报名序号： 000098

赛题题目：大学生平衡膳食食谱的优化设计及评价

# 大学生平衡膳食食谱的优化设计及评价


## 摘要

大学生平衡膳食食谱的优化设计及评价，旨在促进其健康成长，提高生活质量，并培养终身受益的健康饮食习惯 。本文主要研究大学生平衡膳食食谱的优化设计及评价，通过DBI_16指数、AAS等指标建立膳食食谱营养AHP-TOPSIS评价模型，利用TS-GA、TS-NSGA-Ⅱ智能优化算法分别对单、多目标优化问题高效求解。

针对问题一：可以将其定性为评价类问题。首先以本题附件4中“膳食食食谱分析评价”为依据，选取出能够评价大学生平衡膳食食谱的7个有效指标，如必需氨基酸评分（AAS）等，构建了膳食食谱营养AHP-TOPSIS评价模型。接着分别求出男女大学生的食谱评分为0.90和0.69。以参考的摄入量对给定食谱进行分析，发现男、女大学生存在不同问题，根据上述不足，人工对食谱进行改进，求解出男、女大学生食谱评分为：0.91和0.72，膳食食谱得一定的优化。

针对问题二：可以将其定性为优化类问题。分别对第1、2小问建立单目标优化模型、对第3小问建立多目标优化模型。第1小问：以食物的份数为决策变量，以蛋白质氨基酸评分最大为目标函数，约束条件为最经济的餐费用等，根据整数规划的思想建立单目标规划模型。（对于2、3小问，只需调整约束条件和目标函数即可。）为了解决遗传算法求解过程中收敛速度缓慢等问题，我们利用TS-GA、TS-NSGA-Ⅱ智能优化算法分别对上述问题高效求解，得出对应优化后的男、女大学生日食谱。然后通过AHP-TOPSIS评价模型进行膳食营养评价。最后对1、2、3小题日食谱进行**波动性**等比较分析。

针对问题三： 可以将其定性为多目标优化类问题。是在问题二基础上的以以蛋白质氨基酸评分最大、用餐费用最经济、蛋白质氨基酸评分及经济性为目标函数的多目标优化问题，通过对同类日食谱食物品种数约束条件，保证了食物品种的丰富性。最后利用TS-NSGA-Ⅱ智能优化算法求解出男、女大学生周平衡膳食食谱的优化设计方案。（见表13-20）

针对问题四：结合问题一、二、三的研究，撰写健康饮食倡议书。我们严格按照倡议书格式，针对大学生的饮食结构和习惯，提出改进建议，主题为**健康饮食、平衡膳食**。



关键词：AHP-TOPSIS评价系统、TS-NSGA-Ⅱ、TS-GA、整数规划，多目标优化

# 1. 问题重述

## 1.1. 问题背景

大学时代是学知识、长身体的重要阶段，同时也是良好饮食习惯形成的关键时期。这一特定年龄段的年轻人，身体需要充足的营养来支持生长发育，同时，高强度的学习和运动也需要大量能量的支持。

然而，大学生中饮食结构不合理以及不良的饮食习惯问题比较突出，快节奏的大学生活使得许多学生忽视了饮食的重要性，往往选择快速但不健康的饮食方式，缺乏对均衡膳食的重视。主要表现在不规律或不充足的早餐摄入、频繁选择外卖和快餐食品，以及个别学生通过控制饮食来减少脂肪摄入而导致营养不良等情况。大学阶段学习并掌握营养知识，培养健康的饮食习惯，对于促进生长发育、保证身体健康具有至关重要的意义。良好的营养摄入不仅有助于学生的身体健康，还能提高其学习效率和整体生活质量。

## 1.2. 已知信息

1.附件一和附件二分别列出了一名男大学生和一名女大学生的一日食谱，包括每餐的食物名称、主要成分、食物编码、可食部重量以及食用份数。

2.附件 3 提供了某高校学生食堂一日三餐的食物种类、主要成分、编码、分量、价格以及是否可点半份的详细信息统计表。

3.附件 4 提供了膳食食谱分析评价和优化设计的综合指南，涵盖了平衡膳食的基本准则、大学生每日能量摄入的具体目标、营养素的定量要求、食谱营养评价的详细过程以及平衡膳食食谱的优化设计原则。

## 1.3. 要解决的问题

**问题一：**构建一种合理的评价方法，并对食谱进行科学的营养分析评价。再基于学生食堂提供的食物信息调整并重新评价。

**问题二：**分别建立不同的优化模型，根据结果对男女大学生的食谱进行重新设计与再次评价并且比较以上三种食谱的营养价值和成本。

**问题三：**将问题二中优化模型的时间尺度进一步的扩大，建立周为时间尺度的优化模型，并设计男、女大学生的周食谱，并进行评价及比较分析。

**问题四：**结合已做的问题一、二、三，针对大学生的饮食结构和习惯，确定主题，

根据倡议书的语言风格、行文脉络等要求，撰写健康饮食倡议书。

# 2. 问题的假设

## 2.1. 问题的假设

1.假设附件 1 和附件 2 中记录的食物摄入量准确无误。

2.所有大学生均无食物过敏或特殊饮食限制，因此可以自由摄入各类食物。

3.假设附件3中提供的食物信息统计表准确反映了高校食堂提供的食物种类和营养成分。

4.假设高校食堂提供的食物种类、食物价格和营养成分在研究期间保持不变。

# 3. 问题的分析

## 3.1. 问题 1 的分析

首先，从《中国食物成分表》第一册和第二册，精确查找每 100 克可食部分食物的主要营养素含量，进而计算出食谱中各种主要营养素的含量，再参照附件 4 提供的平衡膳食基本准则、相关定量要求和营养评价过程，通过研究相关文献，引入平衡膳食的权威性指数，选取合适的评价指标建立多级指标评价模型，确保评价模型的科学性和权威性，然后对两份食谱做出全面的膳食营养评价。根据评价结果，对食谱中的不足之处进行人工微调和优化，并对更新后的食谱进行再次的全面营养评价。

## 3.2. 问题 2 的分析

首先，我们对于每个小题，需要根据问题的特性和需求，确定目标函数、约束条件以及决策变量，分别建立相应的数学优化模型。接着，我们根据复杂性、求解时间等要求，选择合适的智能优化算法来求解这些模型。在求解过程中，我们可能会发现基本的智能优化算法在某些情况下表现不佳。我们可以对算法进行改进，如调整参数、引入新的启发式规则或者结合多种算法的优点来增强性能。然后，我们将基于问题 1 所得到的评价模型，对优化后各小题的日食谱进行膳食营养评价。最后对 1、2、3 小题日食谱进行比较分析，得出结果。

### 3.3. 问题 3 的分析

基于问题 2 多目标优化的模型，根据问题三的 题目要求新增加新的目标函数，新增约束食物种类经可能丰富。累加七天的日平衡膳食食谱形成周平衡膳食食谱。最后利用问题 2 智能优化算法求解出男、女大学生周平衡膳食食谱的优化设计方案。

### 3.4. 问题 4 的分析

要求分析大学生的饮食结构和习惯，并提出健康饮食和平衡膳食的建议。重点在于理解大学生饮食现状，并给出具体可行的改善方案。倡议书核心内容应包括分析大学生饮食现状，提出健康饮食建议，强调营养均衡、规律用餐、饮食多样化，并倡导合理锻炼和保持良好心态。

## 4. 模型的建立与求解

### 4.1. 问题 1 的模型建立和求解

#### 4.1.1. 膳食食谱营养评价过程

膳食食谱的营养评价是为了全面了解日常饮食提供的营养情况，以指导饮食结构的合理安排。通过分析食物结构、计算主要营养素含量、评价能量来源、蛋白质氨基酸评分等步骤，可以得出食谱的营养特点及不足之处。这种评价不仅能帮助个人合理搭配饮食，也是食堂或餐饮服务提供者改进食谱的重要依据。

**1.食谱的主要营养素含量及其它膳食参数计算**

首先， 我们参考《中国食物成分表》第一册、第二册,从中查出每 100 克可食部食物所含主要营养素的数量，列表如下：

表 1 100g 食物主要营养素含量表

| 序号 | 食物名称 | 主要成分 | | 异亮氨酸 mg/g 蛋白质 | 亮氨酸 mg/g 蛋白质 | 赖氨酸 mg/g 蛋白质 | 含硫氨基酸 mg/g 蛋白质 | |
|---|---|---|---|---|---|---|---|---|
| 1 | 牛奶 | 牛奶 | | 1.46 | 2.91 | 2.3 | 0.92 | |
| 2 | 酸奶 | 酸奶 | | 1.42 | 2.59 | 2.08 | 0.26 | |
| 3 | 豆浆 | 黄豆 | | 18.53 | 28.19 | 22.37 | 9.02 | |
| 4 | 大米粥 | 稻米 | | 3.19 | 6.11 | 2.6 | 3.32 | |
| 5 | 小米粥 | 小米 | | 3.92 | 11.66 | 1.76 | 5.12 | |
| | …… | …… | | …… | …… | …… | …… | |

① 对于每种食物 i，其营养素 j 的含量可以表示为

$$K_{ij} = M_{ij} \times N_i$$

其中，$K_{ij}$ 表示食物 $i$ 中营养素 $j$ 的含量；$M_{\{ij\}}$ 表示食物 i 每 100 克中营养素 j 的含量；$N_i$ 表示食物 i 的可食部重量（克）。

② 一日三餐的总营养素含量为所有食物营养素含量的总和

$$X_j = \sum_{\{i=1\}}^{\{n\}} X_{\{ij\}}$$

其中，$X_j$ 表示一天中摄入的总营养素 j 的含量；n 表示一天中食物的总数。

③ 每种微量元素的摄入量计算公式为

微量元素摄入量（$mg$或μ$g$）=∑每种食物的微量元素含量（$mg$）

微量元素摄入量（mg 或μg）=∑每种食物的微量元素含量（mg 或μg）

**2.膳食结构**

参考《中国居民膳食指南（2016）》中的平衡膳食基本准则，根据制作的数据表格，对两位大学生的食物种类进行分类和统计。衡量他们的膳食种类是否符合准则要求，包括食物种类的多样性、食物分组的合理性以及每日膳食的均衡性。

**3.能量及餐次比评估**

之后对食谱数据进行能量及餐次比的计算。具体步骤包括计算每日总能量摄入和各餐次的能量分配比例，并将结果与《中国居民膳食指南（2016）》中的推荐值进行比较，以评估能量摄入的合理性和餐次分配的科学性。

**4.食谱 ASS 评价**

① 氨基酸评分计算公式

对于每种氨基酸，其评分公式为：

$$氨基酸得分 = \frac{标准氨基酸模式中该氨基酸含量（mg/g蛋白质）}{食物中该氨基酸含量（mg/g蛋白质）} \times 100$$

② 最低氨基酸评分（AAS）

每种食物的蛋白质氨基酸评分为所有氨基酸得分中的最小值，即：

$$\mathrm{AAS} = \min(\text{氨基酸得分})$$

③ 混合食物的蛋白质氨基酸评分

当有多种食物混合时，总氨基酸含量为各食物中氨基酸含量的加权平均，具体公式为：

$$混合食物的氨基酸评分 = \frac{\sum(食物主要成分的AAS \times 该主要成分的可食部重量（g）)}{混合食物总可食部重量（g）}$$

根据以上公式，混合食物的蛋白质氨基酸评分为所有氨基酸得分中的最小值。

### 4.1.2. AHP-TOPSIS 系统的建立

1.TOPSIS 综合评价模型：

本文采用 TOPSIS 综合评价模型，结合推荐的每日营养素摄入量，对两份食谱进行综合评价。TOPSIS 模型将考虑食谱中各营养素含量、非营养素指标（如膳食纤维、抗氧化物质等）以及 DBI_16、AAS 综合这些因素来评估食谱的整体营养质量和健康性。

TOPSIS 是一种常用的多属性决策方法，它通过计算方案与理想解（最优解）和反理想解（最劣解）的距离来评价和排序方案。

① 构建决策矩阵：

设有 m 个评价对象（例如不同的饮食方案），n 个评价指标（如蛋白质含量、维生素含量等），构建决策矩阵 X，其元素$y_{ij}$表示第 i 个对象在第 j 个指标上的表现。

