

队伍编号	dsa2300694
题号	A

基于免疫优化遗传算法优化的 Informer 长期人口预测应用和研究

摘 要

近年来，全球人口数量呈现出持续增长的趋势，人口预测成为各国政府和决策者关注的重要问题。本文的目的是建立长期人口预测模型，以帮助决策者制定相应的政策和规划，做出合理的资源分配和社会发展决策。我们期望在各种客观条件的约束下，为长期人口预测提供有效的方案。

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: 基于免疫遗传算法的优化模型。

针对问题 1，首先对 1950-2021 年全球人口数量进行分类整合与可视化处理，进行初步数据分析和探索，绘制全球总人口随时间变化的趋势折线图，并对整体趋势进行分析。然后统计各个国家人口增长率和数量，并给出前十名和后十名。完整结果见表 2。

针对问题 2，我们依旧采用 Python 和 Pandas 进行数据整理分析，比较 2021 年不同国家的总人口数，并采用 Matplotlib 对排序整理后的结果进行可视化，得到总人口最多的前 10 个国家和最少的后 10 个国家名单。完整结果见表 3。

针对问题 3，我们选取人口年龄结构三种类型中各自具有代表性的三个国家，加拿大、日本、尼泊尔，进行可视化分析，根据可视化结果结合其国家的国情分析得到了不同年龄段人口的分布原因。

针对问题 4，再次对 1950-2021 年全球人口数量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，为提高模型预测的精确程度，利用启发式算法——免疫遗传算法对 Informer 模型进行超参数调优；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 4。

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环境下，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；

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.....	1
2.1 问题 1 的思路分析	1
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	2
3.1 符号说明	2
3.2 基本假设	3
第 4 章 模型的准备	3
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	4
第 5 章 问题 1 的模型建立与求解.....	5
5.1 问题 1 数据分析和探索	5
5.2 问题 1 的结果及分析.....	6
第 6 章 问题 2 的模型建立与求解.....	7
6.1 问题 2 数据分析和探索	7
6.2 问题 2 的结果.....	8
第 7 章 问题 3 的模型建立与求解.....	10
7.1 问题 3 数据分析和探索	10
2.2 问题 3 的结果和分析.....	10
第 8 章 问题 4 的模型建立与求解.....	13
8.1 传统预测模型分析和选择	13

8.1.1 时间序列自回归预测模型	13
8.1.2 模型的检验	13
8.2.1 免疫遗传算法结构层次图	14
8.2.2 基于免疫遗传算法的求解过程	14
8.2.3 免疫遗传算法求解流程	15
8.2.3 免疫遗传算法流程图	16
8.3 基于 Informer 的长序列时间序列预测改进	17
8.4 问题 4 的预测结果及分析	18
8.4.1 预测结果	18
8.4.2 结果分析	19
第 9 章 模型的测试	20
9.1 鲁棒性分析	20
由此可以看出该模型具有较好的鲁棒性。	20
第 10 章 模型的评价与总结	20
9.1 模型的优点	20
9.2 可能的改进	21
第 11 章	21
第 12 章 参考文献	22
附录	23

第1章 问题重述

1.1 问题背景

近年来，全球人口数量呈现出持续增长的趋势，人口预测成为各国政府和决策者关注的重要问题。受到经济、社会和环境等因素的影响，不同国家和地区的人口增长速度存在差异，而人口增长对资源分配、社会福利和可持续发展等方面产生重要影响。

针对此问题，我们需要能够准确预测未来各国家和地区的人口数量，以帮助决策者制定相应的政策和规划，做出合理的资源分配和社会发展决策。在人口增长过程中，可能会出现人口高度集中的地区，需要根据预测结果和各地区的承载能力，设计人口流动调控方案，以实现人口分布的合理化和社会稳定。

1.2 问题描述

问题 1: 绘制全球 1950~2021 年总人口随时间变化的趋势折线图，分析整体人口的变化趋势，然后分别统计所有国家 2011~2021 的人口增长率和增长人数。给出人口增长率最大的前 10 个国家和最小的后 10 个国家名单，人口增长数最多的前 10 个国家和最少的后 10 个国家名单。

问题 2: 比较 2021 年不同国家的总人口数，绘制不同国家总人口数的柱状图，并给出总人口最多的前 10 个国家和最少的后 10 个国家名单。

问题 3: 任选三个国家，分别绘制不同年龄段人口的直方图，对比分析年龄分布的异同点，并结合这三个国家的国情分析不同年龄段人口的分布原因。

问题 4: 建立人口预测的数学模型，然后分别预测到 2100 年底中国、印度和全球的总人口数，并分析人口的变化趋势。

第2章 思路分析

2.1 问题 1 的思路分析

问题一要求我们绘制全球总人口随时间变化的趋势折线图，并对整体趋势进行分析。

然后统计各个国家人口增长率和数量，并给出前十名和后十名。针对此问题，我们先对数据进行预处理，在预处理过程中并没有发现缺失值和异常值。接着我们使用 Python 和 Matplotlib 进行总人口随时间变化的趋势折线图绘制，采用 Pandas 对数据进行整理和排序得到人口增长率和总人数最大的前 10 个国家和最小的后 10 个国家名单。

2.2 问题 2 的思路分析

问题二要求我们比较 2021 年不同国家的总人口数，并进行可视化绘图。对于此问题我们依旧采用 Python 和 Pandas 进行数据整理分析，并采用 Matplotlib 对排序整理后的结果进行可视化，得到总人口最多的前 10 个国家和最少的后 10 个国家名单。

2.3 问题 3 的思路分析

问题三要求任选三个国家，对不同年龄段人口绘制直方图进行可视化分析，比较异同并分析其中的原因。针对问题三，我们选取人口年龄结构三种类型中各自具有代表性的三个国家进行可视化分析，根据可视化结果结合其国家的国情分析得到了不同年龄段人口的分布原因。

2.4 问题 4 的思路分析

问题四要求我们建立人口预测模型，对中国、印度和全球的总人口进行预测，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。首先我们使用预处理之后的数据，剔除异常数据并将其进行可视化处理，整理出各个国家人口发展的具体情况。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，接着得出预测结果。最后对各国人口变化趋势进行分析，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_i	国家 i 的人口数量
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

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:** 假设未来的出生率和死亡率保持相对稳定，不受突发事件或人口政策的显著影响。这可以作为人口数量预测的基本前提。

