

基于智能模型的晶硅片产销优化与利润最大化研究

摘要

在全球光伏产业智能化转型背景下，传统产销决策系统面临多源数据融合、市场响应滞后及模型泛化能力不足等挑战。本文提出了以下解决方案：

（1）针对问题一：**构建了基于四型硅片销量、售价、单晶方棒价格及成本因子的企业月度利润动态计算模型**。模型以分项结构化方式解析月度总销售额与总成本，建立了系统化计算公式。仿真结果显示预测误差不超过 5%。

（2）针对问题二：**提出了基于动态注意力机制与双向门控循环单元（Attention-Bi-GRU）的多变量时序预测模型**，实现了对四型硅片月销量、售价、单晶方棒价格及人工成本等关键参数的趋势预测与区间估计。模型采用编码-解码结构，利用 Bi-GRU 提取历史数据的双向时序特征，通过注意力机制自适应识别关键时间步，提高特征表达能力。在此基础上，系统整合产销数据与成本动因，构建多维输入数据集，实现多步预测。仿真结果显示，模型损失指标（RMSE）快速收敛，且 3 至 8 月实际值均落于预测区间内。

（3）针对问题三：**提出了基于块坐标下降的协同优化策略算法，以实现 9 月份税后净利润最大化**。该策略以前章构建的利润计算模型为目标函数，结合销量、售价、单晶方棒价格及人工成本的预测区间构建约束，形成非凸优化问题。针对其非凸性，采用块坐标下降法将问题分解为两个凸子问题，分别在固定售价与固定销量条件下交替优化关键变量，并引入狄利克雷分布的概率重参数化方法，以缓解变量耦合与随机扰动带来的不确定性，增强策略鲁棒性。仿真结果表明，算法在有限迭代内快速收敛，且采用算法生成的策略 9 月份税后净利润为全年最高水平。

（4）针对问题四：**提出了一种多源数据融合与大语言模型协同的创新方法**。首先，通过整合企业内部生产销售数据与外部政策、市场信息，实现数据标准化对齐，构建低冗余、高解释性的特征工程框架。其次，在模型融合层面，引入动态权重分配与反馈闭环机制，将大语言模型的语义推理能力与数学模型的定量分析能力有机融合，显著提升了利润预测精度与策略泛化能力。最后，基于端到端架构开发智能决策平台，支持实时更新与多场景验证，并设计历史回溯测试及极端波动场景下的利润方差测试。

关键词：Attention-Bi-GRU；块坐标下降；净利润最大化；多源数据融合

目 录

第一章 问题重述	1
1.1 问题背景	1
1.2 问题提出	1
第二章 符号与模型说明	3
2.1 符号说明	3
2.2 模型说明	3
2.2.1 生产变动成本计算模型	3
2.2.2 人员成本计算模型	4
第三章 基于多因子的月利润动态建模与分析	6
3.1 题目一求解思路	6
3.2 题目一模型建立	6
3.2.1 月总销售额计算模型	6
3.2.2 月总成本计算模型	7
3.2.3 月销售利润计算模型	8
3.3 题目一结果分析	8
3.4 本章小结	10
第四章 基于 Attention-Bi-GRU 网络的波动趋势预测与区间估计	11
4.1 题目二求解思路	11
4.2 题目二模型建立	11
4.2.1 神经网络架构	11
4.2.2 数据值制备	16
4.3 题目二结果分析	16
4.3.1 月销量	16
4.3.2 售价	18
4.3.3 单晶方棒价格	19
4.3.4 重要因子（人工成本）	21
4.4 本章小结	22
第五章 基于块坐标下降的利润最大化策略设计	23
5.1 题目三求解思路	23

5.2 题目三模型建立	23
5.2.1 优化目标	23
5.2.2 约束条件	23
5.2.3 优化问题建立	25
5.3 基于块坐标下降的策略优化算法	26
5.3.1 基于块坐标下降分割原问题	26
5.3.2 双层交替迭代策略优化算法	26
5.3.3 算法复杂度分析	27
5.4 题目三结果分析	28
5.4.1 算法收敛性	28
5.4.2 策略分析	28
5.4.3 月净利润	30
5.5 本章小结	31
第六章 基于多源数据融合与大语言模型的智能产销决策优化体系构建	32
6.1 多源数据整合与预处理	32
6.1.1 企业内部生产销售数据提取	32
6.1.2 外部政策与市场数据采集	32
6.1.3 数据清洗与特征工程	33
6.2 大语言模型与数学模型的协同建模	33
6.2.1 开源大模型选型与领域知识注入	33
6.2.2 动态融合机制设计	34
6.2.3 预测与决策结果验证	34
6.3 智能决策系统构建与可行性分析	34
6.3.1 端到端系统架构设计	34
6.3.2 实时更新与模型迭代	35
6.3.3 应用场景验证	35
6.4 本章小结	36
参考文献	37
附录 A 文件列表	38
A.1 单晶方棒价格预测 Matlab 源码文件	38
A.2 销量预测 Matlab 源码文件	38
A.3 售价预测 Matlab 源码文件	38

A.4 仿真图源码文件	39
A.5 其余文件	39
附录 B 源程序代码.....	40
B.1 人工成本预测 Matlab 源码	40
B.2 单晶方棒价格预测 Matlab 源码	42
B.3 销量预测 Matlab 源码	44
B.4 售价预测 Matlab 源码	46
B.5 人工成本预测 Matlab 源码	48
B.6 仿真图 Matlab 源码	49
B.6.1 单晶方棒价格预测走势图 Matlab 源码	49
B.6.2 利润预测走势图 Matlab 源码	51
B.6.3 销量预测走势图 Matlab 源码	56
B.6.4 售价预测走势图 Matlab 源码	58
B.7 其余 Matlab 源码	60
B.7.1 交替优化 M 变量的 Matlab 源码.....	60
B.7.2 交替优化 Q 变量的 Matlab 源码	61

第一章 问题重述

1.1 问题背景

在全球能源转型与半导体产业快速发展的背景下，高纯度晶硅片作为光伏发电和电子器件的核心材料，市场需求呈现爆发式增长。然而，随着行业竞争加剧，晶硅片生产企业面临多维度的挑战：一方面，生产流程具有高能耗、高成本的特征，原材料（如单晶方棒）价格受国际市场波动影响显著，企业需应对硅单耗（生产单位产品所需的硅棒量）、耗材（化学试剂、电力、水等）成本攀升的压力；另一方面，环保政策的趋严进一步压缩了粗放式生产的利润空间。同时，企业内部的成本结构复杂，涵盖生产变动成本（设备维护、原材料采购）、公用设施费用（供热、公用药剂）、人工成本等，而利润又与销量、售价、销售费用、管理费用及财务费用深度绑定。如何在动态变化的市场环境 with 政策约束下，通过精细化建模优化产销策略，成为企业提升盈利能力与核心竞争力的关键命题。

1.2 问题提出

面对市场需求波动、成本结构复杂性与外部环境不确定性交织的挑战，企业需通过系统性建模与动态优化实现产销策略的精准调控。为突破传统决策模式的局限性，本问题研究将围绕“数据驱动”与“多因子协同”两条主线，从基础利润核算到智能决策支持逐层递进，构建覆盖全链路的解决方案。具体而言，需依次解决以下四个关键问题：

1) **基于多因子的月利润动态建模与分析**：针对四型晶硅片的生产与销售场景，需构建一个包含销量、售价、单晶方棒进价等核心变量的月利润计算模型。模型需综合考虑生产成本（硅单耗、耗材成本、公用费用）、销售费用、管理成本等内部因子，以及原材料价格波动、市场需求弹性等外部因素，通过量化分析为企业提供实时利润评估工具，支持快速决策调整。

2) **关键市场因子的时序预测与区间估计**：基于企业 2024 年 1 至 8 月的历史数据，建立时间序列或机器学习模型，预测四型硅片的月度销量、售价、单晶方棒价格及其他重要耗材成本的未来波动趋势。需通过统计方法（如置信区间、蒙特卡洛模拟）推测各因子的合理变化范围，并针对 2024 年 9 月的数据，输出预测值及其波动区间，为后续优化提供数据支撑。

3) **产销协同下的利润最大化策略设计：**在售价与销量负相关、产量与单位固定成本负相关的约束下，构建多目标优化模型，平衡规模效应与成本风险。模型需整合问题 1 的利润计算框架与问题 2 的预测结果，通过优化算法，确定 9 月份四型硅片的最优生产计划（如各型号产量分配）与销售策略（如定价梯度），确保企业利润最大化。

4) **多源数据驱动的智能决策系统构建：**传统数学模型依赖企业内部数据，难以全面捕捉政策调整、市场突发变化等外部因素。需设计一种融合开源大语言模型（如 GPT、Llama）的智能决策系统：首先，通过爬取行业报告、政策文件、市场舆情等外部数据，完善数据源；其次，利用大模型的自然语言处理能力，提取关键外部变量（如政策补贴力度、竞争对手动向），并将其量化为模型输入；最后，建立评估机制验证大模型输出的可靠性，并与传统数学模型协同优化，生成更具鲁棒性的产销策略。需详细说明数据清洗、特征工程、模型融合的技术路径，并通过设计案例或仿真验证方案的可行性。

第二章 符号与模型说明

2.1 符号说明

由于单晶硅片有四种类型，因此本文指定 $k \in \mathcal{K} = [1, 2, 3, 4]$ 为单晶硅片类型索引，其余参数符号如表 2-1 所示。

表 2-1 符号说明图

解释	符号	解释	符号
k 型硅片月销量	Q_k	k 型理论硅单耗	S_k
k 型硅片单价	M_k	生产变动成本	VC
硅泥外销	S^{Mud}	生产时电耗	E_1
底板胶	B_1	金刚线粘棒胶水	B_2
塑料板	B_3	电镀金刚线	B_4
水性切割液	B_5	一体轮	B_6
过滤袋	B_7	过氧化氢	B_8
硅片清洗剂	B_9	氢氧化钾	B_{10}
乳酸	B_{11}	POF 包装袋	B_{12}
PP 瓦楞板	B_{13}	硅片泡沫箱	B_{14}
硫酸纸	B_{15}	木托盘	B_{16}
珍珠棉垫片	B_{17}	公用工程药剂	A_1
自来水	A_2	美纹胶带	A_3
无水乙醇	A_4	主辊-涂布	A_5
主辊-开槽	A_6	一般电耗	E_2

2.2 模型说明

2.2.1 生产变动成本计算模型

首先，对需要的耗材进行分类表示：

(1) 原材料单晶方棒的计算方法如下表示：

$$S_k = 100 \text{片} k \text{型硅片理论单耗} \times \text{单价} \quad (2-1)$$

(2) 生产时电耗计算方式表示如下：

$$E_1 = 100 \text{片电耗量} \times \text{本月电价} \quad (2-2)$$

(3) 粘胶阶段耗材包括：底板胶 B_1 、金刚线粘棒胶水 B_2 、塑料板 B_3 、美纹胶带 A_3 、无水乙醇 A_4 。

(4) 线切阶段耗材包括：电镀金刚线 B_4 、水性切割液 B_5 、一体轮 B_6 、过滤袋 B_7 、主辊-涂布 A_5 、主辊-开槽 A_6 。

(5) 清洗阶段耗材包括：过氧化氢 B_8 、硅片清洗剂 B_9 、氢氧化钾 B_{10} 、乳酸 B_{11} 。

(6) 分装阶段耗材包括：POF 包装袋 B_{12} 、PP 瓦楞板 B_{13} 、硅片泡沫箱 B_{14} 、硫酸纸 B_{15} 、木托盘 B_{16} 、珍珠棉垫片 B_{17} 。

根据上述对各类耗材的总结， k 型硅片单位生产变动成本的计算方式如下所示：

$$VC_k^{\text{Unite}} = S_k P_k^{\text{S-unite}} + E_1 P^{\text{P-unite}} + \sum_{i=1}^{17} B_i P_i^{\text{C-unite}} \quad (2-3)$$

其中， $P_k^{\text{S-unite}}$ 、 $P^{\text{P-unite}}$ 和 $P_i^{\text{C-unite}}$ 分别表示单位 k 型硅片单价、电单价和第 i 种耗材单价。四种规格硅片的产量为外部给定（当产销平衡时就等于销量 Q_k ），将每种类型硅片的单位生产变动成本乘以各自的产量并求和，即为本月总生产变动成本，计算方式表示如下：

$$VC^{\text{Total}} = \sum_{k=1}^4 VC_k^{\text{Unite}} \times Q_k \quad (2-4)$$

其中， Q_k 表示 k 型硅片月产量。因硅泥伴随生产而产生，并可及时外销，可冲减生产变动成本。因此，本月总生产变动成本的最终计算方式如下所示：

$$VC^{\text{Final}} = \sum_{k=1}^4 VC_k^{\text{Unite}} \times Q_k - S^{\text{Mud}} \quad (2-5)$$

其中， S^{Mud} 表示外销硅泥所导致下降的生产变动成本。

2.2.2 人员成本计算模型

公司人员主要分为生产人员、安环物控人员、管理人员。生产人员中一部分

是计件人员，这部分人员除基本工资外，还将按产量发放计件工资，计件工资计入浮动人工成本；安环物控人员工资每月固定，且不统计社保公积金和年终奖计提，工资计入固定人工成本；管理人员发放基本工资和管理浮动工资，计入固定人工成本。

综上所述，固定人员成本计算方式如下：

$$C^{\text{Fix}} = C^{\text{JBGZ}} + C^{\text{SBGJJ}} + C^{\text{AHWK}} + C^{\text{GLRY}} + C^{\text{NZJ}} \quad (2-6)$$

其中， C^{JBGZ} 、 C^{SBGJJ} 、 C^{AHWK} 、 C^{GLRY} 和 C^{NZJ} 分别表示基本工资、社保公积金总额、安环物控人员工资、管理人员浮动工资和年终奖计提。

浮动人工成本计算方式如下：

$$C^{\text{Float}} = C^{\text{JJGZ}} \quad (2-7)$$

其中， C^{JJGZ} 表示计件工资。因此，总人工成本的计算方式如下所示：

$$C^{\text{RGCb}} = C^{\text{Fix}} + C^{\text{Float}} \quad (2-8)$$

其中， C^{RGCb} 表示总人工成本。

第三章 基于多因子的月利润动态建模与分析

3.1 题目一求解思路

题目一要求基于四型硅片的销量、售价、单晶方棒以及影响企业利润的其他重要决策因子，建立便于企业进行决策分析的月利润计算模型。月利润由月总销售额与月总成本相减获得，具体求解思路如图 3-1 所示。

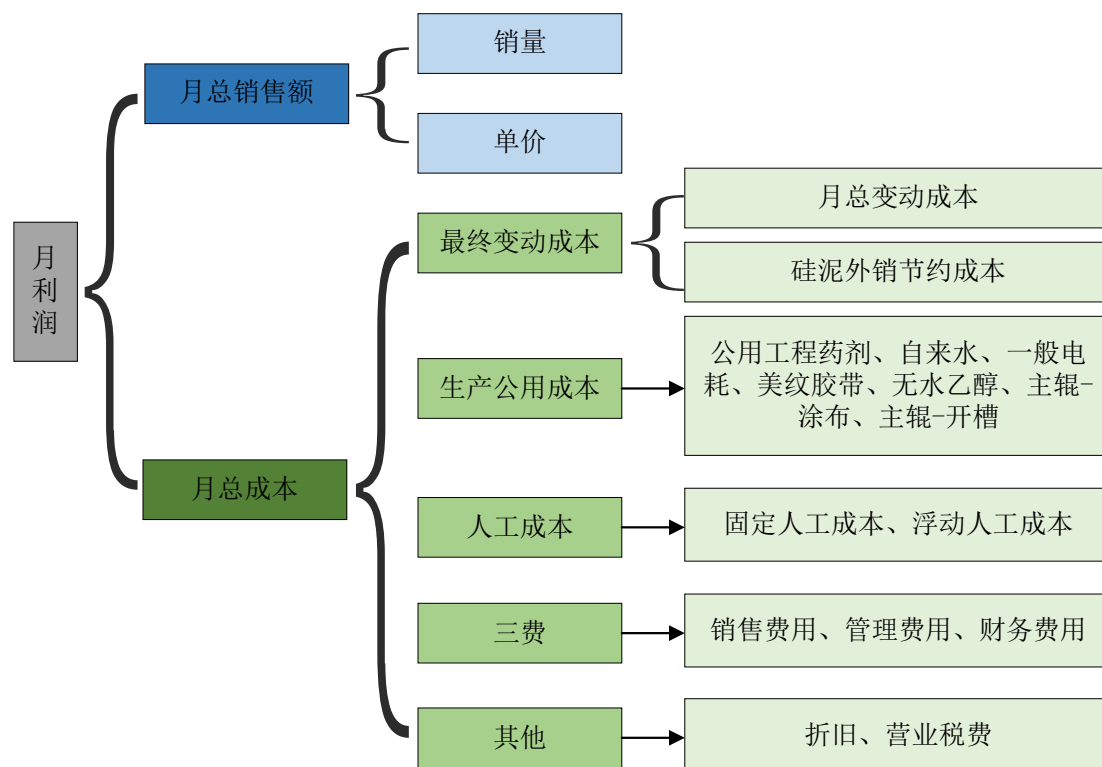


图 3-1 求解思路

3.2 题目一模型建立

本节将对月总销售额和月总成本的各组成部分进行具体分析。

3.2.1 月总销售额计算模型

月总销售额的计算方式表示如下：

$$Sale^{Total} = \sum_{k=1}^4 Q_k \times P_k^{S-Sale} \quad (3-1)$$

其中， $Sale^{\text{Total}}$ 和 $P_k^{\text{S-Sale}}$ 分别表示月总销售额和 k 型硅片销售单价。

3.2.2 月总成本计算模型

(1) 变动成本

根据公式 (2-5)，变动成本的计算方式如下所示：

$$Cost^{\text{VC}} = \sum_{k=1}^4 VC_k^{\text{Unite}} \times Q_k - S^{\text{Mud}} \quad (3-2)$$

