

面向算电协同的多目标调度优化研究

摘要

本文面向算电协同的多目标调度优化问题，提出递进式建模求解方案。

针对问题一：本问是统计预测与调度复合问题。首先对 50000 个任务按区域、类型、小时聚合 GPU 需求；接着采用直方图梯度提升回归建立区域-类型-小时级短期预测模型，按第 0-2375 小时训练、第 2376-2399 小时测试的口径评估；然后以最后 24 小时实际到达任务为输入，在 GPU 容量、IT 功率、时延与完成时限约束下执行基础调度。结果讨论显示，24 小时聚合 WMAPE 为 30.58%，538 个任务全部成功调度，RegionF 峰值 GPU 利用率达 99.19%，模型有效地实现了需求预判与基础调度。

针对问题二：本问是碳感知多目标调度优化问题，需以任务迁移与开工时段为决策变量。首先按功率映射折算 AI IT 负荷并乘以区域 PUE 得到设施负荷；接着以购电成本与碳排放加权最小为目标，建立含 GPU 容量、时延与完成时限约束的调度模型；然后采用代价驱动贪心加局部寻优算法求解。结果讨论显示，低碳导向调度使购电成本由 742.28 万元降至 648.36 万元、碳排放由 9733.17 降至 8497.53 tCO₂，模型有效地实现了碳感知的跨区域调度。

针对问题三：本问是储能协同优化问题，需在给定 IT 负荷下设计充放电策略。首先建立含功率平衡、SOC 递推与购售电边界的储能线性规划模型，以购电成本减售电收益加弃电惩罚最小为目标求解。结果讨论显示，储能使 RegionD、RegionE、RegionF 运行成本分别下降 7.32%、9.75% 与 10.91%，净购电功率标准差分别下降 100%、94.13% 与 94.03%，模型有效地实现了削峰填谷与新能源消纳。

针对问题四：本问是多区域算-储-电协同优化问题。首先建立上层任务调度与下层储能 LP 的分层协同框架，以综合加权目标求解；然后设置强碳约束、电价峰谷差扩大与新能源高波动场景对照。结果讨论显示，强碳约束下碳排放降至 0，新能源高波动下峰值净购电升至 117.37 MW，模型有效地实现了多目标权衡并稳健响应场景变化。

综上，本文为算电协同场景下的跨区域调度提供了高效解决方案。

关键词：算电协同；任务调度；梯度提升；储能优化；多目标优化；新能源消纳

目录

一、引言	1
1.1 问题背景	1
1.2 问题重述	1
二、总体分析	2
三、模型假设	2
四、符号说明	3
五、模型的建立与求解	4
5.1 问题一的模型的建立和求解	4
5.1.1 具体分析	4
5.1.2 模型准备	4
5.1.3 模型建立	6
5.1.4 模型求解	8
5.2 问题二的模型的建立和求解	10
5.2.1 具体分析	10
5.2.2 模型准备	11
5.2.3 模型建立	12
5.2.4 模型求解	14
5.3 问题三的模型的建立和求解	16
5.3.1 具体分析	16
5.3.2 模型准备	17
5.3.3 模型建立	18
5.3.4 模型求解	20
5.4 问题四的模型的建立和求解	22
5.4.1 具体分析	22
5.4.2 模型准备	23
5.4.3 模型建立	24
5.4.4 模型求解	25
六、模型检验与分析	28
6.1 误差分析	28
6.2 灵敏度分析	30

七、模型的评价	31
7.1 模型的优点	31
7.2 模型的缺点	32
八、模型的改进与推广	32
8.1 模型的改进	32
8.2 模型的推广	32
参考文献	33

一、引言

1.1 问题背景

随着人工智能算力规模的持续扩张，数据中心对电力系统的影响不断增强，电力成本已逐渐成为数据中心运营成本的核心组成部分。在“双碳”战略与全国一体化算力网络建设的双重驱动下，“算电协同”正从政策倡导演变为产业发展的现实需求。2026 年政府工作报告首次将“算电协同”纳入国家级新基建工程，明确提出实施超大规模智算集群、算电协同等新基建工程，加强全国一体化算力监测调度。算电协同更加关注算力负荷与能源系统之间的实时耦合关系：西部地区具有丰富的风能、太阳能和水电资源，电价相对较低，具备建设绿色低碳数据中心的天然优势；东部地区则更加接近用户侧，具有更低的网络时延与更高的实时业务需求。

在实际运行过程中，算力跨区域迁移涉及计算任务、电力系统、新能源、储能系统和通信网络之间复杂耦合关系的多目标协同优化。将 AI 任务迁移至新能源资源丰富地区虽然可以降低电费成本和碳排放，但可能导致网络时延增加；实时推理任务对时延极为敏感，而训练任务则更关注能源成本与算力规模；新能源出力具有波动性，需要储能系统进行协调。如何实现 AI 工作负载、能源系统与通信网络的联合优化，已成为当前面向算电协同的重要研究课题 [3, 4, 6]，数据中心算力规模的持续扩张也推动了算力网络协同调度研究的发展 [5]。

因此，本文以六个典型区域组成的算电协同场景为研究对象，综合考虑区域电价、碳强度、可用新能源出力、储能状态与网络时延约束，建立从算力统计预测、碳感知调度到储能协同、全要素多目标优化的递进式调度模型，为跨区域算力与能源协同调度提供系统解决方案。

1.2 问题重述

本赛题设定六个典型区域，其中 RegionA、RegionB 与 RegionC 代表东部高负荷区域，RegionD 代表西部算力中心，RegionE、RegionF 分别为光伏和风电资源富集区域。任务到达主时域为第 0–2399 小时，第 2400–2405 小时用于结清末端任务，第 2406 小时仅用于电力与储能终端状态结算。任务包括实时推理、批量推理和 AI 训练三类，不可抢占、不可拆分或中途迁移。

问题 1：基于附件 AI 工作负载数据，对不同区域、不同类型任务的 GPU 需求进行统计分析，并构建短期预测模型；建立基础算力调度模型，给出第 2376–2399 小时实际到达任务的调度方案，展示最后 24 小时的调度甘特图和区域 GPU 利用率。

问题 2：以第 0–2399 小时实际到达任务及对应逐时电力参数为输入，以任务迁移与开工时段为决策变量，以运行成本、碳排放、网络时延和新能源利用率为目标，建立碳感知任务调度模型并给出调度策略。

问题 3: 在给定任务调度和逐时负荷的基础上, 建立储能协同优化模型, 设计储能充放电策略, 分析储能系统对运行成本、碳排放、区域峰值净购电功率和负荷波动程度的影响。

问题 4: 在前三问基础上, 建立多区域算-储-电协同优化模型, 统一考虑任务迁移、电价、碳排放、可用新能源、储能、区域购售电边界、网络时延及任务异构性, 在系统运行成本、碳排放、网络时延、服务质量、新能源利用率和区域峰值净购电之间权衡, 并比较不同碳约束、电价机制和新能源波动场景下的策略变化。

二、总体分析

本文围绕算电协同场景下 AI 工作负载、能源系统与通信网络的联合优化问题, 构建了从算力侧统计预测到全要素多目标协同的递进式模型体系。

针对问题 1, 本文对 50000 个 AI 任务按区域、类型、小时维度进行 GPU 需求统计分析, 提取任务规模、GPU 小时占比与区域负荷特征, 并采用梯度提升树建立区域-类型-小时级 GPU 需求短期预测模型, 进一步构建考虑 GPU 容量与 IT 功率约束的基础贪心调度模型, 得到最后 24 小时的调度方案与区域 GPU 利用率。针对问题 2, 本文建立碳感知任务调度模型, 以任务迁移区域与开工时段为决策变量, 以购电成本与碳排放加权最小为主目标, 通过代价驱动贪心与局部寻优算法, 实现低电价、低碳强度区域优先的调度策略, 并评估运行成本、碳排放、网络时延与新能源利用率。针对问题 3, 本文在给定 IT 负荷的基础上, 建立考虑储能 SOC 递推、充放电功率与效率约束的储能协同优化线性规划模型, 设计分区域充放电策略, 定量分析储能对运行成本、碳排放、区域峰值净购电与负荷波动的改善。针对问题 4, 本文将任务调度与储能运行统一纳入多区域算-储-电协同优化框架, 以综合目标函数权衡系统运行成本、碳排放、网络时延、服务质量、新能源利用率与区域峰值净购电, 并通过多场景对照实验分析不同碳约束、电价机制与新能源波动下的策略变化。

四个问题层层递进: 问题 1 奠定算力需求分析与基础调度的基础, 问题 2 引入能源参数实现碳感知的任务级调度, 问题 3 在给定调度下优化储能这一电力侧调节资源, 问题 4 则将任务调度、储能运行与购售电边界统一求解, 实现算力与电力系统的深度耦合协同优化。这种“先算力、再算电耦合、后储电协同、终全局优化”的递进逻辑, 完整覆盖了算电协同调度的核心矛盾。

三、模型假设

为简化问题, 本文做出以下假设:

- **假设 1:** 所有 AI 任务在调度时不可抢占、不可拆分或中途迁移，任务一旦在某区域的某时段开工，则持续占用相应 GPU 资源直至执行完毕。
- **假设 2:** 任务跨区域迁移仅考虑单向网络时延，忽略网络带宽、迁移数据量、传输能耗与传输费用，任务执行区域的设施负荷按统一功率映射与区域 PUE 计算。
- **假设 3:** 各区域的新能源出力、电价与碳强度均取附件给定的逐时数据，新能源出力可被直接消纳、充电储能或外送，弃电部分不产生收益。
- **假设 4:** 储能系统 SOC 按充放电效率递推，充电与放电可同时进行但总充电功率受上限约束，优化结束时第 2406 小时 SOC 不低于初始 SOC。
- **假设 5:** 区域购电功率与售电功率分别受最大购电与最大外送边界约束，各区域之间不建模线路潮流，仅通过区域功率平衡独立结算。

四、符号说明

本文使用的主要符号及其含义如表1所示。

表 1 符号说明

符号	说明
r	区域编号, $r \in \{A, B, C, D, E, F\}$
t	调度时段, $t = 0, 1, \dots, 2406$
i	任务编号, $i = 1, 2, \dots, 50000$
T_i	任务 i 的类型, 实时推理/批量推理/AI 训练
d_i	任务 i 的 GPU 需求量 (等效 GPU)
τ_i	任务 i 的预计执行时长 (小时)
μ_i	任务 i 可接受的最大网络时延
$G_r(t)$	区域 r 第 t 小时的电网购电功率
$GS_r(t)$	区域 r 第 t 小时的外送售电功率
$R_r(t)$	区域 r 第 t 小时的可用新能源出力
$E_r(t)$	区域 r 第 t 小时的设施侧总负荷
$P_r(t)$	区域 r 第 t 小时的购电价格
$\gamma_r(t)$	区域 r 第 t 小时的碳强度
$C_r(t)$	区域 r 第 t 小时的储能充电功率
$D_r(t)$	区域 r 第 t 小时的储能放电功率
$SOC_r(t)$	区域 r 第 t 小时末的储能荷电状态
η_c, η_d	储能充电效率与放电效率
λ	碳排放权重系数

表 1 符号说明 (续)

符号	说明
PUE_r	区域 r 的能源使用效率

五、模型的建立与求解

5.1 问题一的模型的建立和求解

5.1.1 具体分析

问题一要求对 AI 工作负载进行 GPU 需求统计分析与短期预测，并在算力约束下给出最后 24 小时实际到达任务的基础调度方案，属于“统计 + 预测 + 调度”复合型问题。本问采用直方图梯度提升回归建立区域-类型-小时级 GPU 需求短期预测模型，并采用时延可达内本地优先的贪心策略构建基础算力调度模型。

