北京郵電大學

数 学 建 模 结 题（论文）

题目：确定最优劳动人口率与老龄化率的人口政策模型

课题名称：数学建模与模拟

姓名：韩志浩

学号：2021213157

学院：国 际 学 院

专业：物 联 网 工 程

摘 要

当前，我国正在面临着生育率过低，人口老龄化速度加快的社会阶段。以往研究仅集中探讨宏观政策问题，然而，忽略了人口生育意愿对生育率的影响以及教育、医疗等政策对于生育意愿的影响。在这篇文章中，我们首先建立了人口随时间演化的差分方程，并引入了迁移矩阵，来模拟人口在都市、城镇、乡村三个地区流动的情况，对未来五十年各地区人口数量变化进行预测，并求出了理想最优老龄化率。随后，建立了生育模式模型，客观地反映出各政策下、各胎次的生育年龄概率分布。接下来，我们分析了不同因素对生育意愿的影响，并探讨了相关基础设施如何影响对应因素，并且在短期和长期时间内给政府提供了修建策略。最后，确定了能够使得老龄化率最低的政策和基础设施的修建模式。

关键词：人口老龄化 人口政策 **生育意愿 凸优化**

目录

题目：确定最优劳动人口率与老龄化率的人口政策模型......................1
一、研究背景......1
二、问题的提出......2
三、假设和符号......3
3.1 问题的假设......3
3.2 符号说明......3
四、模型的建立与求解......4
4.1 模型一:......4
4.2 模型二：......5
4.3 模型三：......5
五、模型的分析......15
参考文献......17
附　　录......18
附录 1：绘制人口数量变化图代码......18
附录 2：计算生育意愿参数代码......19
附录 3：计算最佳基础设施安排代码......22
附录 4：推演人口数量变化代码......23

一、 研究背景

2021 年全年中国出生人口 1062 万人，人口出生率为 7.52‰；死亡人口 1014 万人，人口死亡率为 7.18‰；根据公式(*人口增长率=（年末人口数-年初人口数）/年平均人口×1000‰*)可以计算出，中国的人口自然增长率为 0.34‰，已经创下了中国建国以来人口增长率的最低记录。联合国预计，中国将最早可能在 2023 年进入人口负增长阶段。

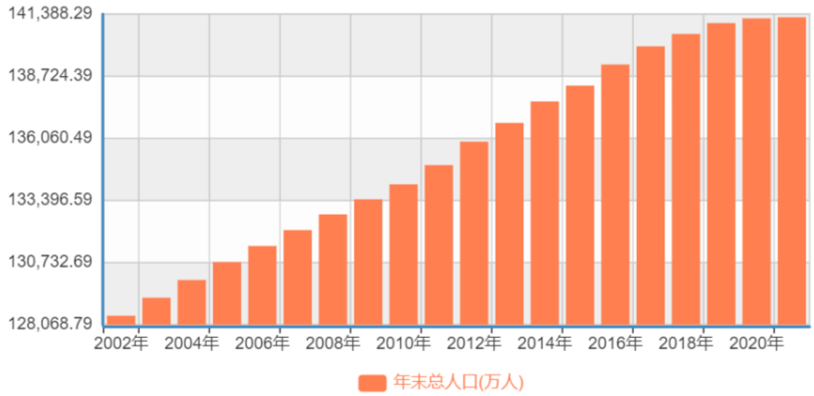

图 1 我国历年人口总数

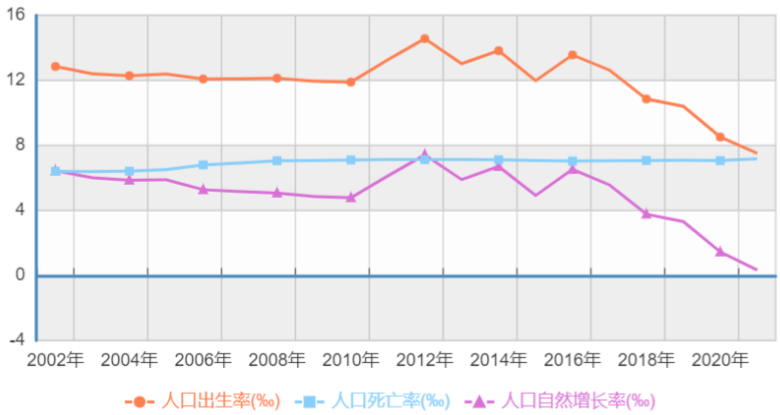

图 2 中国历年人口出生率、死亡率和自然增长率

此外，近年来中国劳动人口处于下降趋势，根据公式(*总抚养比（即赡养率）=（老龄人口+未成年人口）/劳动力人口= 老龄人口抚养比+未成年人口抚养比*)，我们可以得出中国的劳动人口的抚养比。由下图可以看出，中国的总抚养比正在不断上升，而抚养比越大，就意味着中国劳动力平均承担的抚养人数就越多，劳动者的抚养负担就越重，可见我国的劳动人口率（劳动人口占总人口的比例）在未来将会明显下降。而据中国第七次人口普查结果显示，中国 65 岁以上老年人口已经达到两亿以上，占总人口的 14.2%，这意味着约每 7 个中国人中就有一个老年人。按照联合国公布的标准，65 岁以上人口占比达到 7%，即为老龄化社会；65 岁以上人口比例达到 14%，为深度老龄化社

会，达到 20%为超级老龄化社会。可见，我国已经处于深度老龄化社会，并且正在向超级老龄化社会发展。

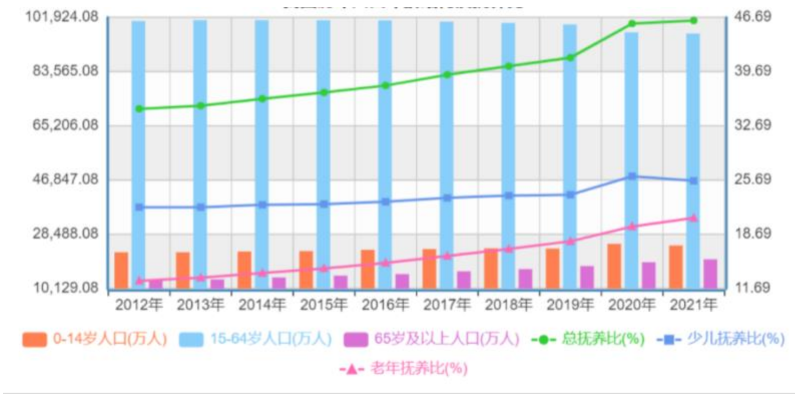

图 3 我国历年人口年龄结构及抚养比

二、 问题的提出

当前我国的人口问题的出现，很大程度上是源于我国 1971 年开始推行的计划生育政策。当时由于人口理论尚不完备，为了提高人口质量，使我国提前进入从经典人口转变模型中的高出生率高死亡率模式转换为低出生率低死亡率的平替模式，故采取了严格的计划生育政策，但是由于错过了政策调整机遇期，我国的人口问题逐渐浮出水面，尽管我国在 2011 年后开始放宽了计划生育政策，开始逐步推出“二孩政策”、“三孩政策”但是效果并不显著。如下图所示，尽管在 2016 年“二孩政策”出台后的短期内有所回升，但是随即生育率又出现了显著的下滑。

