

队伍编号	MC2300777
题号	C

物流网络货量调度问题的优化与探究

摘要

物流网络受电商活动及各种突发情况影响较大。本文的目的是建立物流网络调度模型，以帮助运营者做好应对突发情况的准备，并提高运营效率。我们期望在各种客观条件的约束下，为物流网络的应急调度提供有效的方案。

本文建立了五个主要模型：模型 I：ARIMA 时间序列预测模型；模型 II：基于 Informer 的长序列时间序列预测模型；模型 III：基于免疫遗传算法多目标优化模型；模型 IV：物流线路动态规划模型；模型 V：基于熵权 TOPSIS 分析的线路综合评价模型。

针对问题 1，首先对 2021-2022 年物流网络数据进行分类整合与可视化处理，初步得出其为非平稳变化，因而进行两次差分后建立时间序列自回归模型并对结果进行白噪声检验；接着引入基于 Informer 的长序列时间序列预测模型，以提高模型预测的精确程度；最后通过机器学习训练得出三条线路预测结果，完整结果见表 2。

针对问题 2，为使 DC5 关停后货量发生变化的线路尽可能小，保持线路的工作负荷尽可能均衡，我们构建了多目标优化模型，求出所有线路分配前后货量差值平方的最小值，并利用启发式算法——免疫遗传算法对该模型进行求解。求解得出 69->8 工作负荷达到百分之 100，剩余货件超过 15000 件，因关停受到的影响最大。完整结果见表 3。

针对问题 3，基于问题 2 中的多目标优化模型，对物流网络进行动态调整，建立物流分配动态规划模型，并利用免疫遗传算法进行求解。求解得出路线 27->10 负荷率为 100%，其他线路平均负荷量为 10.34%。具体流转货物量及工作负荷见表 4 和表 5。

针对问题 4，对物流场地和物流线路分别进行评价，分析物流场地的处理能力和物流线路的运输能力对整个物流网络的影响。我们首先运用 Topsis 模型进行物流网络性能的综合评价，用熵权法分别求出路线和场地指标对应的权重，其中货量对应权重为 0.38944，在路线指标中占比最大。具体权重结果见表 6 和表 7。最后得出在场地 10、14 和场地 14、8 之间设置路线能有效改善网络性能。

最后，对模型进行鲁棒性分析。本文的预测模型在具有轻微扰动的原始输入上的预测结果不会显著偏离该原始输入，说明该模型在输入样本存在细微对抗扰动的情况下，模型预测具有不受对抗样本干扰误导的能力。

关键词： ARIMA；Informer；免疫遗传算法；动态规划；熵权 TOPSIS

目 录

第 1 章 问题重述.....	1
1.1 问题背景	1
1.2 问题描述	1
第 2 章 思路分析.....	2
2.1 问题 1 的思路分析	2
2.2 问题 2 的思路分析	2
2.3 问题 3 的思路分析	2
2.4 问题 4 的思路分析	2
第 3 章 符号说明与基本假设.....	3
3.1 符号说明	3
3.2 基本假设	3
第 4 章 模型的准备.....	4
4.1 数据预处理	4
4.1.1 数据的筛选	4
4.1.2 数据可视化	4
4.2 我们的工作	5
第 5 章 问题 1 的模型建立与求解.....	6
5.1 传统预测模型分析和选择	6
5.1.1 时间序列自回归预测模型	6
5.1.2 模型的检验	7
5.2 基于 Informer 的长序列时间序列预测改进	8
5.3 问题 1 的预测结果及分析	9
5.3.1 预测结果	9
5.3.2 结果分析	10
第 6 章 问题 2 的模型建立与求解.....	11

6.1 基于免疫遗传算法多目标优化模型	11
6.1.1 多目标优化模型的建立	11
6.1.2 免疫遗传算法结构层次图	13
6.2 基于免疫遗传算法的求解过程	13
6.2.1 免疫遗传算法求解流程	14
6.2.2 免疫遗传算法流程图	15
6.3 问题 2 的求解结果	16
第 7 章 问题 3 的模型建立与求解	17
7.1 物流线路动态规划模型建立与求解	17
7.2 问题 3 的求解结果	19
第 8 章 问题 4 的模型建立与求解	21
8.1 基于熵权 TOPSIS 分析的线路综合评价模型建立	21
8.2 TOPSIS 流程图	22
8.3 问题 4 的求解结果	23
第 9 章 模型的测试	25
9.1 鲁棒性分析	25
第 10 章 模型的评价与总结	25
9.1 模型的优点	25
9.2 可能的改进	26
第 11 章 参考文献	26
附录	27

第 1 章 问题重述

1.1 问题背景

近年来，我国电子商务行业发展迅猛，包裹物流成为企业关心的一大问题。受电商活动日影响，电商用户的下单量会发生显著波动，而各种自然人为自然灾害导致物流场地临时或永久停用时，会影响到各个物流线路。

针对此问题，我们需要能预测出各物流场地及线路的包裹数量，以提前安排运输、分拣等计划，从而降低运营成本，提高运营效率。在某些物流场地因各种因素停用时，基于预测结果和各个物流场地的处理能力及线路的运输能力，设计物流网络调整方案，以保障物流网络的正常运行。

1.2 问题描述

问题 1：建立线路货量的预测模型，对 2023 年 1 月 1 日至 2023 年 1 月 31 日期间每条线路每天的货量进行预测，并对线路 DC14→DC10、DC20→DC35、DC25→DC62 的货物量进行预测。

问题 2：假设物流场地 DC5 于 2023 年 1 月 1 日开始关停，建立相关模型将资源分配到其他线路，使得各线路工作负荷尽可能均衡，并使得 1 月 1 日至 1 月 31 日期间未正常流转的包裹尽可能少。受 DC5 影响，正常和非正常流转时分别给出正常流转和非正常流转的货量及网络负荷情况。

问题 3：假设被关停的物流场地为 DC9，同时允许对物流网络结构进行动态调整，调整措施为关闭或新开线路，将 DC9 相关线路的货量分配到其他线路，使所有包裹尽可能正常流转，并使得 DC9 关停前后货量发生变化的线路数尽可能少，分别给出正常流转时和非正常流转时，因 DC9 关停导致货量发生变化的线路数、不能正常流转的货量及网络的负荷情况以及每天的线路增减情况。

问题 4：根据附件 1，对该网络的不同物流场地及线路的重要性进行评价，结合问题 1 的预测结果，探讨分析新增物流场地对应的新增线路，新增物流场地的处理能力及新增线路的运输能力的设置情况，并探讨所建网络的鲁棒性。

第 2 章 思路分析

2.1 问题 1 的思路分析

问题一要求我们建立线路货量预测模型，对给定的线路的货量进行预测，针对此问题，根据查阅文献我们可以知道通常采用时间序列预测模型，而近年来使用时间序列的函数型变系数自回归的预测方法对货量进行研究^[1]已成为一大热点。首先我们对 2021-2022 数据进行预处理，剔除异常数据并将其进行可视化处理，整理出每条物流线路的具体情况。然后，我们建立时间序列自回归模型，在此基础上基于 Informer 的长序列时间序列预测^{[2][3]}改进模型，并对题中所给数据选择拟合程度最优的方法进行机器学习训练，最终得出预测结果。

2.2 问题 2 的思路分析

问题二要求我们在物流场地 DC5 关停的情况下，进行合理分配，使各线路工作均衡。针对此问题，通常根据线路的使用频率以及最大最小承载量以表示线路停运时网络的负荷情况，此外需要考虑 DC5 所处的位置，分析 DC5 连接的其他地区与运输的货物量大小，判断与 DC5 链接的其他地区是否有额外的路线。我们利用免疫遗传算法建立多目标优化模型^[4]。首先建立多目标优化模型，构建目标函数，使得 DC5 关停前后货量达到最小值，然后将分配的每条线路总货量限制在一个稳定范围内，建立约束条件，最后利用启发式算法——免疫遗传优化算法进行求解，得出各线路负荷情况。

2.3 问题 3 的思路分析

问题三要求我们对物流网络结构进行动态调整，针对问题二被关停的物流场地为 DC9 的情况，问题三允许对物流网络结构进行动态调整，即建立一个动态调整^[5]的货物业数学模型。该模型需要每天进行一次决策，以确定需要关闭或新增哪些线路。同时，需将 DC9 相关路线的货量分配到其他线路，使所有包裹尽可能正常流转，并使 DC9 关停前后货量发生变化的路线数尽可能少，且保持各条线路的工作负荷尽可能均衡。

2.4 问题 4 的思路分析

问题四要求我们对不同物流场地的重要性进行评价，我们利用熵权 TOPSIS 分析建立线路和场地运输能力的综合评价模型^[6]。给各个场地和路线进行打分排序。最后对新增物流场地及运输路线能力进行分析，并对模型的鲁棒性进行分析。

第 3 章 符号说明与基本假设

3.1 符号说明

表 1：文章中使用的符号及说明

符号	具体说明
y_{ij}	线路 i 到 j 总货物数量
x_{ij}	线路 i 到 j 上增加的 DC5 相关线路的货物
w_{ij}	第 i 到 j 线路的最大包裹量
c_i	DC5 相关线路的总货物量
L_{ij}	所有线路上的物流容量
u_{ij}	表示从 i 到 j 是否开通新的路线
g_{ij}	表示从 i 到 j 是否关闭线路
z_{ijt}	第 t 天新增线路 i 到 j 总货物数量
x_{ijt}	第 t 天线路 i 到 j 上增加的 DC9 相关线路的货物
y_{ij}	第 t 天线路 i 到 j 总货物数量
x'_{ij}	第 i 个样本在第 j 个评价指标上的数值
D_i^-	每级指标与最劣向量的欧式距离
D_i^+	每级指标与最优向量的欧式距离
W_i	每级指标与最优值的接近程度
W	每个指标的权重