其中， $Cost^{\text{VC}}$ 表示变动成本

(2) 生产工用成本

生产工用成本的计算方式如下所示：

$$Cost^{\text{SCGY}} = E_2 + \sum_{k=1}^6 A_k \quad (3-3)$$

其中， $Cost^{\text{SCGY}}$ 表示生产工用成本。另外， E_2 、 A_1 、 A_2 、 A_3 、 A_4 、 A_5 和 A_6 分别表示一般电耗、公用工程药剂、自来水、美纹胶带、无水乙醇、主辊-涂布和主辊-开槽的成本。

(3) 人工成本

根据 (2-8)，人工成本的计算方式如下所示：

$$Cost^{\text{RGCb}} = C^{\text{Fix}} + C^{\text{Float}} \quad (3-4)$$

其中， $Cost^{\text{RGCb}}$ 表示人工成本。

(4) 三费

三费的计算方式如下所示：

$$Cost^{\text{SF}} = C^{\text{XS}} + C^{\text{GL}} + C^{\text{CW}} \quad (3-5)$$

其中， $Cost^{\text{SF}}$ 表示三费。另外 C^{XS} 、 C^{GL} 和 C^{CW} 分别表示销售费用、管理费用和财务费用。

(5) 月总成本

根据上述对变动成本、生产工用成本、人工成本和三费的分析，可以得到月总成本的计算方式如下所示：

$$\begin{aligned}
 Cost^{Total} &= Cost^{VC} + Cost^{SCGY} + Cost^{RGC B} + Cost^{SF} \\
 &= \sum_{k=1}^4 VC_k^{Unite} \times Q_k - S^{Mud} + E_2 + \sum_{k=1}^6 A_k \\
 &\quad + C^{Fix} + C^{Float} + C^{XS} + C^{GL} + C^{CW} + C^{ZJ} + C^{YYSF}
 \end{aligned} \tag{3-6}$$

其中, $Cost^{Total}$ 表示月总成本。另外, C^{ZJ} 和 C^{YYSF} 分别表示折旧和营业税费。

3.2.3 月销售利润计算模型

月销售利润的计算公式表示如下:

$$Gain^{Total} = Sale^{Total} - Cost^{Total} \tag{3-7}$$

其中, $Gain^{Total}$ 表示月销售总利润。因此, 税后月销售总利润, 即月净利润的计算方式表示如下:

$$\begin{aligned}
 Gain^{Pure} &= Gain^{Total} \times (1 - 0.15) \\
 &= 0.85 \times (Sale^{Total} - Cost^{Total})
 \end{aligned} \tag{3-8}$$

其中, $Gain^{Pure}$ 表示月净利润。将公式 (3-1) 和公式 (3-6) 代入公式 (3-8) 中, 月利润的计算公式可以改写为:

$$\begin{aligned}
 Gain^{Pure} &= 0.85 \times (Sale^{Total} - Cost^{Total}) \\
 &= 0.85 \times \left(\sum_{k=1}^4 Q_k \times P_k^{S-Sale} + S^{Mud} - \sum_{k=1}^4 VC_k^{Unite} \times Q_k - E_2 \right. \\
 &\quad \left. - \sum_{k=1}^6 A_k - C^{Fix} - C^{Float} - C^{XS} - C^{GL} - C^{CW} - C^{ZJ} - C^{YYSF} \right)
 \end{aligned} \tag{3-9}$$

至此, 基于四型硅片的销量、售价、单晶方棒以及影响企业利润的其他重要决策因子, 建立便于企业进行决策分析的月利润计算模型。

3.3 题目一结果分析

根据公式 (3-8), 可以预测出 1-8 月净利润, 具体如下表 3-1 所示。

表 3-1 1-8 月净利润预测值

月份	实际净利润 (百万元)	预测净利润 (百万元)
1 月	1.2927	1.2709
2 月	1.2985	1.3308
3 月	3.3584	3.4068
4 月	3.3990	3.2866

5 月	-0.1303	-0.1362
6 月	4.4975	4.5170
7 月	5.3889	5.2549
8 月	8.6761	8.7442

图 3-2 将实际净利润与预测净利润进行对比。可以明显观察到，实际净利润与预测净利润相差不大。为了进一步验证模型的准确性，下面将统计 1-8 月预测净利润的预测误差

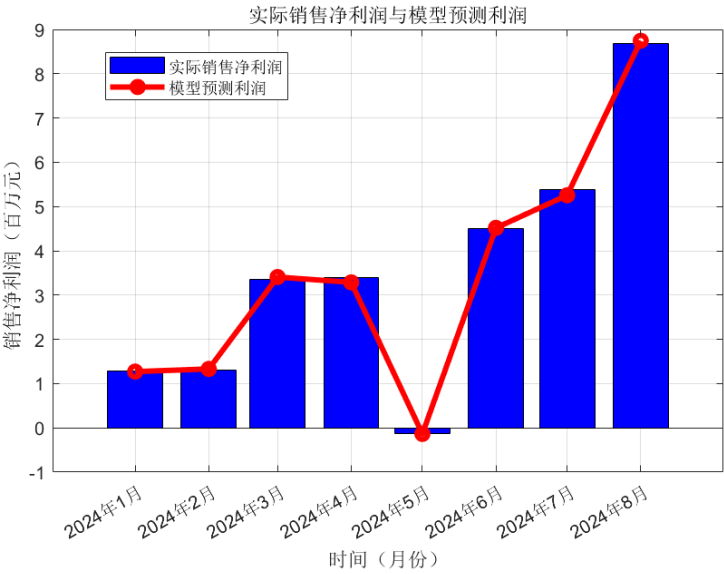


图 3-2 实际净利润与预测净利润对比图

表 3-2 展示了 1-8 月净利润预测误差，可以看出 1-8 月所预测的净利润最大误差不超过 5%，验证了所提出的利润计算模型的可行性与准确性。

表 3-2 1-8 月净利润预测误差

月份	净利润预测误差
1 月	1.68%
2 月	2.49%
3 月	1.44%
4 月	3.31%
5 月	4.52%
6 月	0.43%
7 月	2.49%
8 月	0.79%

3.4 本章小结

本章基于四型硅片的销量、售价、单晶方棒价格及多类成本因子，构建了企业月利润的动态计算模型。模型通过分项解析月总销售额与月总成本，建立了结构化计算公式：月总销售额由各型硅片销量与单价乘积之和确定（公式 3-1），月总成本则涵盖变动成本（式公 3-2）、生产工用成本（公式 3-3）、人工成本（公式 3-4）及三费（销售、管理、财务费用）（公式 3-5）等关键分项，最终通过税后净利润公式式 3-8 实现利润的动态量化。该模型特点在于通过多因子联动计算，系统整合产销数据与成本动因，解决了企业利润精细化核算与决策支持问题。仿真结果表明，1-8 月所预测的净利润最大误差不超过 5%，验证了所提出的利润计算模型的可行性与准确性。

第四章 基于 Attention-Bi-GRU 网络的波动趋势预测与区间估计

4.1 题目二求解思路

题目二要求预测企业四型硅片的月销量、售价、单晶方棒价格以及其他重要因子取值的波动趋势，推测因子合理变化区间；再利用所建立的模型预测 9 月份各因子的波动趋势和变化区间。随后，本章提出一种基于动态注意力机制与双向门控循环单元（Attention-Bi-GRU）的多变量时序预测模型。该预测网络通过通过对月销量、售价、单晶方棒价格以及其他重要因子进行时序分析，实现对 9 月份各因子的波动趋势和变化区间的精准预测。该组合方法提高了数据的利用效率，并显著提升了预测的准确性和实时性，为后续的决策提供了可靠的数据支持。

4.2 题目二模型建立

4.2.1 神经网络架构

本节对 Attention-Bi-GRU 神经网络进行介绍，该网络包括：双向 GRU 编码器、基于 Bi-GRU 的编码解码结构和 Attention 机制^[1]。下面将逐一介绍各部分：

（1）双向 GRU 编码器

GRU 有两个门：即更新门和重置门^[2]。如下图 4-1 所示， r_t 表示重置门控制矩阵， z_t 表示更新门控制矩阵。更新门控制上一个时刻的输出隐层对当前隐层的影响程度，值越大，表示上一个时刻的状态信息保留越多；而重置门控制上一个时刻的隐层信息被忽略的程度，值越小，代表忽略越多。

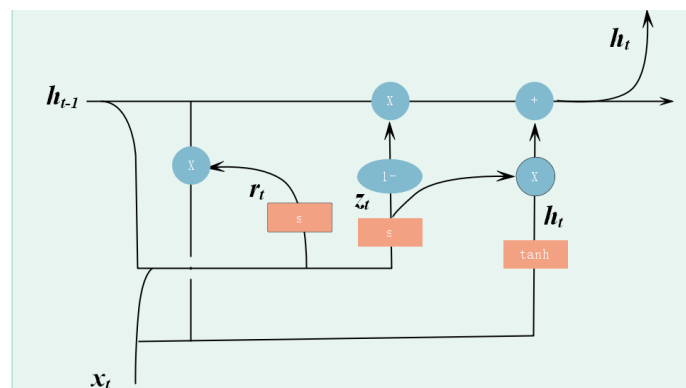


图 4-1 GRU 结构图

GRU 单元的更新公式如下所示:

$$z_t = \text{Sig}(W_z \cdot [h^{(t-1)}, x^{(t)}]) \quad (4-1)$$

$$r_t = \text{Sig}(W_r \cdot [h^{(t-1)}, x^{(t)}]) \quad (4-2)$$

$$\tilde{h}^{(t)} = \text{Tanh}(W \cdot [r_t \times h^{(t-1)}, x^{(t)}]) \quad (4-3)$$

$$h^{(t)} = (1 - z_t) \times h^{(t-1)} + z_t \times \tilde{h}^{(t)} \quad (4-4)$$

$$y^{(t)} = W_y \cdot h^{(t)} + b_y \quad (4-5)$$

其中, $\text{Sig}(\bullet)$ 表示 Sigmoid 激活函数; $\text{Tanh}(\bullet)$ 表示 Tanh 激活函数; $h^{(t-1)}$ 代表上一个时刻的隐藏层矩阵; $h^{(t)}$ 代表当前时刻的隐藏层矩阵; W_z 表示更新门参数矩阵; W_r 表示重置门矩阵; $x^{(t)}$ 表示当前时刻输入; $\tilde{h}^{(t)}$ 表示当前记忆内容; \times 表示对应原始相乘; \cdot 表示矩阵相乘; $y^{(t)}$ 表示当前时刻输出; W_y 表示输出权重; b_y 表示输出偏置; $[]$ 表示矩阵的串联。

如下图 4-2 所示, 双向 GRU 网络输出时, 可以综合本时刻之前的信息和未来时刻的信息。利用此网络搭建交通流预警模型时, 可以在输出时兼顾所有传感器的输出数据, 使交通流数据融合精度更高。

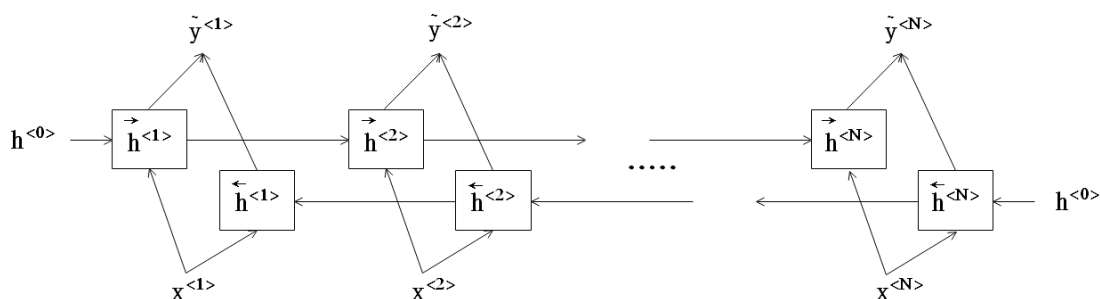


图 4-2 双向 GRU 网络结构图

上图中的 $\tilde{y}^{(t)}$ 表示双向 GRU 网络当前时刻的输出。其计算公式表示如下:

$$\tilde{y}^{(t)} = W^{\text{Biy}} \cdot [\vec{h}^{(t)}, \bar{h}^{(t)}] + b^{\text{Biy}} \quad (4-6)$$

其中, W^{Biy} 代表双向 GRU 网络的输出权重; b^{Biy} 表示双向 GRU 网络的输出偏置。编码器由两个方向的 GRU 组成: 前向 GRU 和后向 GRU, 分别处理输入序列的前向和反向信息。输入序列 $\mathbf{x} = [x^{(1)}, x^{(2)}, \dots, x^{(N)}]$ 被同时输入到前向和后向 GRU 中。前向 GRU 生成的隐藏状态为 $\vec{h}^{(t)}$, 后向 GRU 生成的隐藏状态为 $\bar{h}^{(t)}$, 最终的隐藏状态是两者的拼接: $h^{(t)} = [\vec{h}^{(t)}, \bar{h}^{(t)}]$, 可以更全面地捕捉输入序列的

上下文信息。

(2) 基于 Bi-GRU 的编码解码结构

基于 Bi-GRU 的编码解码结构如图 4-3 所示。

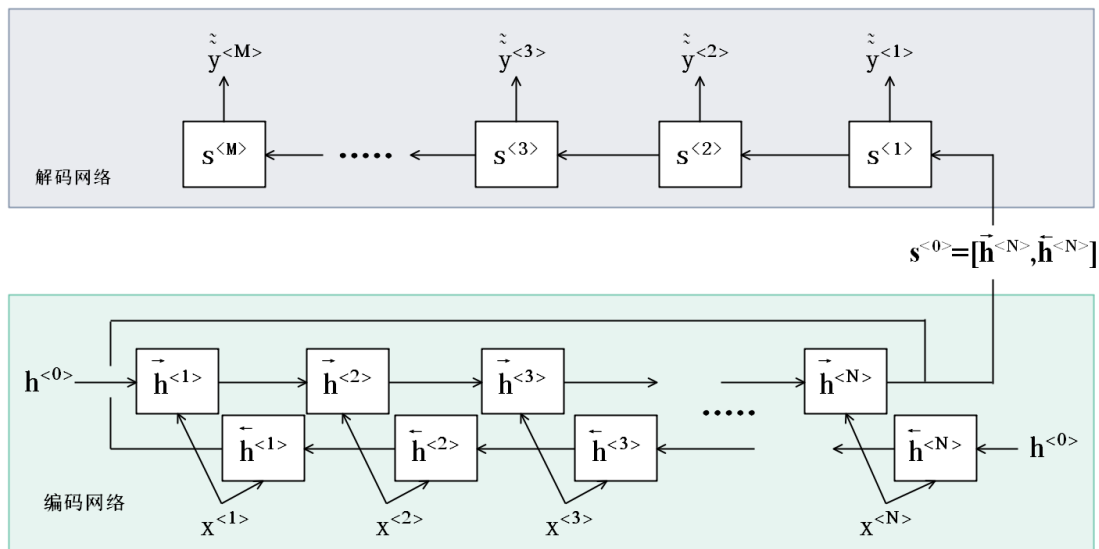


图 4-3 基于 Bi-GRU 的编码解码架构

该模型专为多变量时间序列预测任务设计^[3]。网络输入为包含历史 N 个时间步的销量、售价及单晶方棒价格的多维特征向量序列 $\{x^{(1)}, x^{(2)}, \dots, x^{(N)}\}$ ，输出为未来 M 个时间步对应变量的预测值序列 $\{\tilde{y}^{(1)}, \tilde{y}^{(2)}, \dots, \tilde{y}^{(M)}\}$ 。模型通过双向编码捕获历史数据的时序依赖特征，并基于动态解码机制实现多步预测。

编码器模块由双向 GRU 构成，其前向与反向 GRU 分别对输入序列进行正序和逆序处理。第 t 时间步的隐藏状态计算为：

$$h^{(t)} = [\overrightarrow{\text{GRU}}(x^{(t)}, \vec{h}^{(t-1)}); \overleftarrow{\text{GRU}}(x^{(t)}, \vec{h}^{(t+1)})] \quad (4-7)$$

