报名序号：003430

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

大学时代是知识增长与良好饮食习惯形成的关键阶段。在这一时期，大学生需要关注自己的营养需求和饮食习惯，合理搭配食物、均衡摄入营养素，以保障身体健康和智力发展的需求。本文对运用优化模型对大学生平衡膳食食谱进行了优化设计，并给出了具体的设计方案。

问题一中我们首先基于TOPSIS理论建立了膳食营养评价模型，通过计算膳食中食物结构、各类营养素的摄入量、食谱的能量来源、每餐的蛋白质氨基酸等五个方面与文献中推荐标准值的对比并得到相应的权重，识别出现有食谱的不足。然后，基于附3的学校食堂食材信息，对男女大学生的食谱分别进行了调整，并利用膳食营养评价模型对改进后的食谱进行了再评价，改进后的食谱评分显著提高。

问题二要求分别就“以蛋白质氨基酸评分最大为目标”“以用餐费用最经济为目标”“兼顾蛋白质氨基酸评分及经济性为目标”分别对设计男女生的日食谱。因此，我们进一步设计了针对性的单一目标及多目标优化模型。在构建这些模型的过程中，我们充分考虑了多种约束条件。这些约束条件包括摄入食物种类的多样性、各种营养素供能占比的合理性、营养素摄入量的充足性以及早午晚餐比例的适宜性等。最后我们得到了对应目标下的日食谱，并且通过问题一得到的膳食营养评价模型对所设计的食谱进行了深入的分析和评价并对比。这一评价验证了我们所采用优化方法的有效性。

问题三为在前两问的基础上增加周食谱的膳食营养平衡要求后再分别给出三个优化目标的设计，并同样进行评价及比较分析。相似地，我们将日平衡膳

食食谱优化模型扩展到一周的时间范围内，通过遗传优化算法，我们得到了符合各项约束条件的膳食方案，并对这些方案进行了详细的营养评价和成本分析。营养评价主要包括蛋白质、脂肪、碳水化合物等主要营养素的摄入量分析，以及维生素、矿物质等微量元素的摄入评估。成本分析则主要考虑了食材的售卖价格。通过对比不同优化模型的结果，我们发现多目标优化模型在平衡营养需求和经济性方面表现更为出色。同时，我们也发现男生和女生的膳食需求存在一定的差异，因此在实际应用中需要根据不同人群的特点进行个性化调整。

问题里我们结合前三问的研究结果，撰写了一份倡议书，基于实际得到的数据提出了科学合理的饮食建议，旨在改善大学生的饮食习惯和营养健康状况。

本文综合考虑了各个方面，建立了符合条件的优化模型，采用了线性规划、遗传算法等数学方法，并借助 Matlab 等软件进行了计算与优化，求得的结果完全满足题目的要求，并提出了适用于大学生的平衡膳食方案有一定的实际应用价值。

关键词：线性规划、遗传算法、单目标规划、多目标规划、TOPSIS 评估模型

一、 问题重述

近年来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，大学生的膳食营养平衡是保障年轻一代身体健康的很重要的一个环节。我国很早就提出了标准化的居民膳食指南。根据平衡膳食的基本准则，大学生们存在着对食堂已有食谱的疑问：能否评估以及改进大学生自己的食谱使之达到标准值，才能形成良好的饮食习惯，健康成长。

现在以本次比赛的附件数据为例进行优化设计。

1.1 提出问题

问题一：膳食食谱的营养分析评价及调整

1.1 膳食营养评价

首先我们需要对附件1和附件2中的男、女大学生的膳食进行全面的营养评

价。根据附件4的标准，评价内容包括能量、主要营养素含量（蛋白质、脂肪、碳水化合物等）、 非产能营养素（钙、铁、锌、维生素等）及氨基酸评分等。

1.2 调整改进

基于附件3提供的食堂主要食物信息，对男、女大学生的膳食进行适当调整，使其 更符合营养需求，然后再进行营养评价。

问题二：基于附件 3 的日平衡膳食食谱的优化设计

2.1 目标一：蛋白质氨基酸评分最大化

建立优化模型，设计男、女大学生的日食谱，并进行膳食营养评价。

2.2 目标二：用餐费用最经济

建立优化模型，设计男、女大学生的日食谱，并进行膳食营养评价。

2.3 目标三：兼顾蛋白质氨基酸评分及经济性

建立综合优化模型，设计男、女大学生的日食谱，并进行膳食营养评价。

2.4 比较分析

对上述三种优化方案进行比较分析，找出最优方案。

问题三：基于附件 3 的周平衡膳食食谱的优化设计。

在问题 2 的基础上，分别以蛋白质氨基酸评分最大、用餐费用最经济、兼顾蛋白 质氨基酸评分及经济性为目标，设计男、女大学生的周食谱，并进行评价及比较分析。

问题四：健康饮食、平衡膳食的倡议书。

基于以上分析和设计，针对大学生饮食结构及习惯，写一份健康饮食、平衡膳食的倡议书。

二、条件假设

1) 食谱中列出的所有食材的营养成分数据是准确且全面的；
2) 附件 1 附件 2 和附件 3 的食谱所列食材的食用量是实际摄取量，并不存在浪费行为；
3) 个人的营养需求根据一般推荐标准，不考虑个体差异；
4) 一周内各天的营养需求是均衡的，不考虑周末或特殊情况；
5) 食材的营养成分、价格和可获取性在一周内保持不变，不考虑价格波动。

三、符号说明

本文所涉及的模型符号说明如表 1 所示。

表 1 模型符号说明

序号	符号	说明
1	c_i	第 i 个食品的费用
2	m_{ij}	第 i 个食品 j 个成分的克数
3	f_{kij}	第 i 个食品 j 个成分第 k 个营养素的含量
4	v_{kij}	第 i 个食品 j 个成分第 k 个非产能营养素含量
5	a_{kij}	第 i 个食品 j 个成分第 k 个氨基酸含量
6	x_{pqi}	第 p 天第 i 个食品的早中晚的份数
7	α_k	第 k 类营养素的能量转换系数

四、 问题分析

4.1 问题一分析

要对附件1和附件2两份食谱做出全面的膳食营养评价，我们首先从附件1和附件2中提取两名学生一日三餐的食物摄入情况。然后查询《中国食物成分表》获取每种食物的主要营养成分数据。接下来我们对以下几个重要方面进行计算

（1）营养素含量计算：计算每种食物的营养素含量（包括能量、蛋白质、脂肪、碳水化合物等），并合并得到每日总摄入量。

（2）能量和营养素评价：依据《中国居民膳食指南》中大学生每日能量和营养素的参考摄入量进行比较和评价。确认每日能量、蛋白质、脂肪和碳水化合物的摄入比例是否在合理范围内。评价非产能营养素（如钙、铁、锌、维生素等）的摄入情况。

（3）氨基酸评分（AAS）评价：计算每日摄入食物的蛋白质氨基酸评分，确定蛋白质质量是否合理。

要基于附件3，对附件1、附件2两份食谱进行调整改进并再做出全面的膳食营养评价，我们根据根据附件3提供的食物信息，选择合适的食物替代部分不合理的食物，调整以满足营养需求。我们再次计算调整后食谱的总营养素含量，对调整后的食谱再次进行能量和营养素的全面评价。比较调整前后的变化，确认调整是否达到了优化的目的，以此来验证我们的评估模型是否准确。

4.2 问题二分析

第一小问要求以蛋白质氨基酸评分最大为目标建立优化模型，我们首先构建了一个目标函数，函数的自变量为附件3中的菜谱，函数的因变量为最大化每日食谱中蛋白质的氨基酸评分。约束条件：满足每日能量、蛋白质、脂肪、碳水化合物及其他非产能营养素的需求。这里我们选择用线性规划算法求解得到相应的模型。最后我们用问题一中的评价模型对优化后食谱进行全面的营养评价，确保所有营养素的摄入量在合理范围内。

第二小问同样地参考第一小问的思路流程，将目标函数替换为最小化每日食谱的总费用，约束条件不变。得到模型后，我们同样对优化后食谱进行全面的营养评价，确保所有营养素的摄入量在合理范围内。

第三小问的目标函数兼顾问题一和问题二的目标函数，这里我们采用常用的多目标优化算法，这里我们采用的是加权线性和方法求解该模型。即将多个目标函数进行加权求和，将多目标优化问题转化为单目标优化问题。通过调整权重可以平衡各个目标函数之间的重要性。

第四小问比较三种优化方案的营养评价结果。我们通过分析每种方案的优缺点，最终给出综合评价。

4.3 问题三分析

题目要求以蛋白质氨基酸评分最大、用餐费用最经济、兼顾蛋白质氨基酸评分及经济性为目标，分别建立优化模型，设计男生和女生的周食谱。我们注意到类似于问题 2 的模型建立，但将时间跨度扩展至一周，需要考虑每天的变异，同时也要满足问题二中的日食谱均衡。我们对对每个优化模型下的周食谱进行营养评价，确保每周的食谱在营养素摄入上满足合理性。

4.4 问题四分析

我们将上面三问总结膳食调查和营养评价的结果，找出了主要问题。基于男女生差异、多个优化模型的横向比较，提供了可靠的健康饮食建议，包括食物多样化、均衡营养摄入、考虑到大学生尚未工作，在用餐费用经济的情况下保障自生营养的吸收达到国家标准。

五、问题一模型建立与求解

5.1 问题一求解思路

第一小问，作出全面的膳食营养评价，即根据附件四的各项指标要求对附件一的两份食谱进行计算，对每个食物得到对应的营养物质含量，再对食物结构、主要营养素含量、能量、餐次比及非产能主要营养素含量、能量来源、蛋白质氨基酸评分等分开进行全面的评价；

第二小问，“较少的调整”这句话道出了优化目标：调整的食物量少，约束条件为满足附件四的各个评价要求。

5.2 问题一第一小问模型求解

根据以上模型对于附件一和附件二中的食谱分别进行计算评价指标，并根据评价结果对整体进行评价并给出建议。

评价这个同学的具体摄入情况，标准参考：

1．分析食物结构

按类别将食谱中食物归类排序，列出每种食物的数量，分析五大类别食物是否齐全，食物种类是否大于 12 种（周食谱评价要求大于 25 种）。

2．计算食谱的主要营养素含量

从《中国食物成分表》第一册、第二册中查出每100克可食部食物所含主要营养素的数量，从而算出食谱中各种主要营养素的含量。

3．评价食谱提供的能量、餐次比及非产能主要营养素含量

根据每日能量摄入目标、餐次比参考值以及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 的参考摄入量，对食谱进行评价。

4．评价食谱的能量来源

将食谱中碳水化合物、脂肪、蛋白质供能占总能量的百分比与宏量营养素供能占总能量百分比的参考值比较，对食谱的能量来源进行评价。

5．评价每餐的蛋白质氨基酸评分

计算每餐混合食物的蛋白质氨基酸评分，依此评价每餐食物蛋白质氨基酸组成是否合理。一般地，混合食物蛋白质氨基酸评分小于60为不合理，60-80为不够合理，80-90为比较合理，大于90为合理。

（注意：混合食物蛋白质氨基酸评分按照每餐计算）

4. 3. 1 对男同学的分析

食物种类总的食物数量19种。不同类别食物的数量:谷、薯类:5；蔬菜、菌藻、水果类:7；畜、禽、鱼、蛋类及制品:4；奶、干豆、坚果、种子类及制品:1；植物油类:2

具体营养素计算结果如下：

表1 营养素摄入量（男生）

能量	2262.02kcal
蛋白质	82.45g
脂肪	95.21g
碳水化合物	304.43g
膳食纤维	23.36g
叶酸	328.18μg
钙	804.81mg
铁	11.06mg
锌	19.57mg
维生素A	643.74μg
维生素B1	1.48mg
维生素B2	1.72mg
维生素C	157.12mg

总的来说，该同学的能量摄入明显超过了正常的男生标准2400kcal/d能量摄入的，可以看到该同学的脂肪能量占比明显太高，其蛋白质摄入也超出了正常的15%，所以对于该同学来说，总的需要增加碳水化合物的能量摄入。

男生其日常营养摄入量基本都是满足的，其中超的比较多的是钙、铁、锌以及维生素A，对于维生素C缺少得比较多，在后续的饮食计划中可以考虑将维生素A含量高的食物换成维生素C含量高的食物。

表2 氨基酸评分和氨基酸摄入量（男生）

早餐的 AAS 摄入评分	125.74920070649
午餐的 AAS 摄入评分	88.02551811866526
晚餐的 AAS 摄入评分	101.33517609060078
异亮氨酸	45.784567
亮氨酸	88.272168
色氨酸	13.382265
含硫氨基酸	18.964491
芳香族氨基酸	38.063876
苏氨酸	30.930371
缬氨酸	50.426503
赖氨酸	57.254513

可以看到，对于该同学，其含硫氨基酸、芳香族氨基酸、苏氨酸摄入较少，亮氨酸摄入较多。

4.3.2 对女同学的分析

总的食物数量:15；不同类别食物的数量:谷、薯类:2；蔬菜、菌藻、水果类:6；畜、禽、鱼、蛋类及制品: 5；奶、干豆、坚果、种子类及制品: 1；植物油类: 1。对于这个女同学来说，其满足了总食物数量大于 12 种的标准，且五大种类的食物都有所摄入，但其中奶、干豆、坚果、种子类及制品和植物油类的摄入较少，可以考虑多摄入这种类别的食物。

总的来说，该同学的能量摄入是达到了正常的女生标准 1900kcal/d 能量摄入的，可以看到该同学的脂肪能量占比明显太高，其蛋白质摄入也超出了正常的 15%，所以对于该同学来说，总的需要增加碳水化合物的能量摄入,但相对来说，这个女同学的脂肪摄入相对男同学更少，碳水相对更多，但都是同样的问题 。

除了锌、维生素 A 和维生素 B1 之外，该同学的其他都比较缺乏

表3 氨基酸评分和氨基酸摄入量（女生）

早餐的 AAS 摄入评分	113.48143345475664
午餐的 AAS 摄入评分	117.23301270246674
晚餐的 AAS 摄入评分	93.29575186718046
异亮氨酸	48.159001
亮氨酸	91.323736
色氨酸	14.167410
含硫氨基酸	18.849535
芳香族氨基酸	41.272774
苏氨酸	34.061664
缬氨酸	52.841126
赖氨酸	66.065741

根据上述数据，得出以下结论：

（1）附件 1 中男性的能量、脂肪、碳水化合物、膳食纤维、叶酸、钙、铁、锌、维生素 A 和 维生素 C 的摄入量均超过了建议摄入量。

（2）附件 1 中男性的蛋白质摄入量略高于建议摄入量。

（3）附件 2 中女性的能量、脂肪、碳水化合物、膳食纤维、叶酸、钙、铁、锌、维生素 A 和 维生素 C 的摄入量均超过了建议摄入量。

（4）附件 2 中女性的蛋白质摄入量略高于建议摄入量。

5.3 问题一第二小问，进行较少的调整

对于男同学，其含硫氨基酸、芳香族氨基酸、苏氨酸摄入较少，亮氨酸摄入较多。综上所述，我们对该同学进行食物消减，然后增加维生素、碳水化合物的摄入,消减脂肪的摄入

采用的方案为：将该同学早餐的 2 份油条消减为 1 份，增加橙子的摄入，然后将午餐的红烧肉修改为溜肉段，删除晚餐的包子，

增加 4 份大米粥。

表4 调整后男大学生的食谱

食物名称	主要成分	食物编码	可食部（克/份）	食用份数	餐类
小米粥	小米	015101	15	1	早餐
煎鸡蛋	鸡蛋	111101x	50	1	早餐
	豆油	192004	10		
油条	小麦粉	011201x	50	1	早餐
	豆油	192004	10		
橙子	橙	064101	100	1	早餐
拌海带丝	海带	052004	100	1	早餐
	芝麻油	192017	2		
大米饭	稻米	012001x	25	4	午餐
拌木耳	木耳	051014	100	1	午餐
	芝麻油	192017	2		
地三鲜	茄子	043101x	80	1	午餐
	土豆	021101	80		
	青椒	043123	10		
	豆油	192004	10		
溜肉段	猪肉	081111x	50	1	午餐
	青椒	043123	10		
	胡萝卜	041201	10		
	豆油	192004	15		
砂锅面	玉米面	013109	80	1	晚餐
	白菜	045101x	20		
	油菜	045125	20		
	干豆腐	031509	10		
	豆油	192004	10		
大米粥	稻米	012001x	15	4	晚餐

对于女同学，总的来说，我们需要消减该同学的脂肪摄入、增加碳水和各个营养物质的摄入（特别是钙）

具体方案为，将该同学早餐的鸡排面换成2份橙子，增加午餐海带丝1份。

表5 调整后女大学生的食谱

食物名称	主要成分	食物编码	可食部（克/份）	食用份数	餐类
豆浆	黄豆	031101	10	1	早餐
橙子	橙	064101	100	2	早餐
鸡蛋饼	小麦粉	011201x	25	1	午餐
	鸡蛋	111101x	20		
	火腿肠	081409	20		
	豆油	192004	5		
水饺	小麦粉	011201x	50	1	午餐
	猪肉	081111x	20		
	白菜	045101x	40		
	豆油	192004	10		
葡萄	葡萄	063101x	100	1	午餐
拌海带丝	海带	052004	100	1	午餐
	芝麻油	192017	2		
大米饭	稻米	012001x	25	2	晚餐
香菇炒油菜	油菜	045125	100	1/2	晚餐
	香菇	051019	20		
	豆油	192004	5		
蒜苔炒肉	蒜苔	044106	100	1/2	晚餐
	猪肉	081111x	30		
	豆油	192004	5		
茄汁沙丁鱼	茄汁沙丁鱼	121417	100	1/2	晚餐
苹果	苹果	061101x	100	1	晚餐

六、问题二模型建立与求解

6.1 问题二求解思路

针对问题二，基于附件三的日膳食食堂食谱进行优化设计，自变量为每餐每种食物的份数，分别根据蛋白质氨基酸评分最大、以用餐费用最经济为目标约束条件为满足附件四的各个评价要求，建立单目标优化模型，兼具蛋白质氨基酸评分最大和以用餐费用最经济为目标，约束条件为满足附件四的各个评价要求，对于每天食谱建立多目标优化模型。

6.2 第一小问 以蛋白质氨基酸评分最大为目标建立优化模型

6.2.1 建模

自变量：第i个食品的早中晚的份数$x_{1i}x_{2i}x_{3i}$

目标函数：

$$y=\max\mathrm{AAS}=\max\frac{\sum_{q=1}^{3}\sum_{i=1}^{n}\sum_{j=1}^{m}x_{qi}m_ja_{kij}}{\sum_{q=1}^{3}\sum_{i=1}^{n}\sum_{j=1}^{m}\sum_{k=1}^{8}x_{qi}m_ja_{kij}}\ ,\quad k=1\ldots8$$

约束条件：

1）每天摄入食物种类>12 种：

$$i>12$$

2)每天摄入食物包含五大类

3)男、女生每日能量实际摄入量与摄入量目标相差在±10%之内。摄入总能量摄入目标为：女生 1900kcal/d，男生 2400kcal/d。

设摄入目标为z

$$0.9z<cal=\sum_{q=1}^{3}\sum_{i=1}^{n}\sum_{j=1}^{m}x_{qi}m_ja_kf_{kij}<1.1z$$

其中a_k为能量转换系数

4)宏量营养素供能占比

蛋白质 10%-15%、脂肪 20%-30%、碳水化合物 50%-65%

$$10\% < \frac{\sum_{q=1}^{3}\sum_{i=1}^{n}\sum_{j=1}^{m}\sum_{k=1}^{1} x_{qi}m_j a_k f_{kij}}{cal} < 15\%,$$
$$20\% < \frac{\sum_{q=1}^{3}\sum_{i=1}^{n}\sum_{j=1}^{m}\sum_{k=2}^{2} x_{qi}m_j a_k f_{kij}}{cal} < 30\%,$$
$$50\% < \frac{\sum_{q=1}^{3}\sum_{i=1}^{n}\sum_{j=1}^{m}\sum_{k=3}^{3} x_{qi}m_j a_k f_{kij}}{cal} < 65\%,$$

5）非产能营养素达到摄入量

$$\sum_{q=1}^{3}\sum_{i=1}^{n}\sum_{j=1}^{m} x_{qi}m_j v_k ij > \beta_k,\ \ k = 1\ldots 7,$$

其中β_k为第k类每日膳食非产能主要营养素参考摄入量。

6）餐次比尽可能满足早餐25%-35%，中餐、晚餐各30%-40%

$$25\% < \frac{\sum_{i=1}^{n1}\sum_{j=1}^{m}\sum_{k=1}^{5} x_{1i}m_j a_k f_{kij}}{cal} < 35\%,$$
$$30\% < \frac{\sum_{i=1}^{n2}\sum_{j=1}^{m}\sum_{k=1}^{5} x_{2i}m_j a_k f_{kij}}{cal} < 40\%,$$
$$30\% < \frac{\sum_{i=1}^{n3}\sum_{j=1}^{m}\sum_{k=1}^{5} x_{3i}m_j a_k f_{kij}}{cal} < 40\%.$$

7）食物份数为正整数

综上，我们建立线性规划模型为：

$$
\begin{cases}
x_{qi} \geq 0, \\
y = \max \mathrm{AAS} = \max \dfrac{\sum_{q=1}^{3} \sum_{i=1}^{n} \sum_{j=1}^{m} x_{qi} m_j a_{kij}}{\sum_{q=1}^{3} \sum_{i=1}^{n} \sum_{j=1}^{m} \sum_{k=1}^{8} x_{qi} m_j a_{kij}}, \\
0.9z < cal = \sum_{q=1}^{3} \sum_{i=1}^{n} \sum_{j=1}^{m} x_{qi} m_j a_k f_{kij} < 1.1z, \\
10\% < \dfrac{\sum_{q=1}^{3} \sum_{i=1}^{n} \sum_{j=1}^{m} \sum_{k=1}^{1} x_{qi} m_j a_k f_{kij}}{cal} < 15\%, \\
20\% < \dfrac{\sum_{q=1}^{3} \sum_{i=1}^{n} \sum_{j=1}^{m} \sum_{k=2}^{2} x_{qi} m_j a_k f_{kij}}{cal} < 30\%, \\
50\% < \dfrac{\sum_{q=1}^{3} \sum_{i=1}^{n} \sum_{j=1}^{m} \sum_{k=3}^{3} x_{qi} m_j a_k f_{kij}}{cal} < 65\%, \\
\sum_{q=1}^{3} \sum_{i=1}^{n} \sum_{j=1}^{m} x_{qi} m_j v_k ij > \beta_k, \ \ k = 1 \ldots 7, \\
25\% < \dfrac{\sum_{i=1}^{n1} \sum_{j=1}^{m} \sum_{k=1}^{5} x_{1i} m_j a_k f_{kij}}{cal} < 35\%, \\
30\% < \dfrac{\sum_{i=1}^{n2} \sum_{j=1}^{m} \sum_{k=1}^{5} x_{2i} m_j a_k f_{kij}}{cal} < 40\%, \\
30\% < \dfrac{\sum_{i=1}^{n3} \sum_{j=1}^{m} \sum_{k=1}^{5} x_{3i} m_j a_k f_{kij}}{cal} < 40\%, \\
x_{qi} \geq 0.
\end{cases}
$$