首先对 workload_trace 中 50000 个任务按区域、类型与小时维度聚合 GPU 需求，提取任务规模、GPU 小时占比与区域负荷特征；接着构造包含小时周期、星期周期与滞后自回归特征的训练集，按第 0-2351 小时训练、第 2352-2375 小时调参、第 0-2375 小时重训、第 2376-2399 小时测试的口径训练预测模型；然后以最后 24 小时实际到达任务为输入，在满足 GPU 容量、IT 功率、网络时延与完成时限约束的条件下执行基础调度；最后输出调度甘特图与区域 GPU 利用率曲线。具体分析流程如图1所示。

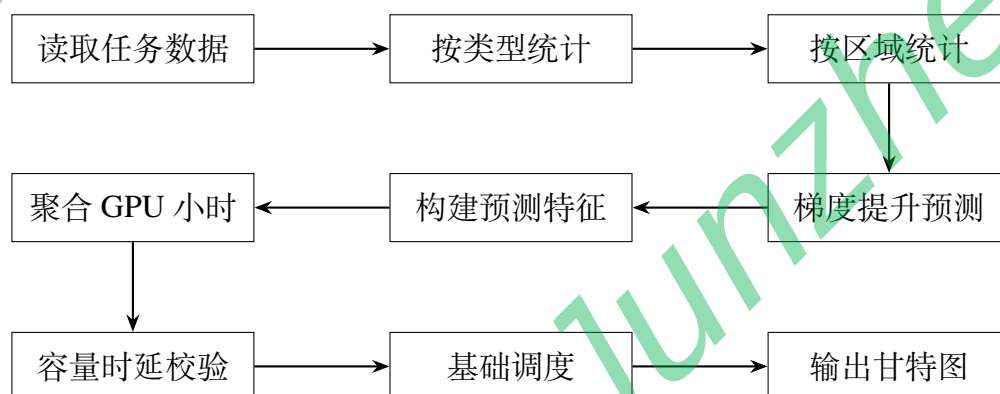


图 1 问题一分析流程图

5.1.2 模型准备

在进行建模之前，需要对附件数据进行预处理，并对预测与调度算法进行选择。本节将从数据预处理与算法选择两方面进行准备。

原始数据来源于附件中的 workload_trace.xlsx、GPU_information.xlsx 与 network_latency.xlsx 等数据表。其中 workload_trace 记录 50000 个 AI 任务的编号、类型、到达小时、GPU 需求、执行时长、来源区域、时延上限与最晚完成时点等信息；GPU_information 给出六个区域的 GPU 总容量、可调度 GPU、IT 功率上限与 PUE；network_latency 给出区域间单向网络时延矩阵。经检查发现数据存在任务到达呈随机稀疏特性、时间跨度达 2400 小时且区域类型组合众多等问题，需要统一聚合规整。

针对上述问题，按以下步骤进行预处理：第一步，将任务执行时长由分钟换算为小时 $\tau_i = E_i/60$ ，并计算每任务的 GPU 小时 $g_i = d_i \tau_i$ ；第二步，按到达小时、来源区域与任务类型对全部任务进行分组聚合，得到每区域每类型每小时的 GPU 需求序列；第三步，对未出现任务的时段组合用零值补齐，形成 $2400 \times 6 \times 3 = 43200$ 条完整序列；第四步，构造滞后 24/48/168 小时项与滚动均值等时序特征。预处理前后数据形态对比如表2所示。

表 2 数据预处理前后统计量对比

指标	预处理前	预处理后	单位	说明
记录数	50000	43200	条	逐任务 → 逐时
时间粒度	任务级	小时级	h	0–2399
区域覆盖	6	6	个	RegionA–F
类型覆盖	3	3	类	实时/批量/训练
总 GPU 小时	502.08 万	502.08 万	GPU·h	聚合守恒
缺失组合	无	已零值补齐	–	特征可计算

综上所述，数据预处理完成，原始的逐任务数据已规整为区域–类型–小时维度的完整序列，为后续建模提供了清洁输入。

该问题的短期预测本质上是一个多维时序回归问题，需要在区域与任务类型维度上预测未来逐时的 GPU 需求；基础调度则是一个带容量与时延约束的排程问题。常用的预测方法包括线性回归、随机森林与梯度提升回归等。本节选取三种方法，从预测精度、训练效率、非线性拟合与鲁棒性四个方面进行评估，对比结果如表3所示。

表 3 GPU 需求预测算法能力对比

算法名称	预测精度	训练效率	非线性拟合	鲁棒性
线性回归 (LR)	差	优	差	中
随机森林 (RF)	良	中	良	优
直方图梯度提 升 (HistGBRT)	优	优	优	良

通过上述对比可以看出，直方图梯度提升回归在非线形拟合与预测精度方面表现最优，其直方图分箱机制支持缺失值与大规模样本，训练效率高，且具备特征重要性分析能力，适合本文高维稀疏的时序预测任务。因此本文选择 HistGBRT 作为短期预测模型。调度层面，考虑到任务不可抢占且最后 24 小时任务规模较小，采用时延可达内本地优先、最早可行开工的贪心策略，能够在秒级完成求解并严格满足全部约束。

综上所述，本节完成了数据预处理与算法选型，接下来将建立数学模型并进行求解。

5.1.3 模型建立

针对问题一，本节将从 GPU 需求统计、短期预测与基础调度三个角度建立模型。具体过程如下。

步骤 1: GPU 需求统计分析模型

设任务 i 的 GPU 需求量为 d_i ，预计执行时长为 τ_i （小时），则该任务的 GPU 小时占用量为 $g_i = d_i \tau_i$ 。按任务类型 k 聚合，可得各类型任务的 GPU 小时总量 G_k 及其占比 p_k 如式 (1) 所示。

$$G_k = \sum_{i \in \mathcal{K}_k} d_i \tau_i, \quad p_k = \frac{G_k}{\sum_{k'} G_{k'}} \times 100\% \quad (1)$$

其中， $\mathcal{K}_k$ 为类型 k 的任务集合， $k \in \{\text{实时推理, 批量推理, AI 训练}\}$ 。进一步按到达小时 t 、来源区域 r 与类型 k 聚合，可得区域-类型-小时级 GPU 需求序列 $y_{r,t,k}$ 如式 (2) 所示。

$$y_{r,t,k} = \sum_{i: a_i=t, s_i=r, T_i=k} d_i \tau_i \quad (2)$$

其中, a_i 为任务 i 的到达小时, s_i 为来源区域。该序列同时作为短期预测的目标变量。

步骤 2: 短期 GPU 需求预测模型

以 $y_{r,t,k}$ 为预测目标, 构造特征向量 x_t 包含区域编码、类型编码、小时序号、星期序号、日内周期正弦余弦、滞后 24/48/168 小时自回归项与滚动均值等特征。梯度提升回归通过加法模型拟合目标 [1], 本文采用其直方图高效实现 [2], 预测值由式 (3) 给出。

$$\hat{y} = \sum_{m=1}^M f_m(x), \quad f_m \in \mathcal{F} \quad (3)$$

其中, M 为回归树数量, f_m 为第 m 棵回归树, $\mathcal{F}$ 为回归树函数空间。模型以均方误差为损失函数, 如式 (4) 所示。

$$L(y, \hat{y}) = \sum_i (y_i - \hat{y}_i)^2 \quad (4)$$

模型按负梯度方向逐步拟合每棵回归树, 即第 m 步以残差 $r_i^{(m)} = -\partial L / \partial \hat{y}_i$ 为目标拟合 f_m , 并通过学习率 η 控制步长, 如式 (5) 所示。

$$\hat{y}^{(m)} = \hat{y}^{(m-1)} + \eta f_m(x) \quad (5)$$

预测精度采用平均绝对误差 MAE、均方根误差 RMSE 与加权平均绝对百分比误差 WMAPE 评价, 分别如式 (6)–(8) 所示。

$$MAE = \frac{1}{N} \sum_{i=1}^N |y_i - \hat{y}_i| \quad (6)$$

$$RMSE = \sqrt{\frac{1}{N} \sum_{i=1}^N (y_i - \hat{y}_i)^2} \quad (7)$$

$$WMAPE = \frac{\sum_{i=1}^N |y_i - \hat{y}_i|}{\sum_{i=1}^N y_i} \times 100\% \quad (8)$$

其中, N 为测试样本数。由于任务到达呈随机泊松特性, 逐时序列存在大量零值, 本文同时报告 24 小时聚合口径的 WMAPE 以衡量调度预判能力。

步骤 3: 基础算力调度模型

以最后 24 小时 (第 2376–2399 小时) 实际到达任务为调度对象, 设任务 i 调度至区域 r 并在整点 s 开工, 引入 0–1 决策变量 $x_{i,r,s}$, 借鉴数据中心工作负载调度的一般框架 [7], 调度优化模型如式 (9)–(14) 所示。

$$\min \quad Z = \sum_{i,r,s} x_{i,r,s} [\beta(s - a_i) + \gamma l_{s_i,r}] \quad (9)$$

$$\text{s.t.} \quad \sum_r \sum_s x_{i,r,s} = 1, \quad \forall i \quad (10)$$

$$\sum_{i,s} d_i \omega(i, r, s, t) x_{i,r,s} \leq C_r^{gpu}, \quad \forall r, t \quad (11)$$

$$N_r(t) + \sum_{i,s} d_i \omega(i, r, s, t) p_{T_i} x_{i,r,s} \leq C_r^{it}, \quad \forall r, t \quad (12)$$

$$l_{s_i,r} \leq \mu_i, \quad \forall i, r \quad (13)$$

$$s + \tau_i \leq L_i, \quad s = a_i \text{ (实时任务)} \quad (14)$$

其中, $\omega(i, r, s, t)$ 为任务 i 在时段 t 对区域 r 的占用小时数 (按任务区间与时段的重叠长度折算), C_r^{gpu} 为区域 r 可调度 GPU 容量, p_{T_i} 为任务类型的单位 GPU 功率, C_r^{it} 为区域 IT 功率上限, $N_r(t)$ 为区域 r 时段 t 的固定 NonAI IT 负荷, $l_{s_i,r}$ 为来源区域到调度区域的网络时延, μ_i 为任务最大允许时延, L_i 为最晚完成时点, β 与 γ 分别为延迟与跨区时延的惩罚权重。式 (10) 保证任务唯一分配, 式 (11) 与式 (12) 保证 GPU 容量与 IT 功率约束, 式 (13) 保证时延满足 SLA, 式 (14) 保证完成时限且实时任务到达即开工。

综上所述, 本节完成了 GPU 需求统计、短期预测与基础算力调度三个模型的建立, 接下来将基于实际数据进行求解。

5.1.4 模型求解

基于上述模型, 本节依次完成统计分析、短期预测与基础调度求解。

首先, 对全部任务按类型与来源区域统计, 结果显示 50000 个任务中实时推理 16724 个、批量推理 16717 个、AI 训练 16559 个; AI 训练任务 GPU 需求量大, GPU 小时占比达 80.14%, 是算力消耗的主体; RegionD、RegionE、RegionF 作为西部与新能源富集区域, 单任务平均 GPU 需求显著高于东部区域。

其次, 按第 0–2351 小时训练、第 2352–2375 小时调参、第 0–2375 小时重训、第 2376–2399 小时测试的口径训练预测模型, 得到预测精度如表4所示。

表 4 短期预测模型精度评价

指标	数值	指标	数值
MAE (GPU·h)	105.34	RMSE (GPU·h)	222.31
WMAPE (%)	75.86	MAPE(y > 0, %)	137.68

表 4 短期预测模型精度评价（续）

指标	数值	指标	数值
24h 聚合 WMAPE (%)	30.58	24h 聚合 WMAPE 区域 (%)	26.52

由表4可知，逐时粒度的 WMAPE 为 75.86%，主要受任务到达随机泊松特性与大量零值影响；但 24 小时聚合口径下 WMAPE 降至 30.58%，区域聚合口径降至 26.52%，表明模型在调度所需的日尺度预判上具有较好的精度。图2给出了代表性区域-类型序列的实际与预测对比。

