中国研究生创新实践系列大赛

“华为杯”第二十届中国研究生

数学建模竞赛

学　　校	浙江科技学院
参赛队号	23110570013
队员姓名	1.俞鸿烽
	2.孙敏
	3.余毅

中国研究生创新实践系列大赛

“华为杯”第二十届中国研究生

数学建模竞赛

题 目： D 题 区域双碳目标与路径规划研究

摘 要：

我国主动提出“双碳”目标，将使碳减排迎来历史性转折。在此背景下，本文根据题意建立了主成分分析、多元线性回归以及组合预测等模型实现碳排放的现状分析、未来趋势预测以及双碳目标下的路径规划。

针对问题一，我们基于细致的分析和对相关文献的深入研究，构建了包括人口、经济、能源消费量以及区域碳排放量在内的 4 个一级指标和 21 个二级指标。同时，通过精确的统计分析，我们获得了“十二五”与“十三五”时期碳排放量的特征信息，例如碳排放总量、同比以及环比增长率。由此可以明显看出，“十三五”时期的碳排总量超过了“十二五”时期，但“十三五”时期的碳排增速却显著低于“十二五”时期。对 10 项与碳排放量相关的指标，我们采用了 **Pearson** 相关系数法进行简要分析后，利用**主成分分析法**对影响碳排放量的各因素以及其贡献量进行量化，可以看出人口以及经济对碳排放量存在显著影响。利用多元线性回归模型可以得出区域碳排放量与区域生产总值、常驻人口数量、区域能源消费量之间的关系式为 $\mathrm{CO2} =- 0.08\mathrm{GDP} - 64.36\mathrm{P} + 8.28\mathrm{E} + 401071.49$，得到的判定系数$R^2$的值为 0.97。在此基础上，本文确定了碳排放预测模型中的参数取值，为后续求解与分析奠定基础。

针对问题二，本文运用线性回归模型探求区域生产总值与时间的函数关系，并且结合多项式回归和线性回归模型，对常驻人口数量相对于时间进行了**组合预测**。根据题目可知，碳排放量不仅与常驻人口总量、区域生产总值以及能源消费量相关，还与各能源消费部门、能源供应部门以及能源消费品种有关。因此我们运用两次**多元线性回归**，首先拟合得到一次能源中化石能源消费量、一次能源非化石能源消费量等五种能源消费量关于区域生产总值（GDP）以及常驻人口数量（P）的关系式，接着拟合得到六个部门碳排放量关于五种能源消费量的关系式，区域总碳排放量即六个部门碳排放量的总和。最终得到的碳排放量关于时间的二次函数式为$CO2 =- 27.85x^2 + 114055.02x - 116638279.04$。

针对问题三，本题设计了自然情景、基准情景以及雄心情景三种情景。其中，自然情景下的最晚碳达峰时间为 2050 年；基准情景的碳达峰时间为 2030 年；雄心情景的碳达峰

时间为 2025 年。在假设区域生产总值 GDP 相关条件的基础上，本文建立了碳达峰的**优化模型**，在满足碳排量关于时间导数为 0 的情况下，求解出最晚碳达峰时间对应的人口数量，从而确定碳中和达成的时间。最后，计算出目标时间节点下的 GDP、人口以及能源消费量的目标值和能源利用效率以及提高化石能源消费比重的目标值。其中，基准情景下 2035 年的区域生产总值为 127568.94 亿元，常驻人口数量为 8602 万人，能源消费量为 33925 万 tce，其余数据见正文。

关键词：双碳目标，主成分分析，组合预测，优化模型

目录

1 问题背景与问题重述

1.1 问题背景

习近平总书记在第七十五届联合国大会上提出中国将提高国家自主贡献力度，二氧化碳排放力争于 2030 年前达到峰值，努力争取 2060 年前实现碳中和，重点申明要把握新一轮经济技术革命和产业转型的历史性机会，保证低碳发展的同时加快疫情后的全球经济复苏进程。经济增长与能源消费量之间以及能源消费量与碳排放量之间均存在关联关系，要想破解发展与减排的矛盾，必须寻求经济增长与碳排放量的负相关变化，进而需要从提高能源利用效率和提高非化石能源消费比重两个方面，实施能效提升、产业（产品）升级、能源脱碳和能源消费电气化等四大重点工程[1]。因此，建立并运用相应的数学模型，进行分析、评价和预测上述四大重点工程对碳排放的影响具有了重要的意义。

1.2 问题重述

本赛题假定以人口密集经济发达且能源及生态碳汇资源相对匮乏的中国东南沿海区域为研究对象，本区域的时间范围为 2010-2020 年，附件中的历史数据的基期是 2010 年，十二五时期为 2011-2015 年；十三五时期为 2016-2020 年。附件主要包含总产值（GDP）及三次产业与部门的产值分布；总能耗及化石能源与非化石能源品种分布；碳排放量总量及产业与部门的分布；碳排放量相关各类能源的碳排放因子。

基于上述背景和附件信息，我们需要建立数学模型以解决下列三个问题：

问题一：建立一个能描述某区域经济、人口、能源消费量、碳排放量的状况和各部门（能源供应部门、工业消费部门、建筑消费部门、交通消费部门、居民生活消费、农林消费部门）的碳排放状况的指标和指标体系。通过分析对该区域碳排放量产生影响的各因素及其贡献进而研判该区域碳排放量以及经济、人口、能源消费量的现状，包括实现碳达峰和碳中和需要面对的主要挑战，以及为该区域双碳路径规划中差异化的路径选择提供依据。通过分析相关指标的变化，建立起各指标间的关联关系模型以及确定碳排放预测模型参数取值。

问题二：建立一个基于人口和经济变化的能源消费量预测模型，使得能源消费量与人口预测、经济（GDP）预测均相关联。运用该预测模型结合中国式现代化两个时间节点（2035 年和 2050 年）预测出某区域十四五至二十一五期间人口、经济和能源消费的变化。建立一个区域碳排放预测模型，不但使得碳排放量与人口、GDP、能源消费量预测相关联，而且使得碳排放量与各能源消费部门以及能源供应部门的能源消费量相关联，还使得碳排放量与各能源消费部门的能源消费品种以及能源供应部门的能源消费品种相关联。

问题三：分别对在无人干预自然情景、按时碳达峰与碳中和的基准情景和率先碳达峰与碳中和的雄心情景等不同情景下，并且满足 2035 年的 GDP 比 2020 年翻一番并且 2060 年的 GDP 比 2020 翻两番、2060 年生态碳汇的碳消纳量为基期碳排放量的 10%和 2060 年工程碳汇或碳交易的碳消纳量为基期碳排放量 10%等基本假设情况下制定在不同年份下 GDP、人口、能源消费量、能源利用率、非化石能源比重等目标以及相应的路径规划。

2 模型假设

1、由于新能源热力 2010-2020 年能源消耗量为 0，假设新能源热力在本文研究期内保持该能源消耗量。

2、假设国家对于人口的政策保持不变。

3、假设未来能源消费结构会发生变化，即非化石能源消费量比例会发生变化

4、假设碳排放量与人口、经济、能源消费量存在一定的线性或非线性关系

3 符号说明

符号	符号意义/单位
CO2	二氧化碳排放量/万 tce
P	人口/万人
E/GDP	单位 GDP 能耗
x	时间（基期是 2010 年）
δX	参数 X 对某基准年的相对变化率
$Growth_{yoy}(x)$	同比增长率
$Growth_{qoq}(x)$	环比增长率
$\overline{x}$	总体数据的均值
GDP	地区生产总值/亿元
Y	因变量
X	自变量
ε	因变量的预测残差
CO2Di	第 i 部门的碳排放量/万 tCO2
EVi	第 i 消费品种的能源消费量/万 tce

4 问题分析

4.1 总体技术流程图

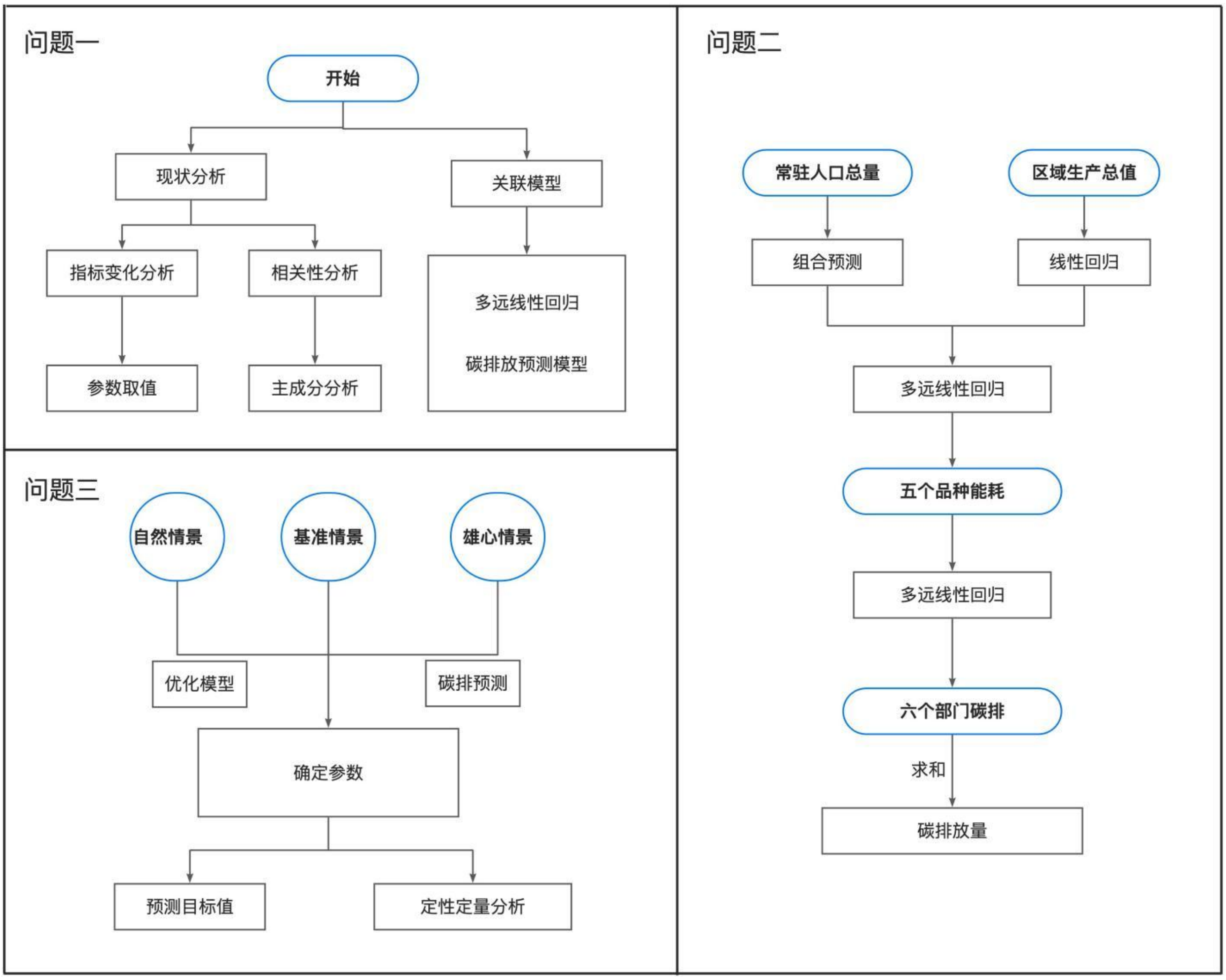

图 4-1 总体技术流程图

4.2 问题一分析

针对问题一，本问涉及分析区域碳排放量以及经济、人口、能源消费现状。

该问通过分析相关数据，科学、全面、合理的选取能够反应区域经济、人口、能源消耗量以及各个部门的碳排放量的量化指标，构成相应的指标体系。对各个主要指标进行相关性分析从而得到它们之间的相关关系，采用与碳排放量相关性强的指标来预测碳排放量的未来趋势。

通过统计分析得到十二五、十三五碳排放量的特征信息，采取图表展示等方法来实现数据可视化。然后结合主成分分析对影响碳排放量的各因素以及各个因素的贡献量进行量化。基于以上数据和 Kaya 模型对该区域碳达峰、碳中和条件进行量化进而研判该区域实现碳达峰、碳中和需要面对的主要挑战，使之成为双碳路径规划中差异化的路径选择的依据。

基于分析相关指标的同比与环比变化，以此反映出近年来各个指标变化趋势并有利于对各指标发展趋势进行预测。根据多元线性回归建立各项指标之间的关联关系模型，基于各项指标数据、双碳政策以及激素进步等多重效应对各项指标参数进行量化，以此得到碳排放量预测模型。

4.3 问题二分析

针对问题二，本问涉及区域碳排放量以及经济、人口、能源消费量的预测模型。

首先依据附件中的数据建立人口、经济（GDP）分别与时间的关系模型，通过 Python 软件进行参数求解，将求得的一次线性回归模型和多项式函数回归模型进行组合预测。然后使用多元线性回归模型分别求解各产业的能源品种消费量与人口、经济之间的关系。根据上述关系模型再结合中国式现代化的两个时间节点（2035 和 2050 年）对十四五（2021-2025 年）至二十一五（2056 年-2060 年）各产业的能源品种消费量进行预测。

其次，根据题目可知，碳排放量不仅与人口、GDP 和能源消费量相关，还与各能源消费部门、能源供应部门以及能源消费品种有关，因此我们需要先对每个能源消费及供应部门和能源消费品种的碳排放量进行预测。我们先建立起能源消费品种的碳排放量与人口、经济的预测模型，然后运用 Python 拟合参数。然后再分别建立起 6 个部门碳排放量与能源消费品种的预测模型，运用 Python 算出参数。六个部门的碳排放量之和等于区域碳排放总量。进而我们就可以得到区域碳排放量的预测模型。

最后通过我们得到的区域碳排放量以及经济、人口、能源消费量的预测模型，预测某区域十四五至二十一五期间人口、经济（GDP）和能源消费量、碳排放量的值。

4.4 问题三分析

针对问题三，本问涉及区域双碳（碳达峰与碳中和）目标与路径规划方法。

本题首先要设计自然情景、基准情景以及雄心情景三种情景。其中，假设在自然情景下的最晚碳达峰时间为 2050 年；基准情景的碳达峰时间为 2030 年；雄心情景的碳达峰时间为 2025 年。并且满足 2035 年的 GDP 比 2020 年翻一番并且 2060 年的 GDP 比 2020 翻两番、2060 年生态碳汇的碳消纳量为基期碳排放量的 10%和 2060 年工程碳汇或碳交易的碳消纳量为基期碳排放量 10%等基本假设，建立碳达峰的优化模型，在满足碳排量关于时间导数为 0 的情况下，求解出最晚碳达峰时间对应的人口数量，从而确定碳中和达成的时间。最后，计算出目标时间节点下的 GDP、人口以及能源消费量的目标值和能源利用效率以及提高化石能源消费比重的目标值。

5 模型建立与求解

5.1 问题一的求解

5.1.1 模型准备

（1）主成分分析

主成分分析（PCA）也称主分量分析，其主要思想是通过数据将更低维度投影，选取合适的投影方向，便可在降低数据维度的同时不丢失原始数据表达的信息，以实现降维。而实现该过程的主要困难为降低由于降维产生的信息损失，即找到投影的方法。降维后的新变量即为主成分，它们能充分解释原始数据变化[2]。

主成分分析是一种降维算法，它能将多个指标转换为少数几个主成分，这些主成分是原始变量的线性组合，且彼此之间互不相关，其能反映出原始数据的大部分信息。一般来说，当研究的问题涉及到多变量且变量之间存在很强的相关性时，我们可考虑使用主成分分析的方法来对数据进行简化。

（2）回归分析模型

一元线性回归是一个主要影响因素作为自变量来解释因变量的变化，在现实问题研究中，因变量的变化往往受几个重要因素的影响，此时就需要用两个或两个以上的影响因素作为自变量来解释因变量的变化，这就是多元回归亦称多重回归。当多个自变量与因变量之间是线性关系时，所进行的回归分析就是多元线性回归。

5.1.2 模型建立与求解

（1）指标与指标体系

人口指标：常驻人口总量和人口增长率是评价人口状况的重要指标，它们可以反映区域人口的规模和增长速度，对能源消费和碳排放有直接影响。

经济指标：地区生产总值（以下简称 GDP）是区域经济发展的重要标志且 GDP 等于第一、第二和第三产业增加值的总和。所以选择 GDP 和第一产业增加值、第二产业增加值、第三产业增加值作为衡量区域经济状况的主要指标，它能够综合反映一个区域的经济发展水平和经济活动的活跃程度。

能源消费指标：能源消费总量和能源消费结构（化石能源与非化石能源比例）以及第一产业的能耗、第二产业的能耗、第三产业的能耗、居民生活能耗是衡量能源消费状况的关键指标，它们直接影响碳排放量的大小和结构。

碳排放指标：总碳排放量、单位 GDP 碳排放量和各部门（能源供应部门、工业消费部门、建筑消费部门、交通消费部门、居民生活消费和农林消费部门）碳排放量是评估碳排放状况的主要指标，它们能够全面描述一个区域的碳排放水平和结构。

本文上述所选的指标能够描述某区域经济、人口、能源消费量和碳排放量的状况，也能够描述各部门（能源供应部门、工业消费部门、建筑消费部门、交通消费部门、居民生活消费、农林消费部门）的碳排放状况。

根据上述的指标分析可以得到如下表 5-1 所示的指标体系。

表 5-1 指标体系表

<table>
<tr><td rowspan="20">指标体系</td><td>一级指标</td><td>二级指标</td></tr>
<tr><td rowspan="2">人口指标</td><td>常驻人口总数</td></tr>
<tr><td>人口增长率</td></tr>
<tr><td rowspan="6">经济指标</td><td>地区生产总值</td></tr>
<tr><td>农林消费部门增加值</td></tr>
<tr><td>能源供应部门增加值</td></tr>
<tr><td>工业消费部门增加值</td></tr>
<tr><td>交通消费部门增加值</td></tr>
<tr><td>建筑消费增加值</td></tr>
<tr><td rowspan="6">能源消耗指标</td><td>总能耗</td></tr>
<tr><td>农林消费部门能耗</td></tr>
<tr><td>能源供应部门能耗</td></tr>
<tr><td>工业消费部门能耗</td></tr>
<tr><td>交通消费部门能耗</td></tr>
<tr><td>建筑消费部门能耗</td></tr>
<tr><td rowspan="3">碳排放指标</td><td>总碳排放量</td></tr>
<tr><td>农林消费部门碳排放量</td></tr>
<tr><td>工业消费部门碳排放量</td></tr>
</table>

		交通消费部门碳排放量
		建筑消费部门碳排放量
		居民生活消费碳排放量
		能源供应部门碳排放量

由上表我们可以看出指标体系能够描述各主要指标之间的相互关系，部分指标的变化（同比或环比）可以成为碳排放量预测的基础。具体的验证部分由下文的模型（5.1.3 中的（2））所给出。