6.2.2 求解

遗传算法的基本原理源于达尔文的自然选择和遗传学的基本概念。在遗传算法中，解决方案的每个实例被视为一个“个体”，整个解决方案空间形成一个“种群”。每个个体通过一串“基因”来表示，这些基因编码了解决方案的具体参数。遗传算法通过迭代过程，不断改进种群的质量，逼近最优解。遗传算法通过模拟自然选择和遗传学原理，能够在复杂的搜索空间中有效地找到问题的近似最优解。

利用遗传算法对该模型就行求解，结果如下：

男生日食谱：

早餐：牛奶 200g、酸奶 125g、大米粥 15g、煮鸡蛋 50g、蒸地瓜 100g

中餐：大米饭 25g、馄饨 40g、水饺 50g、拌海带丝 100g、苹果 100g

晚餐：馒头 50g、鸡排面 50g、馅饼 25g、拌木耳 100g、橙子 100g

女生日食谱：

早餐：豆浆 10g、大米粥 15g、小米粥 15g、馒头 50g、水煎包 20g、苹果 100g

中餐：大米饭 25g、馄饨 40g、水饺 50g、拌豆腐 150g、葡萄 100g

晚餐：馒头 50g、馅饼 25g、拌干豆腐 100g、拌土豆丝 100g、拌芹菜花生米 100g

6.3 第二小问 以用餐费用最经济为目标建立优化模型

6.3.1 建模

自变量：第i个食品的早中晚的份数$x_{1i}x_{2i}x_{3i}$

目标函数：用餐费用最少

$$\min y=\sum_{q=1}^{3}\sum_{i=1}^{n}C_i x_{qi}$$

约束条件：

在第一问约束条件的基础上添加约束条件：

蛋白质氨基酸评分必需氨基酸评分（AAS）合理

$$\mathrm{AAS}=\min\frac{\sum_{q=1}^{3}\sum_{i=1}^{n}\sum_{j=1}^{m}x_{qi}m_j a_{kij}}{\sum_{q=1}^{3}\sum_{i=1}^{n}\sum_{j=1}^{m}\sum_{k=1}^{8}x_{qi}m_j a_{kij}}>90,\ k=1,\dots,8$$

综上，我们建立的线性规划模型为：

$$
\left\{
\begin{array}{r}
\min y=\displaystyle\sum_{q=1}^{3}\sum_{i=1}^{n}C_i x_{qi},\\
i>12,\\
0.9z<cal=\displaystyle\sum_{q=1}^{3}\sum_{i=1}^{n}\sum_{j=1}^{m}x_{qi}m_ja_kf_{kij}<1.1z,\\
10\%<\dfrac{\sum_{q=1}^{3}\sum_{i=1}^{n}\sum_{j=1}^{m}\sum_{k=1}^{1}x_{qi}m_ja_kf_{kij}}{cal}<15\%,\\
20\%<\dfrac{\sum_{q=1}^{3}\sum_{i=1}^{n}\sum_{j=1}^{m}\sum_{k=2}^{2}x_{qi}m_ja_kf_{kij}}{cal}<30\%,\\
50\%<\dfrac{\sum_{q=1}^{3}\sum_{i=1}^{n}\sum_{j=1}^{m}\sum_{k=3}^{3}x_{qi}m_ja_kf_{kij}}{cal}<65\%,\\
\displaystyle\sum_{q=1}^{3}\sum_{i=1}^{n}\sum_{j=1}^{m}x_{qi}m_jv_kij>\beta_k,\ \ k=1,\ldots,7,\\
25\%<\dfrac{\sum_{i=1}^{n1}\sum_{j=1}^{m}\sum_{k=1}^{5}x_{1i}m_ja_kf_{kij}}{cal}<35\%,\\
30\%<\dfrac{\sum_{i=1}^{n2}\sum_{j=1}^{m}\sum_{k=1}^{5}x_{2i}m_ja_kf_{kij}}{cal}<40\%,\\
30\%<\dfrac{\sum_{i=1}^{n3}\sum_{j=1}^{m}\sum_{k=1}^{5}x_{3i}m_ja_kf_{kij}}{cal}<40\%,\\
x_{qi}\geq 0,\\
\min\dfrac{\sum_{q=1}^{3}\sum_{i=1}^{n}\sum_{j=1}^{m}x_{qi}m_ja_{kij}}{\sum_{q=1}^{3}\sum_{i=1}^{n}\sum_{j=1}^{m}\sum_{k=1}^{8}x_{qi}m_ja_{kij}}>90,\ k=1,\ldots,8
\end{array}
\right.
$$

6.3.2 求解

利用遗传算法最终求得结果如下：

男生日食谱：

早餐：豆浆 10g、大米粥 15g、馒头 50g、煮鸡蛋 50g

中餐：大米饭 25g、馄饨 40g、拌海带丝 100g

晚餐：馒头 50g、馅饼 25g、拌豆腐 150g

女生日食谱：

早餐：豆浆 10g、大米粥 15g、馒头 50g、水煎包 20g

中餐：大米饭 25g、馄饨 40g、拌豆腐 150g

晚餐：馒头 50g、馅饼 25g、拌干豆腐 100g

6.4 第三小问. 兼顾蛋白质氨基酸评分及经济性

6.4.1 建模

该小问需要同时考虑氨基酸评分和费用的目标，因此为多目标优化问题，一个较为简单的方法是将两个目标标准化后相加得到新的目标函数更建议使用多目标优化的模型进行求解。此时的目标函数：

$$\begin{cases} \min y = \sum_{q=1}^{3}\sum_{i=1}^{n} C_i x_{qi}, \\ \max\min \dfrac{\sum_{q=1}^{3}\sum_{i=1}^{n}\sum_{j=1}^{m} x_{qi} m_j a_{kij}}{\sum_{q=1}^{3}\sum_{i=1}^{n}\sum_{j=1}^{m}\sum_{k=1}^{8} x_{qi} m_j a_{kij}},\ k=1,\ldots,8 \end{cases}$$

约束条件与第一小问一致，综上，我们建立的线性规划模型为：

$$
\left\{
\begin{aligned}
&\min y=\sum_{q=1}^{3}\sum_{i=1}^{n}C_i x_{qi},\\
&\max\min\frac{\sum_{q=1}^{3}\sum_{i=1}^{n}\sum_{j=1}^{m}x_{qi}m_j a_{kij}}{\sum_{q=1}^{3}\sum_{i=1}^{n}\sum_{j=1}^{m}\sum_{k=1}^{8}x_{qi}m_j a_{kij}},\ k=1,\ldots,8\\
&i>12,\\
&0.9z<cal=\sum_{q=1}^{3}\sum_{i=1}^{n}\sum_{j=1}^{m}x_{qi}m_j a_k f_{kij}<1.1z,\\
&10\%<\frac{\sum_{q=1}^{3}\sum_{i=1}^{n}\sum_{j=1}^{m}\sum_{k=1}^{1}x_{qi}m_j a_k f_{kij}}{cal}<15\%,\\
&20\%<\frac{\sum_{q=1}^{3}\sum_{i=1}^{n}\sum_{j=1}^{m}\sum_{k=2}^{2}x_{qi}m_j a_k f_{kij}}{cal}<30\%,\\
&50\%<\frac{\sum_{q=1}^{3}\sum_{i=1}^{n}\sum_{j=1}^{m}\sum_{k=3}^{3}x_{qi}m_j a_k f_{kij}}{cal}<65\%,\\
&\sum_{q=1}^{3}\sum_{i=1}^{n}\sum_{j=1}^{m}x_{qi}m_j v_{kij}>\beta_k,\ \ k=1,\ldots,7,\\
&25\%<\frac{\sum_{i=1}^{n1}\sum_{j=1}^{m}\sum_{k=1}^{5}x_{1i}m_j a_k f_{kij}}{cal}<35\%,\\
&30\%<\frac{\sum_{i=1}^{n2}\sum_{j=1}^{m}\sum_{k=1}^{5}x_{2i}m_j a_k f_{kij}}{cal}<40\%,\\
&30\%<\frac{\sum_{i=1}^{n3}\sum_{j=1}^{m}\sum_{k=1}^{5}x_{3i}m_j a_k f_{kij}}{cal}<40\%,\\
&x_{qi}\geq 0,
\end{aligned}
\right.
$$

6.4.2 求解

求解结果如下：

男生日食谱：

早餐：牛奶 200g、大米粥 15g、煮鸡蛋 50g

中餐：大米饭 25g、馄饨 40g、拌豆腐 150g

晚餐：馒头 50g、馅饼 25g、拌豆腐 150g

女生日食谱：

早餐：豆浆 10g、酸奶 125g、大米粥 15g、水煎包 20g

中餐：大米饭 25g、馄饨 40g、拌豆腐 150g

晚餐：馒头 50g、馅饼 25g、拌豆腐 100g

七、 问题三模型建立与求解

7.1 问题三求解思路

针对问题三，基于附件三的周膳食食堂食谱进行优化设计，自变量为每日每餐每种食物的份数，兼具根据蛋白质氨基酸评分最大、以用餐费用最经济为目标，约束条件为满足附件四的各个评价要求，建立单目标优化模型，分别根据蛋白质氨基酸评分最大和以用餐费用最经济为目标，约束条件为满足附件四的各个评价要求，对于每周食谱建立多目标优化模型。

7.2 以蛋白质氨基酸评分最大为目标建立优化模型

7.2.1 建模

自变量：第p天第i个食品的早中晚的份数x_{pqi}, $p=1\ldots7,\ q=1\ldots3$

目标函数：

$$y=\max\mathrm{AAS}=\max\min\frac{\sum_{p=1}^{7}\sum_{q=1}^{3}\sum_{i=1}^{n}\sum_{j=1}^{m}x_{qi}m_ja_{kij}}{\sum_{p=1}^{7}\sum_{q=1}^{3}\sum_{i=1}^{n}\sum_{j=1}^{m}\sum_{k=1}^{8}x_{qi}m_ja_{kij}},\ k=1,\ldots,8$$

约束条件：

1）每天摄入食物种类>12种：$i>12$

2）每天摄入食物包含五大类

3）每周摄入食物种类数>25种：

$$\sum_{i=1}^{n}\sum_{p=1}^{7}\sum_{q=1}^{3}\{x_{pqi}>0\}>25$$

4）男、女生每日能量实际摄入量与摄入量目标相差在±10%之内

摄入总能量摄入目标为：女生1900kcal/d，男生2400kcal/d：

设摄入目标为z

$$0.9z<cal_p=\sum_{q=1}^{3}\sum_{i=1}^{n}\sum_{j=1}^{m}\sum_{k=1}^{5}x_{pqi}m_ja_kf_{kij}<1.1z,\ p=1,\ldots,7$$

其中a_k为能量转换系数

宏量营养素供能占比

蛋白质10%-15%、脂肪20%-30%、碳水化合物50%-65%

$$10\% < \frac{\sum_{q=1}^{3}\sum_{i=1}^{n}\sum_{j=1}^{m}\sum_{k=1}^{1} x_{pqi}m_j a_k f_{kij}}{cal} < 15\%,\ p=1,\ldots,7$$

$$20\% < \frac{\sum_{q=1}^{3}\sum_{i=1}^{n}\sum_{j=1}^{m}\sum_{k=2}^{2} x_{pqi}m_j a_k f_{kij}}{cal} < 30\%,\ p=1,\ldots,7$$

$$50\% < \frac{\sum_{q=1}^{3}\sum_{i=1}^{n}\sum_{j=1}^{m}\sum_{k=3}^{3} x_{pqi}m_j a_k f_{kij}}{cal} < 65\%,\ p=1,\ldots,7$$

非产能营养素达到摄入量

$$\sum_{q=1}^{3}\sum_{i=1}^{n}\sum_{j=1}^{m} x_{pqi}m_j v_{kij} > \beta_k,\ \ k=1,\ldots,7,\ p=1,\ldots,7$$

其中β_k为第k类每日膳食非产能主要营养素参考摄入量

7) 餐次比尽可能满足早餐25%-35%，中餐、晚餐各30%-40%

$$25\% < \frac{\sum_{i=1}^{n1}\sum_{j=1}^{m}\sum_{k=1}^{5} x_{p1i}m_j a_k f_{kij}}{cal} < 35\%,\ p=1,\ldots,7$$

$$30\% < \frac{\sum_{i=1}^{n2}\sum_{j=1}^{m}\sum_{k=1}^{5} x_{p2i}m_j a_k f_{kij}}{cal} < 40\%,\ p=1,\ldots,7$$

$$30\% < \frac{\sum_{i=1}^{n3}\sum_{j=1}^{m}\sum_{k=1}^{5} x_{p3i}m_j a_k f_{kij}}{cal} < 40\%,\ p=1,\ldots,7$$

其中x_{p1i}为早餐食物，x_{p2i}为午餐食物，x_{p3i}为晚餐食物

8) 食物份数为正整数

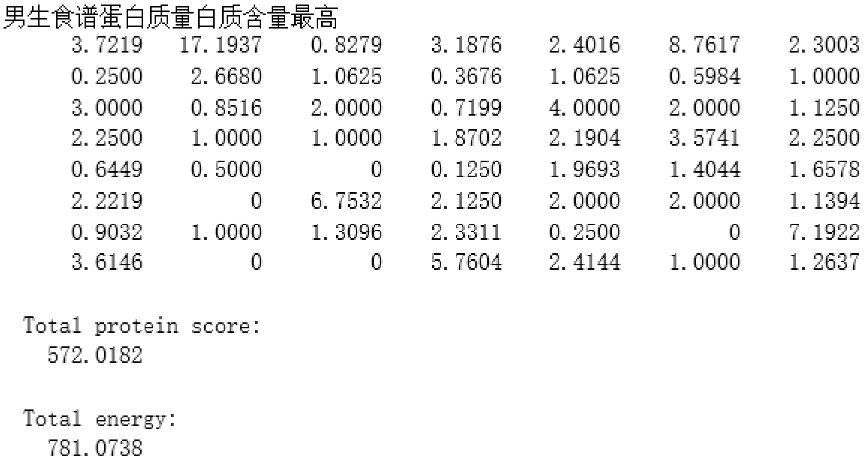
$$x_{pqi} \geq 0$$

综上，我们建立的单目标优化模型为

$$
\begin{cases}
\max\min \dfrac{\sum_{p=1}^{7}\sum_{q=1}^{3}\sum_{i=1}^{n}\sum_{j=1}^{m} x_{qi}m_j a_{kij}}{\sum_{p=1}^{7}\sum_{q=1}^{3}\sum_{i=1}^{n}\sum_{j=1}^{m}\sum_{k=1}^{8} x_{qi}m_j a_{kij}},\ k=1,\ldots,8,\\
\displaystyle\sum_{i=1}^{n}\sum_{p=1}^{7}\sum_{q=1}^{3}\{x_{pqi}>0\}>25,\\
0.9z < cal_p = \displaystyle\sum_{q=1}^{3}\sum_{i=1}^{n}\sum_{j=1}^{m}\sum_{k=1}^{5} x_{pqi}m_j a_k f_{kij} < 1.1z,\ p=1,\ldots,7,\\
i>12,\\
10\% < \dfrac{\sum_{q=1}^{3}\sum_{i=1}^{n}\sum_{j=1}^{m}\sum_{k=1}^{1} x_{pqi}m_j a_k f_{kij}}{cal} < 15\%,\ p=1,\ldots,7,\\
20\% < \dfrac{\sum_{q=1}^{3}\sum_{i=1}^{n}\sum_{j=1}^{m}\sum_{k=2}^{2} x_{pqi}m_j a_k f_{kij}}{cal} < 30\%,\ p=1,\ldots,7,\\
50\% < \dfrac{\sum_{q=1}^{3}\sum_{i=1}^{n}\sum_{j=1}^{m}\sum_{k=3}^{3} x_{pqi}m_j a_k f_{kij}}{cal} < 65\%,\ p=1,\ldots,7,\\
\displaystyle\sum_{q=1}^{3}\sum_{i=1}^{n}\sum_{j=1}^{m} x_{pqi}m_j v_{kij} > \beta_k,\ \ k=1,\ldots,7,\ p=1,\ldots,7,\\
25\% < \dfrac{\sum_{i=1}^{n1}\sum_{j=1}^{m}\sum_{k=1}^{5} x_{p1i}m_j a_k f_{kij}}{cal} < 35\%,\ p=1,\ldots,7,\\
30\% < \dfrac{\sum_{i=1}^{n2}\sum_{j=1}^{m}\sum_{k=1}^{5} x_{p2i}m_j a_k f_{kij}}{cal} < 40\%,\ p=1,\ldots,7,\\
30\% < \dfrac{\sum_{i=1}^{n3}\sum_{j=1}^{m}\sum_{k=1}^{5} x_{p3i}m_j a_k f_{kij}}{cal} < 40\%,\ p=1,\ldots,7,\\
x_{pqi}\geq 0
\end{cases}
$$

7.2.2 求解

利用遗传算法最终求得如图1和图2所示

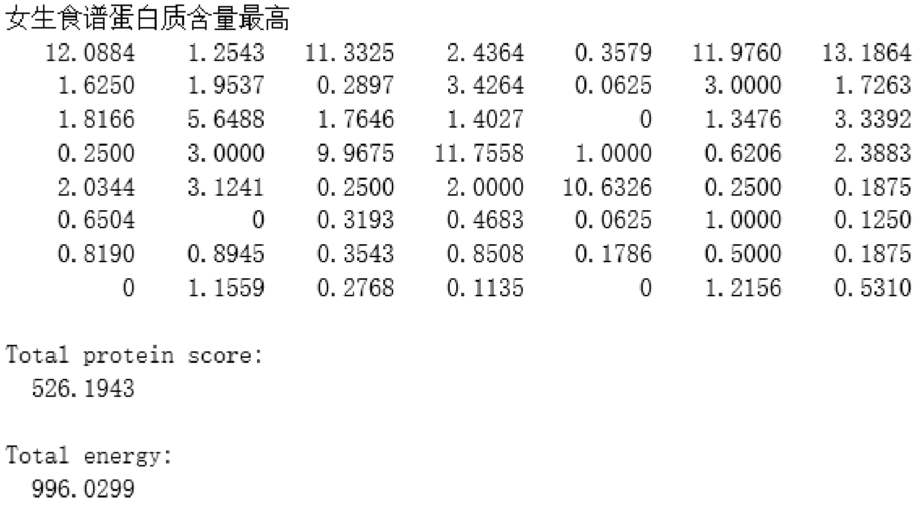


```
男生食谱蛋白质量白质含量最高
      3.7219   17.1937    0.8279    3.1876    2.4016    8.7617    2.3003
      0.2500    2.6680    1.0625    0.3676    1.0625    0.5984    1.0000
      3.0000    0.8516    2.0000    0.7199    4.0000    2.0000    1.1250
      2.2500    1.0000    1.0000    1.8702    2.1904    3.5741    2.2500
      0.6449    0.5000         0    0.1250    1.9693    1.4044    1.6578
      2.2219         0    6.7532    2.1250    2.0000    2.0000    1.1394
      0.9032    1.0000    1.3096    2.3311    0.2500         0    7.1922
      3.6146         0         0    5.7604    2.4144    1.0000    1.2637

  Total protein score:
    572.0182

  Total energy:
    781.0738
```

图1 男生食谱蛋白质含量最高



```
女生食谱蛋白质含量最高
   12.0884    1.2543   11.3325    2.4364    0.3579   11.9760   13.1864
    1.6250    1.9537    0.2897    3.4264    0.0625    3.0000    1.7263
    1.8166    5.6488    1.7646    1.4027         0    1.3476    3.3392
    0.2500    3.0000    9.9675   11.7558    1.0000    0.6206    2.3883
    2.0344    3.1241    0.2500    2.0000   10.6326    0.2500    0.1875
    0.6504         0    0.3193    0.4683    0.0625    1.0000    0.1250
    0.8190    0.8945    0.3543    0.8508    0.1786    0.5000    0.1875
         0    1.1559    0.2768    0.1135         0    1.2156    0.5310

Total protein score:
  526.1943

Total energy:
  996.0299
```

图2 女生食谱蛋白质含量最高

7.3　以用餐费用最经济为目标建立优化模型

7.3.1 建模

自变量：第p天第i个食品的早中晚的份数x_{pqi}, $p=1\ldots7,\ q=1\ldots3$

目标函数：

$$\min y=\sum_{p=1}^{7}\sum_{q=1}^{3}\sum_{i=1}^{n}C_i x_{pqi}$$

约束条件：

在第一问约束条件的基础上添加约束条件：

蛋白质氨基酸评分必需氨基酸评分（AAS）合理

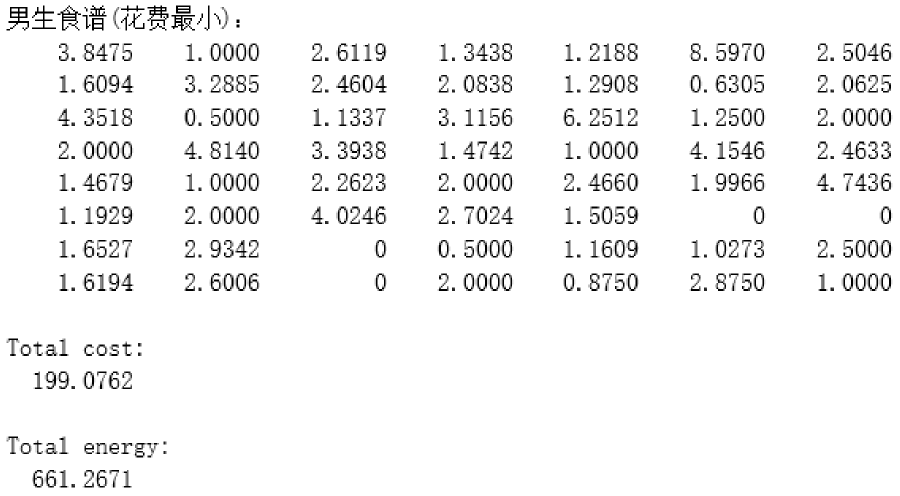
$$\mathrm{AAS}=\min\frac{\sum_{p=1}^{7}\sum_{q=1}^{3}\sum_{i=1}^{n}\sum_{j=1}^{m}x_{pqi}m_j a_{kij}}{\sum_{p=1}^{7}\sum_{q=1}^{3}\sum_{i=1}^{n}\sum_{j=1}^{m}\sum_{k=1}^{8}x_{pqi}m_j a_{kij}}>90,\ k=1,\ldots,8$$