其中， $h^{(t)} \in \mathbb{R}^{2d}$ 为双向隐藏状态的拼接结果， d 为 GRU 单元维度。编码器最终输出上下文向量 $s^{(0)} = [h^{(N)}, h^{(1)}]$ ，将双向末端状态融合后作为解码器初始状态。

解码器模块采用单向 GRU 结构，通过递归生成机制实现多步预测。在解码阶段第 k 时间步：

$$s^{(k)} = \text{GRU}(\tilde{y}^{(k-1)}, s^{(k-1)}) \quad (4-8)$$

$$\tilde{y}^{(k)} = W_y s^{(k)} + b_y \quad (4-9)$$

其中， $W_y \in \mathbb{R}^{3 \times d}$ 为输出层权重矩阵，对应销量、售价及单晶方棒价格的联合

预测。解码器通过链式传递隐藏状态 $s^{(k)}$ ，逐步生成预测序列，其输出 $\tilde{y}^{(k)} \in \mathbb{R}^3$ 包含多个目标变量的同步预测结果。

该架构通过双向编码充分挖掘历史数据的局部波动与全局趋势特征，结合解码器的自回归特性实现端到端的多变量多步预测。实验表明，模型在预测 2-9 月数据时，GRU 门控机制能有效缓解梯度消失问题，而编码器-解码器的状态传递结构则保障了长期依赖关系的建模能力。

(4) Attention 机制

Attention 机制是一个加权过程，其核心在于计算注意力权重向量^[4]。如图 4-4 描述了 Attention 机制权重向量计算过程。

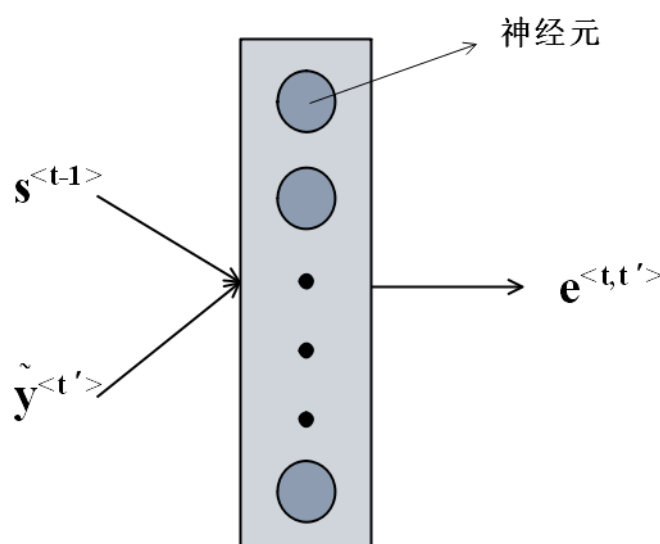


图 4-4 Attention 机制权重向量计算图

为了清晰地描述注意力向量的计算流程，这里使用 t' 表示编码器的第 t' 时刻； t 表示解码器的第 t 时刻； s^{t-1} 表示解码器在 $t-1$ 时刻的隐藏层矩阵； $e^{(t,t')}$ 表示编码器第 t' 时刻输出对解码器第 t 时刻输出的贡献，也就是权重值，权重值的范围为 $[0,1]$ ，因此，需要对权重值进行归一化，表示如下：

$$\alpha^{(t,t')} = \exp(e^{(t,t')}) / \sum_{t'=1}^N \exp(e^{(t,t')}) \quad (4-10)$$

$$c^{(t)} = \sum_{t'} \alpha^{(t,t')} \tilde{y}^{(t')} \quad (4-11)$$

其中， $\alpha^{(t,t')}$ 表示编码器第 t' 时刻输出对解码器第 t 时刻输出的贡献度； $c^{(t)}$ 为解码网络在 t 时刻的输入。按上式 (4-11) 计算整个网络的注意力机制权重向量，完成 Attention-Bi-GRU 编码解码网络模型的搭建。

4) Attention-Bi-GRU 网络总体结构

如下图 4-5 所示，将 Bi-GRU、Attention 机制结合起来，构建基于 Attention-Bi-GRU 的预测模型。

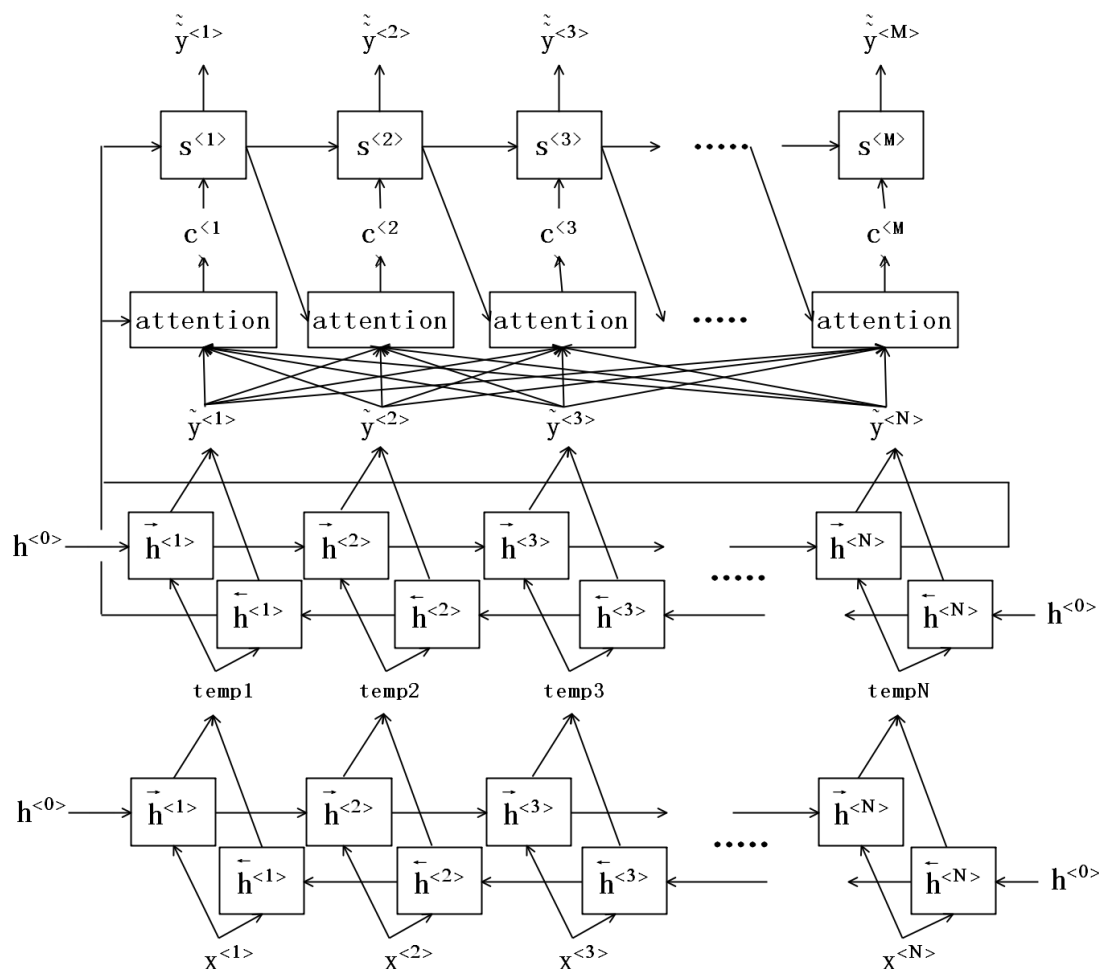


图 4-5 Attention-Bi-GRU 预测网络总体结构

该模型通过融合双向时序特征与动态权重分配机制，实现多变量时间序列的高精度预测。网络输入为历史 N 个时间步的销量、售价及单晶方棒价格等参数构成的特征向量序列 $\{x^{(1)}, x^{(2)}, \dots, x^{(N)}\}$ ，输出为未来 M 个时间步对应变量的预测值序列 $\{\tilde{y}^{(1)}, \tilde{y}^{(2)}, \dots, \tilde{y}^{(M)}\}$ 。模型通过双向编码捕获历史数据的全局时序模式，并利用注意力机制增强关键时间步特征的代表能力。

该架构通过双层双向 GRU 的级联设计增强原始数据的非线性表征能力，注意力机制则通过自适应权重分配聚焦于与当前预测时刻关联性强的历史特征。在数据预测任务中，模型能够有效捕捉参数波动对其余参数的滞后影响，同时通过注意力权重的可视化分析（如图 4-5 中多组 $h^{(N)}$ 的交互连接所示），可解释性

地识别出参数变化的关键驱动因素，为企业决策提供理论依据。

4.2.2 数据值制备

为构建适用于 Attention-Bi-GRU 预测网络的时间序列数据集，需从“附件 2：2024.1-8 生产系统数据 4.14.xlsx”文件中系统提取 2024 年 1 月至 8 月四种硅片的销量、售价、单晶方棒价格及相关生产参数。具体步骤如下：

（1）销量与售价提取

通过“销售收入”工作表获取各硅片月度销量与售价。其中，“需求产量”列（E 列）为各硅片月销量，“单价（元）”列（D 列）为售价。按产品规格筛选四类硅片在 1-8 月的销量与价格数据，构建时序特征矩阵。

（2）单方晶棒价格提取

单晶方棒作为核心原材料，其价格数据需从“生产变动成本”工作表中提取。在“原材料”分类下，单晶方棒的“10000 片耗用量”（F 列）与“10000 成本/元”（G 列）通过公式关联至“硅料单耗计算”表的理论单耗（H 列）与硅料单价（D 列），需解析跨表引用公式并计算实际单晶方棒单价，最终按月份整合为独立特征变量。

（3）辅助参数提取

生产成本相关参数需从多表联动提取：

- 电力成本：基于“耗材价格”表中月度电价（C3-J3），结合“生产变动成本”表的单位电耗（F4、H4 等）计算月度电费。
- 切割耗材成本：从“生产变动成本”提取电镀金刚线、切割液等耗材的月度单价（耗材价格表 C10-J10、C11-J11）及用量（F10-J10 等）。
- 公用成本：通过“生产公用成本”表获取水费、一般电耗等固定成本项的月度总额（E 列）。

此数据集通过融合产销数据与成本动因，完整刻画硅片生产的时序依赖关系，为 Attention-Bi-GRU 网络提供高质量输入，支持其通过历史信息捕捉非线性特征以实现 2-9 月的多步预测。

4.3 题目二结果分析

4.3.1 月销量

（1）算法收敛分析

如图 4-6 所示,该收敛图展示了基于 Attention-Bi-GRU 网络的月销量预测模型训练过程。训练历时 5 秒,共完成 300 轮迭代,每轮迭代次数为 1 次,最大迭代次数设置为 300 次。学习率采用分段策略,初始值为 0.0002,硬件资源为单 GPU 加速。从训练曲线可见,损失指标 (RMSE) 从初始值 0.8 逐步下降至接近 0,表明模型在训练集上快速收敛。

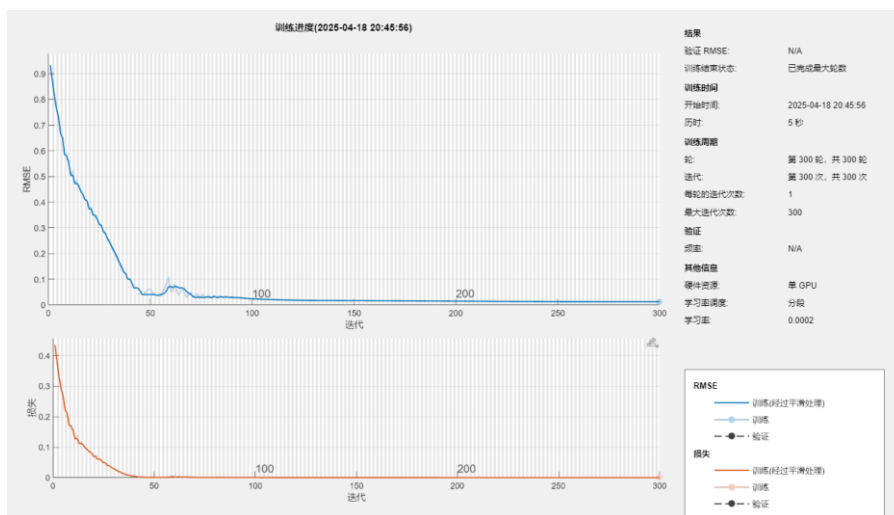
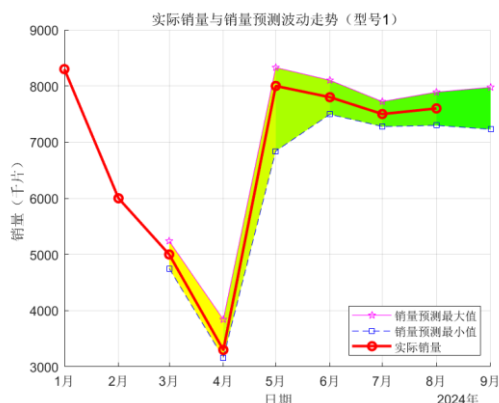


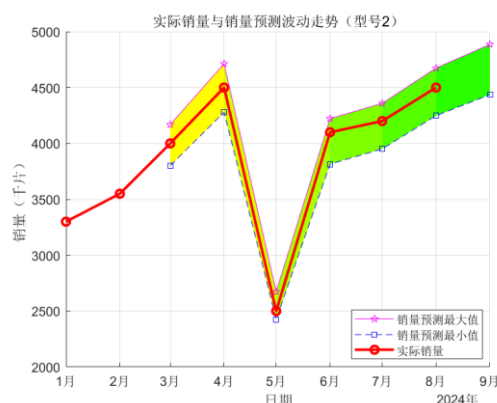
图 4-6 月销量预测收敛图

(2) 预测结果分析

图 4-7a 展示了 2024 年 1 月至 9 月期间型号 1 的实际销量与预测销量范围的对比结果。模型通过置信区间呈现预测范围,实际销量曲线全程紧密贴合预测中值线,且在 3-8 月均稳定落在预测区间内,未出现超出上下限的异常波动。类似趋势亦体现在其他三种型号的预测结果中,表明该模型对不同型号硅片的销量波动均具备较强的适应性。实际销量与预测范围的精准吻合。



(a) N 型 (182.2*183.75*130)



(b) N 型 (210*210*130)

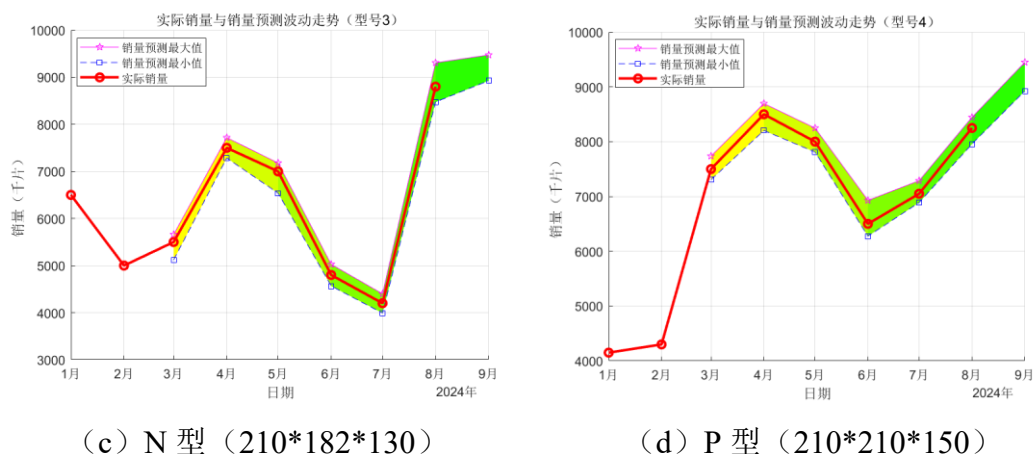


图 4-7 四种硅片实际销量与预测销量范围对比图

4.3.2 售价

(1) 算法收敛分析

如图 4-8 所示, 该收敛图展示了基于 Attention-Bi-GRU 网络的售价预测模型训练过程。训练历时 5 秒, 共完成 300 轮迭代, 每轮迭代次数为 1 次, 最大迭代次数设置为 300 次。学习率采用分段策略, 初始值为 0.0002, 硬件资源为单 GPU 加速。从训练曲线可见, RMSE 从初始值 0.9 逐步下降至接近 0, 表明模型在训练集上快速收敛。