（2）区域碳排放量状况

由题目我们知道区域碳排放总量等于国民经济各行业的碳排放量之和。本赛题将国民经济各行业分为能源供应部门、工业消费部门、建筑消费部门、交通消费部门、居民生活消费、农林消费部门。附件中只给出了工业消费部门、建筑消费部门、交通消费部门、居民生活消费、农林消费部门的碳排放量，所以我们想要知道碳排放总量，就要对能源供应部门的碳排放量进行量化。

由附件中的附表 7，我们可以知道某区域能源供应部门碳排放因子。由附表 4，我们可以知道某区域能源消费品种结构。根据题目给出的注释，我们可以知道碳排放量由各品种能源消费量与对应碳排放因子的乘积之和，所以我们可以得到能源供应部门的碳排放量，进而就可以求得 2010-2020 年中每年的区域碳排放总量。

至此我们对数据的量化基本结束，可以用得到的数据对区域十二五（2011-2015）和十二五（2016-2020 年）期间的碳排放量的状况进行分析。如下图 5-1 十二五和十三五期间碳排放量图所示。

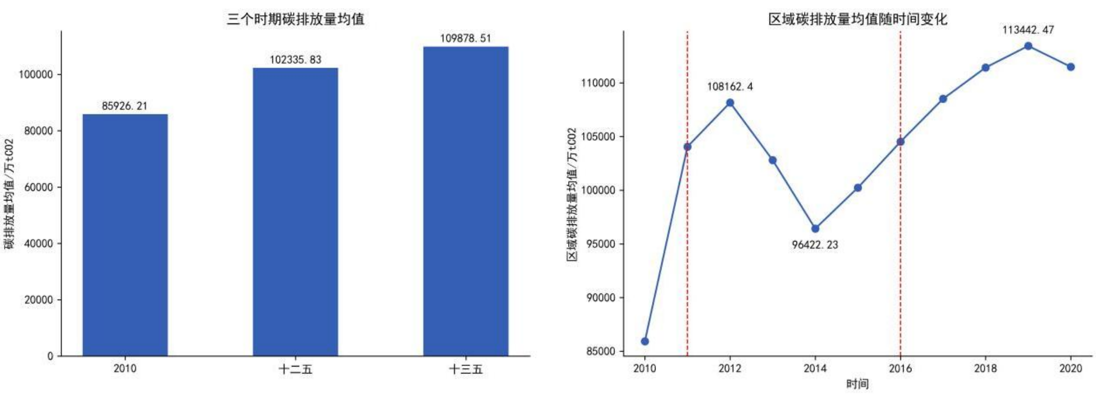

图 5-1 十二五和十三五期间碳排放量图

由上图 5-1，我们可以看出十二五时期年均碳排放量高于 2010 年，十三五时期年均碳排放量高于十二五时期年均排放量，高出约 7542.68 万 tCO_2。由 2010-2020 年年碳排放总量图可以看出年碳排放随着年份的增加是呈波动上升趋势的，其中在 2012 年、2014 年、2019 年分别出现一个拐点，值分别为 108162.4、96422.23、113442.47。值得注意的是，十二五时期的年碳排放总量波动较大，而十三五时期年碳排放总量波动较小，这说明碳排量增速明显放缓。

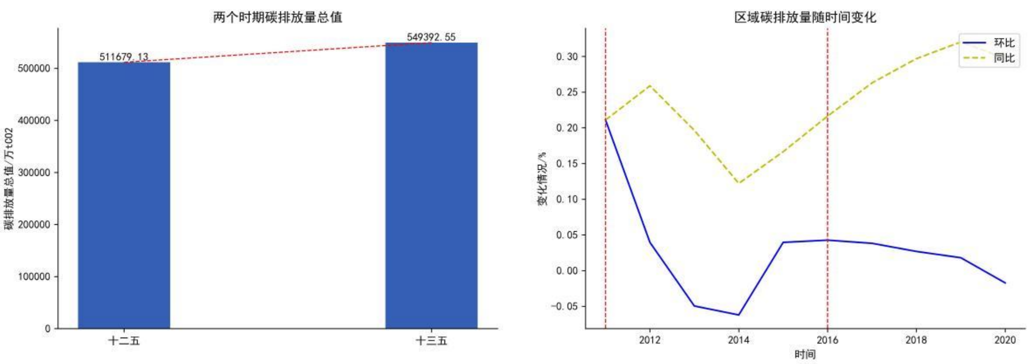

图 5-2 区域碳排放量同比、环比图

由上图 5-2 左边的子图可以看出十三五时期碳排放总量高于十二五时期碳排放总量，整体呈增长趋势。右边的子图为区域碳排放量的同比环比图，具体计算公式如下。

同比增长率：

$$Growth_{yoy}(x)=\frac{Value(x)-Value(2010)}{Value(2010)}\times 100\% \qquad (1)$$

环比增长率：

$$Growth_{qoq}(x)=\frac{Value(x)-Value(x-1)}{Value(x-1)}\times 100\% \qquad (2)$$

由图 5-2 右边的子图，我们可以看出同比的曲线均在 0 之上，这说明十二五时期和十三五时期的每一年的碳排放量均大于 2010 年的碳排放量。环比的曲线除 2013 年和 2014 年数值小于 0 其他年份的数值均大于 0，这说明在 2013 年和 2014 年碳排放量较前一年的碳排放量是有所减少的，而其他年份的碳排放量较前一年的碳排放量均增加。

另外，我们分别对十二五和十三五时期的能源消费品种进行统计。主要为煤炭消费量，油品消费量、天然气消费量、新能源热力、新能源电力、外地调入电、其他新能源。其中新能源热力在 2010-2020 年消费量为 0。根据上述数据，绘制饼图如下图 5-3 所示。

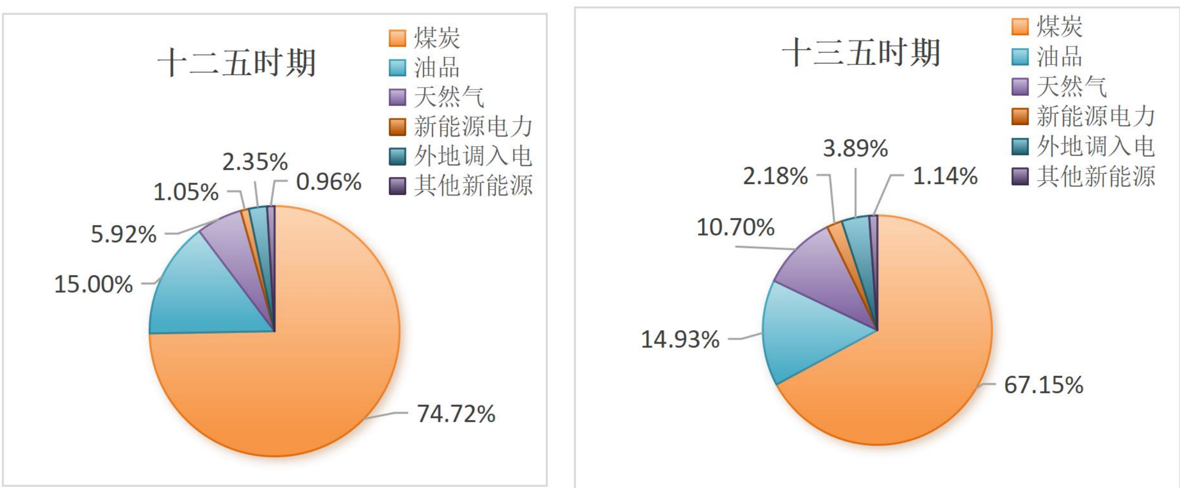

图 5-3 十二五、十三五时期消费品种的占比图

从上面的图我们不难看出，不管是十二五时期还是十三五时期，煤炭的消费量均是总

量的中最多的部分，不管煤炭的占比从十二五时期的74.72%降低到了十三五时期的67.15%。但电力、天然气这种优质能源的占比已经在提高。说明消费部门和供应部门已经将清洁的优质资源纳入消费范围。

总之，2010-2020 年，该区域能源消费变动的总体情况可以大致总结为：能源消费量增长迅速但有放缓趋势；能源消费结构较单一；能源消费和经济增长对化石能源尤其是对煤炭资源依赖性高。这些情况也很好的说明为什么十三五时期的碳排放量比十二五时期碳排放量高，但是十三五时期的较十二五时期的碳排放量较为缓慢了。

（3）主成分分析模型

Step1：读取数据并进行数据标准化。

此步骤的目的是标准化结构化指标的范围，因为 PCA 对于初始变量的方差非常敏感，如果初始变量的范围之间存在较大差异，则会造成很大变差，使用标准化可以将数据转换为可比较的尺度。我们采用零均值归一化。通过下列的转换公式，就可以将原始数据映射到均值为 0，标准差为 1 的分布上。

$$z=\frac{x-\bar{x}}{\sigma} \qquad (3)$$

其中，z 代表标准化后的值，x 代表原始数据，$\bar{x}$代表总体数据的均值，σ代表总体数据的标准差。

读取的数据主要是 2010 年-2020 年该区域每年的常驻人口总量、区域生产总值、第一产业生产总值、第二产业生产总值、第三产业生产总值、区域总能耗、第一产业的能耗、第二产业的能耗、第三产业的能耗。

Step2：求样本相关系数矩阵 R。

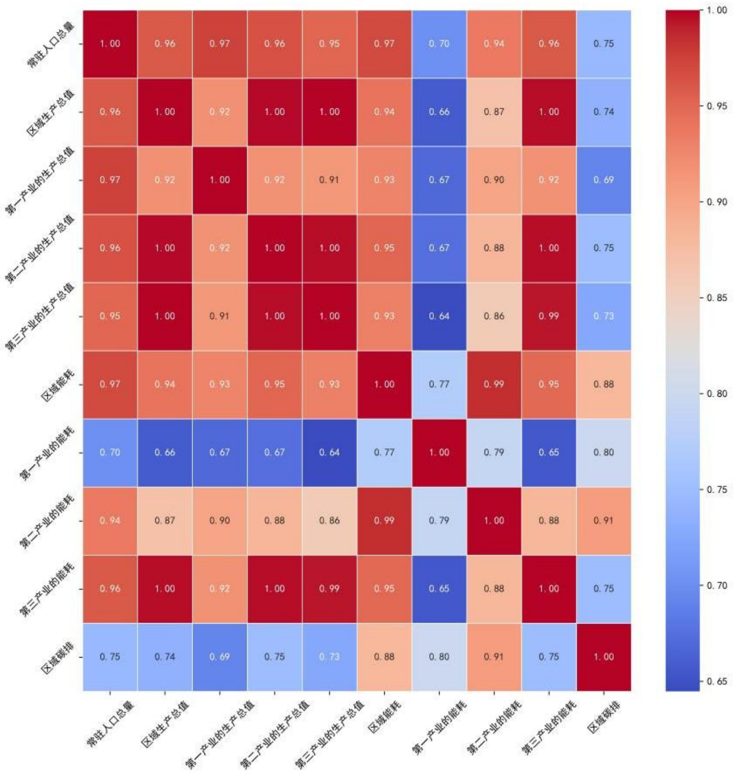

图 5-4 九个主成分之间的热力图

由上图 5-4 所示，热力图右侧的刻度展示了不同相关系数对应的颜色深浅。从图中可

以看出，区域生产总值和第二产业生产总值、第三产业生产总值、第三产业能耗之间的相关性较高，为 1，即存在很强的多重共线性。在进行特征工程时可以考虑剔除四者中的一个变量，以免导致因多重共线性造成的过拟合。

从热力图的最后一列或最后一行可以看出，第二产业能耗和区域碳排放量的相关系数相对最高，为 0.91。因此，在进行区域碳排放量建模时，可以考虑保留第二产业能耗这一变量。

Step3: 求相关系数矩阵对应的特征值 $\lambda_1 \geq \lambda_2 \geq \cdots \geq \lambda_{\mathrm{p}} > 0$ 及其对应的特征向量 $t_1, t_2 \ldots t_p$。

Step4: 按按主成分累计贡献率超过 80%确定主成分的个数 k，并写出此条件下的主成分表达式：

$$y_i = t_i^T x \quad i = 1,2\ldots k \qquad (4)$$

Step5: 对主成分分析结果做统计意义及实际意义的解释。

表 5-2 主成分对应特征向量、特征值、贡献率、累计贡献率

	主成分 1	主成分 2	主成分 3	主成分 4	主成分 5	主成分 6	主成分 7	主成分 8
常驻人口总量	0.345211	0.342804	0.333985	0.344304	0.340550	0.345183	0.265339	0.331575
区域生产总值	0.064884	0.223134	0.076799	0.188297	0.248296	-0.108053	-0.845737	-0.262086
第一产业的生产总值	0.263253	-0.276527	0.472168	-0.260323	-0.318467	0.199512	-0.389834	0.469355
第二产业的生产总值	0.207355	0.005346	0.679197	-0.063915	0.013535	-0.390303	0.245235	-0.512967
第三产业的生产总值	0.827645	-0.056152	-0.417517	0.212150	-0.190612	-0.133243	0.008659	-0.104158
区域能耗	0.004677	-0.297212	0.050557	0.079828	-0.524166	0.030089	0.034151	-0.154656
第一产业能耗	-0.271214	0.034984	0.143828	0.780136	-0.390368	0.126685	-0.030402	-0.051021
第二产业能耗	0.075806	-0.049827	-0.014314	-0.180269	0.022367	0.801451	-0.006868	-0.548242
第三产业能耗	0.000000	-0.813023	0.023191	0.285755	0.506755	0.000000	0.000000	0.000000
特征值	8.153157	0.548303	0.184769	0.094883	0.013326	0.004629	0.000817	0.000116
贡献率	0.905906	0.060923	0.020530	0.010543	0.001481	0.000514	0.000091	0.000013
累计贡献率	0.905906	0.966829	0.987359	0.997901	0.999382	0.999896	0.999987	1.000000

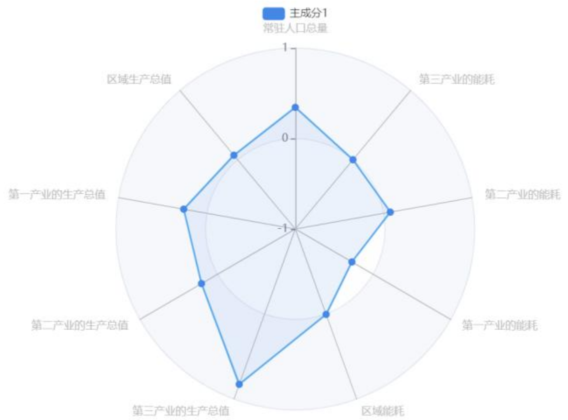

图 5-5 主成分雷达图

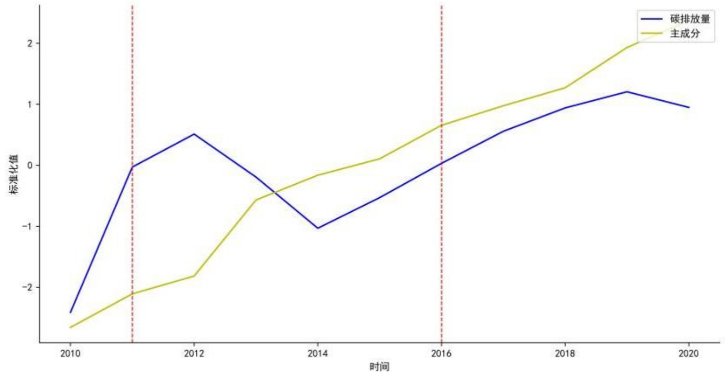

图 5-6 碳排放量与主成分预测关系图

①贡献率分析

根据上述表 5-2 所示，从贡献率来看，第一个主成分贡献率可以达到 0.905906，可以较好的涵盖大部分信息，因此选取一个主成分进行分析即可。第一个主成分：

$$y_i = 0.345211x_1 + 0.064884x_2 + 0.263253x_3 + 0.207355x_4 + 0.827645x_5 + 0.004677x_6 - 0.271214x_7 + 0.075806x_8 + 0.000000x_9 \quad (5)$$

②因子载荷矩阵分析

根据上述表 5-2 所示，从因子载荷矩阵来看，因子载荷范围处于-1 至 1 中间，从相似矩阵出发求解具有较好得优越性。第一主成分在变量x_5上有很高的正载荷，在x_7上有负载荷，可以看出这个主成分度量了受第三产业（建筑消费部门和交通消费部门）生产总值影响的区域碳排放量在区域碳排放总量中占的比重。

③雷达图分析

由图 5-5 雷达图可以看出，九个指标中有八个指标对主成分是正影响，其中第三产业生产总值对其正影响最大，第三产业分为建筑消费部门和交通消费部门。根据附件中的数据对 2010-2020 年各个产业碳排放量进行量化，可以看出第三产业逐渐成为碳排放增量的主要“贡献者”。从碳排放增长率来看，第三产业增速明显。2010-2020 年第一产业碳排放增幅为 38.24%，第二产业增幅为 22.76%，第三产业增幅为 84.90%。

④标准差分析

从表 5-2 可以看出，第一主成分特征值 8.153157，其余主成分特征值均低于 1，所以，我们将标准化后得到的碳排放量与从中计算得到的主成分预测值进行对比，结果如图 5-6 所示，从中不难看出，上述两者趋势一致。

（4）多元线性回归模型

多元线性回归分析是一种用于探究两个或两个以上自变量对一个因变量影响程度的统计分析方法，可以评定各个自变量对因变量影响的相对重要性，并可预测因变量的未来变化趋势，其数学模型可以表示为：

$$\mathrm{y_i} = \beta_0 + \beta_1 \mathrm{x_{1i}} + \beta_2 \mathrm{x_{2i}} + \cdots + \beta_j \mathrm{x_{ji}} + \varepsilon_i \quad (6)$$

上式中：y_i为因变量；x_{ji}为自变量；j 为自变量的数目；i 为第 i 个待计算的样本；β_j为待估计的未知参数，其中β_0为截距、β_1~β_j为各自变量的系数；ε_i是均值为零、方差为$\sigma^2>0$

的不可观测的随机变量，称为误差项。

在多元线性回归模型确定后，利用样本数据对模型的未知参数进行估计，可用最小二乘法。之后，需要对回归模型进行显著性检验，以确定此模型是否真正揭示了被解释变量和解释变量之间的关系。具体而言，可以利用拟合优度检验回归直线对观测值的拟合程度，判定系数R^2的值越接近于 1，模型对实际观测值的拟合效果越好、对因变量变异性的解释程度越高。

通过计算，可以得出区域碳排放量以及经济、人口、能源消费量间的多元线性回归模型为：

$$y_i = 401071.49 - 0.08x_{1i} - 64.36x_{2i} + 8.28x_{3i} \qquad (7)$$

具体的拟合模型如下图 5-7 所示，另外通过拟合优度检验回归直线对观测值的拟合程度，得到的判定系数R^2的值为 0.97，较接近 1，所以我们建立的上述模型对实际观测值的拟合效果较好。