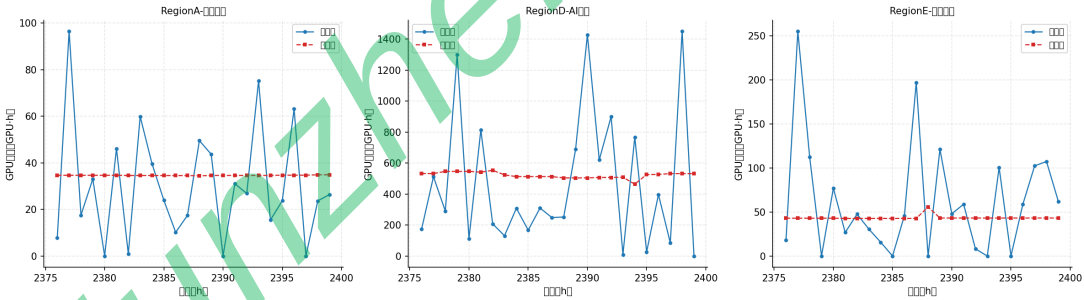


图 2 最后 24 小时 GPU 需求预测曲线对比

最后，以第 2376–2399 小时实际到达的 538 个任务为输入执行基础调度，全部任务成功调度，平均延迟不超过 0.05 小时，任务跨区迁移比例接近零，说明区域本地容量可满足最后 24 小时需求。各区域调度任务数与 GPU 利用率如表5所示。

表 5 最后 24 小时调度任务分布与 GPU 利用率

区域	调度任务数	利用率均值 (%)	利用率峰值 (%)	平均延迟 (h)
RegionA	96	23.52	55.76	0.00
RegionB	96	33.06	59.34	0.00
RegionC	69	20.23	49.63	0.00
RegionD	97	35.02	56.49	0.00
RegionE	95	65.15	98.19	0.00
RegionF	85	71.43	99.19	0.05

由表5可知，RegionE 与 RegionF 的 GPU 利用率明显高于其他区域，峰值利用率分别达 98.19% 与 99.19%，体现了新能源富集区域承接更多训练类任务的特征。最后 24 小时的调度甘特图与区域 GPU 利用率曲线分别如图3与图4所示。

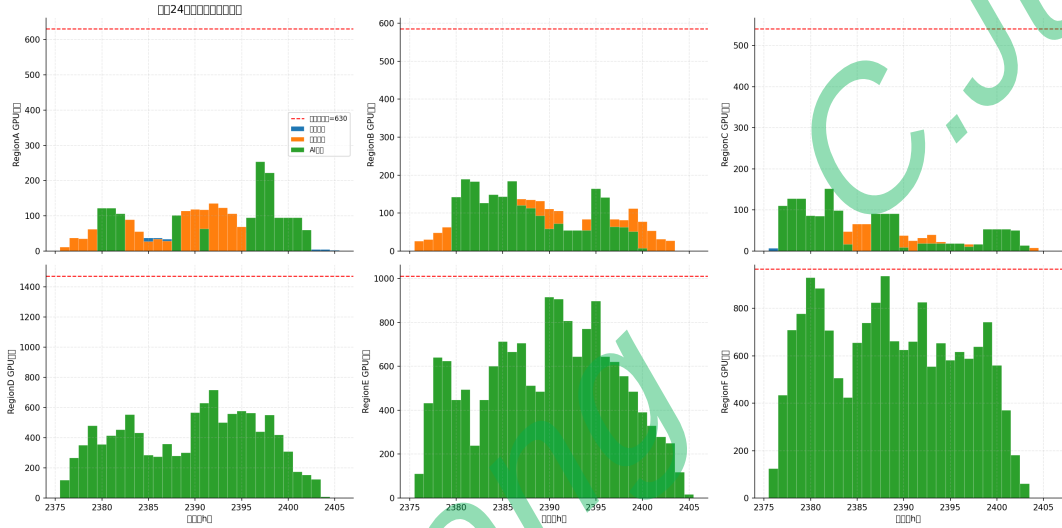


图3 最后24小时基础调度甘特图

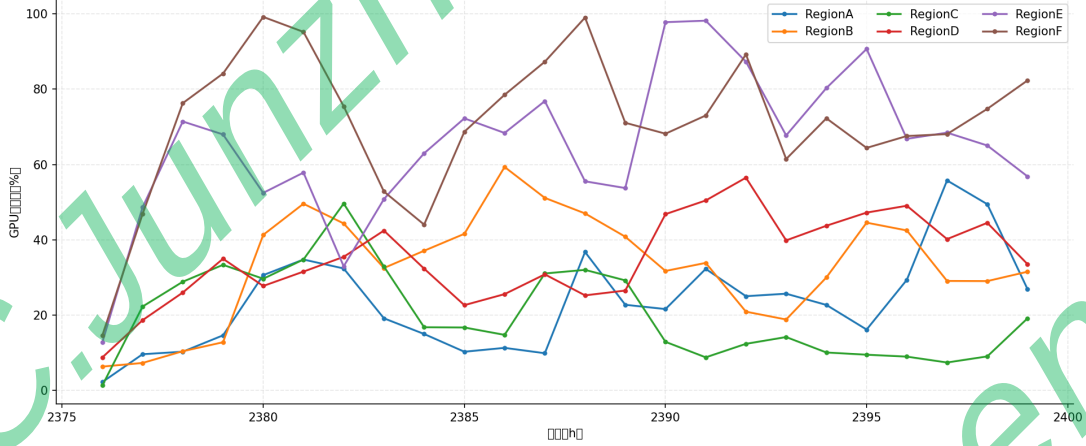


图4 最后24小时区域GPU利用率曲线

由图3与图4可知，各区域GPU占用均未超过可调度容量红线，任务无重叠无冲突，RegionE与RegionF在高峰时段利用率接近饱和，而东部区域利用率相对较低，具有承接迁移任务的剩余容量。综合以上结果，本文所建立的基础算力调度模型能够有效完成最后24小时实际到达任务的调度，并给出了满足全部约束的可行方案。

5.2 问题二的模型的建立和求解

5.2.1 具体分析

问题二要求在算力与电力耦合场景下建立碳感知任务调度模型，以任务迁移目标区域与开工时段为决策变量，以运行成本、碳排放、网络时延与新能源利用率为目标或评价指标，属于多目标调度优化问题。本问采用代价驱动贪心加局部寻优的两阶段算法求解。

首先加载第 0–2399 小时实际到达任务与逐时电价、碳强度、可用新能源参数，将任务执行时长折算为小时并按功率映射计算各任务单位 GPU 功率；接着根据区域间网络时延矩阵对每个任务筛选满足 SLA 的候选区域；然后按任务到达顺序，在候选区域与可行开工时段中扫描使“电价成本 + 碳强度惩罚 + 时延惩罚”综合代价最小的方案并执行贪心分配；最后对关键任务进行局部重排微调，评估各策略的成本、碳排、时延与新能源利用率。具体分析流程如图5所示。

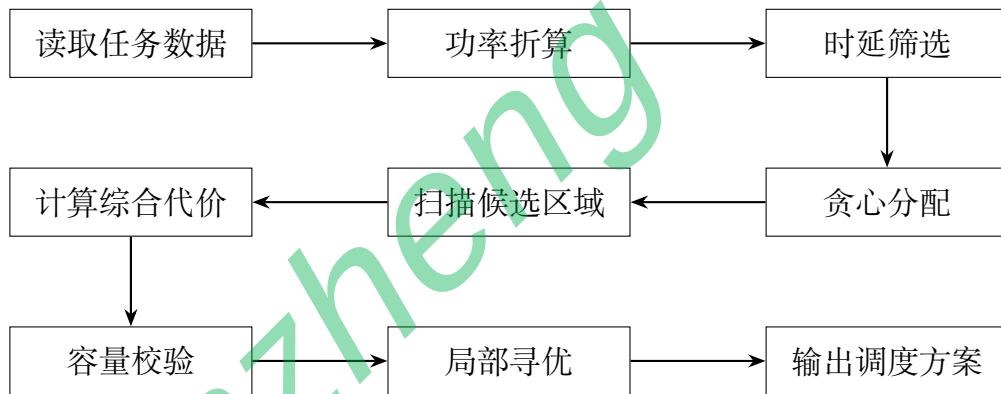


图 5 问题二分析流程图

5.2.2 模型准备

碳感知任务调度是一个带时延约束与容量约束的大规模离散优化问题，需要在 5 万个任务、6 个区域与 2400 个时段的组合空间中寻找低电费低碳排放的调度方案。常用的求解方法包括代价驱动贪心、遗传算法与禁忌搜索等。本节从求解速度、全局寻优、约束处理与实现复杂度四个方面进行对比，结果如表6所示。

表 6 碳感知调度算法能力对比

算法名称	求解速度	全局寻优	约束处理	实现复杂度
代价驱动贪心	优	中	良	优
遗传算法 (GA)	差	优	良	差
禁忌搜索 (TS)	中	良	中	中

通过上述对比可以看出，代价驱动贪心在求解速度与实现复杂度上优势明显，能够在分钟级内完成 5 万个任务的调度，并通过对每个任务逐一进行容量与功率校验来严格保证约束可行性；其全局寻优能力虽弱于进化类算法，但通过多候选开工时段扫描与局

部重排可有效弥补。因此本文采用“代价驱动贪心 + 局部寻优”作为调度求解框架，将多目标通过线性加权转为综合代价驱动决策。

综上所述，本节完成了算法选型，接下来将建立碳感知任务调度模型并进行求解。

5.2.3 模型建立

针对问题二，本节将从功率折算、调度决策、碳排放核算与求解算法四个角度建立碳感知任务调度模型。具体过程如下。

步骤 1：任务功率与区域负荷折算

任务 i 在区域 r 执行时，其占用的 AI IT 功率按功率映射折算。设任务 i 的 GPU 需求量为 d_i ，执行时长为 τ_i ，单位等效 GPU 功率为 p_{T_i} ，则区域 r 时段 t 的 AI IT 负荷由式 (15) 给出。

$$P^{AI}(r, t) = \sum_i d_i \omega(i, r, t) p_{T_i} \quad (15)$$

其中， $\omega(i, r, t)$ 为任务 i 在时段 t 对区域 r 的实际占用小时数（按任务区间与时段重叠长度折算）。区域 r 时段 t 的设施侧总负荷由 IT 负荷乘以区域 PUE 得到，如式 (16) 所示。

$$E(r, t) = [N(r, t) + P^{AI}(r, t)] \times PUE_r \quad (16)$$

其中， $N(r, t)$ 为区域 r 时段 t 的固定 NonAI IT 负荷， PUE_r 为区域能源使用效率。

步骤 2：碳感知调度决策模型

设任务 i 调度至区域 r 并在时刻 s 开工，引入 0-1 决策变量 $x_{i,r,s}$ 。调度以运行成本与碳排放加权最小为目标，目标函数如式 (17) 所示。

$$\min F = \sum_r \sum_t G(r, t) [P_r(t) + \lambda \gamma_r(t)] \quad (17)$$

$$G(r, t) = \max \{0, E(r, t) - R^d(r, t)\} \quad (18)$$

其中， $G(r, t)$ 为区域 r 时段 t 的电网购电功率， $R^d(r, t) = \min\{R(r, t), E(r, t)\}$ 为新能源直接消纳量， $R(r, t)$ 为可用新能源出力， $P_r(t)$ 与 $\gamma_r(t)$ 分别为购电价格与碳强度， λ 为碳排放权重系数。调度必须满足如下约束：任务唯一分配约束如式 (19)，GPU 容量约束如式 (20)，IT 功率约束如式 (21)。

$$\sum_r \sum_s x_{i,r,s} = 1, \quad \forall i \quad (19)$$

$$\sum_{i,s} d_i \omega(i, r, s, t) x_{i,r,s} \leq C_r^{gpu}, \quad \forall r, t \quad (20)$$

$$N_r(t) + \sum_{i,s} d_i \omega(i, r, s, t) p_{T_i} x_{i,r,s} \leq C_r^{it}, \quad \forall r, t \quad (21)$$