3.2 基本假设

为了简化问题，我们做出以下基本假设，并且每一个假设都是合理的。

- ✓ **假设 1：被分配的物流运输线路的能力上限均为其历史货量的最大值。**

理由：因为如果出现超过历史峰值的货量，物流场地和线路可能会出现瓶颈，需要及时调整货量分配方案，以保障网络的正常运转。

- ✓ **假设 2：附件 1 中各物流场地之间运输方式、运输时间及运输成本在本文研究的时间段内不会出现明显的变化。**

理由：在模型的求解中，要保持各物流场地之间运输方式、时间及成本不变，才能

在综合评价模型寻出最优解。

✓ 假设 3：假设本文研究中使用的数据是准确的。

理由：我们假设附件 1 中的数据经过预处理后，与实际情况没有出现明显的测量偏差而被认为是虚假的，因此可以在此基础上建立更合理的量化模型。

第 4 章 模型的准备

4.1 数据预处理

4.1.1 数据的筛选

由于附件 1 给出的数据，为了提高模型的精度，我们对数据进行预处理，将空缺数据补 0 处理，并筛选出流转货物数较大的线路，进行分析。

4.1.2 数据可视化

为了更直观的展示附件 1 所给出的数据，方便对问题进行分析，将 2021 年 1 月至 2022 年 12 月货量总体变换趋势及线路指向情况用如下图 1 图 2 所示：

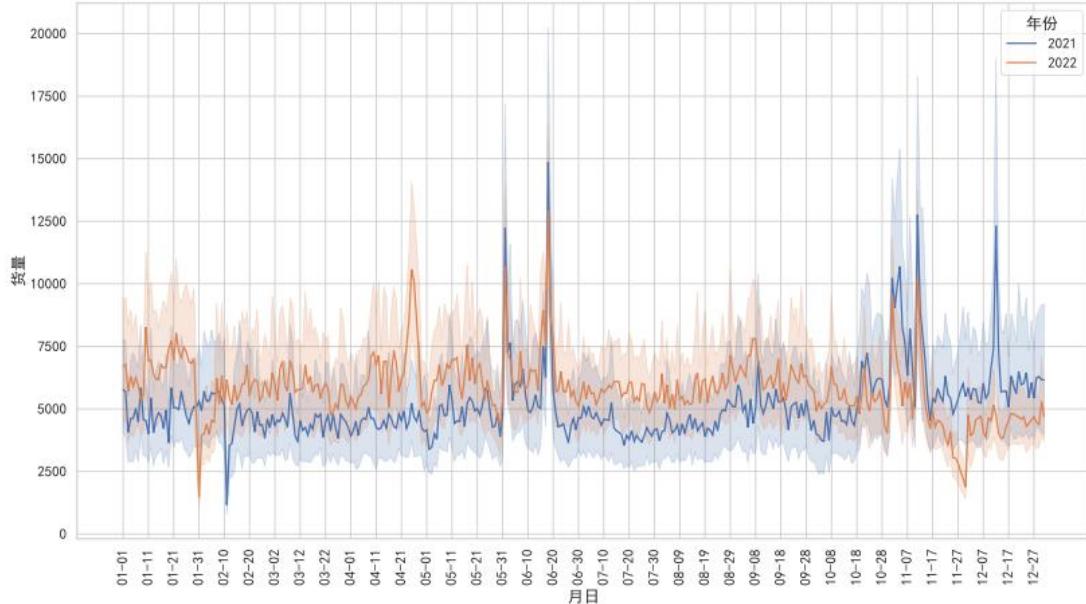
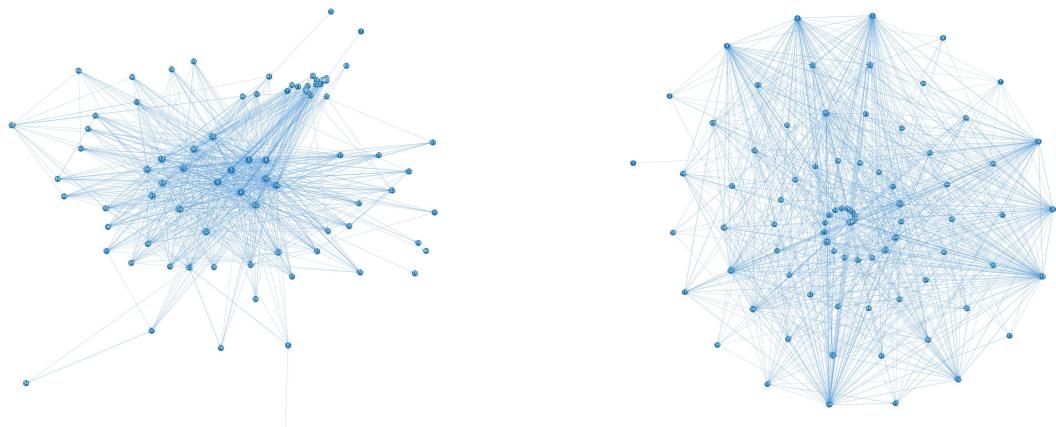


图 1 2021-2022 货量总体变化情况



(a) Kamada 布局的有向图

(b) 螺旋布局有向图

图 2 有向图展示概览

4.2 我们的工作

本文需要我们根据附件 1 数据，建立预测模型对线路货量进行预测，同时建立数学模型分析物流场地关停对其他线路网络负荷情况的影响。首先我们建立时间序列自回归预测模型并基于 Informer 长序列时间序列预测改进；然后，利用启发式算法免疫遗传算法建立多目标优化模型及动态规划模型；最后基于熵权 TOPSIS 分析建立综合评价模型，对新增线路运输能力进行评估。综上所述，整个建模过程可以表示成如下图 3 所示：

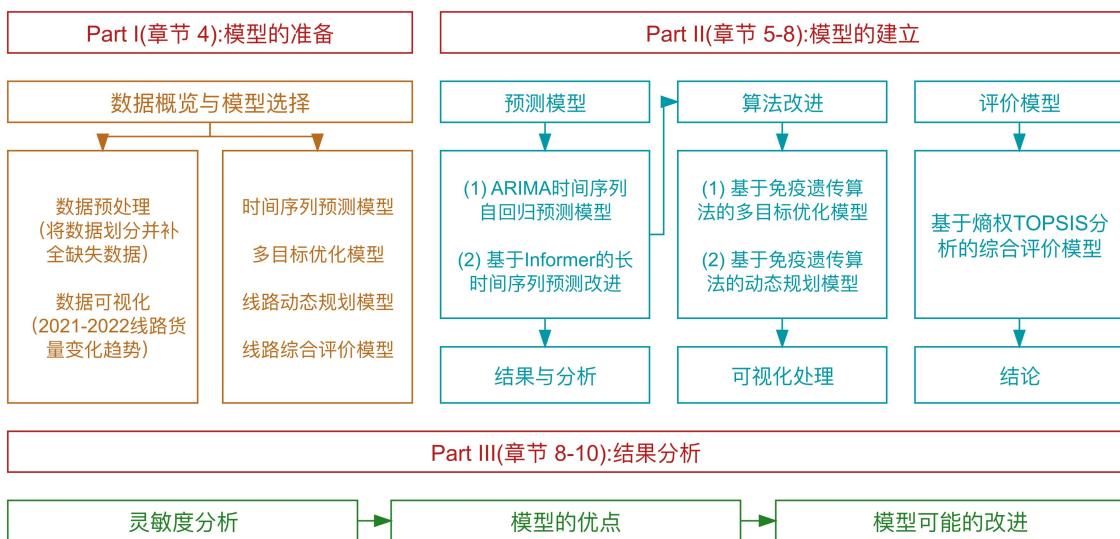


图 3 我们的工作流程图

第 5 章 问题 1 的模型建立与求解

5.1 传统预测模型分析和选择

5.1.1 时间序列自回归预测模型

Step1: 物流网络货量序列的平稳化检验和差分运算

我们用物流网络历史货量数据序列，对时间序列进行观察，发现时间序列存在很明显的增长趋势，这表明时间序列存在异方差。为了减少异方差的影响，对时间序列取自然对数，得到新的数列{TERRT}，然后采用 ADF(Augmented Dickey-Fuller)方法^[7]进行序列的单位根检验。经检验，{TERRT}这个序列在 5% 的显著性水平是非平稳时间序列，则需进行进一步的一阶差分和二阶差分。

Step2: 利用变量 D(TERRT, 2) 对 p 和 q 进行分析

由于参数的确定取决于 D(TERRT, 2) 数列的自回归函数(ACF)和偏自回归函数(PACF)，根据自相关系数的最大滞后阶数确定原则，初步判定最大滞后阶数 K=5。序列的样本自相关系数成衰减正弦波并趋向于零，表现为拖尾衰减。而在 PAC 图中，当 K=2 和 k=4 时，偏相关系数明显不为 0，在 k>5 以后的值都在随机区间以内，可以认为序列的偏自相关函数具有截尾性。分析得出模型为 ARIMA(5, 2, 2)。本文选取使 AIC 到达最小的那一组阶数为理想阶数。通过分析，ARIMA(5, 2, 3) 模型中的 AIC 值最小，因此选定 ARIMA(5, 2, 3) 模型作为预测货运量的模型。具体分析图如图 4、5、6 所示