综上，我们建立的线性规划模型为：

$$
\begin{cases}
\min y = \sum_{p=1}^{7} \sum_{q=1}^{3} \sum_{i=1}^{n} C_i x_{pqi}, \\
\sum_{i=1}^{n} \sum_{p=1}^{7} \sum_{q=1}^{3} \{x_{pqi} > 0\} > 25, \\
i > 12, \\
0.9z < cal_p = \sum_{q=1}^{3} \sum_{i=1}^{n} \sum_{j=1}^{m} \sum_{k=1}^{5} x_{pqi} m_j a_k f_{kij} < 1.1z,\ p = 1, \ldots, 7, \\
10\% < \dfrac{\sum_{q=1}^{3} \sum_{i=1}^{n} \sum_{j=1}^{m} \sum_{k=1}^{1} x_{pqi} m_j a_k f_{kij}}{cal} < 15\%,\ p = 1, \ldots, 7, \\
20\% < \dfrac{\sum_{q=1}^{3} \sum_{i=1}^{n} \sum_{j=1}^{m} \sum_{k=2}^{2} x_{pqi} m_j a_k f_{kij}}{cal} < 30\%,\ p = 1, \ldots, 7, \\
50\% < \dfrac{\sum_{q=1}^{3} \sum_{i=1}^{n} \sum_{j=1}^{m} \sum_{k=3}^{3} x_{pqi} m_j a_k f_{kij}}{cal} < 65\%,\ p = 1, \ldots, 7, \\
\sum_{q=1}^{3} \sum_{i=1}^{n} \sum_{j=1}^{m} x_{pqi} m_j v_{kij} > \beta_k,\ \ k = 1, \ldots, 7,\ p = 1, \ldots, 7, \\
25\% < \dfrac{\sum_{i=1}^{n1} \sum_{j=1}^{m} \sum_{k=1}^{5} x_{p1i} m_j a_k f_{kij}}{cal} < 35\%,\ p = 1, \ldots, 7 \\
30\% < \dfrac{\sum_{i=1}^{n2} \sum_{j=1}^{m} \sum_{k=1}^{5} x_{p2i} m_j a_k f_{kij}}{cal} < 40\%,\ p = 1, \ldots, 7, \\
30\% < \dfrac{\sum_{i=1}^{n3} \sum_{j=1}^{m} \sum_{k=1}^{5} x_{p3i} m_j a_k f_{kij}}{cal} < 40\%,\ p = 1, \ldots, 7, \\
x_{pqi} \geq 0
\end{cases}
$$

7.3.2 求解

利用遗传算法最终求得如图3和图4所示

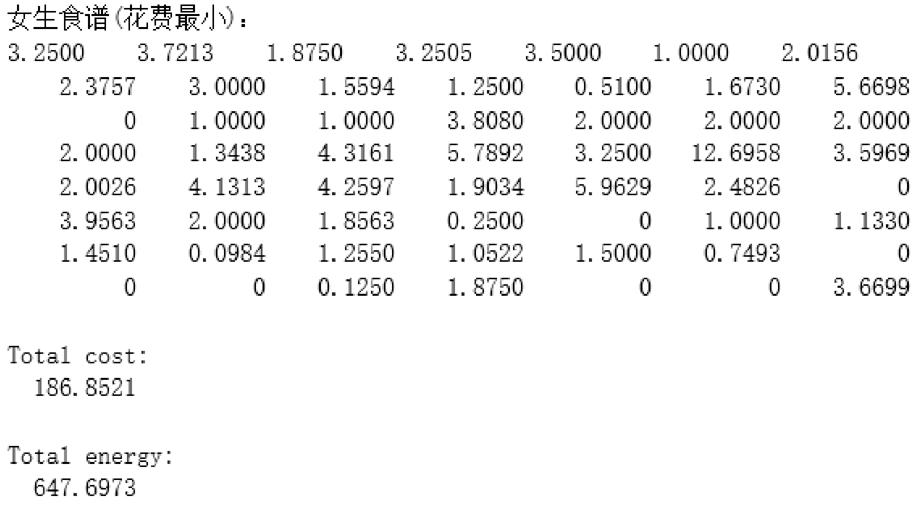


```
男生食谱(花费最小)：
    3.8475    1.0000    2.6119    1.3438    1.2188    8.5970    2.5046
    1.6094    3.2885    2.4604    2.0838    1.2908    0.6305    2.0625
    4.3518    0.5000    1.1337    3.1156    6.2512    1.2500    2.0000
    2.0000    4.8140    3.3938    1.4742    1.0000    4.1546    2.4633
    1.4679    1.0000    2.2623    2.0000    2.4660    1.9966    4.7436
    1.1929    2.0000    4.0246    2.7024    1.5059         0         0
    1.6527    2.9342         0    0.5000    1.1609    1.0273    2.5000
    1.6194    2.6006         0    2.0000    0.8750    2.8750    1.0000

Total cost:
  199.0762

Total energy:
  661.2671
```

图3 男生食谱（花费最小）



```
女生食谱(花费最小):
3.2500    3.7213    1.8750    3.2505    3.5000    1.0000    2.0156
    2.3757    3.0000    1.5594    1.2500    0.5100    1.6730    5.6698
         0    1.0000    1.0000    3.8080    2.0000    2.0000    2.0000
    2.0000    1.3438    4.3161    5.7892    3.2500   12.6958    3.5969
    2.0026    4.1313    4.2597    1.9034    5.9629    2.4826         0
    3.9563    2.0000    1.8563    0.2500         0    1.0000    1.1330
    1.4510    0.0984    1.2550    1.0522    1.5000    0.7493         0
         0         0    0.1250    1.8750         0         0    3.6699

Total cost:
  186.8521

Total energy:
  647.6973
```

图4 女生食谱（花费最小）

7.4 兼顾蛋白质氨基酸评分及经济性

7.4.1 建模

该小问需要同时考虑氨基酸评分和费用的目标，因此为多目标优化问题，一个较为简单的方法是将两个目标标准化后相加得到新的目标函数更建议使用多目标优化的模型进行求解。此时的目标函数：

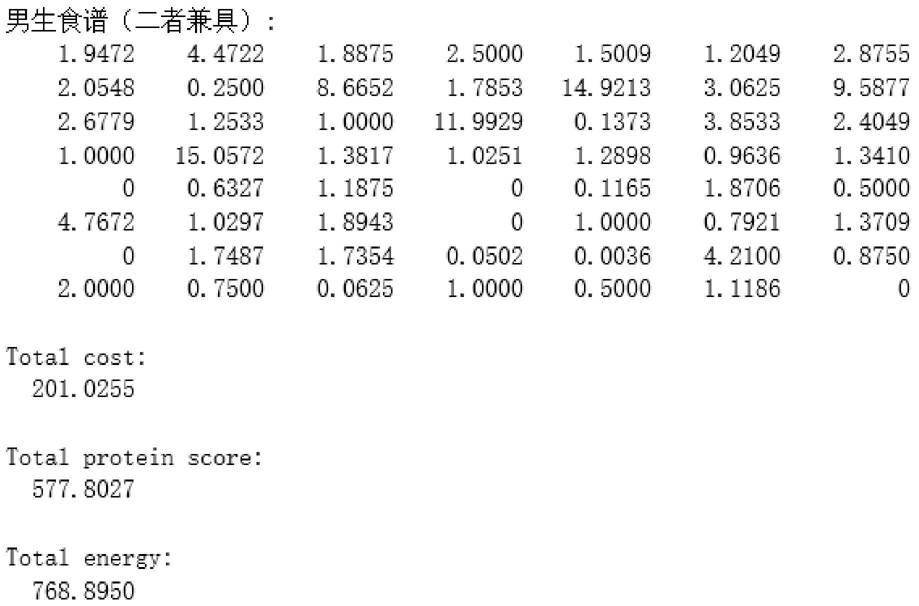
$$\begin{cases} \min y = \sum_{p=1}^{7} \sum_{q=1}^{3} \sum_{i=1}^{n} C_i x_{qi}, \\ \max \min \dfrac{\sum_{p=1}^{7} \sum_{q=1}^{3} \sum_{i=1}^{n} \sum_{j=1}^{m} x_{qi} m_j a_{kij}}{\sum_{p=1}^{7} \sum_{q=1}^{3} \sum_{i=1}^{n} \sum_{j=1}^{m} \sum_{k=1}^{8} x_{qi} m_j a_{kij}}, \ k = 1, \ldots, 8 \end{cases}$$

约束条件：与第一小问一致。

综上，我们建立的线性规划模型为：

$$
\begin{cases}
\min y = \sum_{p=1}^{7}\sum_{q=1}^{3}\sum_{i=1}^{n} C_i x_{pqi}, \\
\max\min \frac{\sum_{p=1}^{7}\sum_{q=1}^{3}\sum_{i=1}^{n}\sum_{j=1}^{m} x_{qi} m_j a_{kij}}{\sum_{p=1}^{7}\sum_{q=1}^{3}\sum_{i=1}^{n}\sum_{j=1}^{m}\sum_{k=1}^{8} x_{qi} m_j a_{kij}},\ k = 1,\ldots,8, \\
\sum_{i=1}^{n}\sum_{p=1}^{7}\sum_{q=1}^{3}\{x_{pqi} > 0\} > 25, \\
0.9z < cal_p = \sum_{q=1}^{3}\sum_{i=1}^{n}\sum_{j=1}^{m}\sum_{k=1}^{5} x_{pqi} m_j a_k f_{kij} < 1.1z,\ p = 1,\ldots,7, \\
i > 12, \\
10\% < \frac{\sum_{q=1}^{3}\sum_{i=1}^{n}\sum_{j=1}^{m}\sum_{k=1}^{1} x_{pqi} m_j a_k f_{kij}}{cal} < 15\%,\ p = 1,\ldots,7, \\
20\% < \frac{\sum_{q=1}^{3}\sum_{i=1}^{n}\sum_{j=1}^{m}\sum_{k=2}^{2} x_{pqi} m_j a_k f_{kij}}{cal} < 30\%,\ p = 1,\ldots,7, \\
50\% < \frac{\sum_{q=1}^{3}\sum_{i=1}^{n}\sum_{j=1}^{m}\sum_{k=3}^{3} x_{pqi} m_j a_k f_{kij}}{cal} < 65\%,\ p = 1,\ldots,7, \\
\sum_{q=1}^{3}\sum_{i=1}^{n}\sum_{j=1}^{m} x_{pqi} m_j v_{kij} > \beta_k,\ \ k = 1,\ldots,7,\ p = 1,\ldots,7, \\
25\% < \frac{\sum_{i=1}^{n1}\sum_{j=1}^{m}\sum_{k=1}^{5} x_{p1i} m_j a_k f_{kij}}{cal} < 35\%,\ p = 1,\ldots,7 \\
30\% < \frac{\sum_{i=1}^{n2}\sum_{j=1}^{m}\sum_{k=1}^{5} x_{p2i} m_j a_k f_{kij}}{cal} < 40\%,\ p = 1,\ldots,7, \\
30\% < \frac{\sum_{i=1}^{n3}\sum_{j=1}^{m}\sum_{k=1}^{5} x_{p3i} m_j a_k f_{kij}}{cal} < 40\%,\ p = 1,\ldots,7, \\
x_{pqi} \geq 0
\end{cases}
$$

利用遗传算法最终求得如图5和图6所示



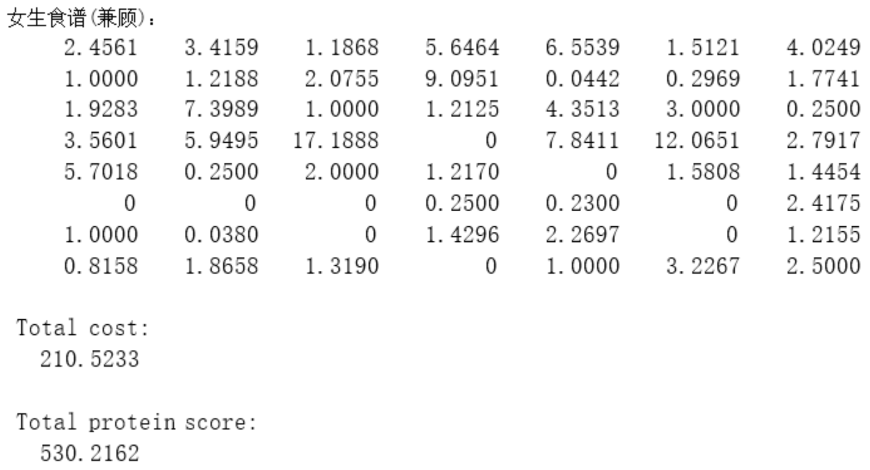
```
男生食谱（二者兼具）:
    1.9472    4.4722    1.8875    2.5000    1.5009    1.2049    2.8755
    2.0548    0.2500    8.6652    1.7853   14.9213    3.0625    9.5877
    2.6779    1.2533    1.0000   11.9929    0.1373    3.8533    2.4049
    1.0000   15.0572    1.3817    1.0251    1.2898    0.9636    1.3410
         0    0.6327    1.1875         0    0.1165    1.8706    0.5000
    4.7672    1.0297    1.8943         0    1.0000    0.7921    1.3709
         0    1.7487    1.7354    0.0502    0.0036    4.2100    0.8750
    2.0000    0.7500    0.0625    1.0000    0.5000    1.1186         0

Total cost:
  201.0255

Total protein score:
  577.8027

Total energy:
  768.8950
```

图5 男生食谱（二者兼具）



```
女生食谱(兼顾):
    2.4561    3.4159    1.1868    5.6464    6.5539    1.5121    4.0249
    1.0000    1.2188    2.0755    9.0951    0.0442    0.2969    1.7741
    1.9283    7.3989    1.0000    1.2125    4.3513    3.0000    0.2500
    3.5601    5.9495   17.1888         0    7.8411   12.0651    2.7917
    5.7018    0.2500    2.0000    1.2170         0    1.5808    1.4454
         0         0         0    0.2500    0.2300         0    2.4175
    1.0000    0.0380         0    1.4296    2.2697         0    1.2155
    0.8158    1.8658    1.3190         0    1.0000    3.2267    2.5000

 Total cost:
   210.5233

 Total protein score:
   530.2162
```

图6 女生食谱（二者兼具）

八、问题四

题目要求针对大学生饮食结构及习惯，写一份健康饮食、平衡膳食的倡议书

倡议主题：大学生健康饮食、平衡膳食

尊敬的大学师生：

根据我们对于大学生平衡膳食食谱的研究发现，大学生的饮食结构和习惯发生了显著变化。在这个过程中，我们应该意识到，健康的饮食结构和良好的饮食习惯对我们的身体健康和学习生活都至关重要。然而，现实中我们也发现了一些不容忽视的问题，比如偏食、挑食现象普遍，高热量、高脂肪食品摄入

过多，不吃早餐、暴饮暴食等不良习惯。这些不健康的饮食习惯可能会导致营养不良、肥胖等问题，影响我们的身心健康和学习效果。为了倡导大家树立健康饮食、平衡膳食的理念，我们发起这份倡议书，希望通过共同努力，营造一个健康、和谐的校园饮食环境。

倡议背景当前，大学生的饮食问题主要表现在以下几个方面：

1) 偏食、挑食现象普遍，营养不均衡；
2) 高热量、高脂肪食品食用过多；
3) 缺乏规律饮食，暴饮暴食、不吃早餐等现象普遍；
4) 缺乏运动，导致能量摄入与消耗不平衡。

这些问题不仅影响大学生的身体健康，还可能引发一系列慢性疾病，如肥胖、高血压、糖尿病等。因此，我们有必要倡导健康饮食、平衡膳食的理念，引导大学生养成良好的饮食习惯。

在此我们提出如下倡议：

1） 树立健康饮食理念：我们要认识到健康饮食对身体健康的重要性，树立“健康饮食、平衡膳食”的理念。在日常饮食中，应注重食物的多样性，合理搭配各类食物，确保摄入足够的营养物质。
2） 均衡膳食搭配：我们的研究发现，大学生的饮食极不均衡，在饮食中，应注重蛋白质、碳水化合物、脂肪、维生素和矿物质等营养素的均衡摄入。合理搭配主食、蔬菜、水果、肉类、豆类等食物，确保身体所需的各种营养素得到充分补充。
3） 规律饮食：养成良好的饮食习惯，保持规律的三餐时间。避免暴饮暴食、不吃早餐等不良习惯。同时，合理安排饮食量，避免过量摄入热量和脂肪。
4） 减少不健康食品摄入：尽量减少外卖、快餐等高热量、高脂肪食品的摄入。在选择食品时，应注重食品的营养价值和健康程度，尽量选择新鲜、健康、低热量的食品。
5） 增加运动量：我们研究发现大学生每日摄入的能量过多，积极参加体育锻炼和户外活动，增加身体运动量。通过运动消耗多余的热量和脂肪，保持身体健康和体态优美。

我们提出如下建议：

1）学校加强饮食健康教育：学校应加强对大学生的饮食健康教育，普及健康饮食、平衡膳食的知识和理念。通过举办讲座、开展宣传活动等方式，引导大学生树立健康饮食的意识。

2）食堂提供健康、营养的餐品：学校食堂应提供健康、营养的餐品，注重食物的品质和口感。同时，加强食堂卫生管理，确保食品的安全和卫生。

3）学生积极参与健康饮食活动：学生应积极参与学校组织的健康饮食活动，如营养餐评比、健康食谱分享等。通过参与活动，了解健康饮食的知识和技巧，培养良好的饮食习惯。

让我们共同倡导健康饮食、平衡膳食的理念，从自身做起，从小事做起，共同营造一个健康、和谐的校园饮食环境！谢谢大家！

九、模型评价与改进

9.1 模型的评价

（1）在问题一中，对问题进行分析，忽略了一些对问题结果影响较小的量，如一些微量元素的摄取等，使模型较简单化，有利于模型的建立与求解；

（2）在问题二、三中，利用分层序列法和主要目标法将多目标规划问题转化为单目标规划，有利于模型的求解；

（3）在用编程过程中采取增加约束的方法来缩小最优解的搜索范围，大大缩短了运行时间。

9.2 模型的改进

（1）在求解过程中忽略了一些对结果影响较小的量，若想模型更加精确，则必须重新考虑这些量，模型的细粒度将会大大提高，但是复杂度也会相应地上升；

（2）在求解周食谱模型的过程中，我们不仅可以讨论整周的膳食营养是否达标，还可以再对周食谱内的每日相隔的日食谱的关联关系进行更加细致的编排进行探讨。

（3）本次目标人群虽然主要针对的是大学生，但也可以很容易将模型迁移至其他人群，例如老龄、幼儿妇女等人群。

十、参考文献

[1]马莉:MATLAB数学实验与建模，清华大学出版社2010年版.

[2]束金龙、闻人凯，柴俊:线性规划理论与模型应用，科学出版社2007年版.

