北京航空航天大学

线性规划模型的
问题分析技巧及应用

第一作者：张作舟 19373260
第二作者：黄泽桓 19373257
第三作者：郭瑞 19373180

目录

摘要

线性规划问题中有两大重要的研究方向，一类是 0-1 型整数线性规划，另一类是多目标规划法。本文围绕着两个核心问题，以数学建模教材中的两道具体问题为例，分别对基础问题按照一般分析问题的步骤步骤进行了建模解决；并重点对两大类规划问题进行了详细的应用总结：分别重点分析探讨了一类问题的分析技巧与方法，并举例概括了其典型的应用场景。

关键词：线性规划、0-1 规划、多目标规划法、数学建模

一、0-1 型整数规划问题

1.1 问题背景

在线性规化问题中，0-1 型整数线性规划是整数线性规划中的特殊情形，0-1 型整数规划的变量 x_i 仅取值 0 或 1。这时称 x_i 为 0-1 变量，或称二进制变量。0-1 型整数规划代表着一类普遍问题，在决策分析中，我们常常需要借助 0-1 型整数规划数学模型来计算最优决策方案。

此部分将以数学建模教材[1]线性规划章节的课后第二小题为例来分析 0-1 型整数规划的问题分析技巧及应用。

1.2 基本问题的解决

1.2.1 问题描述

一家出版社准备在某地向七个区大学生供应图书,每个区的大学生数量如图所示（单位：千人）,出版社准备在该市设立两个图书代理销售点,每个代理点只能向该地区和一个相邻的地区售书,出版社知道售书覆盖的人群越大,所获得的利润也就大,所以出版社要选择两个恰当的代理销售点使覆盖的人群最大。求解合适的代理销售点。

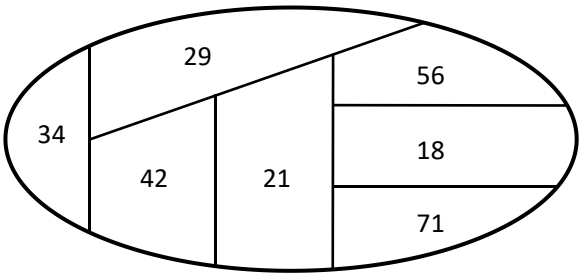

图 1 题一图

1.2.2 符号定义

表 1 题一符号定义表

符号	含义
i、j	唯一标识地区的编号，范围为 1~7
x_{ij}	0-1 变量，为 1 时 i、j 两区建立代售点，为 0 不建立
w_{ij}	地区组合 i、j 的价值（权重），即 i、j 两区人数和

1.2.3 模型假设

1. 每个城市最多只能建立一个代售点，且每个城市最多只能接受一个城市的售书。

2. 这七个区没有人员流动且不考虑迁入迁出等动态变化。

3. 不考虑供给与需求等经济学因素。即书的供应量远远满足学生的需求；且人人的消费能力、需求相等；每本书的价格相同。

1.2.4 模型建立

为将问题规范化、抽象化，可将地区进行编号由 1-7，并且根据地区的相邻关系先将题图转换为一般的具有顶点与边的相邻关系图，转换后如图 2。

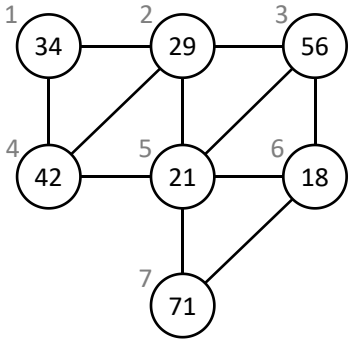

图 2 题一转换图

按照地区相邻才可建立销售点的条件，首先可以筛选出可以建立代售点的地区组合 x_{ij}，组合种类数即图中边数；并计算出此组合的价值 w_{ij}，即边的邻接顶点数值之和，也即该边的边权。

表 2 题一组合权重表

组合	x_{12}	x_{14}	x_{23}	x_{24}	x_{25}	x_{35}	x_{36}	x_{45}	x_{56}	x_{57}	x_{67}
价值	63	76	85	71	50	77	74	63	39	92	89

1. 决策变量：表 2 中的 0-1 变量组合

2. 目标函数：

$$\max W = \sum x_{ij} w_{ij} \qquad (1)$$

3. 约束条件：

（1）代售点数目不超过 2

$$\sum x_{ij} \le 2 \qquad (2)$$

（2）每个区售书点至多为 1

$$\forall i \quad \sum_j x_{ij} + \sum_j x_{ji} \le 1 \qquad (3)$$

（3）0-1 变量约束

$$x_{ij} \in \{0,1\} \qquad (4)$$

整理列出约束条件不等式组（5）

$$\begin{cases} x_{12}+x_{14}+x_{23}+x_{24}+x_{25}+x_{35}+x_{36}+x_{45}+x_{56}+x_{57}+x_{67} \le 2 \\ x_{12}+x_{14} \le 1 \\ x_{12}+x_{23}+x_{24}+x_{25} \le 1 \\ x_{23}+x_{35}+x_{36} \le 1 \\ x_{14}+x_{24}+x_{45} \le 1 \\ x_{25}+x_{35}+x_{45}+x_{56}+x_{57} \le 1 \\ x_{36}+x_{56}+x_{67} \le 1 \\ x_{57}+x_{67} \le 1 \\ x_{ij}=0 \text{或} x_{ij}=1 \end{cases} \qquad (5)$$