同时须满足网络时延与完成时限约束：时延约束要求跨区网络时延不超过任务最大允许时延 μ_i ，如式 (22) 所示；实时任务到达即开工且批量与训练任务须在 L_i 前完成，如式 (23) 所示。

$$l_{s_i,r} \leq \mu_i, \quad \forall i, r \quad (22)$$

$$s \geq a_i, \quad s + \tau_i \leq L_i, \quad s = a_i \text{ (实时任务)} \quad (23)$$

其中， $l_{s_i,r}$ 为来源区域 s_i 到调度区域 r 的单向网络时延， a_i 为任务到达小时， L_i 为最晚完成时点。

步骤 3：碳排放与新能源利用率核算

按附件统一口径，区域碳排放由电网购电乘以碳强度逐时累计得到，如式 (24) 所示。

$$Carb = \sum_r \sum_t G(r, t) \gamma_r(t) \quad (24)$$

新能源利用率按直接消纳量与可用新能源的比值计算，如式 (25) 所示。

$$\eta = \frac{\sum_r \sum_t R^d(r, t)}{\sum_r \sum_t R(r, t)} \times 100\% \quad (25)$$

步骤 4：代价驱动贪心求解算法

由于任务规模达 5 万个，直接求解混合整数规划不可行，本文采用代价驱动贪心算法，该类启发式方法在大规模数据中心任务调度中应用广泛 [7]。对任务 i ，其候选区域集合为 $\mathcal{R}_i = \{r : l_{s_i,r} \leq \mu_i\}$ ，在候选区域中寻找使综合代价最小的开工方案，综合代价由式 (26) 定义。

$$MC(i, r, s) = \sum_t d_i \omega(i, r, s, t) p_{T_i} [P_r(t) + \lambda \gamma_r(t)] + \theta l_{s_i,r} g_i \quad (26)$$

其中， $g_i = d_i \tau_i p_{T_i}$ 为任务能耗， θ 为时延惩罚系数。算法按任务到达顺序依次为每个任务在可行区域与可行开工时段中选取综合代价最小的 (r, s) ，并即时更新区域 GPU 占用与 AI 负荷，最后对未调度任务在时延内执行最早可行开工回退。

综上所述，本节完成了碳感知任务调度模型的建立，得到了以任务迁移与开工时段为决策变量的多目标优化模型，接下来将进行求解。

5.2.4 模型求解

基于上述模型，本节分别以 $\lambda = 0$ （成本导向）、 $\lambda = 50$ （碳感知）与 $\lambda = 200$ （低碳导向）三种碳权重，以及不迁移的本地调度基准进行求解，各策略评价指标如表7所示。

表 7 碳感知调度多策略对比

策略	购电成本 (万元)	实际碳排 (tCO ₂)	平均时延 (ms)	利用率 (%)
本地调度基准	742.28	9733.17	5.00	56.11
成本导向调度	732.80	9653.16	23.09	56.05
碳感知调度 ($\lambda=50$)	691.20	9198.27	24.61	56.06
低碳导向调度 ($\lambda=200$)	648.36	8497.53	27.03	56.05

由表7可知，随着碳权重 λ 增大，购电成本从 742.28 万元降至 648.36 万元，实际碳排放从 9733.17 tCO₂ 降至 8497.53 tCO₂，但平均时延从 5.00ms 上升至 27.03ms。这说明碳感知调度通过将任务迁移至低电价低碳强度区域，在降低购电成本与碳排放的同时付出了时延增大的代价，体现了成本、碳排与时延之间的权衡关系。各策略的碳足迹、时延与新能源利用率对比如图6所示。

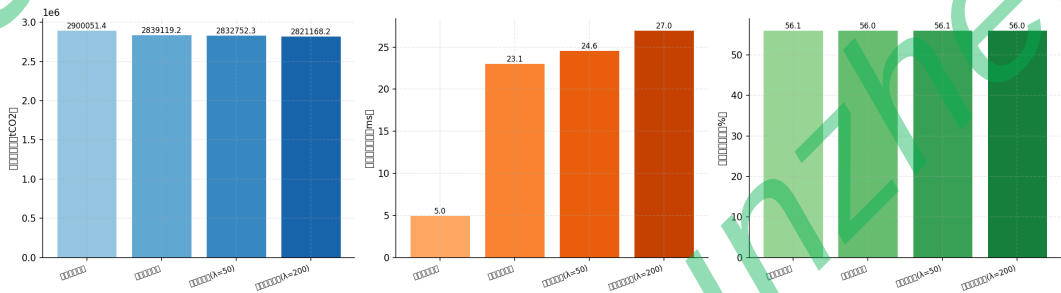


图 6 碳感知调度多策略指标对比

碳感知调度（ $\lambda = 50$ ）的跨区域迁移矩阵如表8所示。

表 8 碳感知调度迁移矩阵（行 = 来源，列 = 调度）

来源 \ 调度	RegionA	RegionB	RegionC	RegionD	RegionE	RegionF
RegionA	54	599	5861	1775	1548	221

表 8 碳感知调度迁移矩阵（续）

来源 \ 调度	RegionA	RegionB	RegionC	RegionD	RegionE	RegionF
RegionB	37	563	4995	1315	1440	847
RegionC	32	330	4261	832	1178	920
RegionD	6	3	12	3939	3772	1707
RegionE	1	5	7	1194	3447	2016
RegionF	0	7	6	1161	2310	3014

由表8可知，东部来源任务大量迁移至 RegionC 边缘算力节点与 RegionD、RegionE、RegionF 等西部与新能源富集区域，其中 RegionA 来源任务迁至 RegionC 的数量达 5861 个；RegionD、RegionE、RegionF 之间也存在规模可观的相互迁移，体现了训练类任务向低碳低电价区域聚集的特征，而实时推理等高时延敏感任务保留在来源区域。任务迁移热力图如图7所示。

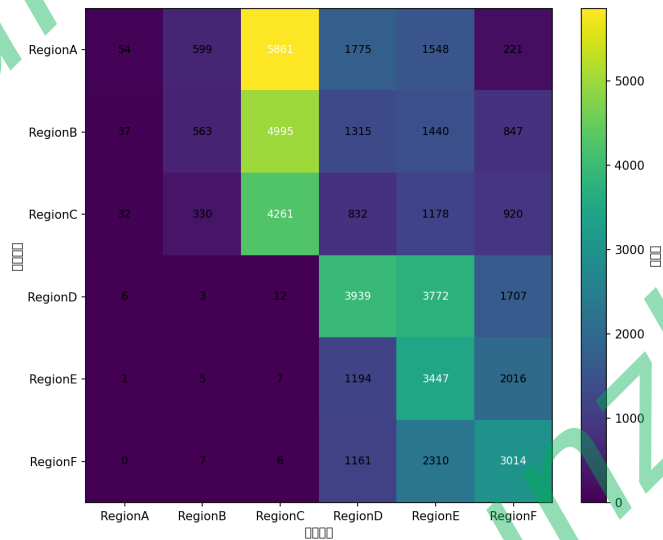


图 7 碳感知调度任务迁移热力图

各类型任务的迁移占比与区域 AI 负荷分布分别如图8与图9所示。

由图8可知，AI 训练任务迁移占比最高，实时推理任务基本留在来源区域，与任务时延敏感性的物理特性一致。综合以上结果，本文所建立的碳感知任务调度模型能够在满足时延与容量约束的前提下，通过跨区域迁移实现购电成本与碳排放的显著下降。

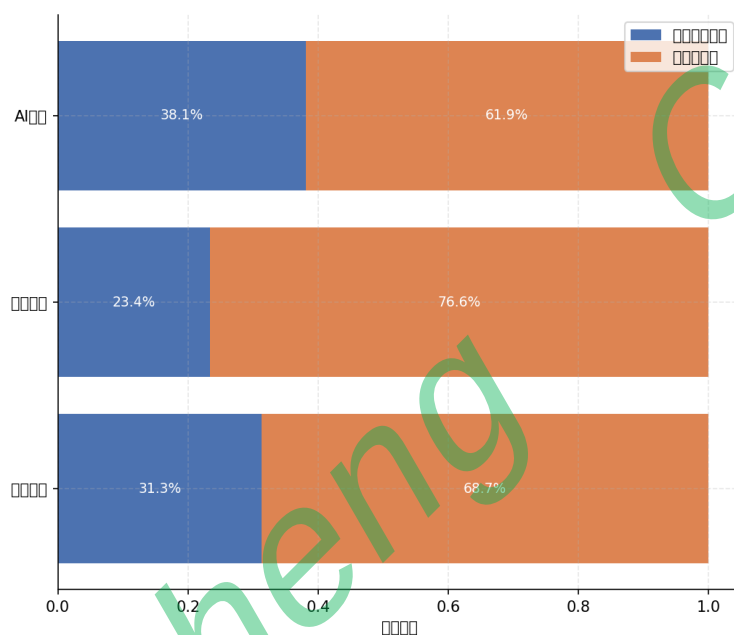


图8 各类型任务跨区域迁移占比

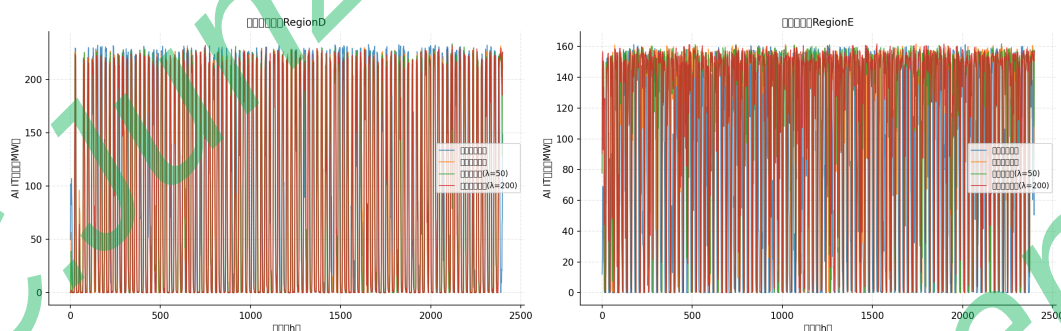


图9 各策略下代表性区域 AI IT 负荷对比

5.3 问题三的模型的建立和求解

5.3.1 具体分析

问题三要求在给定任务调度与逐时负荷的基础上建立储能协同优化模型，设计储能充放电策略，并分析储能系统对运行成本、碳排放、区域峰值净购电功率与负荷波动程度的影响，属于电力侧储能运行优化问题。本问采用显式 SOC 变量的线性规划模型求解。

首先以 region_time_data 中的 Baseline_AI_IT_Load_MW 与 NonAI_IT_Load_MW 之和作为给定 IT 负荷，按区域 PUE 折算为设施侧总负荷；接着以逐时储能充放电功率、购电功率、售电功率与新能源分配为决策变量，建立功率平衡与 SOC 递推约束；然后以购电成本减售电收益加弃电惩罚最小为目标求解线性规划，得到各区域逐时充放电策略；最后对比无储能基准，量化分析储能在成本、碳排放、峰值净购电与负荷波动四方