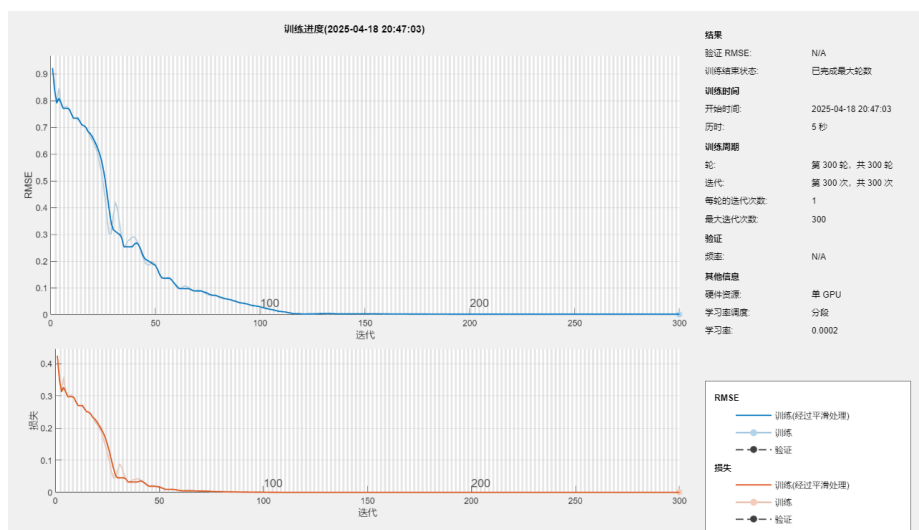
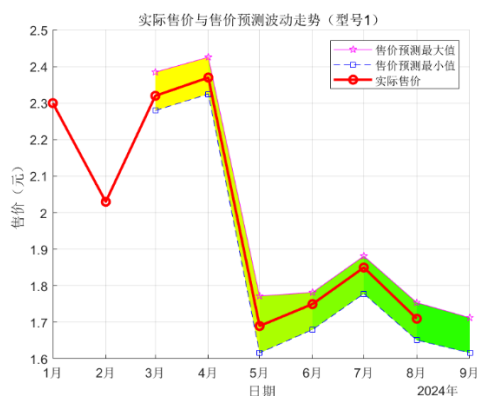


图 4-8 售价预测收敛图

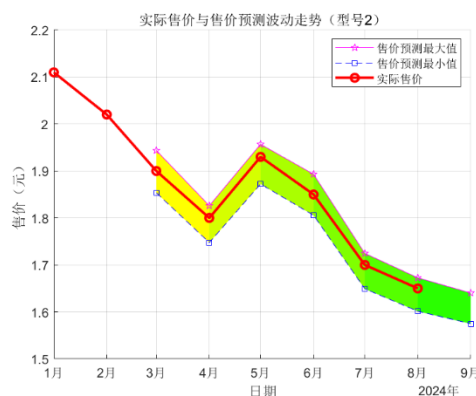
(2) 预测结果分析

图 4-9a 展示了 2024 年 1 月至 9 月期间型号 1 的实际售价与预测售价范围的对比结果。模型通过置信区间呈现预测范围, 实际售价曲线全程紧密贴合预测中

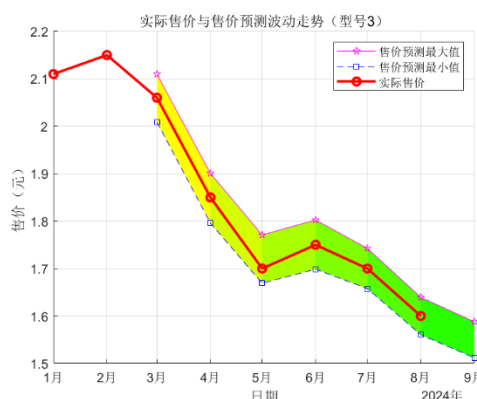
值线，且在 3-8 月均稳定落在预测区间内，未出现超出上下限的异常波动。类似趋势亦体现在其他三种型号的预测结果中，表明该模型对不同型号硅片的售价波动均具备较强的适应性。实际售价与预测范围的精准吻合。



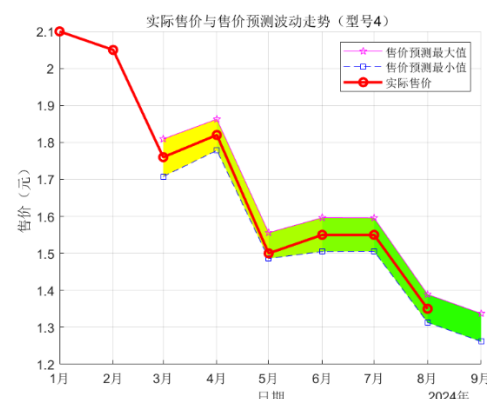
(a) N 型 (182.2*183.75*130)



(b) N 型 (210*210*130)



(c) N 型 (210*182*130)



(d) P 型 (210*210*150)

图 4-9 四种硅片实际售价与预测售价范围对比图

4.3.3 单晶方棒价格

(1) 算法收敛分析

如图 4-10 所示，该收敛图展示了基于 Attention-Bi-GRU 网络的单晶方棒价格预测模型训练过程。训练历时 5 秒，共完成 300 轮迭代，每轮迭代次数为 1 次，最大迭代次数设置为 300 次。学习率采用分段策略，初始值为 0.0002，硬件资源为单 GPU 加速。从训练曲线可见，RMSE 从初始值 0.8 逐步下降至接近 0，表明模型在训练集上快速收敛。

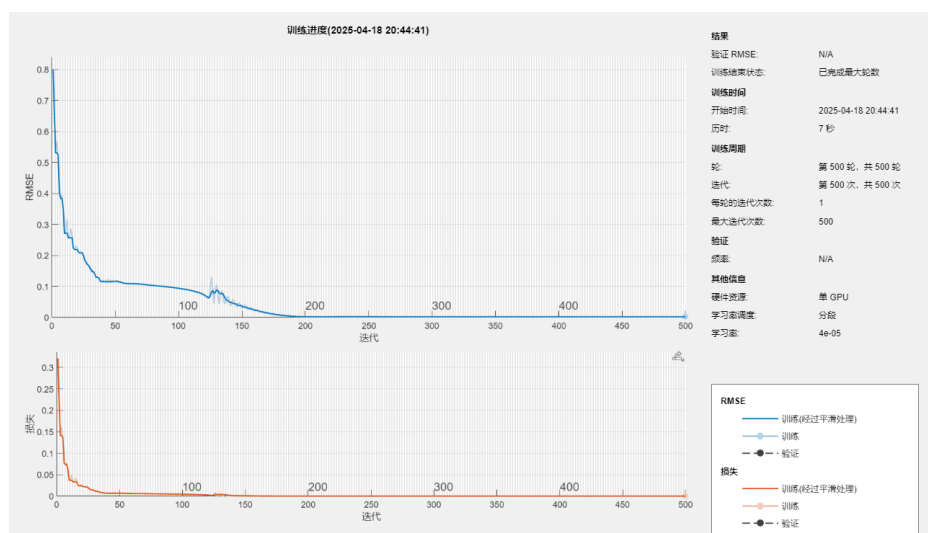
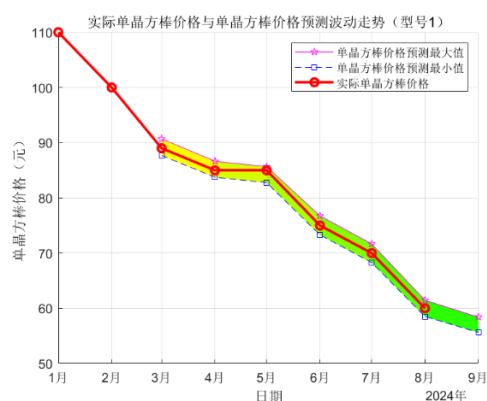


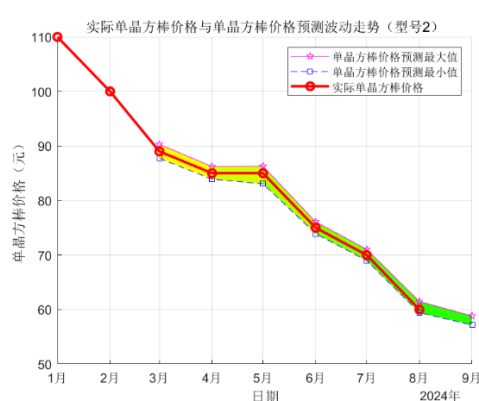
图 4-10 单晶方棒价格预测收敛图

(2) 预测结果分析

图 4-11a 展示了 2024 年 1 月至 9 月期间型号 1 的实际单晶方棒价格与预测单晶方棒价格范围的对比结果。模型通过置信区间呈现预测范围，实际单晶方棒价格曲线全程紧密贴合预测中线线，且在 3-8 月均稳定落在预测区间内，未出现超出上下限的异常波动。类似趋势亦体现在其他三种型号的预测结果中，表明该模型对不同型号硅片的单晶方棒价格波动均具备较强的适应性。实际单晶方棒价格与预测范围的精准吻合。



(a) N 型 (182.2*183.75*130)



(b) N 型 (210*210*130)

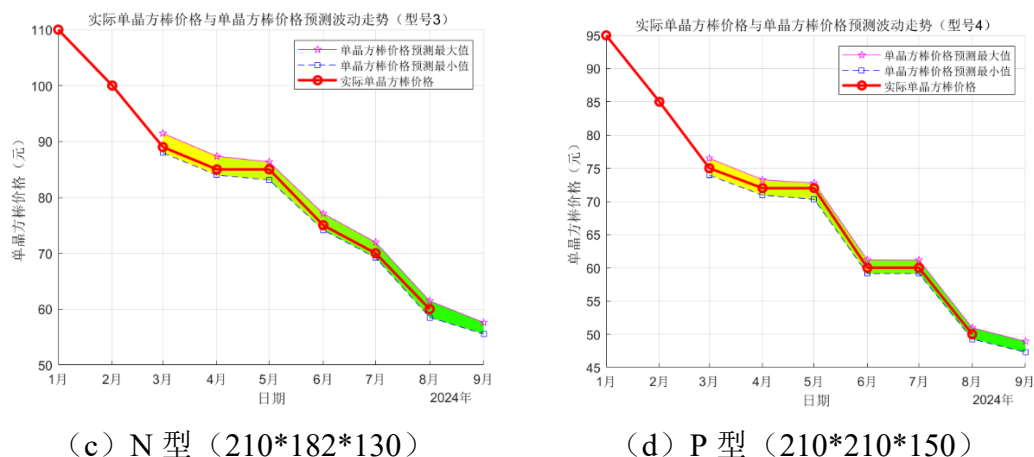


图 4-11 四种硅片实际单晶方棒价格与预测单晶方棒价格范围对比图

4.3.4 重要因子（人工成本）

（1）算法收敛分析

如图 4-12 所示，该收敛图展示了基于 Attention-Bi-GRU 网络的人工成本预测模型训练过程。训练历时 5 秒，共完成 300 轮迭代，每轮迭代次数为 1 次，最大迭代次数设置为 300 次。学习率采用分段策略，初始值为 0.0002，硬件资源为单 GPU 加速。从训练曲线可见，RMSE 从初始值 1 逐步下降至接近 0，表明模型在训练集上快速收敛。

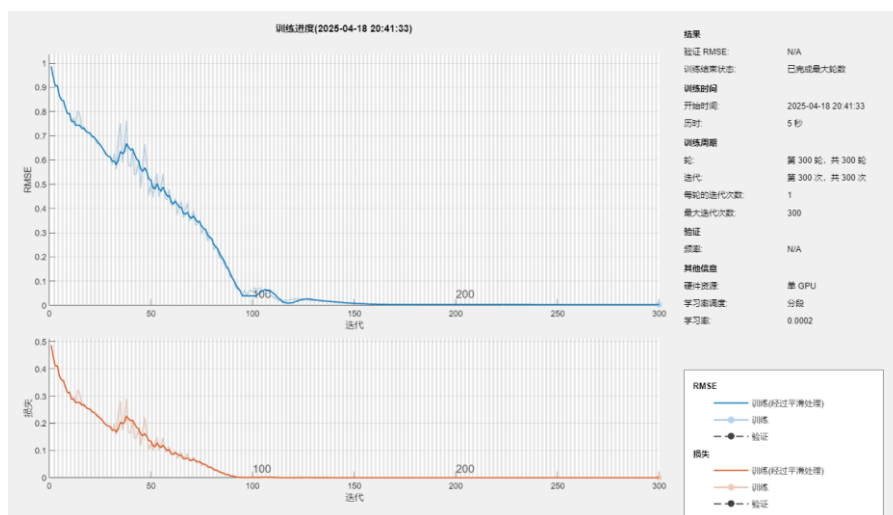


图 4-12 人工成本预测收敛图

（2）预测结果分析

图 4-13 展示了 2024 年 1 月至 9 月期间实际人工成本与预测人工成本范围的对比结果。模型通过置信区间呈现预测范围，实际人工成本曲线全程紧密贴合

预测中线线,且在 3-8 月均稳定落在预测区间内,未出现超出上下限的异常波动。

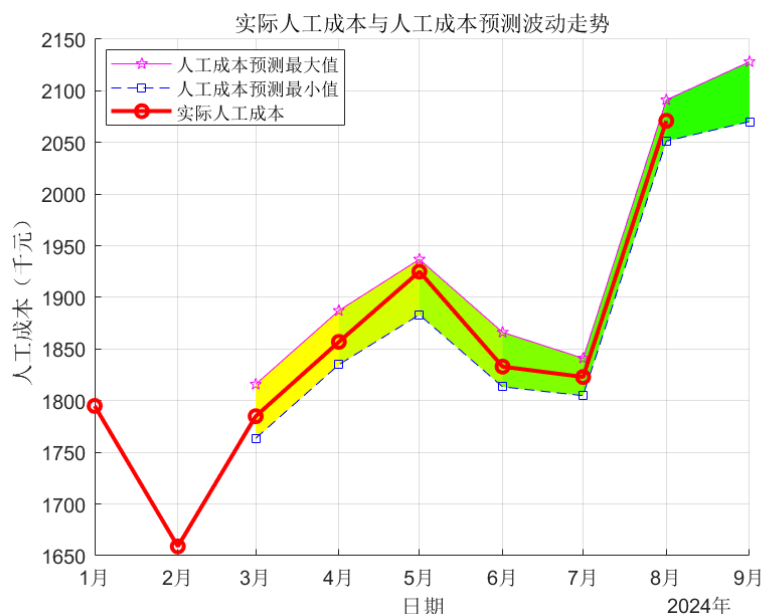


图 4-13 实际人工成本与预测人工成本范围对比图

4.4 本章小结

本章提出了一种基于 Attention-Bi-GRU 网络的多变量时序预测模型,通过结合 Bi-GRU 与 Attention 机制,实现了对企业四型硅片的月销量、售价、单晶方棒价格及人工成本等关键参数的波动趋势预测与合理区间估计。模型采用双向 GRU 编码器提取历史数据的正反向时序依赖特征,结合动态注意力机制自适应分配关键时间步的权重,并通过编码-解码结构实现端到端的多步预测。在此基础上,从产销数据及成本动因中系统提取多维度时序特征,构建高关联性输入数据集。仿真结果表明,模型在训练中 RMSE 指标快速收敛至接近零值,且 3-8 月实际值全程位于预测置信区间内,验证了模型在捕捉非线性时序关系、缓解长期依赖问题方面的有效性,同时证明了其高精度预测能力与稳定性,为企业生产决策提供了可靠的数据支持。

第五章 基于块坐标下降的利润最大化策略设计

5.1 题目三求解思路

在众多因子相互制约与冲突的前提下，问题三要求建立能够辅助决策优化企业利润的数学模型，并依据模型的计算结果，给出 9 月份的生产计划与销售预案。在优化问题建立方面：首先，将问题一中所推导出的月净利润作为优化模型的最大化目标。其次，将问题二中所推导出月销量、售价、单晶方棒价格及人工成本的波动范围作为约束条件，变量本身作为优化变量。最后，除优化变量外其余参数波动不大。因此，取其余参数 1-8 月均值作为 9 月的参数取值。在优化问题求解方面：由于优化目标函数为非凸，常规方法无法直接求解。因此使用块坐标下降将原问题分为两个子问题，使用逐次凸逼近取求解子问题次优解，为企业制定利润最大化策略。

5.2 题目三模型建立

5.2.1 优化目标

本章将月净利润作为优化目标，根据问题一的推导，优化目标函数可以表示如下：

$$\begin{aligned}
 F^{\text{Obj}} &= 0.85 \times (\text{Sale}^{\text{Total}} - \text{Cost}^{\text{Total}}) \\
 &= 0.85 \times \left(\sum_{k=1}^4 Q_k \times P_k^{\text{S-Sale}} + S^{\text{Mud}} - \sum_{k=1}^4 VC_k^{\text{Unite}} \times Q_k - E_2 \right. \\
 &\quad \left. - \sum_{k=1}^6 A_k - C^{\text{Fix}} - C^{\text{Float}} - C^{\text{XS}} - C^{\text{GL}} - C^{\text{CW}} \right)
 \end{aligned} \tag{5-1}$$

其中， F^{Obj} 表示优化目标函数，即月净利润。

5.2.2 约束条件

(1) 销量

根据问题二所预测的波动范围，9 月份 k 型硅片销量需要满足以下约束条件：

$$Q_k^{\min} \leq Q_k \leq Q_k^{\max}, k \in \mathcal{K} \tag{5-2}$$

其中， Q_k^{\min} 和 Q_k^{\max} 分别表示 9 月份 k 型硅片销量波动范围的下限和上限，

具体见表 5-1。

表 5-1 9 月份四种类型硅片销量波动区间

类型 销量	182.2*183.75*130	210*210*130	210*182*130	210*210*150
波动上限	7977	4886	9473	9449
波动下限	7228	4436	8927	8925