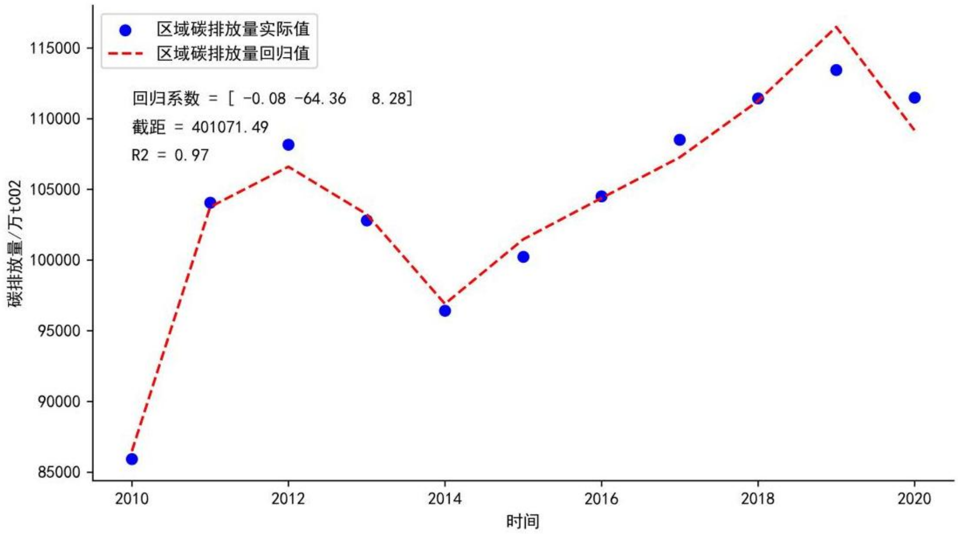

图 5-7 碳排放量多元线性回归模型图

5.1.3 模型结果分析

（1）需要实现双碳目标面对的挑战分析

基于之前数据量化，我们可以得到非化石、化石能源消费比重的数据并绘制成图如下图 5-8 所示，再结合上述模型的分析，可以得出碳排放总量逐年成稳定增长趋势，其中化石能源占比逐年降低，但是非化石能量比重增长速度过于缓慢，要达到区域双碳政策在未来时间内碳排放量应保持不再增长且碳排放量与碳汇消纳量相平衡的要求还相差甚远，并且 GDP 增长与碳排放有强烈的正相关关系，所以实现经济增长的同时控制碳排放量需要着重于提高能源利用率并降低化石能源比重。

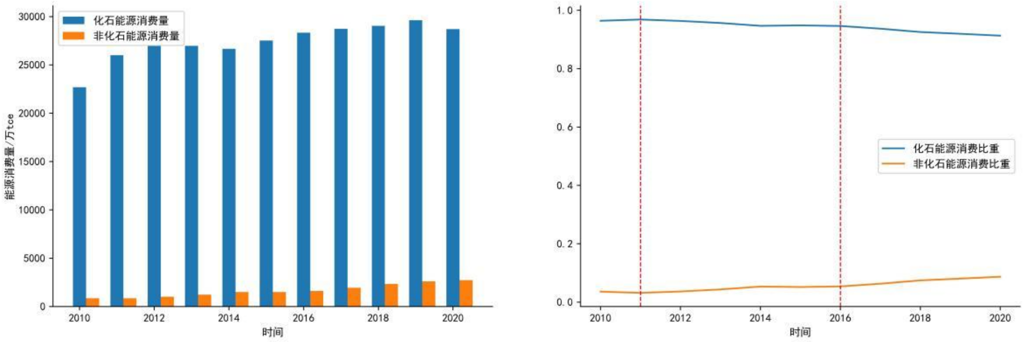

图 5-8 化石、非化石能源消费比重图

（2）对相关指标的变化（同比、环比）分析

针对 2010-2020 年区域碳排放量数据的同比、环比已经在上文中的图 5-2 进行了说明。现在我们主要分析经济（生产总值）、人口（常住人口总量）、能源消费量这三个主要的指标。根据附件中的数据以及同比、环比的计算公式，利用 Python 绘图如下图 5-9 所示。

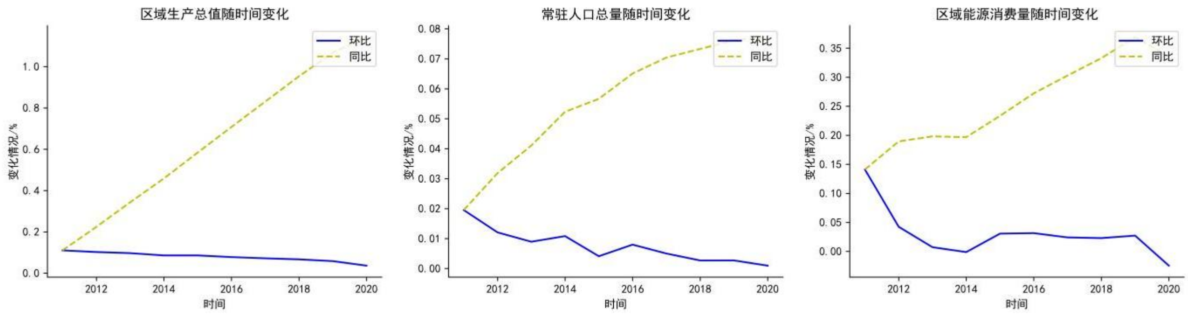

图 5-9 三大指标的同比、环比图

这三个指标的的同比、环比曲线的的走向大致相同，同比均呈上升趋势，环比均呈下降趋势。同比均大于 0，这便可以说明 2011-2020 每年区域生产总值、常驻人口总量、区域消费总量是较于 2010 年而言是增加的。针对环比，随着年份的递增，区域生产总值的环比是不断降低的，但是环比最低点仍大于 0，所以 2010-2020 年区域生产总值是逐年增加的。随着年份的递增，常驻人口总量的环比有增有减，但总体趋势是下降，但是最低点也是大于 0，所以 2010-2020 年区域常驻人口总量是逐年增加的。随着年份的递增，区域能源消费量环比有增有减，但总体趋势是下降的，在 2014 年和 2020 年区域能源消费量环比是低于 0 的，这说明 2014 年和 2020 年的区域年能源消费量较 2013 年和 2019 年是降低分。这三个指标中地区生产总值增速明显。2010-2020 年能源消费量增幅为 33.56%，常驻人口增幅为 7.73%，地区生产总值增幅为 114.29%。

5.2 问题二的求解

5.2.1 模型准备

（1）数据处理

从附件中提取出我们需要的基础数据，即 2010-2020 年间，该区域每年的能源消费总量、常驻人口总量、地区生产总值。具体数据如表 5-3 所示。

表 5-3 区域每年的能源消费总量、常驻人口总量、地区生产总值表

项目	单位	2010	2011	2012	2013	2014	2015	2016	2017	2018	2019	2020
能源消费量	万tce	23539.31	26860.03	27999.22	28203.10	28170.51	29033.61	29947.98	30669.89	31373.13	32227.51	31438.00
常驻人口	万人	7869.34	8022.99	8119.81	8192.44	8281.09	8315.11	8381.47	8423.50	8446.19	8469.09	8477.26
地区生产总值	亿元	41383.87	45952.65	50660.20	55580.11	60359.43	65552.00	70665.71	75752.20	80827.71	85556.13	88683.21

（2）组合预测

组合预测方法是对同一个问题，采用两种以上不同预测方法的预测。它既可是几种定量方法的组合，也可是几种定性的方法的组合，但实践中更多的则是利用定性方法与定量方法的组合。组合的主要目的是综合利用各种方法所提供的信息，尽可能地提高预测精度。理论和实践研究都表明，在诸种单项预测模型各异且数据来源不同的情况下，组合预测模型可能导致一个比任何一个独立预测值更好的预测值，组合预测模型能减少预测的系统误差，显著改进预测效果[3]。

组合预测有两种基本形式：一种是等权组合，即各预测方法的预测值按相同的权数组合成新的预测值。另一种是不等权组合，即赋予不同预测方法的预测值的权数是不一样的。这两种形式的原理和运用方法完全相同，只是权数的取定上有所区别。根据已进行的预测结果，采用不等权组合的组合预测法结果较为准确。

5.2.2 模型建立与求解

（1）人口预测模型

回归分析是对具有相关关系的变量之间的数量变化规律进行测定，研究某一随机变量（因变量）与其他一个或几个普通变量（自变量）之间的数量变动关系，并据此对因变量进行估计和预测的统计分析方法。

一般而言，事物的发展变化并非单一趋势，对于人口预测也是如此，人口规模变化受多种因素影响，单一模型只能模拟人口在某个时段的变化发展，而要对人口发展趋势进行相对合理的预测，就要根据人口历史发展过程，综合采用多种回归模型。

人口预测采用线性模型和多项式函数模型这两种回归方法分别进行预测，假设时间为x，每年常驻人口总量为P。Python 回归结果如表 5-4 和图 5-10 所示。

表 5-4 回归模型预测表

回归模型	回归方程	拟合度R^2
线性回归	$P = 57.8694x - 108334.3790$	0.9168
多项式函数回归	$P = -6.0889x^2 + 24596.2482x - 24830690.1208$	0.9959

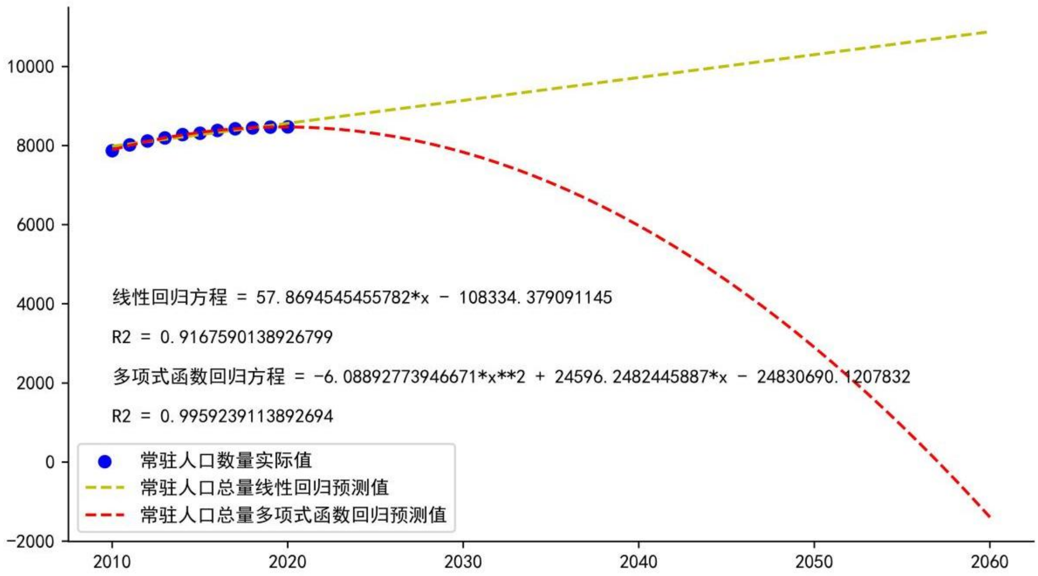

图 5-10 回归模型预测图

由于组合预测模型比单一模型更具有稳定性[4]，因此对两个模型的预测值按三种比例进行组合，得到最终的预测值。这三种比例分别为，情境 1：线性回归预测值占 0.9，多项式函数回归预测值 0.1；情境 2：线性回归预测值占 0.8，多项式函数回归预测值 0.2；情境 3：线性回归预测值占 0.6，多项式函数回归预测值 0.4。如下图 5-11 所示。

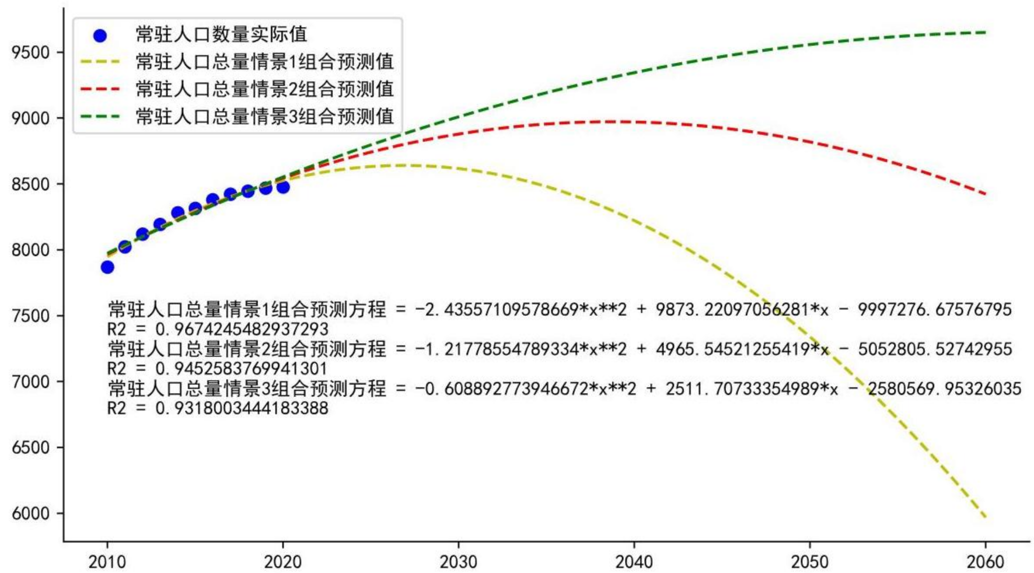

图 5-11 人口预测图

从上图我们可以看出，随着时间的增加，情境 1 的区域常驻人口总量是逐年递增的。情景 2 的区域常驻人口总量是先增加再缓慢减少，在 2039 年达到峰值。情境 3 的区域常驻人口总量也是先增加后减少，但它减少的速率明显比情境 2 的快。再结合中国式现代化的两个时间节点（2035 和 2050），所以综上所述，拟合效果最好的是情境 2。

（2）经济预测模型

经济预测采用线性回归模型进行预测，假设时间为 x，每年经济（GDP）总量为 GDP，python 回归结果如图 5-12 所示。

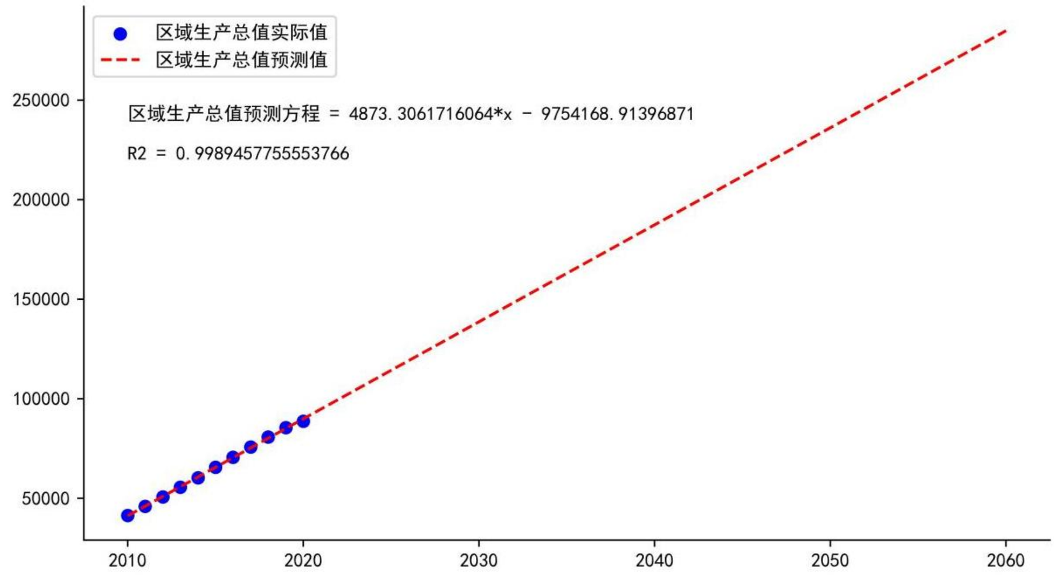

图 5-12 经济预测图

从上图我们可以看出，随着时间的增加，区域经济总量是逐年递增的。且我们 2020 年为基期，计算每五年的经济增长率，如下表 5-5 所示。可以看出经济增长率均大于 0，但区域生产总值增长率在逐年减少。说明区域生产总值的增长变缓。

表 5-5 每五年的区域生产总值增长率表

年份区间	2021-2025	2026-2030	2031-2035	2036-2040	2041-2045	2046-2050	2051-2055	2056-2060
区域生产总值增长率	20.57%	16.36%	13.58%	11.61%	10.14%	9.00%	8.09%	7.35%

（3）能源消费量的预测模型

通过上述的计算，我们已经获得了人口关于时间的预测模型，经济关于时间的预测模型，以及获取到了人口和经济 2021-2060 年的数据，然后基于多元线性回归算法，对能源消费量的预测模型进行构建。利用这一算法，确定模型当中因变量与多个自变量之间的关系，通过上述论述得出，模型中的自变量包括人口、经济（GDP）、时间等，将各个变量代入到下述公式当中，实现对能源消费量的预测模型的构建：

$$Y = X\beta + \varepsilon \qquad (8)$$

式中：Y 为因变量，在模型中，因变量为能源消费量；X 为自变量，即上述所说的数据；β为回归系数；ε为各个因变量的预测残差。公式（8）中，X 又可以用矩阵形式表示：

$$X = \begin{bmatrix} 1 & x_{11} & \cdots & x_{1m} \\ 1 & x_{21} & \cdots & x_{2m} \\ \vdots & \vdots & \vdots & \vdots \\ 1 & x_{n1} & \cdots & x_{nm} \end{bmatrix} \qquad (9)$$

利用式（8）和式（9）对因变量的预测结果进行求解，得出的结果即为能源消费量随

时间 x 变化的预测结果。Python 回归结果如图 5-13 所示。

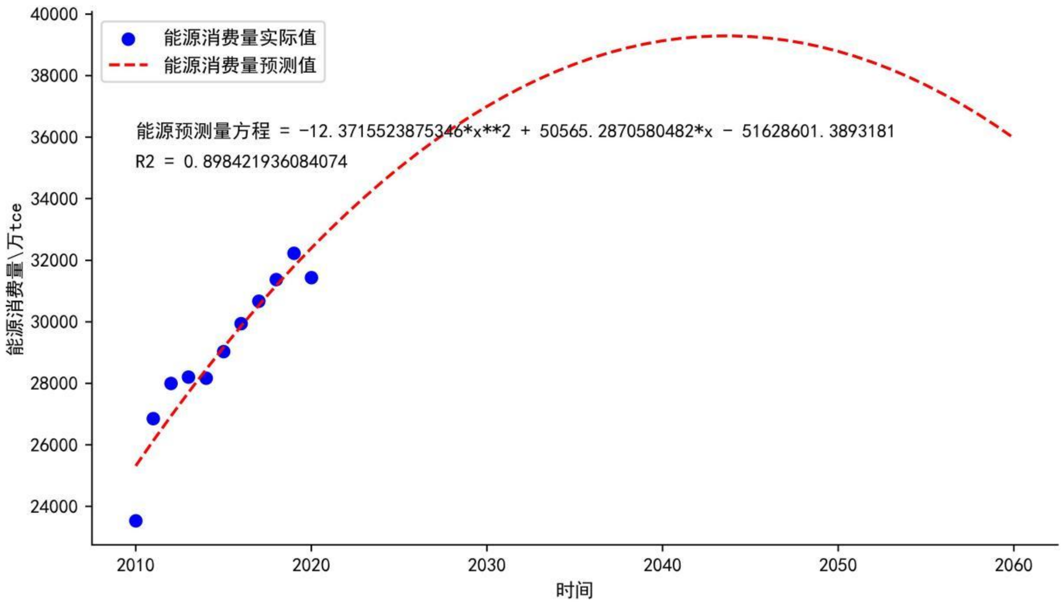

图 5-13 能源消费量预测图

通过 Python 对回归模型参数的求解，我们得到了基于时间 x 的预测区域能源消费量的模型，具体函数为：

$$Y = -12.37155x^2 + 50565.28706x - 51628601.38931 \qquad (10)$$

另外函数的拟合度$R^2 = 0.8984$，说明我们得到的基于时间的预测区域能源消费量的函数拟合度较好。

（4）区域碳排放量预测模型

根据题目可知，碳排放量与人口、GDP 和能源消费量相关，还与各能源消费部门、能源供应部门以及能源消费品种有关，因此我们需要对每个能源消费及供应部门和能源消费品种的碳排放量进行预测[5]。

