进行笛卡尔积运算, 生成原始的联合动作空间. a^n_t表示第n次迭代中t时刻所有空闲设备的一个联合动作向量, 其维度由空闲设备数量$|M^n_{t,idle}|$决定. 其中, 分量$a^n_{t,m} \in A^n_{t,m}$代表分配给机器m的具体动作, $a^n_{t,m} = o(o \in \mathcal{O}^n_t)$和$a^n_{t,m} = \varnothing$分别表示第$n$次迭代中$t$时刻设备$m$接收订单$o$或主动等待. 前者对应决策变量$x^n_{t,o,m} = 1$, 后者表示设备$m$对所有订单$o$的决策$x^n_{t,o,m} = 0$, 非空闲设备$m$对任一订单的决策$x^n_{t,o,m} = 0$. 公式 (13) 为有效性函数$\Psi(a^n_t)$, 保证每台设备至多接收一个订单, 且每个订单至多分配给一台设备.

传统全局调度需同时为所有设备分配订单, 本文仅对当前空闲设备集构造笛卡尔积, 将动作空间规模的上界从$O(Z^M)$降低到$O(Z^H)$, 其中$Z = \max_{t\in T}|\mathcal{O}^n_t|$为最大可选订单数, $H = \max_{t\in\{2,\dots,T\}}|M^n_{t,idle}|$为最大空闲设备数. 由于动态接单过程中通常有$H < M$, 该方法可有效缓解组合动作枚举的计算压力.

2.2.3 价值导向的等待准则与自适应探索

获得有效组合动作集合后, 算法依据第n次迭代的特征近似值进行决策. 本文将等待视为具有机会价值的主动行为: 当且仅当所有当前可用订单带来的长期利润均低于等待的机会价值——即等待所保留的未来状态价值时, 选择等待. 否则, 执行长期利润最大的动作, 即$\arg\max_{x^n_t\in X^n_{t,comb}\setminus X^n_{t,wait}} (R(\tau^n_t, x^n_t) + \gamma\mathbb{E}[\hat{V}^n(\Phi(\tau^n_{t+1}))])$.

$$Wait \Leftrightarrow \max_{x^n_t\in X^n_{t,wait}} (R(\tau^n_t, x^n_t) + \gamma\mathbb{E}[\hat{V}^n(\Phi(\tau^n_{t+1}))]) \geqslant \max_{x^n_t\in X^n_{t,comb}\setminus X^n_{t,wait}} (R(\tau^n_t, x^n_t) + \gamma\mathbb{E}[\hat{V}^n(\Phi(\tau^n_{t+1}))]), \tag{14}$$

其中, $X^n_{t,wait}$是第n次迭代t时刻等待动作集合, 为$\{x^n_t \in X^n_{t,comb} | \exists m \in M^n_{t,idle}, \forall o \in \mathcal{O}^n_t: x_{t,o,m} = 0\}$, 即至少保留一台空闲设备不接单的组合动作. 等待准则两侧均由状态特征决定, 因而可根据系统负载和时间进度评估产能保留的机会价值.

为降低随机环境下陷入局部最优的风险, 本文引入ϵ-贪心策略[17]. 考虑到在较大状态空间和复杂奖励结构下, 探索强度同时受迭代进程和状态可探索性影响[27-28], 在价值导向等待决策中进一步设计基于资源与时间状态的自适应探索率ϵ^n_t. 公式如下:

$$\epsilon^n_t = \epsilon_n \frac{\sum_{m=1}^{M} \mathbb{I}\{\tau^n_{t,m} \leqslant t\}}{M}(1 - \frac{t}{T}), \tag{15}$$

$$\epsilon_n = \max(\epsilon_{min}, \epsilon_0\sigma^n), \tag{16}$$

其中, 第n次迭代中t时刻的动态探索率ϵ^n_t、空闲机器比例与剩余计划期比例共同决定: 当系统空闲资源与剩余时间较充足时, 增大探索率以发现潜在的优质策略; 当空闲资源或剩余时间不足时, 相对降低探索率, 收敛至已知最优策略以规避风险. 基础探索率ϵ_n随迭代次数按系数σ指数衰减, ϵ_{min}为其下限 (设为 0.01), ϵ_0为初始探索率.

2.2.4 特征近似值的自适应平滑更新

本文通过基于特征的值函数近似缓解状态空间的维数灾难, 采用时序差分学习 (Temporal Difference Learning, TD) 增量更新特征近似值. 第n次迭代中, 在决策时观察到状态τ^n_t, 执行动作x^n_t后获得即时利润$R(\tau^n_t, x^n_t)$, 并转移到新的状态τ^n_{t+1}. 对应 TD 误差为:

$$\delta^n_t = R(\tau^n_t, x^n_t) + \gamma\hat{V}^n(\Phi(\tau^n_{t+1})) - \hat{V}^n(\Phi(\tau^n_t)), \tag{17}$$

其中, $\hat{V}^n(\Phi(\tau^n_{t+1}))$表示在第$n$次迭代中$t$时刻设备状态$\tau^n_t$采取行动$x^n_t$后的下一时刻的设备状态$\tau^n_{t+1}$所对应的特征状态映射的近似值, $\hat{V}^n(\Phi(\tau^n_t))$表示在第$n$次迭代中$t$时刻设备状态$\tau^n_t$所对应的特征状态映射的近似值.

为提高迭代稳定性, 本文采用自适应学习率与平滑更新策略. 在 ADP 实际应用中, 易于编码的简单步长规则和常数步长容易导致收敛速度不稳定、修正效果不佳等问题,ADP 中的步长设置需要结合学习过程和预测误差等信息进行自适应调整[29]. 基于这一思想, 本文在基础学习率随迭代衰减的同时, 根据 TD 误差调节有效学习率. 第n次迭代的特征近似值按以下三步更新:

第一步, 计算第n次迭代中t时刻的自适应学习率η^n_t:

$$\eta_t^n = \max(\eta_0\rho^n, 0.1)\min(1, \frac{|\delta_t^n|}{100}), \tag{18}$$

其中, η_0和ρ分别为学习率的初始值和衰减系数. $|\delta_t^n|$为 TD 误差的绝对值, 用于调节更新步长. 当误差较小时降低有效学习率, 减少小幅波动带来的频繁修正; 当误差较大时, 误差缩放因子$\min(1, \frac{|\delta_t^n|}{100})$被截断为 1, 使有效学习率不超过当前迭代的基础学习率, 避免过大更新引起震荡.