在此条件下求

$$\max \quad 63x_{12}+76x_{14}+85x_{23}+71x_{24}+50x_{25}+77x_{35}+74x_{36}+63x_{45}+39x_{56}+92x_{57}+89x_{67}$$

1.2.5 模型求解

利用 LINGO 软件编辑代码[1]，得到运行结果如图 3。

```
Global optimal solution found.
Objective value:                              177.0000
Objective bound:                              177.0000
Infeasibilities:                              0.000000
Extended solver steps:                               0
Total solver iterations:                             0


                     Variable           Value        Reduced Cost
                          X12        0.000000           -63.00000
                          X14        0.000000           -76.00000
                          X23        1.000000           -85.00000
                          X24        0.000000           -71.00000
                          X25        0.000000           -50.00000
                          X35        0.000000           -77.00000
                          X36        0.000000           -74.00000
                          X45        0.000000           -63.00000
                          X56        0.000000           -39.00000
                          X57        1.000000           -92.00000
                          X67        0.000000           -89.00000
```

图 3 题一运行结果图

[1] LINGO 代码见附录 1

可得应在区 2、3 和区 5、7 建立图书代售点，可得到最大的覆盖人群数 177 千人。

1.2.6 模型分析

在解决此决策问题时，先抽象出了题目要求转换后的权重图，并为不同组合设置了 0-1 变量。通过分析目标函数并规定约束条件，建立了线性规划决策模型。该模型的正确性由约束的完备性和 LINGO 软件的正确性保证。

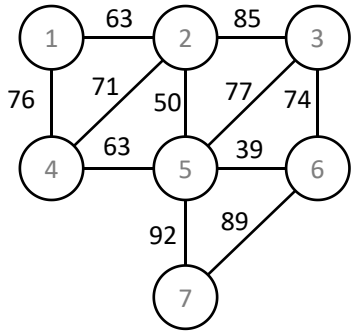

图 4 题一进一步抽象图

为进一步验证模型的正确性，可在抽象后的权重图基础上进一步分析问题。我们首先可以将顶点的价值加和成为边权值，得到进一步抽象的题图 4。此时问题转化为图论问题：寻找两条不相邻的边，使其边权和最大。可编写简单的 Java 程序[2]解决此图论问题。输出结果如图 5 所示。

```
第一个代售点：区2 区3
第二个代售点：区5 区7
最大覆盖人数：177千人
```

图 5 题一运行结果图

程序遍历搜边的结果与线性规划计算结果完全一致，进一步验证了模型的正确性。

1.3 0-1 规划的应用总结

1.3.1 0-1 规划的现实意义

整数规划是线性规化在实际应用中最常见的问题，而 0—1 规划在整数规划占有举足轻重的地位，许多决策问题，例如背包问题、指派问题等都可归结为此类规划；此外，在界定一些变量的约束时，我们可以将其转化为 0-1 规划，而且多种非线性规划问题表示成 0-1 型整数规划问题。

[2] Java 代码见附录 2

1.3.2 0-1 规划现实问题应用

应用场景 1 背包问题

给定若干件物品，已知单件物品的重量为c_i以及价值w_i，现有一负重为C的背包，问该如何取舍，使得背包内物品价值总额最大。

对于类似的背包问题，我们可以简单地为每一件物品设置一个 0-1 变量，用以标识其是否被选取。

$$x_i=\begin{cases}0，\text{不选取}i\text{号物品}\\1，\text{选取i号物品}\end{cases}$$

至此目标函数及约束条件可简单列出。即在约束条件组合（1）下求$\max W=\sum_i x_i w_i$。

$$\begin{cases}\sum_i x_i c_i<C\\x_i=0\text{或}1\end{cases}\qquad(1)$$

应用场景 2 指派问题

有m个人分别完成n项任务中的其中一项。因工作性质和个人专长的差异，不同人和工作组合的完成时间有所差异，i完成任务j的时间为w_{ij}。求总工时最小的工作组合。

对于此类问题，我们同样可以为人和工作的组合设置 0-1 变量，来标识此组合是否被采取。

$$x_{ij}=\begin{cases}0，i\text{不负责完成任务}j\\1，i\text{负责完成任务}j\end{cases}$$

与背包问题不同的是，约束条件中需体现出一个人只能完成一项工作、一项工作只能由一人完成的约束条件组合（7）。在此约束条件下求$\max W=\sum_i\sum_j x_{ij}w_{ij}$。

$$\begin{cases}\sum_i x_{ij}=1\\\sum_j x_{ij}=1\\x_{ij}=0\text{或}1\end{cases}\qquad(2)$$