Step1：拟合人口和经济关于时间的预测函数

在 5.2.2 中的（1）和（2）中我们已经建立了人口预测模型和经济预测模型。

Step2：拟合五个能源消费品种的能源消费量与人口、经济的多元线性回归函数

能源消费量从品种分布看，包含化石能源消费（有碳排放）和非化石能源消费（无碳排放）。化石能源包括煤炭、石油、天然气。

能源消费量从加工转换过程看，包含一次能源（指未经加工转换的能源，如煤炭、石油、天然气、太阳能、风能、水能、核能、生物质能、地热能等）和二次能源（指经过加工转换的能源，如电能、热能、冷能、光伏、风电、水电、核电等）。

根据题目要求，我们选取的五个能源消费品种分别为：一次能源中化石能源消费量、二次能源中化石能源消费量、二次能源消费量、化石能源发电的能源消费量、非化石能源发电的能源消费量。分别设为 EV1、EV2、EV3、EV4、EV5。

对以上这五个能源消费品种，拟合其与人口、经济的多元线性回归函数，具体形式如下所示：

$$EVi = a_{1i}P(x) + a_{2i}GDP(x) + \varepsilon_i \qquad (10)$$

$$EVi = b_{1i}x^2 + b_{2i}x + \gamma_i \quad (11)$$

其中，i 的取值范围为 1、2、3、4、5，P(x)代表常驻人口总量关于时间的函数，GDP(x)代表区域生产总值关于时间的方程，x 代表时间。通过 Python，我们得到了上述每一个多元线性回归函数的参数。具体结果如下表 5-6 所示。

表 5-6 每一个多元线性回归函数的参数

i	EVi	a_{1i}	a_{2i}	ε_i
1	一次能源中化石能源消费量	-0.03661372	11.88155067	-68502.65465777054
2	二次能源中化石能源消费量	-0.00017163	0.3853916	-2877.963175785708
3	二次能源消费量	0.06141583	-2.10788562	14766.760473862283
4	化石能源发电的能源消费量	-0.04767989	8.27075935	-53097.66348118112
5	非化石能源发电的能源消费量	0.06141583	-2.10788562	14766.760473862283

对上面的五个表达式展开化简合并，可以得到五个关于时间的函数，如下式所示。

$$\mathrm{EV1(x)} = -14.4691806933183\mathrm{x}^2 + 58819.9472059816\mathrm{x} - 59746531.1947759 \quad (12)$$

$$\mathrm{EV2(x)} = -0.469324319939295\mathrm{x}^2 + 1912.84299907292\mathrm{x} - 1948512.64569116 \quad (13)$$

$$\mathrm{EV3(x)} = 2.56695265023286\mathrm{x}^2 - 10167.5032464289\mathrm{x} + 10066442.5519702 \quad (14)$$

$$\mathrm{EV4(x)} = -10.0720112056623\mathrm{x}^2 + 40836.4708214353\mathrm{x} - 41378558.5840887 \quad (15)$$

$$\mathrm{EV3(x)} = 2.56695265023286\mathrm{x}^2 - 10167.5032464289\mathrm{x} + 10066442.5519702 \quad (16)$$

Step3：拟合六个部门的碳排放量与五个能源消费品种的能源消费量的多元线性回归函数

本题中的各部门包括：工业消费部门、交通消费部门、建筑消费部门、农林消费部门、居民生活消费、能源供应部门。

对以上这 6 个部门碳排放量，拟合其与五个能源消费品种的能源消费量的多元线性回归函数，具体形式如下列公式所示。

$$CO2Di = c_{1i}EV1 + c_{2i}EV2 + c_{3i}EV3 + c_{4i}EV4 + c_{5i}EV5 + \alpha_i \quad (17)$$

$$CO2Di = d_{1i}x^2 + d_{2i}x + \beta_i \quad (18)$$

其中 i 的取值为 1-6，$CO2Di$分别代表农林消费部门的碳排放量、能源供应部门的碳排放量、工业消费部门的碳排放量、交通消费部门的碳排放量、建筑消费部门的碳排放量、居民生活消费部门。CO2 代表碳排放量，x 代表时间。

通过 Python，我们得到了上述每一个多元线性回归函数的参数。具体结果如下表 5-7 所示。

表 5-7 每一个多元线性回归函数的参数

i	$CO2Di$	d_{1i}	d_{2i}	β_i
1	农林消费部门的碳排放量	-0.562377613477402	2300.13221115463	- 2350240.04056542
2	能源供应部门的碳排放量	-21.8178716322244	88323.3791481631	-89333939.0350974
3	工业消费部门的碳排放量	-0.457711263758086	2093.55628298492	-2355761.82721045
4	交通消费部门的碳排放量	1.18003854878224	-4479.76279145293	4239714.58311904

5	建筑消费部门的碳排放量	1.61975061546172	-6192.82019374946	5907905.53023828
6	居民生活消费部门的碳排量	-7.81202025116154	32010.5397901728	-32745958.2600073

Step4：计算总碳排放量

至此，我们已经得到了 6 个部门碳排放量分别与时间的函数。现在我们再利用这 6 个函数来建立区域碳排放总量的预测模型，如下式（19）（20）所示。

$$CO2 = CO2D1 + CO2D2 + CO2D3 + CO2D4 + CO2D5 + CO2D6 \quad (19)$$

$$CO2 = e_1 x^2 + e_2 x + \delta \quad (20)$$

其中，CO2 代表区域碳排总量。

通过计算，我们可以得到，e_1 的取值为：-27.8501915963774，e_2 的取值为：114055.024447273，δ的取值为：-116638279.049523。将求得的参数带到式（20）中，可以得到区域碳排放总量关于时间的预测函数，即：

$$CO2 = -27.8501915963774x^2 + 114055.024447273x - 116638279.049523 \quad (21)$$

另外区域碳排放总量关于时间的预测函数图如 5-14 所示。

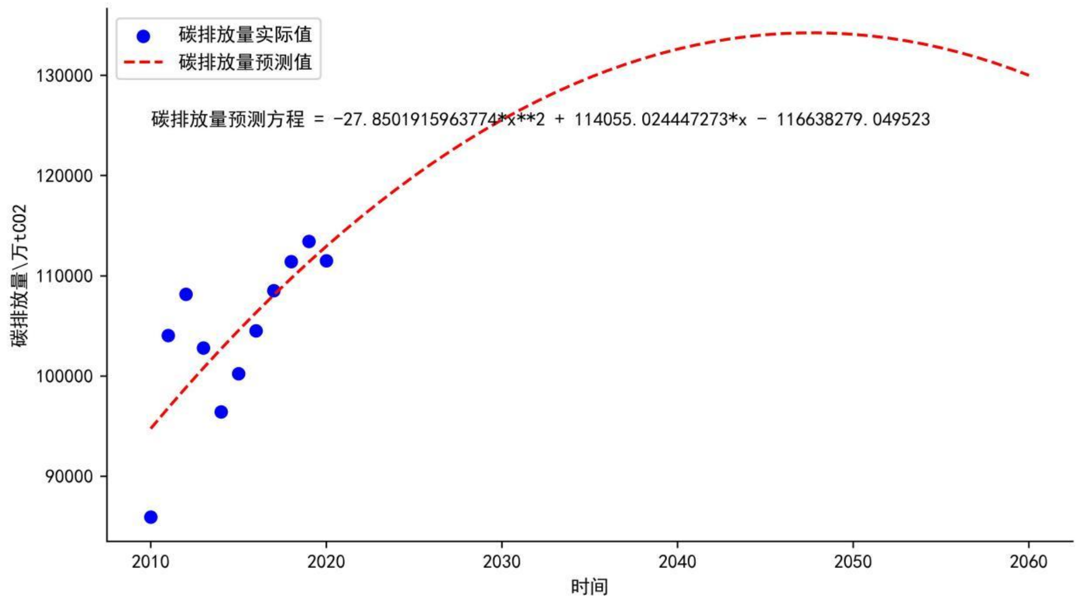

图 5-14 区域碳排放量的预测图

Step5：得到碳排放量的预测值

通过 Step4 我们得到了区域碳排放量的预测函数，进而我们就可以得到区域碳排放量的预测值，具体的预测值在 5.2.3 中的（4）有详细说明。

5.2.3 模型结果分析

（1）人口预测结果

根据上述我们所搭建的人口预测模型，并结合中国式现代化的两个时间节点（2035 和 2050），可以预测出某区域十四五（2021-2025 年）至二十一五（2056-2060 年）期间的人口。部分数据如表 5-8 所示，2021-2060 全部数据见附录。

表 5-8 人口预测结果表

时间	常驻人口总量情景 2 组合预测值	时间	常驻人口总量情景 2 组合预测值
2021	8588.126119	...	...
2022	8630.164362	2052	8758.771075
2023	8669.767033	2053	8725.306613
2024	8706.934133	2054	8689.406581
2025	8741.665663	2055	8651.070977
2026	8773.961621	2056	8610.299802
2027	8803.822008	2057	8567.093056
2028	8831.246823	2058	8521.450739
2029	8856.236068	2059	8473.372851
2030	8878.789742	2060	8422.859392

（2）经济预测结果

根据上述我们所搭建的经济预测模型，并结合中国式现代化的两个时间节点（2035 和 2050），可以预测出某区域十四五（2021-2025 年）至二十一五（2056-2060 年）期间的经济。部分数据如表 5-9 所示，2021-2060 全部数据见附录。

表 5-9 经济预测结果表

时间	区域生产总值预测值	时间	区域生产总值预测值
2021	94782.85885	...	...
2022	99656.16502	2052	245855.3502
2023	104529.4712	2053	250728.6563
2024	109402.7774	2054	255601.9625
2025	114276.0835	2055	260475.2687
2026	119149.3897	2056	265348.5749
2027	124022.6959	2057	270221.881
2028	128896.002	2058	275095.1872
2029	133769.3082	2059	279968.4934
2030	138642.6144	2060	284841.7995

（3）能源消费预测结果

根据上述我们所搭建的能源消费预测模型，并结合中国式现代化的两个时间节点（2035 和 2050），可以预测出某区域十四五（2021-2025 年）至二十一五（2056-2060 年）期间的能源消费量。部分数据如表 5-10 所示，2021-2060 全部数据见附录。

表 5-10 能源消费预测结果表

时间	能源消费量预测值	时间	能源消费量预测值
2021	32967.9497	...	...
2022	33515.05046	2052	38202.5939
2023	34037.40811	2053	37957.91531
2024	34535.02265	2054	37688.4936
2025	35007.8941	2055	37394.3288

2026	35456.02243	2056	37075.42088
2027	35879.40766	2057	36731.76987
2028	36278.04979	2058	36363.37575
2029	36651.94881	2059	35970.23852
2030	37001.10473	2060	38202.5939

（4）区域碳排放量预测结果

根据上述我们所搭建的区域碳排放量预测模型，并结合中国式现代化的两个时间节点（2035 和 2050），可以预测出某区域十四五（2021-2025 年）至二十一五（2056-2060 年）期间的区域碳排放量。部分数据如表 5-11 所示，2021-2060 全部数据见附录。

表 5-11 区域碳排放预测结果表

时间	区域碳排放量预测值	时间	区域碳排量预测值
2021	114460.9443	...	...
2022	115917.6441	2052	133717.9606
2023	117318.6436	2053	133447.9486
2024	118663.9426	2054	133122.2362
2025	119953.5413	2055	132740.8233
2026	121187.4396	2056	132303.7101
2027	122365.6375	2057	131810.8965
2028	123488.135	2058	131262.3826
2029	124554.9322	2059	130658.1682
2030	125566.0289	2060	129998.2535

5.3 问题三的求解

5.3.1 模型准备

（1）优化模型

优化模型三要素：决策变量、目标函数、约束条件。

决策变量：是决策者可以控制的因素，例如根据不同的实际问题，决策变量可以选为产品的产量、物资的运量及工作的天数等。

目标函数：是以函数形式来表示决策者追求的目标。例如目标可以是利润最大或成本最小等。

约束条件：是决策变量需要满足的限定条件。

（2）碳达峰碳中和需要达到的基本要求

基于问题二中对碳总排放量预测，存在碳总排放量关于时间的预测模型，碳总排放量的预测模型为：

$$CO2 = CO2D1 + CO2D2 + CO2D3 + CO2D4 + CO2D5 + CO2D6 \quad (22)$$

$$CO2 = e_1x^2 + e_2x + \delta \quad (23)$$

应双碳政策要求要达到碳达峰，则需满足条件：

$$\frac{dCO2(x)}{dx} = 0 \quad (24)$$

要达到碳中和，需满足条件：

$$\begin{cases} \dfrac{dCO2(x)}{dx} << 0 \\ \dfrac{EV2(x)+EV3(x)}{EV1(x)+EV2(x)+EV3(x)} > 80\% \end{cases}$$

（25）

基于多情景碳排放量核算方法应满足基本假设如下：

①2035 年的 GDP 比基期（2020 年）翻一番；2060 年比基期翻两番；

②2060 年生态碳汇的碳消纳量为基期碳排放量的 10%；

$$H_{生态吸纳} = 10\% \times P_{碳排2020} \quad (26)$$

③2060 年工程碳汇或碳交易的碳消纳量为基期碳排放量 10%。

$$H_{碳交} = 10\% \times P_{碳排2020} \quad (27)$$

5.3.2 模型建立与求解

（1）自然发展情景

基于上述各个函数，建立碳达峰实现时间的优化模型， 并由问题二中的碳排放量的多元回归模型，对其进行求导有：

$$\frac{dCO2(x)}{dx} = -55.7x + 114055.02 = 0 \quad (28)$$

为满足碳达峰要求，在其导函数等于零时，则有

$$x = 2047$$

所以对于无人为干预的自然情景下，至少在 2047 年才能够能够实现碳达峰。

基于碳中和要求，建立实现碳中和关于时间的条件优化模型，其中目标函数为：

$$\min \quad x_{碳中和} \quad (29)$$

约束条件为：

$$\begin{cases} \dfrac{dCO_2(x)}{dx} < 0 \\ \dfrac{EV2(x)+EV3(x)}{EV1(x)+EV2(x)+EV3(x)} > 80\% \\ x_{碳中和} > 0 \end{cases}$$

（30）

基于问题二中碳排放总量关于人口与 GDP 的多元回归模型，并结合能量消费关于时间的一元函数，则有：

$$x_{碳中和} = 2090 \quad (31)$$

基于 2010 至 2060 的数据进行拟合，可以建立经济关于时间的一元回归模型 $f_{GDP}(x)$，

由于在无人为干预的自然情况下，人口与能源消费量与时间的关系与问题二中相同，利用问题二的关系函数并代入时间节点可求得在自然情景时的人口、经济以及消费量在各个时间节点上目标值见表 5-12。

表 5-12 自然情景下各个时间节点的人口、经济与能源消费量的目标值

时间	区域生产总值	常驻人口总量	能源消费量
2025	65211.4421	8284.749672	29157.57457
2030	96390.19358	8543.652306	32555.72345
2035	127568.9451	8741.665663	35335.29268
2050	221105.1995	8970.370068	39962.52244
2060	283462.7024	8818.393285	39954.44402

（2）按时碳达峰与碳中和的基准情景

对于按时碳达峰与碳中和的基准情景，我国需要在 2030 年实现碳达峰，并在 2060 年实现碳中和目标。

基于上述条件，即有：

$$\frac{dCO2(x)}{dx}\bigg|_{x=21}=0 \qquad (32)$$

由于当 $t=2030$ 时，GDP 已知，则 2030 年的人口 $f_{people}(2030)$ 为定值，根据问题二中能源消耗量与人口之间的关系可以得到碳排放量关于人口的关系式 $f_{CO_2}(x)$。再根据碳中和的条件建立相关的条件约束，即有：

$$\begin{cases} \dfrac{dCO2(x)}{dx}\bigg|_{t=51}<0 \\ \dfrac{EV2(x)+EV3(x)}{EV1(x)+EV2(x)+EV3(x)}\bigg|_{t=51}<80\% \end{cases} \qquad (33)$$

根据式（32）可以推出 $f_{people}(2060)$ 的取值范围 w_1，由式（33）可推出 $f_{people}(2060)$ 的第二个取值范围 w_2，则 2060 人口只需要取介于 w_1 与 w_2 的任意值即可。基于 2010 年到 2020 年的数据以及上诉的 2030 年及 2060 年的总人口量的取值则可求出人口与时间的关系式 $f_{people}(x)$ 和经济与时间的关系 $f_{GDP}(x)$。将各个时间节点代入上述关系式中，则可得到各个各个节点的目标值见表 5-13。

表 5-13 基准情景下各个时间节点上关于人口、经济和能源消耗量的目标值

时间	区域生产总值	常驻人口总量	能源消费量
2025	65211.4421	8298.706692	29299.3652
2030	96390.19358	8517.462686	32289.66074

2035	127568.9451	8602.917257	33925.73513
2050	221105.1995	8059.47244	30708.6314
2060	283462.7024	7030.668784	21792.78981

（3）提前实现双碳政策

假设双碳政策提前 T 年能够实现，则建立关于实现双碳政策的优化模型如下所示。

$$\begin{cases} \dfrac{dCO2(x)}{dx}\Big|_{x=21-T}=0 \\ \dfrac{dCO2(x)}{dx}\Big|_{x=51-T}<0 \\ \dfrac{EV2(x)+EV3(x)}{EV1(x)+EV2(x)+EV3(x)}\Big|_{x=51-T}<80\% \end{cases} \tag{34}$$

基于问题二中碳排放量与人口与经济的相关模型，在满足碳达峰的约束条件下，由式一可以推出在 $x=21-T$ 时 $f_{people}(21-T)$ 的值，再由碳中和的满足条件下可推出 $f_{people}(2060)$ 的数值，然后借助回归模型得到人口与经济关于时间的函数关系，再将各个节点的时间代入函数关系式，则可以得到各个节点上人口、经济与能源消耗的目标值见表 5-14。