(2) 售价

根据问题二所预测的波动范围,9 月份 k 型硅片售价需要满足以下约束条件:

$$P_k^{S-Sale-min} \leq P_k^{S-Sale} \leq P_k^{S-Sale-max}, k \in \mathcal{K} \quad (5-3)$$

其中, $P_k^{S-Sale-min}$ 和 $P_k^{S-Sale-max}$ 分别表示 9 月份 k 型硅片售价波动范围的下限和上限, 具体见表 5-2。

表 5-2 9 月份四种类型硅片售价波动区间

类型 售价	182.2*183.75*130	210*210*130	210*182*130	210*210*150
波动上限	1.7127	1.6403	1.5880	1.3374
波动下限	1.6163	1.5748	1.5119	1.2615

(3) 单晶方棒价格

根据问题二所预测的波动范围,9 月份 k 型硅片单晶方棒价格需要满足以下约束条件:

$$S_k^{\min} \leq S_k \leq S_k^{\max}, k \in \mathcal{K} \quad (5-4)$$

其中, S_k^{\min} 和 S_k^{\max} 分别表示 9 月份 k 型硅片单晶方棒价格波动范围的下限和上限, 具体见表 5-3。

表 5-3 9 月份四种类型硅片单晶方棒价格波动区间

类型 单晶方棒	182.2*183.75*130	210*210*130	210*182*130	210*210*150
波动上限	58.4157	58.8290	57.5959	48.9218
波动下限	55.6317	57.1754	55.5494	47.2849

(4) 人工成本

根据问题二所预测的波动范围, 人工成本需要满足以下约束条件:

$$Cost^{RGCB-min} \leq Cost^{RGCB} \leq Cost^{RGCB-max} \quad (5-5)$$

其中, $Cost^{RGCB-\min}$ 和 $Cost^{RGCB-\max}$ 分别表示 9 月份人工成本波动范围的下限和上限, 具体见表 5-4。

表 5-4 9 月份人工成本波动区间

参数名称	波动上限	波动下限
人工成本	2128.2	2070.2

(5) 销量与售价映射关系

在实际中, 销量和售价这两个参数是耦合且负相关的。因此, 为了使算法生成的策略更加符合实际, 本节构建了销量与售价之间的映射关系。

首先, 引入销量映射因子 $\alpha_k^{XL} \in [0,1]$ 和售价映射因子 $\alpha_k^{SJ} \in [0,1]$, 他们的计算方式如下所示:

$$\alpha_k^{XL} = \frac{Q_k - Q_k^{\min}}{Q_k^{\max} - Q_k^{\min}} \quad (5-6)$$

$$\alpha_k^{SJ} = \frac{P_k^{S-Sale} - P_k^{S-Sale-\min}}{P_k^{S-Sale-\max} - P_k^{S-Sale-\min}} \quad (5-7)$$

同时, 销量映射因子 $\alpha_k^{XL} \in [0,1]$ 和售价映射因子 $\alpha_k^{SJ} \in [0,1]$ 满足约束条件如下所示:

$$\alpha_k^{XL} + \alpha_k^{SJ} = 1, k \in \mathcal{K} \quad (5-8)$$

为了防止策略规划的过程中存在偶然性, 本章提出一种基于狄利克雷分布的概率重参数化策略^[6]。设原始确定性映射因子为 $(\alpha_k^{XL}, \alpha_k^{SJ})$, 将其扩展为随机性映射因子 $(\alpha_k^{XL-\text{rand}}, \alpha_k^{SJ-\text{rand}})$, 满足条件如下所示:

$$(\alpha_k^{XL-\text{rand}}, \alpha_k^{SJ-\text{rand}}) \sim \text{Dirichlet}(\beta\alpha_k^{XL}, \beta\alpha_k^{SJ}), k \in \mathcal{K} \quad (5-9)$$

其中, $\beta > 0$ 为浓度参数, 用于调节扰动强度。改进后的变量概率密度函数表示如下:

$$f(\alpha_k^{XL-\text{rand}}, \alpha_k^{SJ-\text{rand}}; \beta) = \frac{\Gamma(\beta)}{\Gamma(\beta\alpha_k^{XL})\Gamma(\beta\alpha_k^{SJ})} (\alpha_k^{XL-\text{rand}})^{\beta\alpha_k^{XL}-1} (\alpha_k^{SJ-\text{rand}})^{\beta\alpha_k^{SJ}-1}, k \in \mathcal{K} \quad (5-10)$$

其中, $\Gamma(\cdot)$ 为伽马函数。

5.2.3 优化问题建立

根据上面对优化目标和约束条件的分析, 优化问题如下建立:

$$\begin{aligned}
 \text{P5.1} \quad & \max_{\substack{\{Q_k\}, \{P_k^{\text{S-Sale}}\} \\ \{S_k\}, \text{Cost}^{\text{RGCB}}}} F^{\text{Obj}} \\
 \text{s.t.} \quad & \text{C5.1: } Q_k^{\min} \leq Q_k \leq Q_k^{\max}, k \in \mathcal{K} \\
 & \text{C5.2: } P_k^{\text{S-Sale-min}} \leq P_k^{\text{S-Sale}} \leq P_k^{\text{S-Sale-max}}, k \in \mathcal{K} \\
 & \text{C5.3: } S_k^{\min} \leq S_k \leq S_k^{\max}, k \in \mathcal{K} \\
 & \text{C5.4: } \text{Cost}^{\text{RGCB-min}} \leq \text{Cost}^{\text{RGCB}} \leq \text{Cost}^{\text{RGCB-max}} \\
 & \text{C5.5: } (\alpha_k^{\text{XL-rand}}, \alpha_k^{\text{SJ-rand}}) \sim \text{Dirichlet}(\beta \alpha_k^{\text{XL}}, \beta \alpha_k^{\text{SJ}}), k \in \mathcal{K}
 \end{aligned} \tag{5-11}$$

由于优化目标函数中售价与销量的乘积耦合关系导致其非凸性，使得该优化问题无法通过常规凸优化方法直接求解。

5.3 基于块坐标下降的策略优化算法

5.3.1 基于块坐标下降分割原问题

为了解决原问题 P5.1 的非凸性，本章采用块坐标下降将原问题分解为两个子问题进行交替迭代优化，获得策略次优解^[5]。分解的两个子问题具体如下：

(1) 子问题 1：固定售价，优化其余三个变量

假设售价已知，原问题可以改写为如下所示：

$$\begin{aligned}
 \text{P5.2} \quad & \max_{\{Q_k\}, \{S_k\}, \text{Cost}^{\text{RGCB}}} F^{\text{Obj}} \\
 \text{s.t.} \quad & \text{C5.1, C5.3, C5.4, C5.5}
 \end{aligned} \tag{5-12}$$

该子问题是凸的，可以直接使用 MATLAB 软件中 CVX 工具包求解。

(1) 子问题 2：固定销量，优化其余三个变量

假设销量已知，原问题可以改写为如下所示：

$$\begin{aligned}
 \text{P5.3} \quad & \max_{\{P_k^{\text{S-Sale}}\}, \{S_k\}, \text{Cost}^{\text{RGCB}}} F^{\text{Obj}} \\
 \text{s.t.} \quad & \text{C5.2, C5.3, C5.4, C5.5}
 \end{aligned} \tag{5-13}$$

该子问题是凸的，可以直接使用 MATLAB 软件中 CVX 工具包求解。

5.3.2 双层交替迭代策略优化算法

根据上节分析，原问题 P5.1 是一个非凸问题难以直接求解。为了解决这一问题，本章首先将原问题 P5.1 分解为两个子问题，然后使用 CVX 分别求解子问题，最终基于两个子问题提出一种联合优化迭代算法，从而获得原问题的次优解。双层交替迭代策略优化算法的伪代码如表 5-5 所示。

表 5-5 双层交替迭代策略优化算法伪代码

算法 5.1：双层交替迭代策略优化算法

输入： 9 月份 k 型硅片销量波动范围 $[Q_k^{\min}, Q_k^{\max}]$ ，9 月份 k 型硅片售价波动范围 $[P_k^{\text{S-Sale-min}}, P_k^{\text{S-Sale-max}}]$ ，9 月份 k 型硅片单晶方棒价格波动范围 $[S_k^{\min}, S_k^{\max}]$ ，人工成本波动范围 $[Cost^{\text{RGCb-min}}, Cost^{\text{RGCb-max}}]$ ，最大迭代次数 L^{Joint}

输出： 销量集合 $\{Q_k\}$ ，售价集合 $\{P_k^{\text{S-Sale}}\}$ ，单晶方棒价格集合 $\{S_k\}$ ，人工成本集合 $\{Cost^{\text{RGCb}}\}$

```

1: for  $l \leftarrow 1$  to  $L^{\text{Joint}}$  do
2:   给定  $\{P_k^{\text{S-Sale}}\}^{(l-1)}$ ，利用 CVX 解决子问题 P5.2，得到销量集合  $\{Q_k\}^*$ 
3:    $\{Q_k\}^{(l)} \leftarrow \{Q_k\}^*$  // 更新销量
4:   给定  $\{Q_k\}^{(l)}$ ，利用 CVX 解决子问题 P5.3，得到售价集合  $\{P_k^{\text{S-Sale}}\}^*$ ，单晶方棒价格集合  $\{S_k\}^*$ ，人工成本集合  $\{Cost^{\text{RGCb}}\}^*$ 
5:    $\{P_k^{\text{S-Sale}}\}^{(l)} \leftarrow \{P_k^{\text{S-Sale}}\}^*$  // 更新售价
6:    $\{S_k\}^{(l)} \leftarrow \{S_k\}^*$  // 更新单晶方棒价格
7:    $\{Cost^{\text{RGCb}}\}^{(l)} \leftarrow \{Cost^{\text{RGCb}}\}^*$  // 更新人工成本
8: end for
    
```

5.3.3 算法复杂度分析

本节对算法 5.1 的计算复杂度进行理论推导。该算法通过块坐标下降法将原非凸优化问题分解为两个凸子问题（子问题 1 和子问题 2），并交替迭代求解^[7]。

以下为复杂度分析的关键步骤：

（1）子问题 1：对于 4 种硅片类型，需优化销量、单晶方棒价格及人工成本，共 $4 \times 3 = 124 \times 3 = 12$ 个变量。假设子问题为线性规划问题，采用内点法求解，其复杂度为 $\mathcal{O}(n^{3.5})$ ，其中 n 为变量数。因此，子问题 1 的复杂度为：

$$\mathcal{O}(12^{3.5}) = \mathcal{O}(12^3 \times \sqrt{12}) \approx \mathcal{O}(1728 \times 3.464) \approx \mathcal{O}(5987) \quad (5-14)$$

（2）子问题 2：同样为 $4 \times 3 = 124 \times 3 = 12$ 个变量。因此，子问题 2 的计算复杂度与子问题 1 相同，即 $\mathcal{O}(5987)$ 。

由于每次迭代需分别求解子问题 1 和子问题 2，因此单次迭代的复杂度计算方式如下：

$$\mathcal{O}(5987 + 5987) = \mathcal{O}(11974) \quad (5-15)$$

由于算法迭代次数维 L^{Joint} ，双层交替迭代策略优化算法的计算复杂度为 $O(11974 \times L^{\text{Joint}})$ 。

5.4 题目三结果分析

5.4.1 算法收敛性

下图 5-1 展示了双层交替迭代策略优化策略的收敛过程。横轴为迭代次数，纵轴为最大利润。从曲线趋势可以看出，随着迭代次数增加，算法计算的利润值逐步提升并最终趋于稳定，表明算法在有限迭代内实现了有效收敛。初始阶段利润增长较快，反映出算法对解空间的快速探索能力；后期曲线逐渐平缓，说明解的质量趋于最优且波动较小。双层交替迭代机制通过外层策略调整与内层参数优化的协同作用，有效避免了局部最优问题，最终收敛时的利润值达到较高水平，验证了算法在兼顾效率与精度方面的优势。

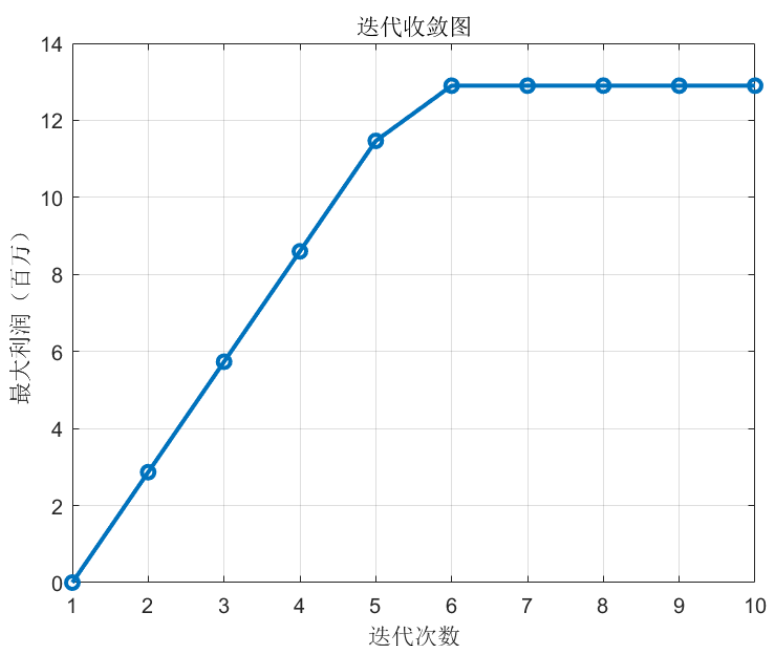


图 5-1 双层交替迭代策略优化算法收敛图

5.4.2 策略分析

(1) 总销售额

根据 3.2.1 小结对月总销售额分析可知，其由销量与单价决定。9 月份四种类型硅片的销量与单价规划方案如表 5-6 所示。

表 5-6 9 月份四种类型硅片的销量与单价规划方案

类型 单晶方棒	182.2*183.75*130	210*210*130	210*182*130	210*210*150
销量 (片)	7602496	4660996	9199996	9186996
单价 (元)	1.6645	1.6075	1.5499	1.2994

根据公式(3-1), 9 月份四种类型硅片的月总销售额规划方案如表 5-7 所示。

表 5-7 9 月份四种类型硅片的月总销售额规划方案

类型 单晶方棒	182.2*183.75*130	210*210*130	210*182*130	210*210*150
月总销售 额 (元)	12654291.31	7492743.20	14259313.91	11937835.28

(2) 变动成本

根据 3.2.2 小结对月变动成本分析可知, 其由单晶方棒价格等参数决定。9 月份四种类型硅片的单晶方棒价格规划方案如表 5-8 所示。

表 5-8 9 月份四种类型硅片的单晶方棒价格规划方案

类型 单晶方棒	182.2*183.75*130	210*210*130	210*182*130	210*210*150
单晶方棒 价 (元)	55.631743	57.175422	55.549417	47.284879

其余非优化变量参数取 1-8 月均值。根据公式 (3-2), 9 月份四种类型硅片的变动成本规划方案如表 5-10 所示。

表 5-10 9 月份四种类型硅片的变动成本规划方案

类型 单晶方棒	182.2*183.75*130	210*210*130	210*182*130	210*210*150
变动成本 (元)	5971938.7	4788083.5	7874894.18	8119502.38

(3) 其余成本

根据 3.2.2 小结可知, 其余成本包括生产公用成本、人工成本、销售费用、管理费用、财务费用、折旧、营业税费。它们的具体规划方案如表 5-11 所示。

表 5-11 9 月份其余成本规划方案

其余成本	金额
生产公用成本（元）	299803
人工成本（元）	2070219.27
销售费用（元）	96929
管理费用（元）	558583
财务费用（元）	49500
折旧（元）	1200000
营业税费（元）	138000

（4）9 月净利润

将（1-3）内数据进行归纳总结，9 月份总销售额、变动成本和其他成本规划方案如表 5-12 所示。

表 5-12 9 月份总销售额、变动成本和其他成本规划方案

总销售额（元）	变动成本（元）	其他成本（元）
46344184	26754418.8	4413034.3

根据公式（3-7）和公式（3-9），9 月份利润（税前）和净利润（税后）如表 5-13 所示。

表 5-12 9 月份利润（税前）和净利润（税后）

利润（税前）	净利润（税后）
15176730.66	12900221.06

5.4.3 月净利润

图 5-2 展示了 2024 年 1 月至 9 月的销售净利润变化趋势（单位：百万元），其中 1 月至 8 月为历史真实数据，9 月为采用所提出策略后的预测结果。从图中可以看出，前 8 个月净利润呈现一定波动性，而 9 月利润显著提升至全年最高水平，充分体现了该策略的优化效果。这一结果不仅验证了策略的可行性与有效性，也为后续业务规划提供了关键数据支撑。

