模型-运筹学-规划论-整数规划【czy】

1.模型名称

整数规划（Integer Programming,IP)

2. 适用范围

我们把要求一部分或者全部决策变量必须取整数值的规划问题称为整数规划。如生产汽车的数量，用人单位招聘员工的数量。

3. 常见类型

3.1 0-1型整数规划

3.1.1 概念

0-1型整数规划是整数规划的特例,其数学模型的目标函数、约束条件与线性规划相同,不同的是其变量只能取0和1,分别表示两种截然相反的结果。

3.1.2 适用范围

如土木工程系统的最优工程配置问题,城建规划中的居民点、给水点、加油站和商业网点的最优布局问题

3.1.3 模型表达式

目标函数：$max(或min)Z=\sum_{j=0}^{n}c_jx_j$ 约束条件：

$$\begin{cases}\sum_{j=i}^{n}a_{ij}x_j\leq（或\ \geq）b_i,(i=1,2,\cdots,m)\\x_j=0或1,(j=1,2,\cdots,n)\end{cases}$$

3.1.4 求解0-1问题通用方法

3.1.4.1 穷举法

由于0-1型整数规划的变量个数有限且取值非0即1,所以不难将解的集合找出来,再检验每个解的可行性,.凡符合全部约束条件者均为可行解,通过比较目标函数的值便可找到最优解,这个解法称为穷举法。但当变量和约束条件很多时,其工作量是非常大的。

3.1.4.2 隐枚举法i

1. 概念

隐枚举法i是以穷举法为基础,通过建立过滤条件而使计算工作量大为减少的方法。

2. 步骤

1. 用试算法找到一个可行解
2. 增加一个约束条件，即**过滤条件**
3. 对解集中的解逐个检验过滤条件,若不符合则直接淘汰该解
4. 对剩下的解**穷举法**

3. 实例

$$max\ Z=3X_1-2X_2+5X_3\qquad s.t.\begin{cases}X_1+2X_2-X_3\leq2\\X_1+4X_2+X_3\leq4\\X_1+X_2\leq3\\4X_1+X_3\leq6\\X_1,X_2,X_3=0或1\end{cases}$$

过滤条件：可知最优解的目标函数一定不小于5（因为是求极大化问题）

本例中有3个变量,共有$2^3=8$个解需要检验,本例通过建立过滤条件,只计算了18次就找到了最优解,而用穷举法需要计算40次,技术工作量大大减少了。如在求解过程发现更好的可行解及时更换过滤条件,计算工作量还可进一步减少。

4. 解法评价

- 优点:其数学模型无须转化为标准型,减少了人工处理的工作量。
- 缺点:需要检验的方案较多,而且在集体操作中存在两个明显的不确定性因素,直接影响到本解法对减少工作量的有效性:
    1. 该方法要求先求出一个可行解,但并没有给出求第一个可行解的方法。当约束方程较多,不可能一眼看出哪一个变量组合是可行解时,为建立过滤条件需要通过试算找到一个可行解,但经过多少次试算才能找到一个可行解,这是无法预知的。
    2. 虽不管通过多少次试算总可找到一个可行解,并据以建立过滤条件,但所建立的过滤条件,有效性如何又是难以确定的。如通过试算所找到的可行解的Z值是全部可行解中Z 值较小或最小的,那么,所建立的过滤条件必然是很弱的,甚至起不到过滤作用。

3.1.4.3 隐枚举法ii

1. 概念

先将数学模型改造为标准型,然后按一定顺序检验解集中各解的可行性,力争只检验较少的解便找到最优解。 此类解法对标准型的要求基本相同,但在确定**检验顺序**上有不同的处理,大致分为两种。

1. 将检验的解点排列成树枝状,称为隐枚举树。(分枝定界法的思路)
2. 通过适当编排解集中各变量分别取值0或1的各种组合的检验顺序,使之符合相应的目标函数值从小到大或从大到小的顺序。(以下)

2. 步骤

1. 将数学模型改造为标准型

对标准型要求目标函数:

$$\min\ Z=\sum_{j=1}^{n}c_jx_j \qquad s.t.\begin{cases}Q_i=\pm b_i\pm\sum_{j=1}^{n}a_{ij}x_j,(i=1,2,\cdots,m)\\ c_j\geq 0,(j=1,2,\cdots,n)\\ x_j=0\text{或}1,(j=1,2,\cdots,n)\end{cases}$$

2. 令各变量均取0值,使 Z 最小。计算值,检验其可行性。
3. 若不是可行解,出发分枝,使某一变量由0变为1(可使 c_j 值小的 x_j 先由0变为1)