应用总结

此外应用场景还有很多，这里不一一列出。需要明确的是，我们解决的基本问题在经转化后亦是在一定约束条件下求组合的最大价值问题，与背包问题、指派问题等都存在共性。

0-1 规划问题万变不离其宗，核心永远是 0-1 变量的设计。我们常常将一些决策变量设置为 0-1 变量，借此数量化地描述诸如开与关、取与弃、有与无等现象所反映的离散变量间的逻辑关系、顺序关系以及互斥的约束条件。

1.3.3 0-1 变量的使用技巧

使用技巧 1 互斥约束转化

现实问题中，对于一些变量通常存在复杂的范围界定，此时可以借助 0-1 变量来描述约束条件。

例如，实际问题中可能存在这样一种形式的约束 $f(x_1,x_2,...)$，其取值域或者为 D_1，或者为 D_2。普通的线性规划并不擅长解决“或”类型约束条件的问题。一种方案我们可以列出不同条件下的约束条件分别做线性规划，并取满足题设的最优解。但在选择分支增加时此法书写计算起来会非常麻烦，不是我们期望的解决方案。另一种解决方案则是借助 0-1 变量，将“或”条件通过映射关系进行转化，列在一组式中解决问题。

为了方便描述，我们首现假设 $D_1=[A,B]$，$D_2=[C,D]$，期望设计这样一个 0-1 变量 b，当其取值为 0 时，$f(x_1,x_2,...)\in D_1$；取值为 1 时，$f(x_1,x_2,...)\in D_2$。

以 $A\le f\le B$ 为基础，引入 0-1 变量 b 和未知数 M 、N，修饰不等式如下：

$$A+bM\le f\le B+bN \qquad (1)$$

为达到要求，未知数 M 、N 需要满足 $b=1$ 时：

$$\begin{cases}A+M=C\\B+N=D\end{cases} \qquad (2)$$

解得未知数 M 、N 代入原式即得我们需要的约束

$$A+b(C-A)\le f\le B+b(D-B) \qquad (3)$$

重新结合整理得最终约束组（4）：

$$\begin{cases}(1-b)A+bC\le f(x_1,x_2,...)\le(1-b)B+bD\\b=0或1\end{cases} \qquad (4)$$

特别的，当存在无界的取值域时，例如 $D_2=[C,+\infty]$，则可在式（4）取 $D>>C$ 即可，类似的情况同理。

有了如上约束关系转化，我们可以方便地利用 LINGO 中通过@bin(b)语句设置 0-1 变量，在一组式中解决类似“或”条件约束问题。

使用技巧 2 简单的非线性约束转化

在某些些情况下变量的约束不是一次的，LINGO 软件在计算此类约束时开销较大，但可以通过 0-1 变量将此类非线性约束问题转换为线性约束。

为简单、讨论方便起见，以单一变量二次约束条件为例讨论此类约束问题。

（1）不考虑无解的情况，对于形如 $ax^2+bx+c<0(a>0)$ 的约束，我们可以解出其零点 $x_1<x_2$，则可简单地将此约束转换为

$$x_1 \le x \le x_2$$

（2）而对于形如 $ax^2+bx+c>0(a>0)$ 的约束，同样计算出二次函数零点 $x_1<x_2$，可得约束为 $x \le x_1$ 或 $x \ge x_2$。此时利用技巧 1 的结论，将此约束转换为

$$\begin{cases}(1-b)M+bx_2 \le x \le (1-b)x_1+bN \\ b=0\text{或}1 \\ M << x_1 \\ N >> x_2\end{cases}$$

技巧总结

在此部分内容中，0-1 变量不再具有实际含义，而是仅仅作为一种标识符，通过巧妙的设计 0-1 变量并建立映射关系，可以将线性规划问题的条件化繁为简，大大提高了解决实际问题的容易程度和效率。

二、多目标规划问题

2.1 问题背景

我们知道，线性规划只能解决一组线性约束条件下一个目标的最大值或最小值问题。而在实际决策中，衡量方案优劣要考虑多个目标，在这些目标中，有主要的也有次要的，有最大值的也有最小值的，有定量的也有定性的，有相互补充的也有相互对立的，对于这些问题线性规划则无能为力。目标规划方法是目前解决多目标规划问题的方法之一，这种方法的基