② 标准化决策矩阵：

使用向量归一化方法来消除不同指标之间的量纲影响。通常采用线性标准化方法：

$$r_{ij} = \frac{x_j}{\sqrt{\sum_{i=1}^{m} x_{ij}^2}}$$

③ 构建加权标准化决策矩阵：

根据每个指标的重要性，给出权重向量 $\mathrm{W} = (\mathrm{w}_1, \mathrm{w}_2, \ldots, \mathrm{w}_\mathrm{n})$ 其中$\sum_{j=1}^{n} w_j = 1$。然后计算加权标准化决策矩阵：$v_{ij} = w_j \cdot r_{ij}$

④ 确定理想解和反理想解：

理想解$A^+$是指标的最佳可能值，反理想解$A^-$是指标的最劣可能值：

$$A^+ = (\underset{i}{max} v_{ij} | j \in \text{benefit criteria}), (\underset{i}{min} v_{ij} | j \in \text{cost criteria})$$

$$A^- = (\underset{i}{min} v_{ij} | j \in \text{benefit criteria}), (\underset{i}{max} v_{ij} | j \in \text{cost criteria})$$

⑤ 计算每个方案与理想解和反理想解的距离：

使用欧几里得距离计算方法：

$$S_i^+ = \sqrt{\sum_{j=1}^{n} (v_{ij} - A_j^+)^2}$$

$$S_i^- = \sqrt{\sum_{j=1}^{n} (v_{ij} - A_j^-)^2}$$

⑥ 计算相对接近度：

相对接近度表示方案与理想解的相对接近程度，计算公式为：

$$C_i = \frac{S_i^-}{S_i^+ + S_i^-}$$

其中，$C_i$的值越接近 1，表示方案越优。

⑦ 排名：

根据 C_i 的值对所有方案进行排序，选择相对接近度最高的方案作为最优方案。

TOPSIS 模型因其直观性和计算简单性而被广泛应用于多种领域，包括资源管理、健康评估和工程设计等。

通过构建 TOPSIS 模型对两份食谱膳食进行整体评价并使用层析分析法对每个指标进行赋权，其中 TOPSIS 的指数组成如下：

表 2 TOPSIS 指数表

| 一级指标 | 二级指标 | | 三级指标 | |
|---|---|---|---|---|
| 日膳食食谱营养 AHP-TOPSIS 评价 | 食物结构 | 0.2 | 食物种类 | 0.1 |
| | | | 食物类别 (DBI_16 指数) | 0.1 |
| | 主要营养素摄入量 | 0.2 | 能量摄入目标 | 0.1 |
| | | | 非产能营养素摄入目标 | 0.1 |
| | 日餐次能量分配 | 0.2 | 餐次比 | 0.2 |
| | 能量来源 | 0.2 | 宏量营养素供能占比 | 0.2 |
| | 氨基酸优质程度 | 0.2 | 必需氨基酸评分 (AAS) | 0.2 |

**2.膳食平衡指数（DBI）**

DBI 是一种综合性指标，用于评估个体或群体的膳食平衡状况。它不仅考虑了能量、蛋白质、脂肪、碳水化合物等宏量营养素的摄入量是否符合健康饮食指南，还包括维生素、矿物质等微量营养素的摄入情况。DBI 的设计旨在确保饮食的多样性和均衡性，从而帮助评估和改善饮食的合理性和健康水平。

其中 DBI 的计算公式可以有很多种，一个常见的简化公式是：

$$DBI\_16 = \frac{1}{N}\sum_{i=1}^{N}\left(\frac{I_i}{RDA_i}\right)^{w}$$

其中，N 是评估的营养素种类数，$I_i$是第$i$种营养素的实际摄入量，$RDA_i$是第$i$种营养素的推荐日摄入量 w 是权重系数，用于调整不同营养素对 DBI 的贡献度，可以根据研究目的进行调整

**3.大学生食谱调整**

① 男大学生营养素评估及调整：

减少热量和脂肪：减少油炸食品和高脂肪食物的摄入。减少 1 份炸鸡块（约 200 kcal, 15 g 脂肪）可以显著降低热量和脂肪摄入。

② 女大学生营养素评估及调整：

增加热量和碳水化合物：增加高碳水化合物食物的摄入。增加 2 份大米饭（每份 150 kcal，30 g 碳水化合物），再增加 1 份红薯（100 g，约 90 kcal，20 g 碳水化合物），可以显著增加热量和碳水化合物摄入。

对食谱进行针对的改进，并对改进后的食谱进行有关评价指数的计算，得出结果如下：

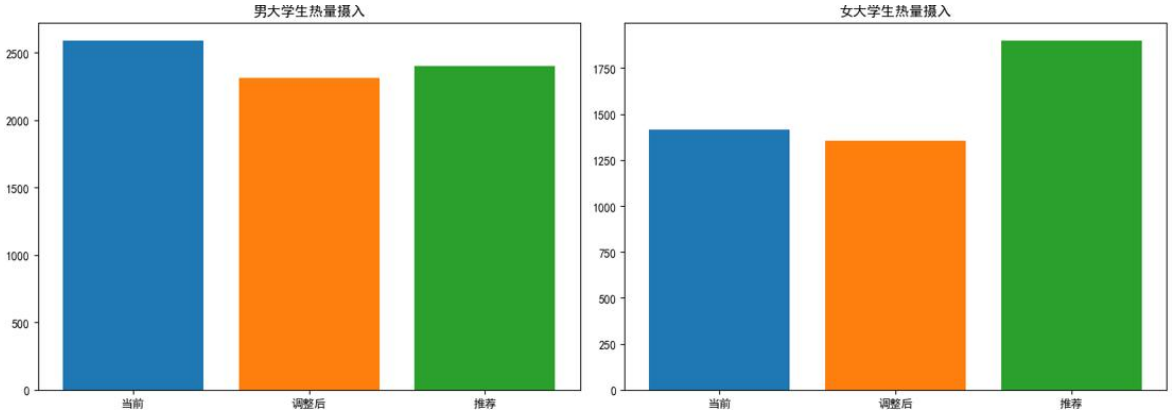



图 1 男女大学生热量摄入图

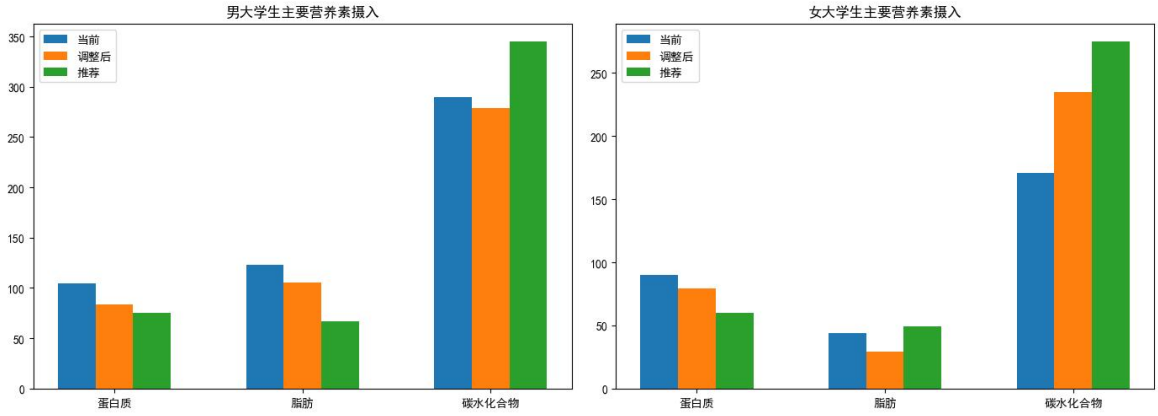



图 2 男女大学生主要营养素摄入图

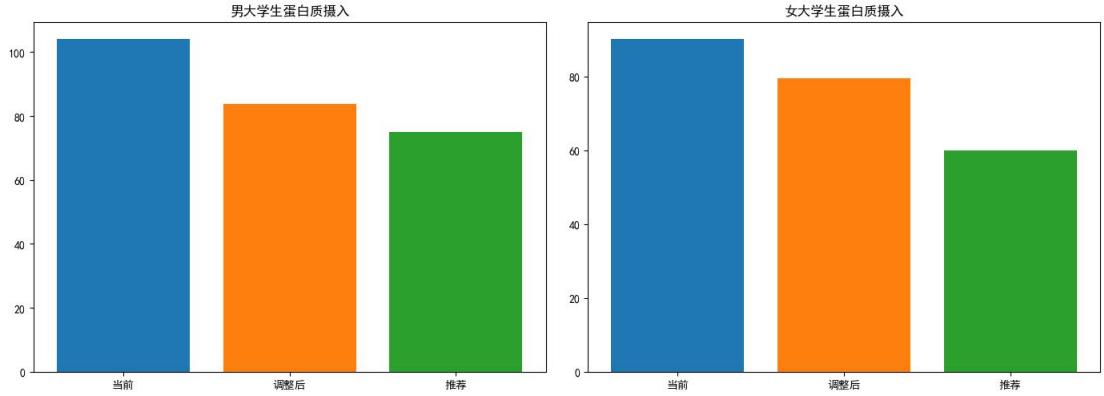


图 3 男女大学生蛋白质摄入图

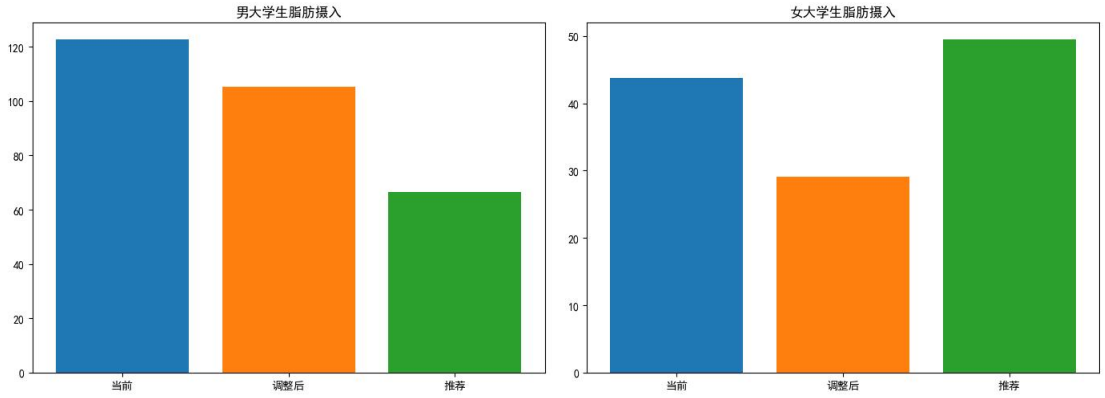



图 4 男女大学生脂肪摄入

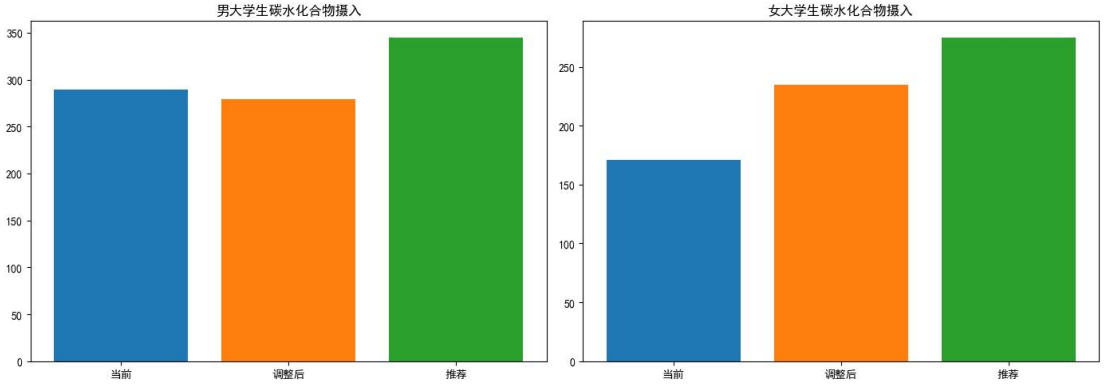



图 5 男女大学生碳水化合物摄入

对食谱进行针对的改进，并对改进后的食谱进行有关评价指数的计算，得出结果如下：

表 3 膳食评分表

| 食谱种类 | 膳食评分 | 改良食谱后膳食评分 |
|---|---|---|
| 男大食谱 | 0.90 | 0.91 |
| 女大食谱 | 0.69 | 0.72 |