   通过使某个变量从0到1的转换使目标函数值从最小开始逐渐增加,并同时使解点与可行解的距离缩短而逐步过滤到可行解。

   用B 表示与可行解的距离,可得:

$$B=-\sum_{i=1}^{m}Q_i(Q_i<0)$$

当$B=0$ 时即为可行解。因为检验的顺序是从 Z 值最小的解点开始并按照Z值递增的顺序,故在检验中发现的第一个可行解即为最优解。

4. 直到停止分枝,已获得本题的最优解。

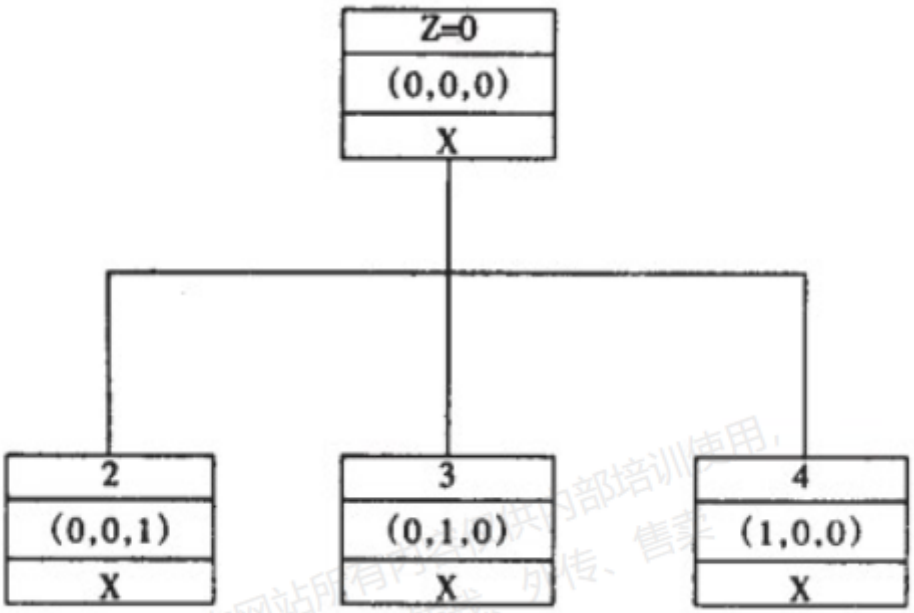
   分枝图上的每一个结点代表解集中的一个解。由于这种分枝的办法是使取值为1的变量在原有的基础上增加而全部 c_j 值都为正值,故分枝后所形成的子问题的Z 值不可能小于原问题的Z值,故出现以下3种情况时不再分枝:

   1. 出现可行解。
   2. 虽为非可行解,但其值已不小于其它分枝上可行解的 Z 值。
   3. 解点为非可行解,其Z值仍小于其它分枝上可行解的 Z 值,但分枝后不可能存在可行解的情况,这种情况称为非可行子域。但这第三种情况只有在极特殊的情况下才能判定。

3. 实例

$$\min\ Z=4X_1-3X_2+2X_3 \qquad s.t.\begin{cases}2X_1-5X_2+3X_3\geq 4\\4X_1+X_2+3X_3\leq 3\\X_2+X_3\leq 1\\X_j=0\text{或}1(j=1,2,3)\end{cases}$$

2. 令各变量均取0值,使 Z 最小。计算值,检验其可行性。
3. 发现不是可行解。从出发分枝,以使某一变量由0变为1,分枝图如下:



4. 从图中的三个一级分枝子问题来看,分别属于步骤4的前两种情况,故停止分枝,已获得本题的最优解。

4. 解法评价

- 优点:所需检验的方案较少。
- 缺点:检验之前必须通过人工处理将数学模型标准化,当问题复杂时,人工处理的工作量很大,易出差错。
- 在具体操作过程中还存在如下问题,即此解法有效性的关键在于检验前必须排出一个能体现各解点 Z 值大小顺序的对各变量取值0或1的各种组合的检验顺序,当变量个数较多的时候,这是相当困难的。这种迂回排序方法实际工作量往往超过穷举法的工作量。穷举法来求解需要检验 $2n$ 个变量的组合,当 n 值很大时不可行。但事实上,用隐枚举法所检验的组合数往往会超过 $2n$ 个。