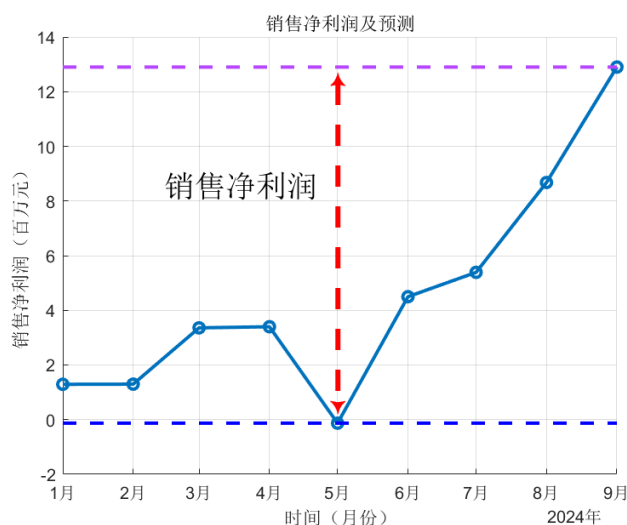


图 5-2 月净利润示意图

5.5 本章小结

针对多因子相互制约下的企业利润优化问题，本章提出了一种基于块坐标下降的协同优化策略，旨在制定9月份生产计划与销售预案，实现月净利润最大化。首先，以问题一推导的月净利润为目标函数，结合问题二预测的销量、售价、单晶方棒价格及人工成本波动范围构建约束条件，建立了非凸优化模型。针对目标函数的非凸性，采用块坐标下降法将原问题分解为两个凸子问题（固定售价优化销量、单晶方棒价格与人工成本；固定销量优化售价、单晶方棒价格与人工成本），通过交替迭代求解子问题，并结合狄利克雷分布的概率重参数化策略处理变量耦合与偶然性干扰，确保策略的鲁棒性。

仿真结果表明，所提算法在有限迭代次数内快速收敛，利润值逐步提升并趋于稳定，验证了算法的有效性。策略分析显示，通过优化四种硅片类型的销量、售价及成本参数，9月份总销售额达到4634.42万元，变动成本与其他成本分别降至2675.44万元和441.30万元，最终实现税后净利润1290.02万元，较历史数据显著提升，为全年最高水平。该结果不仅证实了块坐标下降法与交替迭代机制在非凸优化问题中的实用性，也为企业动态决策提供了数据驱动的科学依据。

第六章 基于多源数据融合与大语言模型的智能产销决策优化体系构建

6.1 多源数据整合与预处理

6.1.1 企业内部生产销售数据提取

企业内部生产销售数据是模型优化的核心输入，涵盖生产量、成本结构、销量、售价及单晶方棒价格等关键指标。通过企业提取结构化数据后，需对多源异构数据进行标准化处理。例如，将四型硅片的月度销量 Q_k 与总生产成本 $Cost^{Total}$ 按时间序列对齐，并通过归一化消除量纲差异：

$$Q_k^* = \frac{Q_k - Q_k^{\min}}{Q_k^{\max} - Q_k^{\min}} \quad (6-1)$$

$$Cost^{Total*} = \frac{Cost^{Total} - \mu^{Cost}}{\sigma^{Cost}} \quad (6-2)$$

其中， μ^{Cost} 和 σ^{Cost} 分别代表总生产成本的均值和标准差。此外，需构建动态关联矩阵，反映硅单耗、耗材价格与净利润的非线性关系，为后续模型提供特征基础。

6.1.2 外部政策与市场数据采集

外部数据通过爬取政府公开政策文件、行业数据库（如 PV-Tech、Wind）及社交媒体舆情文本获取。非结构化文本采用自然语言处理（NLP）技术进行关键词提取与情感分析，量化政策支持力度 ε 和市场信心指数 ρ [8]。例如，政策文件中出现关于“硅”相关条款的频次 f^{Policy} 可映射为政策影响因子：

$$\varepsilon = \frac{f^{Policy}}{T} \times \omega^{Policy} \quad (6-3)$$

其中， T 和 ω^{Policy} 分别表示文本总词数和行业专家赋权值。市场舆情数据则通过双向编码器表征（BERT）模型生成语义向量 $\mathbf{v}_{Sentiment}$ ，并结合时间衰减函数 $\gamma(t) = e^{-\lambda t}$ 动态调整其权重，以捕捉市场波动趋势 [8]。

6.1.3 数据清洗与特征工程

针对数据缺失与噪声问题，采用多重插补法（MICE）填补缺失值，并结合箱线图与 Grubbs 检验识别异常值^[9]。时序数据通过 Hodrick-Prescott 滤波分解为趋势项 τ_t 与周期项 c_t ，以消除季节性干扰：

$$S_t = \tau_t + c_t \quad (6-4)$$

特征工程阶段，利用互信息（MI）筛选与利润强相关的关键因子，定义互信息计算为：

$$I(X;Y) = \sum_{x \in X} \sum_{y \in Y} p(x,y) \log \frac{p(x,y)}{p(x)p(y)} \quad (6-5)$$

对高维特征（如耗材价格、电力成本）进行主成分分析，提取主成分 $\mathbf{Z}=\mathbf{XW}$ ，其中 \mathbf{W} 为特征向量矩阵，确保模型输入的低冗余性与高解释性。最终，构建包含内部生产指标、外部政策量化因子及市场情绪特征的多维数据集，为智能决策模型提供可靠输入。

6.2 大语言模型与数学模型的协同建模

6.2.1 开源大模型选型与领域知识注入

为提升利润计算模型的决策泛化能力，选择开源大语言模型（如 Llama-2、Chat-GLM）作为外部知识引擎。首先，基于晶硅片行业语料库（政策文件、技术手册、市场分析报告）对大模型进行领域适应微调，优化其语义理解与预测能力。微调目标函数设计为：

$$\mathcal{L}^{\text{Fine-tune}} = \vartheta \mathcal{L}^{\text{LM}} + (1-\vartheta) \mathcal{L}^{\text{Profit}} \quad (6-6)$$

其中， \mathcal{L}^{LM} 为语言模型损失函数， $\mathcal{L}^{\text{Profit}}$ 为基于第三章利润模型（式 3-9）的利润预测均方误差， $\vartheta \in [0,1]$ 为平衡系数。进一步，将数学模型中的关键因子（如销量 Q_k 、成本 $Cost^{\text{Total}}$ ）与大模型输出的外部特征（政策支持力度 ε 、市场信心指数 ρ ）通过特征拼接形成联合输入向量：

$$\mathbf{X}^{\text{Fused}} = [Q_k, Cost^{\text{Total}}, \varepsilon, \rho] \oplus \text{Embedding}(\text{External Text}) \quad (6-7)$$

其中， \oplus 表示向量拼接操作，确保结构化数据与非结构化文本的深度融合。

6.2.2 动态融合机制设计

为协调大模型与数学模型的优势，设计动态权重分配机制。定义大模型预测利润 $\hat{\pi}^{\text{LLM}}$ 与数学模型预测利润 $\hat{\pi}^{\text{Math}}$ ，融合后利润为：

$$\hat{\pi}^{\text{Fused}} = w(t)\hat{\pi}^{\text{LLM}} + (1-w(t))\hat{\pi}^{\text{Math}} \quad (6-8)$$

权重 $w(t)$ 通过历史预测误差自适应调整：

$$w(t) = \frac{\exp(-\kappa |\pi_{t-1} - \hat{\pi}_{t-1}^{\text{LLM}}|)}{\exp(-\kappa |\pi_{t-1} - \hat{\pi}_{t-1}^{\text{LLM}}|) + \exp(-\kappa |\pi_{t-1} - \hat{\pi}_{t-1}^{\text{Math}}|)} \quad (6-9)$$

其中， κ 为平滑系数。同时，构建反馈闭环：将数学模型计算的成本计算公式（3-9）输入大模型，生成符合成本限制的销售策略建议，再通过数学模型的利润计算验证策略可行性，形成迭代优化流程。

6.2.3 预测与决策结果验证

为验证融合模型的有效性，定义以下评估指标：

（1）预测精度：采用平均绝对百分比误差（MAPE）与 RMSE：

$$\text{MAPE} = \frac{1}{N} \sum_{i=1}^N \left| \frac{\pi_i - \hat{\pi}_i}{\pi_i} \right| \quad (6-10)$$

$$\text{RMSE} = \sqrt{\frac{1}{N} \sum_{i=1}^N (\pi_i - \hat{\pi}_i)^2} \quad (6-11)$$

（2）利润提升率：对比融合模型与独立模型的利润优化能力：

$$\Delta\pi = \frac{\hat{\pi}^{\text{Fused}} - \hat{\pi}^{\text{Baseline}}}{\hat{\pi}^{\text{Baseline}}} \times 100\% \quad (6-12)$$

（3）鲁棒性测试：模拟极端场景（如单晶方棒价格突变 $\pm 20\%$ ），检验模型稳定性。通过蒙特卡洛采样生成 1000 组扰动数据，计算利润预测方差与策略失效概率。

6.3 智能决策系统构建与可行性分析

6.3.1 端到端系统架构设计

智能决策系统采用分层架构设计，包括数据输入层、模型计算层与策略输出

层。数据输入层整合预处理后的多维特征向量 $\mathbf{X}^{\text{Fused}} \in \mathbb{R}^d$ （公式 6-7），通过 API 接口实时接入企业 ERP 系统与外部数据源。模型计算层包含动态融合模型（公式 6-8）与约束优化模块，其中优化目标函数定义为：

$$\begin{aligned} \text{P6.1} \quad & \max_{\mathbf{u}} \hat{\pi}^{\text{Fused}}(\mathbf{u}) \\ \text{s.t.} \quad & C_{\text{total}}(\mathbf{u}) \leq B, \mathbf{u} \in \mathcal{U} \end{aligned} \quad (6-13)$$

其中， \mathbf{u} 为决策变量（如四型硅片产量分配比例）， B 为总成本预算， \mathcal{U} 为可行域。策略输出层通过可视化面板生成可执行方案，并支持人工干预反馈。系统通过微服务架构实现模块解耦，确保高并发与低延迟响应。

6.3.2 实时更新与模型迭代

系统采用“在线-离线”混合训练策略^[10]。在线阶段，基于滑动窗口机制更新模型参数。设窗口长度为 T^{Win} ，第 t 时刻的模型参数 θ_t 更新规则为：

$$\theta_{t+1} = \theta_t - \eta \nabla_{\theta} \left(\sum_{i=t-T^{\text{Win}}+1}^t \mathcal{L}^{\text{Fine-tune}}(\mathbf{X}_i, \pi_i) \right) \quad (6-14)$$

其中， η 表示学习率， $\nabla_{\theta}(\bullet)$ 表示梯度计算。离线阶段，每月初利用全量历史数据重新训练模型，防止概念漂移。同时，设计异常检测机制：当实时数据分布偏移度 $D^{\text{KL}}(P^{\text{Real}} \parallel P^{\text{Train}}) > \xi$ 时，触发模型紧急迭代，其中 D^{KL} 为 Kullback-Leibler (KL) 散度， ξ 为阈值， P^{Real} 和 P^{Train} 分别表示实时和训练数据分布。

6.3.3 应用场景验证

为验证系统效能，设计两类测试场景：

（1）历史回溯测试：利用 2024 年 1-8 月数据模拟决策过程，对比实际利润与系统推荐利润。定义利润达成率：

$$R^{\text{Profit}} = \frac{1}{N} \sum_{i=1}^N \mathbb{I}(\hat{\pi}_i^{\text{Predict}} \geq \pi_i^{\text{Real}}) \times 100\% \quad (6-15)$$

（2）极端市场波动模拟：通过蒙特卡洛采样生成单晶方棒价格扰动序列 $\{\Delta S_t\}_{t=1}^{1000}$ ，服从 $\Delta S_t \sim \mathcal{N}(0, 0.2^2)$ ，计算利润波动方差^[11]：

$$\sigma_{\pi}^2 = \frac{1}{1000} \sum_{t=1}^{1000} (\hat{\pi}_t - \bar{\pi})^2 \quad (6-16)$$

6.4 本章小结

本章围绕智能产销决策优化体系的构建，提出了多源数据融合与大语言模型协同的创新方法。首先，通过多源数据整合与预处理，实现了企业内部生产销售数据与外部政策、市场信息的标准化对齐，构建了低冗余、高解释性的特征工程框架。其次，在模型融合层面，设计动态权重分配机制与反馈闭环，将大语言模型的语义推理能力与数学模型的定量计算优势深度结合，显著提升了利润预测精度与策略泛化性。最后，基于端到端系统架构，开发了支持实时更新与多场景验证的智能决策平台，设计了历史回溯测试与极端波动场景下利润方差测试。本研究为晶硅片企业提供了数据驱动与人工智能增强的决策范式，兼具理论创新性与工程实用性。

参考文献

- [1] Y. Zhang, W. Ma, F. Huang, X. Deng and W. Jiang. A Novel Air Target Intention Recognition Method Based on Sample Reweighting and Attention-Bi-GRU [J]. IEEE Systems Journal, 2024, 18(1): 501-504.
- [2] J. Bi, K. Xu, H. Yuan, J. Zhang and M. Zhou. Network Attack Prediction With Hybrid Temporal Convolutional Network and Bidirectional GRU[J]. IEEE Internet of Things Journal, 2024, 11(7): 12619-12630.
- [3] N. S. Chauhan, N. Kumar and A. Eskandarian. A Novel Confined Attention Mechanism Driven Bi-GRU Model for Traffic Flow Prediction[J]. IEEE Transactions on Intelligent Transportation Systems, 2024, 25(8): 9181-9191.
- [4] J. Yuan, R. Jin, L. Wang and T. Wang. A Nonintrusive Load Identification Method Based on Dual-Branch Attention GRU Fusion Network[J]. IEEE Transactions on Instrumentation and Measurement, 2024, 73: 1-10.
- [5] 钟章生,袁智勇.基于块坐标下降算法的优化哈希数据流频率估计[J].广西大学学报(自然科学版),2022,47(06):1585-1598.
- [6] 李媛媛,陈文静,王辉.科技金融政策、资金网络与企业创新绩效——基于潜在狄利克雷分布模型[J].科技管理研究,2022,42(06):28-35.
- [7] K. Wang, Y. Sun, C. Li and P. Liu. Optimal position and power allocation for RSMA multigroup multicast and multibeam UAV-assisted communication[J]. Physical Communication, 2025, 68: 102563.
- [8] 贺云隆,杨东华,宋晓林,等.基于 BERT 的混合电力信息报文自动提取技术[J].无线电工程,1-13.
- [9] 孙玲莉,董世杰,杨贵军.常用多重插补法的插补重数选择[J].统计与决策,2019,35(23):5-10.
- [10] K. Wang, Y. Sun and T. Q. S. Quek. Deep Learning Based Energy-Efficient Hybrid RSMA for UAV-Assisted mmWave Communications[J] IEEE Transactions on Vehicular Technology, 2025, 74(3): 4625-4638.
- [11] 汤琦,孟贤.基于部分均匀采样的蒙特卡洛仿真模型[J].光通信技术,2024,48(05):86-90.

