分支定界法 Matlab 程序实现与验证

为了更深入理解分支定界法计算流程，从而决定花费几天时间仔细学习该算法，并编写出该算法的 Matlab 计算程序。同时为了后面个人的借鉴学习，编写本文档。在进行分支定界法计算程序编写过程中，通过网络搜索，发现了 Matlab2014 版之后嵌入了混合整数线性规划求解函数 intlinprog，从而也将该函数的使用方法撰写下来。

1 整数规划问题简介

在线性规划问题中，有些最优解可能是分数或小数，但对于某些具体问题，常有要求解答必须是整数的情形(称为整数解)。例如：所求解是机器的台数、完成工作的人数或装货的车数等，分数或小数的解答就不合要求。为了满足整数解的要求，初看起来，似乎只要把已得到的带有分数或小数的解经过“舍入化整”就可以了。但这常常是不行的，因为化整后不见得是可行解；或虽是可行解，但不一定是最优解。因此，对求最优整数解的问题，有必要另行研究。人们称这样的问题为整数规划(Integer Programming，IP)，整数规划是最近几十年发展起来的规划论中的一个分支。

整数规划中如果所有的变数都限制为(非负)整数，就称为纯整数规划(Pure Integer Programming，PIP)或称为全整数规划(All Integer Programming，AIP)；如果仅一部分变数限制为整数，则称为混合整数计划(Mixed Integer Programming，MIP)。整数规划的一种特殊情形是 0-1 规划，该规划中变量的取值仅限于 0 或 1，指派问题就是一类典型的 0-1 规划问题。

现举例说明用前述单纯形法求得的解不能保证是整数最优解。

例 1：某厂拟用集装箱托运甲乙两种货物，每箱的体积、重量、可获利润以及托运所受限制如表 1 所示。问两种货物各托运多少箱，可使获得利润为最大?

表 1　货物托运示例数据

货物	体积（m^3/箱）	重量（百公斤/箱）	利润（百元/箱）
甲	5	2	20
乙	4	5	10
托运限制	24（m^3）	13 百公斤	

设 x_1、x_2 分别为甲、乙两种货物的托运箱数（为非负整数），列该问题的纯整数规划模型如下：

$$\max \quad z = 20x_1 + 10x_2$$

$$\begin{cases} 5x_1+4x_2\le 24 \\ 2x_1+5x_2\le 13 \\ x_1,x_2\ge 0 \\ x_1,x_2\in \text{int} \end{cases}$$

当不考虑两个变量整数约束时，将上述模型作为一个 LP 问题在 Matlab 中可以求得最优解。

Matlab 中的求解程序如下：

```
f=[-20 -10];
A=[5 4;2 5];
B=[24 13];
lb=[0 0];
[x,fit,exitF]=linprog(f,A,B,[],[],lb,[])
```

运行该程序后得到的结果如下：

```
x = 4.8000      0.0000
fit =    -96.0000
exitF =    1
```

可以看出，当 x_1 取值 4.8、x_2 取值 0 时可以获得最大的盈利 96 个单位，但是由于 x_1 的取值不是整数，不符合实际装箱要求。如果将解｛4.8 0｝转化为｛5 0｝，则不满足第一条约束，转化后的解不是可行解；如果将解｛4.8 0｝转化为｛4 0｝，得到盈利为 80 单位，虽然是可行解，但是并不是最优解。

如果将如果将解｛4.8 0｝转化为｛4 1｝，则可以获得盈利 90 个单位，且该解是可行解，也是该问题的最优解。

从这个例子可以看出，将整数规划问题对应的线性规划问题的解直接整数化虽然很简单，但是最后得到的解可能同最优解之间相距深远，因此需要采取特定的方法来获取证书规划的最优解。

2 分支定界法基本运算逻辑

在求解整数规划时，如果可行域是有界的，首先容易想到的方法就是穷举变量的所有可行的整数组合，然后比较它们的目标函数值以定出最优解。对于小型的问题，变量数很少，可行的整数组合数也是很小时，这个方法是可行的，也是有效的。对于大型的问题，可行的整数组合数是很大的。例如将 n 项任务指派 n 个人去完成，不同的指派方案共有 n !种，当 n = 10，这个数就超过 300 万；当 n = 20，这个数就超过 2×10^{18}，如果一一计算，就是用每秒百万次的计算机，也要几万年的功夫，很明显，解这样的题，穷举法是不可取的。所以我们的方法一般应是仅检查可行的整数组合的一部分，就能定出最优的整数解。分支定界解法(br anch and bound method)就是其中的一个。