表 5-14 雄心情景下各个节点人口、经济与能源消耗的目标值

时间	区域生产总值	常驻人口总量	能源消费量
2025	65211.4421	8309.889291	29412.97022
2030	96390.19358	8496.479122	32076.48683
2035	127568.9451	8491.749705	32796.37396
2050	221105.1995	7329.645962	23294.25848
2060	283462.7024	5598.31389	7241.367439

5.3.3 模型结果分析

通过 python 求解分别求出关于人口、经济和能源消耗量的数据结果，通过对下列数据进行可视化分析。

（1）自然情景

通过求解关于自然条件下的优化模型得到其能源利用率以及非化石嫩管消费比重如表 5-15 所示。

表 5-15 自然情景下各个时间点能源利用率及非化石能源占比情况

时间	能源利用效率	非化石能源消费比重
2025	3. 432957369	0. 131431806
2030	3. 403254485	0. 168483333
2035	3. 394624854	0. 207376221
2050	3. 476549597	0. 351890105
2060	3. 634592839	0. 49529972

由表 5-15 中可以发现在无人干扰的自然情景下，需要在 2050 年才能使碳达峰，但是截至到 2060 年，其能源利用效率为 3.63%且非化石能源占比为 0.49，这与碳中和目标还有很多的差距，从 2025 年到 2060 能源增长率仅为 0.2%，为了实现双碳政策政府必须施加一定的手段进行干预，提高能源利用率和非化石能源比重最重要的在于提高科技创新和提高碳二次回收率。

（2）按时碳达峰与碳中和的基准情景

通过求解关于自然条件下的优化模型得到其能源利用率以及非化石能源消费比重如表 5-16 所示。

表 5-16 基准情景下各个时间点能源利用率及非化石能源占比情况

时间	能源利用效率	非化石能源消费比重
2025	3. 482059636	0. 143937123
2030	3. 514017354	0. 200981685
2035	3. 598350445	0. 274530658
2050	4. 49777489	0. 78644789
2060	9. 200224998	0. 8652452

通过观察在按时碳达峰与碳中和基准情况下，在 2030 年实现碳达峰，2026 年实现碳中和。在 2030 年能源利用率需要达到 3.51%，非化石能源消费比重目标需要达到了 0.2，在 2060 年非化石能源占比目标为 0.98，满足碳中和条件。

（3）率先碳达峰与碳中和的雄心情景

在率先碳达峰与碳中和的雄心情景下，假设提前 T 年实现双碳政策，通过优化模型求解答得到相关数据如表 5-17 所示。

表 5-17 雄心情景下各个时间点能源利用率及非化石能源占比情况

时间	能源利用效率	非化石能源消费比重
2025	3. 524446808	0. 154732247
2030	3. 619412757	0. 231905198
2035	3. 822682909	0. 348477775
2050	9. 012342005	0. 8122264
2060	20. 8707524	0. 952353

通过分析表 5-17 中数据可以发现在雄心情景将提前 5 年实现碳达峰与碳中和，其中预计在 2050 年达到碳中和目标，在实现碳达峰的 2025 年此时的能源利用率目标为 3.52%，而非化石能源消费占比需要达到 0.154，在 2060 年要实现碳中和，其能源利用率应达到 20.87%，非化石能源消费占比需要达到 0.95，相较于前两种情况雄心情景面临挑战更大。

为了能够更加直观综合的展示三种情景下各个指标的发展路径，下面分别对人口、经济与能源消费量做了数据可视化处理，如图 5-15 至 5-17 所示。

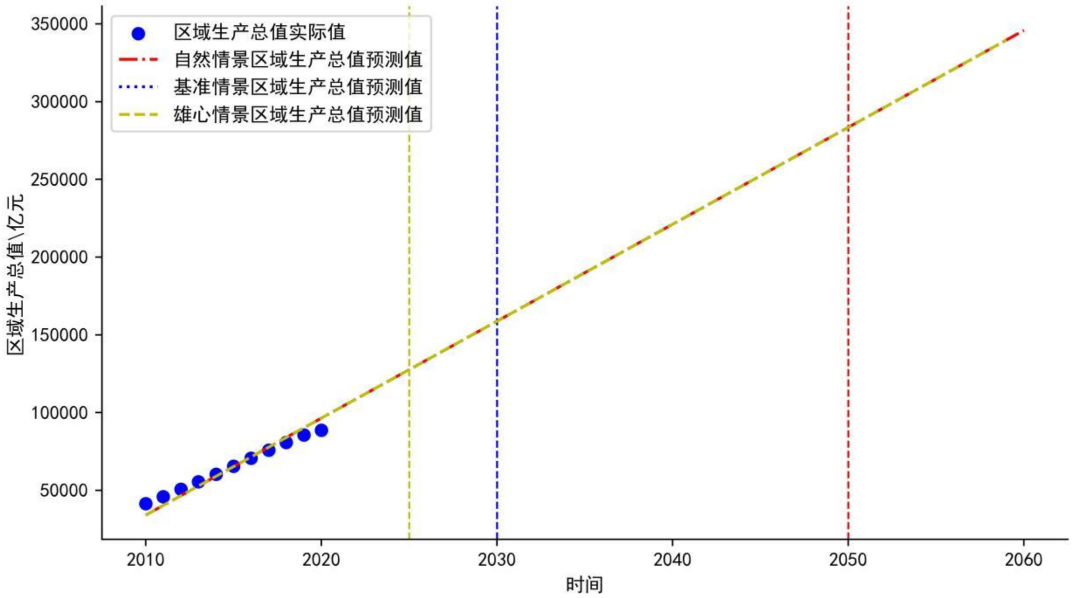

图 5-15 三种设计情景下 GDP 的趋势图

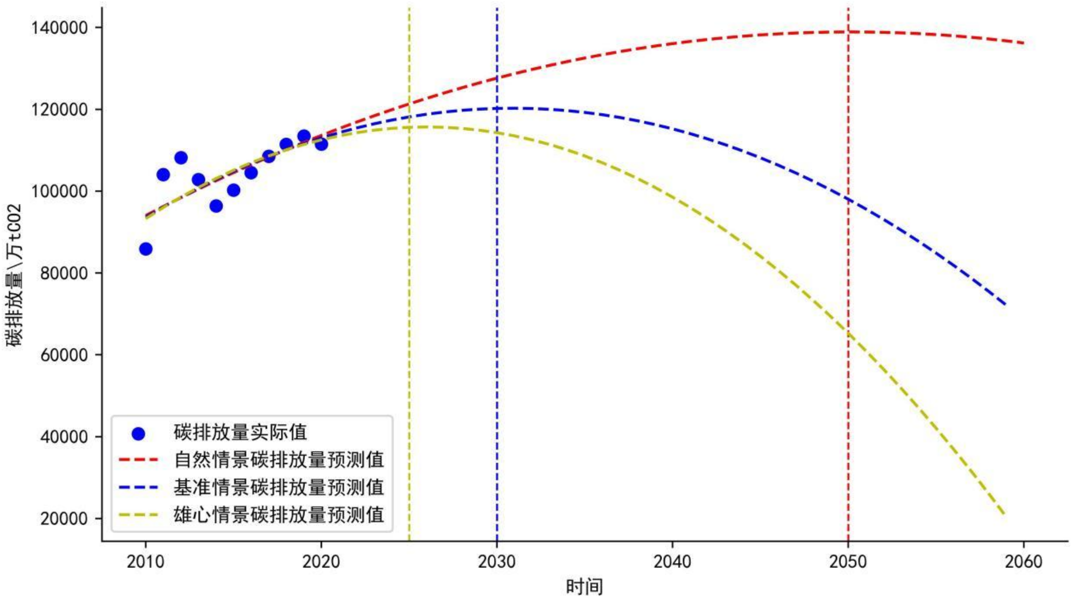

图 5-16 三种设计情景下碳排放量的趋势图

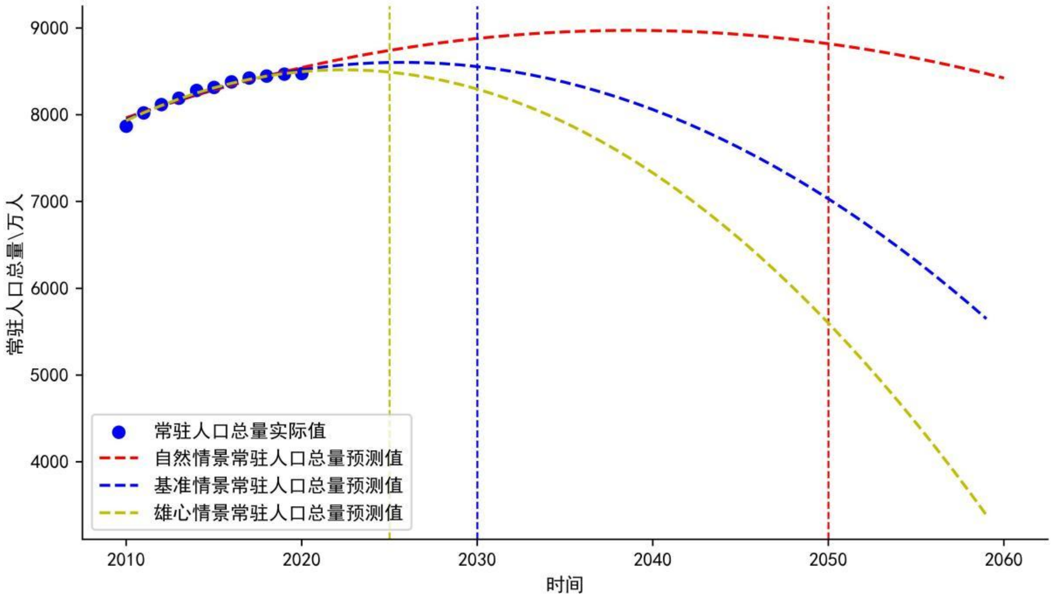

图 5-17 三种设计情景下人口的趋势

通过对比三种情景下的人口、经济和能源消费总量的趋势图，其中在经济成稳定增长的情况，人口数量减少，说人均 GDP 增加，这表明经济呈现出良好的正增长的情况。综合来讲，“碳达峰”和“碳中和”发展目标顺应我国可持续发展的内在要求，有利于构建绿色低碳可持续的循环经济发展，助推绿色生产方式和生活方式，实现社会高质量发展。并且在“碳中和”目标倒逼下，为各部门绿色低碳发展带来了压力与机遇，未来在低碳领域将提供众多就业机会和新的经济增长点，助力我国经济保持稳健增长。

6 模型评价

6.1 模型的优点

（1）问题二运用组合预测模型。组合预测模型能减少预测的系统误差，很好的反映实际情况，实用性强，能够显著改进预测效果。

（2）采用较为成熟的数学理论建立模型，可信度比较高。

（3）本模型采用 Python 软件进行求解，计算出来的值的精确度和稳定性都较高。

（4）本文建立的模型均经过多次修正，综合考虑到了一级指标、二级指标等方面，给出了较优预测值，对于区域双碳目标与路径规划研究有较大的参考价值。

6.2 模型的缺点及改进方法

（1）双碳目标路径规划没有考虑人口因素的影响。

（2）模型虽然综合考虑到了很多因素，但为了建立通用的数学模型，理想化了许多影响因素，具有一定的局限性，得到的最优模型可能与实际有一定的出入。

参考文献

[1] 唐杰等，基于 Kaya 模型的碳排放达峰实证研究[J]，深圳社会科学，第 5 卷第 3 期，2022 年 5 月，第 50-59 页。
[2] 李萍丰、谢守冬等，基于层次分析法与主成分分析法相结合的炮爆破效果评价，第十届全国砂石骨料行业科技大会论文集，2023 年 7 月，第 17-29 页。
[3] 孔春旺、童怀水等，线性组合模型及在人口预测中的应用，江西科学，2009,,27(04)，2009 年 5 月，第 543-547 页。
[4] 王勇胜、薛继量，基于多种模型组合的我国 2015 年人口数预测[J]，西北农林科技大学学报（社会科学版），2009 年 9 月，第 75-79 页。
[5] 冯新惠，李艳等，“双碳”目标下城市形态对碳排放的影响：以长江经济带为例，环境科学，2023 年 9 月 22 日，16 页。
[6] 王艳，“双碳”目标下中国碳排放规模情景预测，硕士毕业论文，2022 年 10 月，80 页。
[7] 李瑞月、钟山、张漓杉等，多元线性回归-分光光度法测定电镀排放废水中重金属[J/OL]. 中国环境监测，2023.05.23，1-10。
[8] 樊丽军，基于多元线性回归模型的建筑能耗预测与建筑节能分析[J]，湘潭大学自然科学学报，2016 年 1 月，123-126。

附录

附录 A：使用工具及版本

1、使用到的软件如下：

（1）pythonl

（2）excel

2、Python 第三方库版本如下：

（1）pandas、

（2）numpy、

（3）sklearn、

（4）matplotlib、

（5）sympy

附录 B：运行结果

问题 2 第（1）问的具体结果：

T2-能源消费量预测表			
时间	区域生产总值预测值	常驻人口总量预测值	能源消费量预测值
2010	41176.49	7964.958	25316.8
2011	46049.8	8033.787	26136.07
2012	50923.1	8100.181	26930.6
2013	55796.41	8164.14	27700.39
2014	60669.71	8225.662	28445.44
2015	65543.02	8284.75	29165.74
2016	70416.33	8341.401	29861.3
2017	75289.63	8395.617	30532.12
2018	80162.94	8447.398	31178.19
2019	85036.24	8496.743	31799.52
2020	89909.55	8543.652	32396.11
2021	94782.86	8588.126	32967.95
2022	99656.16	8630.164	33515.05
2023	104529.5	8669.767	34037.41
2024	109402.8	8706.934	34535.02
2025	114276.1	8741.666	35007.89
2026	119149.4	8773.962	35456.02
2027	124022.7	8803.822	35879.41
2028	128896	8831.247	36278.05
2029	133769.3	8856.236	36651.95
2030	138642.6	8878.79	37001.1
2031	143515.9	8898.908	37325.52
2032	148389.2	8916.59	37625.19
2033	153262.5	8931.837	37900.11
2034	158135.8	8944.649	38150.3

2035	163009.1	8955.025	38375.74
2036	167882.4	8962.965	38576.44
2037	172755.8	8968.469	38752.39
2038	177629.1	8971.539	38903.6
2039	182502.4	8972.172	39030.07
2040	187375.7	8970.37	39131.79
2041	192249	8966.132	39208.77
2042	197122.3	8959.459	39261.01
2043	201995.6	8950.351	39288.51
2044	206868.9	8938.806	39291.26
2045	211742.2	8924.826	39269.27
2046	216615.5	8908.411	39222.54
2047	221488.8	8889.56	39151.06
2048	226362.1	8868.273	39054.84
2049	231235.4	8844.551	38933.88
2050	236108.7	8818.393	38788.17
2051	240982	8789.8	38617.72
2052	245855.3	8758.771	38422.53
2053	250728.6	8725.307	38202.59
2054	255602	8689.407	37957.92
2055	260475.3	8651.071	37688.49
2056	265348.6	8610.3	37394.33
2057	270221.9	8567.093	37075.42
2058	275095.2	8521.451	36731.77
2059	279968.5	8473.373	36363.38
2060	284841.8	8422.859	35970.24

问题 2 第（2）问预测结果

T2-碳排放量预测表						
时间	农林消费部门的碳排放量预测值	一次能源中化石的能源消费量	一次能源中非化石的能源消费量	二次能源的能源消费量	化石能源发电的能源消费量	非化石能源发电的能源消费量
2010	963.9057	24625.77	184.5975	506.4303	10815.3	506.4303
2011	1002.718	25265.14	210.2874	660.6435	11152.21	660.6435
2012	1040.405	25875.58	235.0386	819.9906	11468.98	819.9906
2013	1076.968	26457.07	258.8512	984.4716	11765.6	984.4716
2014	1112.406	27009.63	281.7252	1154.086	12042.09	1154.086
2015	1146.719	27533.25	303.6605	1328.835	12298.42	1328.835
2016	1179.907	28027.93	324.6572	1508.718	12534.62	1508.718
2017	1211.971	28493.67	344.7152	1693.734	12750.67	1693.734

2018	1242.91	28930.47	363.8346	1883.885	12946.57	1883.885
2019	1272.724	29338.33	382.0153	2079.169	13122.33	2079.169
2020	1301.413	29717.26	399.2574	2279.587	13277.95	2279.587
2021	1328.977	30067.25	415.5608	2485.139	13413.42	2485.139
2022	1355.417	30388.3	430.9256	2695.825	13528.75	2695.825
2023	1380.732	30680.41	445.3517	2911.644	13623.94	2911.644
2024	1404.922	30943.58	458.8392	3132.598	13698.98	3132.598
2025	1427.987	31177.82	471.388	3358.685	13753.88	3358.685
2026	1449.928	31383.11	482.9982	3589.907	13788.63	3589.907
2027	1470.744	31559.47	493.6697	3826.262	13803.24	3826.262
2028	1490.435	31706.89	503.4026	4067.751	13797.7	4067.751
2029	1509.001	31825.37	512.1968	4314.374	13772.02	4314.374
2030	1526.442	31914.91	520.0524	4566.131	13726.2	4566.131
2031	1542.759	31975.51	526.9693	4823.021	13660.23	4823.021
2032	1557.951	32007.18	532.9476	5085.046	13574.12	5085.046
2033	1572.018	32009.91	537.9873	5352.204	13467.87	5352.204
2034	1584.96	31983.69	542.0882	5624.497	13341.47	5624.497
2035	1596.778	31928.54	545.2506	5901.923	13194.92	5901.923
2036	1607.471	31844.46	547.4743	6184.483	13028.24	6184.483
2037	1617.039	31731.43	548.7593	6472.177	12841.4	6472.177
2038	1625.482	31589.46	549.1057	6765.005	12634.43	6765.005
2039	1632.8	31418.56	548.5135	7062.966	12407.31	7062.966
2040	1638.994	31218.72	546.9826	7366.062	12160.04	7366.062
2041	1644.063	30989.94	544.513	7674.291	11892.64	7674.291
2042	1648.007	30732.22	541.1048	7987.654	11605.08	7987.654
2043	1650.826	30445.56	536.758	8306.152	11297.39	8306.152
2044	1652.521	30129.97	531.4725	8629.783	10969.55	8629.783
2045	1653.09	29785.43	525.2483	8958.547	10621.56	8958.547
2046	1652.535	29411.96	518.0855	9292.446	10253.43	9292.446
2047	1650.855	29009.55	509.9841	9631.479	9865.163	9631.479
2048	1648.051	28578.2	500.944	9975.645	9456.746	9975.645
2049	1644.121	28117.91	490.9653	10324.95	9028.186	10324.95
2050	1639.067	27628.69	480.0479	10679.38	8579.481	10679.38
2051	1632.888	27110.52	468.1918	11038.95	8110.632	11038.95
2052	1625.585	26563.42	455.3971	11403.65	7621.64	11403.65
2053	1617.156	25987.38	441.6638	11773.49	7112.503	11773.49
2054	1607.603	25382.4	426.9918	12148.46	6583.222	12148.46
2055	1596.925	24748.48	411.3812	12528.56	6033.797	12528.56
2056	1585.122	24085.62	394.8319	12913.8	5464.228	12913.8
2057	1572.194	23393.83	377.344	13304.17	4874.515	13304.17
2058	1558.142	22673.09	358.9174	13699.67	4264.658	13699.67