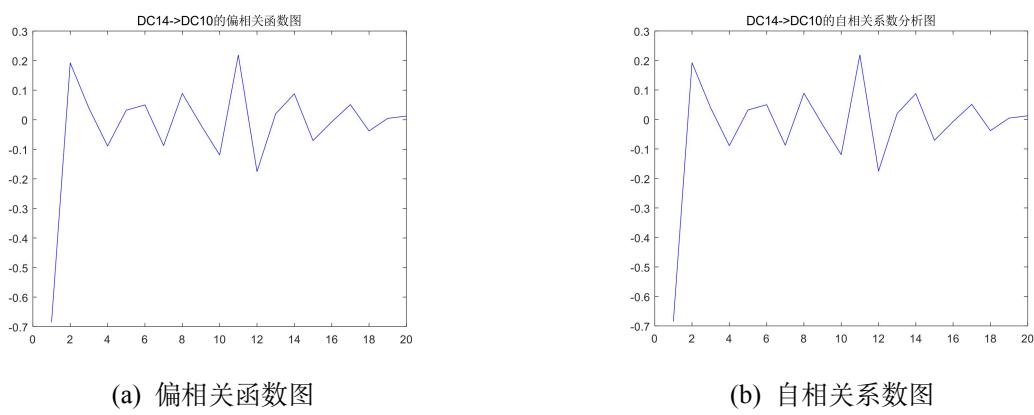
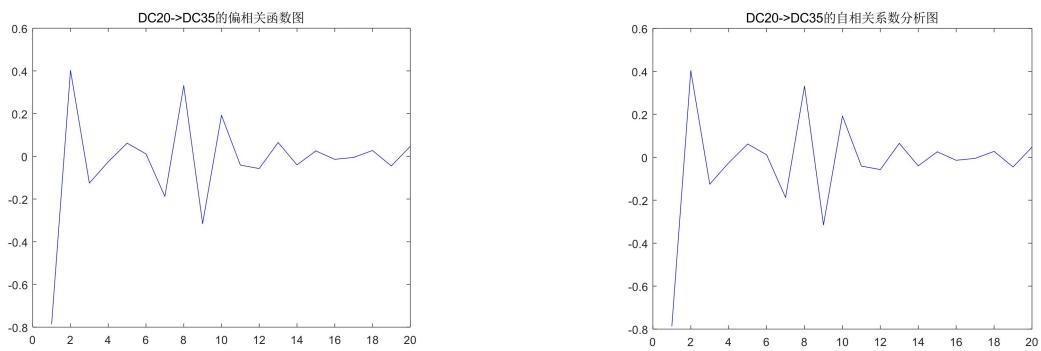


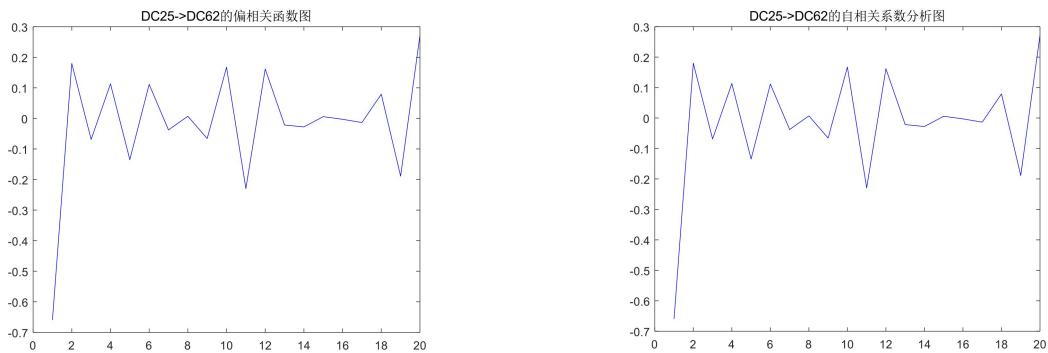
图 4 DC14 —>DC10 相关分析图



(a) 偏相关函数图

(b) 自相关系数图

图 5 DC14 → DC10 相关分析图



(a) 偏相关函数图

(b) 自相关系数图

图 6 DC25 → DC62 相关分析图

5.1.2 模型的检验

由于时间序列自回归模型有随机性，因此我们对模型的残差进行白噪声检验^[6]，对残差进行进行自相关和偏自相关分析。检验结果如图 7 所示：

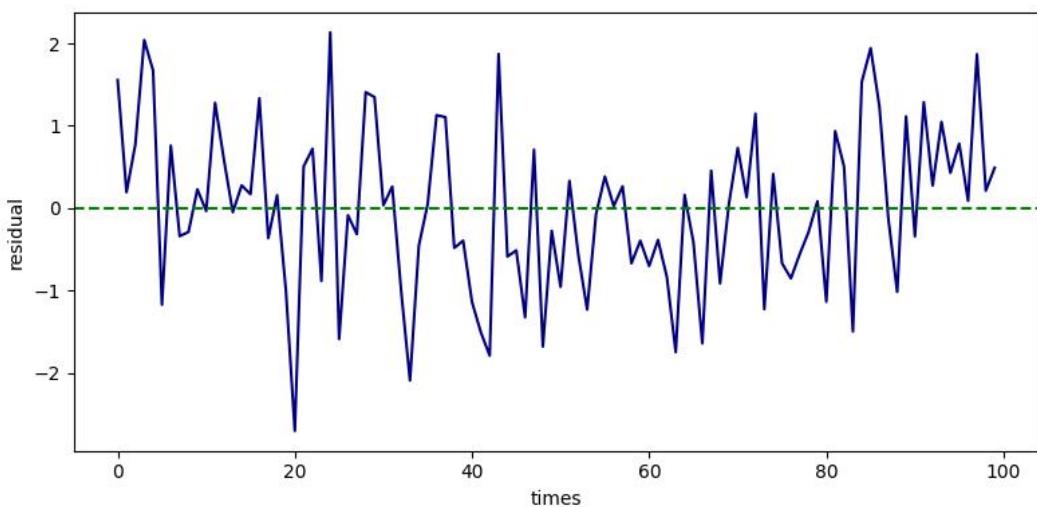


图 7 白噪声检验

5.2 基于 Informer 的长序列时间序列预测改进

由于时间序列自回归模型的先天性缺陷，随着预测期的延长，其预测结果的精度会降低，本文采用 Informer 的长序列时间序列预测对模型进行改进。其中 Informer 的整体框架如图 8 所示：