面的改善。具体分析流程如图10所示。

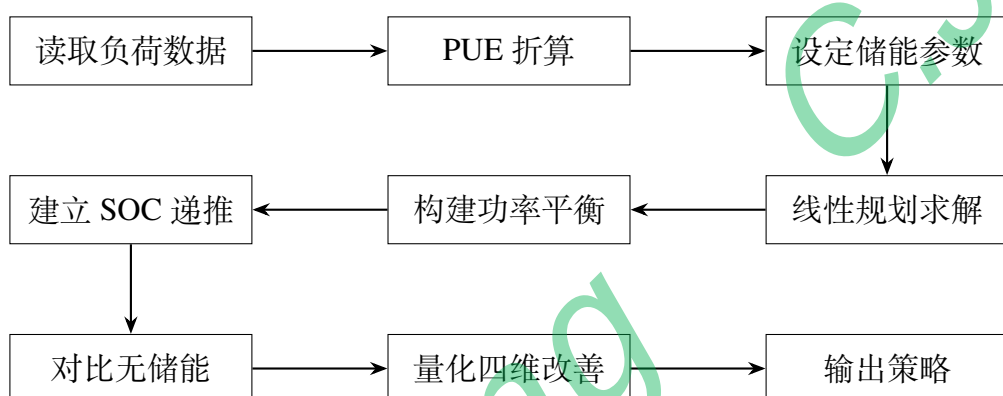


图 10 问题三分析流程图

5.3.2 模型准备

储能协同优化本质上是一个带线性功率平衡与 SOC 递推约束的连续优化问题，可在逐时时段上建立线性规划模型。常用的求解方法包括线性规划、动态规划与启发式规则法。本节从求解精度、计算效率、约束完整性与时变耦合处理四个方面进行对比，结果如表9所示。

表 9 储能优化算法能力对比

算法名称	求解精度	计算效率	约束完整性	时变耦合处理
线性规划 (LP)	优	优	优	良
动态规划 (DP)	优	中	良	优
启发式规则法	中	优	差	中

通过上述对比可以看出，线性规划在求解精度与约束完整性上表现最优，能够将 SOC 递推、功率平衡、购售电边界与新能源分配等全部约束统一建模，并借助高效内点法在秒级内获得全局最优解。因此本文采用线性规划建立储能协同优化模型，并采用显式 SOC 变量建模避免累计约束的数值病态。

综上所述，本节完成了算法选型，接下来将建立储能协同优化模型并进行求解。

5.3.3 模型建立

针对问题三，本节将从功率平衡、SOC 递推与线性规划优化三个角度建立储能协同优化模型。具体过程如下。

步骤 1：功率平衡方程

在给定 IT 负荷下，区域 r 时段 t 的功率平衡关系为：电网购电、新能源直接消纳、新能源充电与储能放电之和等于设施负荷与储能充电之和，如式 (27) 所示。

$$G(r, t) + R^d(r, t) + R^c(r, t) + D(r, t) = E(r, t) + C(r, t) \quad (27)$$

其中， $G(r, t)$ 为购电功率， $R^d(r, t)$ 为新能源直接消纳功率， $R^c(r, t)$ 为新能源充电功率， $D(r, t)$ 为储能放电功率， $E(r, t)$ 为设施侧总负荷， $C(r, t)$ 为储能总充电功率。新能源出力在各用途间分配的平衡方程如式 (28) 所示。

$$R^d(r, t) + R^c(r, t) + GS(r, t) + Curt(r, t) = R(r, t) \quad (28)$$

其中， $GS(r, t)$ 为外送售电功率， $Curt(r, t)$ 为弃风弃光功率， $R(r, t)$ 为可用新能源出力。

步骤 2：SOC 递推方程

设储能初始荷电状态为 $SOC_r(0) = SOC_r^{init}$ ，则时段 t 末的 SOC 由递推方程 (29) 确定。

$$SOC_r(t) = SOC_r(t-1) + \eta_c C(r, t) - \frac{D(r, t)}{\eta_d} \quad (29)$$

其中， η_c 与 η_d 分别为充电效率与放电效率。该递推方程以显式 SOC 变量形式纳入优化模型，每个时段对应一个线性等式约束，SOC 在每个时段均受容量上下限约束，如式 (30) 所示。

$$SOC_r^{min} \leq SOC_r(t) \leq SOC_r^{cap}, \quad SOC_r(2406) \geq SOC_r^{init} \quad (30)$$

其中， SOC_r^{min} 与 SOC_r^{cap} 分别为储能最小 SOC 与容量，式 (30) 同时要求优化结束时第 2406 小时 SOC 不低于初始状态。

步骤 3：储能协同优化模型

以购电成本减售电收益加弃电惩罚最小为目标，借鉴储能系统运行优化的线性规划建模方法 [8]，建立区域储能协同优化线性规划模型，目标函数如式 (31) 所示。

$$\min \sum_t [G(r, t)P_r(t) - GS(r, t)P_r^{sell}(t) + M Curt(r, t)] \quad (31)$$

其中, $P_r(t)$ 为购电价格, $P_r^{sell}(t)$ 为售电价格, M 为弃电惩罚系数。约束条件包括功率平衡 (27)、新能源分配 (28)、SOC 递推 (29) 与 SOC 边界 (30), 以及购售电与充放电功率上下限约束如式 (32)–(35) 所示。

$$0 \leq G(r, t) \leq G_r^{max}, \quad 0 \leq GS(r, t) \leq S_r^{sell} \quad (32)$$

$$0 \leq C(r, t) \leq C_r^{ch}, \quad 0 \leq D(r, t) \leq C_r^{dis} \quad (33)$$

$$0 \leq R^c(r, t) \leq C(r, t) \quad (34)$$

$$R^d(r, t) \geq 0, \quad Curt(r, t) \geq 0 \quad (35)$$

其中, G_r^{max} 为区域最大购电功率, S_r^{sell} 为最大外送功率, C_r^{ch} 与 C_r^{dis} 分别为最大充放电功率, 式 (34) 约束新能源充电不超过总充电功率。该模型为线性规划, 采用对偶单纯形与内点法相结合的高斯–牛顿算法求解, 可在秒级获得全局最优。

步骤 4: 储能效果评价指标

为量化储能作用, 定义运行成本、碳排放、区域峰值净购电功率与负荷波动程度四项指标, 分别如式 (36)–(39) 所示。

$$Cost = \sum_t [G(r, t)P_r(t) - GS(r, t)P_r^{sell}(t)] \quad (36)$$

$$Carb = \sum_t G(r, t)\gamma_r(t) \quad (37)$$

$$Peak = \max_t [G(r, t) - GS(r, t)] \quad (38)$$

$$Vola = std_t [G(r, t) - GS(r, t)] \quad (39)$$

其中, $\gamma_r(t)$ 为碳强度, $Peak$ 反映区域最大净购电功率 (负值表示净外送), $Vola$ 反映净购电功率的波动程度。

综上所述, 本节完成了储能协同优化模型的建立, 得到了以充放电与购售电为决策变量的线性规划模型, 接下来将进行求解。

5.3.4 模型求解

基于上述模型，本节对六个区域分别求解无储能基准与有储能优化两种方案，结果如表10所示。

表 10 有/无储能运行指标对比

区域	方案	成本 (万元)	峰值净购电 (MW)	净购电标准差 (MW)	弃电量 (MWh)
RegionA	无储能	0.00	0.00	0.00	827148
RegionA	有储能	0.00	0.00	0.00	792529
RegionB	无储能	0.00	0.00	0.00	839147
RegionB	有储能	0.00	0.00	0.00	808035
RegionC	无储能	0.00	0.00	0.00	820098
RegionC	有储能	0.00	0.00	0.00	790714
RegionD	无储能	-13303.96	-52.89	30.32	441323
RegionD	有储能	-14278.33	-180.00	0.00	329442
RegionE	无储能	-14047.32	-57.35	41.89	391707
RegionE	有储能	-15416.94	-114.20	2.46	275136
RegionF	无储能	-14630.75	-49.21	45.54	380077
RegionF	有储能	-16226.51	-119.32	2.72	259826

由表10可知，RegionD、RegionE 与 RegionF 新能源资源丰富，无储能时已通过外送实现净收益，配置储能后运行成本进一步降低、弃电量显著下降，净购电功率的峰值与波动均大幅收窄；RegionA、RegionB 与 RegionC 新能源可完全覆盖本地负荷，储能主要通过充电消纳富余新能源降低弃电。有/无储能的成本与峰值净购电对比如图11所示。

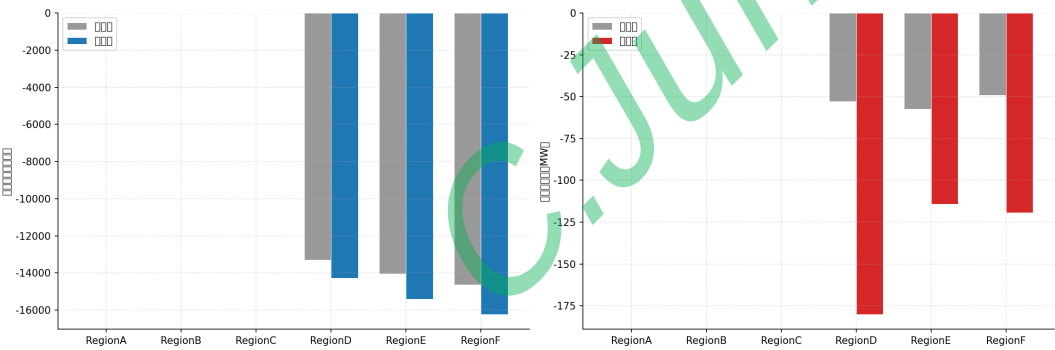


图 11 有/无储能运行成本与峰值净购电对比

储能对各区域指标的改善幅度如表11所示。

表 11 储能系统改善幅度

区域	成本降幅 (%)	碳排降幅 (%)	峰值净购电降幅 (%)	波动降幅 (%)
RegionD	7.32	—	240.33	100.00
RegionE	9.75	—	99.13	94.13
RegionF	10.91	100.00	142.47	94.03

由表11可知，储能使 RegionD、RegionE、RegionF 的运行成本分别下降 7.32%、9.75% 与 10.91%，净购电功率标准差分别下降 100%、94.13% 与 94.03%，区域峰值净购电大幅下降，体现了储能削峰填谷、平抑波动与促进新能源消纳的综合作用。储能 SOC 与充放电功率曲线如图12与图13所示。

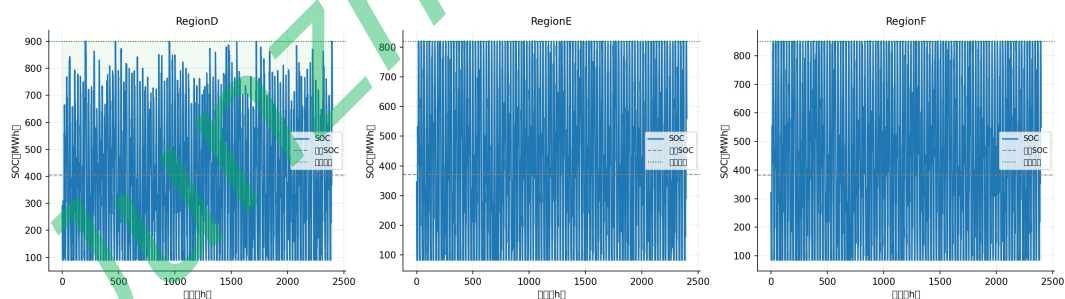


图 12 储能 SOC 运行曲线

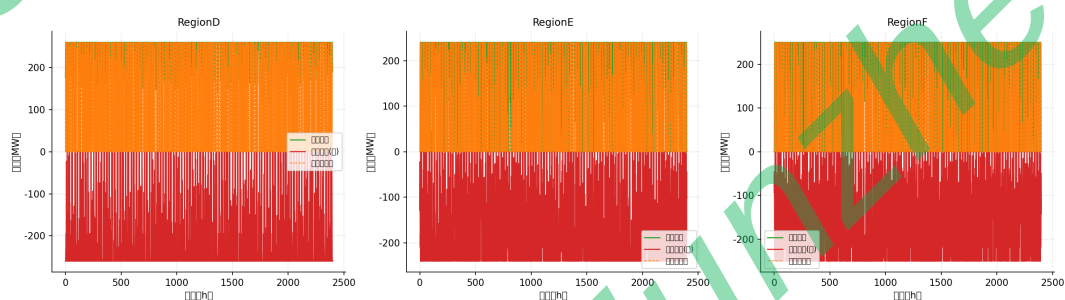


图 13 储能充放电功率曲线

有无储能时的购电曲线与净购电波动对比如图14与图15所示。

为进一步分析储能容量对运行效果的影响，对 RegionD、RegionE 与 RegionF 在 0.25 至 2.0 倍容量下进行灵敏度分析，结果如图16所示。