第二步, 基于自适应学习率和 TD 误差计算特征近似值的更新增量:

$$\Delta\hat{V}^n(\Phi(\tau_t^n)) = \eta_t^n\delta_t^n, \tag{19}$$

该增量反映当前样本经验对特征近似值估计的修正方向与幅度, 其中, δ_t^n决定了更新方向和基本修正量.

第三步, 为抑制特征近似值估计的剧烈波动, 采用如下的平滑更新策略进行特征近似值的更新:

$$\begin{aligned}\hat{V}^{n+1}(\Phi(\tau_t^n)) &= (1-\xi)\hat{V}^n(\Phi(\tau_t^n)) + \\ &\xi(\hat{V}^n(\Phi(\tau_t^n)) + \Delta\hat{V}^n(\Phi(\tau_t^n))) = \\ &\hat{V}^n(\Phi(\tau_t^n)) + \xi\Delta\hat{V}^n(\Phi(\tau_t^n)),\end{aligned} \tag{20}$$

公式 (20) 表示相邻两次迭代间同一特征对应值函数的更新过程. 其中, ξ为平滑因子, $\hat{V}^{n+1}(\Phi(\tau_t^n))$表示在第$n+1$次迭代更新后$\Phi(\tau_t^n)$的价值估计. 同一状态在单次迭代中被多次访问时逐次更新; 未被访问时, 其近似值保持不变. 该策略通过融合新旧信息并保留历史估计, 抑制异常样本或较大 TD 误差引起的波动, 保证价值估计的连续性和稳定性[29].

FD-ADP 在迭代训练中同步优化策略与特征近似值, 采用利润监控早停机制控制计算成本[23-24]. 算法持续记录模拟环境下的期望累积利润及历史最优结果, 以最大迭代次数限定训练时间. 若连续K轮未刷新最优利润, 则终止训练, 锁定历史表现最优时刻所对应的策略及特征近似值, 并输出为最优策略π^*.

在后续数值实验中,FD-ADP 的算法参数设置如下: 最大训练迭代次数$N = 5000$, 早停耐心轮数$K = 300$, 初始学习率$\eta_0 = 1$, 学习率衰减系数$\rho = 0.995$, 初始探索率$\epsilon_0 = 1$, 探索率衰减系数$\sigma = 0.995$, 平滑因子$\xi = 0.1$. 整体伪代码架构如算法 FD-ADP 所示.

算法 1 FD-ADP.

输入: 设备数M, 计划期T, 最大迭代次数N, 初始学习率η_0, 初始探索率ϵ_0.

输出: 最佳策略π*, 最佳利润$\hat{V}^*$.

1: 初始化: 所有特征近似值$\hat{V}(\Phi(.)) = 0$.

2: for 迭代轮次 n=1 to N do:

3: 策略$\pi_n = \varnothing$, 所有设备最早可用时间$\tau_1 = (0, \ldots, 0)$.

4: for 离散计划期 t=1 to T do:

5: 生成订单池$\mathcal{O}_t^n = \mathcal{J}_t^n \cup \mathcal{C}_t^n$, $|\mathcal{C}_t^n|\ Poisson(\lambda)$.

6: 通过公式(10)提取状态特征$\phi(\tau_t^n)$.

7: 通过公式(12)确定有效动作集$X_{t,comb}^n$.

8: 通过公式(15)计算动态探索率ϵ_t^n.

9: 生成一个随机数$rand \in [0,1]$.

10: **if** $rand < \epsilon_t^n$ **then:**

11: 随机选择动作$x_t^n \in X_{t,comb}^n$.

12: else:

13: 通过公式(14)评估选择最优动作x_t^n.

14: end if

15: 记录t时刻及之前的策略$\pi_n \leftarrow \pi_n \cup \{x_t^n\}$.

16: 执行动作x_t^n, 通过公式(1)计算下一阶段状态τ_{t+1}, 公式(2)计算即时利润$R(\tau_t^n, x_t^n)$.

17: 通过公式(17)计算TD误差δ_t^n, 公式(18)更新自适应学习率η_t^n.

18: 通过公式(20)更新特征近似值$\hat{V}^{n+1}(\Phi(\tau_t^n))$.

19: end for

20: 应用早停机制, 连续K轮利润未提升则终止, 否则更新最佳策略π*与最佳利润$\hat{V}^*$.

21: end for

3 数值实验

为验证本文所提出方法和策略的有效性以及分析策略性等待机制的行为特征, 本节进行了一系列数值实验. 鉴于云制造领域对各类制造服务及订单属性的描述尚无公开的标准数据集[7], 本文参考相关研究[14,19,21], 通过随机生成订单属性构造实验算例. 本文假设计划期内订单池持续更新, 且订单需求始终充足. 每个时刻订单池中的标准订单数量$|\mathcal{J}_t|$=10, 定制订单随机到达, 服从参数为 0.3 的泊松分布, $|\mathcal{C}_t|$~ *Poisson*(λ=0.3).

订单的各项属性参数严格遵循第 1 节模型的界定, 在其所属类型的取值区间内独立随机生成. 标准订单作为基准, 其单位时间资源占用系数设为$\beta_{t,o}^J = 1$, 且不产生协调成本$\omega_{t,o}^J = 0$. 其加工时长、收益与单位时间加工成本分别设定为: $p_{t,o}^J \sim \mathcal{U}(3,5)$, $r_{t,o}^J \sim \mathcal{U}(200,400)$, $k_{t,o}^J \sim \mathcal{U}(3,12)$. 相较之下, 定制订单因涉及个性化设计、复杂工艺适配及跨组织协作, 在资源消耗与经济属性上均设定于更高区间. 其资源占用系数为$\beta_{t,o}^C \sim \mathcal{U}(1,1.5)$, 并产生单位时间协调成本$\omega_{t,o}^C \sim \mathcal{U}(3,6)$, 加工时长更长, 设为$p_{t,o}^C \sim \mathcal{U}(6,9)$. 其收益与单位时间加工成本也相应

更高, 分别为$r_{t,o}^C \sim \mathcal{U}(800,1800)$和$k_{t,o}^C \sim \mathcal{U}(5,15)$, 以反映其技术复杂性与定制溢价. 上述参数设置使定制订单的期望单位时间净利润高于标准订单, 即满足$\dfrac{\overline{r}_{t,o}^J - \overline{\beta}_{t,o}^J \overline{k}_{t,o}^J \overline{p}_{t,o}^J}{\overline{p}_{t,o}^J} < \dfrac{\overline{r}_{t,o}^C - (\overline{\omega}_{t,o}^C + \overline{\beta}_{t,o}^C \overline{k}_{t,o}^C)\overline{p}_{t,o}^C}{\overline{p}_{t,o}^C}$, 这构成了制造商进行“策略性等待”以保留产能承接定制订单的核心经济激励. 所有算法均采用 Python 实现, 运行环境为 Apple M1 Pro 处理器和 16 GB 内存.