本思想是对每一个目标函数预先给定一个期望值，在现有的约束条件下，这组期望值也许能够达到，也许达不到，我们的任务是求出尽可能接近这组预定期望值的解。

此部分将以数学建模教材线性规划章节的课后第十三小题为例来分析多目标规划的问题分析技巧及应用。

2.2 基本问题的解决

2.2.1 问题描述

某农场有 3 万亩农田（1 亩=666.67㎡），准备种植玉米、大豆、小麦 3 种农作物。每种农作物每亩需施化肥分别为 0.12t，0.20t，0.15t。预计秋后玉米每亩可收获 500kg，售价为 1.2 元/kg；大豆每亩可收获 200kg，售价为 6 元/kg；小麦每亩可收获 300kg，售价为 3.5 元/kg。农场年初规划时依次考虑以下几个方面：

1) 年终总收益尽量不低于 1650 万元

2) 总产量尽量不低于 1.25×10^4t

3) 小麦产量以 0.5×10^4t 为宜

4) 大豆产量尽量不低于 0.2×10^4t

5) 玉米产量尽量不超过 0.6×10^4t

6) 农场目前能够提供 5000t 化肥，若不够，可额外购买，但希望额外购买量越少越好

请为该农场制定种植计划。

表 3 题二信息表

比较项 / 农作物	每亩须化肥/t	亩产量/kg	每千克售价/元
玉米	0.12	500	1.2
大豆	0.20	200	6
小麦	0.15	300	3.5

2.2.2 模型假设

1. 假设农作物的收成既定，不受自然因素、人为因素等影响。

2. 假设农作物价格不收市场影响，不考虑供给与需求等经济学因素。

2.2.3 符号定义

表 4 题二符号定义表

符号	含义
x_1	玉米耕地面积，单位亩
x_2	大豆耕地面积，单位亩
x_3	小麦耕地面积，单位亩
d_i^+	正偏差变量，非负
d_i^-	负偏差变量，非负

2.2.4 模型建立

记x_1，x_2，x_3分别为玉米、大豆、小麦的耕地面积，单位为亩，是这个问题的决策变量（非负整数）。

首先考虑要求 1）：该农场年终总收益为：1.2 × 500 x_1 +6 × 200 x_2 +3.5 × 300 x_3 =600 x_1 +1200 x_2 +1050 x_3，这个数额尽可能不低于 1650 万，引入非负变量d_1^+和d_1^-，其中d_1^+表示总收益超过目标值的部分，d_1^-表示总收益低于目标值的部分，则有

$$600x_1+1200x_2+1050x_3-d_1^++d_1^-=16500000 \qquad (1)$$

要求 1）希望年终总收益尽量不低于 1650 万元，也就是d_1^-尽可能小。

类似考虑要求 2）：记d_2^+表示总产量超过目标值的部分，d_2^-表示总产量低于目标值的部分，则有

$$500x_1+200x_2+300x_3-d_2^++d_2^-=12500000 \qquad (2)$$

要求 2）希望总产量尽量不低于 1.25×10^4t，也就是d_2^-尽可能小。

对于要求 3）：记d_3^+表示小麦产量超过目标值的部分，d_3^-表示小麦产量低于目标值的部分，则有

$$300x_3-d_3^++d_3^-=5000000 \qquad (3)$$

要求 3）希望小麦产量以 0.5×10^4t 为宜，也就是$d_3^-+d_3^+$尽可能小。

对于要求 4）：记 d_4^+ 表示大豆产量超过目标值的部分，d_4^- 表示大豆产量低于目标值的部分，则有

$$200x_2 - d_4^+ + d_4^- = 2000000 \qquad (4)$$

要求 4）希望大豆产量尽量不低于 0.2×10^4t，也就是 d_4^- 尽可能小。

对于要求 5）：记 d_5^+ 表示玉米产量超过目标值的部分，d_5^- 表示玉米产量低于目标值的部分，则有

$$500x_1 - d_5^+ + d_5^- = 6000000 \qquad (5)$$

要求 5）希望玉米产量尽量不超过 0.6×10^4t，也就是 d_5^+ 尽可能小。

对于要求 6）：记 d_6^+ 表示化肥使用量超过目标值的部分，d_6^- 表示化肥使用量低于目标值的部分，则有

$$0.12x_1 + 0.2x_2 + 0.15x_3 - d_6^+ + d_6^- = 5000 \qquad (6)$$

要求 6）希望化肥使用量不超过 5000t，也就是 d_6^+ 尽可能小。

该农场规划需要依次考虑上述的 6 个要求，记 P_1、P_2、P_3、P_4、P_5 和 P_6 分别为要求 1），2），3），4），5），6）的优先级，则可以认为 $P_1 \geq P_2 \geq P_3 \geq P_4 \geq P_5 \geq P_6 > 0$。此时，可将该农场最后的决策目标表达成

$$\min z = P_1d_1^- + P_2d_2^- + P_3(d_3^- + d_3^+) + P_4d_4^- + P_5d_5^+ + P_6d_6^+ \qquad (7)$$