通过评价指数的变化可以发现：适当调整男、女大学生的饮食结构，可以显著改善二者的营养素摄入，更接近推荐值，同时总体膳食指数更高，这些调整有助于实现附件 4 要求的均衡膳食，促进健康饮食。

### 4.2. 问题 2 的模型建立与求解

问题 2 是一个线性规划问题。我们需要基于附件 3：某高校学生食堂一日三餐主要食物信息统计表，设计出分别适合男大学生和女大学生的日平衡膳食食谱，目标是餐费用最经济、最大化蛋白质氨基酸评分和兼顾蛋白质氨基酸评分及经济性。在满足各项营养素约束条件的同时，找到最优的菜品搭配方案。餐费用最经济和最大化蛋白质氨基酸评分是单目标优化问题，而兼顾蛋白质氨基酸评分及经济性则是多目标优化问题，需要在最大化蛋白质氨基酸评分和最小化膳食总费用之间找到平衡。

根据附件 4 中的平衡膳食食谱优化设计原则，男、女生每日能量实际摄入量与目标摄入量相差在±10%之内；产能营养素占总能量的百分比尽量满足蛋白质 10%-15%、脂肪 20%-30%、碳水化合物 50%-65%；非产能主要营养素钙、铁、锌、维生素 A、维生素 B1、维生素 B2、维生素 C 的实际摄入量尽可能接近参考摄入量；餐次比尽可能满足早餐 25%-35%，中餐和晚餐各 30%-40%；所有计算结果需四舍五入保留到小数点后两位。

#### 4.2.1. 建立以蛋白质氨基酸评分最大的单目标优化模型

决策变量：

$$X_{ij}：第\,j\,种，第\,i\,餐食物的份数$$

其中，i 是食物种类索引 ，$j = 1, 2, \ldots$。

目标函数：

$$\max \sum_{j=1}^{N} a\,as_j x_{ij}$$

其中，$aas_j$ 是第 j 种食物的蛋白质氨基酸评分。

约束条件：

$$0.9 \times E_{target} \leq \sum_{i,j} \left(x_{i,j} \cdot e_i\right) \leq 1.1 \times E_{target}$$

其中，$E_{target}$为每日的能量摄入目标（女生 1900kcal/d，男生 2400kcal/d）。

$$0.25 \leq \frac{\sum_i \left(x_{i,1} \cdot 能量_i\right)}{E_{\text{total}}} \leq 0.35$$

$$0.30 \leq \frac{\sum_i \left(x_{i,2} \cdot 能量_i\right)}{E_{\text{total}}} \leq 0.40$$

$$0.30 \leq \frac{\sum_i \left(x_{i,3} \cdot 能量_i\right)}{E_{\text{total}}} \leq 0.40$$

其中，三餐能量分配占总能量的百分比，即餐次比；早餐 25%-35%，中餐、晚餐各 30%-40%。

$$0.9 \times 800 \leq \sum_{i,j} \left(x_{i,j} \cdot 钙_i\right) \leq 1.1 \times 800$$

$$0.9 \times 参考铁 \leq \sum_{i,j} \left(x_{i,j} \cdot 铁_i\right) \leq 1.1 \times 参考铁$$

$$0.9 \times 参考锌 \leq \sum_{i,j} \left(x_{i,j} \cdot 锌_i\right) \leq 1.1 \times 参考锌$$

$$0.9 \times 参考维生素\ \mathrm{A} \leq \sum_{i,j} \left(x_{i,j} \cdot 维生素\mathrm{A}_i\right) \leq 1.1 \times 参考维生素\ \mathrm{A}$$

$$0.9 \times 参考维生素\ \mathrm{B1} \leq \sum_{i,j} \left(x_{i,j} \cdot 维生素\ \mathrm{B1}_i\right) \leq 1.1 \times 参考维生素\ \mathrm{B1}$$

$$0.9 \times 参考维生素\ \mathrm{B2} \leq \sum_{i,j} \left(x_{i,j} \cdot 维生素\ \mathrm{B2}_i\right) \leq 1.1 \times 参考维生素\ \mathrm{B2}$$

其中，非产能主要营养素含量尽可能接近参考摄入量，男女大学生的日膳食非产能营养素摄入如下表：

表 4　大学生每日膳食非产能主要营养素参考摄入量

| 营养素 | 钙（$mg \cdot d^{-1}$） | 铁（$mg \cdot d^{-1}$） | 锌（$mg \cdot d^{-1}$） | 维生素 A（$\mu g \cdot d^{-1}$） | 维生素 $B_1$/硫胺素（$mg \cdot d^{-1}$） | 维生素 $B_2$/核黄素（$mg \cdot d^{-1}$） | 维生素 C（$mg \cdot d^{-1}$） |
|---|---|---|---|---|---|---|---|
| 男生 | 800 | 12 | 12.5 | 800 | 1.4 | 1.4 | 100 |
| 女生 | 800 | 20 | 7.5 | 700 | 1.2 | 1.2 | 100 |

$$0.10 \leq \frac{\sum_{i,j} \left(x_{i,j} \cdot 蛋白质_i\right) \cdot 4}{E_{\text{total}}} \leq 0.15$$

$$0.20 \leq \frac{\sum_{i,j} \left(x_{i,j} \cdot 脂肪_i\right) \cdot 9}{E_{\text{total}}} \leq 0.30$$

$$0.50 \leq \frac{\sum_{i,j} \left(x_{i,j} \cdot 碳水化合物_i\right) \cdot 4}{E_{\text{total}}} \leq 0.65$$

其中，宏量营养素供能占总能量的百分比：蛋白质 10%-15%、脂肪 20%-30%、碳水化合物 50%-65%。

### 4.2.2. 建立以餐费用最经济的单目标优化模型

目标函数：

$$\text{Min } f = \sum_{i,j} \left(x_{i,j} \cdot c_{i,j}\right)$$

其中，$c_{i,j}$表示第 j 种，第 i 餐食物的费用（单位：元/克）。

决策变量和约束条件不变。

### 4.2.3. TS-GA 求解单目标优化模型

TS-GA 禁忌搜索遗传算法模型是一种结合了禁忌搜索和遗传算法两种优化技术的混合算法模型。该模型通过结合禁忌搜索的局部搜索能力和遗传算法的全局搜索能力，提高了复杂优化问题求解的效率和效果。

**1. 禁忌搜索**

禁忌搜索是一种局部搜索算法，通过引入禁忌表避免搜索陷入局部最优。基本步骤如下：

① 初始解：选择一个初始解。

② 邻域搜索：在当前解的邻域中搜索最优解。

③ 更新禁忌表：将新解加入禁忌表，以避免重复搜索。

④ 步长调整：根据搜索过程中的表现调整步长，以增强搜索能力。

⑤ 终止条件：达到最大迭代次数或满足其他终止条件。

公式如下：

$$x_{k+1} = \text{argmax}_{x \in N(x_k)} f(x) \quad \text{subject to} \quad x \notin \text{Tabu List}$$

其中，$N(x_k)$表示当前解$x_k$的邻域，Tabu List 表示禁忌表。

**2. 禁忌搜索遗传算法模型**

① 种群初始化：随机生成初始种群，对种群中的每个个体进行禁忌搜索，以提升初始种群的质量。

② 进化过程：适应度评价，计算每个个体的适应度值；选择，根据适应度值选择个体进行繁殖；交叉，通过交叉操作生成新的个体；变异，对部分个体进行变异操作；禁忌搜索：对新生成的个体进行禁忌搜索，进一步优化个体。

③ 更新种群：将经过禁忌搜索优化后的个体组成新一代种群。

④ 终止条件：达到预定的迭代次数或满足其他终止条件。

公式如下：

$$P_{t+1} = \text{Tabu Search}\big(\text{Genetic Operators}(P_t)\big)$$

其中，$P_t$表示第$t$代种群。

禁忌搜索遗传算法模型（TS-GA）结合了禁忌搜索和遗传算法的优势，在快速迭代和全局搜索方面具有显著的优点。禁忌搜索通过在解空间内进行局部探索，并使用禁忌表避免陷入局部最优解，增强了局部搜索效率。在每次迭代中，禁忌搜索能够迅速找到当前解的局部最优解，并通过禁忌表记忆已探索过的解，防止回溯到以前的解，使得搜索过程更加高效和集中。

遗传算法以其全局搜索能力而著称，通过选择、交叉和变异等操作，能够在解空间内进行广泛的探索，找到全局最优解的概率较高。遗传算法通过模拟自然选择过程，不断筛选和生成更优的个体（解），提高了全局搜索能力。结合禁忌搜索的局部优化，遗传算法的初始解质量更高，加速了整体收敛速度。

TS-GA 模型在快速迭代方面的优点包括加速收敛、跳出局部最优和平衡探索与开发。禁忌搜索在初期进行快速的局部优化，能够迅速找到高质量的初始解，而遗传算法利用这些高质量的初始解进行全局搜索，加速了整体收敛速度。禁忌搜索通过禁忌表有效避免了重复搜索，减少了陷入局部最优解的风险，结合遗传算法的多样化操作，TS-GA 更容易跳出局部最优，寻找全局最优解。

## 3. 单目标优化模型的求解以及结果分析

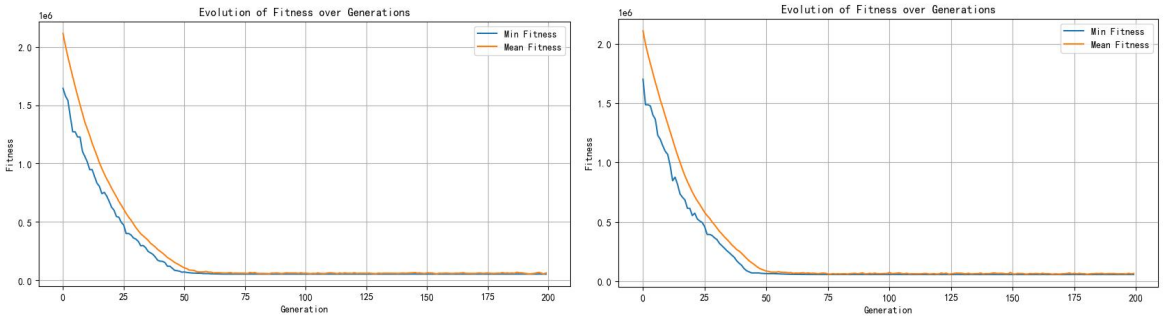



图 6 男大学生两种单目标模型的求解迭代曲线

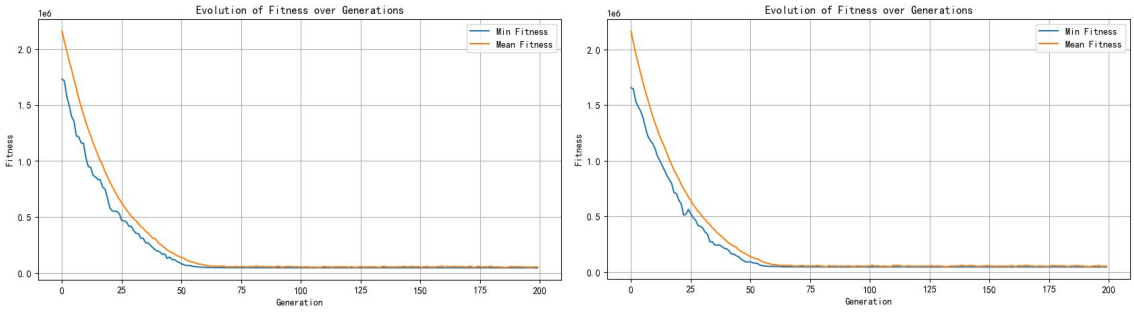



图 7 女大学生两种单目标模型的求解迭代曲线

表 5 男大学生蛋白质氨基酸评分最大推荐食谱

| 食物名称 | 食用份数 | 用餐时间 |
| --- | --- | --- |
| 牛奶 | 2 | 早餐 |
| 鸡排面 | 1 | 早餐 |

| | | |
|---|---|---|
| 拌木耳 | 0.5 | 早餐 |
| 葡萄 | 1 | 早餐 |
| 花卷 | 1 | 午餐 |
| 土豆丝饼 | 1 | 午餐 |
| 菠菜汤 | 1 | 午餐 |
| 鸡肉炖土豆胡萝卜 | 1.5 | 午餐 |
| 干炸黄花鱼 | 0.5 | 午餐 |
| 蜜瓜 | 1 | 午餐 |
| 砂锅面 | 1 | 晚餐 |
| 明太鱼炖豆腐 | 0.5 | 晚餐 |
| 炒肉杏鲍菇 | 0.5 | 晚餐 |
| 炒牛肉 | 0.5 | 晚餐 |
| 西瓜 | 1 | 晚餐 |
| 柚子 | 1 | 晚餐 |