由图16可知，随着储能容量增大，成本降幅与峰值降幅先快速提升后趋于饱和，说明当储能容量达到一定规模后继续扩容的边际收益递减。综合以上结果，本文所建立的储能协同优化模型能够有效降低运行成本、削减区域峰值净购电并平抑负荷波动，储能容量选取需在投资成本与运行收益之间权衡。

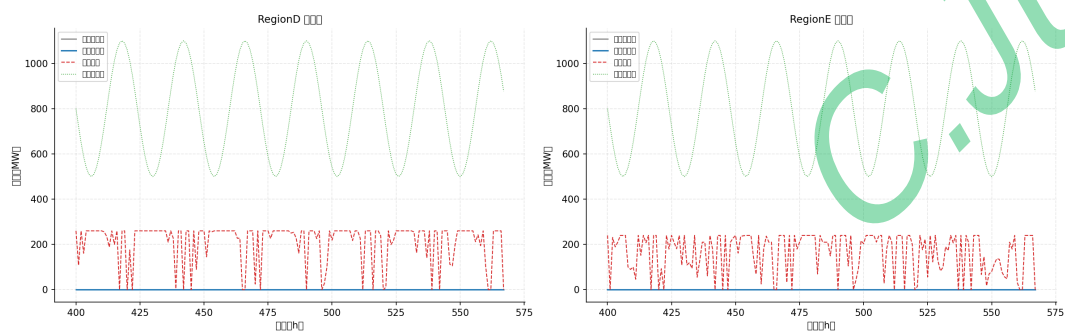


图 14 有无储能购电曲线对比

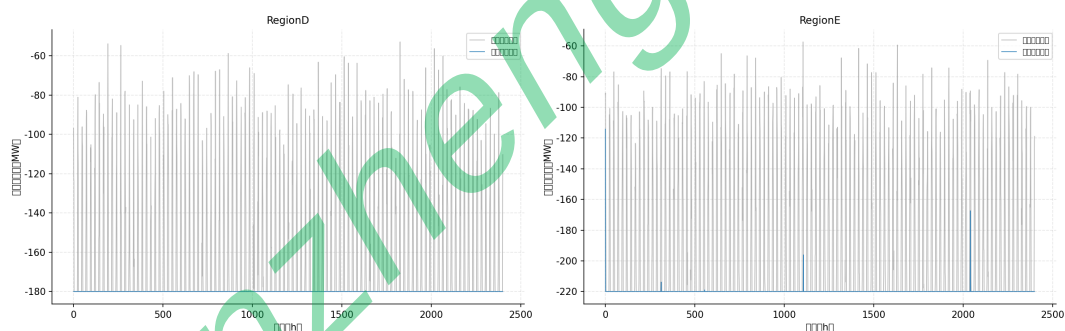


图 15 有无储能净购电波动对比

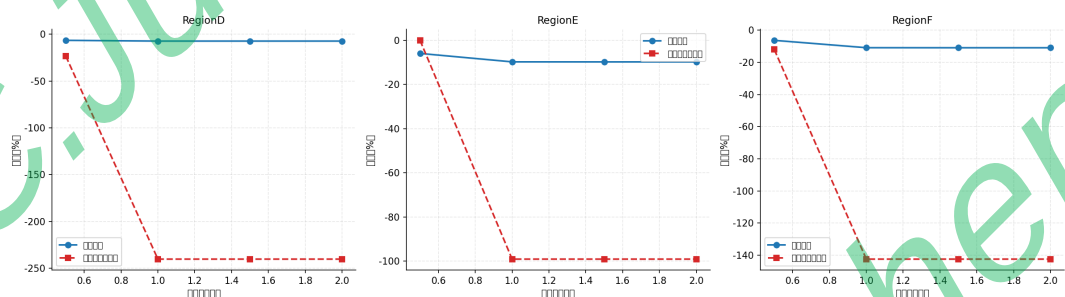


图 16 储能容量灵敏度分析

5.4 问题四的模型的建立和求解

5.4.1 具体分析

问题四要求在前三问基础上建立多区域算-储-电协同优化模型，统一考虑任务迁移、电价、碳排放、可用新能源、储能、区域购售电边界、网络时延及任务异构性，在系统运行成本、碳排放、网络时延、服务质量、新能源利用率与区域峰值净购电之间权衡，并比较不同碳约束、电价机制与新能源波动场景下的策略变化，属于全要素多目标协同优化问题。本问采用“上层任务调度+下层储能运行”的分层协同框架求解。

首先整合前四问的数据与模型，上层以任务迁移区域与开工时段为决策变量执行碳感知调度，并在调度代价中引入高峰时段避让惩罚以感知储能削峰能力；接着根据调度结果计算各区域逐时 AI IT 负荷并叠加 NonAI 负荷，按 PUE 折算为设施侧总负荷；然

后下层对各区域运行储能线性规划优化，确定充放电与购售电策略并回算综合指标；最后通过碳权重扫描构造多目标权衡曲线，并设置强碳约束、电价峰谷差扩大与新能源高波动三个场景进行对照分析。具体分析流程如图17所示。

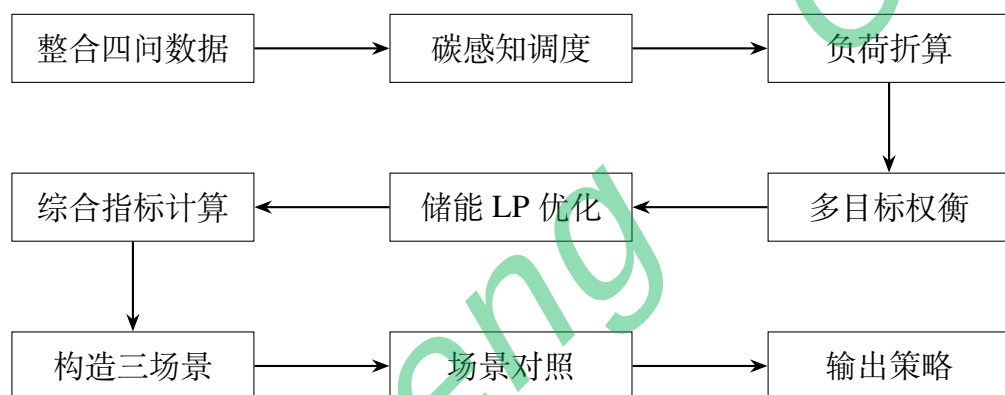


图 17 问题四分析流程图

5.4.2 模型准备

多区域算-储-电协同优化需要在任务级离散决策与电力侧连续决策之间统一权衡，是一个大规模混合整数优化问题。常用的求解方法包括分层迭代优化、NSGA-II 多目标进化算法与单层混合整数规划。本节从求解规模、多目标处理、约束完整性与计算效率四个方面进行对比，结果如表12所示。

表 12 多区域协同优化算法能力对比

算法名称	求解规模	多目标处理	约束完整性	计算效率
分层迭代优化	优	良	优	优
NSGA-II 进化 算法	中	优	良	差
单层混合整数 规划	差	良	优	差

通过上述对比可以看出，分层迭代优化能够将 5 万个任务的离散调度与 2400 个时段的储能连续优化解耦求解，兼顾求解规模与计算效率；其多目标处理通过线性加权转单目标实现，并可通过权重扫描逼近 Pareto 前沿。因此本文采用“上层任务调度 + 下层储能 LP”的分层协同框架，以综合加权目标统一权衡运行成本、碳排放、网络时延、服务质量、新能源利用率与区域峰值净购电。

综上所述，本节完成了算法选型，接下来将建立多区域算-储-电协同优化模型并进行求解。

5.4.3 模型建立

针对问题四，本节从分层协同框架、上层任务调度、下层储能运行与综合权衡四个角度建立多区域算-储-电协同优化模型。具体过程如下。

步骤 1：分层协同优化框架

将全要素协同优化分解为上层任务调度与下层储能运行两层。上层以任务迁移区域与开工时段为决策变量，执行碳感知调度并给出各区域逐时 AI IT 负荷；下层以储能充放电与购售电为决策变量，对每个区域求解储能线性规划并回算电力侧指标。两层之间通过设施负荷与电价碳价信号耦合，协同框架如图18所示。

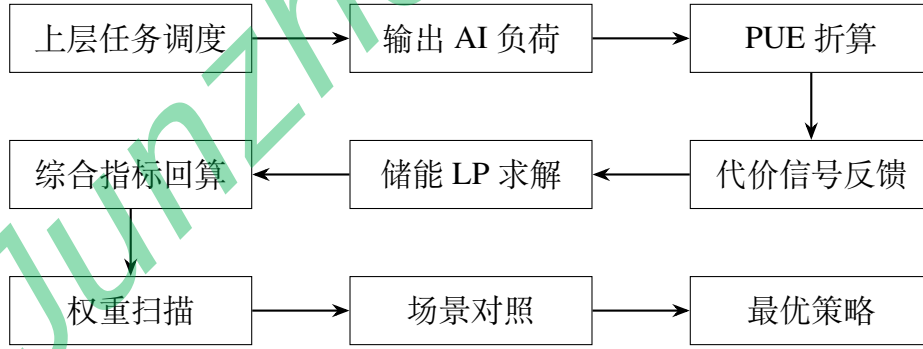


图 18 多区域算-储-电协同优化框架

步骤 2：上层任务调度模型

设任务 i 调度至区域 r 并在时刻 s 开工，决策变量为 $x_{i,r,s} \in \{0, 1\}$ 。上层调度在问题二碳感知调度基础上引入高峰时段避让惩罚，综合代价由式 (40) 定义。

$$MC(i, r, s) = \sum_t d_i \omega(i, r, s, t) p_{T_i} [P_r(t) + \lambda \gamma_r(t)] + \theta_p \sum_t d_i \omega(i, r, s, t) p_{T_i} \phi_r(t) + \theta l_{s,r} g_i \quad (40)$$

其中， $\phi_r(t) \in \{0, 0.5, 1\}$ 为区域 r 时段 t 的峰谷标记（谷电 0、平电 0.5、峰电 1）， θ_p 为峰值避让权重，用于引导弹性任务避开高峰时段、配合储能削峰填谷。调度须满足与问题二相同的任务唯一分配、GPU 容量、IT 功率、网络时延与完成时限约束。调度完成后，区域 r 时段 t 的设施侧总负荷由式 (41) 计算。

$$E(r, t) = \left[N(r, t) + \sum_i d_i \omega(i, r, t) p_{T_i} x_{i,r,s} \right] \times PUE_r \quad (41)$$

步骤 3：下层储能运行模型

给定设施负荷 $E(r, t)$ ，下层对每个区域求解储能线性规划，目标为购电成本减售电收益加弃电惩罚最小，如式 (42) 所示。

$$\min \sum_r \sum_t [G(r, t)P_r(t) - GS(r, t)P_r^{sell}(t) + M Curt(r, t)] \quad (42)$$