3.1 算法有效性分析

为检验 FD-ADP 方法的有效性, 本节开展对比实验: 以 DP 作为可求解规模下的最优性参照; 选取机器学习、数学启发式和元启发式方法进行比较; 通过特征消融实验, 考察特征压缩对算法性能的影响. 问题规模由M和T确定, 其他订单参数与前文一致. 每组算例采用不同随机数种子生成 30 个实例, 各算法在相同订单样本路径和设备初始状态下运行.

FD-ADP 与 DP 的比较结果见表 3. 当问题规模扩大到$M = 4$, $T = 50$时,DP 存在“维数灾难”, 而 FD-ADP 仍能高效给出相应的近似解. FD-ADP 的平均利润损失率为 4.85%, 说明状态压缩和值函数近似仅造成较小的最优性损失; 其平均运行时间较 DP 降低 74.13%, 且计算优势随问题规模扩大而增强. 当 M=3、T≥20 时, 时间降低率均超过 99%; 当 M=4、T≥50 时,DP 由于状态与动作组合规模急剧增长难以继续求解, 而 FD-ADP 仍能在较短时间内得到稳定解. 因此, FD-ADP 能够以有限的利润损失显著提高求解效率, 缓解“维数灾难”.

进一步将 FD-ADP 与其他非精确算法进行比较. 深度 Q 值网络算法 (Deep Q-Network, DQN)[6] 为经典的基于机器学习的在线接单决策方法, 根据设备与订单池状态学习接单和派工动作; 基于时间段分解的订单接收算法 (Time-Bucket Order Acceptance, TB-OAS) 依据 Tarhan 和 Oğuz[30] 的时

表 3 FD-ADP 与 DP 的平均利润与平均运行时间对比

M	T	FD-ADP 平均利润	FD-ADP 平均利润标准差	FD-ADP 平均时间	FD-ADP 平均时间标准差	DP 平均利润	DP 平均利润标准差	DP 平均时间	DP 平均时间标准差	平均利润损失率 (DP- FD-ADP)/DP × 100%	平均时间降低率 (DP -FD-ADP)/DP × 100%
1	10	1 642.78	460.25	0.79	0.10	1 646.65	462.29	0.86	0.12	0.22	6.81
	20	3 564.29	560.00	1.62	0.17	3 596.08	553.51	1.85	0.11	0.91	11.64
	30	5 461.29	693.05	2.55	0.18	5 548.55	716.47	3.33	1.32	1.52	19.12
	40	7 112.38	772.80	3.44	0.84	7 330.03	805.53	6.99	6.15	2.90	35.50
	50	8 800.86	588.94	6.11	0.47	9 090.41	531.76	6.97	2.06	3.18	8.48
	60	10 530.48	773.59	7.20	0.45	11 021.49	755.31	9.52	2.74	4.45	20.80
	70	12 050.42	728.85	7.97	0.35	12 628.71	816.85	11.96	2.79	4.52	30.62
2	10	2 748.67	698.02	2.42	0.49	2 865.23	698.44	8.20	1.93	3.97	69.71
	20	6 135.42	810.95	4.85	0.73	6 340.28	788.17	31.63	4.05	3.25	84.47
	30	9 249.56	976.73	8.30	1.50	9 655.81	970.31	75.68	3.28	4.18	89.04
	40	12 297.75	1 151.09	11.37	1.98	13 107.82	1 248.32	156.10	6.65	6.07	92.72
	50	15 380.93	1 129.94	13.89	1.22	16 304.41	1 013.63	292.41	16.34	5.67	95.24
	60	18 381.70	1 070.65	16.91	1.12	19 680.63	1 198.36	486.03	26.72	6.53	96.51
	70	21 158.36	1 199.33	20.65	5.68	22 615.02	1 261.80	792.18	51.83	6.40	97.40
3	10	3 911.00	749.29	4.06	0.65	3 988.32	785.23	161.45	18.11	1.77	97.47
	20	8 268.57	982.89	9.32	1.44	8 778.47	1 094.79	940.03	39.72	5.67	99.01
	30	12 567.95	1 281.28	14.37	2.53	13 327.72	1 282.32	3 090.71	123.05	5.72	99.54
	40	16 713.66	1 577.02	19.52	3.15	17 837.43	1 442.91	7 965.73	322.40	6.38	99.76
	50	21 063.16	1 589.34	25.36	3.56	22 611.57	1 461.23	16 822.38	558.92	6.88	99.85
	60	25 012.28	1 550.60	30.03	4.39	27 341.10	1 836.63	31 310.12	670.34	8.38	99.90
	70	29 307.07	1 833.36	36.41	4.61	31 876.22	1 812.31	51 550.25	1 465.17	8.05	99.93
4	10	4 958.51	892.10	5.48	1.10	5 068.82	887.87	3 674.72	501.96	2.22	99.85
	20	10 304.85	1 121.54	11.94	2.58	10 982.74	1 123.60	26 664.49	1 791.09	6.13	99.96
	30	15 464.62	1 660.25	19.43	3.97	16 735.58	1 527.76	112 172.98	7 703.15	7.67	99.98
	40	20 510.16	1 515.79	26.59	4.53	22 467.68	1 666.27	414 803.59	23 388.70	8.66	99.99
	50	25 849.36	1 832.40	34.05	7.80	—	—	—	—	—	—
	60	31 332.65	2 128.74	42.40	5.74	—	—	—	—	—	—
	70	34 541.19	3 395.03	42.22	16.38	—	—	—	—	—	—
平均值										4.85	74.13

间段分解思想求解当前订单池快照, 并采用变邻域搜索和禁忌搜索改进求解方案; 遗传算法 (Genetic Algorithm, GA) 和粒子群优化算法 (Particle Swarm Optimization, PSO) 采用通用元启发式搜索; 遗传算法-精细化改进过程-禁忌搜索算法 (Genetic Algorithm-Refining Process-Tabu Search, GA-RP-TS) 参照 Li 等[31] 的混合框架, 通过订单-设备编码、精细化改进和禁忌搜索优化可行方案. 比较结果见表 4, 在 $M=4$、$T=[10,20,30,40,50,60,70]$ 的全部场景下,FD-ADP 均取得最高平均利润, 且多数差异具有统计显著性, 说明其具有较稳定的利润优势. DQN 与 FD-ADP 的平均利润差距最小, TB-OAS 和 GA-RP-TS 的平均利润相对较低, 表明仅依据当前订单池快照或局部组合搜索, 难以充分刻画产能保留的未来机会价值. FD-ADP 利用特征聚合值函数关联资源负载、时间进度与未来期望利润, 使即时接单与策略性等待之间的权衡更具前瞻性, 因此在不同计划期长度下均表现出更好的平均利润水平.