分支定界法可用于解纯整数或混合的整数规划问题。在 20 世纪 60 年代初

由 Land Doig 和 Dakin 等人提出。由于这方法灵活且便于用计算机求解，所以现在它已是解整数规划的重要方法。

设有最大化的整数规划问题 A，与它相应的线性规划为问题 B，从解问题 B 开始，若其最优解不符合 A 的整数条件，那么 B 的最优目标函数必是 A 的最优目标函数 z*的上界，记作 $\bar{z}$，而 A 的任意可行解的目标函数值将是 z*的一个下界 $\underline{z}$。分支定界法就是将 B 的可行域分成子区域(称为分支)的方法，逐步减小 $\bar{z}$ 和增大 $\underline{z}$，最终求到 z*。

现用下例来说明，例 2：

$$\max \quad z = 40x_1 + 90x_2$$

$$\begin{cases} 9x_1 + 7x_2 \le 56 \\ 7x_1 + 20x_2 \le 70 \\ x_1, x_2 \ge 0 \\ x_1, x_2 \in \text{int} \end{cases}$$

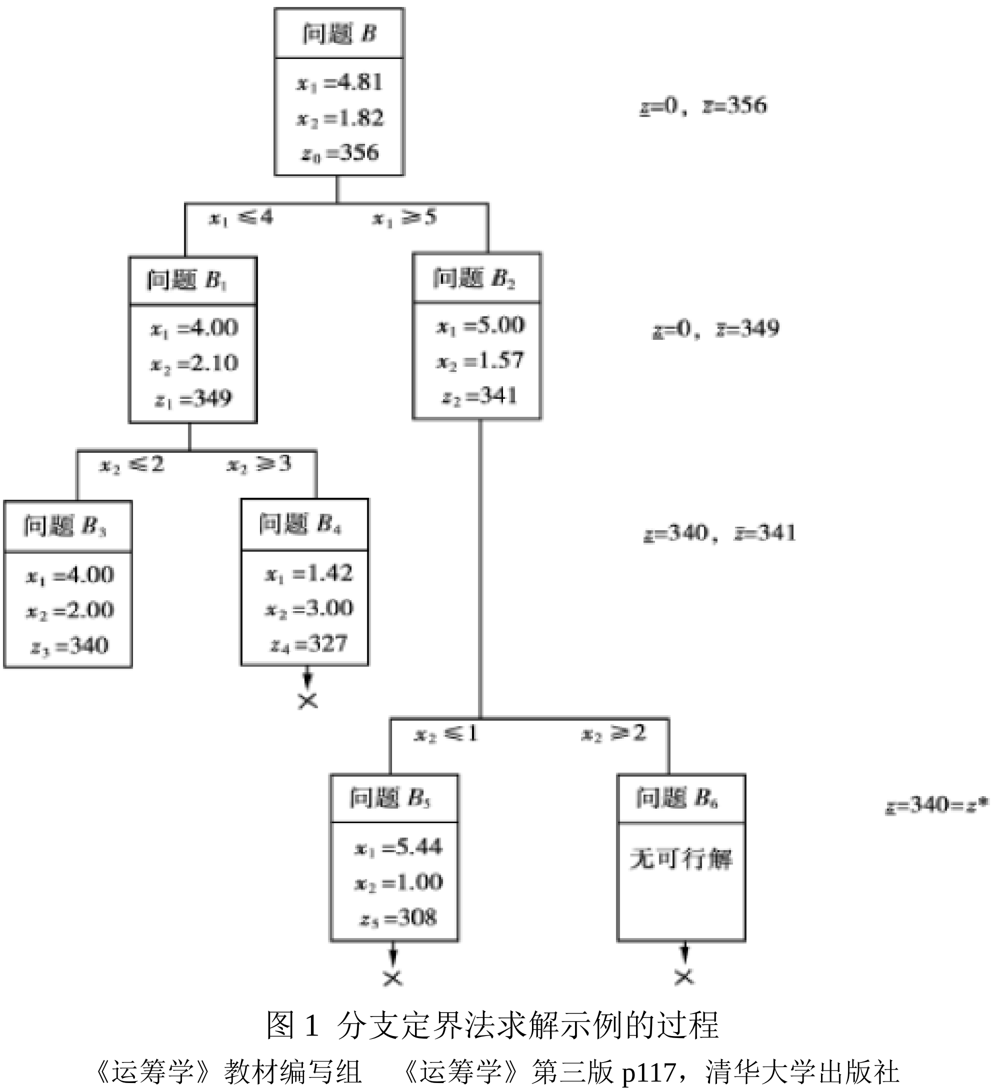

图 1 分支定界法求解示例的过程

《运筹学》教材编写组 《运筹学》第三版 p117，清华大学出版社

为了更好的理解分支定界法的计算流程，下面计算另外两个示例。

例 3：整数规划模型如下，判断使用线性规划求解后采用凑整的办法能够求得最优整数解？

$$\max \quad z = 3x_1 + 2x_2$$

$$\begin{cases} 2x_1 + 3x_2 \le 14.5 \\ 4x_1 + x_2 \le 16.5 \\ x_1, x_2 \ge 0 \\ x_1, x_2 \in \text{int} \end{cases}$$