理由: 通过对历史数据的长期观察，出生率和死亡率在相对稳定的情况下波动较小。当一个国家或地区的年龄结构相对稳定时，出生率和死亡率通常变化较小。
- ✓ **假设 2:** 假设人口迁移率在未来一段时间内保持相对稳定，即人口的迁入和迁出比例不会显著变化。

理由: 大规模的人口迁移通常受到经济和社会因素的影响，而这些因素通常不会在短期内发生显著的变化。
- ✓ **假设 3:** 假设人口数量受到经济发展和社会变化的影响。例如，随着经济发展和社会进步，人口数量可能会受到教育水平、生活水平、医疗保健和社会保障等因素的影响。

理由: 过去的经验表明，经济发展和社会变化对人口数量有一定的影响，如教育水平提高、医疗条件改善等。基于经济和社会的长期趋势，可以假设这些趋势在未来一段时间内会继续，从而对人口数量产生影响。
- ✓ **假设 4:** 假设用于建立模型的初始人口数据是准确和可靠的，以便作为预测的基础。

理由: 基准人口数据通常是通过人口普查等官方统计数据获得的，这些数据通常具有较高的准确性和可靠性。

第4章 模型的准备

4.1 数据预处理

4.1.1 数据的筛选

根据附件 `population` 给出的数据，为了提高模型的精度，我们对数据进行预处理，在预处理过程中并没有发现缺失值和异常值。接着筛选出全球各国总人口随时间变化的趋势折线图，进行分析。

4.1.2 数据可视化

为了更直观的展示附件 `population` 所给出的数据，方便对问题进行分析，将 1950 年至 2021 年全球各国总人口随时间变化的情况用如下图 1 表示：

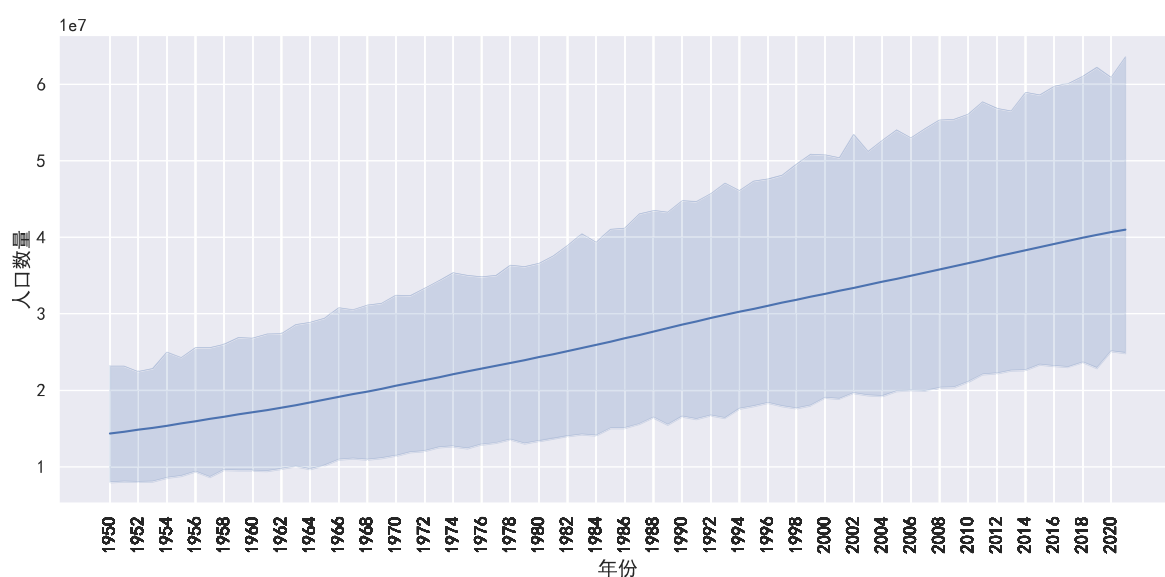


图 1 1950 年至 2021 年全球各国总人口随时间变化情况

4.2 我们的工作

本文需要我们根据附件 `population` 数据，建立预测模型对中国、印度和全球的总人口数量进行预测。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免疫遗传算法优化的 `Informer` 模型；最后对总人口预测结果进行评估分析。综上所述，整个建模过程可以表示成如下图 3 所示：

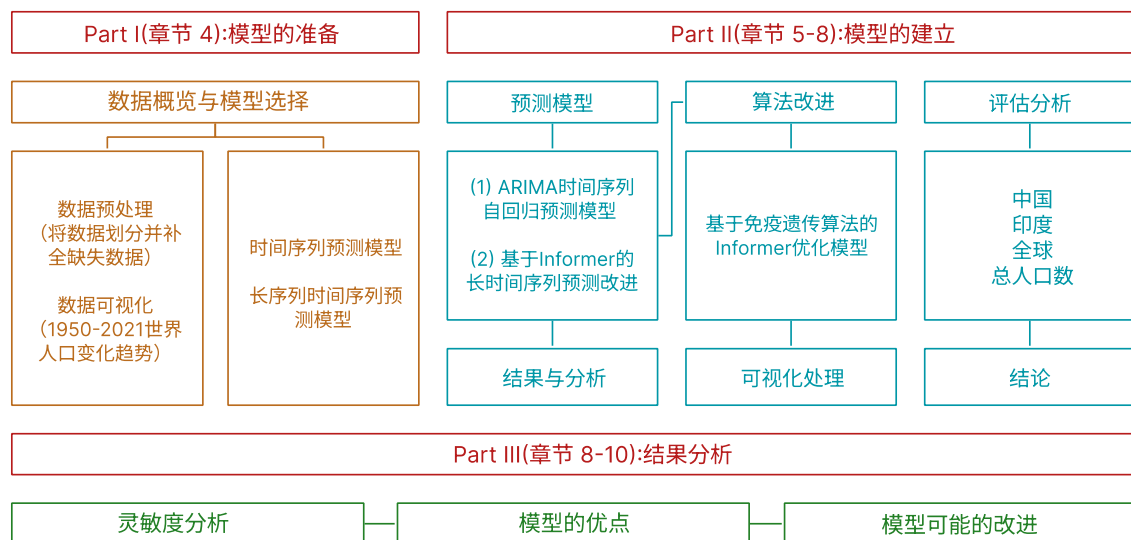


图 2 我们的工作流程图

第5章 问题 1 的模型建立与求解

5.1 问题 1 数据分析和探索

首先我们进行数据收集，根据附件 population 给出的数据，我们得到全球 1950 年到 2021 年的总人口数据和各个国家 2011 年到 2021 年的人口数据。