表 6 女大学生蛋白质氨基酸评分最大推荐食谱

| 食物名称 | 食用份数 | 用餐时间 |
|---|---|---|
| 牛奶 | 2 | 早餐 |
| 鸡排面 | 1 | 早餐 |
| 拌木耳 | 0.5 | 早餐 |
| 葡萄 | 1 | 早餐 |
| 花卷 | 1 | 午餐 |
| 菠菜汤 | 0.5 | 午餐 |
| 拌芹菜花生米 | 0.5 | 午餐 |
| 鸡肉炖土豆胡萝卜 | 1 | 午餐 |
| 香蕉 | 1 | 午餐 |
| 蜜瓜 | 1 | 午餐 |
| 小米粥 | 1 | 晚餐 |
| 明太鱼炖豆腐 | 0.5 | 晚餐 |
| 炒牛肉 | 0.5 | 晚餐 |
| 柚子 | 1 | 晚餐 |

表 7 男大学生用餐费用最经济推荐食谱

| 食物名称 | 食用份数 | 用餐时间 |
|---|---|---|
| 牛奶 | 1 | 早餐 |
| 蜜瓜 | 4 | 午餐 |
| 炖海带白菜豆腐 | 4 | 晚餐 |
| 柚子 | 1 | 晚餐 |

表 8 女大学生用餐费用最经济推荐食谱

| 食物名称 | 食用份数 | 用餐时间 |
|---|---|---|
| 牛奶 | 1 | 早餐 |

| | | |
|---|---|---|
| 蜜瓜 | 3 | 午餐 |
| 韭菜盒子 | 1 | 晚餐 |
| 炖海带白菜豆腐 | 2 | 晚餐 |
| 柚子 | 2 | 晚餐 |

表 9 日单目标膳食评分表

| | 蛋白质氨基酸评分最大 | 用餐费用最经济 |
|---|---|---|
| 男日单目标膳食食谱 | 0.71 | 0.69 |
| 女日单目标膳食食谱 | 0.78 | 0.70 |

针对问题二，建立单目标优化模型进行求解，经过迭代后得出食谱设计结果，下面分别对蛋白质、餐费为目标设计食谱的结果进行评价：

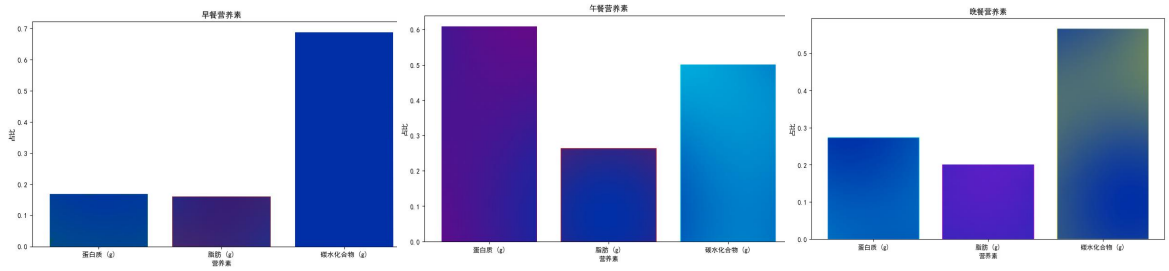



图 8 女生蛋白质目标食谱三餐评价图

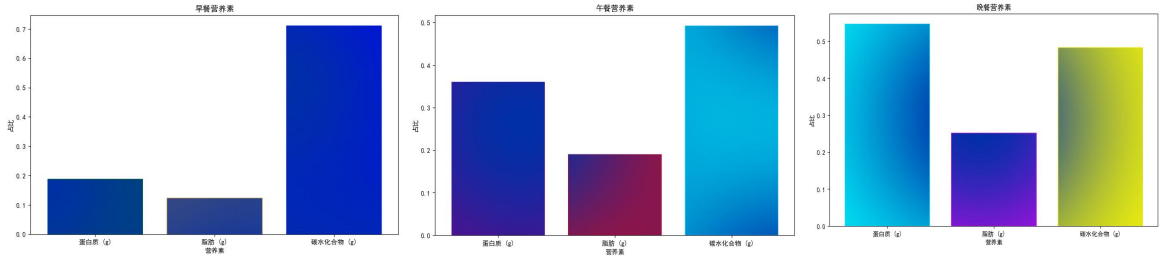



图 9 男生蛋白质目标食谱三餐评价图

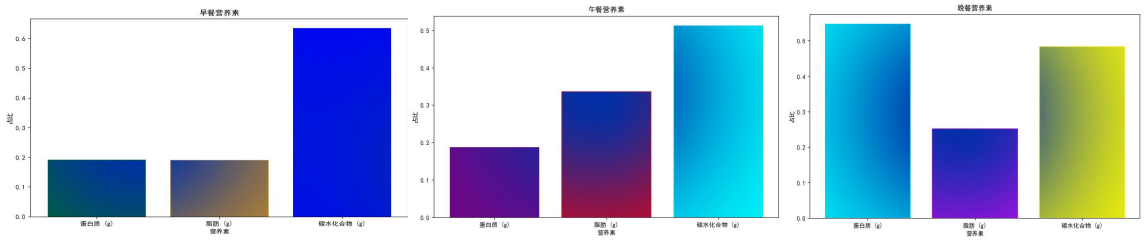



图 10 男生餐费目标食谱三餐评价图

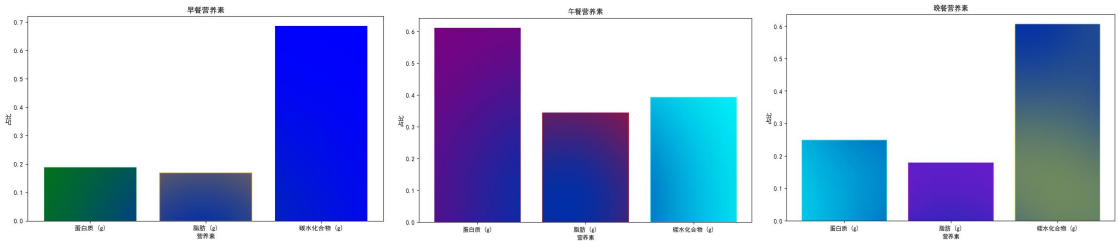


图 11 女生餐费目标食谱三餐评价图

根据所提供的图表数据，我们可以得出以下结论：尽管在食谱设计中特别强调了蛋白质的摄入，但各餐的蛋白质占比并没有达到预期的均衡状态。具体来说，数据显示，随着餐次的推进，蛋白质的占比在不同餐次之间出现了显著的波动，这表明蛋白质的摄入并没有在各餐之间实现均衡分配。

进一步分析，我们发现，尽管蛋白质是食谱设计中的主要关注点，但在实际的营养素分配中，其占比并不是每餐都最高。例如，在某些餐次中，蛋白质的占比可能低于脂肪或碳水化合物。这种营养素分配的不均衡性，随着餐次的增加而变得更加明显，导致蛋白质的摄入在不同餐次之间的差值逐渐增大。

此外，餐费作为另一个重要的考量因素，其在食谱设计中的影响也不容忽视。无论是男性还是女性，在不同餐次中，主要营养素的占比都显示出较大的波动。这种波动不仅体现在蛋白质的摄入上，同样也出现在脂肪和碳水化合物的摄入中。这种营养素占比的波动，显然不符合平衡饮食的要求。

### 4.2.4. 建立以兼顾蛋白质氨基酸评分及经济性的多目标优化模型

目标函数：

$$\min\left(\sum_{j=1}^{N} c_j\, x_{ij} - \lambda \sum_{j=1}^{N} a\, as_j x_{ij}\right)$$

其中，$aas_j$是第 j 种食物的蛋白质氨基酸评分，$\lambda$是权重参数，用于平衡费用和氨基酸评分的影响。

决策变量和约束条件不变。

### 4.2.5. TS-NSGA-II 求解多目标优化模型

TS-NSGA-II，禁忌搜索结合非支配排序遗传算法 II，是一种结合了禁忌搜索和

NSGA-II 算法的多目标优化方法。它利用禁忌搜索的局部搜索能力和 NSGA-II 的全局搜索能力，提高多目标优化问题的求解效果。具体而言，NSGA-II 首先生成初始种群并通过快速非支配排序将种群划分为不同等级，计算每个个体的拥挤度以保持种群的多样性，然后使用二元锦标赛选择、交叉和变异操作生成子代种群。对于子代种群中的部分个体，TS-NSGA-II 引入禁忌搜索进行局部优化，禁忌搜索通过在当前解的邻域内寻找最优解并利用禁忌表避免重复访问，从而提升个体的质量。

### 4.2.6. 多目标优化模型的求解以及结果分析

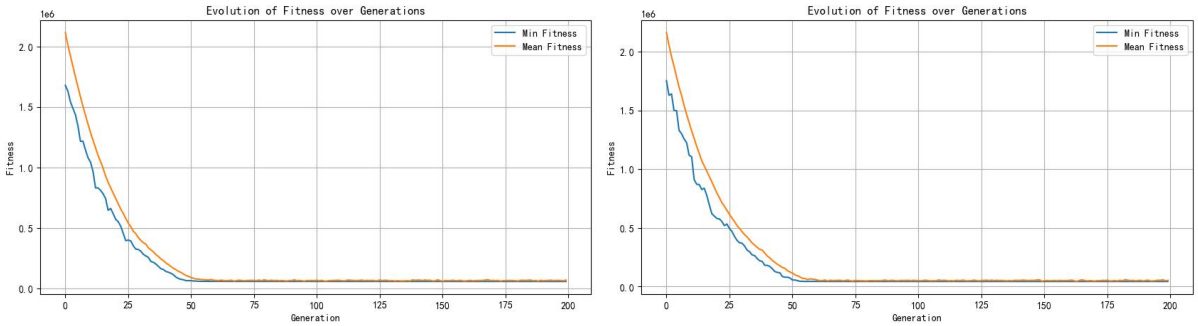



图 12 大学生多目标模型的求解迭代曲线

表 10 男大学生多目标模型推荐食谱

| 食物名称 | 食用份数 | 用餐时间 |
|---|---|---|
| 牛奶 | 2 | 早餐 |
| 煎鸡蛋 | 1 | 早餐 |
| 砂锅面 | 1 | 午餐 |
| 菠菜汤 | 1 | 午餐 |
| 卷心菜炒木耳 | 1 | 午餐 |
| 炖海带白菜豆腐 | 1 | 晚餐 |
| 柚子 | 3 | 晚餐 |

表 11 日多目标膳食评分表

| 食谱类型 | 分值 |
|---|---|
| 男大学生多目标模型推荐食谱 | 0.87 |

| | |
|---|---|
| 女大学生多目标模型推荐食谱 | 0.84 |

表 12 女大学生多目标模型推荐食谱

| 食物名称 | 食用份数 | 用餐时间 |
|---|---|---|
| 牛奶 | 1 | 早餐 |
| 煎鸡蛋 | 1 | 早餐 |
| 拌土豆丝 | 1 | 早餐 |
| 菠菜汤 | 1 | 午餐 |
| 蜜瓜 | 3 | 午餐 |
| 砂锅面 | 1 | 晚餐 |
| 萝卜粉丝汤 | 1 | 晚餐 |
| 炖海带白菜豆腐 | 1 | 晚餐 |
| 柚子 | 3 | 晚餐 |