2059	1542.965	21923.42	339.5522	14100.31	3634.657	14100.31
2060	1526.663	21144.81	319.2483	14506.09	2984.512	14506.09

时间	碳排放量预测值	工业消费部门的碳排放量预测值	交通消费部门的碳排放量预测值	建筑消费部门的碳排放量预测值	居民生活消费的碳排放量预测值	能源供应部门的碳排放量预测值
2010	94761.02	49669.87	3087.026	2865.117	4291.404	33883.7
2011	96830.43	50263.59	3340.125	3130.289	4611.601	34482.11
2012	98844.13	50813.67	3592.309	3397.82	4935.038	35064.89
2013	100802.1	51320.12	3843.577	3667.712	5261.714	35632.05
2014	102704.4	51782.93	4093.93	3939.964	5591.629	36183.58
2015	104551	52202.1	4343.367	4214.576	5924.784	36719.49
2016	106341.9	52577.64	4591.889	4491.548	6261.178	37239.78
2017	108077.1	52909.54	4839.495	4770.88	6600.812	37744.44
2018	109756.6	53197.81	5086.186	5052.572	6943.685	38233.48
2019	111380.4	53442.44	5331.961	5336.624	7289.798	38706.89
2020	112948.5	53643.44	5576.821	5623.037	7639.151	39164.68
2021	114460.9	53800.8	5820.765	5911.809	7991.742	39606.85
2022	115917.6	53914.52	6063.794	6202.942	8347.574	40033.39
2023	117318.6	53984.61	6305.908	6496.435	8706.644	40444.31
2024	118663.9	54011.06	6547.106	6792.288	9068.955	40839.6
2025	119953.5	53993.88	6787.388	7090.501	9434.504	41219.27
2026	121187.4	53933.06	7026.755	7391.074	9803.294	41583.32
2027	122365.6	53828.6	7265.207	7694.007	10175.32	41931.74
2028	123488.1	53680.51	7502.743	7999.3	10550.59	42264.54
2029	124554.9	53488.78	7739.364	8306.954	10929.1	42581.71
2030	125566	53253.42	7975.069	8616.967	11310.85	42883.26
2031	126521.4	52974.42	8209.858	8929.341	11695.83	43169.19
2032	127421.1	52651.79	8443.733	9244.075	12084.06	43439.49
2033	128265.1	52285.51	8676.691	9561.169	12475.52	43694.16
2034	129053.4	51875.61	8908.735	9880.623	12870.23	43933.22
2035	129786	51422.07	9139.863	10202.44	13268.17	44156.65
2036	130462.9	50924.89	9370.075	10526.61	13669.36	44364.45
2037	131084	50384.07	9599.372	10853.15	14073.78	44556.63
2038	131649.5	49799.62	9827.753	11182.04	14481.44	44733.19
2039	132159.3	49171.54	10055.22	11513.29	14892.35	44894.13

2040	132613.4	48499.82	10281.77	11846.91	15306.49	45039.43
2041	133011.8	47784.46	10507.4	12182.88	15723.87	45169.12
2042	133354.5	47025.46	10732.12	12521.22	16144.49	45283.18
2043	133641.5	46222.84	10955.93	12861.91	16568.35	45381.62
2044	133872.8	45376.57	11178.82	13204.97	16995.45	45464.43
2045	134048.3	44486.67	11400.79	13550.38	17425.79	45531.62
2046	134168.2	43553.13	11621.85	13898.16	17859.37	45583.19
2047	134232.4	42575.96	11841.99	14248.3	18296.19	45619.13
2048	134240.9	41555.15	12061.22	14600.79	18736.24	45639.44
2049	134193.7	40490.71	12279.53	14955.65	19179.54	45644.14
2050	134090.8	39382.63	12496.92	15312.86	19626.08	45633.21
2051	133932.1	38230.91	12713.4	15672.44	20075.86	45606.65
2052	133717.8	37035.56	12928.97	16034.37	20528.87	45564.47
2053	133447.8	35796.57	13143.62	16398.67	20985.13	45506.67
2054	133122.1	34513.95	13357.35	16765.33	21444.62	45433.24
2055	132740.7	33187.69	13570.17	17134.34	21907.35	45344.19
2056	132303.6	31817.79	13782.07	17505.72	22373.33	45239.52
2057	131810.7	30404.26	13993.06	17879.46	22842.54	45119.22
2058	131262.2	28947.1	14203.13	18255.55	23314.99	44983.29
2059	130658	27446.29	14412.29	18634.01	23790.69	44831.75
2060	129998.1	25901.85	14620.53	19014.83	24269.62	44664.58

附录 C：程序

T1-主成分分析

```
import pandas as pd
import numpy as np
from sklearn.linear_model import LinearRegression
    data = pd.read_csv(r"E:\参加点比赛\数模研赛\data\碳排放及各指标关联关系数据.csv")
system = data.loc[:, ['时间']]
system['区域生产总值'] = data['农林消费部门的生产总值'] + data['能源供应部门的生产总值'] + data['工业消费部门的生产总值'] + data['交通消费部门的生产总值'] + data['建筑消费部门的生产总值']
system['常驻人口总量'] = data['常驻人口总量']
system['区域能源消费量'] = data['农林消费部门的能源消费量'] + data['工业消费部门的能源消费量'] + data['交通消费部门的能源消费量'] + data['建筑消费部门的能源消费量'] + data['居民生活消费的能源消费量'] + data['能源供应部门的能源消费量']
system['区域碳排放量'] = data['农林消费部门的碳排放量'] + data['工业消费部门的碳排放量'] + data['交通消费部门的碳排放量'] + data['建筑消费部门的碳排放量'] + data['居民生活消费的碳排放量'] + data['能源供应部门的碳排放量']
print(system.columns)
```

```
# 选择自变量与因变量
X = system.loc[:, ['区域生产总值', '常驻人口总量', '区域能源消费量']].values
y = system['区域碳排放量'].values

# 创建线性回归模型
model = LinearRegression()

# 多元线性回归拟合模型
model.fit(X, y)

# 输出参数
coef = model.coef_
intercept = model.intercept_
R2 = model.score(X, y)
print("回归系数为: ", coef)
print("截距为: ", intercept)
print("模型 R2 为: ", R2)

# 输出预测值
system['区域碳排放量回归值'] = 99
for i in range(system.shape[0]):
          system.loc[i, ['区域碳排放量回归值']] = intercept + np.dot(coef, X[i])
# 绘制回归曲线
import matplotlib.pyplot as plt
from pylab import *
import seaborn as sns
mpl.rcParams['font.sans-serif'] = ['SimHei'] # 指定默认字体（解决中文无法显示的问题）
mpl.rcParams['axes.unicode_minus'] = False # 解决保存图像时负号“-”显示方块的问题

fig = plt.figure(figsize=(9, 5))
ax = fig.subplots()
ax.scatter(system['时间'], system['区域碳排放量'], c='b', label='区域碳排放量实际值')
ax.plot(system['时间'], system['区域碳排放量回归值'], c='r', linestyle='--', label='
区域碳排放量回归值')
ax.set_xlabel('时间')
ax.set_ylabel('碳排放量/万 tCO2')
ax.legend()
ax.spines['right'].set_color(None)
ax.spines['top'].set_color(None)
ax.text(system['时间'][0], 111000, '回归系数 = '+str(np.round(coef, 2)))
ax.text(system['时间'][0], 109000, '截距 = '+str(np.round(intercept, 2)))
ax.text(system['时间'][0], 107000, 'R2 = '+str(np.round(R2, 2)))

    plt.savefig(r"E:\参加点比赛\数模研赛\results\T1\区域碳排放量回归图.jpg", dpi=600)
```

```
# 绘制回归曲线
import matplotlib.pyplot as plt
from pylab import *
import seaborn as sns
mpl.rcParams['font.sans-serif'] = ['SimHei'] # 指定默认字体（解决中文无法显示的问题）
mpl.rcParams['axes.unicode_minus'] = False # 解决保存图像时负号“-”显示方块的问题
fig = plt.figure(figsize=(18, 4))
(ax1, ax2, ax3) = fig.subplots(1, 3)

# 相关指标变化
x = system['时间']
y = system['区域生产总值'].values
y1 = []
y2 = []
for i in range(1, len(y)):
      qoq = (y[i] - y[i-1]) / y[i-1] # 计算环比，与前一年
      y1.append(qoq)
      yoy = (y[i] - y[0]) / y[0] # 计算同比，与同一年
      y2.append(yoy)
x=x[1: ] # 从 2011 年开始展示
ax1.plot(x, y1, c='b', linestyle='-', label='环比')
ax1.plot(x, y2, c='y', linestyle='--', label='同比')
ax1.spines['right'].set_color(None)
ax1.spines['top'].set_color(None)
ax1.legend(loc=1)
ax1.set_xlabel('时间')
ax1.set_ylabel('变化情况/%')
ax1.set_title('区域生产总值随时间变化')

# 相关指标变化
x = system['时间']
y = system['常驻人口总量'].values
y1 = []
y2 = []
for i in range(1, len(y)):
      qoq = (y[i] - y[i-1]) / y[i-1] # 计算环比，与前一年
      y1.append(qoq)
      yoy = (y[i] - y[0]) / y[0] # 计算同比，与同一年
      y2.append(yoy)
x=x[1: ] # 从 2011 年开始展示
ax2.plot(x, y1, c='b', linestyle='-', label='环比')
ax2.plot(x, y2, c='y', linestyle='--', label='同比')
ax2.spines['right'].set_color(None)
```

```
ax2.spines['top'].set_color(None)
ax2.legend(loc=1)
ax2.set_xlabel('时间')
ax2.set_ylabel('变化情况/%')
ax2.set_title('常驻人口总量随时间变化')

# 相关指标变化
x = system['时间']
y = system['区域能源消费量'].values
y1 = []
y2 = []
for i in range(1, len(y)):
      qoq = (y[i] - y[i-1]) / y[i-1] # 计算环比，与前一年
      y1.append(qoq)
      yoy = (y[i] - y[0]) / y[0] # 计算同比，与同一年
      y2.append(yoy)
x=x[1: ] # 从 2011 年开始展示
ax3.plot(x, y1, c='b', linestyle='-', label='环比')
ax3.plot(x, y2, c='y', linestyle='--', label='同比')
ax3.spines['right'].set_color(None)
ax3.spines['top'].set_color(None)
ax3.legend(loc=1)
ax3.set_xlabel('时间')
ax3.set_ylabel('变化情况/%')
ax3.set_title('区域能源消费量随时间变化')

    plt.savefig(r"E:\参加点比赛\数模研赛\results\T1\相关指标变化（环比与同比）.jpg",
dpi=600)
```

T1-主成分分析

```
import pandas as pd
from sklearn.preprocessing import StandardScaler
import seaborn as sns
data = pd.read_csv(r"E:\参加点比赛\数模研赛\data\主成分数据.csv")
    print("数据 columns: \n", data.columns.values)
system = data.loc[:, ['时间']]
# 人口指标
system['常驻人口总量'] = data['常驻人口总量']
# 经济指标
system['区域生产总值'] = data['农林消费部门的生产总值'] + data['能源供应部门的生产总值
'] + data['工业消费部门的生产总值'] + data['交通消费部门的生产总值'] + data['建筑消费
部门的生产总值']
system['第一产业的生产总值'] = data['农林消费部门的生产总值']
system['第二产业的生产总值'] = data['能源供应部门的生产总值'] + data['工业消费部门的生
```

```
产总值']
system['第三产业的生产总值'] = data['交通消费部门的生产总值'] + data['建筑消费部门的生产总值']
# 能耗指标
system['区域能耗'] = data['农林消费部门的能源消费量'] + data['工业消费部门的能源消费量'] + data['交通消费部门的能源消费量'] + data['建筑消费部门的能源消费量'] + data['居民生活消费的能源消费量'] + data['能源供应部门的能源消费量']
system['第一产业的能耗'] =   data['农林消费部门的能源消费量']
system['第二产业的能耗'] = data['能源供应部门的能源消费量']+ data['工业消费部门的能源消费量']
system['第三产业的能耗'] = data['交通消费部门的能源消费量'] + data['建筑消费部门的能源消费量']
# 碳排指标
system['区域碳排'] = data['农林消费部门的碳排放量'] + data['工业消费部门的碳排放量'] + data['交通消费部门的碳排放量'] + data['建筑消费部门的碳排放量'] + data['居民生活消费的碳排放量'] + data['能源供应部门的碳排放量']
system['第一产业碳排'] = data['农林消费部门的碳排放量']
system['第二产业碳排'] = data['能源供应部门的碳排放量'] + data['工业消费部门的碳排放量']
system['第三产业碳排'] = data['交通消费部门的碳排放量'] + data['建筑消费部门的碳排放量']
# 各部门碳排指标
system['农林消费部门的碳排放量'] = data['农林消费部门的碳排放量']
system['能源供应部门的碳排放量'] = data['能源供应部门的碳排放量']
system['工业消费部门的碳排放量'] = data['工业消费部门的碳排放量']
system['交通消费部门的碳排放量'] = data['交通消费部门的碳排放量']
system['建筑消费部门的碳排放量'] = data['建筑消费部门的碳排放量']
    system['居民生活消费的碳排放量'] = data['居民生活消费的碳排放量']
import matplotlib.pyplot as plt
from pylab import *
import seaborn as sns
mpl.rcParams['font.sans-serif'] = ['SimHei'] # 指定默认字体（解决中文无法显示的问题）
mpl.rcParams['axes.unicode_minus'] = False # 解决保存图像时负号“-”显示方块的问题

fig = plt.figure(figsize=(16, 5))
(ax1, ax2) = fig.subplots(1, 2)
b = (0.207, 0.372, 0.705)

# 三个时期碳排均值
x = ['2010', '十二五', '十三五']
y1 = system.loc[system['时间']==2010, ['区域碳排']]['区域碳排'].mean()
y2 = system.loc[(system['时间']>=2011) & (system['时间']<=2015), ['区域碳排']]['区域碳排'].mean()
y3 = system.loc[(system['时间']>=2016) & (system['时间']<=2020), ['区域碳排']]['区域碳
```

```
排'].mean()
y = [y1, y2, y3]
ax1.bar(x, y, color=b, width=0.5)
ax1.text(x[0], y1+1000, round(y1, 2), ha='center', va='bottom')
ax1.text(x[1], y2+1000, round(y2, 2), ha='center', va='bottom')
ax1.text(x[2], y3+1000, round(y3, 2), ha='center', va='bottom')
ax1.set_ylabel('碳排放量均值/万 tCO2')
ax1.set_title('三个时期碳排放量均值')
ax1.spines['right'].set_color(None)
ax1.spines['top'].set_color(None)

ax2.plot(system['时间'], system['区域碳排'], c=b, marker='o')
ax2.set_xlabel('时间')
ax2.set_ylabel('区域碳排放量均值/万 tCO2')
ax2.set_title('区域碳排放量随时间变化')
ax2.spines['right'].set_color(None)
ax2.spines['top'].set_color(None)
ax2.text(system['时间'][2], system['区域碳排'][2]+1000, round(system['区域碳排'][2],
2), ha='center', va='bottom')
ax2.text(system['时间'][4], system['区域碳排'][4]-2000, round(system['区域碳排'][4],
2), ha='center', va='bottom')
ax2.text(system['时间'][9], system['区域碳排'][9]+1000, round(system['区域碳排'][9],
2), ha='center', va='bottom')
ax2.axvline(x=system['时间'][1], color='red', linestyle='--', linewidth=1)
ax2.axvline(x=system['时间'][6], color='red', linestyle='--', linewidth=1)

    plt.savefig(r"E:\参加点比赛\数模研赛\results\T1\碳排放量状况图.jpg", dpi=600)
import matplotlib.pyplot as plt
from pylab import *
import seaborn as sns
mpl.rcParams['font.sans-serif'] = ['SimHei'] # 指定默认字体（解决中文无法显示的问题）
mpl.rcParams['axes.unicode_minus'] = False # 解决保存图像时负号“-”显示方块的问题

fig = plt.figure(figsize=(16, 5))
(ax1, ax2) = fig.subplots(1, 2)
b = (0.207, 0.372, 0.705)

# 两个时期碳排总值
x = ['十二五', '十三五']
y1 = system.loc[(system['时间']>=2011) & (system['时间']<=2015), ['区域碳排']]['区域碳
排'].sum()
y2 = system.loc[(system['时间']>=2016) & (system['时间']<=2020), ['区域碳排']]['区域碳
排'].sum()
y = [y1, y2]
```

```
ax1.bar(x, y, color=b, width=0.3)
ax1.text(x[0], y1+1000, round(y1, 2), ha='center', va='bottom')
ax1.text(x[1], y2+1000, round(y2, 2), ha='center', va='bottom')
ax1.set_ylabel('碳排放量总值/万 tCO2')
ax1.set_title('两个时期碳排放量总值')
ax1.spines['right'].set_color(None)
ax1.spines['top'].set_color(None)
ax1.plot(x, y, c='r', linestyle='--', linewidth=1)

# 碳排放量同比
x = system['时间'] # 从 2010 开始
y = system['区域碳排'].values # 2010-2020 的数据
y1 = []
y2 = []
for i in range(1, len(y)):
      qoq = (y[i] - y[i-1]) / y[i-1] # 计算环比，与前一年
      y1.append(qoq)
      yoy = (y[i] - y[0]) / y[0] # 计算同比，与同一年
      y2.append(yoy)
x = x[1:] # 从 2011 开始展示
ax2.plot(x, y1, c='b', linestyle='-', label='环比')
ax2.plot(x, y2, c='y', linestyle='--', label='同比')
ax2.spines['right'].set_color(None)
ax2.spines['top'].set_color(None)
ax2.axvline(x=x.values[0], color='red', linestyle='--', linewidth=1)
ax2.axvline(x=x.values[5], color='red', linestyle='--', linewidth=1)
ax2.legend(loc=1)
ax2.set_xlabel('时间')
ax2.set_ylabel('变化情况/%')
ax2.set_title('区域碳排放量随时间变化')

    plt.savefig(r"E:\参加点比赛\数模研赛\results\T1\区域碳排放量总值及历年变化情
况.jpg", dpi=600)
# 绘制热力图
system_pca = system.loc[:, ['常驻人口总量', '区域生产总值', '第一产业的生产总值', '第
二产业的生产总值', '第三产业的生产总值', '区域能耗',
           '第一产业的能耗', '第二产业的能耗', '第三产业的能耗', '区域碳排']]

# 标准化
system_pca_value = system_pca.values # 取值
scaler = StandardScaler()
system_pca_value_sd = scaler.fit_transform(system_pca_value).astype('float32') # 标准
化后的值
system_pca.loc[:, :]= system_pca_value_sd # 设置标准化后的值，可以退回去原来的值：
```

```
system_pca.loc[:, :]= system_pca_value

# 计算相关系数矩阵
corr = np.corrcoef(system_pca_value_sd.T)

# 绘制热力图
plt.figure(figsize=(12, 12))
sns.heatmap(pd.DataFrame(corr, columns=system_pca.columns, index=system_pca.columns),
annot=True, cmap='coolwarm', fmt='.2f', linewidths=0.5)
plt.xticks(rotation=45)
plt.yticks(rotation=45)
    plt.savefig(r"E:\参加点比赛\数模研赛\results\T1\碳排放量与各指标热力图.jpg",
dpi=500)
system_pca = system.loc[:, ['常驻人口总量', '区域生产总值', '第一产业的生产总值', '第
二产业的生产总值', '第三产业的生产总值', '区域能耗',
           '第一产业的能耗', '第二产业的能耗', '第三产业的能耗']]

# 标准化
system_pca_value = system_pca.values # 取值
scaler = StandardScaler()
system_pca_value_sd = scaler.fit_transform(system_pca_value).astype('float32') # 标准
化后的值
system_pca.loc[:, :]= system_pca_value_sd # 设置标准化后的值，可以退回去原来的值：
system_pca.loc[:, :]= system_pca_value

# 计算相关系数矩阵
corr = np.corrcoef(system_pca_value_sd.T)
# pd.DataFrame(corr, columns=system_pca.columns, index=system_pca.columns)

# 计算特征值和特征向量
eigvalue = np.linalg.eig(corr)[0].astype('float32')
eigvector = np.linalg.eig(corr)[1].astype('float32')
print('特征值为：\n', eigvalue)

# 计算主成分对应的特征向量、特征值、贡献率、累计贡献率
summary = pd.DataFrame(columns=['主成分'+str(i) for i in range(1,
system_pca.shape[1]+1)],
                                   index=system_pca.columns.values.tolist() + ['特征值
', '贡献率', '累计贡献率'])

# 填入特征向量
sort_index=np.argsort(-(eigvalue))
for i in range(len(eigvalue)): # 填入特征向量
      summary.iloc[:9,i]=eigvector[sort_index][i]
```

```
# 填入特征值
summary.loc['特征值']=eigvalue[sort_index]

# 填入共享率
for i in range(len(summary.columns)): # 各个主成分计算贡献率
      summary.iloc[10, i] = summary.iloc[9,i]/summary.iloc[9,:].sum()
summary.iloc[8:,:]

# 各个主成分计算累计贡献率
comulative = 0
for i in range(len(summary.columns)):
      comulative += summary.iloc[10,i]
      summary.iloc[11,i] = comulative
    summary.to_csv(r"E:\参加点比赛\数模研赛\results\T1\主成分结果表.csv", index=True,
encoding='gbk')
# 主成分数据
system_pca.loc[:, :] = system_pca_value_sd
new = system_pca.copy()
new['碳排放量'] = system['区域碳排']
new['主成分 1'] = 99
for j in range(11):
      t = summary.iloc[:9, 0] # 计算第 i 列特征向量
      x = system_pca.iloc[j, :] # 计算第 j 行
      new.loc[j, ['主成分 1']] = np.dot(t, x)

import matplotlib.pyplot as plt
from pylab import *
import seaborn as sns
mpl.rcParams['font.sans-serif'] = ['SimHei'] # 指定默认字体（解决中文无法显示的问题）
mpl.rcParams['axes.unicode_minus'] = False # 解决保存图像时负号“-”显示方块的问题

fig = plt.figure(figsize=(12, 6))
ax = fig.subplots()
ax.plot(system['时间'], scaler.fit_transform(new.loc[:, ['碳排放量']])[:,0], c='b',
label='碳排放量')
ax.plot(system['时间'], new['主成分 1'], c='y', label='主成分',)
ax.legend(loc=1)
ax.spines['right'].set_color(None)
ax.spines['top'].set_color(None)
ax.axvline(x=system['时间'][1], color='red', linestyle='--', linewidth=1)
ax.axvline(x=system['时间'][6], color='red', linestyle='--', linewidth=1)
ax.set_xlabel('时间')
ax.set_ylabel('标准化值')
```

```python
    plt.savefig(r"E:\参加点比赛\数模研赛\results\T1\碳排放量与主成分对应图.jpg",
dpi=500)
from pyecharts import options as opts
from pyecharts.charts import Radar

data1 = [{"value": summary.iloc[:9, 0].values.tolist(), "name": "主成分 1"}]
c_schema = [
      {"name": summary.index[0], "max": 1, "min": -1}, # 范围
      {"name": summary.index[1], "max": 1, "min": -1},
      {"name": summary.index[2], "max": 1, "min": -1},
      {"name": summary.index[3], "max": 1, "min": -1},
      {"name": summary.index[4], "max": 1, "min": -1},
      {"name": summary.index[5], "max": 1, "min": -1},
      {"name": summary.index[6], "max": 1, "min": -1},
      {"name": summary.index[7], "max": 1, "min": -1},
      {"name": summary.index[8], "max": 1, "min": -1},
]
c = (
      Radar()
      .set_colors(["#4587E7"])
      .add_schema(
            schema=c_schema,
            shape="circle",
            center=["50%", "50%"],
            radius="80%",
            angleaxis_opts=opts.AngleAxisOpts(
                  min_=0,
                  max_=360,
                  is_clockwise=False,
                  interval=2,
                  axistick_opts=opts.AxisTickOpts(is_show=False),
                  axislabel_opts=opts.LabelOpts(is_show=False),
                  axisline_opts=opts.AxisLineOpts(is_show=False),
                  splitline_opts=opts.SplitLineOpts(is_show=False),
            ),
            radiusaxis_opts=opts.RadiusAxisOpts(
                  min_=-1, # 范围
                  max_=1,
                  interval=1,
                  splitarea_opts=opts.SplitAreaOpts(
                        is_show=True, areastyle_opts=opts.AreaStyleOpts(opacity=1)
                  ),
            ),
```

```
            polar_opts=opts.PolarOpts(),
            splitarea_opt=opts.SplitAreaOpts(is_show=False),
            splitline_opt=opts.SplitLineOpts(is_show=False),
      )
      .add(
            series_name="主成分 1",
            data=data1,
            areastyle_opts=opts.AreaStyleOpts(opacity=0.1),
            linestyle_opts=opts.LineStyleOpts(width=2, color="#5CACEE"),
      )
      .set_series_opts(label_opts=opts.LabelOpts(is_show=False))

)
c.render_notebook()
```