加上所有变量的非负约束

$$x_1, x_2, x_3 \geq 0, d_i^+, d_i^- \geq 0, i = 1,2,3,4,5,6 \qquad (8)$$

再加上总面积为 3 万亩的刚性约束，有：

$$x_1 + x_2 + x_3 \leq 30000 \qquad (9)$$

于是问题总结为优化模型（1）~（9），是一个多目标规划模型，归纳整理如下：

$$\min z = P_1d_1^- + P_2d_2^- + P_3(d_3^- + d_3^+) + P_4d_4^- + P_5d_5^+ + P_6d_6^+$$

s.t.

$$\begin{cases} x_1+x_2+x_3\le 30000 \\ 600x_1+1200x_2+1050x_3-d_1^++d_1^-=16500000 \\ 500x_1+200x_2+300x_3-d_2^++d_2^-=12500000 \\ 300x_3-d_3^++d_3^-=5000000 \\ 200x_2-d_4^++d_4^-=2000000 \\ 500x_1-d_5^++d_5^-=6000000 \\ 0.12x_1+0.2x_2+0.15x_3-d_6^++d_6^-=5000 \\ x_1,x_2,x_3\ge 0, d_i^+,d_i^-\ge 0, i=1,2,3,4,5,6 \end{cases}$$

2.2.5 模型求解

首先以d_1^-最小为目标，（1）~（7）和（9）为约束求解，利用LINGO软件求解，得d_1^-最小值为0，即可以保证达到第一目标。

将d_1^-=0加入约束，以d_2^-最小为目标，利用LINGO软件求解，可求解其最小值也为0，可以同时保证达到第二目标。

将d_2^-=0加入约束，以$d_3^++d_3^-$最小为目标，利用LINGO软件求解，可求解其最小值为1250000。

将$d_3^++d_3^-$=1250000加入约束，以d_4^-最小为目标，利用LINGO软件求解，可求解其最小值为2000000。

将d_4^-=2000000加入约束，以d_5^+最小为目标，利用LINGO软件求解，可求解其最小值为2750000。

将d_5^+=2750000加入约束，以d_6^+最小为目标，利用LINGO软件求解，可求解其最小值为0，可以达到第六目标。

对应的最优解为$x_1=17500$，$x_2=0$，$x_3=12500$即农场规划为种植1.75万亩玉米，种植1.25万亩小麦，不种植大豆。

2.2.6 模型分析

在解决上述多目标规划问题中，我们采用目标规划方法。首先，从问题所给出的六个要求出发，建立相应的最优约束和目标函数。其次，对每一个目标函数预先给定一个期望值，依次考虑各个变量的最优解，并结合相关约束进行求解，得到尽可能接近预定期望值的解。

可以看到，多目标规划中的每一个目标都会存在正负偏差，解决此类问题无法保证各个

偏差均等于 0，但当我们将目标的优先级进行排序依次考虑时，即指定偏差系数间的远大于关系，就能够分步求解出满足指定目标的最优解及相应偏差。此处的偏差可以理解为为了满足优先级更高的目标的一种妥协，牺牲低级目标的效益以满足高级目标的效益最大化。

2.3 多目标规划总结

2.3.1 模型的一般表达式

求 $x=(x_1,x_2,\cdots,x_n)$，使模型的理想目标达到极大或极小，同时满足模型的现实目标和硬约束的要求，其数学表达式为：

理想目标

$$\max\left(a_{r,1}x_1+a_{r,2}x_2+\cdots+a_{r,n}x_n\right),\forall r$$

$$\min\left(a_{s,1}x_1+a_{s,2}x_2+\cdots+a_{s,n}x_n\right),\forall s$$

现实目标和硬约束

$$a_{t,1}x_1+a_{t,2}x_2+\cdots+a_{t,n}x_b*b,\forall t$$

$$x_1,x_2,\cdots,x_n\geq 0$$

上式中，r 表示极大化理想目标个数，s 表示极小化理想目标个数，t 表示现实目标和刚性约束个数，*表示大小关系符号。

对于每个目标函数确定一个希望达到的期望值就是理想目标，由于受到各种条件的限制，这些目标值往往不能全部都达到，现实中要达到的数值 b 就是现实目标。约束和现实目标的表达形式相同，但现实目标是弹性的，而约束是相对的。因此，当一个现实目标必须满足时，称之为硬约束。

2.3.2 建模步骤

1）选定设计变量。多目标规划中，要求确定的未知量可用一组基本参数来表示。变量的选取是在求解模型过程中不断优化变更的，用向量 $x=(x_1,x_2,\cdots,x_n)$表示。