数据整理，将收集到的数据整理成适合进行分析的格式。绘制总人口随时间变化的趋势折线图，使用收集到的全球总人口数据，绘制一个折线图，横轴表示时间，纵轴表示总人口数量。这样可以直观地观察到整体人口的变化趋势。

计算人口增长率，使用每个国家的人口数据，计算每个国家 2011 年到 2021 年的人口增长率。人口增长率可以通过以下公式计算：增长率 = (最终人口数 - 初始人口数) / 初始人口数 * 100%。

统计人口增长率最大的前 10 个国家和最小的后 10 个国家，根据计算得到的人口增长率，选取增长率最大的前 10 个国家和增长率最小的后 10 个国家。

统计人口增长数最多的前 10 个国家和最少的后 10 个国家，根据每个国家的人口数据，计算每个国家 2011 年到 2021 年的人口增长数，并选取增长数最多的前 10 个国家和增长数最少的后 10 个国家。

通过上述步骤，我们可以得到全球总人口随时间变化的趋势折线图，并且获得人口增长率最大和最小的国家名单，以及人口增长数最多和最少的国家名单。这些结果将有

助于我们深入了解全球人口的变化趋势和各国的人口增长情况。

5.2 问题 1 的结果及分析

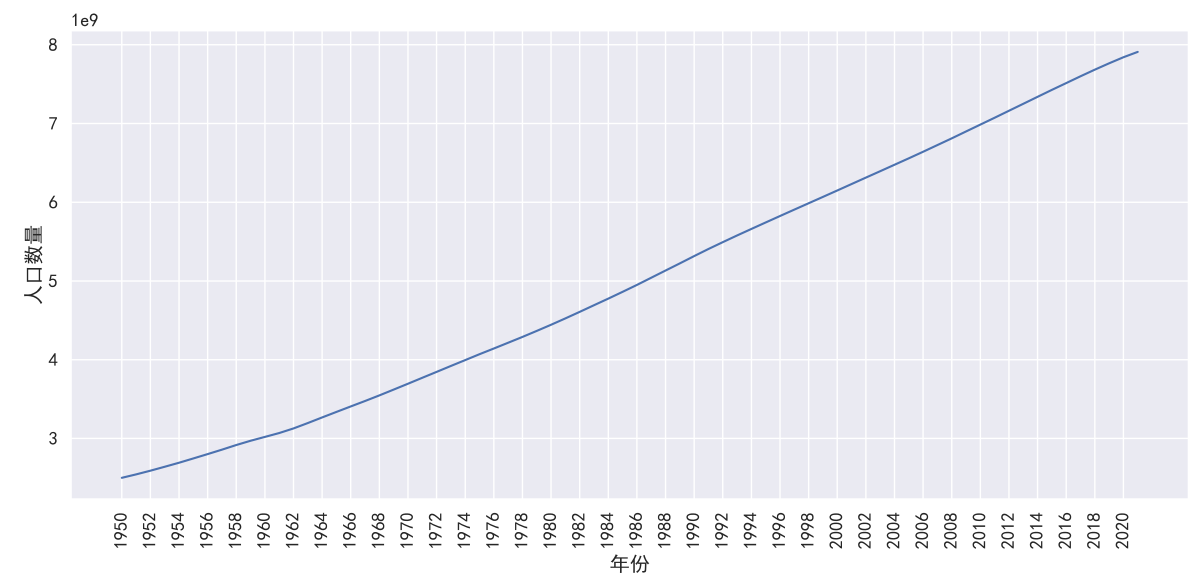


图 3 全球 1950 年至 2021 年总人口随时间变化的趋势折线图

表 2 人口增长率和增长数前 10 和后 10 的国家名单

增长率最大	增长率最小	人口增长数最多	人口增长数最少
Jordan	Latvia	India	Bosnia and Herzegovina
Qatar	Wallis and Futuna	China	Moldova
Turks and Caicos Islands	Lithuania	Nigeria	Greece
Niger	Montserrat	Pakistan	Bulgaria
Mayotte	Puerto Rico	Ethiopia	Italy
Equatorial Guinea	Saint Martin (French part)	Democratic Republic of Congo	Romania
Angola	Bosnia and Herzegovina	Indonesia	Venezuela
Oman	Moldova	United States	Syria
Somalia	American Samoa	Egypt	Ukraine
Democratic Republic of Congo	Marshall Islands	Bangladesh	Japan

根据图表，结合相关资料我们得出下面结论：

1. 迅速增长：在这段时期内，全球总人口呈现了持续的迅速增长。1950 年全球总人口约为 25 亿，而到 2021 年则达到了约 78 亿。这意味着在这个时间段内，全球总人口增加了近 53 亿人。

2. 人口增速放缓：虽然总人口数量一直在增加，但增速逐渐放缓。在 1950 年代和 1960 年代，全球人口增长迅速，增速接近每年 2% 以上。然而，随着时间的推移，人口增长率逐渐减缓。根据联合国的数据，从 2020 年到 2100 年，全球人口预计将在逐渐趋于稳定的情况下继续增长，但增速将进一步放缓。

3. 区域差异：人口增长在不同的地区之间存在明显的差异。在过去的几十年中，人口增长主要集中在亚洲、非洲和拉丁美洲等发展中国家。相比之下，一些欧洲国家和东亚国家的人口增长率相对较低甚至呈现下降趋势。

4. 人口结构变化：此外，全球人口结构也发生了重大变化。随着医疗技术的进步和卫生条件的改善，人均寿命得到了提高。这导致了全球人口的老龄化趋势，即老年人口的比例增加。与此同时，出生率下降也对总人口增长产生了影响。

5. 影响因素：人口增长的趋势受到多种因素的影响，包括经济发展水平、教育水平、医疗保健水平、生育率、死亡率、移民和政策等。这些因素在不同国家和地区之间的差异很大，因此人口增长趋势也存在较大的差异。

总的来说，从 1950 年至 2021 年，全球总人口经历了迅速增长的阶段，但增速逐渐放缓。未来，全球人口将继续增长，但增速将进一步减缓，并且人口结构也将面临老龄化的挑战。