T2-碳排放量预测

```
import pandas as pd
from sklearn.linear_model import LinearRegression
from sklearn.preprocessing import PolynomialFeatures
from scipy.optimize import curve_fit
from sklearn.metrics import r2_score
import sympy as sp
import numpy as np
np.set_printoptions(suppress=True)

data = pd.read_csv(r"E:\参加点比赛\数模研赛\data\碳排预测数据.csv")
print(data.columns)
    system = data.copy()
# 对经济做拟合
x = system['时间'].values.tolist()
GDP = system['区域生产总值'].values.tolist()
def fit_gdp(x, a, b): # 拟合函数
      return a * x + b
params, convariance = curve_fit(fit_gdp, x, GDP) # 将 x、GDP 放入 fit_gdp 中拟合
a, b = params # 获取拟合参数
x = sp.symbols('x')
predict_gdp = a * x + b # 获取拟合方程式
print("经济关于时间的线性回归预测函数 = {}".format(predict_gdp))

# 对人口做拟合
# 做多项式拟合
x = system['时间'].values.tolist()
P = system['常驻人口总量'].values.tolist()
def fit_p_poly(x, a, b, c):
```

```
    return a * x**2 + b * x + c
params, convariance = curve_fit(fit_p_poly, x, P)
a, b, c = params
x = sp.symbols('x')
predict_p_poly = a * x**2 + b * x + c
print("人口关于时间的多项式回归预测函数 = {}".format(predict_p_poly))
# 做线性拟合
x = system['时间'].values.tolist()
P = system['常驻人口总量'].values.tolist()
def fit_p_linear(x, a, b):
    return a * x + b
params, convariance = curve_fit(fit_p_linear, x, P)
a, b = params
x = sp.symbols('x')
predict_p_linear = a * x + b
print("人口关于时间的线性回归预测函数 = {}".format(predict_p_linear))
# 组合预测
predict_p = 0.8 * predict_p_linear + 0.2 * predict_p_poly
print("人口关于时间的组合预测函数 = {}".format(predict_p))

print("已完成建立人口、经济与时间的预测函数"+"-"*100)
print("gdp(x) = {}".format(predict_gdp))
   print("p(x) = {}".format(predict_p))
# 选择自变量与应变量
GDPP = system.loc[:, ['区域生产总值', '常驻人口总量']].values # 自变量
EV1 = system['一次能源中化石的能源消费量'].values
EV2 = system['一次能源中非化石的能源消费量'].values
EV3 = system['二次能源的能源消费量'].values
EV4 = system['化石能源发电的能源消费量'].values
EV5 = system['非化石能源发电的能源消费量'].values

# 求解 EV1 关于 GDP 的线性回归方程
lr_EV1 = LinearRegression()
lr_EV1.fit(GDPP, EV1)
coef_EV1 = lr_EV1.coef_
intercept_EV1 = lr_EV1.intercept_
R2_EV1 = lr_EV1.score(GDPP, EV1)
print("EV1 关于 GDPP 的方程求解完毕" + "-"*100)
print("相关系数 = ", coef_EV1)
print("截距 = ", intercept_EV1)
print("R2 = ", R2_EV1)

# 求解 EV2 关于 GDP 的线性回归方程
lr_EV2 = LinearRegression()
```

```
lr_EV2.fit(GDPP, EV2)
coef_EV2 = lr_EV2.coef_
intercept_EV2 = lr_EV2.intercept_
R2_EV2 = lr_EV2.score(GDPP, EV2)
print("EV2 关于 GDPP 的方程求解完毕" + "-"*100)
print("相关系数 = ", coef_EV2)
print("截距 = ", intercept_EV2)
print("R2 = ", R2_EV2)

# 求解 EV3 关于 GDP 的线性回归方程
lr_EV3 = LinearRegression()
lr_EV3.fit(GDPP, EV3)
coef_EV3 = lr_EV3.coef_
intercept_EV3 = lr_EV3.intercept_
R2_EV3 = lr_EV3.score(GDPP, EV3)
print("EV3 关于 GDPP 的方程求解完毕" + "-"*100)
print("相关系数 = ", coef_EV3)
print("截距 = ", intercept_EV3)
print("R2 = ", R2_EV3)

# 求解 EV4 关于 GDP 的线性回归方程
lr_EV4 = LinearRegression()
lr_EV4.fit(GDPP, EV4)
coef_EV4 = lr_EV4.coef_
intercept_EV4 = lr_EV4.intercept_
R2_EV4 = lr_EV4.score(GDPP, EV4)
print("EV4 关于 GDPP 的方程求解完毕" + "-"*100)
print("相关系数 = ", coef_EV4)
print("截距 = ", intercept_EV4)
print("R2 = ", R2_EV4)

# 求解 EV5 关于 GDP 的线性回归方程
lr_EV5 = LinearRegression()
lr_EV5.fit(GDPP, EV5)
coef_EV5 = lr_EV5.coef_
intercept_EV5 = lr_EV5.intercept_
R2_EV5 = lr_EV5.score(GDPP, EV5)
print("EV5 关于 GDPP 的方程求解完毕" + "-"*100)
print("相关系数 = ", coef_EV5)
print("截距 = ", intercept_EV5)
    print("R2 = ", R2_EV5)
# 选择自变量与应变量
Eall = system.loc[:, ['一次能源中化石的能源消费量', '一次能源中非化石的能源消费量', '
二次能源的能源消费量', '化石能源发电的能源消费量',
```

```
            '非化石能源发电的能源消费量']]
CO2D1 = system['农林消费部门的碳排放量'].values
CO2D2 = system['工业消费部门的碳排放量'].values
CO2D3 = system['交通消费部门的碳排放量'].values
CO2D4 = system['建筑消费部门的碳排放量'].values
CO2D5 = system['居民生活消费部门的碳排放量'].values
CO2D6 = system['能源供应部门的碳排放量'].values
CO2 = system['碳排放量'].values

# 求解 CO2D1 关于五种能源的方程
lr_CO2D1 = LinearRegression()
lr_CO2D1.fit(Eall, CO2D1)
coef_CO2D1 = lr_CO2D1.coef_
intercept_CO2D1 = lr_CO2D1.intercept_
R2_CO2D1 = lr_CO2D1.score(Eall, CO2D1)
print("CO2D1 关于 Eall 的方程求解完毕" + "-"*100)
print("相关系数 = ")
for i in coef_CO2D1:
      print(i)
print("截距 = ", intercept_CO2D1)
print("R2 = ", R2_CO2D1)

# 求解 CO2D2 关于五种能源的方程
lr_CO2D2 = LinearRegression()
lr_CO2D2.fit(Eall, CO2D2)
coef_CO2D2 = lr_CO2D2.coef_
intercept_CO2D2 = lr_CO2D2.intercept_
R2_CO2D2 = lr_CO2D2.score(Eall, CO2D2)
print("CO2D2 关于 Eall 的方程求解完毕" + "-"*100)
print("相关系数 = ")
for i in coef_CO2D2:
      print(i)
print("截距 = ", intercept_CO2D2)
print("R2 = ", R2_CO2D2)

# 求解 CO2D3 关于五种能源的方程
lr_CO2D3 = LinearRegression()
lr_CO2D3.fit(Eall, CO2D3)
coef_CO2D3 = lr_CO2D3.coef_
intercept_CO2D3 = lr_CO2D3.intercept_
R2_CO2D3 = lr_CO2D3.score(Eall, CO2D3)
print("CO2D3 关于 Eall 的方程求解完毕" + "-"*100)
print("相关系数 = ")
for i in coef_CO2D3:
```

```
      print(i)
print("截距 = ", intercept_CO2D3)
print("R2 = ", R2_CO2D3)

# 求解 CO2D4 关于五种能源的方程
lr_CO2D4 = LinearRegression()
lr_CO2D4.fit(Eall, CO2D4)
coef_CO2D4 = lr_CO2D4.coef_
intercept_CO2D4 = lr_CO2D4.intercept_
R2_CO2D4 = lr_CO2D4.score(Eall, CO2D4)
print("CO2D4 关于 Eall 的方程求解完毕" + "-"*100)
print("相关系数 = ")
for i in coef_CO2D4:
      print(i)
print("截距 = ", intercept_CO2D4)
print("R2 = ", R2_CO2D4)

# 求解 CO2D5 关于五种能源的方程
lr_CO2D5 = LinearRegression()
lr_CO2D5.fit(Eall, CO2D5)
coef_CO2D5 = lr_CO2D5.coef_
intercept_CO2D5 = lr_CO2D5.intercept_
R2_CO2D5 = lr_CO2D5.score(Eall, CO2D5)
print("CO2D5 关于 Eall 的方程求解完毕" + "-"*100)
print("相关系数 = ")
for i in coef_CO2D5:
      print(i)
print("截距 = ", intercept_CO2D5)
print("R2 = ", R2_CO2D5)

# 求解 CO2D6 关于五种能源的方程
lr_CO2D6 = LinearRegression()
lr_CO2D6.fit(Eall, CO2D6)
coef_CO2D6 = lr_CO2D6.coef_
intercept_CO2D6 = lr_CO2D6.intercept_
R2_CO2D6 = lr_CO2D6.score(Eall, CO2D6)
print("CO2D6 关于 Eall 的方程求解完毕" + "-"*100)
print("相关系数 = ")
for i in coef_CO2D6:
      print(i)
print("截距 = ", intercept_CO2D6)
print("R2 = ", R2_CO2D6)
```

```
predict_EV1 = coef_EV1[0] * predict_gdp + coef_EV1[1] * predict_p + intercept_EV1
predict_EV2 = coef_EV2[0] * predict_gdp + coef_EV2[1] * predict_p + intercept_EV2
predict_EV3 = coef_EV3[0] * predict_gdp + coef_EV3[1] * predict_p + intercept_EV3
predict_EV4 = coef_EV4[0] * predict_gdp + coef_EV4[1] * predict_p + intercept_EV4
predict_EV5 = coef_EV5[0] * predict_gdp + coef_EV5[1] * predict_p + intercept_EV5

predict_CO2D1 = coef_CO2D1[0] * predict_EV1 + \
                coef_CO2D1[1] * predict_EV2 + \
                coef_CO2D1[2] * predict_EV3 + \
                coef_CO2D1[3] * predict_EV4 + \
                coef_CO2D1[4] * predict_EV5 + intercept_CO2D1

predict_CO2D2 = coef_CO2D2[0] * predict_EV1 + \
                coef_CO2D2[1] * predict_EV2 + \
                coef_CO2D2[2] * predict_EV3 + \
                coef_CO2D2[3] * predict_EV4 + \
                coef_CO2D2[4] * predict_EV5 + intercept_CO2D2

predict_CO2D3 = coef_CO2D3[0] * predict_EV1 + \
                coef_CO2D3[1] * predict_EV2 + \
                coef_CO2D3[2] * predict_EV3 + \
                coef_CO2D3[3] * predict_EV4 + \
                coef_CO2D3[4] * predict_EV5 + intercept_CO2D3

predict_CO2D4 = coef_CO2D4[0] * predict_EV1 + \
                coef_CO2D4[1] * predict_EV2 + \
                coef_CO2D4[2] * predict_EV3 + \
                coef_CO2D4[3] * predict_EV4 + \
                coef_CO2D4[4] * predict_EV5 + intercept_CO2D4

predict_CO2D5 = coef_CO2D5[0] * predict_EV1 + \
                coef_CO2D5[1] * predict_EV2 + \
                coef_CO2D5[2] * predict_EV3 + \
                coef_CO2D5[3] * predict_EV4 + \
                coef_CO2D5[4] * predict_EV5 + intercept_CO2D5

predict_CO2D6 = coef_CO2D6[0] * predict_EV1 + \
                coef_CO2D6[1] * predict_EV2 + \
                coef_CO2D6[2] * predict_EV3 + \
                coef_CO2D6[3] * predict_EV4 + \
                coef_CO2D6[4] * predict_EV5 + intercept_CO2D6

predict_CO2 = predict_CO2D1 + predict_CO2D2 + predict_CO2D3 + predict_CO2D4 +
predict_CO2D5 + predict_CO2D6
```

```python
print("五种能源消费量关于时间" + "-"*100)
print("EV1(x) = ", predict_EV1)
print("EV2(x) = ", predict_EV2)
print("EV3(x) = ", predict_EV3)
print("EV4(x) = ", predict_EV4)
print("EV5(x) = ", predict_EV5)
print("六个部门能耗关于时间" + "-"*100)
print("CO2D1(x) = ", predict_CO2D1)
print("CO2D2(x) = ", predict_CO2D2)
print("CO2D3(x) = ", predict_CO2D3)
print("CO2D4(x) = ", predict_CO2D4)
print("CO2D5(x) = ", predict_CO2D5)
print("CO2D6(x) = ", predict_CO2D6)
print("总能耗关于时间" + "-"*100)
    print("CO2(x) = ", predict_CO2)
# 新建预测结果
system_pre = pd.DataFrame({'时间': [i for i in range(2010, 2061)]})
system_pre['农林消费部门的碳排放量预测值'] = 99
system_pre['农林消费部门的碳排放量预测值'] = 99
system_pre['农林消费部门的碳排放量预测值'] = 99
system_pre['农林消费部门的碳排放量预测值'] = 99
system_pre['农林消费部门的碳排放量预测值'] = 99
system_pre['农林消费部门的碳排放量预测值'] = 99
system_pre['碳排放量预测值'] = 99

# 预测
for i in range(system_pre.shape[0]):
      t = system_pre.loc[i, ['时间']][0] # 获取第 i 个时间
      system_pre.loc[i, ['农林消费部门的碳排放量预测值']] = predict_CO2D1.subs(x, t)
      system_pre.loc[i, ['工业消费部门的碳排放量预测值']] = predict_CO2D2.subs(x, t)
      system_pre.loc[i, ['交通消费部门的碳排放量预测值']] = predict_CO2D3.subs(x, t)
      system_pre.loc[i, ['建筑消费部门的碳排放量预测值']] = predict_CO2D4.subs(x, t)
      system_pre.loc[i, ['居民生活消费的碳排放量预测值']] = predict_CO2D5.subs(x, t)
      system_pre.loc[i, ['能源供应部门的碳排放量预测值']] = predict_CO2D6.subs(x, t)
          system_pre.loc[i, ['碳排放量预测值']] = predict_CO2.subs(x, t)
# 绘制原始曲线、两条拟合结果
import matplotlib.pyplot as plt
from pylab import *
import seaborn as sns
mpl.rcParams['font.sans-serif'] = ['SimHei'] # 指定默认字体（解决中文无法显示的问题）
mpl.rcParams['axes.unicode_minus'] = False # 解决保存图像时负号“-”显示方块的问题
fig = plt.figure(figsize=(9, 5))
(ax) = fig.subplots(1, 1)
```

```
ax.scatter(system['时间'], system['碳排放量'], c='b', label='碳排放量实际值')
ax.plot(system_pre['时间'], system_pre['碳排放量预测值'], c='r', linestyle='--',
label='碳排放量预测值')
ax.text(system_pre['时间'][0], 125000, "碳排放量预测方程 = {}".format(predict_C02))

ax.set_xlabel('时间')
ax.set_ylabel('碳排放量\万 tC02')
ax.spines['right'].set_color(None)
ax.spines['top'].set_color(None)
ax.legend()

plt.savefig(r"E:\参加点比赛\数模研赛\results\T2\碳排放量预测图.jpg", dpi=500)
    system_pre.to_csv(r"E:\参加点比赛\数模研赛\results\T2\碳排放量预测表.csv",
encoding='utf-8-sig', index=False)
```