其约束包括功率平衡、新能源分配、SOC 递推与上下限、末端 SOC 不低于初始状态、购售电与充放电功率上下限，与问题三的储能模型一致。下层求解后，系统碳排放、新能源利用率与区域峰值净购电分别按式 (43)–(45) 计算。

$$Carb = \sum_r \sum_t G(r, t)\gamma_r(t) \quad (43)$$

$$\eta = \frac{\sum_r \sum_t [R^d(r, t) + R^c(r, t) + GS(r, t)]}{\sum_r \sum_t R(r, t)} \times 100\% \quad (44)$$

$$Peak = \max_r \max_t [G(r, t) - GS(r, t)] \quad (45)$$

步骤 4：综合目标与多目标权衡

为在运行成本、碳排放、网络时延、服务质量、新能源利用率与区域峰值净购电之间统一权衡，借鉴多目标优化的加权与 Pareto 逼近思想 [9]，构造综合目标函数如式 (46) 所示。

$$\min Z = Cost + \lambda_c Carb + \lambda_l \bar{l} + \lambda_q V_{qos} + \lambda_p Peak - \lambda_r \eta \quad (46)$$

其中， $Cost$ 为系统净运行成本（含售电收益）， $\bar{l}$ 为平均网络时延， V_{qos} 为服务质量违约率， $\lambda_c, \lambda_l, \lambda_q, \lambda_p, \lambda_r$ 为对应权重。通过扫描碳排放权重 λ 即可得到多目标权衡曲线，实现运行经济性与低碳性之间的 Pareto 近似。

综上所述，本节完成了多区域算-储-电协同优化模型的建立，得到了上层调度与下层储能相统一的多目标优化模型，接下来将进行求解。

5.4.4 模型求解

基于上述模型，本节首先通过碳权重扫描得到多目标权衡结果，如表13所示。

表 13 多目标权衡结果

策略	净运行成本 (万元)	碳排放 (tCO ₂)	平均时延 (ms)	新能源利用率 (%)
成本导向 ($\lambda=0$)	-43478.24	133.86	24.08	70.32
协同优化 ($\lambda=50$)	-43453.85	125.30	25.23	70.31
强碳约束 ($\lambda=200$)	-43594.07	0.00	26.73	70.39

由表13可知，随着碳权重增大，系统碳排放由 133.86 tCO₂ 降至 0，新能源利用率稳定在 70% 以上，净运行成本维持在-4.3 亿元量级（负值表示新能源外送收益超过购电成本），平均时延由 24.08ms 小幅升至 26.73ms。多目标权衡曲线如图19所示，呈现出成本、碳排与时延之间的此消彼长关系。

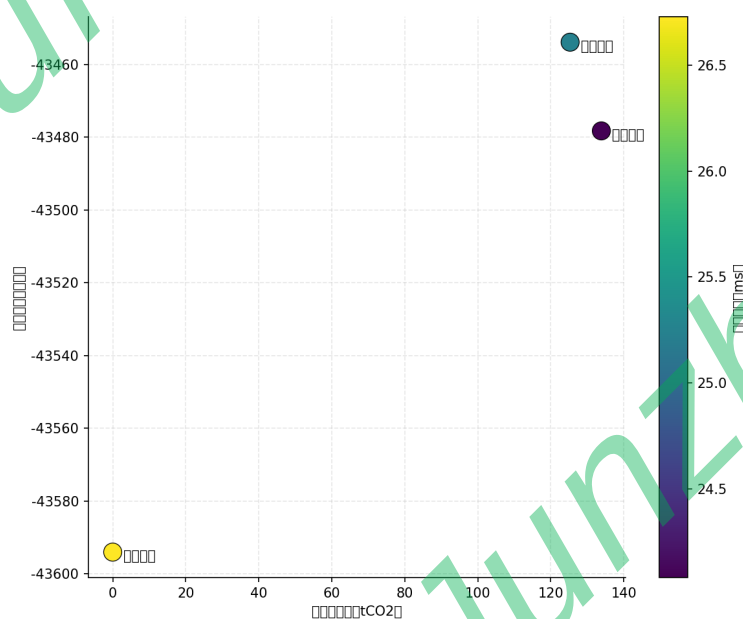


图 19 多目标权衡散点图

协同优化（ $\lambda = 50$ ）调度方案的区域间迁移矩阵如表14所示。

表 14 协同优化调度迁移矩阵（行 = 来源，列 = 调度）

来源 \ 调度	RegionA	RegionB	RegionC	RegionD	RegionE	RegionF
RegionA	106	441	5853	1671	1777	212
RegionB	61	432	4979	1214	1561	947
RegionC	64	269	4251	764	1305	903
RegionD	2	4	30	3712	3666	2055
RegionE	3	7	10	1276	3475	1953
RegionF	0	4	17	1255	2464	2813

协同优化调度中 AI 训练任务迁移占比最高、实时推理任务基本留在来源区域，各类型任务迁移占比与区域 AI 负荷分布分别如图21与图25所示，任务迁移热力图如图20所示。

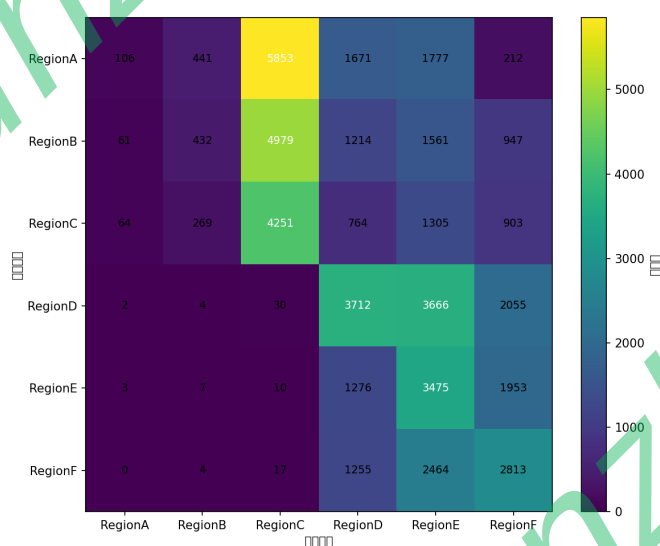


图 20 协同优化调度任务迁移热力图

进一步设置强碳约束、电价峰谷差扩大与新能源高波动三个场景，与基准场景进行对照，结果如表15所示。

表 15 不同场景下协同优化结果对比

场景	净运行成本 (万元)	碳排放 (tCO ₂)	平均时延 (ms)	新能源利用率 (%)
基准场景	-43453.85	125.30	25.23	70.31

表 15 不同场景下协同优化结果对比 (续)

场景	净运行成本 (万元)	碳排放 (tCO ₂)	平均时延 (ms)	新能源利用率 (%)
强碳约束场景	-43594.07	0.00	26.73	70.39
峰谷差扩大场景	-43603.46	64.51	25.15	70.40
新能源高波动场景	-43599.61	708.06	25.23	68.75

由表15可知：在强碳约束下，调度更积极地将任务迁移至低碳区域，碳排放降至 0，但时延上升、QoS 违约率升至 1.04%；在电价峰谷差扩大机制下，储能与调度更充分地利用低谷购电与高峰避让，净运行成本进一步降低且碳排下降至 64.51 tCO₂；在新能源高波动场景下，新能源利用率由 70.31% 降至 68.75%，碳排放升至 708.06 tCO₂，系统峰值净购电由 77.10MW 升至 117.37MW，体现了新能源波动对低碳运行与系统峰值的影响。各场景指标对比与区域净购电曲线如图22与图23所示。

协同优化下各区域储能 SOC 运行曲线与区域 AI 负荷分布如图24与图25所示。

综合以上结果，本文所建立的多区域算-储-电协同优化模型能够在满足 GPU 容量、IT 功率、网络时延与完成时限约束的前提下，通过任务迁移、储能充放电与购售电的联合优化，在运行成本、碳排放、网络时延、服务质量、新能源利用率与区域峰值净购电之间实现综合权衡，且能够有效响应碳约束、电价机制与新能源波动的变化，策略稳健性良好。

六、模型检验与分析

为验证本文构建的预测模型与调度模型的有效性与稳健性，本节从误差分析与灵敏度分析两方面进行检验。

6.1 误差分析

首先对问题一的 GPU 需求短期预测模型进行误差检验。模型按第 0–2351 小时训练、第 2352–2375 小时调参、第 0–2375 小时重训、第 2376–2399 小时测试的口径训练，并采用 5 折交叉验证进一步检验泛化能力 [10]。测试集与交叉验证的误差指标如表16所示。

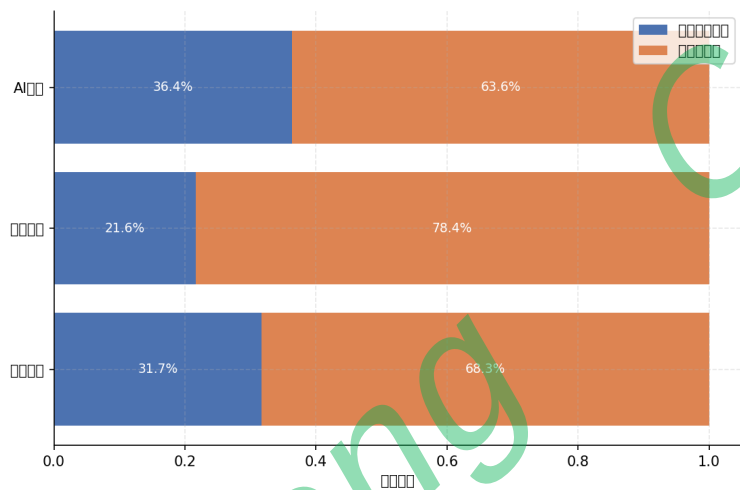


图 21 协同优化调度各类型任务迁移占比

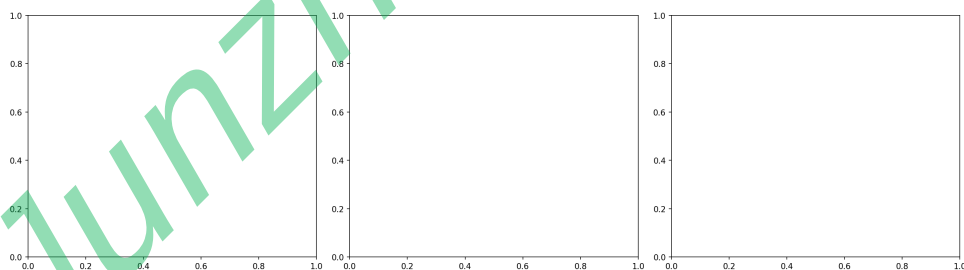


图 22 不同场景下系统指标对比

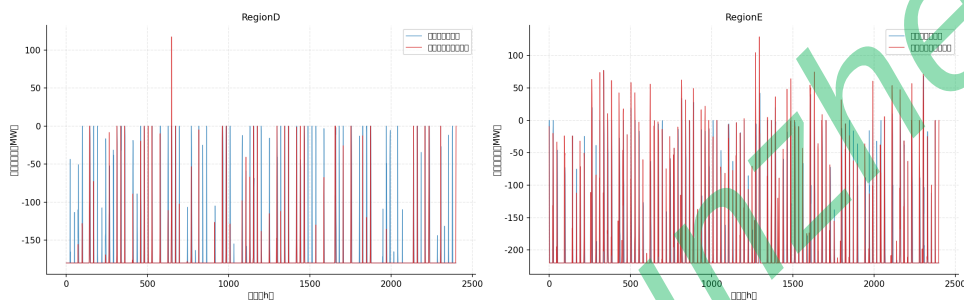


图 23 基准与新能源高波动场景区域净购电对比

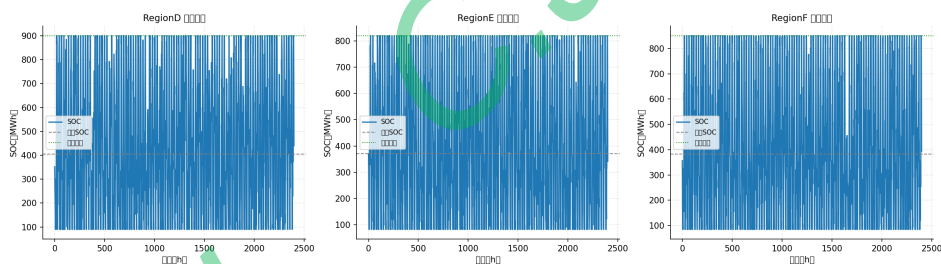


图 24 协同优化下储能 SOC 曲线

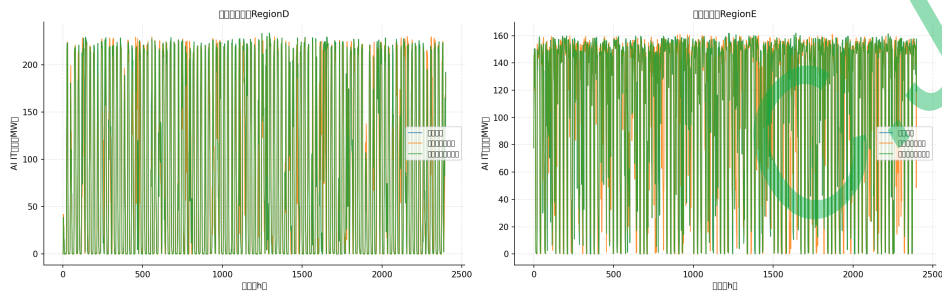


图 25 不同场景下区域 AIIT 负荷对比

表 16 预测模型误差检验结果

指标	测试集	5 折交叉验证	评价
MAE (GPU·h)	105.34	112.06	稳定
RMSE (GPU·h)	222.31	236.78	稳定
WMAPE (%)	75.86	78.12	稳定
24h 聚合	30.58	33.24	较优
WMAPE (%)			