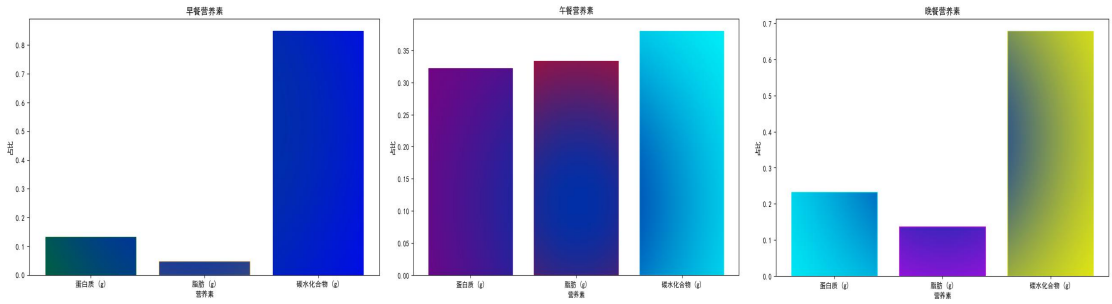



图 13 男生蛋白质餐费目标食谱三餐评价图

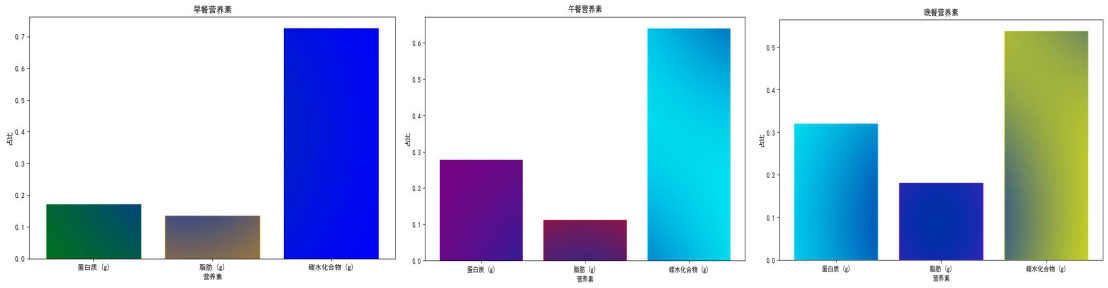



图 14 女生蛋白质餐费目标食谱三餐评价图

在设计男大学生和女大学生的平衡膳食食谱时，除了考虑食物多样性和能量摄入，还需要注意一些具体问题。首先，餐次间的营养素波动即使在相同餐次，不同天的蛋白质摄入比例也显示出波动，这表明需要更细致的食谱规划来确保每日蛋白质摄入的均衡。此外，为了实现平衡饮食，需要对食谱进行优化，确保蛋白质、脂肪和碳水化合物的摄入比例在每餐中都能达到均衡，以满足人体的日常营养需求。

① 餐次间波动：即使是相同餐次，在不同天的蛋白质摄入比例也显示出波动，这表明需要更细致的食谱规划来确保每日蛋白质摄入的均衡。

② 营养素分配优化：为了实现平衡饮食，需要对食谱进行优化，确保蛋白质、脂肪和碳水化合物的摄入比例在每餐中都能达到均衡，以满足人体的日常营养需求。

在设计膳食食谱时，以经济最低为目标虽然可以有效降低膳食成本，但也存在一些显著的弊端。首先，仅以最低成本为目标，可能会忽略其他重要的营养素需求。低成本食物通常营养成分单一，可能导致某些维生素和矿物质摄入不足，从而引起营养不良。此外，廉价食品有时可能质量较低，含有较多的添加剂和保存剂，对健康不利。长期食用低质量食品可能增加健康风险。为了降低成本，食谱中的食物种类可能减少，导致饮食单调乏味，降低用餐愉悦感和长期坚持的可能性。仅以经济为目标还可能忽视了口味、文化习惯和个体偏好等因素，导致膳食方案不具备实际可行性和可接受性。

另一方面，虽然蛋白质氨基酸评分最高的膳食设计能够确保蛋白质质量和氨基酸的摄入，但也有一些潜在的坏处。过于关注蛋白质氨基酸评分，可能导致其他重要的营养素（如脂肪、碳水化合物、纤维素、维生素和矿物质）摄入不足，无法满足全面的营养需求。此外，并不是所有人都需要最高的蛋白质摄入，特别是一些特定人群（如老年人、肾病患者）对高蛋白饮食的需求较低。高蛋白饮食可能加重肾脏负担，对这些人群不利。过高的蛋白质摄入还可能导致脂肪和碳水化合物摄入不足，影响能量平衡和整体饮食结构的合理性。高蛋白食物（如瘦肉、鱼类、奶制品等）通常成本较高，过度追求蛋白质氨基酸评分最高可能导致膳食成本增加，不符合经济最低的初衷。

### 4.3. 问题 3 模型的建立和求解

问题 3 目标为一周制定每日不重复的食谱，同时满足营养需求和经济性。因此我们可以保留问题二中给出的约束条件以及目标函数，增加一定的约束条件将一日食谱拓展成一周食谱，下面我们将介绍改动部分。

目标函数：

$$\max \sum_{i=1}^{7} \sum_{j=1}^{N} a\, as_j x_{ij}$$

$$\text{Min } f = \sum_{i=1}^{7} \sum_{i,j} \left( x_{i,j} \cdot c_{i,j} \right)$$

$$\min \left( \sum_{i=1}^{7} \sum_{j=1}^{N} c_j \, x_{ij} - \lambda \sum_{i=1}^{7} \sum_{j=1}^{N} a \, as_j x_{ij} \right)$$

其中， $c_{i,j}$表示第 j 种，第 i 餐食物的费用（单位：元/克），$aas_j$是第 j 种食物的蛋白质氨基酸评分，$\lambda$ 是权重参数，用于平衡费用和氨基酸评分的影响。

决策变量不变，约束函数新增一条。考虑到一周食谱需要我们摄入的食物种类尽可能丰富，以及总是吃同类食谱可能存在的厌烦情绪，增加了如下约束：

$$\sum_{j=1}^{N} \delta_{ik} \, x_{ij} \leq 1 \quad \forall i, k \neq i$$

其中，$\delta_{ik}$ 是指示函数，如果第 i 天和第 k 天的食谱相同，则为 1，否则为 0。

### 4.3.1. 大学生周平衡膳食的优化食谱以及评价结果

表 13 男大学生周平衡膳食食谱（周一、周二）

| 周一 | | | 周二 | | |
|---|---|---|---|---|---|
| 食物名称 | 食用份数 | 用餐时间 | 食物名称 | 食用份数 | 用餐时间 |
| 牛奶 | 1 | 早餐 | 酸奶 | 1 | 早餐 |
| 煮鸡蛋 | 1 | 早餐 | 煎鸡蛋 | 1 | 早餐 |
| 油条 | 1 | 早餐 | 小米粥 | 1 | 早餐 |
| 大米饭 | 4 | 午餐 | 大米饭 | 4 | 午餐 |
| 红烧肉 | 1 | 午餐 | 炒牛肉 | 1 | 午餐 |
| 地三鲜 | 1 | 午餐 | 拌菠菜 | 1 | 午餐 |
| 拌木耳 | 1 | 午餐 | 木须柿子 | 1 | 午餐 |
| 砂锅面 | 1 | 晚餐 | 鸡排面 | 1 | 晚餐 |
| 包子 | 1 | 晚餐 | 拌菠菜 | 1 | 晚餐 |

表 14 男大学生周平衡膳食食谱（周三、周四）

| 周三 | | | 周四 | | |
|---|---|---|---|---|---|
| 食物名称 | 食用份数 | 用餐时间 | 食物名称 | 食用份数 | 用餐时间 |
| 豆浆 | 1.5 | 早餐 | 小米粥 | 1 | 早餐 |
| 南瓜粥 | 0.5 | 早餐 | 煎鸡蛋 | 1 | 早餐 |
| 花卷 | 1 | 早餐 | 馒头 | 1 | 早餐 |
| 大米饭 | 4 | 午餐 | 大米饭 | 4 | 午餐 |
| 宫保鸡丁 | 1 | 午餐 | 烧排骨 | 1 | 午餐 |

| | | | | | |
|---|---|---|---|---|---|
| 拌干豆腐 | 1 | 午餐 | 炒三丝 | 1 | 午餐 |
| 鸡蛋柿子汤 | 1 | 午餐 | 拌豆腐 | 1 | 午餐 |
| 包子 | 1 | 晚餐 | 拌木耳 | 1 | 晚餐 |
| 拌菠菜 | 2 | 晚餐 | 鸡蛋饼 | 1 | 晚餐 |

表 15 男大学生周平衡膳食食谱（周五、周六）

| 周五 | | | 周六 | | |
|---|---|---|---|---|---|
| 食物名称 | 食用份数 | 用餐时间 | 食物名称 | 食用份数 | 用餐时间 |
| 酸奶 | 1 | 早餐 | 牛奶 | 1 | 早餐 |
| 煎鸡蛋 | 1 | 早餐 | 煎鸡蛋 | 1 | 早餐 |
| 小米粥 | 1 | 早餐 | 小米粥 | 1 | 早餐 |
| 大米饭 | 4 | 午餐 | 大米饭 | 4 | 午餐 |
| 炒牛肉 | 1 | 午餐 | 宫保鸡丁 | 1 | 午餐 |
| 拌木耳 | 2 | 午餐 | 拌干豆腐 | 2 | 午餐 |
| 木须柿子 | 1 | 午餐 | 木须柿子 | 1 | 午餐 |
| 鸡排面 | 1 | 晚餐 | 红烧肉 | 1 | 晚餐 |
| 拌海带丝 | 1 | 晚餐 | 拌木耳 | 1 | 晚餐 |

表 16 男大学生周平衡膳食食谱（周日）

| 周日 | | |
|---|---|---|
| 食物名称 | 食用份数 | 用餐时间 |
| 豆浆 | 1 | 早餐 |
| 鸡排面 | 1 | 早餐 |
| 煎鸡蛋 | 1 | 午餐 |
| 大米饭 | 4 | 午餐 |
| 红烧肉 | 1 | 午餐 |
| 地三鲜 | 1 | 午餐 |
| 拌木耳 | 2 | 午餐 |
| 包子 | 1 | 晚餐 |
| 拌菠菜 | 1 | 晚餐 |

表 17 女大学生周平衡膳食食谱（周一、周二）

| **周一** | | | **周二** | | |
|---|---|---|---|---|---|
| 食物名称 | 食用份数 | 用餐时间 | 食物名称 | 食用份数 | 用餐时间 |
| 豆浆 | 1 | 早餐 | 牛奶 | 1 | 早餐 |
| 鸡排面 | 1 | 早餐 | 鸡蛋饼 | 1 | 早餐 |
| 鸡蛋饼 | 1 | 午餐 | 煎鸡蛋 | 1 | 午餐 |
| 水饺 | 1 | 午餐 | 大米饭 | 2 | 午餐 |
| 葡萄 | 1 | 午餐 | 红烧肉 | 1 | 午餐 |

| | | | | | |
|---|---|---|---|---|---|
| 大米饭 | 2 | 晚餐 | 苹果 | 1 | 午餐 |
| 香菇炒油菜 | 0.50 | 晚餐 | 大米饭 | 2 | 晚餐 |
| 炒肉蒜台 | 0.50 | 晚餐 | 拌菠菜 | 0.50 | 晚餐 |
| 茄汁沙丁鱼 | 0.50 | 晚餐 | 炒牛肉 | 0.50 | 晚餐 |
| 苹果 | 1 | 晚餐 | 香菇炒油菜 | 0.50 | 晚餐 |
| | | | 葡萄 | 1 | 晚餐 |

表 18 女大学生周平衡膳食食谱（周三、周四）

| 周三 | | | 周四 | | |
|---|---|---|---|---|---|
| 食物名称 | 食用份数 | 用餐时间 | 食物名称 | 食用份数 | 用餐时间 |
| 豆浆 | 1 | 早餐 | 牛奶 | 1 | 早餐 |
| 馒头 | 1 | 早餐 | 鸡蛋饼 | 1 | 早餐 |
| 鸡排面 | 1 | 午餐 | 煎鸡蛋 | 1 | 午餐 |
| 大米饭 | 2 | 午餐 | 大米饭 | 2 | 午餐 |
| 红烧肉 | 1 | 午餐 | 红烧肉 | 1 | 午餐 |
| 葡萄 | 1 | 午餐 | 葡萄 | 1 | 午餐 |
| 大米饭 | 2 | 晚餐 | 大米饭 | 2 | 晚餐 |
| 香菇炒油菜 | 0.50 | 晚餐 | 香菇炒油菜 | 0.50 | 晚餐 |
| 炒肉蒜台 | 0.50 | 晚餐 | 炒牛肉 | 0.50 | 晚餐 |
| 茄汁沙丁鱼 | 0.50 | 晚餐 | 茄汁沙丁鱼 | 0.50 | 晚餐 |
| 苹果 | 1 | 晚餐 | 苹果 | 1 | 晚餐 |