图 4 中国常住人口生育率

由于我国现有政策并没有达到预期的效果，因此我们希望建立一个确定最优劳动人口率与老龄化率的人口政策模型，通过对生育意愿等因素的分析来得出合理的政策使得中国的劳动人口率稳定维持在较高水平，让人口老龄化率实现软着陆。

三、 假设和符号

3.1 问题的假设

人口政策是一个涉及到多种因素的复杂问题。它涉及到政治、经济、文化、地理等多个方面，我们不可能把所有情况都加以考虑并模拟，因此我们做出了以下合理的假设，并给出了假设原因。

a. 忽略我国的人口迁入和迁出

原因：由于我国的民族主体观念较强，比较排斥人口的迁入迁出，并且人口的迁入迁出可以在一定程度上相互抵消，每年迁入迁出的人口数量差相较总人口数来说较少，所以忽略不计。

b. 将医疗水平以卫生人口占比代替

原因：医疗、养老、教育这些因素对个体生育意愿的影响是复杂、难以量化的，所以我们将医疗水平以卫生人口占比代替，认为医院、养老院、公立学校数量分别会按比例影响卫生人口占比、赡养负担、抚养负担。

c. 假设分年龄段死亡率是一个常数

原因：分年龄段死亡率在几乎对我们的模型不影响。

d. 忽略三孩政策影响

原因：三孩政策刚刚实施影响尚不明确，且数据尚未公布。

3.2 符号说明

表 1 符号说明

符号	**说明**
r	人口增长率
y	年
m	地
a	岁
m	都市
c	城镇
v	农村
$X_m^a(y)$	人口数量

$P_k^j(i)$	j 政策下第 k 胎在 i 岁生的概率。
W_k^i	i 岁第 k 胎的生育意愿。
$n_k^m(i,j)$	j 政策下 m 地 i 岁人口生第 k 胎数量。
$e_m^t(y)$	第 y 年向 m 地迁入 t 岁的人口数量
C_m	表示 m 地的生育常数
S	最大人口年龄
GDP	国内生产总值
P_r	赡养压力

四、 模型的建立与求解

我们将在第 y 年 m 地 t 岁的人口数量定义为$X_m^t(y)$，以及在第 y 年向 m 地迁入 t 岁的人口数量为$e_m^t(y)$。

m 地人口分布向量：

$$X_m(y)=\begin{bmatrix}X_m^1(y)\\X_m^2(y)\\\vdots\\X_m^n(y)\end{bmatrix} \tag{1}$$

$$e_m(y)=\begin{bmatrix}e_m^1(y)\\e_m^2(y)\\\vdots\\e_m^n(y)\end{bmatrix} \tag{2}$$

4.1 模型一:

首先假设人口增长率不变，设人口数关于时间的连续函数为$X(y)$，可得：

$$\frac{dX}{dy}=rX(y) \tag{3}$$

$$X(0)=X_0 \tag{4}$$

得出：

$$X(y)=X_0e^{ry} \tag{5}$$

- 结果分析：建模结果与实际严重不符

- 原因分析：假定的人口增长率不变过于理想，显然不成立
- 解决方案：由于最大环境容纳量 K 的限制，增长率 r 应与人口数目成负相关，假设 $r = r_0(1 - X/K)$,建立模型二。

4.2 模型二：

$$\frac{dX}{dy} = r_0 \cdot \frac{1-X}{K} \tag{6}$$

$$X(0) = X_0 \tag{7}$$

得出：

$$X(y) = \frac{K}{1 + \frac{K \cdot e^{-ry}}{X_0 - 1}} \tag{8}$$

- 结果分析：

模型在人口增长阶段大致拟合，但在人口稳定阶段和实际情况偏差较大

- 原因分析：

1. 由于生育意愿、地区政策等原因，人口年龄结构无法一直保持正常，随着生育意愿下降等原因，我国老龄化比例日益增高，破坏了原有的稳定的人口年龄组成结构，使得人口难以在 K 值处保持稳定，甚至会出现人口下降等情况；
2. 由于城市之间的经济、政策、风俗之间存在差异，使得城市间人口变化存在差异；

- 解决方案：

针对原因一，我们建立模型三来进一步展示老龄化对人口的影响

4.3 模型三：

首先假设男女比为 1:1，设人口年龄最大为 S，我们将人口按年龄分为 m 段。

假设每单位时间（即S/m年）获取一次结果，设$X_i(y)$为第 i 组年龄人口（即年龄为（i-1）*m/S 到 i*m/S 的人口）在第 t 次获取的数量结果。

设第 i 组的生育率为 a_i(0-14 岁、50 岁以上默认生育率为 0)，死亡率为 d_i，则存活率$S_i = 1 - d_i$。

我们选取某政策下比较有代表性的一年的各年龄段生育率进行归一化处理，作为生育模式模型的生育率参数，即 j 政策下 i 岁人口生育第 k 胎的概率表示$P_k^j(i)$。