表 4　FD-ADP 与其他非精确算法的平均利润对比

算法	10	20	30	40	50	60	70
FD-ADP	4958.51	10304.85	15464.62	20510.16	25849.36	31332.65	34541.19
DQN	4581.26	9360.25	14773.58	19880.63	25699.17	30657.33	33781.94
GA-RP-TS	4579.10	9427.89	14643.82	19583.21	25006.50	29881.17	33114.30
TB-OAS	4248.66	8996.77	13429.12	17912.46	22507.36	26739.60	30829.07
PSO	4146.44	8710.47	13350.65	18127.30	23251.33	28252.36	33181.34
GA	4180.57	8901.86	13710.69	18658.34	23789.63	28900.88	32312.37

注:表格颜色表示显著低于FD-ADP的统计检验结果p<0.05, p<0.01, p<0.001

为验证 FD-ADP 的状态压缩设计的合理性, 在 $M=4$, $T=50$场景下开展特征消融实验, 分别单独去掉空闲设备数、最大剩余忙碌时间、平均剩余忙碌时间和时间进度比特征, 比较平均利润、运行时间与等待频率. 等待频率为存在空闲设备且存在可行订单却未尽可能接满空闲设备的时期数占存在空闲设备且存在可行订单的时期数的比例.

结果见表 5. 完整 FD-ADP 的平均利润最高, 说明四类特征均具有有效贡献. 删除最大或平均剩余忙碌时间后, 等待频率分别升至 0.197 和 0.190, 表明负载信息有助于识别产能占用风险, 避免形成偏高等待倾向. 删除时间进度比造成的利润降幅最大, 说明该特征对有限计划期内未来订单完成窗口和等待价值的判断最为关键. 有限计划期下, 等待价值会随剩余时间缩短而快速变化, 缺少时间进度信息将使算法难以判断未来订单是否有足够完成窗口, 从而削弱其动态权衡能力. 上述结果表明, 完整四维特征组合能有效地支撑 FD-ADP 的价值估计与等待决策.

表 5　FD-ADP 状态特征消融实验的性能比较

	平均利润	平均运行时间	平均等待频率
FD-ADP	25 849.36	34.05	0.174
去掉空闲设备数	25 609.13	29.69	0.166
去掉最大剩余忙碌时间	25 633.09	24.31	0.197
去掉平均剩余忙碌时间	25 618.73	17.99	0.190
去掉时间进度比	21 394.97	4.11	0.151

3.2　关键参数敏感性分析

本节在 $M=3$, $T=50$场景下, 对算法部分关键参数进行敏感性分析. 学习率衰减系数直接影响特征近似值的学习速度, 探索率衰减系数影响算法由随机探索向价值利用转移的速度, 平滑因子影响随机订单样本下价值函数更新的稳定性. 在参数分析的设定中, 除学习率衰减系数ρ、探索率衰减系数σ和平滑因子ξ外, 其余参数保持基准设置不变, 并在不同随机数种子生成的 30 个随机实例上重复运行. 每组参数从平均利润、平均利润的 95% 置信区间、平均等待频率、收敛轮次等维度进行比较, 以评估基准参数在利润水平、策略稳定性和计算效率之间的平衡性. 结果见图 1, 当学习率衰减系数$\rho=0.995$、探索率衰减系数$\sigma=0.995$、平滑因子$\xi=0.1$时, 均处于较高平均利润和较低平均等待频率对应的参数区域, 能够在价值学习速度、探索-利用转换和更新稳定性之间取得较好平衡.

3.3　策略性等待机制有效性分析

为检验策略性等待机制及其在不同订单结构下的有效性, 本节在 $M=3$, $T=50$场景下, 将 FD-ADP 与五类策略方法进行比较: 非策略性等待下的 FD-ADP(Non-Strategic FD-ADP,NS FD-ADP) 保留 FD-ADP 的状态特征提取和值函数近似框架, 但移除策略性等待准则; 先到先服务规则 (First-Come-First-Served,FCFS) 按订单到达顺序依次接收可行订单; 单位时间收益密度规则 (Revenue-Per-Time Rule,RPT) 按照订单单位加工时间净利润由高到低接收订单; 基于阈值的等待规则 (Threshold-Based Waiting Rule,TBW) 根据随时间递减的动态阈值决