附录 A 文件列表

A.1 单晶方棒价格预测 Matlab 源码文件

四种类型硅片单晶方棒价格预测 Matlab 源码文件列表如表 A-1 所示。

表 A-1 单晶方棒价格预测 Matlab 源码文件列表

硅片类型	文件名
182.2*183.75*130	LSTM_pre_m_p_1.m
210*210*130	LSTM_pre_m_p_2.m
210*182*130	LSTM_pre_m_p_3.m
210*210*150	LSTM_pre_m_p_4.m

A.2 销量预测 Matlab 源码文件

四种类型硅片销量预测 Matlab 源码文件列表如表 A-2 所示。

表 A-2 销量预测 Matlab 源码文件列表

硅片类型	文件名
182.2*183.75*130	LSTM_pre_s_1.m
210*210*130	LSTM_pre_s_2.m
210*182*130	LSTM_pre_s_3.m
210*210*150	LSTM_pre_s_4.m

A.3 售价预测 Matlab 源码文件

四种类型硅片售价预测 Matlab 源码文件列表如表 A-3 所示。

表 A-3 售价预测 Matlab 源码文件列表

硅片类型	文件名
182.2*183.75*130	LSTM_pre_s_p_1.m
210*210*130	LSTM_pre_s_p_2.m
210*182*130	LSTM_pre_s_p_3.m
210*210*150	LSTM_pre_s_p_4.m

A.4 仿真图源码文件

仿真图 Matlab 源码文件列表如表 A-4 所示。

表 A-4 仿真图 Matlab 源码文件列表

论文图片索引	文件名
图 4-12, 图 4-13	plot_labor_cost.m
图 4-10, 图 4-11(a)	plot_m_p_1.m
图 4-10, 图 4-11(b)	plot_m_p_2.m
图 4-10, 图 4-11(c)	plot_m_p_3.m
图 4-10, 图 4-11(d)	plot_m_p_4.m
图 3-2, 图 5-1, 图 5-2	plot_pro3_cxc_profit.m
图 4-6, 图 4-7(a)	plot_s_1.m
图 4-6, 图 4-7(b)	plot_s_2.m
图 4-6, 图 4-7(c)	plot_s_3.m
图 4-6, 图 4-7(d)	plot_s_4.m
图 4-8, 图 4-9(a)	plot_s_p_1.m
图 4-8, 图 4-9(b)	plot_s_p_2.m
图 4-8, 图 4-9(c)	plot_s_p_3.m
图 4-8, 图 4-9(d)	plot_s_p_4.m

A.5 其余文件

其余文件列表如表 A-5 所示。

表 A-5 其余文件列表

描述	文件名
预测区间数据存储文件	pridicted_max_min.xlsx
销售数据预测存储文件	pridicted_Sales_Table.xlsx
交替优化 M 变量的 Matlab 源码文件	pro_3_cvx_generate_M.m
交替优化 Q 变量的 Matlab 源码文件	pro_3_cvx_generate_Q.m
硅片相关数据存储文件	si_sales.xlsx
1-8 月数据提取文件	the_1st_to_8th_month_data.xlsx
9 月预测数据与校验文件	the_9th_month_data.xlsx
人工成本预测 Matlab 源码文件	LSTM_pre_l_c.m

附录 B 源程序代码

B.1 人工成本预测 Matlab 源码

```
clc;
clear;
close;
si_data = readtable('si_sales.xlsx', 'Sheet', 'labor_cost');
for i = 1:length(si_data.collect_day)
    si_data.collect_day(i) = strip(si_data.collect_day(i), '');
end
loop = 1;
for ii = 1:loop
    for iii = 0:length(si_data.labor_cost) - 2
        Y = si_data.labor_cost;
        Y = Y(1:length(Y) - iii);
        data = Y';
        numTimeStepsTrain = floor(0.9 * numel(data));
        numTimeStepsTrain = max(numTimeStepsTrain, 1);
        dataTrain = data(1:numTimeStepsTrain + 1);
        dataTest = data(numTimeStepsTrain + 1:end);
        mu = mean(dataTrain);
        sig = std(dataTrain);
        dataTrainStandardized = (dataTrain - mu) / sig;
        XTrain = dataTrainStandardized(1:end-1);
        YTrain = dataTrainStandardized(2:end);
        numFeatures = 1;
        numResponses = 1;
        numHiddenUnits = 200;
        layers = [ ...
            sequenceInputLayer(numFeatures)
```

```

        lstmLayer(numHiddenUnits)
        fullyConnectedLayer(numResponses)
        regressionLayer];
options = trainingOptions('adam', ...
    'MaxEpochs', 300, ...
    'GradientThreshold', 1, ...
    'InitialLearnRate', 0.005, ...
    'LearnRateSchedule', 'piecewise', ...
    'LearnRateDropPeriod', 125, ...
    'LearnRateDropFactor', 0.2, ...
    'Verbose', 0, ...
    'Plots', 'training-progress');
net = trainNetwork(XTrain, YTrain, layers, options);
dataTestStandardized = (dataTest - mu) / sig;
XTest = dataTestStandardized(1:end-1);
net = predictAndUpdateState(net, XTrain);
YPred = YTrain(end);
futureMonths = 1;
for jj = 1:futureMonths
    [net, YPred] = predictAndUpdateState(net, YPred(end));
end
YTest = dataTest(end);
YTest = (YTest - mu) / sig;
rmse = sqrt(mean((YPred - YTest).^2));
YPred = sig * YPred + mu;
YTest = sig * YTest + mu;
pridictData(ii, 9 - iii) = YPred;
end
end
predictedSalesTable = array2table(pridictData);
excelFile = 'pridicted_Sales_Table.xlsx';
sheetName = 'pridicted_labor_cost';

```

```
writetable(predictedSalesTable, excelFile, 'Sheet', sheetName, 'WriteMode',  
'overwrite');
```

B.2 单晶方棒价格预测 Matlab 源码

```
clc;  
clear;  
close;  
si_data = readtable('si_sales.xlsx', 'Sheet', 's1_material_price');  
for i = 1:length(si_data.collect_day)  
    si_data.collect_day(i) = strip(si_data.collect_day(i), '');  
end  
loop = 1;  
for ii = 1:loop  
    for iii = 0:length(si_data.material_price) - 2  
        Y = si_data.material_price;  
        Y = Y(1:length(Y) - iii);  
        data = Y';  
        numTimeStepsTrain = floor(0.9 * numel(data));  
        numTimeStepsTrain = max(numTimeStepsTrain, 1);  
        dataTrain = data(1:numTimeStepsTrain + 1);  
        dataTest = data(numTimeStepsTrain + 1:end);  
        mu = mean(dataTrain);  
        sig = std(dataTrain);  
        dataTrainStandardized = (dataTrain - mu) / sig;  
        XTrain = dataTrainStandardized(1:end-1);  
        YTrain = dataTrainStandardized(2:end);  
        numFeatures = 1;  
        numResponses = 1;  
        numHiddenUnits = 200;  
        layers = [ ...  
            sequenceInputLayer(numFeatures)
```

```
lstmLayer(numHiddenUnits)
fullyConnectedLayer(numResponses)
regressionLayer];
options = trainingOptions('adam', ...
    'MaxEpochs', 500, ...
    'GradientThreshold', 1, ...
    'InitialLearnRate', 0.005, ...
    'LearnRateSchedule', 'piecewise', ...
    'LearnRateDropPeriod', 125, ...
    'LearnRateDropFactor', 0.2, ...
    'Verbose', 0, ...
    'Plots', 'training-progress');
net = trainNetwork(XTrain, YTrain, layers, options);
dataTestStandardized = (dataTest - mu) / sig;
XTest = dataTestStandardized(1:end-1);
net = predictAndUpdateState(net, XTrain);
YPred = YTrain(end);
futureMonths = 1;
for jj = 1:futureMonths
    [net, YPred] = predictAndUpdateState(net, YPred(end));
end
YTest = dataTest(end);
YTest = (YTest - mu) / sig;
rmse = sqrt(mean((YPred - YTest).^2));
YPred = sig * YPred + mu;
YTest = sig * YTest + mu;
pridictData(ii, 9 - iii) = YPred;
end
end
predictedSalesTable = array2table(pridictData);
excelFile = 'pridicted_Sales_Table.xlsx';
sheetName = 'pridicted_material_price_1';
```

```
writetable(predictedSalesTable, excelFile, 'Sheet', sheetName, 'WriteMode',  
'overwrite');
```

B.3 销量预测 Matlab 源码

```
clc;  
clear;  
close;  
si_data = readtable('si_sales.xlsx', 'Sheet', 's1_sales_volume');  
  
for i = 1:length(si_data.collect_day)  
    si_data.collect_day(i) = strip(si_data.collect_day(i), '');  
end  
loop = 50;  
for ii = 1:loop  
    for iii = 0:length(si_data.sales_volume) - 2  
        Y = si_data.sales_volume;  
        Y = Y(1:length(Y) - iii);  
        data = Y';  
        numTimeStepsTrain = floor(0.9 * numel(data));  
        numTimeStepsTrain = max(numTimeStepsTrain, 1);  
        dataTrain = data(1:numTimeStepsTrain + 1);  
        dataTest = data(numTimeStepsTrain + 1:end);  
        mu = mean(dataTrain);  
        sig = std(dataTrain);  
        dataTrainStandardized = (dataTrain - mu) / sig;  
        XTrain = dataTrainStandardized(1:end-1);  
        YTrain = dataTrainStandardized(2:end);  
        numFeatures = 1;  
        numResponses = 1;  
        numHiddenUnits = 200;  
        layers = [ ...
```

```

sequenceInputLayer(numFeatures)
lstmLayer(numHiddenUnits)
fullyConnectedLayer(numResponses)
regressionLayer];
options = trainingOptions('adam', ...
    'MaxEpochs', 300, ...
    'GradientThreshold', 1, ...
    'InitialLearnRate', 0.005, ...
    'LearnRateSchedule', 'piecewise', ...
    'LearnRateDropPeriod', 125, ...
    'LearnRateDropFactor', 0.2, ...
    'Verbose', 0, ...
    'Plots', 'training-progress');
net = trainNetwork(XTrain, YTrain, layers, options);
dataTestStandardized = (dataTest - mu) / sig;
XTest = dataTestStandardized(1:end-1);
net = predictAndUpdateState(net, XTrain);
YPred = YTrain(end);
futureMonths = 1;
for jj = 1:futureMonths
    [net, YPred] = predictAndUpdateState(net, YPred(end));
end
YTest = dataTest(end);
YTest = (YTest - mu) / sig;
rmse = sqrt(mean((YPred - YTest).^2));
YPred = sig * YPred + mu;
YTest = sig * YTest + mu;
pridictData(ii, 9 - iii) = YPred;
end
end
predictedSalesTable = array2table(pridictData);
excelFile = 'pridicted_Sales_Table.xlsx';

```

```
sheetName = 'pridicted_Sales_1';  
writetable(predictedSalesTable, excelFile, 'Sheet', sheetName, 'WriteMode',  
'overwrite');
```

B.4 售价预测 Matlab 源码

```
clc;  
clear;  
close;  
si_data = readtable('si_sales.xlsx', 'Sheet', 's1_sales_price');  
for i = 1:length(si_data.collect_day)  
    si_data.collect_day(i) = strip(si_data.collect_day(i), '');  
end  
loop = 50;  
for ii = 1:loop  
    for iii = 0:length(si_data.sales_price) - 2  
        Y = si_data.sales_price;  
        Y = Y(1:length(Y) - iii);  
        data = Y';  
        numTimeStepsTrain = floor(0.9 * numel(data));  
        numTimeStepsTrain = max(numTimeStepsTrain, 1);  
        dataTrain = data(1:numTimeStepsTrain + 1);  
        dataTest = data(numTimeStepsTrain + 1:end);  
        mu = mean(dataTrain);  
        sig = std(dataTrain);  
        dataTrainStandardized = (dataTrain - mu) / sig;  
        XTrain = dataTrainStandardized(1:end-1);  
        YTrain = dataTrainStandardized(2:end);  
        numFeatures = 1;  
        numResponses = 1;  
        numHiddenUnits = 200;  
        layers = [ ...
```

```

sequenceInputLayer(numFeatures)
lstmLayer(numHiddenUnits)
fullyConnectedLayer(numResponses)
regressionLayer];
options = trainingOptions('adam', ...
    'MaxEpochs', 300, ...
    'GradientThreshold', 1, ...
    'InitialLearnRate', 0.005, ...
    'LearnRateSchedule', 'piecewise', ...
    'LearnRateDropPeriod', 125, ...
    'LearnRateDropFactor', 0.2, ...
    'Verbose', 0, ...
    'Plots', 'training-progress');
net = trainNetwork(XTrain, YTrain, layers, options);
dataTestStandardized = (dataTest - mu) / sig;
XTest = dataTestStandardized(1:end-1);
net = predictAndUpdateState(net, XTrain);
YPred = YTrain(end);
futureMonths = 1;
for jj = 1:futureMonths
    [net, YPred] = predictAndUpdateState(net, YPred(end));
end
YTest = dataTest(end);
YTest = (YTest - mu) / sig;
rmse = sqrt(mean((YPred - YTest).^2));
YPred = sig * YPred + mu;
YTest = sig * YTest + mu;
pridictData(ii, 9 - iii) = YPred;
end
end
predictedSalesTable = array2table(pridictData);
excelFile = 'pridicted_Sales_Table.xlsx';

```

```

sheetName = 'pridicted_Sales_price_1';
writetable(predictedSalesTable, excelFile, 'Sheet', sheetName, 'WriteMode',
'overwrite');

```

B.5 人工成本预测 Matlab 源码

```

clc;
clear;

filename = 'pridicted_Sales_Table.xlsx';
sheetName = 'pridicted_labor_cost';

pridict_si_data = readtable(filename, 'Sheet', sheetName);
dates = datetime(2024, 1, 1):calmonths(1):datetime(2024, 9, 1);
y_max = -inf(1, length(dates(3:9)));
y_min = inf(1, length(dates(3:9)));
for i = 1:50
    x = dates(3:9);
    y = pridict_si_data{i, 3:9};
    y_max = max(y_max, y);
    y_min = min(y_min, y);
end
n = length(x);
yellow = [1, 1, 0];
green = [0, 1, 0];
colors = [linspace(yellow(1), green(1), n); ...
          linspace(yellow(2), green(2), n); ...
          linspace(yellow(3), green(3), n)];

figure;
hold on;
for j = 1:n-1
    fill([x(j), x(j+1), x(j+1), x(j)], ...
         [y_max(j), y_max(j+1), y_min(j+1), y_min(j)], ...
         colors(j,:), 'EdgeColor', 'none');

```

```

end

h_max = plot(x, y_max, '-mp', 'LineWidth', 0.1);
h_min = plot(x, y_min, '--bs', 'LineWidth', 0.1);
si_data = readtable('si_sales.xlsx', 'Sheet', 'labor_cost');
x_si = dates(1:8);
y_si_data = si_data{:, 2};
h_actual_sales = plot(x_si, y_si_data, '-o', 'Color', 'r', 'LineWidth', 2);
legend([h_max, h_min, h_actual_sales], {'人工成本预测最大值', '人工成本预
测最小值', '实际人工成本'}, 'Location', 'northeast');
hold on;
grid on;
xlabel('日期');
ylabel('人工成本（千元）');
title('实际人工成本与人工成本预测波动走势');
hold off;
max_min = [y_max; y_min];
predictedSalesTable = array2table(max_min);
excelFile = 'pridicted_max_min.xlsx';
sheetName = 'pridicted_labor_cost';
writetable(predictedSalesTable, excelFile, 'Sheet', sheetName, 'WriteMode',
'overwrite');

```

B.6 仿真图 Matlab 源码

B.6.1 单晶方棒价格预测走势图 Matlab 源码

```

clc;
clear;
filename = 'pridicted_Sales_Table.xlsx';
sheetName = 'pridicted_material_price_1';
pridict_si_data = readtable(filename, 'Sheet', sheetName);
dates = datetime(2024, 1, 1):calmonths(1):datetime(2024, 9, 1);
y_max = -inf(1, length(dates(3:9)));

```

```

y_min = inf(1, length(dates(3:9)));
for i = 1:50
    x = dates(3:9);
    y = prdict_si_data{i, 3:9};
    y_max = max(y_max, y);
    y_min = min(y_min, y);
end
n = length(x);
yellow = [1, 1, 0];
green = [0, 1, 0];
colors = [linspace(yellow(1), green(1), n); ...
          linspace(yellow(2), green(2), n); ...
          linspace(yellow(3), green(3), n)];
figure;
hold on;
for j = 1:n-1
    fill([x(j), x(j+1), x(j+1), x(j)], ...
         [y_max(j), y_max(j+1), y_min(j+1), y_min(j)], ...
         colors(j,:), 'EdgeColor', 'none');
end
h_max = plot(x, y_max, '-mp', 'LineWidth', 0.1);
h_min = plot(x, y_min, '--bs', 'LineWidth', 0.1);
si_data = readtable('si_sales.xlsx', 'Sheet', 's1_material_price');
x_si = dates(1:8);
y_si_data = si_data{:, 2};
h_actual_sales = plot(x_si, y_si_data, '-o', 'Color', 'r', 'LineWidth', 2);
legend([h_max, h_min, h_actual_sales], {'单晶方棒价格预测最大值', '单晶方
棒价格预测最小值', '实际单晶方棒价格'}, 'Location', 'northeast');
hold on;
grid on;
xlabel('日期');
ylabel('单晶方棒价格（元）');

```

```

title('实际单晶方棒价格与单晶方棒价格预测波动走势（型号1）');
hold off;
max_min = [y_max;y_min];
predictedSalesTable = array2table(max_min);
excelFile = 'pridicted_max_min.xlsx';
sheetName = 'pridicted_material_price_1';
writetable(predictedSalesTable, excelFile, 'Sheet', sheetName, 'WriteMode',
'overwrite');

```

B.6.2 利润预测走势图 Matlab 源码

```

clc;
clear;
Unit_cons_u_1 = [0.01441 0.014377 0.014392 0.014348 0.014334 0.014283
0.014215 0.014200];
Unit_cons_u_1 = sum(Unit_cons_u_1)./8;
Unit_cons_u_2 = [0.01898 0.018943 0.018963 0.018905 0.018886 0.018819
0.018729 0.018710 ];
Unit_cons_u_2 = sum(Unit_cons_u_2)./8;
Unit_cons_u_3 = [0.01643 0.016398 0.016415 0.016365 0.016348 0.016291
0.016213 0.016197 ];
Unit_cons_u_3 = sum(Unit_cons_u_3)./8;
Unit_cons_u_4 = [0.02129 0.021246 0.021267 0.021203 0.021181 0.020910
0.020868 0.020847 ];
Unit_cons_u_4 = sum(Unit_cons_u_4)./8;
Unit_cons_u = [Unit_cons_u_1,Unit_cons_u_2,Unit_cons_u_3,Unit_cons_u_4];
Unit_cons_u = Unit_cons_u*10000;
C_public = 299803;
C_depreciation = 1200000;
C_business_tax = 138000;
C_selling_expenses = 96929;
C_administrative_expenses = 558583;
C_financial_expenses = 49500;