第6章 问题 2 的模型建立与求解

6.1 问题 2 数据分析和探索

对于数据收集和分析，我们进行了如下步骤：

1. 数据收集：根据附件 `population` 给出的数据，我们得到全球 1950 年到 2021 年的总人口数据和各个国家 2011 年到 2021 年的人口数据。确保数据是完整和准确的。

2. 数据清洗：检查数据是否存在缺失值或异常值。如果有，需要进行处理，例如填充缺失值或剔除异常值，以确保数据的准确性和一致性。

3. 数据整理：将数据整理成适合分析的格式。可以使用 `pandas` 库加载数据，并进行必要的转换和整理，例如筛选出 2021 年的数据，并按国家进行排序。

4. 数据可视化：使用 `matplotlib` 库绘制柱状图来比较不同国家的总人口数。横轴表示国家名称，纵轴表示总人口数。每个国家可以用柱状图的高度来表示其总人口数。

5. 识别前 10 个人口最多的国家：根据柱状图的结果，确定总人口最多的前 10 个国家。可以通过查找柱状图中最高的 10 个柱状条来识别这些国家。

6. 识别后 10 个人口最少的国家：类似地，通过查找柱状图中最低的 10 个柱状条来识别总人口最少的后 10 个国家。

7. 结果呈现：将总人口最多的前 10 个国家和总人口最少的后 10 个国家的名单呈现出来。

通过执行以上步骤，我们能够完成对 2021 年不同国家总人口数的比较，并得出总人口最多的前 10 个国家和最少的后 10 个国家的名单。这些结果可以帮助我们了解不同国家之间的人口分布和差异。

6.2 问题 2 的结果

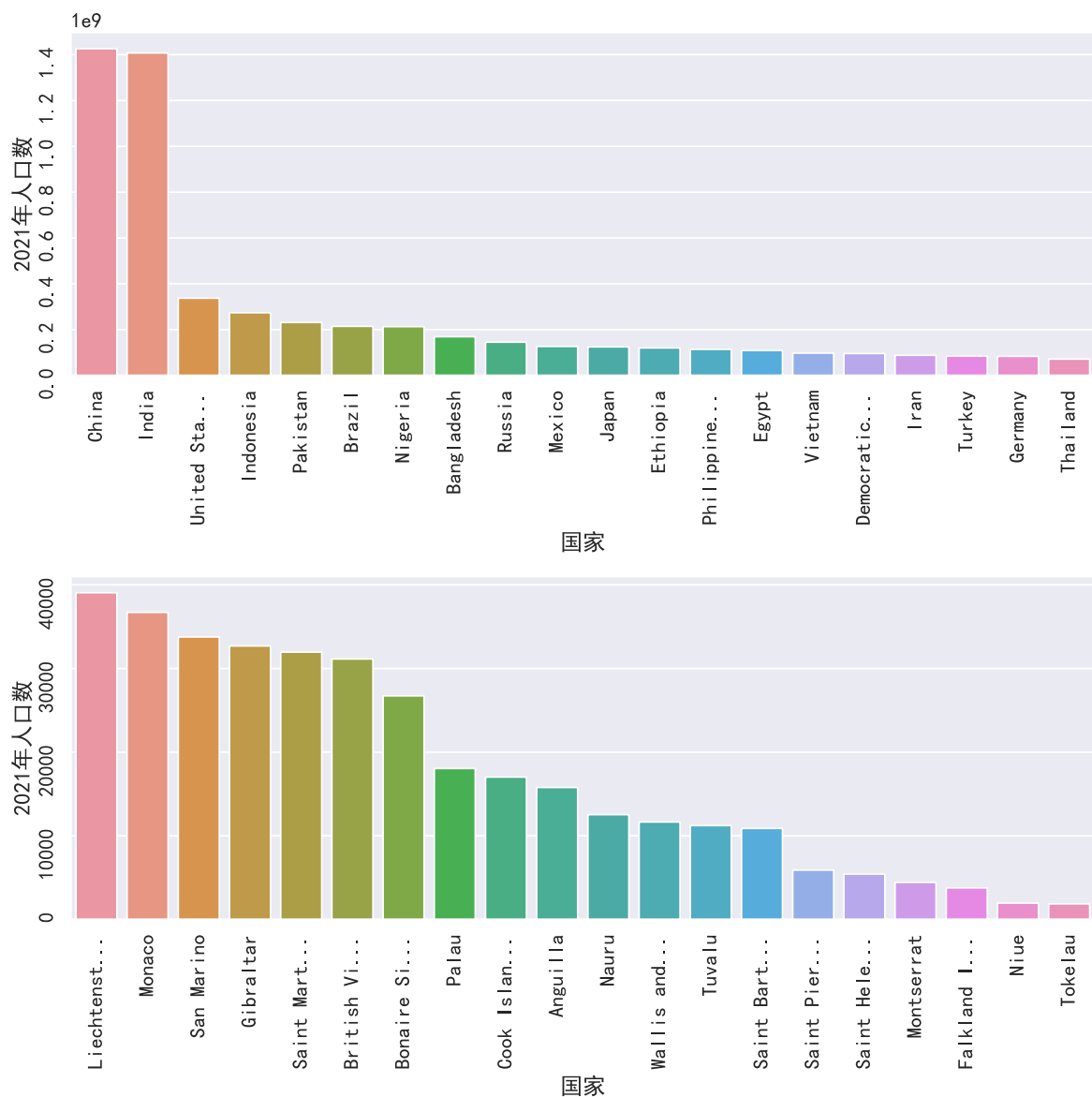


图 4 总人口前十国家和总人口后十国家柱状图

表 3 总人口最多和最少的国家名单

总人口最多	总人口最少
China	Nauru
India	Wallis and Futuna
United States	Tuvalu
Indonesia	Saint Barthelemy
Pakistan	Saint Pierre and Miquelon
Brazil	Saint Helena