2008 年（计划生育）：

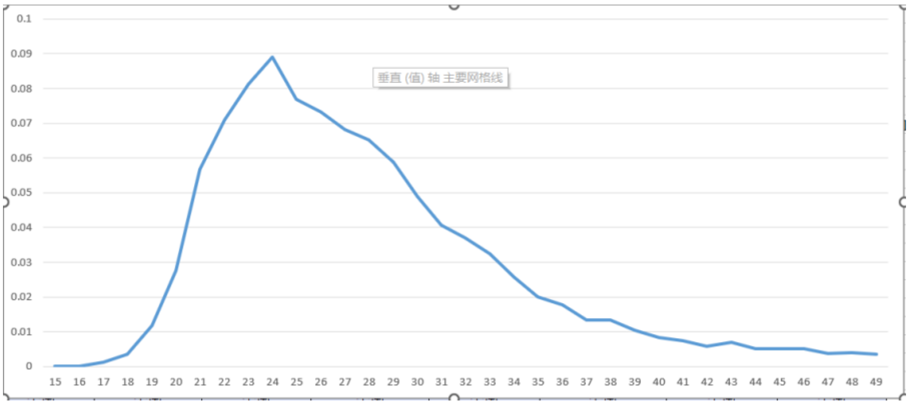

图 5 独生子女政策下 15 岁到 49 岁生育率参数

2014 年（双独二孩）：

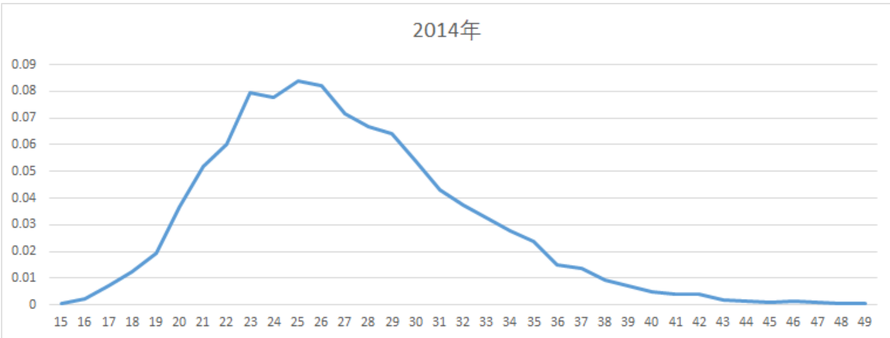

图 6 双独二孩政策下 15 岁到 49 岁生育率参数

2020 年（全面二孩）：

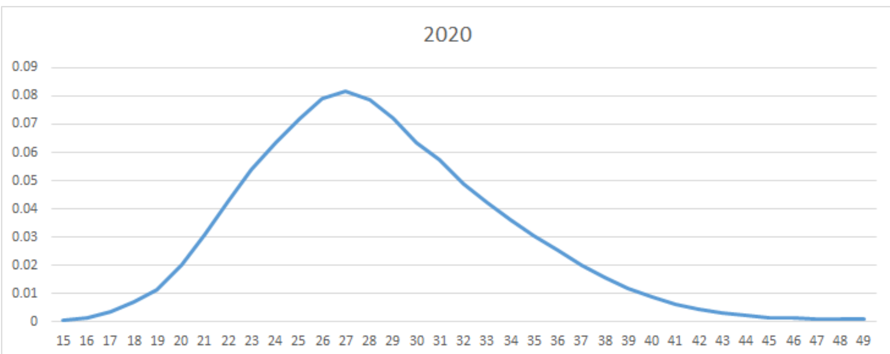

图 7 全面二孩政策下 15 岁到 49 岁生育率参数

分析得到，每个地区的生育率为生育模式的概率乘以不同参数，以 m 地为例：

$$b_m^t(y) = C_m P_k^t(i) \tag{9}$$

其中，C_m表示 m 地的生育常数。

在第 y+1 次观察时，第一组人口为第 y 次观察时所有人口所生育的人口，默认地区为 m 地，即：

$$X_1^m(y+1) = \sum_{i=1}^{n} a_i \cdot X_i^m(y) \tag{10}$$

此时第 i 组（i>1）的人口为上一次观察时第 i-1 组人口乘以存活率，即：

$$X_{i+1}^m(y+1) = s_i X_i^m(y) \tag{11}$$

将上述公式转换为矩阵形式，可以得到：

$$L = \begin{bmatrix} 0 & \cdots\cdots & 0 & b_m^{a_1}(y) & \cdots & b_m^{a_2}(y) & 0 & \cdots & 0 \\ S^1(h) & \cdots\cdots & \cdots & \cdots\cdots & \cdots & \cdots\cdots & \cdots & \cdots & \cdots\cdots \\ 0 & S^2(h) & \cdots & \cdots\ \cdots & \cdots & \cdots\cdots & \cdots & \cdots & \cdots\cdots \\ \cdots & \cdots\cdots & \cdots & \cdots\ \cdots & \cdots & \cdots\cdots & \cdots & \cdots & 0 \\ 0 & \cdots\cdots & \cdots & \cdots\ \cdots & \cdots & \cdots\cdots & \cdots & 0 & S^n(h) \end{bmatrix} \tag{12}$$

由此我们可以得出新的一年的各年龄段的人口数，将上述算法进行多次迭代即可预估一个城市 x 年后的人口数。

将预计人口数与城市起始人口老龄化率进行相关性分析，即可得出老龄化对人口增长的影响。

a 岁人口迁移矩阵 T_a：

$$[e_m^t(y) \quad e_c^t(y) \quad e_v^t(y)] = [X_m^t(y) \quad X_c^t(y) \quad X_v^t(y)] \times \begin{bmatrix} r_{mm} & r_{mc} & r_{mv} \\ r_{cm} & r_{cc} & r_{cv} \\ r_{vm} & r_{vc} & r_{vv} \end{bmatrix} \tag{13}$$

初始状态下，我们假设有 c、m、v 三座城市，以第一行第二列的元素 r_{cm} 为例,该元素表示从 c 地到 m 地迁移人口和 c 地总人口的比值，我们将其称作迁移率，根据人口迁移率我们可以轻松看到不同地区的迁移意愿。

因此，结合上文的人口分布矩阵和人口迭代矩阵，可以推出：

$$X_m(y+1) = L \times X_m(y) + e_m(y)$$

根据上述模型，我们建模求得的近 50 年三个地区的人口变化如下。

都市的 50 年人口数量变化：

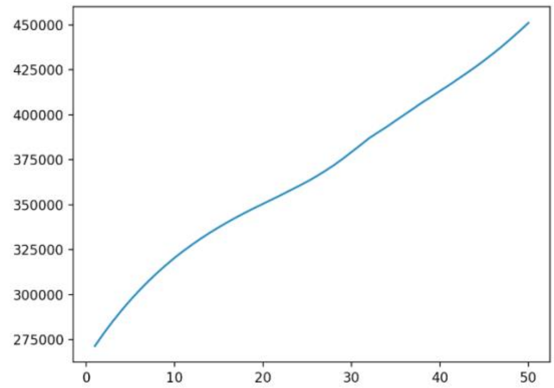

图 8 都市的 50 年人口数量变化

城镇的 50 年人口数量变化：

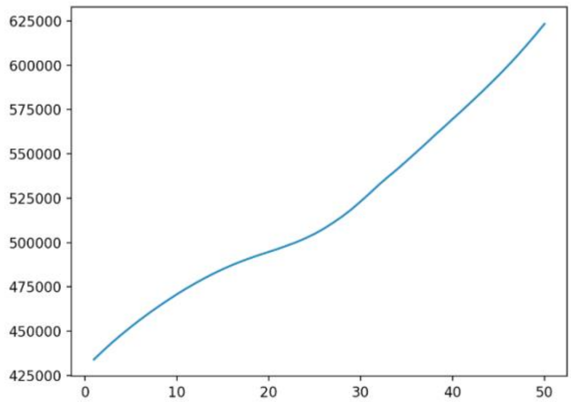

图 9 城镇的 50 年人口数量变化

乡村的 50 年人口数量变化：

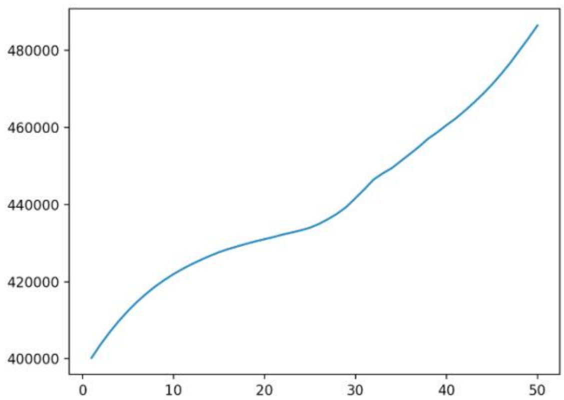

图 10 乡村的 50 年人口数量变化

普遍而言，适龄生育人口进行生育的时间与所处年龄有较强的相关性。根据上文分析，我们发现在不同年份实行同一政策时，不同年龄段的人生育率出现了同步调的变化，因此认为基础的生育模式只与政策相关。我们分别对计划生育、双独二孩、全面二孩进行建模，模型反映了适龄人口群体在期望生育人口为一时在对应年龄的概率。