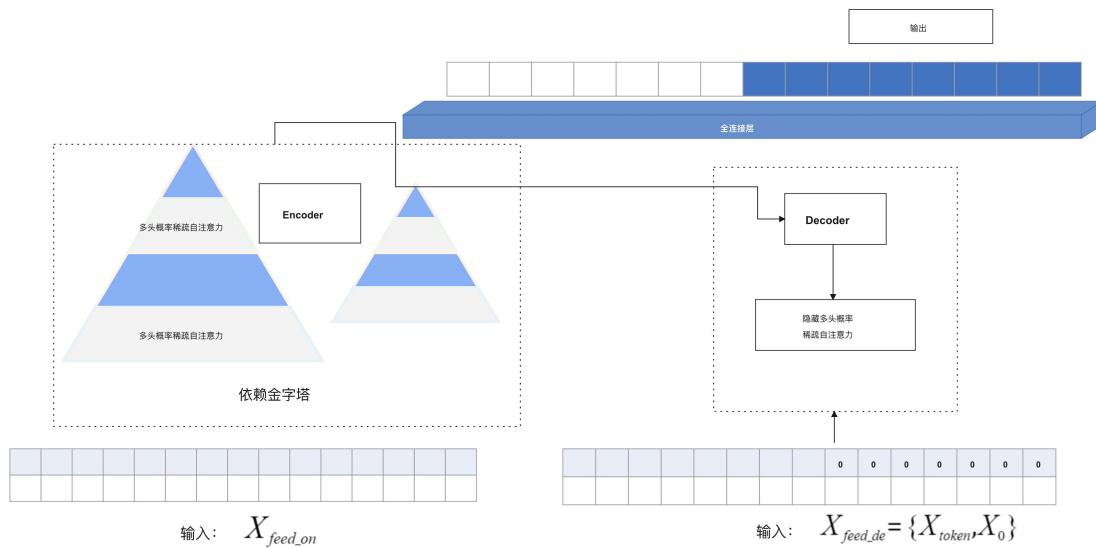


图 8 Informer 整体框架

利用 Informer 长序列时间序列预测改进后的与时间序列自回归预测对比如图 9 所示：

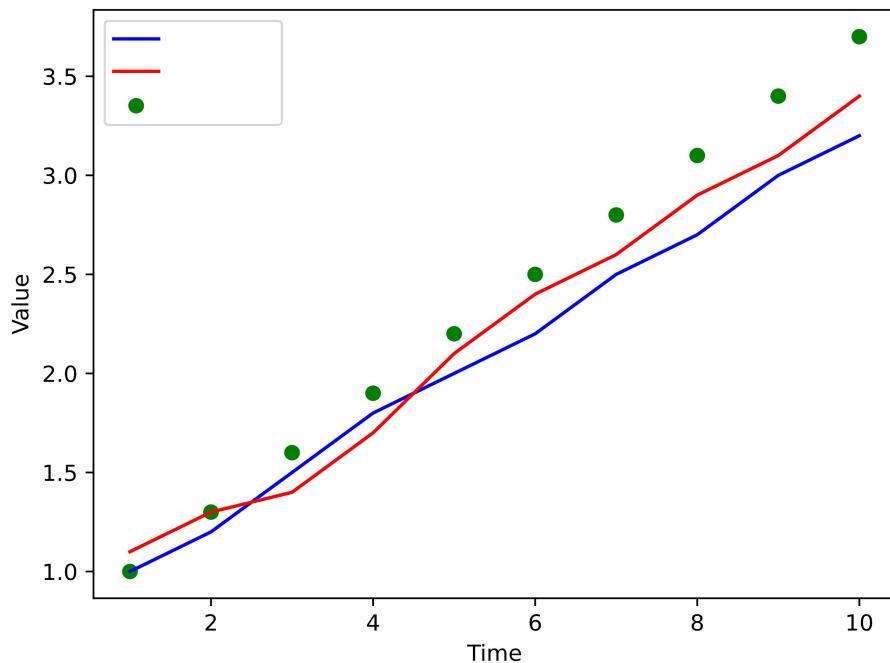


图 9 时间序列自回归与 Informer 对比图

5.3 问题 1 的预测结果及分析

5.3.1 预测结果

DC14—>DC10、DC20—>DC35、DC25—>DC62 三条路线预测结果如表 2 所示：

表 2 三条路线预测结果

时间	DC14—>DC10	DC20—>DC35	DC25—>DC62
2023 年 1 月 1 日	28485.93	38	6320
2023 年 1 月 2 日	28556.6	48.5	9932
2023 年 1 月 3 日	28051	74	8185.667
2023 年 1 月 4 日	27076.71	64.75	9586
2023 年 1 月 5 日	26116.67	74	10552.2
2023 年 1 月 6 日	26462.47	64.33	11059.33
2023 年 1 月 7 日	27406.45	57.29	10122
2023 年 1 月 8 日	27983.71	53	9391.625
2023 年 1 月 9 日	28834.32	70.11	9991.444
2023 年 1 月 10 日	28565.7	65.6	9570.1
2023 年 1 月 11 日	28209.79	62.36	9675.636
2023 年 1 月 12 日	28196.8	58.08	9220.833
2023 年 1 月 13 日	28939.81	69.77	8876.154
2023 年 1 月 14 日	28568.15	65.36	8405.643
2023 年 1 月 15 日	28198	61.33	8889.4
2023 年 1 月 16 日	27769.17	57.69	8630.75
2023 年 1 月 17 日	27493	55.24	8449.765
2023 年 1 月 18 日	26715.94	54.61	8256.222
2023 年 1 月 19 日	28485.93	53.26	8683.474
2023 年 1 月 20 日	28556.6	51.75	8882.3
2023 年 1 月 21 日	28051	57.24	8742.048
2023 年 1 月 22 日	27076.71	54.95	8610.682
2023 年 1 月 23 日	26116.67	67.57	8837.696
2023 年 1 月 24 日	26462.47	65.92	8755.75
2023 年 1 月 25 日	27406.45	63.64	8841.2

2023 年 1 月 26 日	27983.71	62.23	8940.5
2023 年 1 月 27 日	28834.32	62.7	8926.926
2023 年 1 月 28 日	28565.7	61.36	8745.929
2023 年 1 月 29 日	28209.79	59.55	8992.379
2023 年 1 月 30 日	28196.8	58.33	8813.967
2023 年 1 月 31 日	28939.81	56.87	8882.806

5.3.2 结果分析

经过两次差分后两次差分后的序列图和原数情况如图 10 所示:

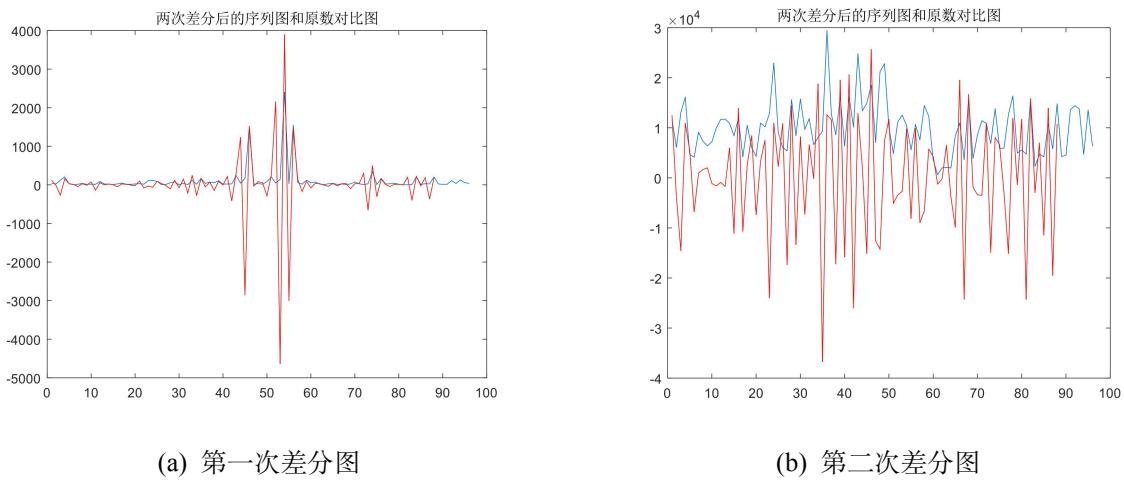


图 10 两次差分后的序列图和原数对比图

DC14 → DC10 相关预测分析图如图 11 所示:

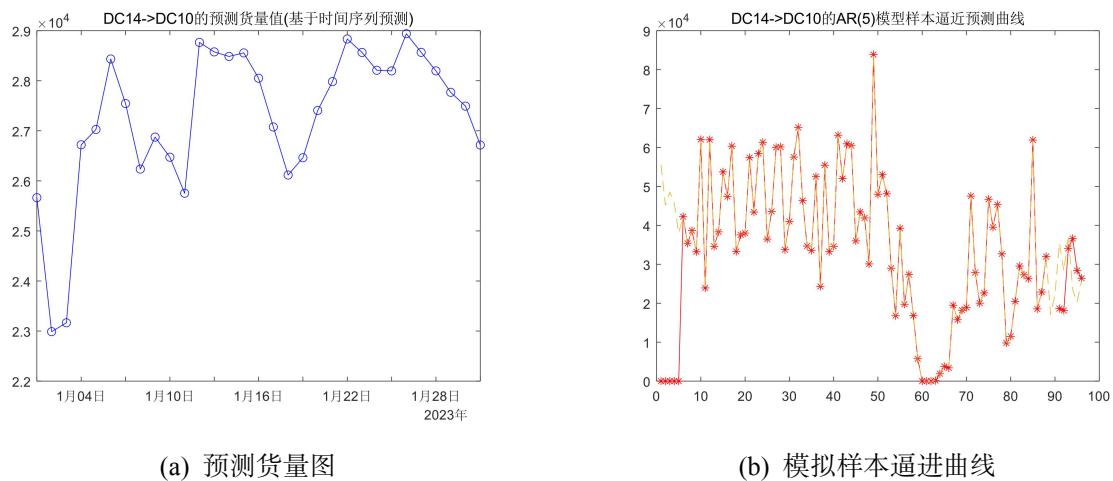
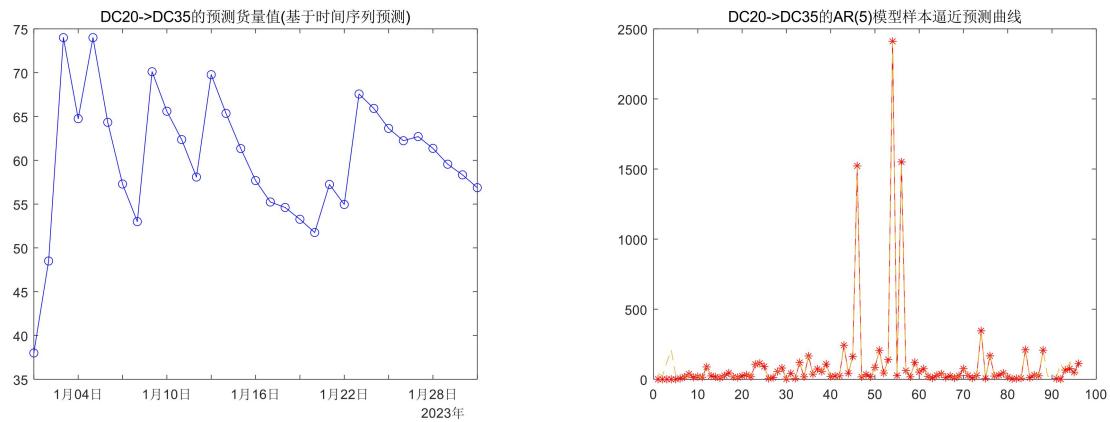


图 11 DC14 → DC10 相关预测分析图

DC20 → DC35 相关预测分析图如图 12 所示:

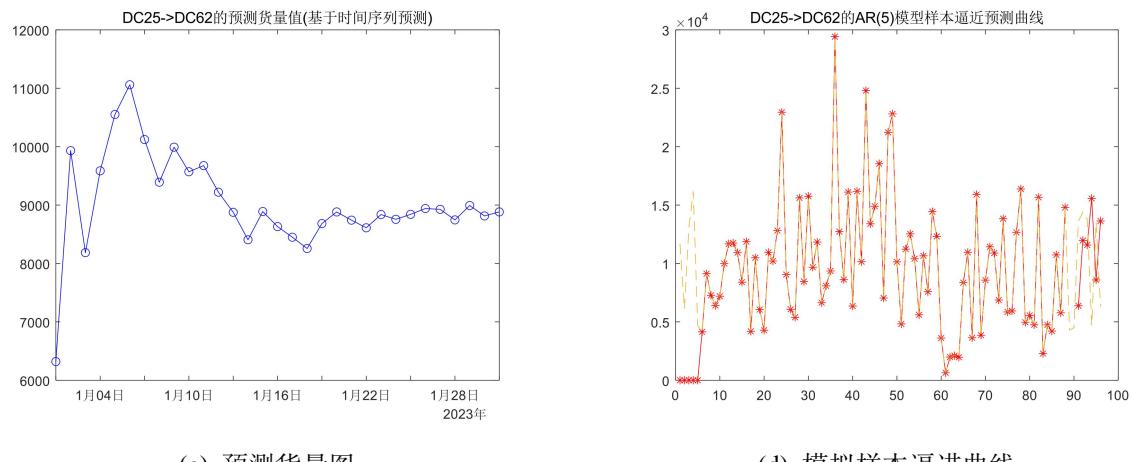


(a) 预测货量图

(b) 模拟样本逼近曲线

图 12 DC14 → DC10 相关预测分析图

DC25 → DC62 相关预测分析图如图 13 所示:



(c) 预测货量图

(d) 模拟样本逼近曲线

图 13 DC25 → DC62 相关预测分析图

第 6 章 问题 2 的模型建立与求解

6.1 基于免疫遗传算法多目标优化模型

6.1.1 多目标优化模型的建立

根据问题 2 的相关要求, 建立线性规划模型来优化货量分配, 并通过免疫遗传算法进行求解, 使得所有包裹尽可能正常流转, 同时减少货量发生变化的线路数, 并保持各条线路的工作负荷尽可能均衡。具体建模步骤如下:

Step1: 确定决策变量

假设 DC5 关停后, DC5 相关线路的货物需要分配到其他线路, 其中 DC5 相关线路为 $i=1,2,L,n$, $j=1,2,L,n$, 线路 i 到 j 总货量数量为 y_{ij} , 线路 i 到 j 上增加的 DC5 相关线路

的货物为 x_{ij} , w_{ij} 则表示第 i 到 j 线路的最大包裹量, 当第 i 到 j 条线路发生改变, 即 x_{ij} 大于 1 时, $\text{count}(x_{ij})$ 为 1, 否则为 0.