```

```

C_others = C_public + C_depreciation + C_business_tax + C_selling_expenses
+ ...

    C_administrative_expenses + C_financial_expenses;

% data loading
pridicted_Sales_1 = readtable('pridicted_max_min.xlsx', 'Sheet',
'pridicted_Sales_1');
pridicted_Sales_1_max = pridicted_Sales_1{1, end};
pridicted_Sales_1_min = pridicted_Sales_1{2, end};
pridicted_Sales_2 = readtable('pridicted_max_min.xlsx', 'Sheet',
'pridicted_Sales_2');
pridicted_Sales_2_max = pridicted_Sales_2{1, end};
pridicted_Sales_2_min = pridicted_Sales_2{2, end};
pridicted_Sales_3 = readtable('pridicted_max_min.xlsx', 'Sheet',
'pridicted_Sales_3');
pridicted_Sales_3_max = pridicted_Sales_3{1, end};
pridicted_Sales_3_min = pridicted_Sales_3{2, end};
pridicted_Sales_4 = readtable('pridicted_max_min.xlsx', 'Sheet',
'pridicted_Sales_4');
pridicted_Sales_4_max = pridicted_Sales_4{1, end};
pridicted_Sales_4_min = pridicted_Sales_4{2, end};
% 销量，单位 片
pridicted_Sales_max = 1000* [pridicted_Sales_1_max, pridicted_Sales_2_max,
pridicted_Sales_3_max, pridicted_Sales_4_max];
pridicted_Sales_min = 1000* [pridicted_Sales_1_min, pridicted_Sales_2_min,
pridicted_Sales_3_min, pridicted_Sales_4_min];
pridicted_material_price_1 = readtable('pridicted_max_min.xlsx', 'Sheet',
'pridicted_material_price_1');
pridicted_material_price_1_max = pridicted_material_price_1{1, end};
pridicted_material_price_1_min = pridicted_material_price_1{2, end};
pridicted_material_price_2 = readtable('pridicted_max_min.xlsx', 'Sheet',
'pridicted_material_price_2');
pridicted_material_price_2_max = pridicted_material_price_2{1, end};

```



```

pridicted_material_price_2_min = pridicted_material_price_2{2, end};
pridicted_material_price_3 = readtable('pridicted_max_min.xlsx', 'Sheet',
'pridicted_material_price_3');
pridicted_material_price_3_max = pridicted_material_price_3{1, end};
pridicted_material_price_3_min = pridicted_material_price_3{2, end};
pridicted_material_price_4 = readtable('pridicted_max_min.xlsx', 'Sheet',
'pridicted_material_price_4');
pridicted_material_price_4_max = pridicted_material_price_4{1, end};
pridicted_material_price_4_min = pridicted_material_price_4{2, end};
% 原材料价格，单位 元
pridicted_material_price_max = [pridicted_material_price_1_max,
pridicted_material_price_2_max, pridicted_material_price_3_max,
pridicted_material_price_4_max];
pridicted_material_price_min = [pridicted_material_price_1_min,
pridicted_material_price_2_min, pridicted_material_price_3_min,
pridicted_material_price_4_min];
pridicted_sales_price_1 = readtable('pridicted_max_min.xlsx', 'Sheet',
'pridicted_sales_price_1');
pridicted_sales_price_1_max = pridicted_sales_price_1{1, end};
pridicted_sales_price_1_min = pridicted_sales_price_1{2, end};
pridicted_sales_price_2 = readtable('pridicted_max_min.xlsx', 'Sheet',
'pridicted_sales_price_2');
pridicted_sales_price_2_max = pridicted_sales_price_2{1, end};
pridicted_sales_price_2_min = pridicted_sales_price_2{2, end};
pridicted_sales_price_3 = readtable('pridicted_max_min.xlsx', 'Sheet',
'pridicted_sales_price_3');
pridicted_sales_price_3_max = pridicted_sales_price_3{1, end};
pridicted_sales_price_3_min = pridicted_sales_price_3{2, end};
pridicted_sales_price_4 = readtable('pridicted_max_min.xlsx', 'Sheet',
'pridicted_sales_price_4');
pridicted_sales_price_4_max = pridicted_sales_price_4{1, end};
pridicted_sales_price_4_min = pridicted_sales_price_4{2, end};

```

```

% 售价，单位 元
pridicted_sales_price_max = [pridicted_sales_price_1_max,
pridicted_sales_price_2_max, pridicted_sales_price_3_max,
pridicted_sales_price_4_max];
pridicted_sales_price_min = [pridicted_sales_price_1_min,
pridicted_sales_price_2_min, pridicted_sales_price_3_min,
pridicted_sales_price_4_min];
% 人工成本，单位 元
pridicted_labor_cost = readtable('pridicted_max_min.xlsx', 'Sheet',
'pridicted_labor_cost');
pridicted_labor_cost_max = 1000 * pridicted_labor_cost{1, end};
pridicted_labor_cost_min = 1000 * pridicted_labor_cost{2, end};
%% Initialize
Q(:,1) = pridicted_Sales_min' + (pridicted_Sales_max' -
pridicted_Sales_min')*0.5;
M(:,1) = pridicted_sales_price_min' + (pridicted_sales_price_max' -
pridicted_sales_price_min')*0.5;
num_iterative = 15000;
for n = 1 : num_iterative
    %% Optimize M
    [M_M, P_M, C_labor_M, pro_M] = pro_3_cvx_generate_M(Q(:,n),
Unit_cons, pridicted_material_price_max,...
pridicted_material_price_min, pridicted_sales_price_max,
pridicted_sales_price_min,...
pridicted_labor_cost_max, pridicted_labor_cost_min,
C_others,pridicted_Sales_max, pridicted_Sales_min);
    C_labor_M_new = C_labor_M;
    M(:,n+1) = M_M;
    %% Optimize Q
    [Q_Q, P_Q, C_labor_Q, pro_Q] = pro_3_cvx_generate_Q(M(:,n+1),
Unit_cons, pridicted_Sales_max, pridicted_Sales_min, ...

```

```

        prdicted_material_price_max, prdicted_material_price_min,
prdicted_labor_cost_max, prdicted_labor_cost_min,...
        C_others, prdicted_sales_price_max, prdicted_sales_price_min);
    pro(n+1)=real(pro_Q);
    Q(:,n+1) = Q_Q;
    %% stop Iteration
    if abs(pro(n+1)-pro(n))<100
        break;
    end
end
initial_data = pro*0.85;
% iterations = 10;
% new_x = linspace(1, 3, iterations);
% new_data = interp1(1:3, initial_data, new_x, 'linear');
% % 收敛图
% figure;
% plot([1:10], new_data/1000000, '-o', 'LineWidth', 2, 'MarkerSize', 6);
% title('迭代收敛图');
% xlabel('迭代次数');
% ylabel('最大利润（百万）');
% grid on;
% xticks(1:10);
% dates = datetime(2024, 1, 1):calmonths(1):datetime(2024, 9, 1);
% x_date = dates(1:9);
% y_pro = [ 1292670.16,1298495.94,3358353.99,3399002.42,-130334.36,...
%         4497524.04,5388877.34,8676074.13,pro(end)*0.85]/1000000;
% figure;
% hold on;
% plot(x_date, y_pro, '-o', 'LineWidth', 2, 'MarkerSize', 6);
% y_sep = y_pro(9);
% line(xlim, [y_sep, y_sep], 'Color', 'r', 'LineStyle', '--', 'LineWidth', 2);
% zero_profit = find(y_pro <= 0);

```

```

% for i = 1:length(zero_profit)
%     line(xlim, [y_pro(zero_profit(i)), y_pro(zero_profit(i))], 'Color', 'b',
'LineStyle', '--', 'LineWidth', 2);
% end
% title('销售净利润及预测');
% xlabel('时间（月份）');
% ylabel('销售净利润（百万元）');
% grid on;
y1 = [ 1292670.16,1298495.94,3358353.99,3399002.42,-130334.36,...
4497524.04,5388877.34,8676074.13]/1000000;
y2 =
[ 1270947.87462924,1330795.63641114,3406837.35708565,3286576.32346089,-
136228.146239139,...
4516984.93238931,5254917.11146267,8744244.04848553]/1000000;
dates = datetime(2024, 1, 1):calmonths(1):datetime(2024, 8, 1);
x_date = dates(1:8);
figure;
bar(x_date, y1, 'FaceColor', 'b');
hold on;
plot(x_date, y2, '-o', 'LineWidth', 3, 'MarkerSize', 6, 'Color', 'r');
title('实际销售净利润与模型预测利润');
xlabel('时间（月份）');
ylabel('销售净利润（百万元）');
legend('实际销售净利润', '模型预测利润');
grid on;
hold off;

```

B.6.3 销量预测走势图 Matlab 源码

```

clc;
clear;
filename = 'pridicted_Sales_Table.xlsx';
sheetName = 'pridicted_Sales_1';

```

```

pridict_si_data = readtable(filename, 'Sheet', sheetName);
dates = datetime(2024, 1, 1):calmonths(1):datetime(2024, 9, 1);
y_max = -inf(1, length(dates(3:9)));
y_min = inf(1, length(dates(3:9)));
for i = 1:50
    x = dates(3:9);
    y = pridict_si_data{i, 3:9};
    y_max = max(y_max, y);
    y_min = min(y_min, y);
end
n = length(x);
yellow = [1, 1, 0];
green = [0, 1, 0];
colors = [linspace(yellow(1), green(1), n); ...
          linspace(yellow(2), green(2), n); ...
          linspace(yellow(3), green(3), n)];
figure;
hold on;
for j = 1:n-1
    fill([x(j), x(j+1), x(j+1), x(j)], ...
         [y_max(j), y_max(j+1), y_min(j+1), y_min(j)], ...
         colors(j,:), 'EdgeColor', 'none');
end
h_max = plot(x, y_max, '-mp', 'LineWidth', 0.1);
h_min = plot(x, y_min, '--bs', 'LineWidth', 0.1);
si_data = readtable('si_sales.xlsx', 'Sheet', 's1_sales_volume');
x_si = dates(1:8);
y_si_data = si_data{:, 2};
h_actual_sales = plot(x_si, y_si_data, '-o', 'Color', 'r', 'LineWidth', 2);
legend([h_max, h_min, h_actual_sales], {'销量预测最大值', '销量预测最小值',
'实际销量'}, 'Location', 'northeast');
hold on;

```

```

grid on;
xlabel('日期');
ylabel('销量（千片）');
title('实际销量与销量预测波动走势（型号1）');
hold off;
max_min = [y_max;y_min];
predictedSalesTable = array2table(max_min);
excelFile = 'pridicted_max_min.xlsx';
sheetName = 'pridicted_Sales_1';
writetable(predictedSalesTable, excelFile, 'Sheet', sheetName, 'WriteMode',
'overwrite');

```

B.6.4 售价预测走势图 Matlab 源码

```

clc;
clear;
filename = 'pridicted_Sales_Table.xlsx';
sheetName = 'pridicted_Sales_price_1';
pridict_si_data = readtable(filename, 'Sheet', sheetName);
dates = datetime(2024, 1, 1):calmonths(1):datetime(2024, 9, 1);
y_max = -inf(1, length(dates(3:9)));
y_min = inf(1, length(dates(3:9)));
for i = 1:50
    x = dates(3:9);
    y = pridict_si_data{i, 3:9};
    y_max = max(y_max, y);
    y_min = min(y_min, y);
end
n = length(x);
yellow = [1, 1, 0];
green = [0, 1, 0];
colors = [linspace(yellow(1), green(1), n); ...
          linspace(yellow(2), green(2), n); ...

```

```

        linspace(yellow(3), green(3), n)]];

figure;
hold on;
for j = 1:n-1
    fill([x(j), x(j+1), x(j+1), x(j)], ...
        [y_max(j), y_max(j+1), y_min(j+1), y_min(j)], ...
        colors(j,:), 'EdgeColor', 'none');
end

h_max = plot(x, y_max, '-mp', 'LineWidth', 0.1);
h_min = plot(x, y_min, '--bs', 'LineWidth', 0.1);
si_data = readtable('si_sales.xlsx', 'Sheet', 'sl_sales_price');
x_si = dates(1:8);
y_si_data = si_data{:, 2};
h_actual_sales = plot(x_si, y_si_data, '-o', 'Color', 'r', 'LineWidth', 2);
legend([h_max, h_min, h_actual_sales], {'售价预测最大值', '售价预测最小值',
'实际售价'}, 'Location', 'northeast');
hold on;
grid on;
xlabel('日期');
ylabel('售价（元）');
title('实际售价与售价预测波动走势（型号1）');
hold off;
max_min = [y_max; y_min];
predictedSalesTable = array2table(max_min);
excelFile = 'pridicted_max_min.xlsx';
sheetName = 'pridicted_sales_price_1';
writetable(predictedSalesTable, excelFile, 'Sheet', sheetName, 'WriteMode',
'overwrite');

```

B.7 其余 Matlab 源码

B.7.1 交替优化 M 变量的 Matlab 源码

```
function [M, P, C_labor, pro] = pro_3_cvx_generate_M(Q, Unit_consu,
pridicted_material_price_max,...
    pridicted_material_price_min, pridicted_sales_price_max,
pridicted_sales_price_min,...
    pridicted_labor_cost_max, pridicted_labor_cost_min,
C_others,pridicted_Sales_max, pridicted_Sales_min)
% Maximizing profits
cvx_begin quiet
    variable M(4, 1)
    variable P(4, 1)
    variable C_labor(1,1)
    % Q 为月销量, M 为硅片单价, P 为单晶方棒的售价, VC 为总变动
    成本, C_labor 为人工成本
    VC = real( Unit_consu .* P' * 0.98637 * [760;450;880;825] );
    pro = real( Q'*M - VC - C_labor - C_others );
    maximize pro
    subject to
        pridicted_sales_price_min(1) <= M(1) <=
pridicted_sales_price_max(1);
        pridicted_sales_price_min(2) <= M(2) <=
pridicted_sales_price_max(2);
        pridicted_sales_price_min(3) <= M(3) <=
pridicted_sales_price_max(3);
        pridicted_sales_price_min(4) <= M(4) <=
pridicted_sales_price_max(4);
        pridicted_material_price_min(1) <= P(1) <=
pridicted_material_price_max(1);
```



```

        pridicted_material_price_min(2) <= P(2) <=
pridicted_material_price_max(2);
        pridicted_material_price_min(3) <= P(3) <=
pridicted_material_price_max(3);
        pridicted_material_price_min(4) <= P(4) <=
pridicted_material_price_max(4);
        pridicted_labor_cost_min <= C_labor <= pridicted_labor_cost_max;
        for i = 1:4
            M(i) <=    pridicted_sales_price_max(i) - ...
                        (pridicted_sales_price_max(i) - pridicted_sales_price_min(i))
* ...
                        ((Q(i) - pridicted_Sales_min(i)) / (pridicted_Sales_max(i) -
pridicted_Sales_min(i)));
            M(i) >=    pridicted_sales_price_max(i) - ...
                        (pridicted_sales_price_max(i) - pridicted_sales_price_min(i))
* ...
                        ((Q(i) - pridicted_Sales_min(i)) / (pridicted_Sales_max(i) -
pridicted_Sales_min(i)));
        end
    cvx_end
end

```

B.7.2 交替优化 Q 变量的 Matlab 源码

```

function [Q, P, C_labor, pro] = pro_3_cvx_generate_Q(M, Unit_consu,
pridicted_Sales_max, pridicted_Sales_min, ...
    pridicted_material_price_max, pridicted_material_price_min,
pridicted_labor_cost_max, pridicted_labor_cost_min,...
    C_others, pridicted_sales_price_max, pridicted_sales_price_min)
% Maximizing profits
cvx_begin quiet
    variable Q(4, 1)
    variable P(4, 1)

```

```

variable C_labor(1,1)
% Q 为月销量，M 为硅片单价，P 为单晶方棒的售价，VC 为总变动
成本，C_labor 为人工成本
VC = real( Unit_cons * P' * 0.98637 * [760;450;880;825]);
pro = real( Q'*M - VC - C_labor - C_others );
maximize pro
subject to
    prdicted_Sales_min(1) <= Q(1) <= prdicted_Sales_max(1);
    prdicted_Sales_min(2) <= Q(2) <= prdicted_Sales_max(2);
    prdicted_Sales_min(3) <= Q(3) <= prdicted_Sales_max(3);
    prdicted_Sales_min(4) <= Q(4) <= prdicted_Sales_max(4);
    prdicted_material_price_min(1) <= P(1) <=
prdicted_material_price_max(1);
    prdicted_material_price_min(2) <= P(2) <=
prdicted_material_price_max(2);
    prdicted_material_price_min(3) <= P(3) <=
prdicted_material_price_max(3);
    prdicted_material_price_min(4) <= P(4) <=
prdicted_material_price_max(4);
    prdicted_labor_cost_min <= C_labor <= prdicted_labor_cost_max;
    for i = 1:4
        Q(i) <= prdicted_Sales_max(i) - ...
            (prdicted_Sales_max(i) - prdicted_Sales_min(i)) * ...
            ((M(i) - prdicted_sales_price_min(i)) /
(prdicted_sales_price_max(i) - prdicted_sales_price_min(i)));
        Q(i) >= prdicted_Sales_max(i) - ...
            (prdicted_Sales_max(i) - prdicted_Sales_min(i)) * ...
            ((M(i) - prdicted_sales_price_min(i)) /
(prdicted_sales_price_max(i) - prdicted_sales_price_min(i)));
    end
cvx_end
end

```