表 19 女大学生周平衡膳食食谱（周五、周六）

| 周五 | | | 周六 | | |
|---|---|---|---|---|---|
| 食物名称 | 食用份数 | 用餐时间 | 食物名称 | 食用份数 | 用餐时间 |
| 豆浆 | 1 | 早餐 | 牛奶 | 1 | 早餐 |
| 鸡排面 | 1 | 早餐 | 鸡蛋饼 | 1 | 早餐 |
| 煎鸡蛋 | 1 | 午餐 | 煎鸡蛋 | 1 | 午餐 |
| 大米饭 | 2 | 午餐 | 大米饭 | 2 | 午餐 |
| 红烧肉 | 1 | 午餐 | 红烧肉 | 1 | 午餐 |
| 葡萄 | 1 | 午餐 | 葡萄 | 1 | 午餐 |
| 大米饭 | 2 | 晚餐 | 大米饭 | 2 | 晚餐 |
| 香菇炒油菜 | 0.50 | 晚餐 | 香菇炒油菜 | 0.50 | 晚餐 |
| 炒肉蒜台 | 0.50 | 晚餐 | 炒牛肉 | 0.50 | 晚餐 |
| 茄汁沙丁鱼 | 0.50 | 晚餐 | 茄汁沙丁鱼 | 0.50 | 晚餐 |
| 苹果 | 1 | 晚餐 | 苹果 | 1 | 晚餐 |

表 20 女大学生周平衡膳食食谱（周日）

| 周日 | | |
|---|---|---|
| 食物名称 | 食用份数 | 用餐时间 |
| 豆浆 | 1 | 早餐 |
| 鸡排面 | 1 | 早餐 |
| 煎鸡蛋 | 1 | 午餐 |
| 大米饭 | 2 | 午餐 |
| 红烧肉 | 1 | 午餐 |
| 葡萄 | 1 | 午餐 |
| 大米饭 | 2 | 晚餐 |
| 香菇炒油菜 | 0.50 | 晚餐 |
| 炒肉蒜台 | 0.50 | 晚餐 |
| 茄汁沙丁鱼 | 0.50 | 晚餐 |
| 苹果 | 1 | 晚餐 |

表 21 周膳食评分系数表

| 食谱类型 | 周一 | 周二 | 周三 | 周四 | 周五 | 周六 | 周日 |
|---|---|---|---|---|---|---|---|
| 男 | 0.94 | 0.91 | 0.95 | 0.90 | 0.87 | 0.91 | 0.92 |
| 女 | 0.91 | 0.91 | 0.87 | 0.94 | 0.91 | 0.92 | 0.95 |

根据所提供的男大学生和女大学生的周平衡膳食食谱，从几个方面进行评价和比较分析：根据所提供的男大学生和女大学生的周平衡膳食食谱，从几个方面进行评价和比较分析：

首先，从食物多样性来看，男大学生的食谱中食物种类较为丰富，包括肉类、蛋类、蔬菜、豆制品和主食等，每天的菜品有所变化，能够提供全面的营养素。相对而言，女大学生的食谱虽然也较为丰富，但变化较少，主食和菜品的重复度较高，需要更多样化的选择来提高膳食质量。

在营养均衡方面，男大学生的食谱每餐都包含蛋白质、碳水化合物和脂肪，且蔬菜和豆制品的摄入较为充足，能够提供必要的维生素和矿物质。女大学生的食谱同样涵盖了各类营养素，但可能需要根据个人的能量需求进行适当调整，以确保营养均衡。

能量摄入是另一个重要方面。男大学生的食谱中主食（大米饭）的摄入量较大，有助于满足较高的能量需求，适合其较高的活动量和能量消耗。相比之下，女大学生的食

谱中主食摄入量相对较少，可能需要根据个人的活动量和能量需求进行调整，确保摄入足够的能量以支持日常活动。

膳食评分系数的比较显示，男大学生的膳食评分系数在 0.87 到 0.95 之间，而女大学生的评分系数也在相似范围内。这表明两份食谱的营养均衡程度都较高，但男大学生的食谱在某些天的评分略高，这可能与其食物多样性和能量摄入有关。

从性别差异的角度来看，男大学生的食谱中考虑到了男性通常较高的能量需求和蛋白质需求，因此主食和肉类的摄入量较大。女大学生的食谱中，主食和肉类的摄入量相对较少，这可能与女性较低的能量需求相关。

针对改进建议，对于男大学生，可以考虑增加一些低脂肪的肉类和鱼类，以及更多的蔬菜和水果，以进一步提高营养均衡。而对于女大学生，可以考虑增加主食的多样性，如全谷物、杂粮等，同时适当增加蛋白质的来源，如豆制品、鱼类等。

总结来说，两份食谱都较为均衡，但男大学生的食谱在食物多样性和能量摄入方面略胜一筹。女大学生的食谱则需要根据个人的能量需求适当调整主食和蛋白质的摄入量，以达到更好的营养平衡。

### 4.4. 问题四的求解

关注大学生饮食健康，倡导平衡膳食

亲爱的大学生们：

在校园里，我们共同享受着青春的美好与活力。然而，我们发现，不少大学生在饮食习惯上存在一些不良现象，这些不良习惯不仅影响了我们的身体健康，也阻碍了我们的学业进步和人生发展。为了引导大家树立健康的饮食观念，养成平衡膳食的好习惯，我们特发起此次倡议，希望能够引起大家的共鸣和行动。

一、倡议背景与意义

作为新时代的青年，我们肩负着建设祖国、服务社会的重任。然而，身体健康是实现这一使命的基础。近年来，随着生活节奏的加快和饮食习惯的改变，不少大学生在饮食方面出现了诸多问题，主要表现在以下几个方面：

(1) 不吃早餐或早餐马虎应付。早餐是一天中最重要的一餐，它能为我们的身体提供所需的能量和营养，帮助我们保持清醒的头脑和充沛的精力。然而，一些大学生由于起床晚、时间紧张等原因，经常不吃早餐或只是简单地应付一下。长期如此，会导致营养摄入不足，影响身体健康和学业表现。

(2) 经常性食用外卖及快餐食品。随着外卖平台的兴起，大学生们越来越依赖外卖和快餐食品。这些食品虽然方便快捷，但往往存在油脂过多、营养不均衡等问题。长期食用不仅容易导致肥胖、高血压等健康问题，还会影响我们的饮食口味和习惯。

(3) 通过控制进食来减少皮下脂肪的积存。为了保持苗条的身材，一些大学生选择通过控制饮食来减少皮下脂肪的积存。然而，这种做法往往会导致营养不良和身体健康问题。身体缺乏必要的营养支持，会影响我们的免疫力和抵抗力，增加患病的风险。

针对以上问题，我们发起此次倡议，旨在引导大学生树立健康的饮食观念，养成平衡膳食的好习惯。这不仅有助于我们保持身体健康和良好状态，还有助于我们提高学习效率和生活质量，为未来的发展奠定坚实的基础。

二、倡议内容与措施

为了实现倡议目标，我们提出以下具体内容和措施：

(1) 倡导健康饮食理念。我们要积极宣传健康饮食的重要性，让大家了解健康饮食对身体的好处和不良饮食习惯的危害。同时，我们要倡导“食育”教育，让大家了解食物的营养成分和热量，掌握科学的饮食方法。

(2) 推广平衡膳食理念。平衡膳食是指合理搭配各种食物，保证身体所需的各种营养素得到充足供应。我们要引导大家了解平衡膳食的原则和方法，学会根据自己的身体状况和营养需求制定合理的饮食计划。同时，我们要推广多样化的饮食方式，鼓励大家尝试各种健康食材和烹饪方法。

(3) 倡导早餐的重要性。早餐是一天中最重要的一餐，它能为我们的身体提供所需的能量和营养。我们要倡导大家按时吃早餐，并尽量选择营养丰富、易于消化的食物。学校可以设立早餐供应点，提供健康、美味的早餐选择，方便大家购买和食用。

(4) 减少外卖及快餐食品的摄入。我们要引导大家减少外卖及快餐食品的摄入，尽量自己烹饪食物。学校可以加强食堂建设和管理，提供多样化、健康美味的饭菜选择，满足大家的口味和营养需求。同时，我们也要鼓励大家学会简单的烹饪技能，自己动手做美食，享受烹饪的乐趣。

(5) 合理安排饮食计划。我们要引导大家根据自己的身体状况和营养需求制定合理的饮食计划。不要盲目控制饮食或暴饮暴食，要保证身体所需的各种营养素得到充足供应。同时，我们也要关注饮食的多样性和平衡性，尽量让食物种类丰富多样，保证营养均衡。

(6) 加强健康教育。学校可以加强健康教育课程的开设和宣传力度，让大学生们更加深入地了解健康饮食和平衡膳食的重要性。同时，学校也可以邀请专业营养师和医生来校开展健康讲座和义诊活动，为大家提供科学的健康指导和建议。

(7) 设立健康饮食宣传栏。学校可以在校园内设立健康饮食宣传栏，定期发布健康饮食知识和营养食谱等内容。通过图文并茂的形式吸引大家的注意力，让大家更加直观地了解健康饮食的重要性和方法。

(8) 举办健康饮食主题活动。学校可以定期举办健康饮食主题活动，如健康饮食知识竞赛、烹饪大赛等。这些活动不仅可以增强大家对健康饮食的认识和兴趣，还可以提高大家的实际操作能力和团队协作能力。

三、倡议实施与保障

为了确保倡议的有效实施和持续推进，我们提出以下实施与保障措施：

(1) 加强组织领导。学校可以成立专门的领导小组或工作小组来负责倡议的组织实施和推进工作。领导小组或工作小组要制定详细的工作计划和实施方案，明确各项任务和责任分工，确保倡议的顺利推进。

(2) 加强宣传教育。学校可以通过各种渠道和形式加强宣传教育力度，如开展主题班会、发放宣传资料等。同时，学校也可以利用社交媒体等新媒体平台来扩大宣传范围和影响力。

(3) 加强监督检查。学校可以定期对倡议实施情况进行监督检查和评估总结，及时发现问题和不足之处并采取措施加以改进和完善。同时，学校也可以鼓励大家积极参与倡议的实施和推进工作，形成全校共同关注和支持的良好氛围。

(4) 寻求社会支持。学校可以积极寻求社会各界的支持和帮助来推进倡议的实施工作。例如与政府、企业、社会组织等建立合作关系，共同开展健康饮食和平衡膳食的推广和普及工作。

四、结语

健康饮食是保持身体健康和提高生活质量的重要保障。作为新时代的大学生，我们应该树立健康的饮食观念，养成平衡膳食的好习惯。通过此次倡议的实施和推进工作，我们希望能够引导大家关注饮食健康、关注身体健康，共同营造一个健康、美好的校园氛围。让我们携手共进，为未来的发展奠定坚实的基础！

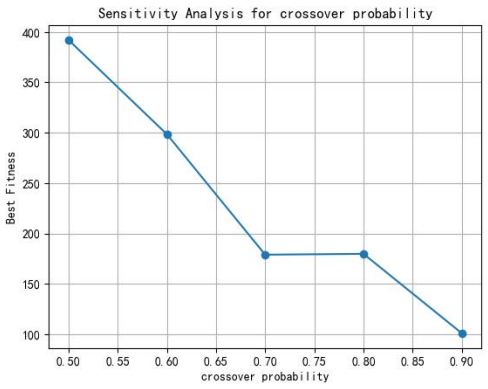
谢谢大家！

[团队 000098]