解：

（1） 首先将该模型转化为 linprog 理解的 LP 标准型如下：

$$\min \quad z = -3x_1 - 2x_2$$

$$\begin{cases} 2x_1 + 3x_2 \le 14.5 \\ 4x_1 + x_2 \le 16.5 \\ x_1, x_2 \ge 0 \end{cases}$$

（2） 通过如下 Matlab 程序进行求解

```
f=[-3 -2];
A=[2 3;4 1];
B=[14.5 16.5];
lb=[0 0];
ub=[inf,inf];
[x,fit,exitF]=linprog(f,A,B,[],[],lb,ub)
```

求解结果为：z=15.5，x=[3.5,2.5]

（3） 采用将两个变量取整来获取可行解，可以将解转化为如下四种组合：

[3 2]　　[3 3]　　[4 2]　　[4 3]

其中[3 3]，[4 3]不满足第一个约束条件，[4 2]不满足第二个约束条件，为非可行解。

[3 2]对应的目标函数值为：13。那么这个解是不是最优解呢？下面通过分支定界获得最优解来对比。

（4） 分支定界

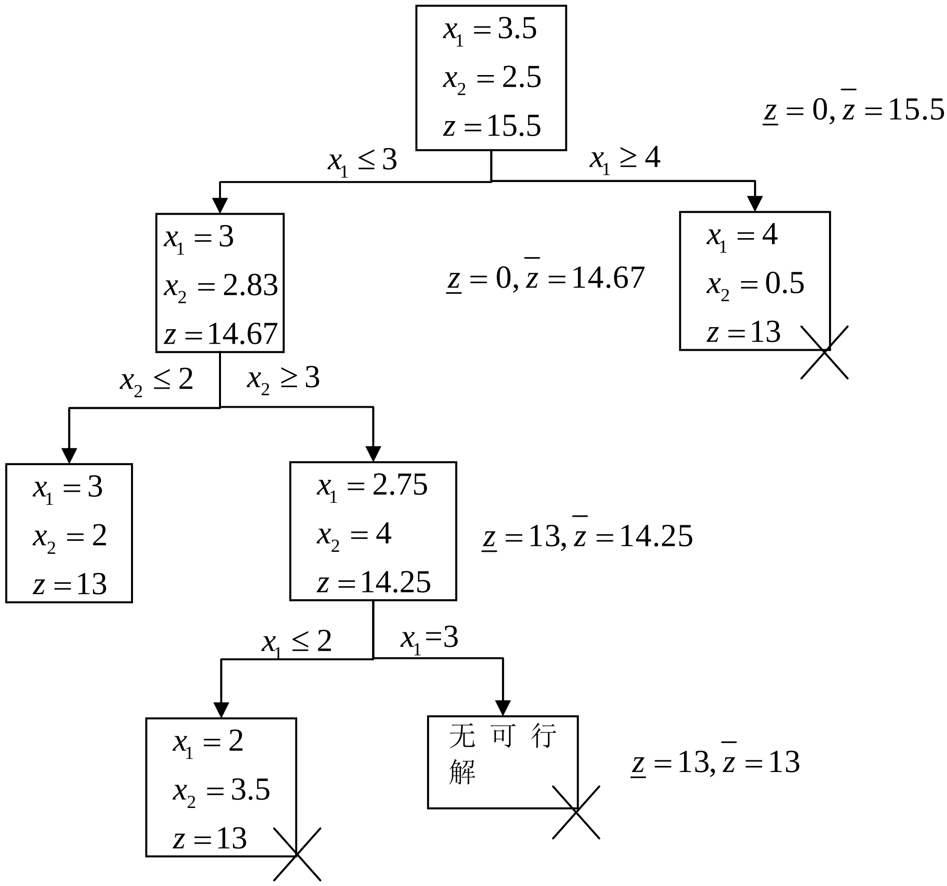

图 2 例 3 分支定界计算过程图

根据上图的分支定界求解可知，最优解为[3,2]，目标函数最大值为 13，所以通过简单的取整得到了本模型的最优解。

当一个分支中下界和上界相等时，或者找到了下界，其他分支无可行解则表示找到了最优解；

剪枝规则：当找到了函数的整数可行解下界后，其他分支 LP 解值不能大于下界时，剪枝；或者无可行解时，剪枝。

例 4：整数规划模型如下，判断使用线性规划求解后采用凑整的办法能够求得最优整数解？

$$\max \quad z = 3x_1 + 2x_2$$

$$\begin{cases} 2x_1 + 3x_2 \le 14 \\ 2x_1 + x_2 \le 9 \\ x_1, x_2 \ge 0 \\ x_1, x_2 \in \text{int} \end{cases}$$