我们将寻求理想老龄化问题定义为在 k 年后的最高老龄化率（以年为单位统计）。因此可以转换为一个优化问题，默认在 m 地区：

$$\min \max_{y} \frac{\sum_{i=65}^{n} X_m^i(y)}{\sum_{i=1}^{n} X_m^i(y)} \tag{14}$$

$$s.t. y \geq k;\ n > 0;\ X_m^i(y) \geq 0, 1 \leq i \leq n \tag{15}$$

假设当前处于最佳状态，即老龄化率维持稳定。

若忽略人口迁移，则第 k 岁人口数可表示为：

$$X_m^k(y) = \left(\prod_{i=1}^{k-1}(1 - S_i)\right) \times X_b \tag{16}$$

X_b 为常数，即每年出生人口。

因此，最优老龄化率表示为：

$$\frac{\sum_{i=65}^{n}(\prod_{j=1}^{i-1} 1 - S_j)}{\sum_{i=1}^{n}(\prod_{j=1}^{i-1} 1 - S_j)} \tag{17}$$

我们发现，该结果只与每岁的死亡率有关，经过计算，发现最理想的老龄化率为 4.027%。

下图是不同政策下出生的一孩、二孩母亲的年龄分布。并且根据题目，我们注意到，实际生育情况与我们求得的生育模式有一定出入。因而，除实行政策外，还有其他因素影响了生育情况。在本文中，我们考虑了地区发展、抚养压力、赡养压力三个因素，并将其归于生育意愿。

2014 年：

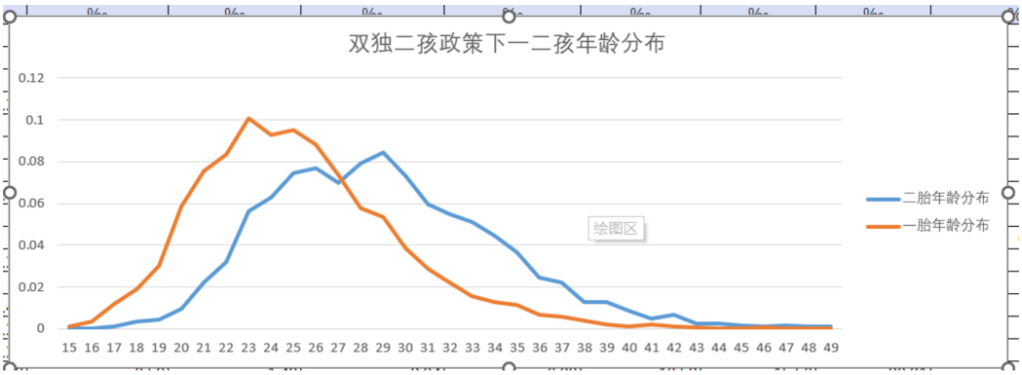

图 11 双独二孩政策下一二孩年龄分布

14 年总二孩/一孩=0. 62982

2020 年：

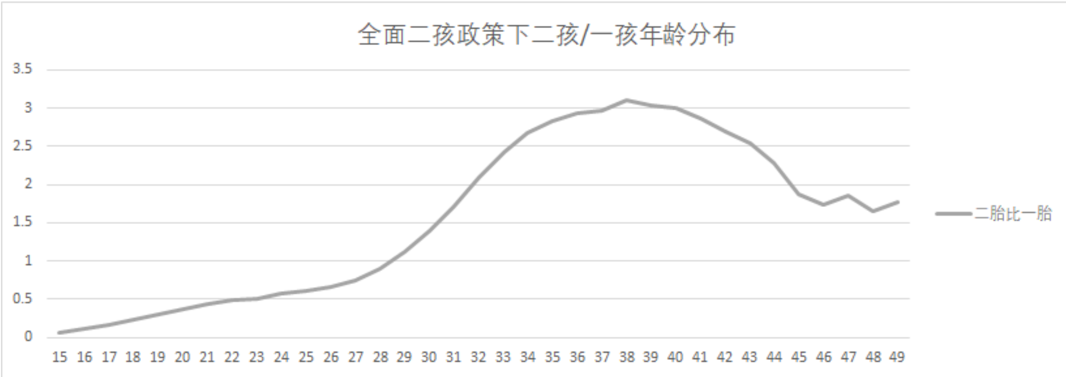

图 12 全面二孩政策下二孩/一孩年龄分布

20 年总二孩/一孩=0. 99766

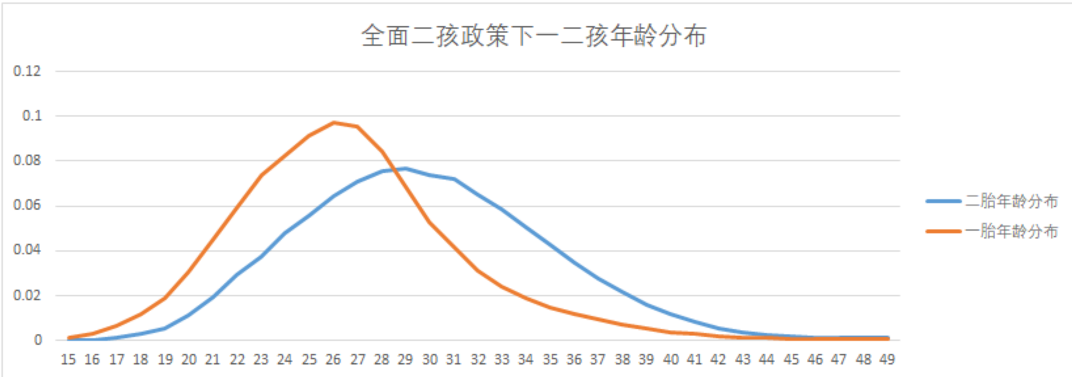

图 13 全面二孩政策下一二孩年龄分布

在本文中，默认只有三种生育政策可供选择，即计划生育、双独二孩、全面二孩，并依次命名为 1、2、3。

我们定义赡养压力为 $P_r = \frac{老年人口数}{劳动人口数}$，抚养压力为$P_s = \frac{k}{2}$，其中，k 为生育的是第 k 胎，由于孩子由父母双方抚养，所以需要对压力除以二，此处两个变量结果均小于等于 1。因此可以得到生育意愿的结果：

$W_k^i = \alpha_{k0} \times GDP^{\alpha_{k1}} \cdot P_r^{\alpha_{k2}} \cdot P_s{}^{\alpha_{k3}}$，其中，$\alpha$为可优化的参数，$\alpha$的初始值是期望为 1、方差为 0.3 正态分布的随机数。

之后，对上述式子两边取对数，可得：

$$lnW_k^i = \alpha_{0k}^i + \alpha_{1k}^i lnGDP + \alpha_{2k}^i lnP_r + \alpha_{3k}^i lnP_s \tag{18}$$