日期：2024 年 05 月 27 日

# 5. 模型的评价与推广

## 5.1. 模型灵敏度分析



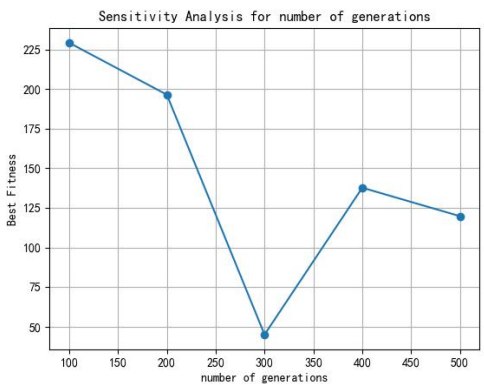




图 15 crossover probability 敏感度曲线　图 16 number of generations 敏感度曲线

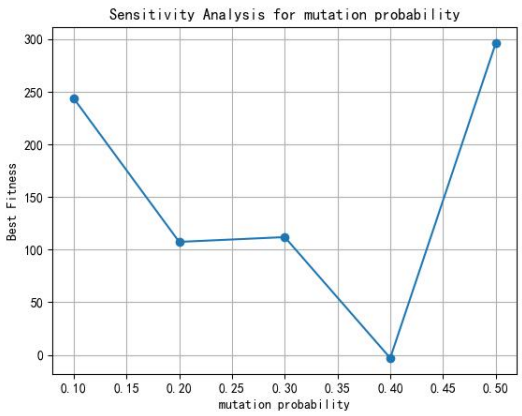



图 17 mutation probability 敏感度曲线

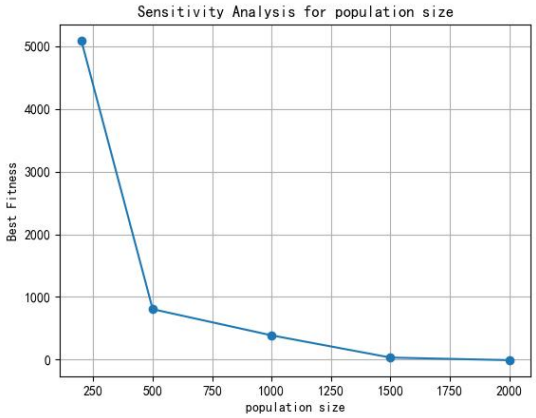



图 18 population size 敏感度曲线

遗传算法（Genetic Algorithm, GA）是一种受自然选择和遗传机制启发的优化算法，其性能在很大程度上依赖于关键参数的设置。本文对代数、变异概率、种群大小和交叉概率四个关键参数进行了敏感度分析，以探讨其对算法性能的影响。

代数敏感度分析： 在代数敏感度分析中，我们观察到随着代数的增加，性能指标（可能是适应度）从 75 提升至 225，表明在初期代数中，算法的性能有显著的增长。然而，当代数超过某个值（例如 400 代）时，性能增长趋于平缓，这可能意味着算法已经接近或达到其最优解。因此，对于遗传算法的迭代次数，选择一个合理的上限是重要的，以避免不必要的计算开销。

变异概率敏感度分析： 变异概率的敏感度分析显示，性能指标在变异概率为 0.20 左右时达到峰值，这表明在这个变异概率下，算法能够平衡探索和开发，从而获得最佳性能。过低的变异概率可能导致算法过早收敛，而过高的变异概率则可能破坏有益的遗传信息，导致性能下降。

种群大小敏感度分析： 种群大小的敏感度分析表明，随着种群大小的增加，性能指标从 1000 提升至 5000。这表明较大的种群能够提供更多的遗传多样性，有助于算法探索解空间。然而，当种群大小超过某个阈值（例如 1500）时，性能的提升开始减缓，这表明进一步增加种群大小的边际效益降低。

交叉概率敏感度分析： 交叉概率的敏感度分析揭示了交叉概率对算法性能的影响。图中显示，当交叉概率从 0.50 增加至 0.85 时，性能指标有所提升。这表明在一定范围内增加交叉概率有助于改善算法性能，因为它促进了遗传信息的重组和有益特征的传播。然而，图中没有显示超过 0.85 的交叉概率对性能的影响，因此我们无法确定是否存在一个最优的交叉概率值。

遗传算法的性能受代数、变异概率、种群大小和交叉概率等关键参数的显著影响。通过敏感度分析，可以优化这些参数设置，从而提升算法的效率和效果。合理的参数设置不仅能有效提高算法性能，还能控制计算开销，确保优化过程在合理的时间和资源范围内完成。这为遗传算法在实际应用中的成功实施提供了重要参考。

### 5.2. 模型的推广

#### 1. 工业生产优化

在工业生产过程中，优化问题无处不在，如生产调度、供应链管理、资源分配等。传统的优化方法在面对复杂的、多变量的非线性问题时往往力不从心。遗传算法通过模拟生物进化过程，以群体为基础进行搜索，能够有效避免陷入局部最优，找到全局最优解。例如，在生产调度问题中，遗传算法可以通过编码生产任务和机器的排列组合，快速找到最优的调度方案，提高生产效率，降低成本。此外，遗传算法在物流和供应链管理中的应用，也能够优化运输路径、库存管理等，显著提升运营效益。

#### 2. 科学研究与工程设计

在科学研究和工程设计中，许多问题的求解需要在庞大的解空间中寻找最优解。例如，结构优化设计、化学反应路径优化、基因工程中的基因序列排列等问题，传统的穷举法和梯度下降法效率低下且容易陷入局部最优。遗传算法的多点搜索和群体进化机制，能够有效探索复杂的解空间，找到接近最优的解。例如，在结构优化设计中，遗传算法可以根据设计参数的变化，自动生成并评估不同的设计方案，找到最佳的结构配置，既节约了时间，又提高了设计质量。

#### 3. 金融与经济决策

在金融与经济领域，市场预测、投资组合优化、风险管理等问题复杂多变，传统的方法难以应对动态变化的市场环境。遗传算法通过模拟进化过程中的选择、交叉和变异操作，能够快速适应市场变化，找到最优的投资组合和风险控制策略。例如，在投资组合优化中，遗传算法可以根据不同资产的历史数据，生成并评估多个投资组合，通过进化过程找到风险和收益的最佳平衡点，帮助投资者制定科学的投资策略。此外，遗传算法在金融衍生品定价和信用风险评估中的应用，也展现了其强大的适应性和优化能力。

## 6. 参考文献

[1]何宇纳，房玥晖，杨晓光，丁钢强.中国健康膳食指数建立与应用[J].营养学报,201

第 39 卷(5): 436-441
[2]何宇纳，房玥晖，夏娟.中国膳食平衡指数的修订:DBI_16[J].营养学报,2018,第 40 卷(6): 526-530
[3]L. Jiang, B. Qiu, X. Liu, C. Huang and K. Lin, "DeepFood: Food Image Analysis and Dietary Assessment via Deep Model," in IEEE Access, vol. 8, pp. 47477-47489, 2020, doi: 10.1109/ACCESS.2020.2973625.
[4]周明，孙树栋编著.遗传算法原理及应用[M].北京：国防工业出版社,1999
[5]Subrata Chakraborty.TOPSIS and Modified TOPSIS: A comparative analysis[J].Decision Analytics Journal,2022,Vol.2: 100021

## 7. 附录

T1.py

```
import pandas as pd
df1 = pd.read_excel('q2.xlsx',sheet_name='蛋白质最大_male')
excel_file = pd.ExcelFile('附件 3：某高校学生食堂一日三餐主要食物信息统计表.xlsx')
sheet_names = excel_file.sheet_names
foods = df1['男生蛋白质氨基酸评分最大'].dropna().values
meals = {}
name = ['早餐','午餐','晚餐']
meal = []

for item in foods:
    if item in name:
        key = item
        meal = []
        continue
    if item != '食物名称':
        meal.append(item)
    else:
        meals[key] = meal
df2 = pd.read_excel('q2.xlsx',sheet_name='蛋白质最大_female')
foods = df2['女生蛋白质氨基酸评分最大'].dropna().values
meals_nv = {}
name = ['早餐','午餐','晚餐']
meal = []

for item in foods:
    if item in name:
        key = item
        meal = []
        continue
    if item != '食物名称':
```

```
            meal.append(item)
        else:
            meals_nv[key] = meal
food_female_unique = food_female.drop_duplicates(subset=['主要成分'])['主要成分'].values
food_male_unique = food_male.drop_duplicates(subset=['主要成分'])['主要成分'].values
food_male_uniqu_len=len(food_female_unique)
food_female_unique_len=len(food_male_unique)
# 读取食物营养素
foods_nutrients = pd.read_excel('种类成分表 1.xlsx')
foods_nutrients.head()
# 计算每餐的总营养素
whole_day_nutrients_male = {'热量  (kcal)': 0, '蛋白质  (g)': 0, '脂肪  (g)': 0, '碳水化合物
(g)': 0}
male_ls1 = []
for key in meals:
    total_nutrients = {'热量  (kcal)': 0, '蛋白质  (g)': 0, '脂肪  (g)': 0, '碳水化合物  (g)': 0}
    for item in meals[key]:
        for _, food in food_male.iterrows():
            if food['食物名称'] == item:
                nutrient = foods_nutrients[foods_nutrients['主要成分'] == food['主要成分
']].iloc[0]

                portion_size = food['可食部（克/份）'] * food['食用份数'] / 100  # 换算成
100g 标准

                total_nutrients['热量  (kcal)'] += nutrient['热量  (kcal)'] * portion_size
                total_nutrients['蛋白质  (g)'] += nutrient['蛋白质  (g)'] * portion_size

                total_nutrients['脂肪  (g)'] += nutrient['脂肪  (g)'] * portion_size
                total_nutrients['碳水化合物  (g)'] += nutrient['碳水化合物  (g)'] *
portion_size
    print(f"{key}的总营养素：", total_nutrients)
    male_ls1.append(total_nutrients)
    whole_day_nutrients_male['热量  (kcal)'] += total_nutrients['热量  (kcal)']
    whole_day_nutrients_male['蛋白质  (g)'] += total_nutrients['蛋白质  (g)']
    whole_day_nutrients_male['脂肪  (g)'] += total_nutrients['脂肪  (g)']
    whole_day_nutrients_male['碳水化合物  (g)'] += total_nutrients['碳水化合物  (g)']
print(f"一天的总营养素：", whole_day_nutrients_male)
```