[3]中国食物成分表2018年（标准版）第6版第一册.网址：https://www.doc88.com/p-69816594797495.html

十一、附录

附录清单：

11.1问题1的程序

11.1.1 男生膳食指标评估

```
nutrient_content_per_100g = struct( ...
    'XiaomiPorridge_Millet', struct('Carbohydrate', 73.5, 'Protein', 9.0, 'Fat', 1.6), ...
    'Youtiao_WheatFlour', struct('Carbohydrate', 70.0, 'Protein', 10.5, 'Fat', 1.8), ...
    'Youtiao_SoybeanOil', struct('Fat', 99.9, 'Carbohydrate', 0.0, 'Protein', 0.0), ...
    'FriedEgg_Egg', struct('Carbohydrate', 2.5, 'Protein', 13.1, 'Fat', 10.6), ...
    'FriedEgg_SoybeanOil', struct('Fat', 99.9, 'Carbohydrate', 0.0, 'Protein', 0.0), ...
    'SeaweedSalad_Seaweed', struct('Carbohydrate', 10.6, 'Protein', 1.2, 'Fat', 0.1), ...
    'SeaweedSalad_SesameOil', struct('Fat', 99.7, 'Carbohydrate', 0.0, 'Protein', 0.0), ...
    'Rice_Rice', struct('Carbohydrate', 77.2, 'Protein', 7.4, 'Fat', 0.6), ...
    'BlackFungusSalad_BlackFungus', struct('Carbohydrate', 6.0, 'Protein', 1.5, 'Fat', 0.2), ...
    'BlackFungusSalad_SesameOil', struct('Fat', 99.7, 'Carbohydrate', 0.0, 'Protein', 0.0), ...
    'DiSanXian_Eggplant', struct('Carbohydrate', 5.0, 'Protein', 1.0, 'Fat', 0.2), ...
    'DiSanXian_Potato', struct('Carbohydrate', 18.4, 'Protein', 2.0, 'Fat', 0.2), ...
    'DiSanXian_GreenPepper', struct('Carbohydrate', 5.4, 'Protein', 1.4, 'Fat', 0.3), ...
    'DiSanXian_SoybeanOil', struct('Fat', 99.9, 'Carbohydrate', 0.0, 'Protein', 0.0), ...
    'BraisedPorkBelly_PorkBelly', struct('Carbohydrate', 0.0, 'Protein', 20.6, 'Fat', 39.8), ...
    'BraisedPorkBelly_DriedTofu', struct('Carbohydrate', 10.0, 'Protein', 44.6, 'Fat',
18.0), ...
    'BraisedPorkBelly_SoybeanOil', struct('Fat', 99.9, 'Carbohydrate', 0.0, 'Protein', 0.0), ...
    'HotPotNoodles_CornFlour', struct('Carbohydrate', 74.0, 'Protein', 8.5, 'Fat', 1.5), ...
    'HotPotNoodles_Cabbage', struct('Carbohydrate', 3.6, 'Protein', 1.5, 'Fat', 0.2), ...
    'HotPotNoodles_BokChoy', struct('Carbohydrate', 3.8, 'Protein', 1.8, 'Fat', 0.4), ...
    'HotPotNoodles_DriedTofu', struct('Carbohydrate', 10.0, 'Protein', 44.6, 'Fat', 18.0), ...
    'HotPotNoodles_SoybeanOil', struct('Fat', 99.9, 'Carbohydrate', 0.0, 'Protein', 0.0), ...
    'Baozi_WheatFlour', struct('Carbohydrate', 70.0, 'Protein', 10.5, 'Fat', 1.8), ...
    'Baozi_Pork', struct('Carbohydrate', 0.0, 'Protein', 20.6, 'Fat', 39.8), ...
    'Baozi_Sauerkraut', struct('Carbohydrate', 5.0, 'Protein', 1.1, 'Fat', 0.2), ...
    'Baozi_SoybeanOil', struct('Fat', 99.9, 'Carbohydrate', 0.0, 'Protein', 0.0), ...
    'FriedChicken', struct('Carbohydrate', 0.0, 'Protein', 25.0, 'Fat', 20.0) ...
);

recipe = struct( ...
    'Breakfast', struct( ...
        'XiaomiPorridge_Millet', [15, 1], ...
        'Youtiao_WheatFlour', [50*2, 2], ...
        'Youtiao_SoybeanOil', [10*2, 2], ...
        'FriedEgg_Egg', [50, 1], ...
        'FriedEgg_SoybeanOil', [10, 1], ...
        'SeaweedSalad_Seaweed', [100, 1], ...
        'SeaweedSalad_SesameOil', [2, 1] ...
    ), ...
    'Lunch', struct( ...
        'Rice_Rice', [25*4, 4], ...
        'BlackFungusSalad_BlackFungus', [100, 1], ...
        'BlackFungusSalad_SesameOil', [2, 1], ...
        'DiSanXian_Eggplant', [80, 1], ...
        'DiSanXian_Potato', [80, 1], ...
        'DiSanXian_GreenPepper', [10, 1], ...
        'DiSanXian_SoybeanOil', [10, 1], ...
        'BraisedPorkBelly_PorkBelly', [50, 1], ...
        'BraisedPorkBelly_DriedTofu', [50, 1], ...
        'BraisedPorkBelly_SoybeanOil', [10, 1] ...
    ), ...
    'Dinner', struct( ...
        'HotPotNoodles_CornFlour', [80, 1], ...
        'HotPotNoodles_Cabbage', [20, 1], ...
        'HotPotNoodles_BokChoy', [20, 1], ...
        'HotPotNoodles_DriedTofu', [10, 1], ...
        'HotPotNoodles_SoybeanOil', [10, 1], ...
        'Baozi_WheatFlour', [25, 1], ...
        'Baozi_Pork', [15, 1], ...
        'Baozi_Sauerkraut', [20, 1], ...
        'Baozi_SoybeanOil', [5, 1], ...
        'FriedChicken', [100, 1] ...
    ) ...
);

total_energy = 0;
energy_from_protein = 0;
energy_from_fat = 0;
energy_from_carbs = 0;

energy_breakfast = 0;
```

```
energy_lunch = 0;
energy_dinner = 0;

meals = fieldnames(recipe);

for i = 1:numel(meals)
    meal = meals{i};
    foods = recipe.(meal);
    food_names = fieldnames(foods);
    meal_energy = 0;

    for j = 1:numel(food_names)
        food = food_names{j};
        amount = foods.(food)(1);
        servings = foods.(food)(2);
        total_amount = amount * servings;

        nutrients = fieldnames(nutrient_content_per_100g.(food));
        for k = 1:numel(nutrients)
            nutrient = nutrients{k};
            content_per_100g = nutrient_content_per_100g.(food).(nutrient);
            nutrient_total = (content_per_100g / 100) * total_amount;

            if strcmp(nutrient, 'Protein')
                energy_from_protein = energy_from_protein + nutrient_total * 4;
            elseif strcmp(nutrient, 'Fat')
                energy_from_fat = energy_from_fat + nutrient_total * 9;
            elseif strcmp(nutrient, 'Carbohydrate')
                energy_from_carbs = energy_from_carbs + nutrient_total * 4;
            end
            if ismember(nutrient, {'Protein', 'Carbohydrate'})
                meal_energy = meal_energy + nutrient_total * 4;
            elseif strcmp(nutrient, 'Fat')
                meal_energy = meal_energy + nutrient_total * 9;
            end
        end
    end

    if strcmp(meal, 'Breakfast')
        energy_breakfast = meal_energy;
    elseif strcmp(meal, 'Lunch')
        energy_lunch = meal_energy;
    elseif strcmp(meal, 'Dinner')
        energy_dinner = meal_energy;
    end
end

total_energy = energy_from_protein + energy_from_fat + energy_from_carbs;

% 输出结果
fprintf('蛋白质总摄入量: %.2f克\n', energy_from_protein / 4);
fprintf('脂肪总摄入量: %.2f克\n', energy_from_fat / 9);
fprintf('碳水化合物总摄入量: %.2f克\n', energy_from_carbs / 4);

fprintf('蛋白质占总能量百分比: %.2f%%\n', (energy_from_protein / total_energy) *
100);
fprintf('脂肪占总能量百分比: %.2f%%\n', (energy_from_fat / total_energy) * 100);
fprintf('碳水化合物占总能量百分比: %.2f%%\n', (energy_from_carbs /
total_energy) * 100);
```

```
ideal_protein_percent = 12.5;
ideal_fat_percent = 25;
ideal_carbs_percent = 62.5;

fprintf('蛋白质占比与理想差距: %.2f%%\n', (energy_from_protein / total_energy) * 100 - ideal_protein_percent);
fprintf('脂肪占比与理想差距: %.2f%%\n', (energy_from_fat / total_energy) * 100 - ideal_fat_percent);
fprintf('碳水化合物占比与理想差距: %.2f%%\n', (energy_from_carbs / total_energy) * 100 - ideal_carbs_percent);

total_meal_energy = energy_breakfast + energy_lunch + energy_dinner;
fprintf('早餐餐次比: %.2f%%\n', (energy_breakfast / total_meal_energy) * 100);
fprintf('午餐餐次比: %.2f%%\n', (energy_lunch / total_meal_energy) * 100);
fprintf('晚餐餐次比: %.2f%%\n', (energy_dinner / total_meal_energy) * 100);

ideal_breakfast_percent = 30;
ideal_lunch_percent = 35;
ideal_dinner_percent = 35;

fprintf('早餐餐次比与理想差距: %.2f%%\n', (energy_breakfast / total_meal_energy) * 100 - ideal_breakfast_percent);
fprintf('午餐餐次比与理想差距: %.2f%%\n', (energy_lunch / total_meal_energy) * 100 - ideal_lunch_percent);
fprintf('晚餐餐次比与理想差距: %.2f%%\n', (energy_dinner / total_meal_energy) * 100 - ideal_dinner_percent);

% 绘制营养成分占比图
figure;
```

```
categories = categorical({'蛋白质', '脂肪', '碳水化合物'});
values = [(energy_from_protein / total_energy) * 100, (energy_from_fat /
total_energy) * 100, (energy_from_carbs / total_energy) * 100];
ideal_values = [ideal_protein_percent, ideal_fat_percent, ideal_carbs_percent];

bar(categories, values, 'FaceColor', 'flat');
hold on;
plot(1:3, ideal_values, 'r--', 'LineWidth', 2);
hold off;

title('Energy Nutrient Proportion');
xlabel('Nutrient');
ylabel('Percentage');
legend({'Actual', 'Ideal'}, 'Location', 'northwest');
ylim([0 100]);

% 绘制餐次比例图
figure;
labels = {'早餐', '午餐', '晚餐'};
sizes = [(energy_breakfast / total_meal_energy) * 100, (energy_lunch /
total_meal_energy) * 100, (energy_dinner / total_meal_energy) * 100];

pie(sizes, labels);
title('Meal Proportion');
legend(labels, 'Location', 'southoutside', 'Orientation', 'horizontal');
```

11.1.2 女生膳食指标评估

```
% Nutrient content per 100g
nutrient_content_per_100g = struct( ...
    'Soy_Milk_Soybean', struct('Carbohydrates', 11.1, 'Protein', 35.1, 'Fat', 18.4), ...
    'Chicken_Noodle_Wheat_Flour', struct('Carbohydrates', 70.0, 'Protein', 10.5, 'Fat', 1.8), ...
    'Chicken_Noodle_Chicken_Meat', struct('Carbohydrates', 0.0, 'Protein', 23.0, 'Fat', 6.0), ...
    'Chicken_Noodle_Soybean_Oil', struct('Carbohydrates', 0.0, 'Protein', 0.0, 'Fat', 99.9), ...
    'Egg_Cake_Wheat_Flour', struct('Carbohydrates', 70.0, 'Protein', 10.5, 'Fat', 1.8), ...
    'Egg_Cake_Egg', struct('Carbohydrates', 2.5, 'Protein', 13.1, 'Fat', 10.6), ...
    'Egg_Cake_Sausage', struct('Carbohydrates', 9.0, 'Protein', 14.0, 'Fat', 26.0), ...
    'Egg_Cake_Soybean_Oil', struct('Carbohydrates', 0.0, 'Protein', 0.0, 'Fat', 99.9), ...
    'Dumpling_Wheat_Flour', struct('Carbohydrates', 70.0, 'Protein', 10.5, 'Fat', 1.8), ...
    'Dumpling_Pork', struct('Carbohydrates', 0.0, 'Protein', 20.6, 'Fat', 39.8), ...
    'Dumpling_Cabbage', struct('Carbohydrates', 3.6, 'Protein', 1.5, 'Fat', 0.2), ...
    'Dumpling_Soybean_Oil', struct('Carbohydrates', 0.0, 'Protein', 0.0, 'Fat', 99.9), ...
    'Grape', struct('Carbohydrates', 18.0, 'Protein', 0.5, 'Fat', 0.3), ...
    'Rice_Rice', struct('Carbohydrates', 77.2, 'Protein', 7.4, 'Fat', 0.6), ...
    'Stir_fried_Kale_Kale', struct('Carbohydrates', 3.8, 'Protein', 2.0, 'Fat', 0.4), ...
    'Stir_fried_Kale_Mushroom', struct('Carbohydrates', 5.0, 'Protein', 2.2, 'Fat', 0.3), ...
    'Stir_fried_Kale_Soybean_Oil', struct('Carbohydrates', 0.0, 'Protein', 0.0, 'Fat', 99.9), ...
    'Stir_fried_Meat_with_Garlic_Stir_fried_Garlic', struct('Carbohydrates', 5.4, 'Protein', 2.0, 'Fat', 0.2), ...
    'Stir_fried_Meat_with_Garlic_Pork', struct('Carbohydrates', 0.0, 'Protein', 20.6, 'Fat', 39.8), ...
    'Stir_fried_Meat_with_Garlic_Soybean_Oil', struct('Carbohydrates', 0.0, 'Protein', 0.0, 'Fat', 99.9), ...
    'Sardine_in_Tomato_Sauce', struct('Carbohydrates', 0.0, 'Protein', 22.0, 'Fat', 12.0), ...
    'Apple', struct('Carbohydrates', 25.0, 'Protein', 0.3, 'Fat', 0.2) ...
);

% Recipe
recipe = struct( ...
    'Breakfast', struct( ...
        'Soy_Milk_Soybean', [10, 1], ...
        'Chicken_Noodle_Wheat_Flour', [50, 1], ...
        'Chicken_Noodle_Chicken_Meat', [40, 1], ...
        'Chicken_Noodle_Soybean_Oil', [5, 1] ...
    ), ...
    'Lunch', struct( ...
        'Egg_Cake_Wheat_Flour', [25, 1], ...
        'Egg_Cake_Egg', [20, 1], ...
        'Egg_Cake_Sausage', [20, 1], ...
        'Egg_Cake_Soybean_Oil', [5, 1], ...
        'Dumpling_Wheat_Flour', [50, 1], ...
        'Dumpling_Pork', [20, 1], ...
        'Dumpling_Cabbage', [40, 1], ...
        'Dumpling_Soybean_Oil', [10, 1], ...
        'Grape', [100, 1] ...
    ), ...
    'Dinner', struct( ...
        'Rice_Rice', [25 * 2, 2], ...
        'Stir_fried_Kale_Kale', [100, 0.5], ...
        'Stir_fried_Kale_Mushroom', [20, 0.5], ...
        'Stir_fried_Kale_Soybean_Oil', [5, 0.5], ...
        'Stir_fried_Meat_with_Garlic_Stir_fried_Garlic', [100, 0.5], ...
        'Stir_fried_Meat_with_Garlic_Pork', [30, 0.5], ...
        'Stir_fried_Meat_with_Garlic_Soybean_Oil', [5, 0.5], ...
        'Sardine_in_Tomato_Sauce', [100, 0.5], ...
        'Apple', [100, 1] ...
    ) ...
);

total_energy = 0;
energy_from_protein = 0;
energy_from_fat = 0;
energy_from_carbs = 0;

energy_breakfast = 0;
energy_lunch = 0;
energy_dinner = 0;

meals = fieldnames(recipe);

for i = 1:numel(meals)
    meal = meals{i};
    foods = recipe.(meal);
    food_names = fieldnames(foods);
    meal_energy = 0;

    for j = 1:numel(food_names)
        food = food_names{j};
        amount = foods.(food)(1);
        servings = foods.(food)(2);
        total_amount = amount * servings;
        nutrients = fieldnames(nutrient_content_per_100g.(food));
        for k = 1:numel(nutrients)
            nutrient = nutrients{k};
            content_per_100g = nutrient_content_per_100g.(food).(nutrient);
            nutrient_total = (content_per_100g / 100) * total_amount;

            if strcmp(nutrient, 'Protein')
                energy_from_protein = energy_from_protein + nutrient_total * 4;
            elseif strcmp(nutrient, 'Fat')
                energy_from_fat = energy_from_fat + nutrient_total * 9;
            elseif strcmp(nutrient, 'Carbohydrate')
                energy_from_carbs = energy_from_carbs + nutrient_total * 4;
            end

            if ismember(nutrient, {'Protein', 'Carbohydrate'})
                meal_energy = meal_energy + nutrient_total * 4;
            elseif strcmp(nutrient, 'Fat')
                meal_energy = meal_energy + nutrient_total * 9;
            end
        end
    end

    if strcmp(meal, 'Breakfast')
        energy_breakfast = meal_energy;
    elseif strcmp(meal, 'Lunch')
        energy_lunch = meal_energy;
    elseif strcmp(meal, 'Dinner')
        energy_dinner = meal_energy;
    end
end
```

```
total_energy = energy_from_protein + energy_from_fat + energy_from_carbs;

fprintf('Total Protein Intake: %.2f grams\n', energy_from_protein / 4);
fprintf('Total Fat Intake: %.2f grams\n', energy_from_fat / 9);
fprintf('Total Carbohydrates Intake: %.2f grams\n', energy_from_carbs / 4);

fprintf('Protein Percentage of Total Energy: %.2f%%\n', (energy_from_protein / total_energy) * 100);
fprintf('Fat Percentage of Total Energy: %.2f%%\n', (energy_from_fat / total_energy) * 100);
fprintf('Carbohydrates Percentage of Total Energy: %.2f%%\n', (energy_from_carbs / total_energy) * 100);

ideal_protein_percent = 12.5;
ideal_fat_percent = 25;
ideal_carbs_percent = 62.5;

fprintf('Protein Percentage Difference from Ideal: %.2f%%\n', ((energy_from_protein / total_energy) * 100) - ideal_protein_percent);
fprintf('Fat Percentage Difference from Ideal: %.2f%%\n', ((energy_from_fat / total_energy) * 100) - ideal_fat_percent);
fprintf('Carbohydrates Percentage Difference from Ideal: %.2f%%\n', ((energy_from_carbs / total_energy) * 100) - ideal_carbs_percent);

total_meal_energy = energy_breakfast + energy_lunch + energy_dinner;
fprintf('Breakfast Meal Ratio: %.2f%%\n', (energy_breakfast / total_meal_energy) * 100);
fprintf('Lunch Meal Ratio: %.2f%%\n', (energy_lunch / total_meal_energy) * 100);
fprintf('Dinner Meal Ratio: %.2f%%\n', (energy_dinner / total_meal_energy) * 100);
```

```
ideal_breakfast_percent = 30;
ideal_lunch_percent = 35;
ideal_dinner_percent = 35;

fprintf('Breakfast Meal Ratio Difference from Ideal: %.2f%%\n', ((energy_breakfast /
total_meal_energy) * 100) - ideal_breakfast_percent);
fprintf('Lunch Meal Ratio Difference from Ideal: %.2f%%\n', ((energy_lunch /
total_meal_energy) * 100) - ideal_lunch_percent);
fprintf('Dinner Meal Ratio Difference from Ideal: %.2f%%\n', ((energy_dinner /
total_meal_energy) * 100) - ideal_dinner_percent);

% Plotting
figure;
categories = {'Protein', 'Fat', 'Carbohydrates'};
values = [(energy_from_protein / total_energy) * 100, (energy_from_fat /
total_energy) * 100, (energy_from_carbs / total_energy) * 100];
ideal_values = [ideal_protein_percent, ideal_fat_percent, ideal_carbs_percent];

bar(categories, values, 'FaceColor', 'flat');
hold on;
plot(1:3, ideal_values, 'r--', 'LineWidth', 2);
hold off;

title('Energy Nutrient Proportion');
xlabel('Nutrient');
ylabel('Percentage');
legend({'Actual', 'Ideal'}, 'Location', 'northwest');
ylim([0 100]);
```

```
figure;
labels = {'Breakfast', 'Lunch', 'Dinner'};
sizes = [(energy_breakfast / total_meal_energy) * 100, (energy_lunch /
total_meal_energy) * 100, (energy_dinner / total_meal_energy) * 100];

pie(sizes, labels);
title('Meal Proportion');
legend(labels, 'Location', 'southoutside', 'Orientation', 'horizontal');
```