T3-情景参数调整

```
import pandas as pd
from sklearn.linear_model import LinearRegression
from sklearn.preprocessing import PolynomialFeatures
from scipy.optimize import curve_fit
from sklearn.metrics import r2_score
import sympy as sp
from tqdm import tqdm
import numpy as np
np.set_printoptions(suppress=True)

data = pd.read_csv(r"E:\参加点比赛\数模研赛\data\碳排预测数据.csv")
print(data.columns)
    system = data.copy()
data = pd.read_csv(r"E:\参加点比赛\数模研赛\data\碳排预测数据.csv")
system = data.copy()

w_list = np.linspace(4863, 4868, 20)
for i in tqdm(range(len(w_list))):
      w = w_list[i] # 权重
      sign1 = False # 判断碳达峰时间约束
      sign2 = False # 判断预测结果正负约束
      sign3 = False # 判断碳中和约束
      # 假设条件下拟合得到人口、经济与时间的关系式
      ##############################################################################
################################################################
      # 对人口做拟合
```

```
# 做多项式拟合
x = system['时间'].values.tolist()
P = system['常驻人口总量'].values.tolist()
x = x + [2030]
P = P + [w]
def fit_p_poly(x, a, b, c):
    return a * x**2 + b * x + c
params, convariance = curve_fit(fit_p_poly, x, P)
a, b, c = params
x = sp.symbols('x')
predict_p_poly = a * x**2 + b * x + c
# 做线性拟合
x = system['时间'].values.tolist()
P = system['常驻人口总量'].values.tolist()
def fit_p_linear(x, a, b):
    return a * x + b
params, convariance = curve_fit(fit_p_linear, x, P)
a, b = params
x = sp.symbols('x')
predict_p_linear = a * x + b
# 组合预测
predict_p = 0.8 * predict_p_linear + 0.2 * predict_p_poly
# 对经济做拟合：满足给定条件
x = system['时间'].values.tolist()
GDP = system['区域生产总值'].values.tolist()
x = x + [2035, 2060]
GDP = GDP + [2*GDP[-1], 4*GDP[-1]]
def fit_gdp(x, a, b): # 拟合函数
    return a * x + b
params, convariance = curve_fit(fit_gdp, x, GDP) # 将 x、GDP 放入 fit_gdp 中拟合
a, b = params # 获取拟合参数
x = sp.symbols('x')
predict_gdp = a * x + b # 获取拟合方程式


# 多元线性回归 得到五种能源与人口、经济关系
###############################################################################
###############################################################################
# 选择自变量与应变量
GDPP = system.loc[:, ['区域生产总值', '常驻人口总量']].values # 自变量
EV1 = system['一次能源中化石的能源消费量'].values
EV2 = system['一次能源中非化石的能源消费量'].values
EV3 = system['二次能源的能源消费量'].values
EV4 = system['化石能源发电的能源消费量'].values
```

```
    EV5 = system['非化石能源发电的能源消费量'].values

    # 求解 EV1 关于 GDP 的线性回归方程
    lr_EV1 = LinearRegression()
    lr_EV1.fit(GDPP, EV1)
    coef_EV1 = lr_EV1.coef_
    intercept_EV1 = lr_EV1.intercept_
    R2_EV1 = lr_EV1.score(GDPP, EV1)

    # 求解 EV2 关于 GDP 的线性回归方程
    lr_EV2 = LinearRegression()
    lr_EV2.fit(GDPP, EV2)
    coef_EV2 = lr_EV2.coef_
    intercept_EV2 = lr_EV2.intercept_
    R2_EV2 = lr_EV2.score(GDPP, EV2)

    # 求解 EV3 关于 GDP 的线性回归方程
    lr_EV3 = LinearRegression()
    lr_EV3.fit(GDPP, EV3)
    coef_EV3 = lr_EV3.coef_
    intercept_EV3 = lr_EV3.intercept_
    R2_EV3 = lr_EV3.score(GDPP, EV3)

    # 求解 EV4 关于 GDP 的线性回归方程
    lr_EV4 = LinearRegression()
    lr_EV4.fit(GDPP, EV4)
    coef_EV4 = lr_EV4.coef_
    intercept_EV4 = lr_EV4.intercept_
    R2_EV4 = lr_EV4.score(GDPP, EV4)

    # 求解 EV5 关于 GDP 的线性回归方程
    lr_EV5 = LinearRegression()
    lr_EV5.fit(GDPP, EV5)
    coef_EV5 = lr_EV5.coef_
    intercept_EV5 = lr_EV5.intercept_
    R2_EV5 = lr_EV5.score(GDPP, EV5)

    # 多元线性回归 得到六个部门碳排放量与五种能源消费量关系
    ################################################################################
################################################################
    # 选择自变量与应变量
    Eall = system.loc[:, ['一次能源中化石的能源消费量', '一次能源中非化石的能源消费量', '二次能源的能源消费量', '化石能源发电的能源消费量',
          '非化石能源发电的能源消费量']]
```

```
CO2D1 = system['农林消费部门的碳排放量'].values
CO2D2 = system['工业消费部门的碳排放量'].values
CO2D3 = system['交通消费部门的碳排放量'].values
CO2D4 = system['建筑消费部门的碳排放量'].values
CO2D5 = system['居民生活消费部门的碳排放量'].values
CO2D6 = system['能源供应部门的碳排放量'].values
CO2 = system['碳排放量'].values

# 求解 CO2D1 关于五种能源的方程
lr_CO2D1 = LinearRegression()
lr_CO2D1.fit(Eall, CO2D1)
coef_CO2D1 = lr_CO2D1.coef_
intercept_CO2D1 = lr_CO2D1.intercept_
R2_CO2D1 = lr_CO2D1.score(Eall, CO2D1)

# 求解 CO2D2 关于五种能源的方程
lr_CO2D2 = LinearRegression()
lr_CO2D2.fit(Eall, CO2D2)
coef_CO2D2 = lr_CO2D2.coef_
intercept_CO2D2 = lr_CO2D2.intercept_
R2_CO2D2 = lr_CO2D2.score(Eall, CO2D2)

# 求解 CO2D3 关于五种能源的方程
lr_CO2D3 = LinearRegression()
lr_CO2D3.fit(Eall, CO2D3)
coef_CO2D3 = lr_CO2D3.coef_
intercept_CO2D3 = lr_CO2D3.intercept_
R2_CO2D3 = lr_CO2D3.score(Eall, CO2D3)

# 求解 CO2D4 关于五种能源的方程
lr_CO2D4 = LinearRegression()
lr_CO2D4.fit(Eall, CO2D4)
coef_CO2D4 = lr_CO2D4.coef_
intercept_CO2D4 = lr_CO2D4.intercept_
R2_CO2D4 = lr_CO2D4.score(Eall, CO2D4)

# 求解 CO2D5 关于五种能源的方程
lr_CO2D5 = LinearRegression()
lr_CO2D5.fit(Eall, CO2D5)
coef_CO2D5 = lr_CO2D5.coef_
intercept_CO2D5 = lr_CO2D5.intercept_
R2_CO2D5 = lr_CO2D5.score(Eall, CO2D5)

# 求解 CO2D6 关于五种能源的方程
```

```
    lr_CO2D6 = LinearRegression()
    lr_CO2D6.fit(Eall, CO2D6)
    coef_CO2D6 = lr_CO2D6.coef_
    intercept_CO2D6 = lr_CO2D6.intercept_
    R2_CO2D6 = lr_CO2D6.score(Eall, CO2D6)

    # 计算得到 CO2 的方程
    ################################################################################
################################################################
    predict_EV1 = coef_EV1[0] * predict_gdp + coef_EV1[1] * predict_p + intercept_EV1
    predict_EV2 = coef_EV2[0] * predict_gdp + coef_EV2[1] * predict_p + intercept_EV2
    predict_EV3 = coef_EV3[0] * predict_gdp + coef_EV3[1] * predict_p + intercept_EV3
    predict_EV4 = coef_EV4[0] * predict_gdp + coef_EV4[1] * predict_p + intercept_EV4
    predict_EV5 = coef_EV5[0] * predict_gdp + coef_EV5[1] * predict_p + intercept_EV5

    predict_CO2D1 = coef_CO2D1[0] * predict_EV1 + \
                    coef_CO2D1[1] * predict_EV2 + \
                    coef_CO2D1[2] * predict_EV3 + \
                    coef_CO2D1[3] * predict_EV4 + \
                    coef_CO2D1[4] * predict_EV5 + intercept_CO2D1

    predict_CO2D2 = coef_CO2D2[0] * predict_EV1 + \
                    coef_CO2D2[1] * predict_EV2 + \
                    coef_CO2D2[2] * predict_EV3 + \
                    coef_CO2D2[3] * predict_EV4 + \
                    coef_CO2D2[4] * predict_EV5 + intercept_CO2D2

    predict_CO2D3 = coef_CO2D3[0] * predict_EV1 + \
                    coef_CO2D3[1] * predict_EV2 + \
                    coef_CO2D3[2] * predict_EV3 + \
                    coef_CO2D3[3] * predict_EV4 + \
                    coef_CO2D3[4] * predict_EV5 + intercept_CO2D3

    predict_CO2D4 = coef_CO2D4[0] * predict_EV1 + \
                    coef_CO2D4[1] * predict_EV2 + \
                    coef_CO2D4[2] * predict_EV3 + \
                    coef_CO2D4[3] * predict_EV4 + \
                    coef_CO2D4[4] * predict_EV5 + intercept_CO2D4

    predict_CO2D5 = coef_CO2D5[0] * predict_EV1 + \
                    coef_CO2D5[1] * predict_EV2 + \
                    coef_CO2D5[2] * predict_EV3 + \
                    coef_CO2D5[3] * predict_EV4 + \
                    coef_CO2D5[4] * predict_EV5 + intercept_CO2D5
```

```
    predict_CO2D6 = coef_CO2D6[0] * predict_EV1 + \
                    coef_CO2D6[1] * predict_EV2 + \
                    coef_CO2D6[2] * predict_EV3 + \
                    coef_CO2D6[3] * predict_EV4 + \
                    coef_CO2D6[4] * predict_EV5 + intercept_CO2D6
    predict_CO2 = predict_CO2D1 + predict_CO2D2 + predict_CO2D3 + predict_CO2D4 +
predict_CO2D5 + predict_CO2D6
    # 求解碳达峰时间
    predict_CO2_d = sp.diff(predict_CO2, x)
    t1 = sp.solve(predict_CO2_d)
    t1 = int(t1[0])

    # 满足 2030 年碳达峰条件
    if (t1>2020) & (t1<2050):
        sign1 = True

    # 新建预测结果
    system_pre = pd.DataFrame(columns=system.columns)
    system_pre['时间'] = [i for i in range(2010, 2061)]

    for i in range(system_pre.shape[0]):
        t = system_pre.loc[i, ['时间']][0] # 获取第 i 个时间
        system_pre.loc[i, ['常驻人口总量']] = predict_p.subs(x, t)
        system_pre.loc[i, ['区域生产总值']] = predict_gdp.subs(x, t)
        system_pre.loc[i, ['一次能源中化石的能源消费量']] = predict_EV1.subs(x, t)
        system_pre.loc[i, ['一次能源中非化石的能源消费量']] = predict_EV2.subs(x,
t)
        system_pre.loc[i, ['二次能源的能源消费量']] = predict_EV3.subs(x, t)
        system_pre.loc[i, ['化石能源发电的能源消费量']] = predict_EV4.subs(x, t)
        system_pre.loc[i, ['非化石能源发电的能源消费量']] = predict_EV5.subs(x, t)
        system_pre.loc[i, ['农林消费部门的碳排放量']] = predict_CO2D1.subs(x, t)
        system_pre.loc[i, ['工业消费部门的碳排放量']] = predict_CO2D2.subs(x, t)
        system_pre.loc[i, ['交通消费部门的碳排放量']] = predict_CO2D3.subs(x, t)
        system_pre.loc[i, ['建筑消费部门的碳排放量']] = predict_CO2D4.subs(x, t)
        system_pre.loc[i, ['居民生活消费部门的碳排放量']] = predict_CO2D5.subs(x,
t)
        system_pre.loc[i, ['能源供应部门的碳排放量']] = predict_CO2D6.subs(x, t)
        system_pre.loc[i, ['碳排放量']] = predict_CO2.subs(x, t)

    # 判断是否全都为正
    has_negative = (system_pre['碳排放量'] > 0).all()
    if has_negative:
        sign2 = True
```

```
    time = [2020, 2025, 2030, 2035, 2050, 2060]
    goal_2 = pd.DataFrame({'时间' : time})
    goal_2['一次能源中化石的能源消费量'] = 99
    goal_2['一次能源中非化石的能源消费量'] = 99
    goal_2['二次能源的能源消费量'] = 99
    goal_2['化石能源发电的能源消费量'] = 99
    goal_2['非化石能源发电的能源消费量'] = 99
    goal_2['碳排放量'] = 99
    for i in time:
        if i == 2020:
            goal_2.loc[goal_2['时间']==i, ['一次能源中化石的能源消费量']] =
system.loc[system['时间']==i, ['一次能源中化石的能源消费量']].values[0]
            goal_2.loc[goal_2['时间']==i, ['一次能源中非化石的能源消费量']] =
system.loc[system['时间']==i, ['一次能源中非化石的能源消费量']].values[0]
            goal_2.loc[goal_2['时间']==i, ['二次能源的能源消费量']] =
system.loc[system['时间']==i, ['二次能源的能源消费量']].values[0]
            goal_2.loc[goal_2['时间']==i, ['化石能源发电的能源消费量']] =
system.loc[system['时间']==i, ['化石能源发电的能源消费量']].values[0]
            goal_2.loc[goal_2['时间']==i, ['非化石能源发电的能源消费量']] =
system.loc[system['时间']==i, ['非化石能源发电的能源消费量']].values[0]
            goal_2.loc[goal_2['时间']==i, ['碳排放量']] = system.loc[system['时
间']==i, ['碳排放量']].values[0]
        else:
            goal_2.loc[goal_2['时间']==i, ['一次能源中化石的能源消费量']] =
system_pre.loc[system_pre['时间']==i, ['一次能源中化石的能源消费量']].values[0]
            goal_2.loc[goal_2['时间']==i, ['一次能源中非化石的能源消费量']] =
system_pre.loc[system_pre['时间']==i, ['一次能源中非化石的能源消费量']].values[0]
            goal_2.loc[goal_2['时间']==i, ['二次能源的能源消费量']] =
system_pre.loc[system_pre['时间']==i, ['二次能源的能源消费量']].values[0]
            goal_2.loc[goal_2['时间']==i, ['化石能源发电的能源消费量']] =
system_pre.loc[system_pre['时间']==i, ['化石能源发电的能源消费量']].values[0]
            goal_2.loc[goal_2['时间']==i, ['非化石能源发电的能源消费量']] =
system_pre.loc[system_pre['时间']==i, ['非化石能源发电的能源消费量']].values[0]
            goal_2.loc[goal_2['时间']==i, ['碳排放量']] =
system_pre.loc[system_pre['时间']==i, ['碳排放量']].values[0]
    goal_2['能源利用效率'] = goal_2['碳排放量'] / \
                                      (goal_2['一次能源中化石的能源消费量'] + \
                                      goal_2['一次能源中非化石的能源消费量'] + \
                                      goal_2['二次能源的能源消费量'])
    goal_2['非化石能源消费比重'] = (goal_2['一次能源中非化石的能源消费量'] + goal_2['
二次能源的能源消费量']) / \
                                            (goal_2['一次能源中化石的能源消费量']
+ goal_2['一次能源中非化石的能源消费量'] + goal_2['二次能源的能源消费量'] )
```

```
    if goal_2.loc[goal_2['时间']==2060, ['非化石能源消费比重']].values[0][0] > 0.8:
        sign3 = True

    if sign1 & sign2:
        print(goal_2.loc[goal_2['时间']==2060, ['非化石能源消费比重
']].values[0][0])
        print("满足线性规划模型的人口={}碳达峰时间={}".format(w, t1))
```