2）确立目标函数。理想目标由数学函数表示，目标函数表示决策者的某种愿望，如最

大利润或最小成本等，用$\max f(x)$或$\min f(x)$表示。

3）建立约束条件。约束与现实目标的表达形式相似，只是在求解模型时必须满足条件。

4）数学模型。使用数学表达式来描述决策问题，通过化多为少的方法、或分层序列法等，将多目标直接或间接地转化为较容易求解的单目标问题，并使用 LINGO 进行求解。

2.3.3 应用场合

多目标规划问题，一般应用于多目标、多因素、多期望的场合。

经济学

在经济学中，许多问题涉及多目标，例如最大化消费者对各种商品的需求，将取决于这些商品所产生的功能和具有的价值，且受制于这些商品的成本和价格等。加之消费者对各种商品目标的首选消费（期望）是相互矛盾的，因此可使用目标规划法，表示消费者的偏好和预算约束等，从而研究消费者面临的消费权衡。

最佳控制

在工程方面，许多问题不能描述为最好或最坏，相反，每个目标都有一个理想的目标值，并且希望每个元素尽可能接近每个目标的期望值。例如　　，能源系统通常在性能和成本之间取舍，应用程序在运行效率和要求的硬件算力之间取舍，社会工程则综合考虑质量、时间、人力、效力等多因素。

通常，这些问题受线性等式约束或现实约束，从而阻止所有目标同时达到期望。我们通常使用多目标二次目标函数，建立与客观物体的距离与其理想值的二次方成正相关的关系。由于这类问题通常受到时间因素影响，各类变量和期望会随着时间点的变动而发生改变，因此采用跨期优化技术[2]。

最佳设计

在设计领域中，如何平衡设计的好坏，是最佳设计的关键问题之一[3]。而良好的设计通常涉及多个标准（目标），如资本成本、投资、运营成本、利润、质量、产品回收率，效率、工艺安全性、操作时间等。例如，在设计造纸厂时，可以减少造纸厂投入的资金，同时提高纸张的质量。则该多目标规划问题，将可以包括如最小化质量参数的预期变化、最小化预期休息时间、最小化存储量的投资成本等目标。

2.3.4 现实意义

在现实生活中，我们遇到的问题往往不会是单一的，而是受到多种因素的影响，且人的期望大多并非一成不变，而是允许结果不断地靠近期望，不断地往我们希望的方向进步。而这种问题，在数学中便体现为多目标规划问题——通过做出合理的决策，得到距期望最近的最优解。

在上述应用场合部分，我们也能看到，多目标规划在资源分配、计划编制、生产调度等方面是具有一定的作用。尽管有些理论问题尚在探讨之中，但作为一种决策问题，它的应用场景还是十分乐观的。企业决策者掌握和运用这种方法，将有助于提高管理和决策水平。

参考文献

[1] 姜启源，谢金星，叶俊.数学模型[M].第 5 版.北京：高等教育出版社，2018.5：1-18.
[2] 姜杰. 煤矿跨期生产规模的优化研究[D]. 西安科技大学，2014.
[3] 涂春鸣，罗安，刘娟. 无源滤波器的多目标优化设计[D]. ，2002.

附录

附录 1 题一 LINGO 代码段

```
model:
max = 63*x12 + 76*x14 + 85*x23 + 71*x24 +
      50*x25 + 77*x35 + 74*x36 + 63*x45 +
      39*x56 + 92*x57 + 89*x67;
x12+x14+x23+x24+x25+x35+x36+x45+x56+x57+x67 <= 2;
x12+x14 <= 1;
x12+x23+x24+x25 <= 1;
x23+x35+x36 <= 1;
x14+x24+x45 <= 1;
x25+x35+x45+x56+x57 <= 1;
x36+x56+x67 <= 1;
x57+x67 <= 1;
@bin(x12);@bin(x14);@bin(x23);@bin(x24);
@bin(x25);@bin(x35);@bin(x36);@bin(x45);
@bin(x56);@bin(x57);@bin(x67);
end
```