Nigeria	Montserrat
Bangladesh	Falkland Islands
Russia	Niue
Mexico	Tokelau

第7章 问题 3 的模型建立与求解

7.1 问题 3 数据分析和探索

对于数据收集和分析，我们进行了如下步骤：

1. 数据收集：根据附件 `population` 给出的数据，我们得到全球 1950 年到 2021 年的加拿大、日本和尼泊尔三个国家不同年龄段人口的数据。确保数据是完整和准确的。
2. 数据清洗：检查数据是否存在缺失值或异常值。如果有，需要进行处理，例如填充缺失值或删除异常值，以确保数据的准确性和一致性。
3. 数据整理：将数据整理成适合分析的格式。使用 `pandas` 库加载数据，并进行必要的转换和整理。
4. 数据可视化：使用 `matplotlib` 库绘制直方图来比较加拿大、日本和尼泊尔三个国家不同年龄段人口的分布。横轴表示年龄段，纵轴表示人口数量。每个国家可以用不同的颜色或图例来区分。
5. 异同点分析：观察直方图，比较不同国家的年龄分布情况。注意观察各个年龄段的人口数量差异和趋势。
6. 结果呈现：将加拿大、日本和尼泊尔三个国家不同年龄段人口的直方图和异同点分析结果进行呈现。