11.1.3 男生氨基酸评估

```
% 初始化存储每100g营养成分的结构体
nutrient_content_per_100g = struct();

% 遍历每一行并提取营养成分
for i = 1:height(nutrition_composition_man)
    food_name = nutrition_composition_man.food_name{i};
    nutrient_content_per_100g.(food_name) = struct( ...
        'Carbohydrates', nutrition_composition_man.Carbohydrates_g_100g(i), ...
        'Protein', nutrition_composition_man.Protein_g_100g(i), ...
        'Fat', nutrition_composition_man.Fat_g_100g(i), ...
        'Calcium', nutrition_composition_man.Calcium_mg_100g(i), ...
        'Iron', nutrition_composition_man.Iron_mg_100g(i), ...
        'Zinc', nutrition_composition_man.Zinc_mg_100g(i), ...
        'VitaminA', nutrition_composition_man.VitaminA_mcg_100g(i), ...
        'VitaminB1', nutrition_composition_man.VitaminB1_mg_100g(i), ...
        'VitaminB2', nutrition_composition_man.VitaminB2_mg_100g(i), ...
        'VitaminC', nutrition_composition_man.VitaminC_mg_100g(i), ...
        'Isoleucine', nutrition_composition_man.Isoleucine_g_100g(i), ...
        'Leucine', nutrition_composition_man.Leucine_g_100g(i), ...
        'Lysine', nutrition_composition_man.Lysine_g_100g(i), ...
        'SulfurAminoAcids',
nutrition_composition_man.SulfurAminoAcids_g_100g(i), ...
        'AromaticAminoAcids',
nutrition_composition_man.AromaticAminoAcids_g_100g(i), ...
        'Threonine', nutrition_composition_man.Threonine_g_100g(i), ...
        'Tryptophan', nutrition_composition_man.Tryptophan_g_100g(i), ...
        'Valine', nutrition_composition_man.Valine_g_100g(i) ...
    );
end

% 预定义食谱
recipe = struct( ...
    'Breakfast', struct( ...
        'XiaomiPorridge_Millet', [15, 1], ...
        'Youtiao_WheatFlour', [50*2, 2], ...
        'Youtiao_SoybeanOil', [10*2, 2], ...
        'FriedEgg_Egg', [50, 1], ...
        'FriedEgg_SoybeanOil', [10, 1], ...
        'SeaweedSalad_Seaweed', [100, 1], ...
        'SeaweedSalad_SesameOil', [2, 1] ...
    ), ...
    'Lunch', struct( ...
        'Rice_Rice', [25*4, 4], ...
        'BlackFungusSalad_BlackFungus', [100, 1], ...
        'BlackFungusSalad_SesameOil', [2, 1], ...
        'DiSanXian_Eggplant', [80, 1], ...
        'DiSanXian_Potato', [80, 1], ...
        'DiSanXian_GreenPepper', [10, 1], ...
        'DiSanXian_SoybeanOil', [10, 1], ...
        'BraisedPorkBelly_PorkBelly', [50, 1], ...
        'BraisedPorkBelly_DriedTofu', [50, 1], ...
        'BraisedPorkBelly_SoybeanOil', [10, 1] ...
    ), ...
    'Dinner', struct( ...
        'HotPotNoodles_CornFlour', [80, 1], ...
        'HotPotNoodles_Cabbage', [20, 1], ...
        'HotPotNoodles_BokChoy', [20, 1], ...
        'HotPotNoodles_DriedTofu', [10, 1], ...
        'HotPotNoodles_SoybeanOil', [10, 1], ...
        'Baozi_WheatFlour', [25, 1], ...
        'Baozi_Pork', [15, 1], ...
        'Baozi_Sauerkraut', [20, 1], ...
        'Baozi_SoybeanOil', [5, 1], ...
        'FriedChicken', [100, 1] ...
    ) ...
);

% 计算每日食物种类数量
total_food_types = 0;
meal_names = fieldnames(recipe);
for i = 1:numel(meal_names)
    total_food_types = total_food_types +
numel(fieldnames(recipe.(meal_names{i})));
end

average_food_types_per_day = total_food_types / numel(meal_names);

if average_food_types_per_day > 12
    disp('平均每天摄入食物种类数量大于12种');
else
```

```
    disp('平均每天摄入食物种类数量不足12种');
end

% 计算总能量
total_energy = 0;
for i = 1:numel(meal_names)
    foods = recipe.(meal_names{i});
    food_names = fieldnames(foods);
    for j = 1:numel(food_names)
        food = food_names{j};
        amount = foods.(food)(1);
        servings = foods.(food)(2);
        total_energy = total_energy + sum([ ...
            nutrient_content_per_100g.(food).Protein, ...
            nutrient_content_per_100g.(food).Fat, ...
            nutrient_content_per_100g.(food).Carbohydrates ...
        ] .* amount * servings / 100);
    end
end

% 计算早餐、午餐和晚餐的能量
energy_breakfast = 0;
foods = recipe.Breakfast;
food_names = fieldnames(foods);
for j = 1:numel(food_names)
    food = food_names{j};
    amount = foods.(food)(1);
    servings = foods.(food)(2);
    energy_breakfast = energy_breakfast + sum([ ...
        nutrient_content_per_100g.(food).Protein, ...
        nutrient_content_per_100g.(food).Fat, ...
        nutrient_content_per_100g.(food).Carbohydrates ...
    ] .* amount * servings / 100);
end

energy_lunch = 0;
foods = recipe.Lunch;
food_names = fieldnames(foods);
for j = 1:numel(food_names)
    food = food_names{j};
    amount = foods.(food)(1);
    servings = foods.(food)(2);
    energy_lunch = energy_lunch + sum([ ...
        nutrient_content_per_100g.(food).Protein, ...
        nutrient_content_per_100g.(food).Fat, ...
        nutrient_content_per_100g.(food).Carbohydrates ...
    ] .* amount * servings / 100);
end

energy_dinner = 0;
foods = recipe.Dinner;
food_names = fieldnames(foods);
for j = 1:numel(food_names)
    food = food_names{j};
    amount = foods.(food)(1);
    servings = foods.(food)(2);
    energy_dinner = energy_dinner + sum([ ...
        nutrient_content_per_100g.(food).Protein, ...
        nutrient_content_per_100g.(food).Fat, ...
        nutrient_content_per_100g.(food).Carbohydrates ...
    ] .* amount * servings / 100);
end

% 计算能量占比
percent_breakfast = (energy_breakfast / total_energy) * 100;
percent_lunch = (energy_lunch / total_energy) * 100;
percent_dinner = (energy_dinner / total_energy) * 100;

fprintf('早餐能量占比：%.2f%%\n', percent_breakfast);
fprintf('午餐能量占比：%.2f%%\n', percent_lunch);
fprintf('晚餐能量占比：%.2f%%\n', percent_dinner);

% 能量参考值
reference_percent_breakfast = [30, 30];
reference_percent_lunch = [30, 40];
reference_percent_dinner = [30, 40];

% 判断能量占比是否符合参考值
if percent_breakfast < reference_percent_breakfast(1)
    disp('早餐能量占比低于参考值');
elseif percent_breakfast > reference_percent_breakfast(2)
    disp('早餐能量占比高于参考值');
else
    disp('早餐能量占比符合参考值');
end

if percent_lunch < reference_percent_lunch(1)
    disp('午餐能量占比低于参考值');
elseif percent_lunch > reference_percent_lunch(2)
```

```
    disp('午餐能量占比高于参考值');
else
    disp('午餐能量占比符合参考值');
end

if percent_dinner < reference_percent_dinner(1)
    disp('晚餐能量占比低于参考值');
elseif percent_dinner > reference_percent_dinner(2)
    disp('晚餐能量占比高于参考值');
else
    disp('晚餐能量占比符合参考值');
end


% Initialize nutrient intake variables
carbs_intake = 0;
protein_intake = 0;
fat_intake = 0;

% Calculate nutrient intakes
meals = fieldnames(recipe);
for m = 1:numel(meals)
    foods = recipe.(meals{m});
    food_names = fieldnames(foods);
    for f = 1:numel(food_names)
        food = food_names{f};
        amount = foods.(food)(1);
        servings = foods.(food)(2);
        if isfield(nutrient_content_per_100g, food)
            carbs_intake = carbs_intake +
nutrient_content_per_100g.(food).Carbohydrates * amount * servings / 100;
            protein_intake = protein_intake +
nutrient_content_per_100g.(food).Protein * amount * servings / 100;
            fat_intake = fat_intake + nutrient_content_per_100g.(food).Fat * amount
* servings / 100;
        end
    end
end

% Daily energy requirement
daily_energy_requirement = 2400;

% Calculate nutrient percentages
protein_percent = (protein_intake * 4 / daily_energy_requirement) * 100;
fat_percent = (fat_intake * 9 / daily_energy_requirement) * 100;
carbs_percent = (carbs_intake * 4 / daily_energy_requirement) * 100;

% Print nutrient intake and percentages
fprintf('每日摄入的蛋白质量: %.2f克\n', protein_intake);
fprintf('每日摄入的脂肪量: %.2f克\n', fat_intake);
fprintf('每日摄入的碳水化合物量: %.2f克\n', carbs_intake);
fprintf('蛋白质占总能量的百分比: %.2f%%\n', protein_percent);
fprintf('脂肪占总能量的百分比: %.2f%%\n', fat_percent);
fprintf('碳水化合物占总能量的百分比: %.2f%%\n', carbs_percent);

% Reference values
reference_protein_percent = [10, 15];
reference_fat_percent = [20, 30];
reference_carbs_percent = [50, 65];
```

```
% Check nutrient intake against reference values
if protein_percent < reference_protein_percent(1)
    disp('蛋白质摄入量低于参考值');
elseif protein_percent > reference_protein_percent(2)
    disp('蛋白质摄入量高于参考值');
else
    disp('蛋白质摄入量符合参考值');
end

if fat_percent < reference_fat_percent(1)
    disp('脂肪摄入量低于参考值');
elseif fat_percent > reference_fat_percent(2)
    disp('脂肪摄入量高于参考值');
else
    disp('脂肪摄入量符合参考值');
end

if carbs_percent < reference_carbs_percent(1)
    disp('碳水化合物摄入量低于参考值');
elseif carbs_percent > reference_carbs_percent(2)
    disp('碳水化合物摄入量高于参考值');
else
    disp('碳水化合物摄入量符合参考值');
end

% Amino acid reference values
reference_amino_acids = struct( ...
    'Isoleucine', 40, ...
    'Leucine', 70, ...
    'Lysine', 55, ...
    'SulfurAminoAcids', 35, ...
    'AromaticAminoAcids', 60, ...
    'Threonine', 40, ...
    'Tryptophan', 10, ...
    'Valine', 50 ...
);

% Calculate total amino acid scores
total_protein = 0;
total_amino_acid_scores = struct();

for m = 1:numel(meals)
    foods = recipe.(meals{m});
    food_names = fieldnames(foods);
    for f = 1:numel(food_names)
        food = food_names{f};
        amount = foods.(food)(1);
        servings = foods.(food)(2);
        if isfield(nutrient_content_per_100g, food)
            total_protein = total_protein + nutrient_content_per_100g.(food).Protein
* (amount / 100) * servings;
            amino_acid_names = fieldnames(reference_amino_acids);
            for a = 1:numel(amino_acid_names)
                amino_acid = amino_acid_names{a};
                if isfield(nutrient_content_per_100g.(food), amino_acid)
                    amino_acid_content =
nutrient_content_per_100g.(food).(amino_acid);
                    score = min((amino_acid_content /
reference_amino_acids.(amino_acid)), 1);
                    if isfield(total_amino_acid_scores, amino_acid)
```

```
                        total_amino_acid_scores.(amino_acid) = total_amino_acid_scores.(amino_acid) + score * (amount / 100) * servings;
                    else
                        total_amino_acid_scores.(amino_acid) = score * (amount / 100) * servings;
                    end
                end
            end
        end
    end
end

total_aas_score = sum(cell2mat(struct2cell(total_amino_acid_scores))) / numel(fieldnames(reference_amino_acids));

fprintf('总氨基酸评分: %.2f\n', total_aas_score);
```

11.1.4 女生氨基酸评估

```
% 初始化存储每100g营养成分的结构体
nutrient_content_per_100g_woman = struct();

% 遍历每一行并提取营养成分
for i = 1:height(nutrition_composition_woman)
    food_name = nutrition_composition_woman.food_name{i};
    nutrient_content_per_100g_woman.(food_name) = struct( ...
        'Carbohydrates', nutrition_composition_woman.Carbohydrates_g_100g(i), ...
        'Protein', nutrition_composition_woman.Protein_g_100g(i), ...
        'Fat', nutrition_composition_woman.Fat_g_100g(i), ...
        'Calcium', nutrition_composition_woman.Calcium_mg_100g(i), ...
        'Iron', nutrition_composition_woman.Iron_mg_100g(i), ...
        'Zinc', nutrition_composition_woman.Zinc_mg_100g(i), ...
        'VitaminA', nutrition_composition_woman.VitaminA_mcg_100g(i), ...
        'VitaminB1', nutrition_composition_woman.VitaminB1_mg_100g(i), ...
        'VitaminB2', nutrition_composition_woman.VitaminB2_mg_100g(i), ...
        'VitaminC', nutrition_composition_woman.VitaminC_mg_100g(i), ...
        'Isoleucine', nutrition_composition_woman.Isoleucine_g_100g(i), ...
        'Leucine', nutrition_composition_woman.Leucine_g_100g(i), ...
        'Lysine', nutrition_composition_woman.Lysine_g_100g(i), ...
        'SulfurAminoAcids',
nutrition_composition_woman.SulfurAminoAcids_g_100g(i), ...
        'AromaticAminoAcids',
nutrition_composition_woman.AromaticAminoAcids_g_100g(i), ...
        'Threonine', nutrition_composition_woman.Threonine_g_100g(i), ...
        'Tryptophan', nutrition_composition_woman.Tryptophan_g_100g(i), ...
        'Valine', nutrition_composition_woman.Valine_g_100g(i) ...
    );
end

% Recipe
recipe = struct( ...
    'Breakfast', struct( ...
        'Soy_Milk_Soybean', [10, 1], ...
        'Chicken_Noodle_Wheat_Flour', [50, 1], ...
        'Chicken_Noodle_Chicken_Meat', [40, 1], ...
        'Chicken_Noodle_Soybean_Oil', [5, 1] ...
    ), ...
    'Lunch', struct( ...
        'Egg_Cake_Wheat_Flour', [25, 1], ...
        'Egg_Cake_Egg', [20, 1], ...
        'Egg_Cake_Sausage', [20, 1], ...
        'Egg_Cake_Soybean_Oil', [5, 1], ...
        'Dumpling_Wheat_Flour', [50, 1], ...
        'Dumpling_Pork', [20, 1], ...
        'Dumpling_Cabbage', [40, 1], ...
        'Dumpling_Soybean_Oil', [10, 1], ...
        'Grape', [100, 1] ...
    ), ...
    'Dinner', struct( ...
        'Rice_Rice', [25 * 2, 2], ...
        'Stir_fried_Kale_Kale', [100, 0.5], ...
        'Stir_fried_Kale_Mushroom', [20, 0.5], ...
        'Stir_fried_Kale_Soybean_Oil', [5, 0.5], ...
        'Stir_fried_Meat_with_Garlic_Stir_fried_Garlic', [100, 0.5], ...
        'Stir_fried_Meat_with_Garlic_Pork', [30, 0.5], ...
        'Stir_fried_Meat_with_Garlic_Soybean_Oil', [5, 0.5], ...
        'Sardine_in_Tomato_Sauce', [100, 0.5], ...
        'Apple', [100, 1] ...
    ) ...
);

% 计算每日食物种类数量
total_food_types = 0;
meal_names = fieldnames(recipe);
for i = 1:numel(meal_names)
    total_food_types = total_food_types +
numel(fieldnames(recipe.(meal_names{i})));
end

average_food_types_per_day = total_food_types / numel(meal_names);
```

```
if average_food_types_per_day > 12
    disp('平均每天摄入食物种类数量大于12种');
else
    disp('平均每天摄入食物种类数量不足12种');
end

% 计算总能量
total_energy = 0;
for i = 1:numel(meal_names)
    foods = recipe.(meal_names{i});
    food_names = fieldnames(foods);
    for j = 1:numel(food_names)
        food = food_names{j};
        amount = foods.(food)(1);
        servings = foods.(food)(2);
        total_energy = total_energy + sum([ ...
            nutrient_content_per_100g_woman.(food).Protein, ...
            nutrient_content_per_100g_woman.(food).Fat, ...
            nutrient_content_per_100g_woman.(food).Carbohydrates ...
        ] .* amount * servings / 100);
    end
end

% 计算早餐、午餐和晚餐的能量
energy_breakfast = 0;
foods = recipe.Breakfast;
food_names = fieldnames(foods);
for j = 1:numel(food_names)
    food = food_names{j};
    amount = foods.(food)(1);
    servings = foods.(food)(2);
    energy_breakfast = energy_breakfast + sum([ ...
        nutrient_content_per_100g_woman.(food).Protein, ...
        nutrient_content_per_100g_woman.(food).Fat, ...
        nutrient_content_per_100g_woman.(food).Carbohydrates ...
    ] .* amount * servings / 100);
end

energy_lunch = 0;
foods = recipe.Lunch;
food_names = fieldnames(foods);
for j = 1:numel(food_names)
    food = food_names{j};
    amount = foods.(food)(1);
    servings = foods.(food)(2);
    energy_lunch = energy_lunch + sum([ ...
        nutrient_content_per_100g_woman.(food).Protein, ...
        nutrient_content_per_100g_woman.(food).Fat, ...
        nutrient_content_per_100g_woman.(food).Carbohydrates ...
    ] .* amount * servings / 100);
end

energy_dinner = 0;
foods = recipe.Dinner;
food_names = fieldnames(foods);
for j = 1:numel(food_names)
    food = food_names{j};
    amount = foods.(food)(1);
    servings = foods.(food)(2);
    energy_dinner = energy_dinner + sum([ ...
        nutrient_content_per_100g_woman.(food).Protein, ...
        nutrient_content_per_100g_woman.(food).Fat, ...
        nutrient_content_per_100g_woman.(food).Carbohydrates ...
    ] .* amount * servings / 100);
end

% 计算能量占比
percent_breakfast = (energy_breakfast / total_energy) * 100;
percent_lunch = (energy_lunch / total_energy) * 100;
percent_dinner = (energy_dinner / total_energy) * 100;

fprintf('早餐能量占比：%.2f%%\n', percent_breakfast);
fprintf('午餐能量占比：%.2f%%\n', percent_lunch);
fprintf('晚餐能量占比：%.2f%%\n', percent_dinner);

% 能量参考值
reference_percent_breakfast = [30, 30];
reference_percent_lunch = [30, 40];
reference_percent_dinner = [30, 40];

% 判断能量占比是否符合参考值
if percent_breakfast < reference_percent_breakfast(1)
    disp('早餐能量占比低于参考值');
elseif percent_breakfast > reference_percent_breakfast(2)
    disp('早餐能量占比高于参考值');
else
    disp('早餐能量占比符合参考值');
end
```

```
if percent_lunch < reference_percent_lunch(1)
    disp('午餐能量占比低于参考值');
elseif percent_lunch > reference_percent_lunch(2)
    disp('午餐能量占比高于参考值');
else
    disp('午餐能量占比符合参考值');
end

if percent_dinner < reference_percent_dinner(1)
    disp('晚餐能量占比低于参考值');
elseif percent_dinner > reference_percent_dinner(2)
    disp('晚餐能量占比高于参考值');
else
    disp('晚餐能量占比符合参考值');
end



% Initialize nutrient intake variables
carbs_intake = 0;
protein_intake = 0;
fat_intake = 0;

% Calculate nutrient intakes
meals = fieldnames(recipe);
for m = 1:numel(meals)
    foods = recipe.(meals{m});
    food_names = fieldnames(foods);
    for f = 1:numel(food_names)
        food = food_names{f};
        amount = foods.(food)(1);
        servings = foods.(food)(2);
        if isfield(nutrient_content_per_100g, food)
            carbs_intake = carbs_intake +
nutrient_content_per_100g.(food).Carbohydrates * amount * servings / 100;
            protein_intake = protein_intake +
nutrient_content_per_100g.(food).Protein * amount * servings / 100;
            fat_intake = fat_intake + nutrient_content_per_100g.(food).Fat * amount
* servings / 100;
        end
    end
end

% Daily energy requirement
daily_energy_requirement = 2400;

% Calculate nutrient percentages
protein_percent = (protein_intake * 4 / daily_energy_requirement) * 100;
fat_percent = (fat_intake * 9 / daily_energy_requirement) * 100;
carbs_percent = (carbs_intake * 4 / daily_energy_requirement) * 100;

% Print nutrient intake and percentages
fprintf('每日摄入的蛋白质量: %.2f克\n', protein_intake);
fprintf('每日摄入的脂肪量: %.2f克\n', fat_intake);
fprintf('每日摄入的碳水化合物量: %.2f克\n', carbs_intake);
fprintf('蛋白质占总能量的百分比: %.2f%%\n', protein_percent);
fprintf('脂肪占总能量的百分比: %.2f%%\n', fat_percent);
fprintf('碳水化合物占总能量的百分比: %.2f%%\n', carbs_percent);

% Reference values
reference_protein_percent = [10, 15];
```

```
reference_fat_percent = [20, 30];
reference_carbs_percent = [50, 65];

% Check nutrient intake against reference values
if protein_percent < reference_protein_percent(1)
    disp('蛋白质摄入量低于参考值');
elseif protein_percent > reference_protein_percent(2)
    disp('蛋白质摄入量高于参考值');
else
    disp('蛋白质摄入量符合参考值');
end

if fat_percent < reference_fat_percent(1)
    disp('脂肪摄入量低于参考值');
elseif fat_percent > reference_fat_percent(2)
    disp('脂肪摄入量高于参考值');
else
    disp('脂肪摄入量符合参考值');
end

if carbs_percent < reference_carbs_percent(1)
    disp('碳水化合物摄入量低于参考值');
elseif carbs_percent > reference_carbs_percent(2)
    disp('碳水化合物摄入量高于参考值');
else
    disp('碳水化合物摄入量符合参考值');
end

% Amino acid reference values
reference_amino_acids = struct( ...
    'Isoleucine', 40, ...
    'Leucine', 70, ...
    'Lysine', 55, ...
    'SulfurAminoAcids', 35, ...
    'AromaticAminoAcids', 60, ...
    'Threonine', 40, ...
    'Tryptophan', 10, ...
    'Valine', 50 ...
);

% Calculate total amino acid scores
total_protein = 0;
total_amino_acid_scores = struct();

for m = 1:numel(meals)
    foods = recipe.(meals{m});
    food_names = fieldnames(foods);
    for f = 1:numel(food_names)
        food = food_names{f};
        amount = foods.(food)(1);
        servings = foods.(food)(2);
        if isfield(nutrient_content_per_100g, food)
            total_protein = total_protein + nutrient_content_per_100g.(food).Protein
* (amount / 100) * servings;
            amino_acid_names = fieldnames(reference_amino_acids);
            for a = 1:numel(amino_acid_names)
                amino_acid = amino_acid_names{a};
                if isfield(nutrient_content_per_100g.(food), amino_acid)
                    amino_acid_content =
nutrient_content_per_100g.(food).(amino_acid);
```

```
                    score = min((amino_acid_content /
reference_amino_acids.(amino_acid)), 1);
                    if isfield(total_amino_acid_scores, amino_acid)
                        total_amino_acid_scores.(amino_acid) =
total_amino_acid_scores.(amino_acid) + score * (amount / 100) * servings;
                    else
                        total_amino_acid_scores.(amino_acid) = score * (amount /
100) * servings;
                    end
                end
            end
        end
    end
end

total_aas_score = sum(cell2mat(struct2cell(total_amino_acid_scores))) /
numel(fieldnames(reference_amino_acids));

fprintf('总氨基酸评分: %.2f\n', total_aas_score);
```

11.1.5膳食总指标评估

```
%
通过考虑三个主要营养素（蛋白质、脂肪和碳水化合物）的摄入量和占比差距
，以及餐次比例的差距来计算评分。
% 评分使用sigmoid函数标准化差距，并按权重加权平均得到总评分。

% 1.Sigmoid标准化：将差距标准化为0到100的分数。
%
2.营养素评分：根据标准化后的分数和营养素的权重计算每种营养素的评分。
```

```
% 3.餐次比评分：根据实际餐次比和理想餐次比的差距计算评分。
% 4.总评分：将所有营养素的评分和餐次比的评分相加得到总评分。

weights = struct( ...
    'Protein', 0.2, ...
    'Fat', 0.2, ...
    'Carbohydrates', 0.3, ...
    'MealRatio', 0.3 ...
);

% Sigmoid function
function result = sigmoid(x)
    result = 1 ./ (1 + exp(-x));
end

% Function to standardize the score
function result = standardize_score(gap, scale_factor)
    x = (scale_factor / 2 - abs(gap)) / scale_factor;
    result = round(sigmoid(x) * 100, 2);
end

% Function to score nutrients
function result = score_nutrient(gap, weight, scale_factor)
    result = standardize_score(gap, scale_factor) * weight;
end

% Function to score meal ratios
function result = score_meal_ratio(actual_ratios, ideal_ratios, weight, scale_factor)
    scores = arrayfun(@(x, y) standardize_score(abs(x - y), scale_factor),
actual_ratios, ideal_ratios);
```

```
    result = round(sum(scores) * (weight / length(ideal_ratios)), 2);
end

male_data = struct( ...
    'Protein', struct('Intake', 139.19, 'Gap', 0.38), ...
    'Fat', struct('Intake', 159.57, 'Gap', 8.23), ...
    'Carbohydrates', struct('Intake', 582.12, 'Gap', -8.62), ...
    'MealRatio', [30.60, 48.73, 20.67] ...
);

female_data = struct( ...
    'Protein', struct('Intake', 60.48, 'Gap', 2.12), ...
    'Fat', struct('Intake', 60.23, 'Gap', 7.76), ...
    'Carbohydrates', struct('Intake', 217.65, 'Gap', -9.88), ...
    'MealRatio', [18.58, 39.83, 41.59] ...
);

ideal_meal_ratios = [30, 40, 30];

scale_factor = 5;

male_scores = struct();
female_scores = struct();

for nutrient = fieldnames(male_data)'
    nutrient = nutrient{1};
    if ~strcmp(nutrient, 'MealRatio')
        male_scores.(nutrient) = score_nutrient(male_data.(nutrient).Gap,
weights.(nutrient), scale_factor);
        female_scores.(nutrient) = score_nutrient(female_data.(nutrient).Gap,
weights.(nutrient), scale_factor);
    else
        male_scores.(nutrient) = score_meal_ratio(male_data.MealRatio, ideal_meal_ratios, weights.MealRatio, scale_factor);
        female_scores.(nutrient) = score_meal_ratio(female_data.(nutrient), ideal_meal_ratios, weights.(nutrient), scale_factor);
    end
end

male_total_score = sum(cell2mat(struct2cell(male_scores)));
female_total_score = sum(cell2mat(struct2cell(female_scores)));

fprintf('男生膳食质量指数总评分: %.2f\n', male_total_score);
fprintf('女生膳食质量指数总评分: %.2f\n', female_total_score);
```