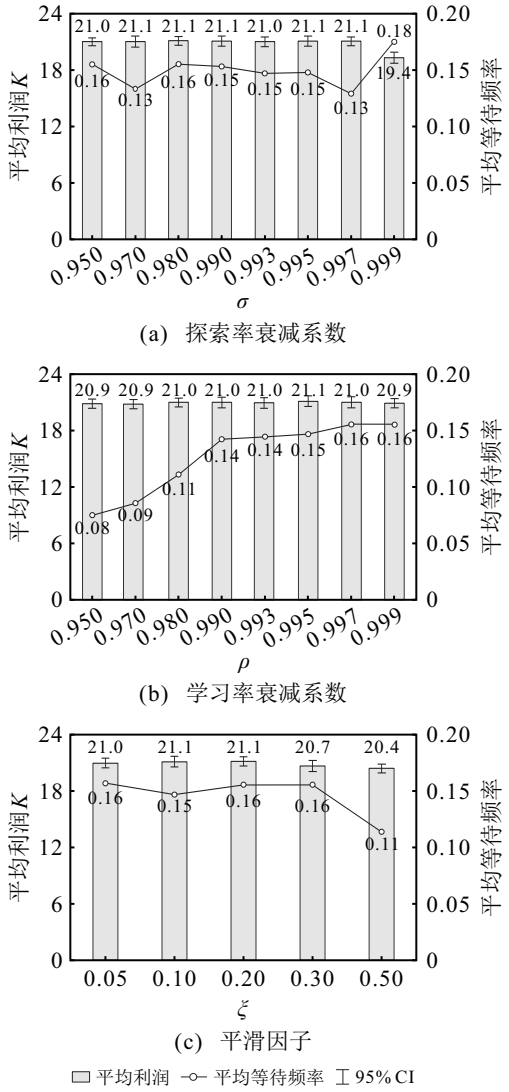


图 1 算法参数对平均利润与等待频率的影响

定是否等待, 并在验证实例上确定参数后用于测试. 滚动时域优化方法 (Rolling Horizon Optimization, RHO) 枚举当前可行动作, 通过有限预测窗口和有限期 DP 评估其短期未来收益. 订单属性参数与前文设置一致, 所有策略实验均采用不同随机种子生成 30 个随机实例, 并以平均利润和平均等待频率作为主要评价指标. 等待频率定义为存在空闲设备且存在可行订单时, 未完全接满空闲设备的时期数占相应可决策时期数的比例. 对比结果见图 2(a). FD-ADP 的平均利润高于 NS FD-ADP, 也高于其他四种规则. 该结果说明, 单纯提高即时收益密度或设置固定等待规则, 难以充分刻画随机异质订单流下产能保留的未来机会价值. RHO 具有较高等待频率, 但平均利润低于 FD-ADP, 表明等待行为本身并不必然带来利润的提升, 关键在于能否结合系统负载、剩余时间和未来价值进行状态依赖判断. FD-ADP 的等待频率处于适中水平, 说明其并非简单增加等待次数, 而是在资源紧张和未来高价值订单具有承接可能时选择性保留产能.

本节进一步构造不同定制订单到达强度、非平稳到达模式、标准订单规模、定制订单利润溢价和定制订单加工时间五类扩展实验, 以检验策略性等待机制在不同订单结构和需求环境下的稳健性. 各实验除所考察的参数外, 其余参数均保持不变.

在不同定制订单到达强度的实验中, 令 $\lambda \in \{0.1, 0.3, 0.6, 1.2, 2.4, 4.8\}$, 对应计划期内定制订单期望总数 5~240, 覆盖极低、中等和极高定制需求强度. 实验结果见图 2(b). FD-ADP 在全部到达强度下均获得最高利润, 其等待频率随 λ 增大呈先升后降趋势. 定制需求处于中等水平时, 保留产能的机会价值较高. 需求过于频繁时, 系统更容易直接获得高价值订单, 主动等待的边际利润反而降低.

在计划期内期望定制订单到达总数不变的条件下, 设置平稳、递增、递减、先增后减和 W 型五种到达模式. 结果见图 2(c). FD-ADP 在不同到达模式下均保持较高平均利润水平, 且利润波动相对较小. 需求后移或存在阶段性峰值时, 短视策略容易过早占用产能或错失后续订单高峰.FD-ADP 通过长期价值近似调整产能保留决策, 在不同到达模式下均具有较好的稳健性.

在不同标准订单规模的实验中, 设置每期标准订单数量 $|\mathcal{J}_t| \in \{1, 2, 3, 5, 10\}$, 分别对应标准订单非常稀缺、较为稀缺、刚好等于设备数量、较为充足和非常充足的现实场景. 结果见图 2(d). 随着标准订单增加, 各方法平均利润整体提高, FD-ADP 始终保持最高水平. 在标准订单稀缺时, 系统空闲可能来自订单不足或主动保留产能, 采用上述等待频率定义可将二者进行区分. 结果表明, FD-ADP 在不同标准订单规模下均能维持适度等待, 其有效性并不依赖订单持续充足这一有利条件, 表现出较好的适应性.

在定制订单利润溢价实验中, 以定制订单与标准订单的单位时间期望净利润之比 g 表示利润溢价, 设置 $g \in \{1.3, 1.8, 2.3, 2.8, 3.3\}$, 其中 $g = 2.3$ 对应本文基准设置, 其余场景仅调整定制订单收益分布. 图 2(e) 显示, 随着 g 提高, 六种算法的平均利润均上升, FD-ADP 的利润优势逐渐扩大. 当溢价较低时, 两类订单的利润差异有限, 等待的边际收益较弱. 溢价提高后, 保留产能的机会价值增大, FD-ADP 能够更有效地将其转化为长期利润. RHO 和 TBW 在较高的 g 下也表现出一定等待倾向, 但其利润仍低于 FD-ADP, 说明基于阈值或有限预测窗口的方法难以替代基于状态价值学习的等待策略.