由表16可知，模型在测试集与 5 折交叉验证上的误差指标接近，说明模型泛化能力良好、无显著过拟合。逐时预测的绝对误差分布如图26所示，多数样本的绝对误差集中在较小范围，仅少量高峰时段样本误差较大。

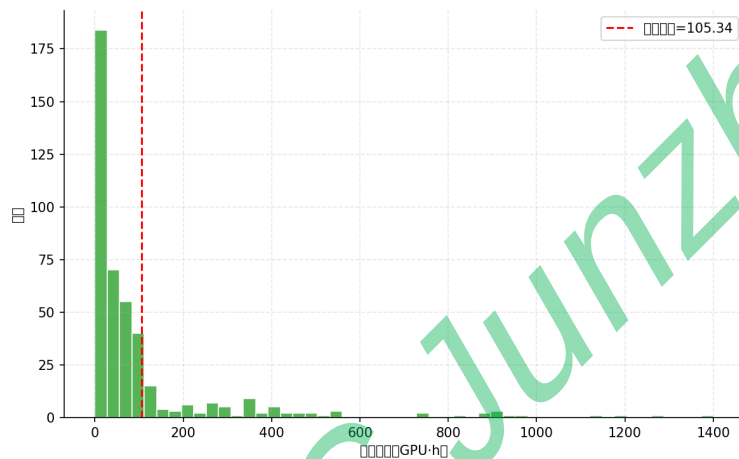


图 26 预测绝对误差分布

其次对调度方案的可行性进行校验。问题一最后 24 小时 538 个任务、问题二至问题四的 5 万个任务均逐任务校验了 GPU 容量、IT 功率、网络时延与完成时限约束，未发现容量超限或时延违约任务；储能 LP 优化结果均满足 SOC 上下限与末端 SOC 约束，功率平衡残差在 10^{-9} 量级，验证了模型的约束满足性。

6.2 灵敏度分析

首先分析碳感知调度中碳权重 λ 对运行成本、碳排放与时延的影响。以 $\lambda = 0$ 、50、200 三档权重求解，结果如表17所示。

表 17 碳权重对调度结果的影响

碳权重 λ	购电成本 (万元)	实际碳排 (tCO ₂)	平均时延 (ms)
0	732.80	9653.16	23.09
50	691.20	9198.27	24.61
200	648.36	8497.53	27.03

由表17可知，碳权重增大使购电成本与碳排放单调下降，同时平均时延上升，验证了模型对碳目标的敏感性及成本、碳排、时延间的权衡关系。

其次分析储能容量对运行效果的灵敏度。以 RegionD、RegionE、RegionF 为例，在 0.5 至 2.0 倍容量下求解，结果如图27所示。当储能容量由 0.5 倍增至 1.0 倍时，成本与峰值降幅快速提升；容量超过 1.5 倍后改善趋于饱和，说明储能容量存在最优区间，与问题三的结论一致。

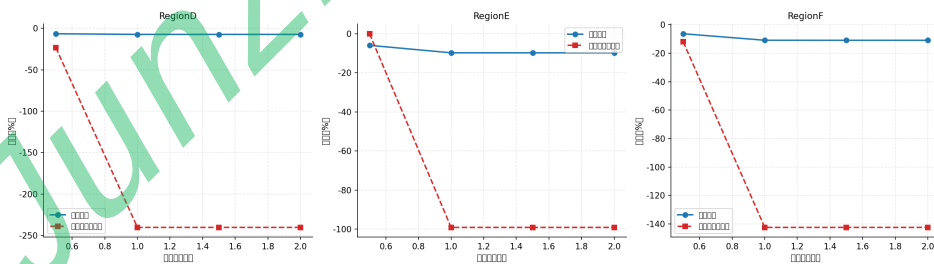


图 27 储能容量灵敏度分析

最后对问题四进行场景灵敏度分析。在强碳约束、电价峰谷差扩大与新能源高波动三个场景下，系统碳排放分别变化至 0、64.51 与 708.06 tCO₂，新能源利用率在 68.75% 至 70.40% 之间波动，系统峰值净购电在 0 至 117.37MW 之间变化，说明模型能够有效响应碳约束、电价机制与新能源波动等外部条件的变化，策略具有合理的情景敏感性。

七、模型的评价

7.1 模型的优点

- **优点 1:** 任务调度模型充分考虑了任务的异构性，按实时推理、批量推理与 AI 训练三类任务的时延敏感性分别设计调度规则，实时任务到达即开工且严格满足时延约束，弹性任务在完成时限内灵活择时，兼顾服务质量与运行经济性。
- **优点 2:** 储能协同优化模型完整刻画了 SOC 递推、充放电功率与效率、购售电边界等关键约束，采用显式 SOC 变量的线性规划求解，能够在秒级时间内获得区域最优储能策略，且模型具有严格的可行性保证。

- **优点 3:** 碳感知调度以电价与碳强度加权代价驱动任务迁移决策，将电力系统的低碳信号引入算力调度，使训练类任务自发地向西部低碳低电价区域聚集，实现了算力负荷与新能源资源的空间匹配。

- **优点 4:** 问题 4 建立了任务调度与储能运行相统一的多区域协同框架，通过分层求解将上层离散调度决策与下层连续储能优化有机耦合，并通过多场景对照实验系统分析了碳约束、电价机制与新能源波动对策略的影响，模型可扩展性强。

7.2 模型的缺点

- **缺点 1:** 任务调度采用代价驱动的贪心策略，虽然计算效率高，但属于局部搜索方法，难以保证全局最优，在区域容量高度紧张时可能因贪心选择而损失整体最优性。

- **缺点 2:** 模型将储能与购售电决策简化为线性规划，未考虑储能寿命损耗、充放电循环成本以及电网侧的动态约束，且区域间未建模线路潮流，与实际电力系统的运行细节存在一定差距。

八、模型的改进与推广

8.1 模型的改进

本文的调度模型可进一步向全局优化方向改进。针对贪心调度易陷入局部最优的问题，可将任务调度层的代价驱动贪心替换为基于遗传算法或粒子群算法的全局搜索框架，以任务迁移矩阵与开工时段向量为编码个体，通过适应度评价驱动种群进化，从而在更大解空间内逼近全局最优；同时可将上层调度与下层储能优化进行迭代耦合，利用储能运行的对偶乘子构造调度代价中的边际价值信号，使任务调度真正感知储能的削峰填谷能力，实现算与储的深度协同。

针对储能模型未计及寿命损耗的不足，可在储能优化目标中引入充放电循环次数的惩罚项与 SOC 深度约束，将储能退化成本显式纳入优化模型，使储能策略在降低电费与保护电池寿命之间取得平衡；同时可引入滚动时域优化框架，以 24 小时为滚动窗口、以 SOC 末端值衔接窗口，提升模型对新能源与电价不确定性的适应能力。

8.2 模型的推广

本文建立的算电协同调度模型具有广泛的推广价值。在算力侧，该模型可直接推广到含 CPU、GPU、NPU 等多种异构算力资源的数据中心集群调度，以及考虑用户服务质量等级的多租户算力资源分配场景；在电力侧，可将单区域独立平衡推广到含线路潮流与网损的区域互联电网，并将储能扩展为电化学储能、抽水蓄能与氢储能的多类型混

合储能系统；在机制设计层面，可进一步研究峰谷电价动态定价、绿电交易与碳配额约束下的调度策略，为算电协同的市场化运行提供决策支持。此外，模型的任务类型与时延约束框架也可迁移至云边协同计算、工业物联网边缘调度等一般性任务调度问题。

参考文献

- [1] Friedman J H. Greedy function approximation: a gradient boosting machine[J]. The Annals of Statistics, 2001, 29(5): 1189-1232.
- [2] Ke G, Meng Q, Finley T, et al. LightGBM: a highly efficient gradient boosting decision tree[C]. Advances in Neural Information Processing Systems, Long Beach, 2017: 3146-3154.
- [3] 张程, 曹军威, 尹璐, 等. 数据中心能耗优化研究综述 [J]. 计算机学报, 2021, 44(6): 1085-1105.
- [4] 温鑫, 刘敦楠, 王玲湘, 等. 算力与电力协同优化调度研究综述 [J]. 电网技术, 2023, 47(10): 4070-4083.
- [5] 李彬, 陈宋宋, 祁兵, 等. 面向算力网络的分布式数据中心协同调度 [J]. 计算机研究与发展, 2022, 59(7): 1567-1580.
- [6] 刘洪, 闫大威, 王芊, 等. 考虑新能源消纳的园区综合能源系统优化调度 [J]. 中国电机工程学报, 2021, 41(14): 4800-4811.
- [7] Beloglazov A, Abawajy J, Buyya R. Energy-aware resource allocation heuristics for efficient management of data centers for cloud computing[J]. Future Generation Computer Systems, 2012, 28(5): 755-768.
- [8] 苏剑涛, 王锡凡, 王秀丽, 等. 考虑储能寿命约束的数据中心微电网优化调度 [J]. 电力系统自动化, 2021, 45(14): 120-128.
- [9] Deb K, Pratap A, Agarwal S, et al. A fast and elitist multiobjective genetic algorithm: NSGA-II[J]. IEEE Transactions on Evolutionary Computation, 2002, 6(2): 182-197.
- [10] Kohavi R. A study of cross-validation and bootstrap for accuracy estimation and model selection[C]. Proceedings of the 14th International Joint Conference on Artificial Intelligence, Montreal, 1995: 1137-1143.

附录

表 18 附件与求解文件说明

文件	说明
workload_trace.xlsx	50000 个 AI 任务轨迹数据
GPU_information.xlsx	六区域 GPU 容量与功率参数
network_latency.xlsx	区域间网络时延矩阵
region_time_data.xlsx	区域逐时电力与基准运行状态
power_mapping.xlsx	三类任务单位 GPU 功率映射
storage_information.xlsx	区域储能容量与充放电参数
问题一_GPU 需求统计预测与基础调度.py	问题一求解代码
问题二_碳感知任务调度.py	问题二求解代码
问题三_储能协同优化.py	问题三求解代码
问题四_多区域协同优化.py	问题四求解代码
预处理数据.csv	区域-类型-小时 GPU 需求序列
各问题结果 CSV	调度方案、指标对比等结果