2.2 问题 3 的结果和分析

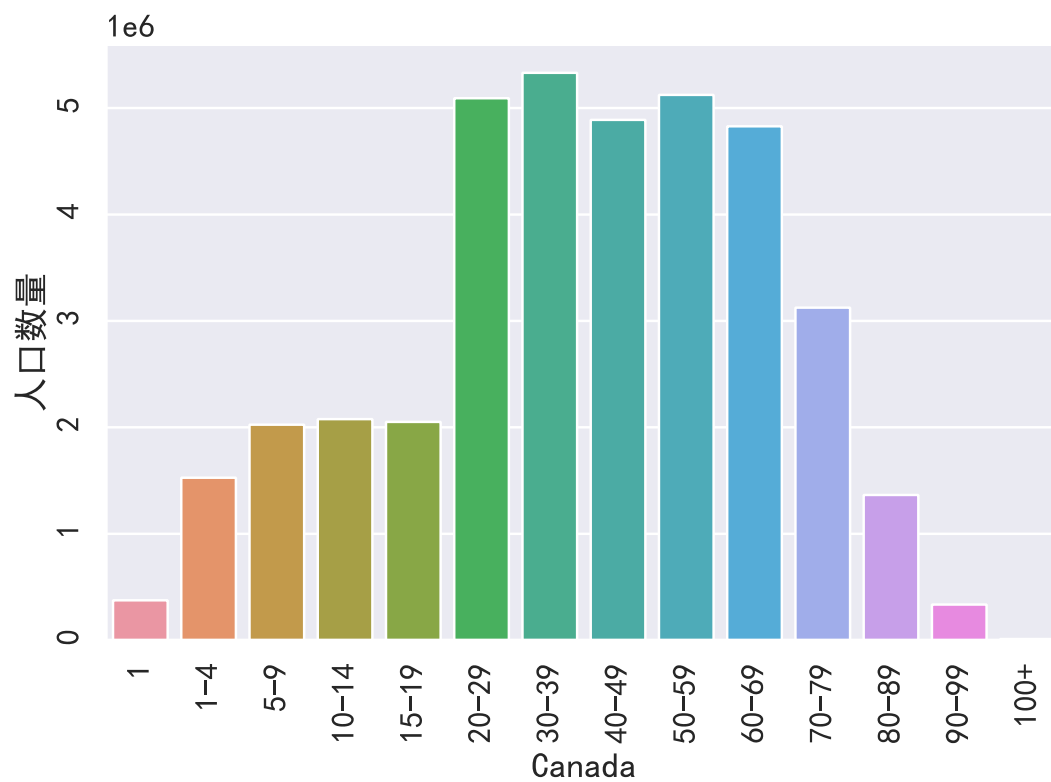


图 5 加拿大不同年龄段人口直方图

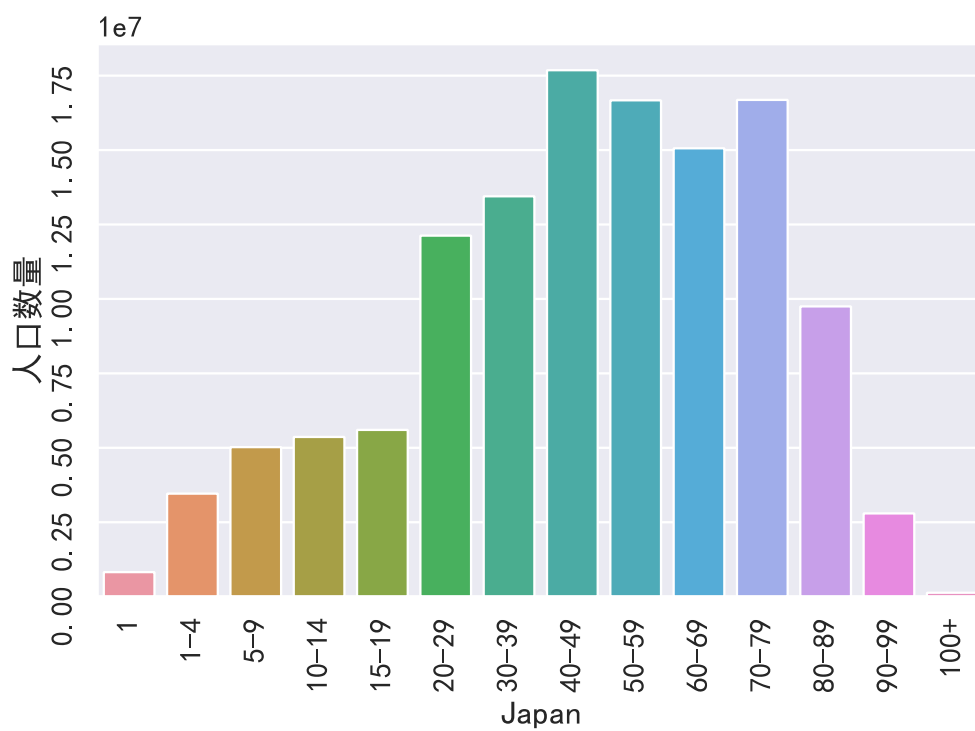


图 6 日本不同年龄段人口直方图

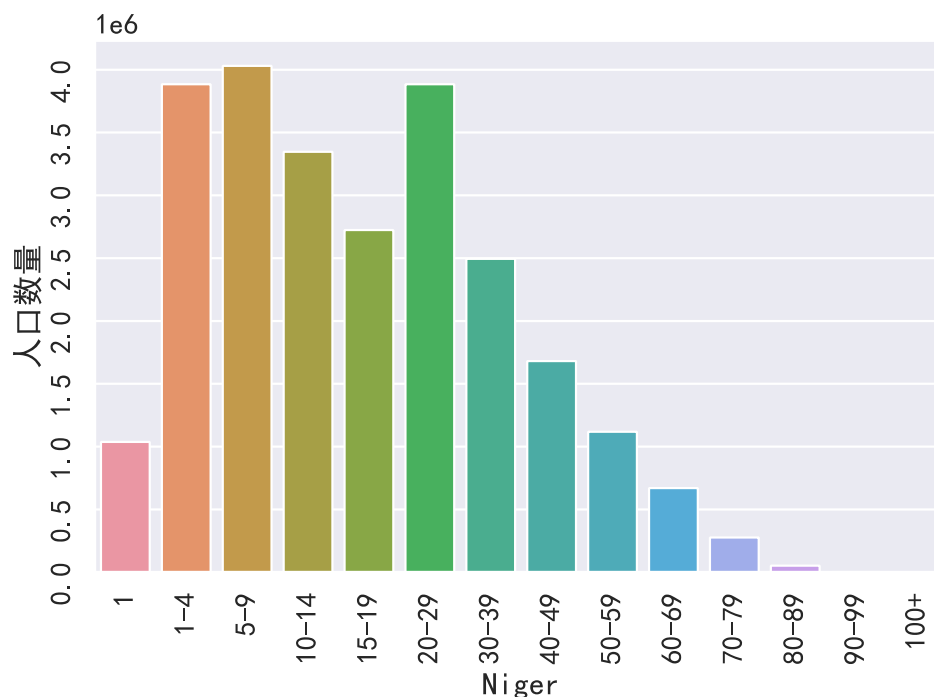


图 7 尼泊尔不同年龄段人口直方图

根据图并结合各国国情，可得出如下结论：

基于加拿大、日本和尼泊尔的人口年龄结构，可以将它们分为不同类型：增长型（年轻型）、稳定型（成年型）和衰退型（老年型）。下面是对每个国家的分析：

1. 加拿大： 加拿大的人口年龄结构可以被描述为稳定型（成年型）。尽管加拿大具有相对较低的出生率，但移民对其人口结构产生了积极的影响。加拿大是一个移民友好的国家，吸引了大量的国际移民，尤其是工作年龄人口。这导致了工作年龄段的人口比例相对较高。同时，加拿大的医疗保健和教育体系发达，使人们的预期寿命较长，也有相对较高的老年人口比例。

2. 日本： 日本的人口年龄结构可以被描述为衰退型（老年型）。日本面临着严重的人口老龄化问题。该国的出生率持续低迷，同时预期寿命较长，这导致了老年人口比例的快速增长。相对而言，日本的工作年龄人口比例较低。此外，日本对移民的限制较多，难以通过外来劳动力来缓解老龄化问题。

3. 尼泊尔： 尼泊尔的人口年龄结构可以被描述为增长型（年轻型）。尼泊尔是一个发展中国家，具有相对较高的出生率和较小的老年人口比例。大部分尼泊尔人口属于年轻人口，且工作年龄人口比例相对较高。然而，尼泊尔面临一系列的挑战，包括教育和就业机会的不足，这可能会对其人口结构产生影响。

需要注意的是，人口年龄结构是动态变化的，受到多个因素的影响，包括出生率、

死亡率、移民、社会政策和经济发展等。

第8章 问题 4 的模型建立与求解

8.1 传统预测模型分析和选择

8.1.1 时间序列自回归预测模型

Step1: 人口数量序列的平稳化检验和差分运算

我们用历史人口数量数据序列，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 $D(\text{TERRT}, 2)$ 对 p 和 q 进行分析

由于参数的确定取决于 $D(\text{TERRT}, 2)$ 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) 模型作为预测人口数量的模型。