附录 2 题一 Java 代码段

```java
import java.util.ArrayList;
import java.util.Scanner;

class Map {
    ArrayList<Edge> edges;

    public int numEdge() {
        return edges.size();
    }

    public Map(ArrayList<Edge> edges) {
        this.edges = edges;
    }
}

class Edge {
    int node1;
    int node2;
    int weight;

    Edge(int node1, int node2, int weight) {
        this.node1 = node1;
        this.node2 = node2;
        this.weight = weight;
    }

    public static boolean isAdjoin(Edge edge1, Edge edge2) {
        return edge1.node1 == edge2.node1 || edge1.node1 == edge2.node2
                || edge1.node2 == edge2.node1 || edge1.node2 == edge2.node2;
    }
}

public class Test {
    public static void main(String[] args) {
        Scanner scanner = new Scanner(System.in);
        // 建图
        ArrayList<Edge> edges = new ArrayList<>();
        while (scanner.hasNextInt()) {
            int n1 = scanner.nextInt();
            int n2 = scanner.nextInt();
            int w =scanner.nextInt();
            Edge edge = new Edge(n1, n2, w);
            edges.add(edge);
```

```
        }
        Map map = new Map(edges);
        // 遍历搜边
        int max = 0;
        Edge e1 = map.edges.get(0);
        Edge e2 = map.edges.get(1);
        for (Edge edge1 : map.edges) {
            for (Edge edge2 : map.edges) {
                if (!Edge.isAdjoin(edge1, edge2)) {
                    if (edge1.weight + edge2.weight > max) {
                        max = edge1.weight + edge2.weight;
                        e1 = edge1;
                        e2 = edge2;
                    }
                }
            }
        }
        // 输出结果
        System.out.println("第一个代售点: " + "区" + e1.node1 +" " + "区" + e1.node2);
        System.out.println("第二个代售点: " + "区" + e2.node1 +" " + "区" + e2.node2);
        System.out.println("最大覆盖人数: " + max + "千人");
    }
}
```

附录 3 题二 LINGO 代码段

代码段 1

```
model:
sets:
variable/1..3/:x;!规定变量;
s_con_num/1..6/:target,dplus,dminus;!软约束条件个数以及相关参数;
s_con(s_con_num,variable):c;!软约束系数;
endsets
data:
target = 16500000 12500000 5000000 2000000 6000000 5000;
c = 600 1200 1050 500 200 300 0 0 300 0 200 0 500 0 0 0.12 0.2 0.15;
enddata
min = dminus(1);!第一个目标函数;
x(1) + x(2) + x(3) < 30000;!硬约束;
@for(s_con_num(i):@sum(variable(j):c(i,j)*x(j))+dminus(i)-dplus(i)=target(i));!软约束表达式;
@for(variable:@gin(x));!限制变量为整数;
end
```

代码段 2

```
model:
sets:
variable/1..3/:x;!规定变量;
s_con_num/1..6/:target,dplus,dminus;!软约束条件个数以及相关参数;
s_con(s_con_num,variable):c;!软约束系数;
endsets
data:
target = 16500000 12500000 5000000 2000000 6000000 5000;
c = 600 1200 1050 500 200 300 0 0 300 0 200 0 500 0 0 0.12 0.2 0.15;
enddata
min = dminus(2);!第一个目标函数;
x(1) + x(2) + x(3) < 30000;!硬约束;
@for(s_con_num(i):@sum(variable(j):c(i,j)*x(j))+dminus(i)-dplus(i)=target(i));!软约束表达式;
dminus(1) = 0;!代入一级目标的解;
@for(variable:@gin(x));!限制变量为整数;
end
```

代码段 3

```
model:
sets:
variable/1..3/:x;!规定变量;
s_con_num/1..6/:target,dplus,dminus;!软约束条件个数以及相关参数;
s_con(s_con_num,variable):c;!软约束系数;
```

```
endsets
data:
target = 16500000 12500000 5000000 2000000 6000000 5000;
c = 600 1200 1050 500 200 300 0 0 300 0 200 0 500 0 0 0.12 0.2 0.15;
enddata
min = dminus(3) + dplus(3);!第三个目标函数;
x(1) + x(2) + x(3) < 30000;!硬约束;
@for(s_con_num(i):@sum(variable(j):c(i,j)*x(j))+dminus(i)-dplus(i)=target(i));!软约束表达式;
dminus(1) = 0;!代入一级目标的解;
dminus(2) = 0;!代入二级目标的解;
@for(variable:@gin(x));!限制变量为整数;
end
```

代码段 4

```
model:
sets:
variable/1..3/:x;!规定变量;
s_con_num/1..6/:target,dplus,dminus;!软约束条件个数以及相关参数;
s_con(s_con_num,variable):c;!软约束系数;
endsets
data:
target = 16500000 12500000 5000000 2000000 6000000 5000;
c = 600 1200 1050 500 200 300 0 0 300 0 200 0 500 0 0 0.12 0.2 0.15;
enddata
min = dminus(4);!第一个目标函数;
x(1) + x(2) + x(3) < 30000;!硬约束;
@for(s_con_num(i):@sum(variable(j):c(i,j)*x(j))+dminus(i)-dplus(i)=target(i));!软约束表达式;
dminus(1) = 0;!代入一级目标的解;
dminus(2) = 0;!代入二级目标的解;
dminus(3) + dplus(3) = 1250000;!代入二级目标的解;
@for(variable:@gin(x));!限制变量为整数;
end
```