注意到，为了确保 $W_k^i \in (0,1]$，我们对W_k^i进行归一化处理，选取 softmax 为激活函数，即：

$$W_k^i = \frac{e^{W_k^i}}{\sum_{j=15}^{49} e^{W_k^j}} \tag{19}$$

对 GDP 该常量的处理亦采用该方式，以 m 地的 GDP 为例：

$$GDP_m = \frac{e^{GDP_m}}{\sum_{i=15}^{(m,c,v)} e^{GDP_i}} \tag{20}$$

我们可以得到 j 政策下 m 地 i 岁人口生第 k 胎数量：

$$n_k^m(i,j) = W_k^i \times P_k^j(i) \times X_m^i(y) \tag{21}$$

根据上述公式，将 j 政策下的损失函数定义为：

$$L_j = -\frac{1}{3*y}\sum_y \sum_m (X_m^0(y) - \sum_k^{(1,2)} \sum_{i=15}^{49} n_k^m(i,j))^2 + \sum_k^{(1,2)} \sum_{i=15}^{49} \sum_j^{(0,1,2,3)} (\ \alpha_{jk}^i)^2 \tag{22}$$

其中，$(\alpha_{jk}^i)^2$为正则化项，防止模型过拟合。

之后，使用梯度下降法更新α，更新后的结果为α'。

$$\alpha' = \alpha - \beta \cdot \frac{dL_j}{d\alpha} \tag{23}$$

其中，β为学习率，默认取值为 0.1。

这里我们以为α_2例，计算每一轮中对参数的更新。

$$\begin{aligned} \alpha_2' &= \alpha_2 - 2\beta \cdot \left(-\frac{1}{3*y}\right)\left(n_k^m(i,j) - X_m^0(y)\right) \cdot \frac{dn_k^m(i,j)}{d\alpha} + 2 \cdot \alpha_2 \\ &= 3\alpha_2 + 2\beta \cdot \frac{1}{3*y}\left(n_k^m(i,j) - X_m^0(y)\right) \cdot \frac{dn_k^m(i,j)}{d\alpha} \end{aligned} \tag{24}$$

$$= 3\alpha_2 + 2\beta \cdot \frac{1}{3*y}\left(n_k^m(i,j) - X_m^0(y)\right) \cdot P_k^j(i) \times X_m^i(y) \cdot W_k^i \cdot (1 - W_k^i) \cdot \frac{dW_k^i}{d\alpha}$$

$$= 3\alpha_2 + 2\beta \cdot \frac{1}{3*y}\left(n_k^m(i,j) - X_m^0(y)\right) \cdot P_k^j(i) \times X_m^i(y) \cdot W_k^i \cdot (1 - W_k^i) \cdot W_k^i \times lnP_r$$

根据上述模型，我们求得了α矩阵，其形状为 4*35*2，其中“35”代表本文中定义的可生育年龄的区间为 35 岁，“2”代表生育第一胎、第二胎不同的意愿情况。部分α的结果展示如下：

表 2 部分α结果

年龄 \ 参数	α_{01}	α_{11}	α_{21}	α_{31}
25 岁	0.3423	-0.5769	-0.3842	-0.5621
26 岁	0.3428	-0.5781	-0.3876	-0.5582

根据图 xx 反映的生育累计概率分布，我们对每个α进行加权求和，得到新的参数α_j^p：

$$\alpha_j^p = \sum_{k}^{(1,2)} \sum_{i=15}^{49} P_k(i) * \alpha_{jk}^i \tag{25}$$

其中， p 是指当前采取的宏观人口政策。

表 3

政策 \ 参数	α_0	α_1	α_2	α_3
计划生育	0.2468	-0.2416	-0.5482	-0.2271
双独二孩	0.4627	-0.2742	-0.3781	-0.5369
全面二孩	0.6483	-0.3657	-0.2039	-0.7245

通过观察表格中的数据，我们可以发现，随着政策逐渐由收紧转向放松，整体的生育意愿呈现出稳定上升的趋势。但是，与赡养压力相关的参数显示出减小的趋势（有利于促进生育），而抚养压力和地区经济相关的参数显示出增大的趋势（有利于抑制生育）。这表明，政策的放松在经济较为落后的地区效果更好，并且可以减轻赡养压力，加重抚养压力。由于开设学校、养老院、医院等基础设施难以估计其影响，我们在此分别假设这三种基础设施对上述模型产生的影响：

学校：[9]认为教育资源的有限性使得多数家庭为获取优质教育资源展开激烈竞争,导致家庭生育焦虑的滋生,降低了家庭生育意愿。因此，我们假设当政府建立 n 个学校时，抚养压力P_s会衰减到$0.95^{\sqrt{n}} * P_s$，根号起衰减作用，反映基础设施会饱和。

养老院：[10]提到，打造健全的新型养老模式，完善养老制度，与育龄人口生育意愿呈正相关。我们假设当政府建立 n 个养老院时，赡养压力P_r会衰减到$0.95^{\sqrt{n}} * P_r$，根号起衰减作用，反映基础设施会饱和。

医院：[11]认为医疗发展的提高对于死亡率的改善在低年龄段的人口更加明显。因此，我们假设当政府建立 n 个医院时，对 i 岁人口的死亡率会进行如下变换：

$$S_i = (1 - 0.05^{0.02*i/\sqrt[4]{(n+1)}}) * S_i \quad , \qquad 15 \le i \le n \tag{25}$$

注意到，新建基础设施产生的影响是在未修建时的生育意愿上进行乘法运算，不妨假设未修建基础设施的生育意愿为 C，且该城市修建了k_1所学校，k_1所养老院，k_3所医院。

在短期过程中，医院对结果产生的影响可以忽略不计，故 j 政策下新的生育意愿W_j可以表示为：

$$W_j = C * 0.95^{\alpha_3 * \sqrt{k_1}} * 0.95^{\alpha_2 * \sqrt{k_2}} \tag{26}$$

因此，我们只需求解:

$$\begin{gathered} \min_{k1} \ \alpha_3 * \sqrt{k}_1 + \alpha_2 * \sqrt{(6-k_1)} \\ s.t.\ 0 \le k_1 \le 6,\ \alpha_3 < 0,\ \alpha_2 < 0 \end{gathered} \tag{27}$$