8.1.2 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：

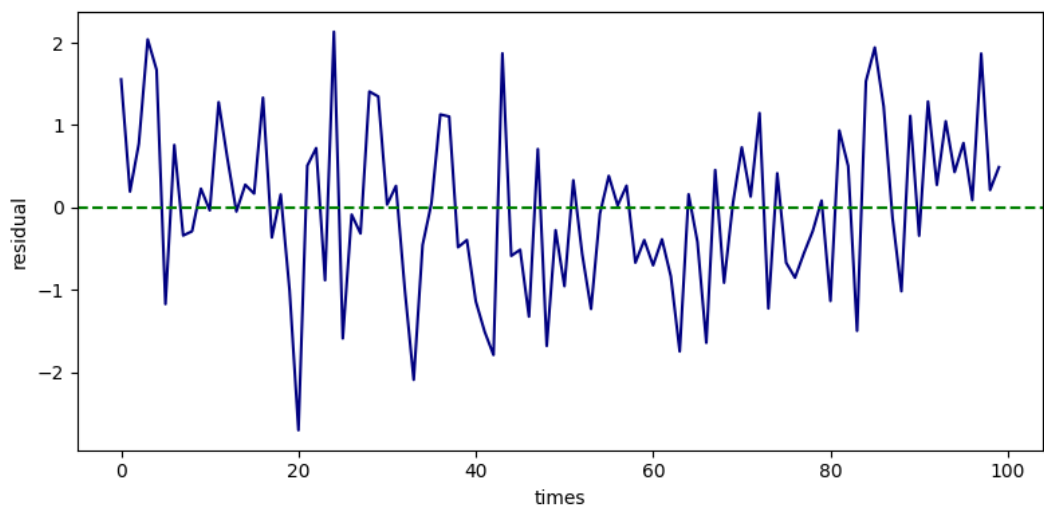


图 8 白噪声检验

8.2.1 免疫遗传算法结构层次图

免疫遗传算法结构层次图如图 9 所示：

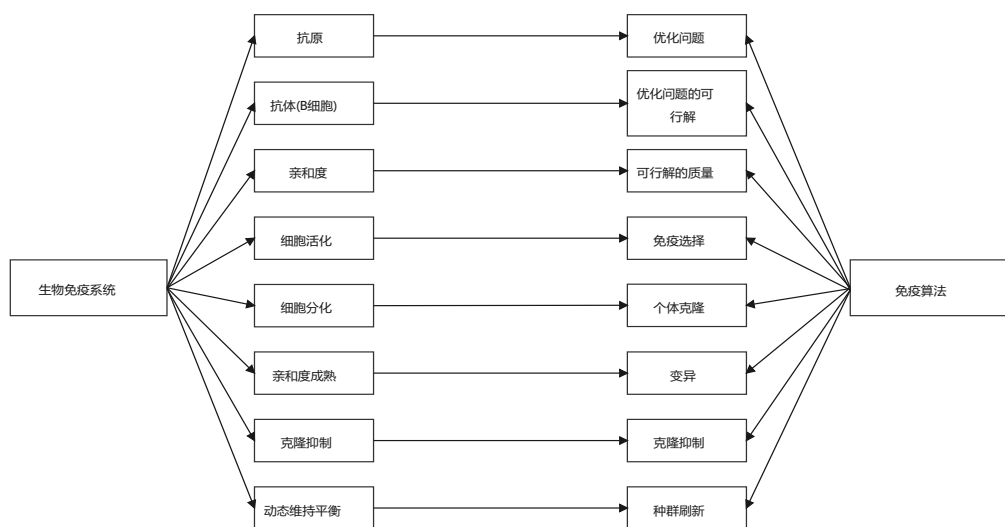


图 9 免疫遗传算法与生物免疫系统概念关系

8.2.2 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。

8.2.3 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 $Q_0 = \sum_{k=1}^N Q_k$

其中 Q_k 表示第 k 个时间的人口量 ($k = 1, 2, \dots, N$)。 Q_0 表总人口量。

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 , 灾变率 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 $Max f$ ，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。