Step2: 确定目标函数

设定 Y 为所有线路分配前后货物总量的差值的平方, 即

$$Y = \sum_{j=1}^n [w_{ij} - (x_{ij} + y_{ij})]^2 \quad (1)$$

将 DC5 相关线路的货量分配到其他线路后, 为保证 DC5 关闭前后货量变化的线路尽可能少, 且保持各条线路的工作负荷尽可能均衡, 需要将每条线路的总货物数量限制在一个相对稳定的范围内, 即在保证所有线路的工作负荷尽量相近的情况下最小化分配前后货物总量的差值, 即

$$\min Y \quad (2)$$

$$\min \sum_{j=1}^n \text{count}(x_{ij}) \quad (3)$$

Step3: 确定约束条件

为保持各条线路的工作负荷尽可能均衡, 需要将每条线路的总货物数量限制在一个相对稳定的范围内, 即所有线路的工作负荷尽量相近。因此, 需要添加以下约束条件:
所有线路上的货物数量必须满足物流容量限制, 即

$$0 \leq \sum_{i=1}^n \sum_{j=1}^n (x_{ij} + y_{ij}) \leq L_{ij} \quad (3)$$

分配后各个线路上的货物数量不超过其容量, 即

$$x_{ij} + y_{ij} \leq w_{ij}, \quad (4)$$

DC5 相关线路上货物数量分配到其他线路上的总量等于 DC5 相关线路上的货物量:

$$\sum_{i=1}^n x_{ij} = c_i \quad (5)$$

每个时间段每个线路的货物数量为非负整数

$$x_{ij} + y_{ij} \in Z \quad (6)$$

综合以上分析得到的多目标优化模型总的表述如下:

$$\begin{aligned} & \min Y \\ & \min \sum_{j=1}^n \text{count}(x_{ij}) \end{aligned} \quad (7)$$

$$s.t. \left\{ \begin{array}{l} x_{ij} + y_{ij} \leq w_{ij} \\ 0 \leq \sum_{i=1}^n \sum_{j=1}^n x_{ij} + y_{ij} \leq L_{ij} \\ \sum_{i=1}^n x_{ij} = c_i \\ x_{ij} + y_{ij} \in Z \end{array} \right. \quad (8)$$

6.1.2 免疫遗传算法结构层次图

免疫遗传算法结构层次图如图 14 所示：

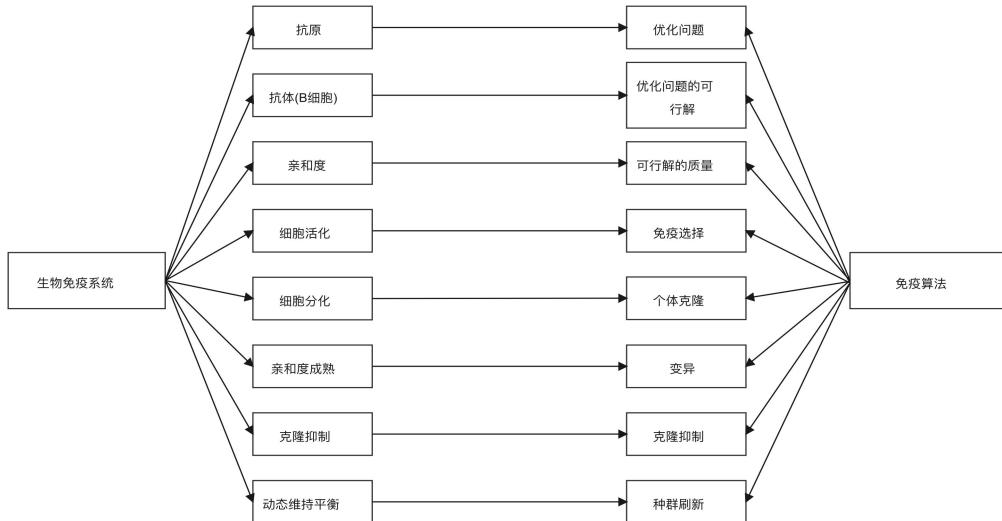


图 14 免疫遗传算法与生物免疫系统概念关系

6.2 基于免疫遗传算法的求解过程

针对此题数据量大，模型变量维数多，变量空间大的特点，采用启发式算法进行求解。传统的遗传算法具有收敛速度较慢，局部搜索能力较弱的缺点，并且运行时间长，容易受参数的影响。而人工免疫算法模拟免疫系统使其具有自我调节能力[8]，以及基于抗体浓度的产生和维持多样性抗体的能力。基于此，我们采用免疫遗传算法，将遗传算法和免疫算法的优点结合了起来，通过保存最优个体记忆信息，从而加速算法的局部搜索能力，使得进化计算的收敛速度更快，效率更高。

6.2.1 免疫遗传算法求解流程

Step1: 准备

(1) 抗体表示: 采用实数编码 $(Q_1 \ Q_2 \ \dots \ Q_n)$, 总 $Q_O = \sum_{k=1}^N Q_k$

其中 Q_k 表示第 k 个物流场地的货物量 ($k = 1, 2, \dots, N$)。 Q_O 表示总货物量。

与二进制相比, 实数编码在数值优化方面具有更高的精度和效率, 并且搜索范围更大。

(2) 适应度 f : 取 $f = Y$

(3) 初始化参数: 抗体规模 N , 记忆种群比率 M , 浓度抑制半径 r , 浓度阈值 T , 遗传算子 (交叉率 p_c , 变异率 p_m , 突变率 p_{m2})

(4) 终止条件: 最大迭代代数 $MAXGEN$

Step2: 初始抗体群生成

在可行域中, 随机产生一个抗体, 应用随机模拟, 检验满足率的要求, 若不满足要求, 重新产生一个抗体。直到产生可行的符合满足率的 N 个抗体, 形成初始抗体群 $(POP_1, POP_2, \dots, POP_N)$ 。

Step3: 初始抗体群生成

抗体 v 和抗原亲和力 A_v 是指抗体与抗原的匹配度, 由目标函数得来。这里采用 $f + 1$ 的倒数。

$$A_v = \frac{1}{1+f} \quad (9)$$

抗体 u 与抗体 v 之间的亲和力 B_{uv} 反映了抗体之间的相似程度, 本文采用基于 *Euclidean* 距离的亲和力计算方法:

$$B_{uv} = \|u - v\| = \sqrt{\sum_{j=1}^N (u_j - v_j)^2} \quad (10)$$

两个抗体间的距离在浓度抑制半径 r 内, 表示这两个抗体相似程度较高。

Step4: 抗体浓度计算

抗体浓度 C_v 采用下面公式:

$$C_v = \frac{\sum_{u \neq v} S_{uv}}{N} \quad (11)$$

其中 N 表示抗体个数,

$$S_{uv} = \begin{cases} 1 & \text{若 } \|u - v\| \leq r \\ 0 & \text{其它} \end{cases} \quad (12)$$

在计算抗体浓度时，相似程度比较高的抗体看作是一种抗体。

Step5: 抗体记忆与抑制

在运算过程中，当一种抗体的浓度超过设定的阈值 T 时，表明抗体在种群中占据了较大优势，达到了一个相对最优点，此时生成一个记忆细胞，以记录此时的局部最优解。每次将亲和度高的部分抗体复制进记忆种群中去，并在每次的迭代中通过记忆种群更新整个种群。

Step6: 遗传

(1) 选择

抗体的选择将抗体群中较好的候选解选择后参与进化。根据适应度公式，得到各个抗体的适应度。按适应度排序，找出抗体群中最大的适应度 $\text{Max } f$, 定义抗体 v 的浓度， α 、 β 为 0-1 之间的可调参数，可以取一个固定值，可以实现自适应度变化。此模型中取 $\alpha = 0.2$, $\beta = 1 - \alpha$ 。采用轮盘赌的方式，选择 N 次，复制 N 个个体到匹配池中等待交叉操作。