根据求解，我们得出以下结论：

1、计划生育政策下的最优解为 k1=1、k2=5、k3=0；

2、双独二孩政策下的最优解为 k1=4、k2=2、k3=0；

3、全面二孩政策下的最优解为 k1=5、k2=1、k3=0；

可见，当政策逐渐开放时，短期内学校相比养老院对于提高生育意愿的程度更大。

而在长期过程中，赡养压力P_r会产生如下变化，记原赡养压力为C_p，且当 i 大于 15 岁时，$0.05^{0.2*i/n}$在零点附近，根据不等式，有$e^x \approx x + 1$，我们可以求解：

$$\begin{aligned} P_r &= \frac{\sum_{i=65}^{n} \prod_{j=1}^{i} (1 - 0.05^{0.2*j/\sqrt[4]{(n+1)}}) * S_j}{\sum_{i=15}^{64} \prod_{j=1}^{i} (1 - 0.05^{0.2*j/\sqrt[4]{(n+1)}}) * S_j} \\ &\approx \frac{\sum_{i=65}^{114} \prod_{j=1}^{i} (0.02j * \ln{(20)}/\sqrt[4]{(n+1)}) * S_j}{\sum_{i=15}^{64} \prod_{j=1}^{i} (0.02j * \ln{(20)}/\sqrt[4]{(n+1)}) * S_j} \\ &= \frac{\sum_{i=65}^{114} \prod_{j=i-50}^{i} (0.02j * \ln{(20)}/\sqrt[4]{(n+1)}) * S_j}{\sum_{i=15}^{64} \prod_{j=1}^{i} j * S_j} \end{aligned} \tag{28}$$

$$\approx C_p \times \frac{0.02 * \ln(20)}{\sqrt[4]{(n+1)}}$$

因此可以得到长期过程中，j 政策下新的生育意愿W_j可以表示为：

$$W_j = C \times 0.95^{\alpha_3 * \sqrt{k_1}} \times 0.95^{\alpha_2 * \sqrt{k_2}} \times (\frac{0.02 * \ln(20)}{\sqrt[4]{(n+1)}})^{\alpha_2} \tag{29}$$

类似短期的求解过程，我们也得出以下结论：

1、计划生育政策下的最优解为 k1=0、k2=1、k3=5；

2、双独二孩政策下的最优解为 k1=1、k2=1、k3=4；

3、全面二孩政策下的最优解为 k1=3、k2=0、k3=3；

根据我们假设的环境，我们发现，长期来看，养老院的作用要弱于医院和学校来提高生育意愿。随着政策的放松，学校的重要性会增强，而医院的作用则会减弱。如果我们考虑地区因素，即生育意愿中的 GDP 因素，我们发现，在都市和城市建设学校会更有利于提高整体生育意愿。这可能是因为发达的经济会导致“少子化”的生育理念，而且这些地区的年轻人口比例较大，教育资源也较为紧张，使得学校的促进效果更显著。而在乡村地区，修建医院和养老院可能会更有助于提高整体生育意愿。这可能是由于落后地区的老年人口比例较大，导致养老负担较重，而且该类地区医疗资源相对稀缺，因此死亡率有较大的改善空间。

结合我们所提出的人口迁移迭代模型、生育模式模型、生育意愿模型、基础设施效益模型，假设某地级市（这里按照城镇地区分析）每十年修建一个基础设施。发现以全面二孩政策，并且依次以（医院、医院、学校、养老院、学校）总数为 5 的循环进行修建，所得到的老龄化结果最优。

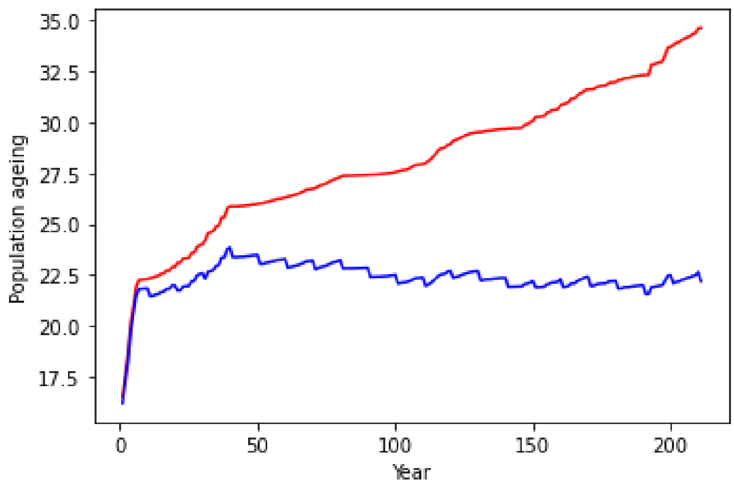

图 14 使用模型前后老龄化对比

五、 模型的分析

政策建议：由上述模型可得老龄人口的最优占比为 4.027%，相比于当前中国 18.70%的老龄化占比，该数值明显偏低，所以当前的首要目标是降低老龄人口的占比以此来缓解老龄化进一步加重导致的种种问题，根据 2022 年中国人口年龄金字塔（见下图）来看

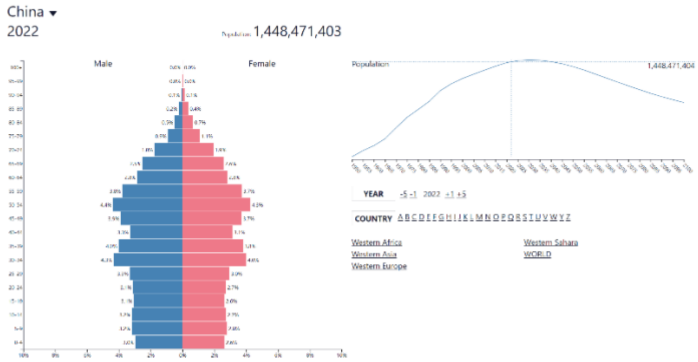

图 15 2022 年中国人口金字塔

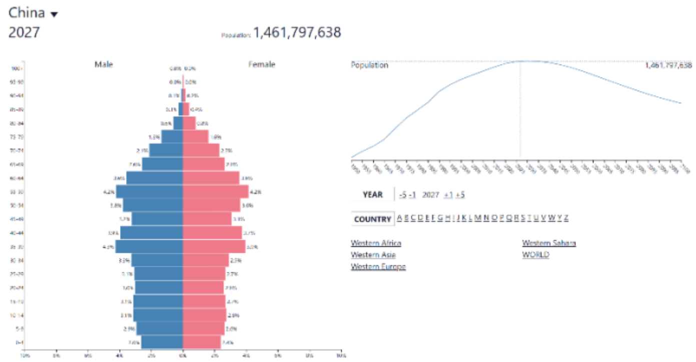

图 16 2027 年预测中国人口金字塔

我们可以发现中国目前应当处于稳定型，再观察该网站预测的 2027 年人口年龄金字塔，我们可以在此确认我们之前结论的正确性，所以如果想要进一步改善当前中国目前的人口，最好的办法是继续开放多胎政策同时鼓励一个家庭有多个孩子，但是生育成本是影响育龄家庭生育意愿的最重要因素之一。原国家卫计委在 2017 年进行的全国生育状况抽样调查结果显示，育龄妇女不打算再生育的前三位原因依次是“经济负担重”、“年龄太大”、“没人带孩子”，分别占 77.4%、45.6%和 33.2%。我们可以发现经济压力是使得生育率停滞不前的主要原因，因此出台减轻生育成本的政策也应在考虑范围之内，目前为止，部分地方已经率先出台了一些减轻家庭养育负担的政策：

例一、2021 年 7 月 28 日，四川省攀枝花市公布并详细解读攀枝花市《关于促进人力资源聚集的十六条政策措施》，其中，对按政策生育二、三孩的攀枝花户籍家庭，每月每孩发放 500 元育儿补贴金，直至孩子 3 岁。