3.1.4.4 隐枚举法iii——目标排序法

1. 概念

结合上述两种解法的优点并克服两者的缺陷,就会得到一种明显地优于前两种解法的新的解法,这就是隐枚举法III——目标排序法。

2. 步骤

1. 求出解集中各解点的Z值
2. 将Z值按大小排序
3. 最后**按序检验**每个解的可行性。

如求极大化则从**Z** 值最大的解开始检验,如求极小化则从**Z** 值最小的解开始检验。在检验中发现的**第一个**可行解即为最优解。

3. 实例

表 1　隐枚举法 III 对例 1 的求解过程

解点	Z值	值序(大→小)	约束条件 ①	②	③	④	是否符合是(√)否(×)
(0,0,0)	0	7					
(0,0,1)	5	3					
(0,1,0)	−2	8					
(1,0,0)	3	4					
(1,0,1)	8	1	√	√	√	√	√
(0,1,1)	3	5					
(1,1,0)	1	6					
(1,1,1)	6	2					

表 2　隐枚举法 III 对例 2 的求解过程

解点	Z值	值序(大→小)	约束条件 ①	②	③	是否符合是(√) 否(×)
(0,0,0)	0	1	√	×		×
(0,0,1)	2	2	√	√	√	√
(0,1,0)	3	3				
(1,0,0)	4	4				
(1,1,0)	7	7				
(1,0,1)	6	6				
(0,1,1)	5	5				
(1,1,1)	9	8				

4. 评价

非常简便而有效。无须将数学模型转化为标准型,它保留了隐枚举法i的优点而克服了隐枚举法ii的缺陷;它不需要过滤条件,既避免了通过试算寻求可行解的麻烦,也略去了用过滤条件检验每个目标函数的工作,自然也就排除了两个不确定性因素的干扰。

它只需要在隐枚举法i求出解集中各解点 Z 值的基础上将 Z 值按大小排个顺序,然后按Z值顺序检验相应各解的可行性,就可以确保通过检验的第一个可行解即为最优解。 这种直接以Z值大小编排检验顺序的解法,保持了隐枚举法ii按序检验的优点,同时又排除了迂回排序和重复检验带来的困扰。更重要的是,这种新的解法省略了所有需要人工处理的环节,对实施验算极为有利。

3.2 分枝定界法

3.2.1 基本思路和参数说明

设最大化的整数规划问题为A，相应的不含整数约束的线性规划为B。

若B的最优解不符合A的整数条件,那么B的最优目标函数值必为A的最优目标函数值Z^*的一个上界,记作Z^+

而A的任意可行解的目标函数值将是Z^* 的一个下界,记作Z^-

对B的非整数解的**相邻整数**作**附加条件**,从而形成两个分枝,即两个子问题,两个子问题的可行域中包含原整数规划问题的所有可行解。不断分枝,逐步减小,Z^+增大,最Z^-终求得Z^*。

3.2.2 适用范围

纯整数或混合的整数规划问题。

3.2.3 步骤

1. 先不考虑整数约束,变成一般线性规划问题,求出最优解,记为$X*_{(0)}$
2. 若$X*_{(0)}$刚好是整数解,则它就是原整数规划的最优解;否则,转入下一步
3. 对原问题进行分枝寻求整数最优解。

   选取非整数解$X*_{(0)}$ 的一个非整数分量$x_i^* = b_i$,以该非整数分量的相邻整数$[b_i]$和$[b_i]+1$为边界,将原问题分枝为两个子问题,并抛弃这两个整数之间的非整数区域:

   > 在原线性规划模型中添加分枝约束$x_i \le [b_i]$, 构成第一个子问题。

   > 在原线性规划模型中添加分枝约束$x_i \ge [b_i]+1$, 构成第二个子问题。

4. 对上面两个子问题求出最优解。若某个子问题的解是整数解,则停止该子问题的分枝,并且把它的目标函数值与上一步求得的最优整数解对应的目标函数值比较以决定取舍;否则,对该子问题继续进行分枝。
5. 重复第三、第四步直至获得原问题最优整数解为止。

经验表明,优先选择较小的、不符合整数要求的变量进行分枝;并在可能的情况下,根据对实际问题的了解,事先选择一个合理的"界限",可以更好地发挥分枝定界法的优点。

3.2.4 实例

求: $\max Z = 40x_1 + 90x_2, \text{s.t. } 9x_1 + 7x_2 \le 56, 7x_1 + 20x_2 \le 70, \text{其中} \, x_1, x_2 \in Z_+$

解:

1. 忽略$x_1, x_2 \in Z_+$条件,求得$x_1 = 4.81 x_2 = 1.82$,Z的初始上界$Z_0 = 356$,它是最大的。但不满足$x_1, x_2 \in Z_+$, 所以该问题没有结束。(当然,如果Z 没有上界,说明无解,也应当结束) 接着应确定Z的下界Z_0。若容易求得一整数解,则可将对应的值作为初始下界Z_0。反之可以令$Z_0 = -\infty$或待分枝定界法求出一个整数可行解后再给出,其作用是解的目的仅在求得比Z_0 更好的目标值。此例有$x = (0,0)^T$故取Z_0
2. 分出各枝并求出相应各枝的解。根据经验应先选择较小者,取$x_2 = 1.82$, 在原问题上增加约束分枝$x_2 \le 1$形成一分枝,类似的加上$x2 \ge 2$形成又一分枝。接着就来求解分枝。在求的过程中可能会遇到下列情形:

   a.无可行解，不再分枝。

   b.得到整数解，该枝停止。

   c.得到非整数解。若$Z < Z_0$还应续分枝。

   当一对分枝都需要继续分解时,对极大化而言,一般将目标函数值较小的一枝暂时放下,留待以后处理而沿着另一枝继续分解下去,直到搜寻完毕,然后将那些留待处理的分枝按照"后进先出"原则,依次取出进行搜寻。
3. 修改原来的上、下界。

   a.修改下界Z

   Z一般是迄今为止最好的整数可行解相应的目标函数值,因而每求出一个新的整数解,都应将其对应的Z值与前面的比较,谁大就取谁。下界Z的值在求解过程中不断增大。

   b. 修改上界Z

   每求完一对分枝,都要考虑一下修改Z,新的Z应该比原始问题的小些,而且是目前为止所有未被分枝的问题的目标函数值中最大的一个,在求解过程中Z不断变小。

求解过程见下面的“树枝”形图:

 屏幕截图 2022-01-28 150752

故本例最优解为$x_1^* = 4.00, x_2^* = 2.00, Z^* = 340$。

3.2.5 评价

- 优点:“分枝”为整数规划最优解的出现创造条件,而“定界”则提高了搜索的效率。
- 缺点:必须在每个节点解一个完整的线性规划。在大型问题中这将很费时,但在目前来讲,它在解决实践问题上还是最有效的,但不是说每个整数规划都能用它来解。

3.3 割平面法

3.3.1 基本思路

首先不考虑变量xi是整数这一条件,不断增加线性约束条件(几何术语称为割平面)将原规划问题的可行域切割掉一部分,使其切割掉的部分只包含非整数解,没有切割掉任何整数可行解,直到切割后得到的可行域有一个整数坐标的极点恰好是问题的最优解为止[1]。下面介绍**分数算法**。

3.3.2 适用条件

它的基本要求是每一个约束条件的系数和右边的常数都必须为整数。如$x_1 + 13x_2 \leq 13$必须将它变为$6x_1 + 2x_2 \leq 39$ 才能进行下一步操作。割平面法对具有特殊结构的整数规划问题是很有效的。

3.3.3 步骤

问题:求$max\ x_0 = x_1 + 2x_2, x_1 + x_2/2 \leq 13/4, x_1, x_2 \in Z_+$

1. 忽略整数条件的约束,当作一般线性问题求解。如求得最优解则结束,否则转下一步。
2. 添加辅助约束条件。如何添加是割平面法的核心问题。其方法为:假设上一步的最后表格如下:

    屏幕截图 2022-02-02 1

x_i表示基变量ω_j表示非基变量。对x_i取非整数值的第i个方程即$x_i + \sum \alpha_i^j \omega_j = \beta_i$有:
$x_i = \beta_i - \sum \alpha_j^i \omega_j (\beta_i为分数)$。称这样的行为来源行。令$\beta_i = [\beta_i] + f_i,\ \alpha_i^j = [\alpha_i^j] + f_{ij}$,显然$0 < f_i < 1,\ 0 \leq f_{ij} < 1$,那么来源行变为:

$$f_i - \sum_{j=1}^{n} f_{ij}\omega_j = x_i - [\beta_i] + \sum_{j=1}^{n} [\alpha_{ji}]\omega_j$$

要满足所有的变量取整,则其左、右两边为整数,所以$f_1 - \sum_{j=1}^{n} f_{ij}\omega_j < f_i < 1$,显然$0 \leq S_i$(分数切割)$S_i$按规定应是正整数且为松驰变量。故当$\omega_j = 0$时$S_i = -f_i < 0$ 不可行,它表明得到的解不满足新的约束条件,如何才能消除这种不可行性呢?