在定制订单加工时间实验中, 本文设置加工时

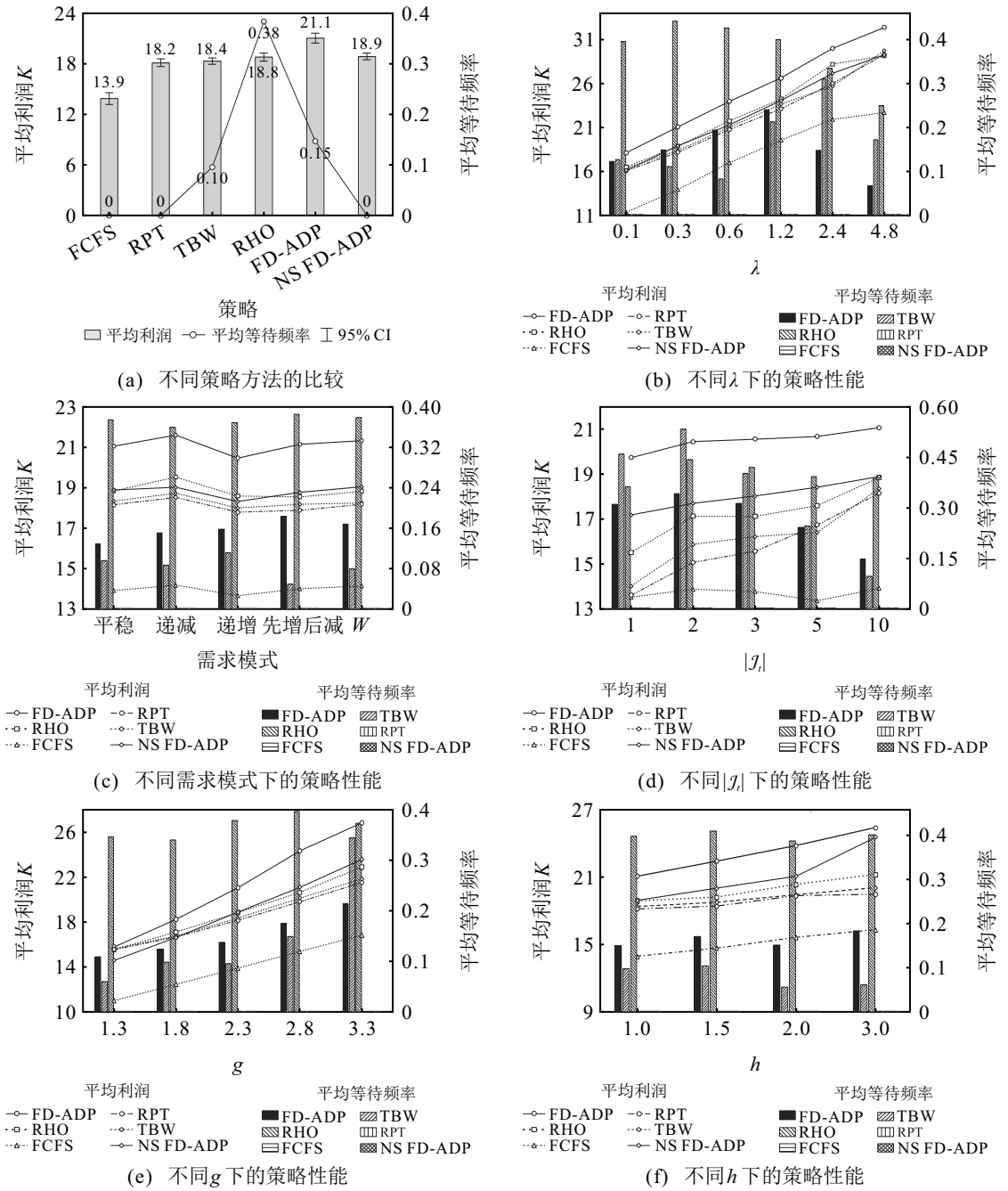


图 2　FD-ADP 的策略性能与敏感性分析

间放大系数$h \in \{1.0, 1.5, 2.0, 3.0\}$, 同步放大定制订单收益, 使其单位时间净利润水平基本保持稳定, 考察长加工时间对产能占用的影响. 结果见图 2(f). FD-ADP 在各参数水平下均取得最高利润. 当$h = 3.0$时, NS FD-ADP 与其利润差距缩小, 说明单笔收益同步提高且加工时间极长时, 直接接收高收益订单可获得较高利润, 策略性等待的边际作用减弱. RHO 的高等待频率仍未形成利润优势, 原因在于过多等待会压缩后续订单的可完成窗口. FD-ADP 并非在所有场景下都通过更高等待频率获得优势, 而是能够根据加工时间和剩余产能状态调整等待强度, 从而保持较为稳定的利润表现.

3.4　策略性等待行为分析

为揭示策略性等待的行为模式及其状态依赖性, 本文对 FD-ADP 在$M = 3$, $T = 50$场景下多次模拟运行中记录的 2 202 次有效等待事件进行聚类分析. 每个等待事件均关联一个由四个指标构成的状态特征向量. 表 6 给出了各聚类簇在原始特征空间中的中心值. 根据聚类结果, 可将等待事件归纳为三类具有明确物理含义的典型系统状态: 第一类状态中, 空闲设备数较低、负载较高且处于计划期前段, 称为“高负载早期”; 第二类状态中同样空闲设备数有限、负载较高, 但时间进度已进入中后期, 定义为“高负载中后期”; 第三类状态中, 空闲设备较多、负

载较轻且相较于前面两类状态较靠近计划期中前期, 称为“低负载中前期”. 前两类状态表明, 当设备负载较高且未来定制订单仍有完成可能时, 制造商倾向保留有限的空闲产能, 避免当前标准订单挤占后续高价值订单的加工机会. 第三类状态说明, 即使设备负载较低, 只要计划期仍处于中前段, 制造商也可能通过适度等待提高未来产能的利润密度. 因此, 等待既是资源紧张时维持产能灵活性的防御性选择, 也是资源相对宽松时追求未来收益的主动决策. 策略性等待并非由单一因素驱动, 而是由资源负载与时序演进状态共同驱动的动态响应行为.

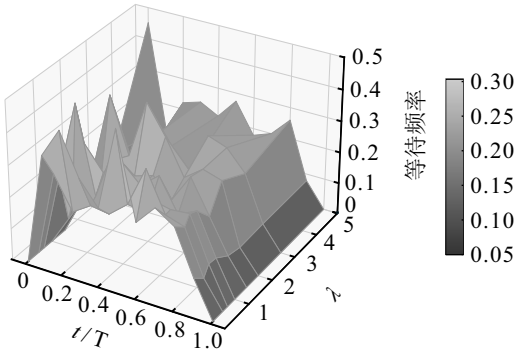
表 6 策略性等待行为聚类中心特征值

模式类型	空闲设备数	最大剩余忙碌时间	平均剩余忙碌时间	时间进度比
高负载早期	1.139	4.477	2.238	0.291
高负载中后期	1.443	3.722	1.668	0.709
低负载中前期	2.286	1.499	0.507	0.394

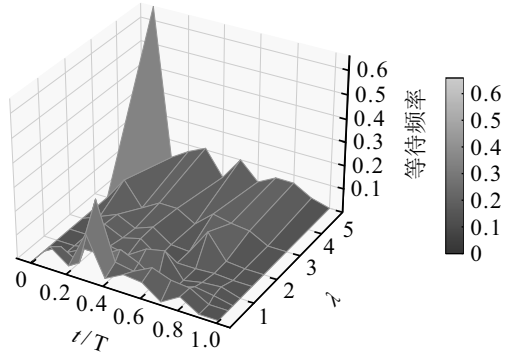
进一步在$M=3, T=50$场景下开展多因素控制实验, 从空闲设备数、时间进度比及定制订单到达率三个关键维度分析不同条件下等待行为的发生概率. 其中$\lambda \in \{0.1,0.5,0.7,1.0,1.5,2.0,3.0,5.0\}$, 对应不同定制订单的需求强度. 等待频率为特定状态下等待事件次数占该状态出现次数的比例.