T2.py

```
import random
```

```
from deap import creator, base, tools, algorithms

# 删除已存在的类
if hasattr(creator, "FitnessMin"):
    del creator.FitnessMin
if hasattr(creator, "Individual"):
    del creator.Individual

# 创建最小化问题的 fitness function
creator.create("FitnessMin", base.Fitness, weights=(-1.0,))

# 创建个体和种群
creator.create("Individual", list, fitness=creator.FitnessMin)

toolbox = base.Toolbox()

# 定义个体生成函数，生成整数列表，表示每种食物的份数
toolbox.register("attr_int", random.randint, 0, 4)
toolbox.register("individual", tools.initRepeat, creator.Individual, toolbox.attr_int,
n=len(df_morning) + len(df_noon) + len(df_evening))

# 定义种群生成函数
toolbox.register("population", tools.initRepeat, list, toolbox.individual)

def evaluate(individual):
    AAS_total = 0
    x_morning = individual[:len(df_morning)]
    x_lunch = individual[len(df_morning):len(df_morning) + len(df_noon)]
    x_dinner = individual[len(df_morning) + len(df_noon):]

    AAS_total = np.dot(x_morning, df_morning['AAS'].values) + np.dot(x_lunch,
df_noon['AAS'].values) + np.dot(x_dinner, df_evening['AAS'].values)
    penalty = 0

    # 设置不同的惩罚系数
    energy_penalty_weight = 50
    meal_penalty_weight = 30
    nutrient_penalty_weight = 20
    ratio_penalty_weight = 40

    # Energy constraints
    penalty += max(0, energy_constraint_min(individual)) * energy_penalty_weight
    penalty += max(0, energy_constraint_max(individual)) * energy_penalty_weight
```

```
    # Breakfast constraints
    penalty += max(0, breakfast_min(individual)) * meal_penalty_weight
    penalty += max(0, breakfast_max(individual)) * meal_penalty_weight

    # Lunch constraints
    penalty += max(0, lunch_min(individual)) * meal_penalty_weight
    penalty += max(0, lunch_max(individual)) * meal_penalty_weight

    # Dinner constraints
    penalty += max(0, dinner_min(individual)) * meal_penalty_weight
    penalty += max(0, dinner_max(individual)) * meal_penalty_weight

    # Nutrient constraints
    penalty += max(0, calcium_min(individual)) * nutrient_penalty_weight
    penalty += max(0, calcium_max(individual)) * nutrient_penalty_weight
    penalty += max(0, iron_min(individual)) * nutrient_penalty_weight
    penalty += max(0, iron_max(individual)) * nutrient_penalty_weight
    penalty += max(0, zinc_min(individual)) * nutrient_penalty_weight
    penalty += max(0, zinc_max(individual)) * nutrient_penalty_weight
    penalty += max(0, vitamin_A_min(individual)) * nutrient_penalty_weight
    penalty += max(0, vitamin_A_max(individual)) * nutrient_penalty_weight
    penalty += max(0, vitamin_B1_min(individual)) * nutrient_penalty_weight
    penalty += max(0, vitamin_B1_max(individual)) * nutrient_penalty_weight
    penalty += max(0, vitamin_B2_min(individual)) * nutrient_penalty_weight
    penalty += max(0, vitamin_B2_max(individual)) * nutrient_penalty_weight
    penalty += max(0, vitamin_C_min(individual)) * nutrient_penalty_weight
    penalty += max(0, vitamin_C_max(individual)) * nutrient_penalty_weight

    # Protein, fat, and carb ratio constraints
    penalty += max(0, protein_energy_ratio_min(individual)) * ratio_penalty_weight
    penalty += max(0, protein_energy_ratio_max(individual)) * ratio_penalty_weight
    penalty += max(0, fat_energy_ratio_min(individual)) * ratio_penalty_weight
    penalty += max(0, fat_energy_ratio_max(individual)) * ratio_penalty_weight
    penalty += max(0, carb_energy_ratio_min(individual)) * ratio_penalty_weight
    penalty += max(0, carb_energy_ratio_max(individual)) * ratio_penalty_weight

    return -AAS_total + penalty,

toolbox.register("mate", tools.cxTwoPoint)
toolbox.register("mutate", tools.mutUniformInt, low=0, up=10, indpb=0.2)
toolbox.register("select", tools.selTournament, tournsize=3)
toolbox.register("evaluate", evaluate)
```

```
## 设置遗传算法的参数
# population = toolbox.population(n=1300)
# ngen = 200
# cxpb = 0.7
# mutpb = 0.2

## 运行遗传算法
# algorithms.eaSimple(population, toolbox, cxpb, mutpb, ngen, stats=None, halloffame=None, verbose=True)

## 获取最优解
# best_individual = tools.selBest(population, k=1)[0]

# print("最优解：", best_individual)
# print("最优值：", evaluate(best_individual))



import matplotlib.pyplot as plt
def plot_evolution(logbook):
    generations = logbook.select("gen")
    min_fitness_values = logbook.select("min")
    mean_fitness_values = logbook.select("mean")

    plt.figure(figsize=(10, 5))
    plt.plot(generations, min_fitness_values, label='Min Fitness')
    plt.plot(generations, mean_fitness_values, label='Mean Fitness')
    plt.xlabel('Generation')
    plt.ylabel('Fitness')
    plt.title('Evolution of Fitness over Generations')
    plt.legend()
    plt.grid()
    plt.show()

def run_ea():
    population = toolbox.population(n=1300)
    ngen = 200
    cxpb = 0.7
    mutpb = 0.2

    stats = tools.Statistics(key=lambda ind: ind.fitness.values)
    stats.register("min", np.min)
```

```
    stats.register("mean", np.mean)

    logbook = tools.Logbook()
    logbook.header = ['gen', 'nevals'] + stats.fields

    for gen in range(ngen):
        offspring = algorithms.varAnd(population, toolbox, cxpb, mutpb)
        fits = list(map(toolbox.evaluate, offspring))
        for fit, ind in zip(fits, offspring):
            ind.fitness.values = fit
        population = toolbox.select(offspring, k=len(population))

        record = stats.compile(population)
        logbook.record(gen=gen, nevals=len(population), **record)
        print(logbook.stream)
    global best_individual
    best_individual = tools.selBest(population, k=1)[0]
    print("最优解：", best_individual)
    print("最优值：", evaluate(best_individual))

    return logbook

# 运行遗传算法并获取日志记录
logbook = run_ea()

# 绘制迭代曲线
plot_evolution(logbook)
```

T3.py

```
import random
import numpy as np
import matplotlib.pyplot as plt
from deap import base, creator, tools, algorithms

# 删除已存在的类
if hasattr(creator, "FitnessMin"):
    del creator.FitnessMin
if hasattr(creator, "Individual"):
    del creator.Individual

# 创建最小化问题的 fitness function
```

```
creator.create("FitnessMin", base.Fitness, weights=(-1.0, 1.0))  # 最小化 total_cost，最大化 total_aas

# 创建个体和种群
creator.create("Individual", list, fitness=creator.FitnessMin)

toolbox = base.Toolbox()

# 定义个体生成函数，生成整数列表，表示每种食物的份数
toolbox.register("attr_int", random.randint, 0, 4)
toolbox.register("individual", tools.initRepeat, creator.Individual, toolbox.attr_int,
n=len(foods_grouped) * 7)

# 定义种群生成函数
toolbox.register("population", tools.initRepeat, list, toolbox.individual)

# 注册选择函数
toolbox.register("select", tools.selNSGA2)

# 注册交叉和变异函数
toolbox.register("mate", tools.cxTwoPoint)
toolbox.register("mutate", tools.mutUniformInt, low=0, up=10, indpb=0.2)

# 评估函数：兼顾蛋白质氨基酸评分及经济性
def evaluate_weekly_combined(individual):
    total_cost = 0
    total_aas = 0
    penalty = 0
    unique_meals = set()

    for day in range(7):
        x_day = individual[day*len(foods_grouped):(day+1)*len(foods_grouped)]
        x_morning = x_day[:len(df_morning)]
        x_lunch = x_day[len(df_morning):len(df_morning)+len(df_noon)]
        x_dinner = x_day[len(df_morning)+len(df_noon):]

        day_cost = np.dot(x_morning, df_morning['价格\n（元/份）'].values) + \
                   np.dot(x_lunch, df_noon['价格\n（元/份）'].values) + \
                   np.dot(x_dinner, df_evening['价格\n（元/份）'].values)
        total_cost += day_cost

        day_aas = np.dot(x_morning, df_morning['AAS'].values) + \
                  np.dot(x_lunch, df_noon['AAS'].values) + \
```

```
                    np.dot(x_dinner, df_evening['AAS'].values)
        total_aas += day_aas

        day_energy = np.dot(x_morning, df_morning['热量  (kcal)'].values) + \
                     np.dot(x_lunch, df_noon['热量  (kcal)'].values) + \
                     np.dot(x_dinner, df_evening['热量  (kcal)'].values)
        day_protein = np.dot(x_morning, df_morning['蛋白质  (g)'].values) + \
                      np.dot(x_lunch, df_noon['蛋白质  (g)'].values) + \
                      np.dot(x_dinner, df_evening['蛋白质  (g)'].values)
        day_fat = np.dot(x_morning, df_morning['脂肪  (g)'].values) + \
                  np.dot(x_lunch, df_noon['脂肪  (g)'].values) + \
                  np.dot(x_dinner, df_evening['脂肪  (g)'].values)
        day_carb = np.dot(x_morning, df_morning['碳水化合物  (g)'].values) + \
                   np.dot(x_lunch, df_noon['碳水化合物  (g)'].values) + \
                   np.dot(x_dinner, df_evening['碳水化合物  (g)'].values)
        day_servings = np.sum(x_day)

        # 计算不重复食谱的惩罚
        day_meal = tuple(x_day)
        if day_meal in unique_meals:
            penalty += 1000  # 对重复食谱进行高惩罚
        else:
            unique_meals.add(day_meal)

        if day_energy > 4000:
            penalty += (day_energy - 4000) * 10
        if day_energy < 2400:
            penalty += (2400 - day_energy) * 10
        if day_protein < 60:
            penalty += (60 - day_protein) * 10
        if day_fat < 53:
            penalty += (53 - day_fat) * 10
        if day_carb < 300:
            penalty += (300 - day_carb) * 10
        if day_servings > 25:
            penalty += (day_servings - 25) * 10

    return total_cost + penalty, total_aas

toolbox.register("evaluate", evaluate_weekly_combined)

# 设置遗传算法的参数
```

```
population = toolbox.population(n=1000)  # 种群大小
ngen = 1000  # 迭代次数
cxpb = 0.7  # 交叉概率
mutpb = 0.2  # 变异概率

# 迭代曲线数据收集
stats = tools.Statistics(lambda ind: ind.fitness.values)
stats.register("avg", np.mean, axis=0)
stats.register("min", np.min, axis=0)

# 运行 NSGA-II
population, logbook = algorithms.eaMuPlusLambda(population, toolbox, mu=1000,
lambda_=1000, cxpb=cxpb, mutpb=mutpb, ngen=ngen, stats=stats, halloffame=None,
verbose=True)

# 获取帕累托前沿
pareto_front = tools.sortNondominated(population, len(population), first_front_only=True)[0]

# 绘制帕累托前沿
costs = [ind.fitness.values[0] for ind in pareto_front]
aas_scores = [ind.fitness.values[1] for ind in pareto_front]

plt.figure(figsize=(12, 5))

plt.subplot(1, 2, 1)
plt.scatter(costs, aas_scores, c='r', marker='o')
plt.xlabel('Total Cost')
plt.ylabel('Total AAS Score')
plt.title('Pareto Front')
plt.grid(True)

# 迭代曲线绘制
gen = logbook.select("gen")
min_costs = [entry[0] for entry in logbook.select("min")]

plt.subplot(1, 2, 2)
plt.plot(gen, min_costs, label='Minimum Cost')
plt.xlabel('Generation')
plt.ylabel('Cost')
plt.title('Evolutionary Algorithm Convergence')
plt.legend()
plt.grid(True)
```

```
plt.tight_layout()
plt.show()

# 输出每天的最优值
best_individual = tools.selBest(population, k=1)[0]
print("最优解：", best_individual)
print("最优值：", toolbox.evaluate(best_individual))

for day in range(7):
    x_day = best_individual[day*len(foods_grouped):(day+1)*len(foods_grouped)]
    x_morning = x_day[:len(df_morning)]
    x_lunch = x_day[len(df_morning):len(df_morning)+len(df_noon)]
    x_dinner = x_day[len(df_morning)+len(df_noon):]

    day_cost = np.dot(x_morning, df_morning['价格\n（元/份）'].values) + \
               np.dot(x_lunch, df_noon['价格\n（元/份）'].values) + \
               np.dot(x_dinner, df_evening['价格\n（元/份）'].values)
    day_aas = np.dot(x_morning, df_morning['AAS'].values) + \
              np.dot(x_lunch, df_noon['AAS'].values) + \
              np.dot(x_dinner, df_evening['AAS'].values)

    print(f"Day {day+1} - Cost: {day_cost}, AAS: {day_aas}")

# 将每天的最优值存储在表格中
data = []
for day in range(7):
    x_day = best_individual[day*len(foods_grouped):(day+1)*len(foods_grouped)]
    x_morning = x_day[:len(df_morning)]
    x_lunch = x_day[len(df_morning):len(df_morning)+len(df_noon)]
    x_dinner = x_day[len(df_morning)+len(df_noon):]

    day_cost = np.dot(x_morning, df_morning['价格\n（元/份）'].values) + \
               np.dot(x_lunch, df_noon['价格\n（元/份）'].values) + \
               np.dot(x_dinner, df_evening['价格\n（元/份）'].values)
    day_aas = np.dot(x_morning, df_morning['AAS'].values) + \
              np.dot(x_lunch, df_noon['AAS'].values) + \
              np.dot(x_dinner, df_evening['AAS'].values)

    data.append([day+1, day_cost, day_aas])

# 创建数据框并显示
df_results = pd.DataFrame(data, columns=["Day", "Cost", "AAS"])
print(df_results)
```

```
# 保存表格到文件
df_results.to_csv("weekly_best_values.csv", index=False)
```