例二、2021 年 9 月 15 日，甘肃省临泽县发布《临泽县优化生育政策促进人口长期均衡发展的实施意见（试行）》规定，二孩每年发放 5000 元育儿补贴，三孩每年发放 10000 元育儿补贴，直至孩子 3 岁。在辖区内公办幼儿园就读的临泽户籍常住家庭，二孩每生每学年给予 1000 元的资助，三孩每生每学年给予 2000 元的资助。对生育二孩、三孩的临泽户籍常住家庭，在城区购买商品房时给予 4 万元的政府补助。

不过，到目前为止，这些都是地方性的政策，而且力度远远不够，对于几十万到上百万的养育成本可以说是杯水车薪。只有中央政府才有实力给到实质性的养育负担的减轻。我们可以做一个简单的估算，中国的养育成本收入比是 6.9，而其他发达国家的养育成本收入比是 2-4，如果按照未来的出生率是 1%来计算的话，那就需要每年需要 3-5% 的 GDP 投入才能把养育成本比降到其他发达国家的水平。（具体计算是(6.9-2）/1%到(6.9-4）/1%）.这是几万亿的总投入，中央财政才有这样的财力。可惜的是，中央层面尚未出台发放育儿补贴的政策。我们建议，全国层面尽快出台减轻育龄家庭生育成本的政策。以下为减轻生育成本的部分措施：

一、现金和税收补贴

由于不同地区和人群之间存在很大的收入差距，我们建议个人所得税减免和现金补贴的方式并重，对高收入家庭通过孩子人头抵税的方式减免个人所得税。由于收入较低者不需要缴纳个人所得税，所以减免税收不适用于低收入家庭，对于这些家庭，可直接发放现金补贴。

二、购房补贴

现在制约育龄夫妇生育孩子的一个重要原因就是高房价，中国大城市的房价收入比是世界上最高的，虽然大城市的收入也高，但是房价更高。大城市生活成本高主要是因为房价高，其他如衣食住行，大城市并不比小城市贵很多，教育成本如果是公立教育， 也不会贵很多。所以大城市里，养育的高成本主要体现在房价上。这是大城市的生育率要低于小城市的重要原因之一。根据七普数据，2020 年全国总和生育率为 1.3，其中上海和北京的总和生育率分别仅有 0.74 和 0.87，而山东、河南、江西等省份的总和生育率在 1.4 左右。要减轻育儿家庭的负担，除了现金和税收补贴以外，还需要对多孩家庭买房被贴的政策。具体方式可以通过按揭利息返还或房价打折进行补贴。

三、增建各类幼托机构

根据世界银行数据，2019 年中国 15-64 岁女性劳动参与率达到 68.6%，而世界平均水平为 52.6%。由于中国的女性劳动参与率比较高，当今中国很多夫妻都是双职工。大量年轻人不敢生育二孩三孩的主要原因之一是，看护孩子的时间和精力成本高昂，这特别体现在孩子入托、入幼、入学的困难上。尤其是未满三岁孩子的托儿服务严重缺乏。原国家卫计委的数据显示，0-3 岁婴幼儿在我国各类托幼机构的入托率仅为 4%。我们建议把 0-3 岁孩子的入托率提高到 50%左右。要实现这一目标，政府有必要直接或者牵头。

参考文献

[1] 李玉琴. 我国公共教育资源配置对生育率的影响研究[D]. 贵州财经大学, 2022. DOI:10.27731/d.cnki.ggzcj.2022.000214.
[2] 何剑钢,郑宇西.中国死亡率改善率预测及实践研究[J].保险研究,2022(01):79-96.DOI:10.13497/j.cnki.is.2022.01.006.
[3] 李英嘉. 东北地区育龄人口生育意愿影响因素及其激励策略研究[D].辽宁大学,2022.DOI:10.27209/d.cnki.glniu.2022.000190.
[4] 银鱼.《中国活法Ⅱ》[J].新阅读,2018(03):54.
[5] 新峰.《国民经济统计概论》[J
[6] 王庆德.我国应对人口老龄化养老服务的模式研究[J].现代商贸工业,2022,43(18):82-83.DOI:10.19311/j.cnki.1672-3198.2022.18.036.
[7] 钟森丽. 中国城镇家庭的育儿时间成本[D].内蒙古大学,2022.

附　　录

附录 1：绘制人口数量变化图代码

```
import matplotlib.pyplot as plt
import numpy as np
import matplotlib.dates as mdates
import matplotlib as mpl
import pandas as pd
s = "数据矩阵"
a = s.split(' ')
city1 = []
city2 = []
city3 = []
for i in range(len(a)):
    a[i] = int(float(a[i]))
for i in range(len(a)):
    if i % 3 == 0:
        city1.append(a[i])
    elif i % 3 == 1:
        city2.append(a[i])
    else:
        city3.append(a[i])
x = np.arange(1,51,1)
plt.plot(x, city3)
plt.show()
```