11.2问题2的程序

11.2.1 question2_1

```
clear;clc

load('matlab.mat')
%
% N1, N2, N3分别为早餐、中餐、晚餐的菜品数
N1 = height(breakfast.recipe_name); % 早餐菜品数
N2 = height(lunch.recipe_name); % 中餐菜品数
N3 = height(dinner.recipe_name); % 晚餐菜品数

% 分别定义三餐的营养成分数据
% 早餐数据
breakfast_energy = (breakfast.energy_kcal .* breakfast.avail_part_g) / 100; % 能量(kcal/100g)
```

```
breakfast_protein = (breakfast.protein_g .* breakfast.avail_part_g) / 100; % 蛋白质 (g/100g)
breakfast_fat = (breakfast.fat_g .* breakfast.avail_part_g) / 100; % 脂肪 (g/100g)
breakfast_carbs = (breakfast.carbohydrates_g .* breakfast.avail_part_g) / 100; % 碳水化合物 (g/100g)
breakfast_cost = (breakfast.price_per_part .* breakfast.avail_part_g) / 100; % 费用 (元/100g)
breakfast_essential_aa = ((breakfast.isoleucine_g + breakfast.leucine_g + breakfast.lysine_g + breakfast.total_g + breakfast.valine_g + breakfast.cystine_g + breakfast.tyrosine_g + breakfast.histidine_g + breakfast.arginine_g + breakfast.alanine_g + breakfast.aspartic_acid_g + breakfast.glutamic_acid_g + breakfast.glycine_g) .* breakfast.avail_part_g) / 100; % 必需氨基酸含量
breakfast_reference_protein_aa = ((breakfast.isoleucine_g + breakfast.leucine_g + breakfast.lysine_g + breakfast.methionine_g + breakfast.phenylalanine_g + breakfast.tryptophan_g + breakfast.tryptophan_g + breakfast.valine_g + breakfast.histidine_g) .* breakfast.avail_part_g) / 100;
breakfast_calcium = (breakfast.calcium_mg .* breakfast.avail_part_g) / 100; % 钙 (mg/100g)
breakfast_iron = (breakfast.iron_mg .* breakfast.avail_part_g) / 100; % 铁 (mg/100g)
breakfast_zinc =   (breakfast.zinc_mg .* breakfast.avail_part_g) / 100; % 锌 (mg/100g)
breakfast_vitA = (breakfast.total_vitaminA_mcg .* breakfast.avail_part_g) / 100; % 维生素A (ug/100g)
breakfast_vitB1 = (breakfast.thiamine_mg .* breakfast.avail_part_g) / 100; % 维生素B1 (mg/100g)
breakfast_vitB2 = (breakfast.riboflavin_mg .* breakfast.avail_part_g) / 100; % 维生素B2 (mg/100g)
breakfast_vitC = (breakfast.vitaminC_mg .* breakfast.avail_part_g) / 100; % 维生素C (mg/100g)
```

```
% 中餐数据
lunch_energy = (lunch.energy_kcal .* lunch.avail_part_g) / 100; % 能量 (kcal/100g)
lunch_protein = (lunch.protein_g .* lunch.avail_part_g) / 100; % 蛋白质 (g/100g)
lunch_fat = (lunch.fat_g .* lunch.avail_part_g) / 100; % 脂肪 (g/100g)
lunch_carbs = (lunch.carbohydrates_g .* lunch.avail_part_g) / 100; % 碳水化合物 (g/100g)
lunch_cost = (lunch.price_per_part .* lunch.avail_part_g) / 100; % 费用 (元/100g)
lunch_essential_aa = ((lunch.isoleucine_g + lunch.leucine_g + lunch.lysine_g + lunch.total_g + lunch.valine_g + lunch.cystine_g + lunch.tyrosine_g + lunch.histidine_g + lunch.arginine_g + lunch.alanine_g + lunch.aspartic_acid_g + lunch.glutamic_acid_g + lunch.glycine_g) .* lunch.avail_part_g) / 100; % 必需氨基酸含量
lunch_reference_protein_aa = ((lunch.isoleucine_g + lunch.leucine_g + lunch.lysine_g + lunch.methionine_g + lunch.phenylalanine_g + lunch.tryptophan_g + lunch.tryptophan_g + lunch.valine_g + lunch.histidine_g) .* lunch.avail_part_g) / 100; % 参考蛋白质氨基酸含量
lunch_calcium = (lunch.calcium_mg .* lunch.avail_part_g) / 100; % 钙 (mg/100g)
lunch_iron = (lunch.iron_mg .* lunch.avail_part_g) / 100; % 铁 (mg/100g)
lunch_zinc =   (lunch.zinc_mg .* lunch.avail_part_g) / 100; % 锌 (mg/100g)
lunch_vitA = (lunch.total_vitaminA_mcg .* lunch.avail_part_g) / 100; % 维生素A (ug/100g)
lunch_vitB1 = (lunch.thiamine_mg .* lunch.avail_part_g) / 100; % 维生素B1 (mg/100g)
lunch_vitB2 = (lunch.riboflavin_mg .* lunch.avail_part_g) / 100; % 维生素B2 (mg/100g)
lunch_vitC = (lunch.vitaminC_mg .* lunch.avail_part_g) / 100; % 维生素C (mg/100g)

% 晚餐数据
```

```
dinner_energy = (dinner.energy_kcal .* dinner.avail_part_g) / 100; % 能量
(kcal/100g)
dinner_protein = (dinner.protein_g .* dinner.avail_part_g) / 100; % 蛋白质 (g/100g)
dinner_fat = (dinner.fat_g .* dinner.avail_part_g) / 100; % 脂肪 (g/100g)
dinner_carbs = (dinner.carbohydrates_g .* dinner.avail_part_g) / 100; % 碳水化合物
(g/100g)
dinner_cost = (dinner.price_per_part .* dinner.avail_part_g) / 100; % 费用 (元/100g)
dinner_essential_aa = ((dinner.isoleucine_g + dinner.leucine_g + dinner.lysine_g +
dinner.total_g + dinner.valine_g + dinner.cystine_g + dinner.tyrosine_g +
dinner.histidine_g + dinner.arginine_g + dinner.alanine_g + dinner.aspartic_acid_g +
dinner.glutamic_acid_g + dinner.glycine_g) .* dinner.avail_part_g) / 100; %
必需氨基酸含量
dinner_reference_protein_aa = ((dinner.isoleucine_g + dinner.leucine_g +
dinner.lysine_g + dinner.methionine_g + dinner.phenylalanine_g +
dinner.tryptophan_g + dinner.tryptophan_g + dinner.valine_g + dinner.histidine_g) .*
dinner.avail_part_g) / 100; % 参考蛋白质氨基酸含量
dinner_calcium = (dinner.calcium_mg .* dinner.avail_part_g) / 100; % 钙 (mg/100g)
dinner_iron = (dinner.iron_mg .* dinner.avail_part_g) / 100; % 铁 (mg/100g)
dinner_zinc =   (dinner.zinc_mg .* dinner.avail_part_g) / 100; % 锌 (mg/100g)
dinner_vitA = (dinner.total_vitaminA_mcg .* dinner.avail_part_g) / 100; % 维生素A
(ug/100g)
dinner_vitB1 = (dinner.thiamine_mg .* dinner.avail_part_g) / 100; % 维生素B1
(mg/100g)
dinner_vitB2 = (dinner.riboflavin_mg .* dinner.avail_part_g) / 100; % 维生素B2
(mg/100g)
dinner_vitC = (dinner.vitaminC_mg .* dinner.avail_part_g) / 100; % 维生素C
(mg/100g)

% 性别和相应的目标能量
```

```
gender = 'male'; % 或 'male'
if strcmp(gender, 'female')
    E_target = 1900;
    iron_ref = 20;
    zinc_ref = 7.5;
    vitA_ref = 700;
    vitB1_ref = 1.2;
    vitB2_ref = 1.2;
else
    E_target = 2400;
    iron_ref = 12;
    zinc_ref = 12.5;
    vitA_ref = 800;
    vitB1_ref = 1.4;
    vitB2_ref = 1.4;
end

% 决策变量，食物的选择量 (100g 单位)
x_breakfast = optimvar('x_breakfast', N1, 'LowerBound', 0);
x_lunch = optimvar('x_lunch', N2, 'LowerBound', 0);
x_dinner = optimvar('x_dinner', N3, 'LowerBound', 0);

% 构建目标函数，最大化总的蛋白质氨基酸评分
AAS_breakfast = (x_breakfast' * breakfast_essential_aa) / (x_breakfast' * breakfast_reference_protein_aa) * 100;
AAS_lunch = (x_lunch' * lunch_essential_aa) / (x_lunch' * lunch_reference_protein_aa) * 100;
AAS_dinner = (x_dinner' * dinner_essential_aa) / (x_dinner' * dinner_reference_protein_aa) * 100;
AAS_total = AAS_breakfast + AAS_lunch + AAS_dinner;
```

```
prob = optimproblem('Objective', -AAS_total); % 最大化AAS, 所以使用负号

% 总能量摄入约束
prob.Constraints.energy_total_min = sum(x_breakfast .* breakfast_energy) + sum(x_lunch .* lunch_energy) + sum(x_dinner .* dinner_energy) >= 0.9 * E_target;
prob.Constraints.energy_total_max = sum(x_breakfast .* breakfast_energy) + sum(x_lunch .* lunch_energy) + sum(x_dinner .* dinner_energy) <= 1.1 * E_target;

% 早餐能量分配
prob.Constraints.energy_breakfast_min = sum(x_breakfast .* breakfast_energy) >= 0.25 * sum([x_breakfast; x_lunch; x_dinner] .* [breakfast_energy; lunch_energy; dinner_energy]);
prob.Constraints.energy_breakfast_max = sum(x_breakfast .* breakfast_energy) <= 0.35 * sum([x_breakfast; x_lunch; x_dinner] .* [breakfast_energy; lunch_energy; dinner_energy]);

% 中餐能量分配
prob.Constraints.energy_lunch_min = sum(x_lunch .* lunch_energy) >= 0.30 * sum([x_breakfast; x_lunch; x_dinner] .* [breakfast_energy; lunch_energy; dinner_energy]);
prob.Constraints.energy_lunch_max = sum(x_lunch .* lunch_energy) <= 0.40 * sum([x_breakfast; x_lunch; x_dinner] .* [breakfast_energy; lunch_energy; dinner_energy]);

% 晚餐能量分配
prob.Constraints.energy_dinner_min = sum(x_dinner .* dinner_energy) >= 0.30 * sum([x_breakfast; x_lunch; x_dinner] .* [breakfast_energy; lunch_energy; dinner_energy]);
prob.Constraints.energy_dinner_max = sum(x_dinner .* dinner_energy) <= 0.40 * sum([x_breakfast; x_lunch; x_dinner] .* [breakfast_energy; lunch_energy;
dinner_energy]);

% 宏量营养素供能占比约束
total_energy = sum(x_breakfast .* breakfast_energy) + sum(x_lunch .* lunch_energy) + sum(x_dinner .* dinner_energy);
prob.Constraints.protein_min = (sum(x_breakfast .* breakfast_protein) * 4 + sum(x_lunch .* lunch_protein) * 4 + sum(x_dinner .* dinner_protein) * 4) >= 0.10 * total_energy;
prob.Constraints.protein_max = (sum(x_breakfast .* breakfast_protein) * 4 + sum(x_lunch .* lunch_protein) * 4 + sum(x_dinner .* dinner_protein) * 4) <= 0.15 * total_energy;

prob.Constraints.fat_min = (sum(x_breakfast .* breakfast_fat) * 9 + sum(x_lunch .* lunch_fat) * 9 + sum(x_dinner .* dinner_fat) * 9) >= 0.20 * total_energy;
prob.Constraints.fat_max = (sum(x_breakfast .* breakfast_fat) * 9 + sum(x_lunch .* lunch_fat) * 9 + sum(x_dinner .* dinner_fat) * 9) <= 0.30 * total_energy;

prob.Constraints.carbs_min = (sum(x_breakfast .* breakfast_carbs) * 4 + sum(x_lunch .* lunch_carbs) * 4 + sum(x_dinner .* dinner_carbs) * 4) >= 0.50 * total_energy;
prob.Constraints.carbs_max = (sum(x_breakfast .* breakfast_carbs) * 4 + sum(x_lunch .* lunch_carbs) * 4 + sum(x_dinner .* dinner_carbs) * 4) <= 0.65 * total_energy;

% 非产能营养素摄入量相差在±10%之内
% 钙
prob.Constraints.calcium_min = (sum(x_breakfast .* breakfast_calcium) + sum(x_lunch .* lunch_calcium) + sum(x_dinner .* dinner_calcium)) >= 0.9 * 800;
prob.Constraints.calcium_max = (sum(x_breakfast .* breakfast_calcium) + sum(x_lunch .* lunch_calcium) + sum(x_dinner .* dinner_calcium)) <= 1.1 * 800;
```

```
% 铁
prob.Constraints.iron_min = (sum(x_breakfast .* breakfast_iron) + sum(x_lunch .*
lunch_iron) + sum(x_dinner .* dinner_iron)) >= 0.9 * iron_ref;
prob.Constraints.iron_max = (sum(x_breakfast .* breakfast_iron) + sum(x_lunch .*
lunch_iron) + sum(x_dinner .* dinner_iron)) <= 1.1 * iron_ref;

% 锌
prob.Constraints.zinc_min = (sum(x_breakfast .* breakfast_zinc) + sum(x_lunch .*
lunch_zinc) + sum(x_dinner .* dinner_zinc)) >= 0.9 * zinc_ref;
prob.Constraints.zinc_max = (sum(x_breakfast .* breakfast_zinc) + sum(x_lunch .*
lunch_zinc) + sum(x_dinner .* dinner_zinc)) <= 1.1 * zinc_ref;

% 维生素A
prob.Constraints.vitA_min = (sum(x_breakfast .* breakfast_vitA) + sum(x_lunch .*
lunch_vitA) + sum(x_dinner .* dinner_vitA)) >= 0.9 * vitA_ref;
prob.Constraints.vitA_max = (sum(x_breakfast .* breakfast_vitA) + sum(x_lunch .*
lunch_vitA) + sum(x_dinner .* dinner_vitA)) <= 1.1 * vitA_ref;

% 维生素B1
prob.Constraints.vitB1_min = (sum(x_breakfast .* breakfast_vitB1) + sum(x_lunch .*
lunch_vitB1) + sum(x_dinner .* dinner_vitB1)) >= 0.9 * vitB1_ref;
prob.Constraints.vitB1_max = (sum(x_breakfast .* breakfast_vitB1) +
sum(x_lunch .* lunch_vitB1) + sum(x_dinner .* dinner_vitB1)) <= 1.1 * vitB1_ref;

% 维生素B2
prob.Constraints.vitB2_min = (sum(x_breakfast .* breakfast_vitB2) + sum(x_lunch .*
lunch_vitB2) + sum(x_dinner .* dinner_vitB2)) >= 0.9 * vitB2_ref;
prob.Constraints.vitB2_max = (sum(x_breakfast .* breakfast_vitB2) +
sum(x_lunch .* lunch_vitB2) + sum(x_dinner .* dinner_vitB2)) <= 1.1 * vitB2_ref;
```

```
% 维生素C
prob.Constraints.vitC_min = (sum(x_breakfast .* breakfast_vitC) + sum(x_lunch .* lunch_vitC) + sum(x_dinner .* dinner_vitC)) >= 0.9 * 100;
prob.Constraints.vitC_max = (sum(x_breakfast .* breakfast_vitC) + sum(x_lunch .* lunch_vitC) + sum(x_dinner .* dinner_vitC)) <= 1.1 * 100;

% 提供初始点
x0.x_breakfast = ones(N1, 1) * 100; % 初始点，每种早餐食物100g
x0.x_lunch = ones(N2, 1) * 100; % 初始点，每种中餐食物100g
x0.x_dinner = ones(N3, 1) * 100; % 初始点，每种晚餐食物100g

% 求解
options = optimoptions('fmincon', 'Display', 'iter', 'Algorithm', 'interior-point', 'ConstraintTolerance', 1e-6, 'StepTolerance', 1e-10, 'MaxIterations', 1000, 'MaxFunctionEvaluations', 5000, 'UseParallel', true);
[sol, fval, exitflag, output] = solve(prob, x0, 'Options', options);

% 显示结果
disp('Optimal food quantities for breakfast (100g units):');
disp(sol.x_breakfast);
disp('Optimal food quantities for lunch (100g units):');
disp(sol.x_lunch);
disp('Optimal food quantities for dinner (100g units):');
disp(sol.x_dinner);
disp('Optimal AAS:');
disp(-fval);
```