(2) 交叉

采用顺序算术交叉。根据题目提示 DC5 关停可能存在不能正常流转的情况，这里交叉算子可能搜索不到靠近边缘的解，应依靠变异。

(3) 变异

确定变异个体，随机产生变异方向 $d = (d_1, d_2, \dots, d_N)$, 其中 d_i 是 Q_i 的允许变化量

Step7: 重复 3-6 步，直到满足终止条件，记忆池中的解为最优解。

6.2.2 免疫遗传算法流程图

免疫遗传算法流程图如图 15 所示：

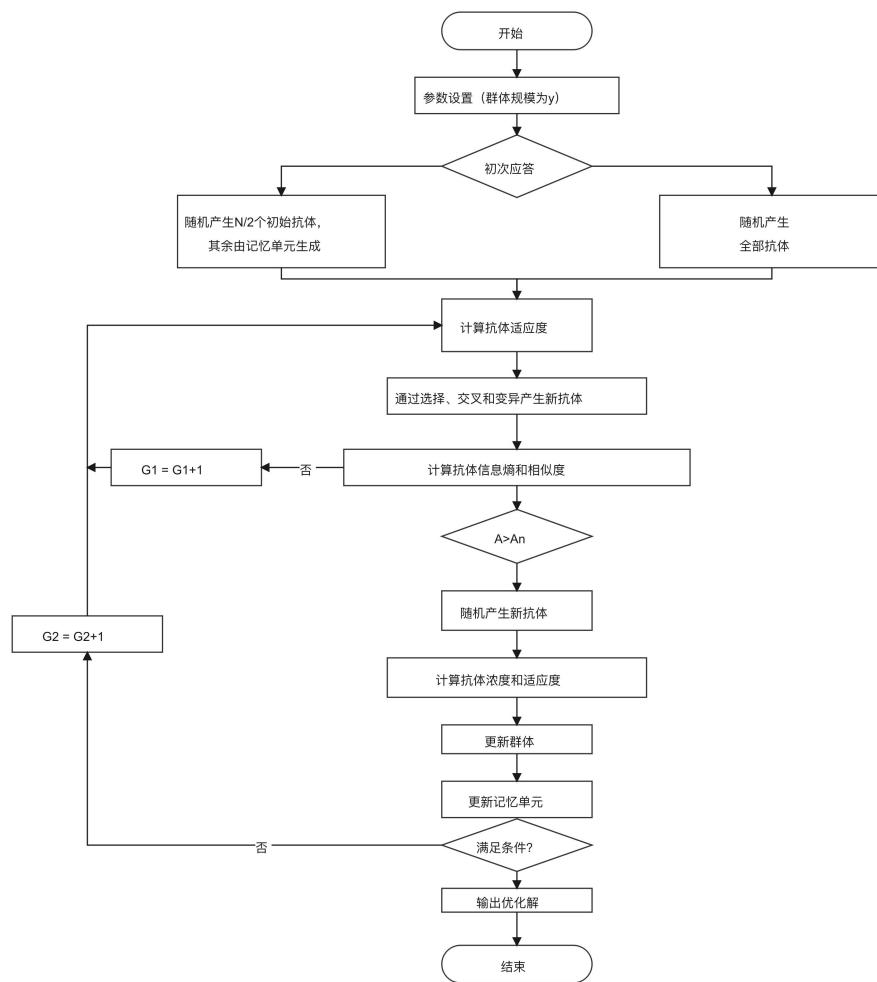


图 15 免疫算法流程图

6.3 问题 2 的求解结果

正常流转货物量以及工作负荷如下表 3 所示:

表 3: 时间序列预测流量结果

路线	过量/剩余货量(件)	工作负荷(%)
69-8	15303.161	100.00%
69-14	10018.26	100.00%
69-62	349.7419	100.00%
69-10	919.9355	100.00%
23-32	11591.1	94.86%
64-8	3880.226	87.90%
60-8	8303.161	84.48%
60-10	16591.9	81.19%
27-10	2018.194	68.67%

8-14	5653.935	51.47%
52-8	1712.774	51.38%
26-64	186.5806	47.00%
4-23	6707.71	46.25%
70-4	26392.87	44.31%
38-64	6942.29	43.62%

问题 2 结论的部分可视化如图 16 所示:

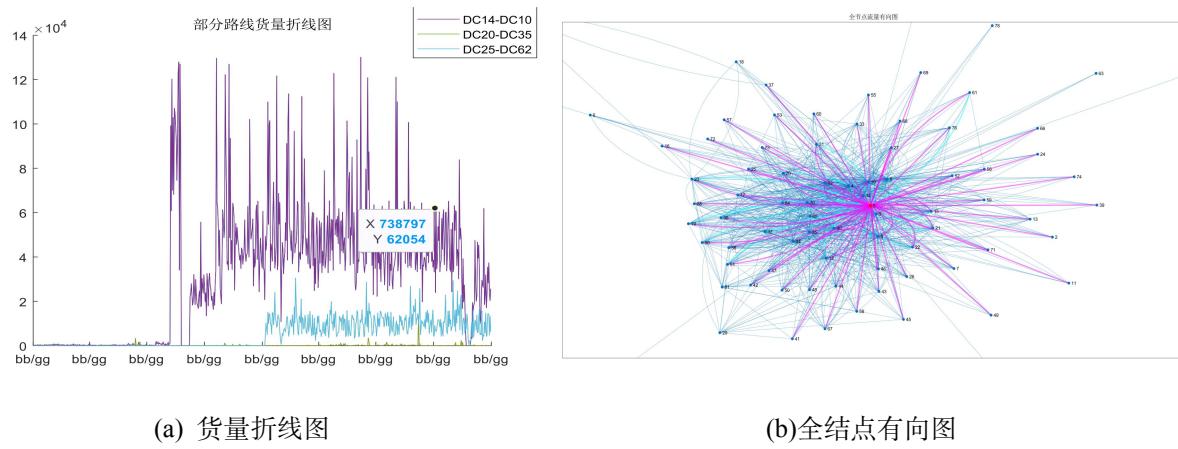


图 16 问题 2 结论的部分可视化

第 7 章 问题 3 的模型建立与求解

7.1 物流线路动态规划模型建立与求解

Step1: 确定决策变量

首先构建以下决策变量以及定义开通路线的集合:

- (1) $p_{ij} = 0$ 表示从 i 到 j 未开通新的路线;
- (2) $p_{ij} = 1$ 表示从 i 到 j 开通新的线路;
- (3) $q_{ij} = 0$ 表示从 i 到 j 不关闭的线路;
- (4) $q_{ij} = 1$ 表示从 i 到 j 关闭的线路;
- (5) V 表示网络中所有节点的集合。

Step2: 多目标规划的目标体系

目标函数为最小化 DC9 关停前后货量发生变化的线路数、保持各条线路的工作负荷

尽可能均衡和最小化 2023-01-01 至 2023-01-31 期间未能正常流转的包裹日累计总量，上述的三个目标既互相独立又互相联系，因此我们将问题三考虑成多目标优化问题。

目标函数 1：最小化货量变化的线路数

DC9 关停后通过关闭、新增线路来分配 DC9 的相关货物量，因此定义一个二元变量，通过将所有线路的变量相加，得到变化的线路数，使新增和关闭的总路线次数最小。

$$\sum_{i \in V} \sum_{j \in V} \sum_i^T u_{ijt} + \sum_{i \in V} \sum_{j \in V} \sum_t^T g_{ijt} \quad (13)$$

目标函数 2：最小化各线路运载的不平衡

为使各条线路的工作负荷尽可能均衡，建立如下目标函数使得各线路在 DC9 关停前后的运载量的差最小化：

$$\sum_{i,j \in V} \sum_t^T [w_{ij} - (x_{ijt} + y_{ijt})]^2 \quad (14)$$

目标函数 3：最小化无法正常运转的包裹日累计总量

为使所有包裹尽可能正常运转，需尽可能减少不能正常流转的包裹数量，由此我们定义了一个变量来表示时间为 t 时线路 i 到 j 上不能正常运转的包裹数量。我们的目标是最小化的总和，即在整个时间段内不能正常运转的包裹数量之和，即

$$\sum_{j \in V} \sum_{i \in V} m_{ij}(t) \quad (15)$$

其中 m_{ij} : 路线 i 到路线 j 上不能正常流转的包裹数量。

Step3: 确定约束条件

(1) 关停 DC9 后要保证所有包裹均能正常流转，应限制因 DC9 关停导致的新增线路及原有线路改变后的货运量的总和，使其与 DC9 关停前的总货物量相等

$$\sum_{i \in V} \sum_{j \in V} [(1 - q_{ijt}) (x_{ijt} + y_{ijt}) + u_{ijt} * z_{ijt}] = c_i \quad (16)$$

其中 c_i 表示有关 DC9 线路的货物量，该约束条件保证所有包裹在任何情况下都能正常流转，即保证货物流通的连通性。

(2) 当关闭某条线路时，这条线路的货物流量为 0

$$\sum_{i \in V} g_{ijt} * y_{ijt} = 0, \forall i, j \in V \quad (17)$$

分配后线路的运输能力不能超过原有线路的运输能力的最大值

$$\sum_{j \in V} [(1 - g_{ijt}) (x_{ijt} + y_{ijt}) + u_{ijt} * z_{ijt}] \leq R_j \quad (18)$$

其中 R_j 为已有线路的运输能力的最大值

(3) 每条线路上的货物数量都为非负整数

$$x_{ij} \in \mathbb{Z} \quad (19)$$

最后，我们利用免遗传算法对模型进行求解。

7.2 问题 3 的求解结果

流量的时间序列预测结果如下表 4 所示：

表 4：时间序列预测流量结果

起始站点 1	到达站点 2	添加时间	流量
3	1	2023 年 1 月 1 日	116.9537
3	1	2023 年 1 月 2 日	90.5448
3	1	2023 年 1 月 3 日	202.9713
3	1	2023 年 1 月 4 日	487.4328
3	1	2023 年 1 月 5 日	828.4849
5	8	2023 年 1 月 6 日	705.4949
5	8	2023 年 1 月 7 日	519.8781
5	8	2023 年 1 月 8 日	463.2876
3	8	2023 年 1 月 9 日	393.8699
5	8	2023 年 1 月 10 日	372.7428
19	1	2023 年 1 月 11 日	400.6607
19	1	2023 年 1 月 15 日	429.3333
19	1	2023 年 1 月 16 日	473.0966
19	14	2023 年 1 月 17 日	544.0233
64	14	2023 年 1 月 18 日	635.3227
64	14	2023 年 1 月 19 日	753.0309
64	14	2023 年 1 月 20 日	906.2025
64	9	2023 年 1 月 21 日	1094.083
64	9	2023 年 1 月 22 日	1327.236
64	9	2023 年 1 月 23 日	1620.752
64	9	2023 年 1 月 24 日	1991.986

64	9	2023 年 1 月 25 日	2459.046
----	---	-----------------	----------

正常流转货物量以及工作负荷如下表 5 所示:

表 5: 正常流转货物量以及工作负荷

路线	过量/剩余货量(件)	工作负荷(%)
27-10	15303.161	100.00%
8-14	16591.9	100.00%
52-8	2018.194	99.48%
26-64	5653.935	81.19%
4-23	1712.774	68.67%
70-4	186.5806	51.47%
38-64	6707.71	51.38%
27-8	26392.87	50.48%
27-14	6942.29	49.19%
59-62	1761.097	48.67%
30-14	3314.387	48.46%
10-4	5538.97	47.00%
10-62	13821.65	46.25%
8-62	17496.84	44.31%
23-14	12919.32	43.62%
62-8	4438.23	43.62%
62-14	11743.16	42.91%
18-14	6051.871	42.77%
8-10	23510.39	41.36%
26-10	2968.355	39.44%

由表 4 和表 5 可知,只有路线 27-10 与路线 8-14 的负荷率为 100%, 其他线路负荷量都在范围以内, 并且所有线路平均负荷率为 10.34%, 所有路线的平均剩余货物量在 10000 以内, 说明 DC9 停运后, 随着每日新增站点对整体配送效果影响不大, 没有线路超负荷, 故满足题中要求。

第 8 章 问题 4 的模型建立与求解

8.1 基于熵权 TOPSIS 分析的线路综合评价模型建立

针对问题 4 的相关要求，需要对不同物流场地及线路的状况进行评价，我们采用极差标准化方法对原始数据消除指标的量纲影响来进行标准化，标准差越小其偏离程度也越小，具体处理方式如下：

Step1: 路线重要性指标选取极差标准化处理

$$\text{正向指标: } x'_{ij} = \frac{x_j - x_{min}}{x_{max} - x_{min}}, \quad \text{负向指标: } x'_{ij} = \frac{x_{max} - x_j}{x_{max} - x_{min}}$$

x'_{ij} 是第 i 个样本在第 j 个评价指标上的数值，从而得到各评价指标统计值，设其矩阵表示为：

评价不同物流线路的重要性需要考虑以下因素：

- ① 货物运输数量 A_1
- ② 货物运输次数 A_2
- ③ 最大货物运输量 A_3
- ④ 货物运输均衡度 A_4

评价不同物流场地的重要性需要考虑以下因素，数学符号如下所示：

- ① 场地发送货物总量 B_1
- ② 场地接受货物总量 B_2
- ③ 场地发送货物次数 B_3
- ④ 场地接受货物次数 B_4
- ⑤ 场地发送货物平均值 B_5
- ⑥ 场地接受货物平均值 B_6

Step2: 确定最优向量、最劣向量

$$Z^+ = \max_{1 \leq i \leq n} \{Z_{ij}\} = (Z_1^+, Z_2^+, \dots, Z_n^+) \quad (20)$$

$$Z^- = \min_{1 \leq i \leq n} \{Z_{ij}\} = (Z_1^-, Z_2^-, \dots, Z_n^-) \quad (21)$$

Step3: 熵权法计算指标的权重

熵权法求解出路线指标对应的权重如下表 6 所示：

表 6: 熵权法求解出路线指标对应的权重

指标	信息熵值 e	信息效用值 d	权重
货物量	0.7568	0.3692	38.944%
运输次数	0.9953	0.1307	13.815%
最大运输货量	0.8598	0.2634	28.159%
运输均衡度	0.8206	0.3054	32.283%

熵权法求解出场地指标对应的权重如下表 7 所示:

表 7: 熵权法求解出场地指标对应的权重

指标	信息熵 e	信息效用值 d	权重
发送货物总量	0.76382	0.33218	15.74369
接受货物总量	0.50331	0.59269	28.20348
发送次数	0.95566	0.14034	6.56589
接受次数	0.61558	0.48042	22.81888
发送平均值	0.84339	0.25261	11.96684
接受平均值	0.59705	0.49895	23.71922

8.2 TOPSIS 流程图

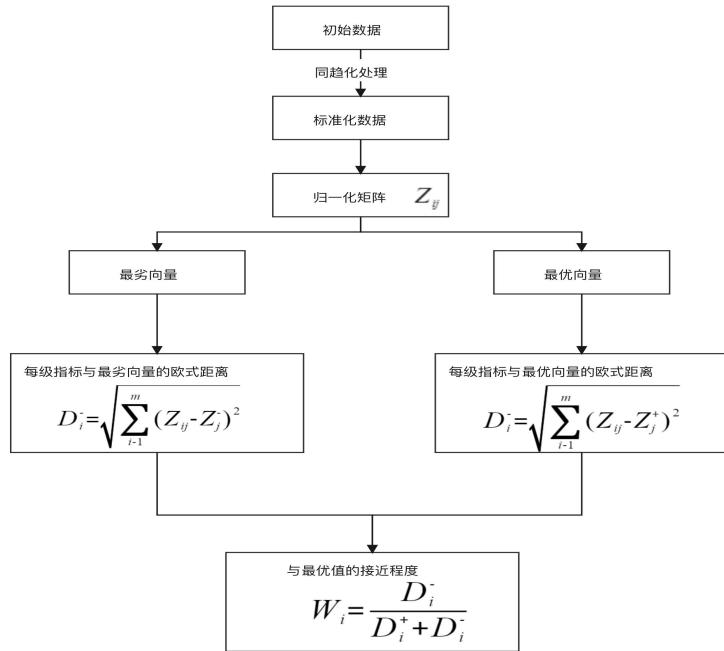


图 17 TOPSIS 流程示意图

8.3 问题 4 的求解结果

Topsis 模型评价结果-场地如下表 8 所示:

表 8: Topsis 模型评价结果-场地

排名	场地	正理想解距离(D^+)	负理想解距离(D^-)	综合得分指数
1	DC14	0.38664	0.80016	0.76880
2	DC10	0.39692	0.78698	0.64608
3	DC69	0.51929	0.77288	0.63101
4	DC60	0.46701	0.61494	0.62224
5	DC27	0.63108	0.54382	0.60861
6	DC8	0.68123	0.35287	0.59340
7	DC51	0.77638	0.35398	0.58195
8	DC53	0.83703	0.38018	0.44745
9	DC62	0.77631	0.36316	0.38974
10	DC30	0.84246	0.25156	0.35518
11	DC22	0.83425	0.23220	0.33580
12	DC9	0.83201	0.27058	0.34177
13	DC38	0.85319	0.25036	0.33513
14	DC36	0.85471	2.40732	0.24701
15	DC25	0.87578	2.66182	0.24098
16	DC30	0.84247	2.40201	0.23322
17	DC52	0.76206	2.35393	0.22937
18	DC25	0.85220	2.20655	0.21705
19	DC47	0.87704	2.07338	0.21521
20	DC12	0.87927	2.06502	0.20215

Topsis 模型评价结果-路线如下表 9 所示:

表 9: Topsis 模型评价结果-路线

排名	路线	正理想解距离(D^+)	负理想解距离(D^-)	综合得分指数
1	14-8	0.08919	0.94382	0.86253
2	14-9	0.39354	0.68051	0.59526

3	36-4	0.52265	0.46033	0.43872
4	20-10	0.57860	0.47506	0.42226
5	44-8	0.54718	0.43683	0.41572
6	21-10	0.59003	0.45884	0.40962
7	20-4	0.56168	0.43181	0.40695
8	17-4	0.56295	0.42907	0.40495
9	25-10	0.60251	0.45687	0.40377
10	14-10	0.62207	0.43726	0.38633
11	22-4	0.58822	0.41178	0.38540
12	22-10	0.63248	0.44145	0.38472
13	19-10	0.64837	0.41892	0.36724
14	10-4	0.61182	0.38840	0.36329
15	36-10	0.72333	0.38007	0.32201
16	23-4	0.69905	0.33643	0.30363
17	23-10	0.69736	0.33540	0.30350
18	19-4	0.69057	0.32537	0.29928
19	21-4	0.66669	0.31126	0.29741
20	17-10	0.77716	0.35441	0.29264

其中 DC10、DC14、DC4 场地对整个物流网络的影响比较大。综上考虑，在原有 81 个场地的基础上构建 82 号场地，将 82 号场地作为中转站，并在场地 10、14；场地 14、8 之间设置相应路线。

新增线路的运输能力应该根据场地的转运能力进行调整。即新线路的运输能力应该与已有线路的最大运输能力相近或略高，以确保流量能够合理分配。

该模型的鲁棒性分析详见第 9 章。

第 9 章 模型的测试

9.1 鲁棒性分析

本文是基于机器学习算法建立模型。机器学习模型的鲁棒性问题会导致模型在环境扰动影响下得到错误的、甚至特定的结果，危害使用者的安全。正确性衡量模型对未知数据的预测性能，即对于给定未知输入，模型能够正常预测或分类的能力；如图 18 所示，本文的预测模型在具有轻微扰动的原始输入上的预测结果不会显著偏离该原始输入，即鲁棒性较强。说明该模型在输入样本存在细微对抗扰动的情况下，模型预测具有不受对抗样本干扰或误导的能力。