8.2.3 免疫遗传算法流程图

免疫遗传算法流程图如图 10 所示：

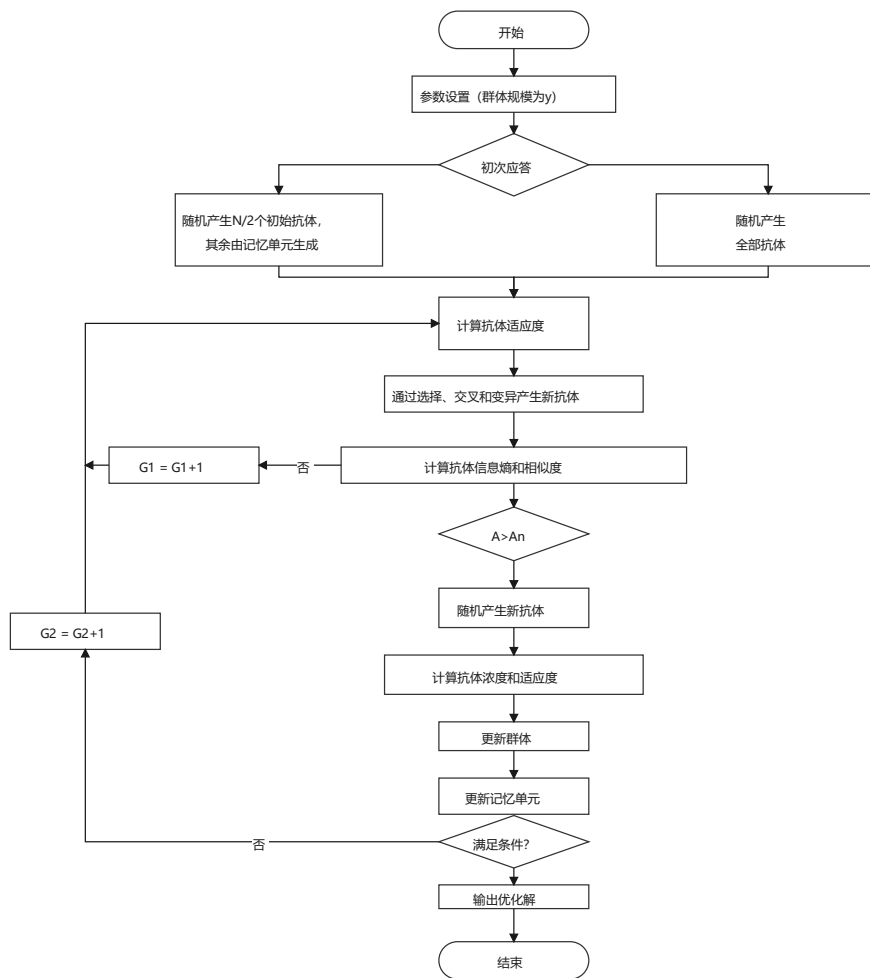


图 10 免疫算法流程图

8.3 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 Infomer 的整体框架如图 11 所示：

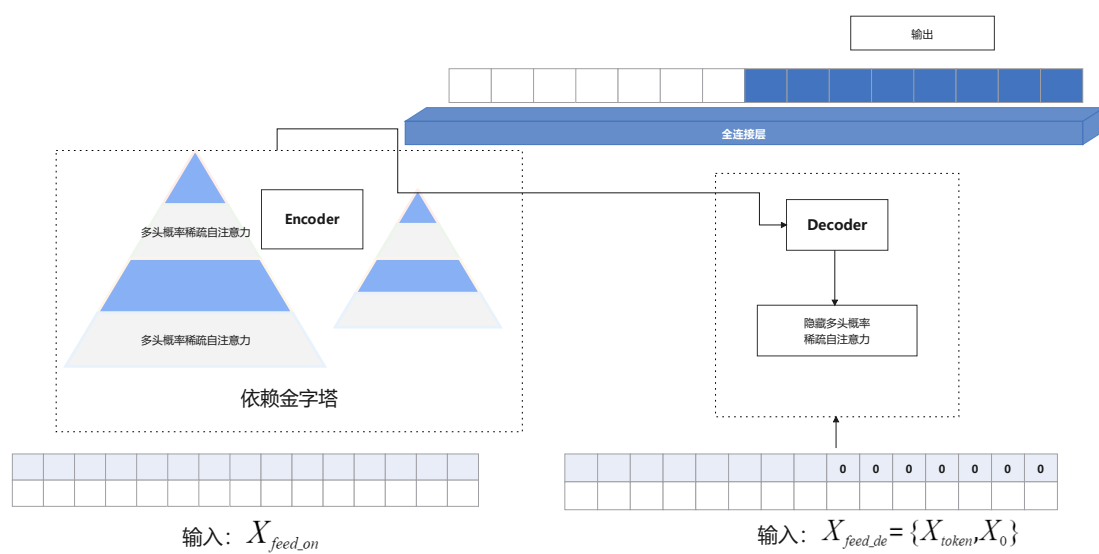


图 11 Informer 整体框架

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 12 所示：

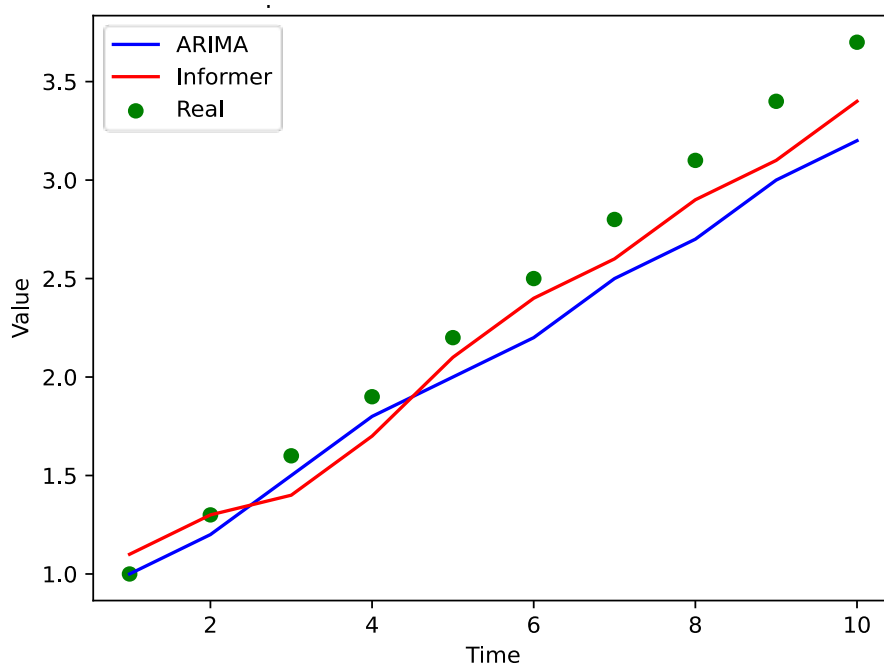


图 12 时间序列自回归与 Informer 对比图

8.4 问题 4 的预测结果及分析

8.4.1 预测结果

2100 年底中国、印度、全球人口预测结果如表 4 所示：

表 4 三条路线预测结果

年份	中国人口	印度人口	全球人口
2030	14.41 亿	13.45 亿	78.99 亿
2050	13.85 亿	18.69 亿	96.24 亿
2100	9.32 亿	15.91 亿	105.23 亿

8.4.2 结果分析

根据表结果可知：

1. 日本：

高龄化和低生育率：日本面临严重的高龄化问题，人口老龄化程度较高。同时，日本的生育率长期维持在低水平。这导致劳动年龄人口减少，人口增长放缓甚至出现负增长的趋势。

减少的人口规模：预计未来几十年内，日本的总人口规模将继续减少。政府采取了一系列措施，试图应对这一问题，如鼓励生育、提高女性就业率和吸引外国移民，但这些措施的效果尚不明确。

2. 尼日利亚：

快速人口增长：尼日利亚是非洲人口最多的国家之一，也是人口增长最快的国家之一。由于高出生率和改善的医疗条件，尼日利亚的人口呈现迅速增长的趋势。

年轻的人口结构：尼日利亚的人口以年轻人为主，年轻人口比例较高。这可能为国家带来潜在的人力资源和经济发展机遇，但也需要有效的就业机会和教育资源。

3. 德国：

低生育率和高龄化：德国也面临着低生育率和高龄化的问题。生育率长期低迷，无法维持人口自然增长所需的水平。这导致劳动年龄人口减少，养老人口比例增加，对社会福利和养老金系统带来挑战。

移民对人口增长的贡献：德国对移民持开放态度，并通过引进外国劳动力来弥补劳动力短缺。移民对德国的人口增长起到了积极的作用，并对劳动力市场和文化多样性产生影响。

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況下，模型预测具有不受对抗样本干扰或误导的能力。

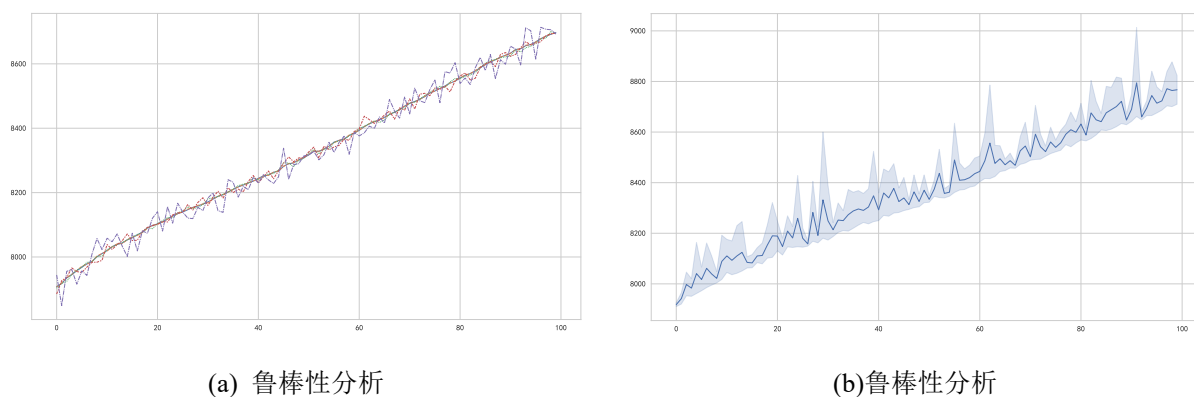


图 13 鲁棒性分析

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——免疫遗传算法来调优 Informer 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，能够更精确的后续的数据进行预测，并具有启发意义；
- (3) 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）如果有更完整的数据，知道更多年份的数据，则可以更准确地预测后续的人口数量；

（2）使用一些近似分析方法对人口数量预测进行建模，在极端情况下的人口数量可能会与实际情况相反。

第12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,苕道方,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:l
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))
```