11.2.2 question2_2

```
clear;clc
```

```
load('matlab.mat')
%
% N1, N2, N3分别为早餐、中餐、晚餐的菜品数
N1 = height(breakfast.recipe_name); % 早餐菜品数
N2 = height(lunch.recipe_name); % 中餐菜品数
N3 = height(dinner.recipe_name); % 晚餐菜品数

% 分别定义三餐的营养成分数据
% 早餐数据
breakfast_energy = (breakfast.energy_kcal .* breakfast.avail_part_g) / 100; % 能量 (kcal/100g)
breakfast_protein = (breakfast.protein_g .* breakfast.avail_part_g) / 100; % 蛋白质 (g/100g)
breakfast_fat = (breakfast.fat_g .* breakfast.avail_part_g) / 100; % 脂肪 (g/100g)
breakfast_carbs = (breakfast.carbohydrates_g .* breakfast.avail_part_g) / 100; % 碳水化合物 (g/100g)
breakfast_cost = (breakfast.price_per_part .* breakfast.avail_part_g) / 100; % 费用 (元/100g)
breakfast_essential_aa = ((breakfast.isoleucine_g + breakfast.leucine_g + breakfast.lysine_g + breakfast.total_g + breakfast.valine_g + breakfast.cystine_g + breakfast.tyrosine_g + breakfast.histidine_g + breakfast.arginine_g + breakfast.alanine_g + breakfast.aspartic_acid_g + breakfast.glutamic_acid_g + breakfast.glycine_g) .* breakfast.avail_part_g) / 100; % 必需氨基酸含量
breakfast_reference_protein_aa = ((breakfast.isoleucine_g + breakfast.leucine_g + breakfast.lysine_g + breakfast.methionine_g + breakfast.phenylalanine_g + breakfast.tryptophan_g + breakfast.tryptophan_g + breakfast.valine_g + breakfast.histidine_g) .* breakfast.avail_part_g) / 100;
breakfast_calcium = (breakfast.calcium_mg .* breakfast.avail_part_g) / 100; % 钙 (mg/100g)
```

```
breakfast_iron = (breakfast.iron_mg .* breakfast.avail_part_g) / 100; % 铁 (mg/100g)
breakfast_zinc =   (breakfast.zinc_mg .* breakfast.avail_part_g) / 100; % 锌 (mg/100g)
breakfast_vitA = (breakfast.total_vitaminA_mcg .* breakfast.avail_part_g) / 100; % 维生素A (ug/100g)
breakfast_vitB1 = (breakfast.thiamine_mg .* breakfast.avail_part_g) / 100; % 维生素B1 (mg/100g)
breakfast_vitB2 = (breakfast.riboflavin_mg .* breakfast.avail_part_g) / 100; % 维生素B2 (mg/100g)
breakfast_vitC = (breakfast.vitaminC_mg .* breakfast.avail_part_g) / 100; % 维生素C (mg/100g)

% 中餐数据
lunch_energy = (lunch.energy_kcal .* lunch.avail_part_g) / 100; % 能量 (kcal/100g)
lunch_protein = (lunch.protein_g .* lunch.avail_part_g) / 100; % 蛋白质 (g/100g)
lunch_fat = (lunch.fat_g .* lunch.avail_part_g) / 100; % 脂肪 (g/100g)
lunch_carbs = (lunch.carbohydrates_g .* lunch.avail_part_g) / 100; % 碳水化合物 (g/100g)
lunch_cost = (lunch.price_per_part .* lunch.avail_part_g) / 100; % 费用 (元/100g)
lunch_essential_aa = ((lunch.isoleucine_g + lunch.leucine_g + lunch.lysine_g + lunch.total_g + lunch.valine_g + lunch.cystine_g + lunch.tyrosine_g + lunch.histidine_g + lunch.arginine_g + lunch.alanine_g + lunch.aspartic_acid_g + lunch.glutamic_acid_g + lunch.glycine_g) .* lunch.avail_part_g) / 100; % 必需氨基酸含量
lunch_reference_protein_aa = ((lunch.isoleucine_g + lunch.leucine_g + lunch.lysine_g + lunch.methionine_g + lunch.phenylalanine_g + lunch.tryptophan_g + lunch.tryptophan_g + lunch.valine_g + lunch.histidine_g) .* lunch.avail_part_g) / 100; % 参考蛋白质氨基酸含量
lunch_calcium = (lunch.calcium_mg .* lunch.avail_part_g) / 100; % 钙 (mg/100g)
```

```
lunch_iron = (lunch.iron_mg .* lunch.avail_part_g) / 100; % 铁 (mg/100g)
lunch_zinc =   (lunch.zinc_mg .* lunch.avail_part_g) / 100; % 锌 (mg/100g)
lunch_vitA = (lunch.total_vitaminA_mcg .* lunch.avail_part_g) / 100; % 维生素A (ug/100g)
lunch_vitB1 = (lunch.thiamine_mg .* lunch.avail_part_g) / 100; % 维生素B1 (mg/100g)
lunch_vitB2 = (lunch.riboflavin_mg .* lunch.avail_part_g) / 100; % 维生素B2 (mg/100g)
lunch_vitC = (lunch.vitaminC_mg .* lunch.avail_part_g) / 100; % 维生素C (mg/100g)

% 晚餐数据
dinner_energy = (dinner.energy_kcal .* dinner.avail_part_g) / 100; % 能量 (kcal/100g)
dinner_protein = (dinner.protein_g .* dinner.avail_part_g) / 100; % 蛋白质 (g/100g)
dinner_fat = (dinner.fat_g .* dinner.avail_part_g) / 100; % 脂肪 (g/100g)
dinner_carbs = (dinner.carbohydrates_g .* dinner.avail_part_g) / 100; % 碳水化合物 (g/100g)
dinner_cost = (dinner.price_per_part .* dinner.avail_part_g) / 100; % 费用 (元/100g)
dinner_essential_aa = ((dinner.isoleucine_g + dinner.leucine_g + dinner.lysine_g + dinner.total_g + dinner.valine_g + dinner.cystine_g + dinner.tyrosine_g + dinner.histidine_g + dinner.arginine_g + dinner.alanine_g + dinner.aspartic_acid_g + dinner.glutamic_acid_g + dinner.glycine_g) .* dinner.avail_part_g) / 100; % 必需氨基酸含量
dinner_reference_protein_aa = ((dinner.isoleucine_g + dinner.leucine_g + dinner.lysine_g + dinner.methionine_g + dinner.phenylalanine_g + dinner.tryptophan_g + dinner.tryptophan_g + dinner.valine_g + dinner.histidine_g) .* dinner.avail_part_g) / 100; % 参考蛋白质氨基酸含量
dinner_calcium = (dinner.calcium_mg .* dinner.avail_part_g) / 100; % 钙 (mg/100g)
```

```
dinner_iron = (dinner.iron_mg .* dinner.avail_part_g) / 100; % 铁 (mg/100g)
dinner_zinc =   (dinner.zinc_mg .* dinner.avail_part_g) / 100; % 锌 (mg/100g)
dinner_vitA = (dinner.total_vitaminA_mcg .* dinner.avail_part_g) / 100; % 维生素A (ug/100g)
dinner_vitB1 = (dinner.thiamine_mg .* dinner.avail_part_g) / 100; % 维生素B1 (mg/100g)
dinner_vitB2 = (dinner.riboflavin_mg .* dinner.avail_part_g) / 100; % 维生素B2 (mg/100g)
dinner_vitC = (dinner.vitaminC_mg .* dinner.avail_part_g) / 100; % 维生素C (mg/100g)

% 性别和相应的目标能量
gender = 'male'; % 或 'male'
if strcmp(gender, 'female')
    E_target = 1900;
    iron_ref = 20;
    zinc_ref = 7.5;
    vitA_ref = 700;
    vitB1_ref = 1.2;
    vitB2_ref = 1.2;
else
    E_target = 2400;
    iron_ref = 12;
    zinc_ref = 12.5;
    vitA_ref = 800;
    vitB1_ref = 1.4;
    vitB2_ref = 1.4;
end
```

```
% 决策变量，食物的选择量 (100g 单位)
x_breakfast = optimvar('x_breakfast', N1, 'LowerBound', 0);
x_lunch = optimvar('x_lunch', N2, 'LowerBound', 0);
x_dinner = optimvar('x_dinner', N3, 'LowerBound', 0);

% 构建目标函数，最小化总的用餐费用
cost_total = sum(x_breakfast .* breakfast_cost) + sum(x_lunch .* lunch_cost) +
sum(x_dinner .* dinner_cost);
prob = optimproblem('Objective', cost_total); % 最小化用餐费用

% 总能量摄入约束
prob.Constraints.energy_total_min = sum(x_breakfast .* breakfast_energy) +
sum(x_lunch .* lunch_energy) + sum(x_dinner .* dinner_energy) >= 0.9 * E_target;
prob.Constraints.energy_total_max = sum(x_breakfast .* breakfast_energy) +
sum(x_lunch .* lunch_energy) + sum(x_dinner .* dinner_energy) <= 1.1 * E_target;

% 早餐能量分配
prob.Constraints.energy_breakfast_min = sum(x_breakfast .* breakfast_energy) >=
0.25 * E_target;
prob.Constraints.energy_breakfast_max = sum(x_breakfast .* breakfast_energy) <=
0.35 * E_target;

% 中餐能量分配
prob.Constraints.energy_lunch_min = sum(x_lunch .* lunch_energy) >= 0.30 *
E_target;
prob.Constraints.energy_lunch_max = sum(x_lunch .* lunch_energy) <= 0.40 *
E_target;

% 晚餐能量分配
prob.Constraints.energy_dinner_min = sum(x_dinner .* dinner_energy) >= 0.30 *
```

```
E_target;
prob.Constraints.energy_dinner_max = sum(x_dinner .* dinner_energy) <= 0.40 *
E_target;

% 宏量营养素供能占比约束
total_energy = sum(x_breakfast .* breakfast_energy) + sum(x_lunch .* lunch_energy)
+ sum(x_dinner .* dinner_energy);
prob.Constraints.protein_min = (sum(x_breakfast .* breakfast_protein) * 4 +
sum(x_lunch .* lunch_protein) * 4 + sum(x_dinner .* dinner_protein) * 4) >= 0.10 *
total_energy;
prob.Constraints.protein_max = (sum(x_breakfast .* breakfast_protein) * 4 +
sum(x_lunch .* lunch_protein) * 4 + sum(x_dinner .* dinner_protein) * 4) <= 0.15 *
total_energy;

prob.Constraints.fat_min = (sum(x_breakfast .* breakfast_fat) * 9 + sum(x_lunch .*
lunch_fat) * 9 + sum(x_dinner .* dinner_fat) * 9) >= 0.20 * total_energy;
prob.Constraints.fat_max = (sum(x_breakfast .* breakfast_fat) * 9 + sum(x_lunch .*
lunch_fat) * 9 + sum(x_dinner .* dinner_fat) * 9) <= 0.30 * total_energy;

prob.Constraints.carbs_min = (sum(x_breakfast .* breakfast_carbs) * 4 +
sum(x_lunch .* lunch_carbs) * 4 + sum(x_dinner .* dinner_carbs) * 4) >= 0.50 *
total_energy;
prob.Constraints.carbs_max = (sum(x_breakfast .* breakfast_carbs) * 4 +
sum(x_lunch .* lunch_carbs) * 4 + sum(x_dinner .* dinner_carbs) * 4) <= 0.65 *
total_energy;

% 非产能营养素摄入量相差在±10%之内
% 钙
prob.Constraints.calcium_min = (sum(x_breakfast .* breakfast_calcium) +
sum(x_lunch .* lunch_calcium) + sum(x_dinner .* dinner_calcium)) >= 0.9 * 800;
```

```
prob.Constraints.calcium_max = (sum(x_breakfast .* breakfast_calcium) +
sum(x_lunch .* lunch_calcium) + sum(x_dinner .* dinner_calcium)) <= 1.1 * 800;

% 铁
prob.Constraints.iron_min = (sum(x_breakfast .* breakfast_iron) + sum(x_lunch .*
lunch_iron) + sum(x_dinner .* dinner_iron)) >= 0.9 * iron_ref;
prob.Constraints.iron_max = (sum(x_breakfast .* breakfast_iron) + sum(x_lunch .*
lunch_iron) + sum(x_dinner .* dinner_iron)) <= 1.1 * iron_ref;

% 锌
prob.Constraints.zinc_min = (sum(x_breakfast .* breakfast_zinc) + sum(x_lunch .*
lunch_zinc) + sum(x_dinner .* dinner_zinc)) >= 0.9 * zinc_ref;
prob.Constraints.zinc_max = (sum(x_breakfast .* breakfast_zinc) + sum(x_lunch .*
lunch_zinc) + sum(x_dinner .* dinner_zinc)) <= 1.1 * zinc_ref;

% 维生素A
prob.Constraints.vitA_min = (sum(x_breakfast .* breakfast_vitA) + sum(x_lunch .*
lunch_vitA) + sum(x_dinner .* dinner_vitA)) >= 0.9 * vitA_ref;
prob.Constraints.vitA_max = (sum(x_breakfast .* breakfast_vitA) + sum(x_lunch .*
lunch_vitA) + sum(x_dinner .* dinner_vitA)) <= 1.1 * vitA_ref;

% 维生素B1
prob.Constraints.vitB1_min = (sum(x_breakfast .* breakfast_vitB1) + sum(x_lunch .*
lunch_vitB1) + sum(x_dinner .* dinner_vitB1)) >= 0.9 * vitB1_ref;
prob.Constraints.vitB1_max = (sum(x_breakfast .* breakfast_vitB1) +
sum(x_lunch .* lunch_vitB1) + sum(x_dinner .* dinner_vitB1)) <= 1.1 * vitB1_ref;

% 维生素B2
prob.Constraints.vitB2_min = (sum(x_breakfast .* breakfast_vitB2) + sum(x_lunch .*
lunch_vitB2) + sum(x_dinner .* dinner_vitB2)) >= 0.9 * vitB2_ref;
```

```
prob.Constraints.vitB2_max = (sum(x_breakfast .* breakfast_vitB2) + sum(x_lunch .* lunch_vitB2) + sum(x_dinner .* dinner_vitB2)) <= 1.1 * vitB2_ref;

% 维生素C
prob.Constraints.vitC_min = (sum(x_breakfast .* breakfast_vitC) + sum(x_lunch .* lunch_vitC) + sum(x_dinner .* dinner_vitC)) >= 0.9 * 100;
prob.Constraints.vitC_max = (sum(x_breakfast .* breakfast_vitC) + sum(x_lunch .* lunch_vitC) + sum(x_dinner .* dinner_vitC)) <= 1.1 * 100;

% 提供初始点
x0.x_breakfast = ones(N1, 1) * 100; % 初始点，每种早餐食物100g
x0.x_lunch = ones(N2, 1) * 100; % 初始点，每种中餐食物100g
x0.x_dinner = ones(N3, 1) * 100; % 初始点，每种晚餐食物100g

% 使用 fmincon 求解
options = optimoptions('fmincon', 'Display', 'iter', 'Algorithm', 'interior-point', ...
    'ConstraintTolerance', 1e-6, 'StepTolerance', 1e-10, 'MaxIterations', 1000, ...
    'MaxFunctionEvaluations', 5000, 'UseParallel', true);
[sol, fval, exitflag, output] = solve(prob, x0, 'Options', options);

% 显示结果
disp('Optimal food quantities for breakfast (100g units):');
disp(sol.x_breakfast);
disp('Optimal food quantities for lunch (100g units):');
disp(sol.x_lunch);
disp('Optimal food quantities for dinner (100g units):');
disp(sol.x_dinner);
disp('Total cost:');
disp(fval);
```

11.2.3 question2_3

```
clear;clc

load('matlab.mat')
%
% N1, N2, N3分别为早餐、中餐、晚餐的菜品数
N1 = height(breakfast.recipe_name); % 早餐菜品数
N2 = height(lunch.recipe_name); % 中餐菜品数
N3 = height(dinner.recipe_name); % 晚餐菜品数

% 分别定义三餐的营养成分数据
% 早餐数据
breakfast_energy = (breakfast.energy_kcal .* breakfast.avail_part_g) / 100; % 能量 (kcal/100g)
breakfast_protein = (breakfast.protein_g .* breakfast.avail_part_g) / 100; % 蛋白质 (g/100g)
breakfast_fat = (breakfast.fat_g .* breakfast.avail_part_g) / 100; % 脂肪 (g/100g)
breakfast_carbs = (breakfast.carbohydrates_g .* breakfast.avail_part_g) / 100; % 碳水化合物 (g/100g)
breakfast_cost = (breakfast.price_per_part .* breakfast.avail_part_g) / 100; % 费用 (元/100g)
breakfast_essential_aa = ((breakfast.isoleucine_g + breakfast.leucine_g + breakfast.lysine_g + breakfast.total_g + breakfast.valine_g + breakfast.cystine_g + breakfast.tyrosine_g + breakfast.histidine_g + breakfast.arginine_g + breakfast.alanine_g + breakfast.aspartic_acid_g + breakfast.glutamic_acid_g + breakfast.glycine_g) .* breakfast.avail_part_g) / 100; % 必需氨基酸含量
breakfast_reference_protein_aa = ((breakfast.isoleucine_g + breakfast.leucine_g + breakfast.lysine_g + breakfast.methionine_g + breakfast.phenylalanine_g + breakfast.tryptophan_g + breakfast.tryptophan_g + breakfast.valine_g +
breakfast.histidine_g) .* breakfast.avail_part_g) / 100;
breakfast_calcium = (breakfast.calcium_mg .* breakfast.avail_part_g) / 100; % 钙
(mg/100g)
breakfast_iron = (breakfast.iron_mg .* breakfast.avail_part_g) / 100; % 铁 (mg/100g)
breakfast_zinc =   (breakfast.zinc_mg .* breakfast.avail_part_g) / 100; % 锌
(mg/100g)
breakfast_vitA = (breakfast.total_vitaminA_mcg .* breakfast.avail_part_g) / 100; %
维生素A (ug/100g)
breakfast_vitB1 = (breakfast.thiamine_mg .* breakfast.avail_part_g) / 100; %
维生素B1 (mg/100g)
breakfast_vitB2 = (breakfast.riboflavin_mg .* breakfast.avail_part_g) / 100; %
维生素B2 (mg/100g)
breakfast_vitC = (breakfast.vitaminC_mg .* breakfast.avail_part_g) / 100; %
维生素C (mg/100g)

% 中餐数据
lunch_energy = (lunch.energy_kcal .* lunch.avail_part_g) / 100; % 能量 (kcal/100g)
lunch_protein = (lunch.protein_g .* lunch.avail_part_g) / 100; % 蛋白质 (g/100g)
lunch_fat = (lunch.fat_g .* lunch.avail_part_g) / 100; % 脂肪 (g/100g)
lunch_carbs = (lunch.carbohydrates_g .* lunch.avail_part_g) / 100; % 碳水化合物
(g/100g)
lunch_cost = (lunch.price_per_part .* lunch.avail_part_g) / 100; % 费用 (元/100g)
lunch_essential_aa = ((lunch.isoleucine_g + lunch.leucine_g + lunch.lysine_g +
lunch.total_g + lunch.valine_g + lunch.cystine_g + lunch.tyrosine_g +
lunch.histidine_g + lunch.arginine_g + lunch.alanine_g + lunch.aspartic_acid_g +
lunch.glutamic_acid_g + lunch.glycine_g) .* lunch.avail_part_g) / 100; %
必需氨基酸含量
lunch_reference_protein_aa = ((lunch.isoleucine_g + lunch.leucine_g +
lunch.lysine_g + lunch.methionine_g + lunch.phenylalanine_g + lunch.tryptophan_g
+ lunch.tryptophan_g + lunch.valine_g + lunch.histidine_g) .* lunch.avail_part_g) /
```

```
100; % 参考蛋白质氨基酸含量
lunch_calcium = (lunch.calcium_mg .* lunch.avail_part_g) / 100; % 钙 (mg/100g)
lunch_iron = (lunch.iron_mg .* lunch.avail_part_g) / 100; % 铁 (mg/100g)
lunch_zinc =   (lunch.zinc_mg .* lunch.avail_part_g) / 100; % 锌 (mg/100g)
lunch_vitA = (lunch.total_vitaminA_mcg .* lunch.avail_part_g) / 100; % 维生素A
(ug/100g)
lunch_vitB1 = (lunch.thiamine_mg .* lunch.avail_part_g) / 100; % 维生素B1
(mg/100g)
lunch_vitB2 = (lunch.riboflavin_mg .* lunch.avail_part_g) / 100; % 维生素B2
(mg/100g)
lunch_vitC = (lunch.vitaminC_mg .* lunch.avail_part_g) / 100; % 维生素C
(mg/100g)

% 晚餐数据
dinner_energy = (dinner.energy_kcal .* dinner.avail_part_g) / 100; % 能量
(kcal/100g)
dinner_protein = (dinner.protein_g .* dinner.avail_part_g) / 100; % 蛋白质 (g/100g)
dinner_fat = (dinner.fat_g .* dinner.avail_part_g) / 100; % 脂肪 (g/100g)
dinner_carbs = (dinner.carbohydrates_g .* dinner.avail_part_g) / 100; % 碳水化合物
(g/100g)
dinner_cost = (dinner.price_per_part .* dinner.avail_part_g) / 100; % 费用 (元/100g)
dinner_essential_aa = ((dinner.isoleucine_g + dinner.leucine_g + dinner.lysine_g +
dinner.total_g + dinner.valine_g + dinner.cystine_g + dinner.tyrosine_g +
dinner.histidine_g + dinner.arginine_g + dinner.alanine_g + dinner.aspartic_acid_g +
dinner.glutamic_acid_g + dinner.glycine_g) .* dinner.avail_part_g) / 100; %
必需氨基酸含量
dinner_reference_protein_aa = ((dinner.isoleucine_g + dinner.leucine_g +
dinner.lysine_g + dinner.methionine_g + dinner.phenylalanine_g +
dinner.tryptophan_g + dinner.tryptophan_g + dinner.valine_g + dinner.histidine_g) .*
dinner.avail_part_g) / 100; % 参考蛋白质氨基酸含量
dinner_calcium = (dinner.calcium_mg .* dinner.avail_part_g) / 100; % 钙 (mg/100g)
dinner_iron = (dinner.iron_mg .* dinner.avail_part_g) / 100; % 铁 (mg/100g)
dinner_zinc =   (dinner.zinc_mg .* dinner.avail_part_g) / 100; % 锌 (mg/100g)
dinner_vitA = (dinner.total_vitaminA_mcg .* dinner.avail_part_g) / 100; % 维生素A (ug/100g)
dinner_vitB1 = (dinner.thiamine_mg .* dinner.avail_part_g) / 100; % 维生素B1 (mg/100g)
dinner_vitB2 = (dinner.riboflavin_mg .* dinner.avail_part_g) / 100; % 维生素B2 (mg/100g)
dinner_vitC = (dinner.vitaminC_mg .* dinner.avail_part_g) / 100; % 维生素C (mg/100g)

% 性别和相应的目标能量
gender = 'female'; % 或 'male'
if strcmp(gender, 'female')
    E_target = 1900;
    iron_ref = 20;
    zinc_ref = 7.5;
    vitA_ref = 700;
    vitB1_ref = 1.2;
    vitB2_ref = 1.2;
else
    E_target = 2400;
    iron_ref = 12;
    zinc_ref = 12.5;
    vitA_ref = 800;
    vitB1_ref = 1.4;
    vitB2_ref = 1.4;
end
```

```
% 决策变量，食物的选择量 (100g 单位)
x_breakfast = optimvar('x_breakfast', N1, 'LowerBound', 0);
x_lunch = optimvar('x_lunch', N2, 'LowerBound', 0);
x_dinner = optimvar('x_dinner', N3, 'LowerBound', 0);

% 定义餐费最小化目标
cost_total = sum(x_breakfast .* breakfast_cost) + sum(x_lunch .* lunch_cost) +
sum(x_dinner .* dinner_cost);

% 定义总的蛋白质氨基酸评分最大化目标
AAS_breakfast = (x_breakfast' * breakfast_essential_aa) / (x_breakfast' *
breakfast_reference_protein_aa) * 100;
AAS_lunch = (x_lunch' * lunch_essential_aa) / (x_lunch' *
lunch_reference_protein_aa) * 100;
AAS_dinner = (x_dinner' * dinner_essential_aa) / (x_dinner' *
dinner_reference_protein_aa) * 100;
AAS_total = AAS_breakfast + AAS_lunch + AAS_dinner;

% 定义新的综合目标函数
objective_function = 0.5 * cost_total - 0.5 * AAS_total;

% 设置优化问题
prob = optimproblem('Objective', objective_function); % 综合目标函数

% 总能量摄入约束
prob.Constraints.energy_total_min = sum(x_breakfast .* breakfast_energy) +
sum(x_lunch .* lunch_energy) + sum(x_dinner .* dinner_energy) >= 0.9 * E_target;
prob.Constraints.energy_total_max = sum(x_breakfast .* breakfast_energy) +
sum(x_lunch .* lunch_energy) + sum(x_dinner .* dinner_energy) <= 1.1 * E_target;
```

```
% 早餐能量分配
prob.Constraints.energy_breakfast_min = sum(x_breakfast .* breakfast_energy) >=
0.25 * sum([x_breakfast; x_lunch; x_dinner] .* [breakfast_energy; lunch_energy;
dinner_energy]);
prob.Constraints.energy_breakfast_max = sum(x_breakfast .* breakfast_energy) <=
0.35 * sum([x_breakfast; x_lunch; x_dinner] .* [breakfast_energy; lunch_energy;
dinner_energy]);

% 中餐能量分配
prob.Constraints.energy_lunch_min = sum(x_lunch .* lunch_energy) >= 0.30 *
sum([x_breakfast; x_lunch; x_dinner] .* [breakfast_energy; lunch_energy;
dinner_energy]);
prob.Constraints.energy_lunch_max = sum(x_lunch .* lunch_energy) <= 0.40 *
sum([x_breakfast; x_lunch; x_dinner] .* [breakfast_energy; lunch_energy;
dinner_energy]);

% 晚餐能量分配
prob.Constraints.energy_dinner_min = sum(x_dinner .* dinner_energy) >= 0.30 *
sum([x_breakfast; x_lunch; x_dinner] .* [breakfast_energy; lunch_energy;
dinner_energy]);
prob.Constraints.energy_dinner_max = sum(x_dinner .* dinner_energy) <= 0.40 *
sum([x_breakfast; x_lunch; x_dinner] .* [breakfast_energy; lunch_energy;
dinner_energy]);

% 宏量营养素供能占比约束
total_energy = sum(x_breakfast .* breakfast_energy) + sum(x_lunch .* lunch_energy)
+ sum(x_dinner .* dinner_energy);
prob.Constraints.protein_min = (sum(x_breakfast .* breakfast_protein) * 4 +
sum(x_lunch .* lunch_protein) * 4 + sum(x_dinner .* dinner_protein) * 4) >= 0.10 *
```

```
total_energy;
prob.Constraints.protein_max = (sum(x_breakfast .* breakfast_protein) * 4 + sum(x_lunch .* lunch_protein) * 4 + sum(x_dinner .* dinner_protein) * 4) <= 0.15 * total_energy;

prob.Constraints.fat_min = (sum(x_breakfast .* breakfast_fat) * 9 + sum(x_lunch .* lunch_fat) * 9 + sum(x_dinner .* dinner_fat) * 9) >= 0.20 * total_energy;
prob.Constraints.fat_max = (sum(x_breakfast .* breakfast_fat) * 9 + sum(x_lunch .* lunch_fat) * 9 + sum(x_dinner .* dinner_fat) * 9) <= 0.30 * total_energy;

prob.Constraints.carbs_min = (sum(x_breakfast .* breakfast_carbs) * 4 + sum(x_lunch .* lunch_carbs) * 4 + sum(x_dinner .* dinner_carbs) * 4) >= 0.50 * total_energy;
prob.Constraints.carbs_max = (sum(x_breakfast .* breakfast_carbs) * 4 + sum(x_lunch .* lunch_carbs) * 4 + sum(x_dinner .* dinner_carbs) * 4) <= 0.65 * total_energy;

% 非产能营养素摄入量相差在±10%之内
% 钙
prob.Constraints.calcium_min = (sum(x_breakfast .* breakfast_calcium) + sum(x_lunch .* lunch_calcium) + sum(x_dinner .* dinner_calcium)) >= 0.9 * 800;
prob.Constraints.calcium_max = (sum(x_breakfast .* breakfast_calcium) + sum(x_lunch .* lunch_calcium) + sum(x_dinner .* dinner_calcium)) <= 1.1 * 800;

% 铁
prob.Constraints.iron_min = (sum(x_breakfast .* breakfast_iron) + sum(x_lunch .* lunch_iron) + sum(x_dinner .* dinner_iron)) >= 0.9 * iron_ref;
prob.Constraints.iron_max = (sum(x_breakfast .* breakfast_iron) + sum(x_lunch .* lunch_iron) + sum(x_dinner .* dinner_iron)) <= 1.1 * iron_ref;
```

```
% 锌
prob.Constraints.zinc_min = (sum(x_breakfast .* breakfast_zinc) + sum(x_lunch .*
lunch_zinc) + sum(x_dinner .* dinner_zinc)) >= 0.9 * zinc_ref;
prob.Constraints.zinc_max = (sum(x_breakfast .* breakfast_zinc) + sum(x_lunch .*
lunch_zinc) + sum(x_dinner .* dinner_zinc)) <= 1.1 * zinc_ref;

% 维生素A
prob.Constraints.vitA_min = (sum(x_breakfast .* breakfast_vitA) + sum(x_lunch .*
lunch_vitA) + sum(x_dinner .* dinner_vitA)) >= 0.9 * vitA_ref;
prob.Constraints.vitA_max = (sum(x_breakfast .* breakfast_vitA) + sum(x_lunch .*
lunch_vitA) + sum(x_dinner .* dinner_vitA)) <= 1.1 * vitA_ref;

% 维生素B1
prob.Constraints.vitB1_min = (sum(x_breakfast .* breakfast_vitB1) + sum(x_lunch .*
lunch_vitB1) + sum(x_dinner .* dinner_vitB1)) >= 0.9 * vitB1_ref;
prob.Constraints.vitB1_max = (sum(x_breakfast .* breakfast_vitB1) +
sum(x_lunch .* lunch_vitB1) + sum(x_dinner .* dinner_vitB1)) <= 1.1 * vitB1_ref;

% 维生素B2
prob.Constraints.vitB2_min = (sum(x_breakfast .* breakfast_vitB2) + sum(x_lunch .*
lunch_vitB2) + sum(x_dinner .* dinner_vitB2)) >= 0.9 * vitB2_ref;
prob.Constraints.vitB2_max = (sum(x_breakfast .* breakfast_vitB2) +
sum(x_lunch .* lunch_vitB2) + sum(x_dinner .* dinner_vitB2)) <= 1.1 * vitB2_ref;

% 维生素C
prob.Constraints.vitC_min = (sum(x_breakfast .* breakfast_vitC) + sum(x_lunch .*
lunch_vitC) + sum(x_dinner .* dinner_vitC)) >= 0.9 * 100;
prob.Constraints.vitC_max = (sum(x_breakfast .* breakfast_vitC) + sum(x_lunch .*
lunch_vitC) + sum(x_dinner .* dinner_vitC)) <= 1.1 * 100;
```

```
% 提供初始点
x0.x_breakfast = ones(N1, 1) * 100; % 初始点，每种早餐食物100g
x0.x_lunch = ones(N2, 1) * 100; % 初始点，每种中餐食物100g
x0.x_dinner = ones(N3, 1) * 100; % 初始点，每种晚餐食物100g

% 求解
options = optimoptions('fmincon', 'Display', 'iter', 'Algorithm', 'interior-point', 'ConstraintTolerance', 1e-6, 'StepTolerance', 1e-10, 'MaxIterations', 1000, 'MaxFunctionEvaluations', 5000, 'UseParallel', true);
[sol, fval, exitflag, output] = solve(prob, x0, 'Options', options);

% 显示结果
disp('Optimal food quantities for breakfast (100g units):');
disp(sol.x_breakfast);
disp('Optimal food quantities for lunch (100g units):');
disp(sol.x_lunch);
disp('Optimal food quantities for dinner (100g units):');
disp(sol.x_dinner);
disp('Optimal:');
disp(fval);
```