对偶单纯形法提供了可靠的保障。它能消除这种不可行性。这也是割平面法与分枝定界法在解题过程中差异的所在。因此在上表中加入分数之后新的表格为:

 屏幕截图 2022-02-02 2

3. 从上表中又可构成一个新的分数切割,并用对偶单纯形法来消除不可行性,重复下去,直到求出结果为止。

3.3.4 实例

 屏幕截图 2022-02-02 3

1. 按一般线性规划求得下表:

    屏幕截图 2022-02-02 4

$x_1 = 29, x_2 = 27$不是整数转入下一步。

2. 一般来说:对应一个非整数解的任何约束方程均可以选来作为切割方程,但据经验对极大化问题常取对应于$\max(f_i)$的方程,因为它对目标函数相对来讲影响略大,此例中$f_1 = f_2 = \frac{1}{2}$,所以选谁都一样就取x_2得到$x_2 + \frac{7}{22}x_3 + \frac{1}{22}x_4 = 3\frac{1}{2}$对应的分数切割为$S_1 - \frac{7}{22}x_3 - \frac{1}{22}x_4 = -\frac{1}{2}$有表如图27:

    屏幕截图 2022-02-025

其解仍不为整数,所以应构造一个新的切割方程,把x1的方程写为$x_1 + \frac{1}{7}x_4 + (-1 + \frac{6}{7})S_1 = 4 + \frac{4}{7}$得$S_2 - \frac{1}{7}x_4 - \frac{6}{7}S_1 = -\frac{4}{7}$转入下步。

3. 由上得表如图29:

基	x_0	x_1	x_2	x_3	x_4	S_1	S_2	右边
x_0	1	0	0	0	1	8	0	59
x_2	0	0	1	0	0	1	0	3
x_1	0	1	0	0	1/7	-1/7	0	44/7
x_3	0	0	0	1	1/7	-22/7	0	4/7
S_2	0	0	0	0	-1/7	-6/7	1	-4/7

由对偶单纯形法解得$x_0 = 55, x_1 = 4, x_2 = 3$,达到目的,结束求解。

5.Matlab实现

```
%使用intlinprog函数求解混合整数规划问题
f = [-3;-2;-1]; %目标函数
intcon = 3;  %整数变量的下标
% 线性限制条件
A = [1,1,1];
b = 7;
% 等式条件
Aeq = [4,2,1];
beq = 12;
% 上下界限制
lb = zeros(3,1);
ub = [Inf;Inf;1]; % 强迫 x(3) 为0-1变量
x = intlinprog(f,intcon,A,b,Aeq,beq,lb,ub)\
```

```
% max f(x)=3x1+5x2
% s.t.  x1       <=4
%            2x2 <=12
%      3x1 + 2x2 <=16    18改为16,顶点不再是整数
%       x1       >=0
%             x2 >=0
% x1 x2 均为int
f=[3 5];
A=[1 0;0 2;3 2];
```

```
b=[4;12;16];
Aeq=[];
beq=[];
lb=[0 0]';
ub=[];
intcon=[1 2]; % x1 x2均为整数
[x1,fvallin] = linprog(-f ,A,b,Aeq,beq,lb,ub);
[x2,fvalint,exitflag]= intlinprog(-f,intcon,A,b,Aeq,beq,lb,ub);

% 设一下lb,ub即可求解0-1规划, 看0-1背包问题(用整数规划很容易推广到有限背包)
% max f(x)= \sum v_i * x_i = v * x
% s.t.   \sum w_i * x_i= w * x  <=W
% OJ3030
N=4;W=100;
f2=[100 1 3 3]; %v_i
A2= [99 101 97 2];  %w_i
b2=[W];
Aeq=[];
beq=[];
lb=zeros(N,1)';
ub=ones(N,1);
intcon=[1 2 3 4];
[x,value]= intlinprog(-f2,intcon,A2,b2,Aeq,beq,lb,ub);

% OJ3024
N=3;W=9;
f2=[3 4 5]; %v_i
A2= [2 3 4];  %w_i
b2=[W];
Aeq=[];
beq=[];
lb=zeros(N,1)';
ub=[1;2;3];
intcon=[1 2 3];
[xx,value2]= intlinprog(-f2,intcon,A2,b2,Aeq,beq,lb,ub);
```

6. 参考资料

1. 数学建模培训营----整数规划

---

1. 针对求解的决策变量是全部取整还是部分取整问题,割平面算法中就分成了两种计算方法,前者称为**分数算法**,后者称为**混合整数算法**,本处着重探讨分数算法。 ↩