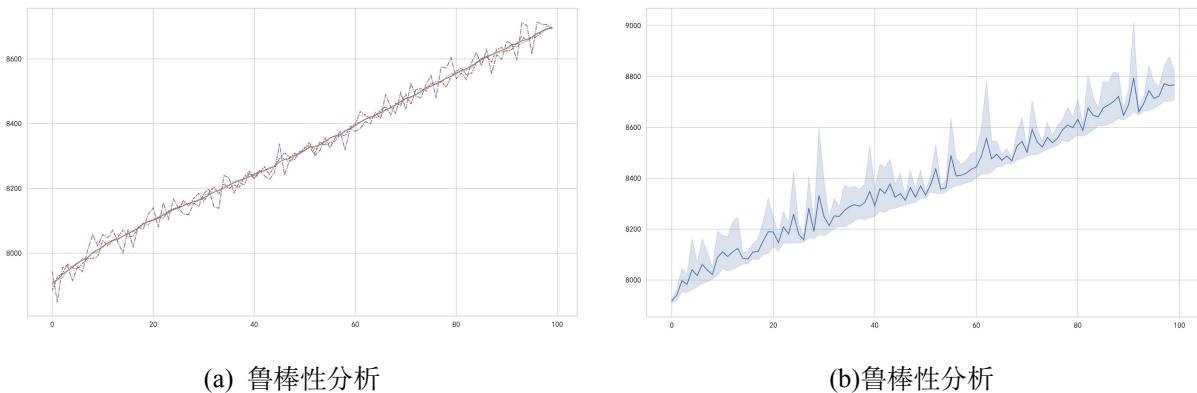


图 18 鲁棒性分析

由此可以看出该模型具有较好的鲁棒性。

第 10 章 模型的评价与总结

9.1 模型的优点

- (1) 模型的求解采用启发式算法——免疫遗传算法来求解线路货量的多目标优化模型，将免疫理论和基本遗传算法各自的优点结合在一起，收敛于局部极值；
- (2) 本文基于 Informer 的长序列时间序列预测（LSTF）对原有的 AR 模型进行改进，能够更精确的对 2023 年的数据进行预测，并具有启发意义；
- (3) 采用熵权 TOPSIS 建立综合评价模型，具有较好的客观性；
- (4) 模型的敏感性分析表明了模型在不同参数组合下的有效性，证明了模型的鲁棒性，并且文章可视化效果良好。

9.2 可能的改进

- (1) 如果我们有更完整的数据，知道更多年份如 2010-2020 每条线路的货量，则可以更准确地预测 2023 年 1 月 1 日至 2023 年 1 月 31 日期间各线路的货量；
- (2) 使用一些近似分析方法对物流线路的货量的预测进行建模，在极端情况下的流转货物量及工作负荷可能会与实际情况相反。

第 11 章 参考文献

- [1] 王丽. 时间序列的函数型变系数自回归预测方法研究[D].山西大学,2016.
- [2] 刘洪笑,向勉,周丙涛,段亚穷,伏德粟.基于 Informer 的长序列时间序列电力负荷预测[J].湖北民族大学学报(自然科学版),2021,39(03):326-331
- [3] Zhou H, Zhang S, Peng J, 等。Informer: Beyond efficient transformer for long sequence time-series forecasting[C]//人工智能 AAAI 会议论文集。2021, 35(12): 11106-11115。
- [4] 何向婷. 基于免疫遗传算法的多目标优化研究[D].西安科技大学,2013.
- [5] 张迪,刘婷婷,宋家友. 基于动态规划的无人机编队最优协同容错控制 [J]. 电光与控制,2023,30(04):34-39.
- [6] 宋丹丹. 高维时间序列的白噪声检验[D].东北师范大学,2020.
- [7] 杨艳军,唐迪.我国有色金属期货市场价格泡沫实时预警研究——基于上确界 ADF 检验方法的分析[J].价格理论与实践,2022(12):114-117+202.
- [8] 王子,苌道方,范志川,项乔.基于免疫遗传算法考虑割缝及间隙补偿的矩形件排样优化研究[J].制造业自动化,2023,45(03):211-215.
- [9] 黄莲琴,刘明玥,梁晨.基于熵权 TOPSIS 法的公司绿色治理观测指标与评价研究[J].电子科技大学学报(社科版),2023,25(02):95-106.

附录

附录 1

介绍: 工具和软件

通过 Office 2023 编写和生成的论文。

使用 Python 和 Matlab2023a 生成计算图像和数据。

附录 2

介绍: Matlab 代码的自回归模型

```
clc
clear all
close all hidden
data21=xlsread("C:\桌面\25-62.xlsx");
GDP1=data21(:,1);
figure(1)
subplot(1,1,1);
plot(GDP1,'-ob','LineWidth',1.5,'MarkerSize',3);
title('原始 DC14→DC10');
hold on
datayuce=[GDP1];
out=[];
for i=1:1
    dataTrain=datayuce(:,i);
    mu = mean(dataTrain);
    sig = std(dataTrain);
    dataTrainStandardized = (dataTrain - mu) / sig;
    XTrain = dataTrainStandardized(1:end-1);
    YTrain = dataTrainStandardized(2:end);
    numFeatures = 1;
    numResponses = 1;
    numHiddenUnits = 250;
    layers = [sequenceInputLayer(numFeatures)
              lstmLayer(numHiddenUnits)
              fullyConnectedLayer(numResponses)
              regressionLayer];
    options = trainingOptions('adam', ...
        'MaxEpochs', 500, ...
        'GradientThreshold', 1, ...
        'InitialLearnRate', 0.005, ...
        'LearnRateSchedule', 'piecewise', ...
```

```

'LearnRateDropPeriod', 120, ...
'LearnRateDropFactor', 0.25, ...
'Verbose', 0, ...
'Plots', 'training-progress');

XTrain=XTrain';
YTrain=YTrain';

net = trainNetwork(XTrain,YTrain,layers,options);
net = predictAndUpdateState(net, XTrain);
[net, YPred] = predictAndUpdateState(net, YTrain(end));
numTimeStepsTest = 30;
for i = 2:numTimeStepsTest
[net, YPred(:, i)] = predictAndUpdateState(net, YPred(:, i-1), 'ExecutionEnvironment', 'cpu');
end
YPredout{i} = sig*YPred + mu;
out=[out YPredout{i}'];
end
figure(2)
plot(GDP1,'-ob','LineWidth',1.5,'MarkerSize',3);
hold on
plot(366:1:395,out(:,1),'LineWidth",1.5,"MarkerSize", ...
3,"LineStyle", "-.", "Marker", "*","MarkerEdgeColor",'r');
title('DC25→DC62');

```

附录 3

介绍: Python 代码的免疫遗传算法

```

import pandas as pd
import numpy as np
from pulp import *
import random
import math
from datetime import datetime

data = pd.read_excel('data.xlsx', header=0, index_col=[0,1])

date_columns = [datetime.strptime(col, '%Y/%m/%d') for col in data.columns[2:]]

sites = list(data.index.levels[0])
num_sites = len(sites)

model = LpProblem('Routing Optimization', LpMinimize)

routes = [i for i in range(num_sites*num_sites)]

```

```

days = [i for i in range(len(date_columns))]
X = LpVariable.dicts('X', ((i,j,t) for i in sites for j in sites for t in days), lowBound=0,
upBound=None, cat='Continuous')
Y = LpVariable.dicts('Y', (i for i in routes), lowBound=0, upBound=None,
cat='Continuous')

obj = lpSum(Y[i] for i in routes)
model += obj

for t, date_col in enumerate(date_columns):
    for j in sites:
        model += lpSum(X[i,j,t] for i in sites if i != j) <= data.loc[(j,date_col)]
    for i in sites:
        model += lpSum(X[i,j,t] for j in sites if i != j) <= data.loc[(i,date_col)]

    for i in sites:
        for j in sites:
            if i != j:
                model += lpSum(X[i,j,t] for t in days) <= Y[int(i*num_sites+j)]

    for t, date_col in enumerate(date_columns):
        for i in sites:
            for j in sites:
                if i != j:
                    model += X[i,j,t] <= Y[int(i*num_sites+j)] * data.loc[(i,date_col)]

    for i in sites:
        for j in sites:
            if i != j:
                for t in range(len(date_columns)-1):
                    if data.loc[(i,date_columns[t])] > 0 and data.loc[(j,date_columns[t+1])] > 0:
                        model += X[i,j,t] == X[j,i,t+1]

# 免疫遗传算法
def fitness(X):
    # 计算目标函数值
    obj_value = sum([X[i] for i in routes])
    return obj_value

def select_parents(population, fitness_values):
    # 选择父代
    # 使用锦标赛选择法
    tournament_size = 3
    parents = []
    for i in range(len(population)):
        participants = random.sample(range(len(population)), tournament_size)
        winner = participants[0]
        for j in range(1, tournament_size):
            if fitness_values[participants[j]] < fitness_values[winner]:

```

```

winner = participants[j]
parents.append(population[winner])
return parents

def crossover(parents):
    # 交叉操作
    # 随机选择两个父代进行交叉
    p1, p2 = random.sample(parents, 2)
    # 随机选择一个位置点进行交叉
    crossover_point = random.randint(1, num_routes - 1)
    # 生成两个子代
    c1 = p1[:crossover_point] + p2[crossover_point:]
    c2 = p2[:crossover_point] + p1[crossover_point:]
    # 返回两个子代
    return c1, c2

def mutation(child, mutation_rate):
    # 变异操作
    # 对每个基因以概率 mutation_rate 进行变异
    for i in range(len(child)):
        if random.random() < mutation_rate:
            # 生成随机值
            r = random.random()
            # 判断随机值所处的区间并改变基因
            if r < 0.33:
                child[i] = 0
            elif r < 0.66:
                child[i] = 1
            else:
                child[i] = 2
    # 返回变异后的子代
    return child

# 初始化种群
population_size = 50
population = [[random.randint(0, 2) for _ in range(num_routes)] for _ in range(population_size)]

# 设置参数
mutation_rate = 0.1
num_generations = 100

# 遗传算法主循环
for g in range(num_generations):
    # 计算适应度并进行排序
    fitness_values = [fitness(x) for x in population]
    sorted_indices = sorted(range(len(fitness_values)), key=lambda k: fitness_values[k])
    sorted_population = [population[i] for i in sorted_indices]

    # 输出每代最优解

```

```
print('Generation: {}, Best fitness: {}'.format(g, fitness_values[sorted_indices[0]]))

# 选择父代并进行交叉和变异操作
new_population = []
for i in range(int(population_size/2)):
    parents = select_parents(sorted_population, fitness_values)
    c1, c2 = crossover(parents)
    c1 = mutation(c1, mutation_rate)
    c2 = mutation(c2, mutation_rate)
    new_population.append(c1)
    new_population.append(c2)

# 更新种群
population = new_population

best_solution = population[sorted_indices[0]]
print('Best solution: {}'.format(best_solution))
```