11.3问题3的程序

```
clear; clc;

load('matlab.mat')
%
% 假设N1, N2, N3分别为早餐、中餐、晚餐的菜品数
N1 = height(breakfast.recipe_name); % 早餐菜品数
N2 = height(lunch.recipe_name); % 中餐菜品数
```

```
N3 = height(dinner.recipe_name); % 晚餐菜品数

% 分别定义三餐的营养成分数据
% 早餐数据
breakfast_energy = (breakfast.energy_kcal .* breakfast.avail_part_g) / 100; % 能量 (kcal/100g)
breakfast_protein = (breakfast.protein_g .* breakfast.avail_part_g) / 100; % 蛋白质 (g/100g)
breakfast_fat = (breakfast.fat_g .* breakfast.avail_part_g) / 100; % 脂肪 (g/100g)
breakfast_carbs = (breakfast.carbohydrates_g .* breakfast.avail_part_g) / 100; % 碳水化合物 (g/100g)
breakfast_cost = (breakfast.price_per_part .* breakfast.avail_part_g) / 100; % 费用 (元/100g)
breakfast_essential_aa = ((breakfast.isoleucine_g + breakfast.leucine_g + breakfast.lysine_g + breakfast.total_g + breakfast.valine_g + breakfast.cystine_g + breakfast.tyrosine_g + breakfast.histidine_g + breakfast.arginine_g + breakfast.alanine_g + breakfast.aspartic_acid_g + breakfast.glutamic_acid_g + breakfast.glycine_g) .* breakfast.avail_part_g) / 100; % 必需氨基酸含量
breakfast_reference_protein_aa = ((breakfast.isoleucine_g + breakfast.leucine_g + breakfast.lysine_g + breakfast.methionine_g + breakfast.phenylalanine_g + breakfast.tryptophan_g + breakfast.tryptophan_g + breakfast.valine_g + breakfast.histidine_g) .* breakfast.avail_part_g) / 100;
breakfast_calcium = (breakfast.calcium_mg .* breakfast.avail_part_g) / 100; % 钙 (mg/100g)
breakfast_iron = (breakfast.iron_mg .* breakfast.avail_part_g) / 100; % 铁 (mg/100g)
breakfast_zinc =   (breakfast.zinc_mg .* breakfast.avail_part_g) / 100; % 锌 (mg/100g)
breakfast_vitA = (breakfast.total_vitaminA_mcg .* breakfast.avail_part_g) / 100; % 维生素A (ug/100g)
breakfast_vitB1 = (breakfast.thiamine_mg .* breakfast.avail_part_g) / 100; %
```

```
维生素B1 (mg/100g)
breakfast_vitB2 = (breakfast.riboflavin_mg .* breakfast.avail_part_g) / 100; % 维生素B2 (mg/100g)
breakfast_vitC = (breakfast.vitaminC_mg .* breakfast.avail_part_g) / 100; % 维生素C (mg/100g)

% 中餐数据
lunch_energy = (lunch.energy_kcal .* lunch.avail_part_g) / 100; % 能量 (kcal/100g)
lunch_protein = (lunch.protein_g .* lunch.avail_part_g) / 100; % 蛋白质 (g/100g)
lunch_fat = (lunch.fat_g .* lunch.avail_part_g) / 100; % 脂肪 (g/100g)
lunch_carbs = (lunch.carbohydrates_g .* lunch.avail_part_g) / 100; % 碳水化合物 (g/100g)
lunch_cost = (lunch.price_per_part .* lunch.avail_part_g) / 100; % 费用 (元/100g)
lunch_essential_aa = ((lunch.isoleucine_g + lunch.leucine_g + lunch.lysine_g + lunch.total_g + lunch.valine_g + lunch.cystine_g + lunch.tyrosine_g + lunch.histidine_g + lunch.arginine_g + lunch.alanine_g + lunch.aspartic_acid_g + lunch.glutamic_acid_g + lunch.glycine_g) .* lunch.avail_part_g) / 100; % 必需氨基酸含量
lunch_reference_protein_aa = ((lunch.isoleucine_g + lunch.leucine_g + lunch.lysine_g + lunch.methionine_g + lunch.phenylalanine_g + lunch.tryptophan_g + lunch.tryptophan_g + lunch.valine_g + lunch.histidine_g) .* lunch.avail_part_g) / 100; % 参考蛋白质氨基酸含量
lunch_calcium = (lunch.calcium_mg .* lunch.avail_part_g) / 100; % 钙 (mg/100g)
lunch_iron = (lunch.iron_mg .* lunch.avail_part_g) / 100; % 铁 (mg/100g)
lunch_zinc =   (lunch.zinc_mg .* lunch.avail_part_g) / 100; % 锌 (mg/100g)
lunch_vitA = (lunch.total_vitaminA_mcg .* lunch.avail_part_g) / 100; % 维生素A (ug/100g)
lunch_vitB1 = (lunch.thiamine_mg .* lunch.avail_part_g) / 100; % 维生素B1 (mg/100g)
```

```
lunch_vitB2 = (lunch.riboflavin_mg .* lunch.avail_part_g) / 100; % 维生素B2
(mg/100g)
lunch_vitC = (lunch.vitaminC_mg .* lunch.avail_part_g) / 100; % 维生素C
(mg/100g)

% 晚餐数据
dinner_energy = (dinner.energy_kcal .* dinner.avail_part_g) / 100; % 能量
(kcal/100g)
dinner_protein = (dinner.protein_g .* dinner.avail_part_g) / 100; % 蛋白质 (g/100g)
dinner_fat = (dinner.fat_g .* dinner.avail_part_g) / 100; % 脂肪 (g/100g)
dinner_carbs = (dinner.carbohydrates_g .* dinner.avail_part_g) / 100; % 碳水化合物
(g/100g)
dinner_cost = (dinner.price_per_part .* dinner.avail_part_g) / 100; % 费用 (元/100g)
dinner_essential_aa = ((dinner.isoleucine_g + dinner.leucine_g + dinner.lysine_g +
dinner.total_g + dinner.valine_g + dinner.cystine_g + dinner.tyrosine_g +
dinner.histidine_g + dinner.arginine_g + dinner.alanine_g + dinner.aspartic_acid_g +
dinner.glutamic_acid_g + dinner.glycine_g) .* dinner.avail_part_g) / 100; %
必需氨基酸含量
dinner_reference_protein_aa = ((dinner.isoleucine_g + dinner.leucine_g +
dinner.lysine_g + dinner.methionine_g + dinner.phenylalanine_g +
dinner.tryptophan_g + dinner.tryptophan_g + dinner.valine_g + dinner.histidine_g) .*
dinner.avail_part_g) / 100; % 参考蛋白质氨基酸含量
dinner_calcium = (dinner.calcium_mg .* dinner.avail_part_g) / 100; % 钙 (mg/100g)
dinner_iron = (dinner.iron_mg .* dinner.avail_part_g) / 100; % 铁 (mg/100g)
dinner_zinc =   (dinner.zinc_mg .* dinner.avail_part_g) / 100; % 锌 (mg/100g)
dinner_vitA = (dinner.total_vitaminA_mcg .* dinner.avail_part_g) / 100; % 维生素A
(ug/100g)
dinner_vitB1 = (dinner.thiamine_mg .* dinner.avail_part_g) / 100; % 维生素B1
(mg/100g)
```

```
dinner_vitB2 = (dinner.riboflavin_mg .* dinner.avail_part_g) / 100; % 维生素B2 (mg/100g)
dinner_vitC = (dinner.vitaminC_mg .* dinner.avail_part_g) / 100; % 维生素C (mg/100g)

% 性别和相应的目标能量
gender = 'female'; % 或 'male'
if strcmp(gender, 'female')
    E_target = 1900;
    iron_ref = 20;
    zinc_ref = 7.5;
    vitA_ref = 700;
    vitB1_ref = 1.2;
    vitB2_ref = 1.2;
else
    E_target = 2400;
    iron_ref = 12;
    zinc_ref = 12.5;
    vitA_ref = 800;
    vitB1_ref = 1.4;
    vitB2_ref = 1.4;
end

% 决策变量，7天内每餐的食物选择量 (100g 单位)
x_breakfast = optimvar('x_breakfast', N1, 7, 'LowerBound', 0);
x_lunch = optimvar('x_lunch', N2, 7, 'LowerBound', 0);
x_dinner = optimvar('x_dinner', N3, 7, 'LowerBound', 0);

% 定义总费用最小化目标
cost_total = sum(sum(x_breakfast .* repmat(breakfast_cost, 1, 7))) + ...
```

```
             sum(sum(x_lunch .* repmat(lunch_cost, 1, 7))) + ...
             sum(sum(x_dinner .* repmat(dinner_cost, 1, 7)));

% 定义总的蛋白质氨基酸评分最大化目标
AAS_breakfast = sum((sum(x_breakfast .* repmat(breakfast_essential_aa, 1, 7)) ./ ...
                    sum(x_breakfast .* repmat(breakfast_reference_protein_aa, 1, 7))) * 100);

AAS_lunch = sum((sum(x_lunch .* repmat(lunch_essential_aa, 1, 7)) ./ ...
                sum(x_lunch .* repmat(lunch_reference_protein_aa, 1, 7))) * 100);

AAS_dinner = sum((sum(x_dinner .* repmat(dinner_essential_aa, 1, 7)) ./ ...
                 sum(x_dinner .* repmat(dinner_reference_protein_aa, 1, 7))) * 100);

AAS_total = AAS_breakfast + AAS_lunch + AAS_dinner;

% 目标1：最大化蛋白质氨基酸评分
prob1 = optimproblem('Objective', -AAS_total); % 最大化AAS, 所以使用负号

% 目标2：最小化总费用
prob2 = optimproblem('Objective', cost_total); % 最小化用餐费用

% 目标3：兼顾蛋白质氨基酸评分及经济性
objective_function = 0.5 * cost_total - 0.5 * AAS_total;
prob3 = optimproblem('Objective', objective_function); % 综合目标函数

% 约束条件
% 总能量摄入约束
energy_total = sum(sum(x_breakfast .* repmat(breakfast_energy, 1, 7))) + ...
              sum(sum(x_lunch .* repmat(lunch_energy, 1, 7))) + ...
```

```
                 sum(sum(x_dinner .* repmat(dinner_energy, 1, 7)));

prob1.Constraints.energy_total_min = energy_total >= 0.9 * E_target;
prob1.Constraints.energy_total_max = energy_total <= 1.1 * E_target;

prob2.Constraints.energy_total_min = prob1.Constraints.energy_total_min;
prob2.Constraints.energy_total_max = prob1.Constraints.energy_total_max;

prob3.Constraints.energy_total_min = prob1.Constraints.energy_total_min;
prob3.Constraints.energy_total_max = prob1.Constraints.energy_total_max;

% 宏量营养素供能占比约束
total_energy = energy_total;

protein_total = sum(sum(x_breakfast .* repmat(breakfast_protein, 1, 7) * 4)) + ...
                sum(sum(x_lunch .* repmat(lunch_protein, 1, 7) * 4)) + ...
                sum(sum(x_dinner .* repmat(dinner_protein, 1, 7) * 4));

prob1.Constraints.protein_min = protein_total >= 0.10 * total_energy;
prob1.Constraints.protein_max = protein_total <= 0.15 * total_energy;

prob2.Constraints.protein_min = prob1.Constraints.protein_min;
prob2.Constraints.protein_max = prob1.Constraints.protein_max;

prob3.Constraints.protein_min = prob1.Constraints.protein_min;
prob3.Constraints.protein_max = prob1.Constraints.protein_max;

fat_total = sum(sum(x_breakfast .* repmat(breakfast_fat, 1, 7) * 9)) + ...
            sum(sum(x_lunch .* repmat(lunch_fat, 1, 7) * 9)) + ...
            sum(sum(x_dinner .* repmat(dinner_fat, 1, 7) * 9));
```

```
prob1.Constraints.fat_min = fat_total >= 0.20 * total_energy;
prob1.Constraints.fat_max = fat_total <= 0.30 * total_energy;

prob2.Constraints.fat_min = prob1.Constraints.fat_min;
prob2.Constraints.fat_max = prob1.Constraints.fat_max;

prob3.Constraints.fat_min = prob1.Constraints.fat_min;
prob3.Constraints.fat_max = prob1.Constraints.fat_max;

carbs_total = sum(sum(x_breakfast .* repmat(breakfast_carbs, 1, 7) * 4)) + ...
              sum(sum(x_lunch .* repmat(lunch_carbs, 1, 7) * 4)) + ...
              sum(sum(x_dinner .* repmat(dinner_carbs, 1, 7) * 4));

prob1.Constraints.carbs_min = carbs_total >= 0.50 * total_energy;
prob1.Constraints.carbs_max = carbs_total <= 0.65 * total_energy;

prob2.Constraints.carbs_min = prob1.Constraints.carbs_min;
prob2.Constraints.carbs_max = prob1.Constraints.carbs_max;

prob3.Constraints.carbs_min = prob1.Constraints.carbs_min;
prob3.Constraints.carbs_max = prob1.Constraints.carbs_max;

% 非产能营养素摄入量约束
nutrient_ref = [800, iron_ref, zinc_ref, vitA_ref, vitB1_ref, vitB2_ref, 100];
nutrient_names = {'calcium', 'iron', 'zinc', 'vitA', 'vitB1', 'vitB2', 'vitC'};
for k = 1:length(nutrient_ref)
    nutrient_total = sum(sum(x_breakfast .* repmat(eval(['breakfast_'
nutrient_names{k}]), 1, 7))) + ...
                     sum(sum(x_lunch .* repmat(eval(['lunch_' nutrient_names{k}]),
1, 7))) + ...
                              sum(sum(x_dinner .* repmat(eval(['dinner_'
nutrient_names{k}]), 1, 7)));
    prob1.Constraints.([nutrient_names{k}, '_min']) = nutrient_total >= 0.9 *
nutrient_ref(k);
    prob1.Constraints.([nutrient_names{k}, '_max']) = nutrient_total <= 1.1 *
nutrient_ref(k);
    prob2.Constraints.([nutrient_names{k}, '_min']) =
prob1.Constraints.([nutrient_names{k}, '_min']);
    prob2.Constraints.([nutrient_names{k}, '_max']) =
prob1.Constraints.([nutrient_names{k}, '_max']);
    prob3.Constraints.([nutrient_names{k}, '_min']) =
prob1.Constraints.([nutrient_names{k}, '_min']);
    prob3.Constraints.([nutrient_names{k}, '_max']) =
prob1.Constraints.([nutrient_names{k}, '_max']);
end

% 提供初始点
x0.x_breakfast = ones(N1, 7) * 100; % 初始点，每种早餐食物100g
x0.x_lunch = ones(N2, 7) * 100; % 初始点，每种中餐食物100g
x0.x_dinner = ones(N3, 7) * 100; % 初始点，每种晚餐食物100g

% 求解优化问题
options = optimoptions('fmincon', 'Display', 'iter', 'Algorithm', 'interior-point',
'ConstraintTolerance', 1e-6, 'StepTolerance', 1e-10, 'MaxIterations', 1000,
'MaxFunctionEvaluations', 5000, 'UseParallel', true);

% 求解目标1：最大化蛋白质氨基酸评分
[sol1, fval1, exitflag1, output1] = solve(prob1, x0, 'Options', options);
```

```
% 求解目标2：最小化总费用
[sol2, fval2, exitflag2, output2] = solve(prob2, x0, 'Options', options);

% 求解目标3：兼顾蛋白质氨基酸评分及经济性
[sol3, fval3, exitflag3, output3] = solve(prob3, x0, 'Options', options);

% 输出目标1
disp('Optimal food quantities for breakfast (100g units) for max AAS:');
disp(sol1.x_breakfast);
disp('Optimal food quantities for lunch (100g units) for max AAS:');
disp(sol1.x_lunch);
disp('Optimal food quantities for dinner (100g units) for max AAS:');
disp(sol1.x_dinner);
disp('Optimal AAS:');
disp(-fval1);

% 输出目标2
disp('Optimal food quantities for breakfast (100g units) for min cost:');
disp(sol2.x_breakfast);
disp('Optimal food quantities for lunch (100g units) for min cost:');
disp(sol2.x_lunch);
disp('Optimal food quantities for dinner (100g units) for min cost:');
disp(sol2.x_dinner);
disp('Total cost:');
disp(fval2);

% 输出目标3
disp('Optimal food quantities for breakfast (100g units) for balanced objective:');
disp(sol3.x_breakfast);
disp('Optimal food quantities for lunch (100g units) for balanced objective:');
disp(sol3.x_lunch);

disp('Optimal food quantities for dinner (100g units) for balanced objective:');

disp(sol3.x_dinner);

disp('Objective value:');

disp(fval3);
```