代码段 5

```
model:
sets:
variable/1..3/:x;!规定变量;
s_con_num/1..6/:target,dplus,dminus;!软约束条件个数以及相关参数;
s_con(s_con_num,variable):c;!软约束系数;
endsets
data:
```

```
target = 16500000 12500000 5000000 2000000 6000000 5000;
c = 600 1200 1050 500 200 300 0 0 300 0 200 0 500 0 0 0.12 0.2 0.15;
enddata
min = dplus(5);!第一个目标函数;
x(1) + x(2) + x(3) < 30000;!硬约束;
@for(s_con_num(i):@sum(variable(j):c(i,j)*x(j))+dminus(i)-dplus(i)=target(i));!软约束表达式;
dminus(1) = 0;!代入一级目标的解;
dminus(2) = 0;!代入二级目标的解;
dminus(3) + dplus(3) = 1250000;!代入三级目标的解;
dminus(4) = 2000000;!代入四级目标的解;
@for(variable:@gin(x));!限制变量为整数;
end
```

代码段 6

```
model:
sets:
variable/1..3/:x;!规定变量;
s_con_num/1..6/:target,dplus,dminus;!软约束条件个数以及相关参数;
s_con(s_con_num,variable):c;!软约束系数;
endsets
data:
target = 16500000 12500000 5000000 2000000 6000000 5000;
c = 600 1200 1050 500 200 300 0 0 300 0 200 0 500 0 0 0.12 0.2 0.15;
enddata
min = dplus(6);!第一个目标函数;
x(1) + x(2) + x(3) < 30000;!硬约束;
@for(s_con_num(i):@sum(variable(j):c(i,j)*x(j))+dminus(i)-dplus(i)=target(i));!软约束表达式;
dminus(1) = 0;!代入一级目标的解;
dminus(2) = 0;!代入二级目标的解;
dminus(3) + dplus(3) = 1250000;!代入三级目标的解;
dminus(4) = 2000000;!代入四级目标的解;
dplus(5) =2750000;!代入五级目标的解;
@for(variable:@gin(x));!限制变量为整数;
end
```

附录 4 题二 LINGO 最终结果截图

```
Global optimal solution found.
Objective value:                              0.000000
Objective bound:                              0.000000
Infeasibilities:                              0.000000
Extended solver steps:                               0
Total solver iterations:                             0


                     Variable           Value        Reduced Cost
                         X( 1)        17500.00            0.000000
                         X( 2)        0.000000            0.000000
                         X( 3)        12500.00            0.000000
                    TARGET( 1)       0.1650000E+08        0.000000
                    TARGET( 2)       0.1250000E+08        0.000000
                    TARGET( 3)        5000000.            0.000000
                    TARGET( 4)        2000000.            0.000000
                    TARGET( 5)        6000000.            0.000000
                    TARGET( 6)        5000.000            0.000000
                     DPLUS( 1)        7125000.            0.000000
                     DPLUS( 2)        0.000000            0.000000
                     DPLUS( 3)        0.000000            0.000000
                     DPLUS( 4)        0.000000            0.000000
                     DPLUS( 5)        2750000.            0.000000
                     DPLUS( 6)        0.000000            1.000000
                    DMINUS( 1)        0.000000            0.000000
                    DMINUS( 2)        0.000000            0.000000
                    DMINUS( 3)        1250000.            0.000000
                    DMINUS( 4)        2000000.            0.000000
                    DMINUS( 5)        0.000000            0.000000
                    DMINUS( 6)        1025.000            0.000000
                      C( 1, 1)        600.0000            0.000000
                      C( 1, 2)        1200.000            0.000000
                      C( 1, 3)        1050.000            0.000000
                      C( 2, 1)        500.0000            0.000000
                      C( 2, 2)        200.0000            0.000000
                      C( 2, 3)        300.0000            0.000000
                      C( 3, 1)        0.000000            0.000000
                      C( 3, 2)        0.000000            0.000000
                      C( 3, 3)        300.0000            0.000000
                      C( 4, 1)        0.000000            0.000000

                      C( 4, 2)        200.0000            0.000000
                      C( 4, 3)        0.000000            0.000000
                      C( 5, 1)        500.0000            0.000000
                      C( 5, 2)        0.000000            0.000000
                      C( 5, 3)        0.000000            0.000000
                      C( 6, 1)       0.1200000            0.000000
                      C( 6, 2)       0.2000000            0.000000
                      C( 6, 3)       0.1500000            0.000000

                          Row    Slack or Surplus      Dual Price
                            1        0.000000           -1.000000
                            2        0.000000            0.000000
                            3        0.000000            0.000000
                            4        0.000000            0.000000
                            5        0.000000            0.000000
                            6        0.000000            0.000000
                            7        0.000000            0.000000
                            8        0.000000            0.000000
                            9        0.000000            0.000000
                           10        0.000000            0.000000
                           11        0.000000            0.000000
                           12        0.000000            0.000000
                           13        0.000000            0.000000
```