实验结果见图 3. 当空闲设备数由 3 台减至 1 台时, 等待频率非线性上升, 说明资源越紧张, 保留剩余产能的价值越高, 这与两类高负载聚类结果一致. 其次, 定制订单到达率对等待频率呈非单调影响. 空闲设备数为 1 时, λ=0.1 对应的需求过低, 未来高价值订单出现概率较小, 等待动机不足. 随着λ增至约 2.0, 未来订单机会增加, 等待频率快速上升. 当λ继续提高时, 高价值订单频繁到达并可即时接收, 主动等待的必要性反而下降. 由此可见, 等待最活跃的并非极高需求场景, 而是定制订单到达强度适中的区间. 最后, 时间进度比进一步限定了等待的有效窗口. 等待主要集中在计划期中前段 ($t/T \in [0.2, 0.6]$), 此时剩余时间较充足, 制造商能够承担暂时闲置产能的风险. 随着计划期推进, 后续订单的可完成窗口缩短, 等待频率逐渐下降. 当空闲设备数为 1 且$t/T>$ 0.8 时, 等待频率迅速降低, 说明临近计划期末时继续等待的风险已超过等待的潜在收益.

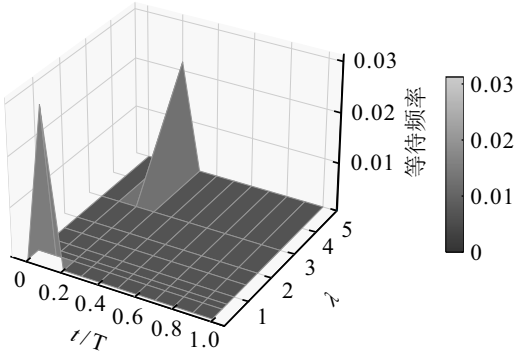
基于上述对策略性等待行为的聚类分析与多因素控制实验, 本节从内部状态与外部参数两个维度, 揭示了策略性等待行为的复杂驱动机制. 研究表明, 等待决策并非单一因素的简单响应, 而受到资源负



(a) 空闲机器数=1



(b) 空闲机器数=2



(c) 空闲机器数=3

图 3 等待行为发生频率三维曲面

载、时序演进与市场需求三者动态交互的影响: 高负载早期、高负载中后期、低负载中前期三类典型内部系统状态刻画了等待的决策逻辑与触发条件, 而外部需求强度动态调节着等待行为的边际价值. 策略性等待的管理含义在于识别“何时保留产能”, 而非追求更高的设备闲置比例. 实时决策中, 制造商可依据空闲设备数和剩余忙碌时间判断资源紧张程度, 并结合计划期进度识别等待的可行窗口; 在需求评估中, 还需考虑定制订单到达强度, 因为极低需求下等待机会不足, 极高需求下高价值订单可即时到达, 均可能削弱主动等待的必要性.

4 结 论

本文围绕云制造环境下在线并行机动态订单接收问题展开研究, 重点探讨在标准订单与定制订单

共存的生产场景中, 制造商面对随机、动态、异质的在线订单流, 如何通过策略性等待实现长期利润最大化. 本研究提出了一种基于策略性等待的特征驱动近似动态规划方法, 通过状态特征压缩与值函数近似, 缓解传统动态规划中的“维数灾难”, 并实现对动态订单接收中策略性等待行为的智能决策支持.

数值实验结果表明, 本文提出的 FD-ADP 方法在求解效率与策略性能之间达到较好平衡. 相较于精确动态规划所得理论最优解, FD-ADP 的平均利润损失率仅为 4.85%, 同时大幅降低了计算时间. 与 DQN、TB-OAS、GA、PSO 和 GA-RP-TS 等非精确算法相比, 其在不同计划期长度下均获得更高的平均利润. 特征消融实验表明, 时间进度比是判断有限计划期内等待价值的关键信息. 策略对比与扩展实验进一步验证了 FD-ADP 在不同订单结构和需求模式下的适应性. 行为分析发现, 等待主要集中于高负载早期、高负载中后期和低负载中前期三类状态, 并受到资源负载、计划期进度与定制订单到达率的共同影响.

本研究突破了传统“空闲即接单”的被动决策范式, 将“策略性等待”明确建模为制造商在随机动态订单接收中的主动决策机制, 为随机订单到达环境下的动态订单接收问题提供了高效且可行的近似求解方法. 制造商在面对定制订单的随机冲击时, 不宜机械采用传统“空闲即接单”策略, 而应根据系统的资源负载、时序演进以及订单到达模式等多维因素, 灵活调整等待策略.

本文以经济利润最大化为单一目标, 尚未考虑设备能耗和订单公平性等因素. 未来可构建多目标决策模型, 进一步考察更复杂的订单到达模式, 增强算法在更广泛制造场景中的适用性.

参考文献 (References)

[1] 孙晓晨, 李勇建, 张双, 等. 制造系统生态化概念、内涵及展望[J]. 控制与决策, 2024, 39(8): 2465-2483.
(Sun X C, Li Y J, Zhang S, et al. Concepts, connotations, and future research of ecologization in manufacturing systems[J]. Control and Decision, 2024, 39(8): 2465-2483.)

[2] Gourgand M, Grangeon N, Norre S. A review of the static stochastic flow-shop scheduling problem[J]. Journal of Decision Systems, 2000, 9(2): 1-31.

[3] 于超, 刘洋, 樊治平. 考虑风险传导情形的订单接受决策方法[J]. 控制与决策, 2019, 34(3): 611-619.
(Yu C, Liu Y, Fan Z P. Decision making method for order acceptance considering risk propagation[J]. Control and Decision, 2019, 34(3): 611-619.)

[4] Wang Z, Qi Y Q, Cui H R, et al. A hybrid algorithm for order acceptance and scheduling problem in make-to-stock/make-to-order industries[J]. Computers & Industrial Engineering, 2019, 127: 841-852.

[5] 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.
(Xie N M, Wu Q, Zheng S X. Cross-vendor order allocation model for centralized integrated scheduling of cloud platforms[J]. Control and Decision, 2020, 35(3): 667-676.)

[6] 丁祥海, 张梦钗, 刘春来, 等. 基于深度强化学习的云订单动态接受与调度问题研究[J]. 运筹与管理, 2024, 33(9): 221-226.
(Ding X H, Zhang M C, Liu C L, et al. Research on cloud order dynamic acceptance and scheduling based on deep reinforcement learning[J]. Operations Research and Management Science, 2024, 33(9): 221-226.)