附录 2：计算生育意愿参数代码

```
import numpy as np
import matplotlib.pyplot as plt

def softmax(z):

    new_z = np.zeros(z)

    total = 0

    for i in z:
        total += np.exp(i)

    for i in z:
        new_z[i] = np.exp(i) / total

    return new_z

def initialize(dim):

    alpha = np.zeros(shape = dim)
    alpha += np.random.normal(1, 0.3, (dim))
    return alpha

def propagate(X, alpha, P, num_iterations , learning_rate):

    y = X.shape[0]
    Pi = P[1]
    #正向传播
    W = alpha[0] + np.multiply(alpha[1],np.log(Pi[0])) +
np.multiply(alpha[2],np.log(Pi[1])) + np.multiply(alpha[3],np.log(Pi[2]))
    A = np.sum(np.dot(np.dot(P[0], W.T), X),axis=1)
    loss = (- 1 / (3*y)) * np.sum((X[0] - A)**2)

    #反向传播
    dalpha0 = 2/(3*len(X))*(X[0] - A)*Pi[0]*X*(W**2)*(1-W)
    dalpha1 = 2/(3*len(X))*(X[0] - A)*Pi[0]*X*(W**2)*(1-W)*np.log(Pi[0])
    dalpha2 = 2/(3*len(X))*(X[0] - A)*Pi[0]*X*(W**2)*(1-W)*np.log(Pi[1])
    dalpha3 = 2/(3*len(X))*(X[0] - A)*Pi[0]*X*(W**2)*(1-W)*np.log(Pi[2])

    #创建一个字典，把 dalpha 保存起来。
    grads = {
                "dalpha0": dalpha0,
```

```
                "dalpha1": dalpha1,
                "dalpha2": dalpha2,
                "dalpha3": dalpha3,
             }
    return (grads , loss)

def optimize(X_train, alpha, P, Pi, num_iterations , learning_rate):

    costs = []

    for i in range(num_iterations):

        grads, loss = propagate(X_train, alpha, Pi, num_iterations ,
learning_rate)

        dalpha0 = grads["dalpha0"]
        dalpha1 = grads["dalpha1"]
        dalpha2 = grads["dalpha2"]
        dalpha3 = grads["dalpha3"]

        alpha0 = alpha[0] - learning_rate * dalpha0
        alpha1 = alpha[1] - learning_rate * dalpha1
        alpha2 = alpha[2] - learning_rate * dalpha2
        alpha3 = alpha[3] - learning_rate * dalpha3

        #记录成本
        if i % 100 == 0:
            costs.append(loss)

    params  = {
                "alpha0" : alpha0,
                "alpha1" : alpha1,
                "alpha2" : alpha2,
                "alpha3" : alpha3, }
    return (params, costs)

def predict(alpha0, alpha1, alpha2, alpha3, X, Pi):
    W = alpha0 + np.multiply(alpha1,np.log(Pi[1][0])) +
np.multiply(alpha2,np.log(Pi[1][1])) + np.multiply(alpha3,np.log(Pi[1][2]))
    A = np.dot(np.sum(Pi[0],axis=0).reshape(3,80),W.T)
    A = np.sum(A, axis=-1)
    A = np.sum(np.dot(X, A),axis=1)
    return A
```

```
def model(X_train , Pi_train, Y_train , X_test , Pi_test, Y_test,
num_iterations = 2000 , learning_rate = 0.1 , print_cost = False):

    alpha = initialize((4, X_train.shape[1], 3))
    params, costs = optimize(X_train, alpha, Pi_train[0], Pi_train[1],
num_iterations , learning_rate)

    #从字典“参数”中检索参数 w 和 b
    alpha0, alpha1, alpha2, alpha3 = params["alpha0"] , params["alpha1"],
params["alpha2"], params["alpha3"]

    #预测测试/训练集的例子
    Y_prediction_test = predict(alpha0, alpha1, alpha2, alpha3, X_test,
Pi_test)
    Y_prediction_train = predict(alpha0, alpha1, alpha2, alpha3, X_train,
Pi_train)

    #打印训练后的准确性
    print("训练集准确性: "  , format(100 - np.mean(np.abs(Y_prediction_train -
Y_train)) * 100) ,"%")
    print("测试集准确性: "  , format(100 - np.mean(np.abs(Y_prediction_test -
Y_test)) * 100) ,"%")

    return alpha

X_train = np.zeros(shape=[180, 80, 3])
Y_train = np.zeros(shape=[180, 3])
X_test = np.zeros(shape=[20, 80, 3])
Y_test = np.zeros(shape=[20, 3])
Pi_train = np.zeros(shape=[2, 4, X_train.shape[1], X_train.shape[2]])
Pi_test = np.zeros(shape=[2, 4, X_test.shape[1], X_test.shape[2]])
#数据集来自于中经数据，为保障数据来源方的隐私，这里去掉了敏感数据。

d = model(X_train , Pi_train, Y_train , X_test , Pi_test, Y_test,
num_iterations = 2000, learning_rate = 0.1, print_cost = True)
```

附录 3：计算最佳基础设施安排代码

```
#include <stdio.h>
#include <stdlib.h>
#include <math.h>
void cal_short(void){
 for(int k1=0; k1 <= 6; k1++){
  float result = 0.7245*sqrt(k1) + 0.2039*sqrt(6-k1);
  printf("k1=%d、k2=%d 的时候, 结果为%f\n", k1, 6-k1, result);
 }
}
void cal_long(void){
 float max=0;
 int k1_m, k2_m, k3_m;
 for(int k1=0; k1 <= 6; k1++){
  for(int k2=0; k2 <= 6-k1; k2++){
   double result = pow(0.95, -0.7245*sqrt(k1) -
0.2039*sqrt(k2))*pow(0.2*log(20)/pow(6-k1-k2+1,0.25), - 0.2039);
   if(result > max){
    max = result;
    k1_m = k1;
    k2_m = k2;
    k3_m = 6 - k1 - k2;
   }
   printf("k1=%d、k2=%d、k3=%d 的时候, 结果为%lf\n", k1, k2, 6-k1-k2, result);
  }
 }
 printf("当 k1=%d,k2=%d,k3=%d 时，生育意愿取到最大值",k1_m,k2_m,k3_m);
}

int main(){
 cal_short();
}
```

附录 4：推演人口数量变化代码

```
#include <stdio.h>
int main(){
FILE* p=fopen("新建文本文档.txt","r");

double People[90][3]={0};//起始人口
double tmpPeople[90][3]={0};
double SYL[90][3]={0};//各个年龄段生育率
double SWL[90][3]={0};
double Qianyi[90][6]={0};
double a;
for(int n=0;n<3;n++){
 for(int i=0;i<90;i++){
 fscanf(p,"%lf",&People[i][n]);
 }
}
for(int n=0;n<3;n++){
for(int i=0;i<90;i++){
 fscanf(p,"%lf",&SYL[i][n]);
}
}
for(int n=0;n<3;n++){
for(int i=0;i<90;i++){
 fscanf(p,"%lf",&SWL[i][n]);
}
}
for(int n=0;n<6;n++){
for(int i=0;i<90;i++){
 fscanf(p,"%lf",&Qianyi[i][n]);
}
}
for(int n=0;n<3;n++){
for(int i=0;i<90;i++){

 SYL[i][n]/=1000.0;
 SWL[i][n]/=1000.0;


}
}
for(int n=0;n<6;n++){
for(int i=0;i<90;i++){
 Qianyi[i][n]/=100.0;
}
```

```
}
//各个年龄段死亡率
 for(int i=0;i<50;i++){//50 年
  for(int n=0;n<3;n++){
   double T=0;
   for(int m=0;m<90;m++){
   T+=People[m][n]*SYL[m][n];
  }
   tmpPeople[0][n]=T;
   for(int m=0;m<89;m++){
   tmpPeople[m+1][n]=People[m][n]*(1-SWL[m][n]);
  }
  for(int m=0;m<90;m++){
   People[m][n]=tmpPeople[m][n];
  }

  }
  for(int n=0;n<3;n++){
   for(int m=0;m<90;m++){
   if(n==0){

    People[m][n]=People[m][n]+tmpPeople[m][2]*Qianyi[m][2]+tmpPeople[m][1]*Qia
nyi[m][3];
    People[m][n]=People[m][n]-tmpPeople[m][0]*Qianyi[m][0]-
tmpPeople[m][0]*Qianyi[m][5];
    }
   else if(n==1){
    People[m][n]=People[m][n]+tmpPeople[m][2]*Qianyi[m][4]+tmpPeople[m][0]*Qia
nyi[m][0];
    People[m][n]=People[m][n]-tmpPeople[m][1]*Qianyi[m][1]-
tmpPeople[m][1]*Qianyi[m][3];
    }
   else if(n==2){
    People[m][n]=People[m][n]+tmpPeople[m][0]*Qianyi[m][5]+tmpPeople[m][1]*Qia
nyi[m][1];
    People[m][n]=People[m][n]-tmpPeople[m][2]*Qianyi[m][2]-
tmpPeople[m][2]*Qianyi[m][4];
    }

  }

  }
  printf("Year:%d\n",i);
 for(int n=0;n<3;n++){
```

```
printf("City:%d\n",n);
double TotalP=0.0;
for(int i=0;i<90;i++){

 TotalP+=People[i][n];
}
printf("%lf \n",TotalP);

}

}


 //printf("\n");



return 0;}
```