[7] 张晓冬, 燕洁晨, 孙家正. 动态扰动下云制造服务组合的区间多目标优化方法[J]. 控制与决策, 2026, 41(1): 19-30.
(Zhang X D, Yan J C, Sun J Z. Interval multi-objective optimization for cloud manufacturing service composition under dynamic disturbances[J]. Control and Decision, 2026, 41(1): 19-30.)

[8] 孙琳, 禹春霞, 陈萌, 等. 考虑可持续性的云制造服务组合优化研究[J]. 控制与决策, 2026, 41(5): 1477-1488.
(Sun L, Yu C X, Chen M, et al. Research on cloud manufacturing service composition and optimal selection[J]. Control and Decision, 2026, 41(5): 1477-1488.)

[9] Zheng C Y, Yu J, Wan G H. Online order acceptance and scheduling in a single machine environment[J]. Computers & Operations Research, 2025, 179: 107028.

[10] Arredondo F, Martinez E. Learning and adaptation of a policy for dynamic order acceptance in make-to-order manufacturing[J]. Computers & Industrial Engineering, 2010, 58(1): 70-83.

[11] Zhang L F, Mu Y P, Chao X R, et al. Optimal reservation control strategies in shared parking systems considering two types of customers[J]. Computers & Operations Research, 2023, 155: 106235.

[12] Chen S, Lei J F, Moinzadeh K. Cost optimization in cloud computing: Capacity reservation for intermittent random demand surges[J]. Production and Operations Management, 2024, 33(6): 1265-1284.

[13] Ahmadian M M, Salehipour A. The just-in-time job-shop scheduling problem with distinct due-dates for operations[J]. Journal of Heuristics, 2021, 27(1): 175-204.

[14] Ronconi D P, Powell W B. Minimizing total tardiness in a stochastic single machine scheduling problem using approximate dynamic programming[J]. Journal of Scheduling, 2010, 13(6): 597-607.

[15] Wulan Q. Order scheduling optimization in manufacturing enterprises based on MDP and dynamic programming[J]. Scientific Reports, 2023, 13: 9783.

[16] Powell W B. Approximate dynamic programming: Solving the curses of dimensionality[M]. Hoboken: Wiley-Interscience, 2007.

[17] 温广辉, 杨涛, 周佳玲, 等. 强化学习与自适应动态规划: 从基础理论到多智能体系统中的应用进展综述[J]. 控制与决策, 2023, 38(5): 1200-1230.
(Wen G H, Yang T, Zhou J L, et al. Reinforcement learning and adaptive/approximate dynamic programming: A survey from theory to applications in multi-agent systems[J]. Control and Decision, 2023, 38(5): 1200-1230.)

[18] Satic U, Jacko P, Kirkbride C. A simulation-based approximate dynamic programming approach to dynamic and stochastic resource-constrained multi-project scheduling problem[J]. European Journal of Operational Research, 2024, 315(2): 454-469.

[19] 何雯晴, 唐亮. 基于近似动态规划的共享制造平台订单动态定价优化[J]. 计算机集成制造系统, 2025, 31(6): 2237-2244.
(He W Q, Tang L. Order dynamic pricing optimization for shared manufacturing platform based on approximate dynamic programming[J]. Computer Integrated Manufacturing Systems, 2025, 31(6): 2237-2244.)

[20] Abedi A, Zhu W H. An advanced order acceptance model for hybrid production strategy[J]. Journal of Manufacturing Systems, 2020, 55: 82-93.

[21] Li X, Ventura J A. Exact algorithms for a joint order acceptance and scheduling problem[J]. International Journal of Production Economics, 2020, 223: 107516.

[22] Wu Q, Xie N M, Zheng S X, et al. Online order scheduling of multi 3D printing tasks based on the additive manufacturing cloud platform[J]. Journal of Manufacturing Systems, 2022, 63: 23-34.

[23] Bertsekas D P. Feature-based aggregation and deep reinforcement learning: A survey and some new implementations[J]. IEEE/CAA Journal of Automatica Sinica, 2019, 6(1): 1-31.

[24] Bertsekas D. Dynamic programming and optimal control: Volume I[M]. Belmont: Athena Scientific, 2017.

[25] 胡金昌, 吴耀华, 吴颖颖, 等. 考虑学习效应的最小化延误总时间的单机批次排序问题[J]. 控制与决策, 2019, 34(12): 2708-2712.
(Hu J C, Wu Y H, Wu Y Y, et al. Batch scheduling with learning effect on single-machine to minimize the total tardiness[J]. Control and Decision, 2019, 34(12): 2708-2712.)

[26] 巴智勇, 刘金朵, 李贤, 等. 面向作业车间调度问题的约束增强型邻域结构与批量近似评估方法[J]. 控制与决策, 2026, 41(6): 1709-1721.
(Ba Z Y, Liu J D, Li X, et al. Constraint-enhanced neighborhood structure and batch approximate evaluation method for job shop scheduling problem[J]. Control and Decision, 2026, 41(6): 1709-1721.)

[27] Usama M, Chang D E. Learning-driven exploration for reinforcement learning[C]. Proceedings of the 21st International Conference on Control, Automation and Systems. Jeju, 2021: 1146-1151. DOI: 10.23919/iccas52745.2021.9649810.

[28] Wang G J, Wu F G, Zhang X, et al. Adaptive trajectory-constrained exploration strategy for deep reinforcement learning[J]. Knowledge-Based Systems, 2024, 285: 111334.

[29] Ryzhov I O, Frazier P I, Powell W B. A new optimal stepsize for approximate dynamic programming[J]. IEEE Transactions on Automatic Control, 2015, 60(3): 743-758.

[30] Tarhan İ, Oğuz C. A matheuristic for the generalized order acceptance and scheduling problem[J]. European Journal of Operational Research, 2022, 299(1): 87-103.

[31] Li Y T, Tang L H, Bai D Y, et al. Profit maximization through integrated order acceptance and scheduling: A metaheuristic approach[J]. Applied Soft Computing, 2024, 159: 111570.

作者简介

侯林辉 (1998−), 女, 博士生, 主要研究方向为供应链管理、云制造生产调度管理, E-mail: houlinhui1029@163.com;

孙晓晨 (1978−), 男, 副教授, 博士, 主要研究方向为供应链管理、平台运营, E-mail: xcsun@tju.edu.cn;

李勇建 (1973−), 男, 教授, 博士, 博士生导师, 主要研究方向为现代物流与供应链管理、平台经济与运营管理、系统优化与决策分析, E-mail: liyongjian@nankai.edu.cn.