解：

（1） 首先将该模型转化为 linprog 理解的 LP 标准型如下：

$$\min \quad z = -3x_1 - 2x_2$$

$$\begin{cases} 2x_1 + 3x_2 \le 14 \\ 2x_1 + x_2 \le 9 \\ x_1, x_2 \ge 0 \end{cases}$$

（2） 通过如下 Matlab 程序进行求解

```
f=[-3 -2];
A=[2 3;2 1];
B=[14 9];
lb=[0 0];
ub=[inf,inf];
[x,fit,exitF]=linprog(f,A,B,[],[],lb,ub)
```

求解结果为：z=14.75，x=[3.25,2.5]

（3） 采用将两个变量取整来获取可行解，可以将解转化为如下四种组合：

[3 2]　　　[3 3]　　　[4 2]　　　[4 3]

其中[3 3]，[4 3]不满足第一个约束条件，[4 2]不满足第二个约束条件，为非可行解。

[3 2]对应的目标函数值为：13。那么这个解是不是最优解呢？下面通过分支定界获得最优解来对比。

（4） 分支定界

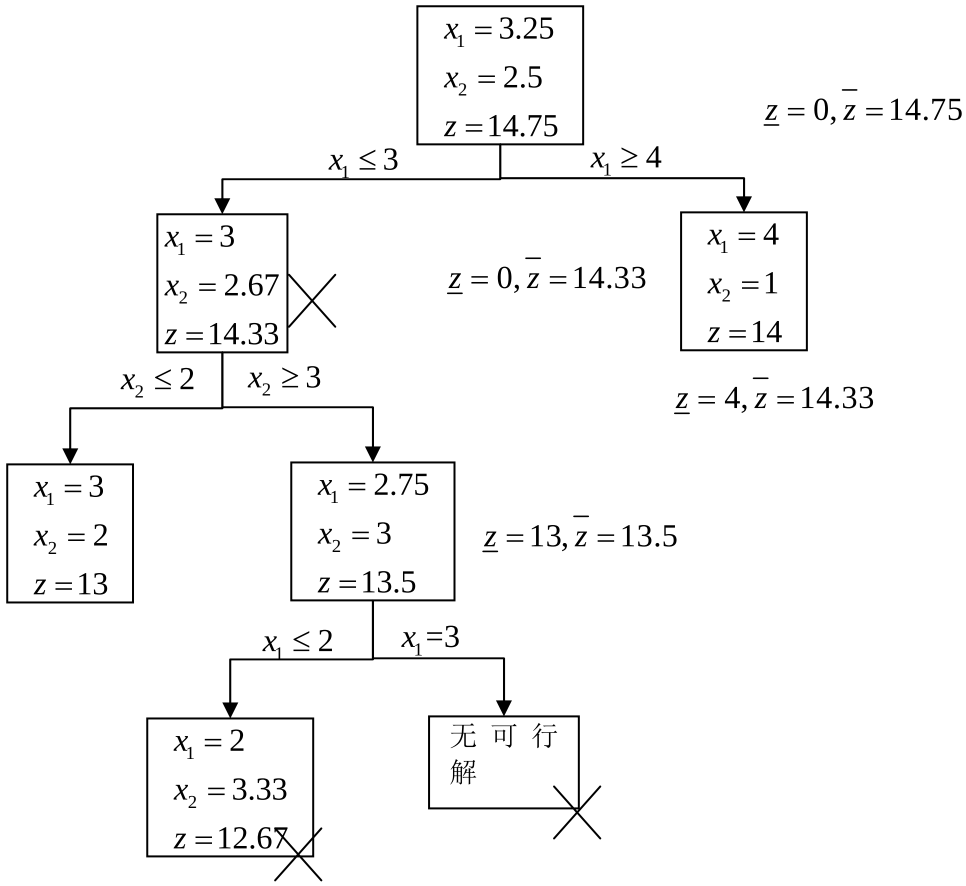

图 3 例 4 分支定界计算过程图

根据上图的分支定界求解可知，最优解为[4,1]，目标函数最大值为 14，所以通过简单的取整不能得到了本模型的最优解。

3 分支定界法程序设计逻辑框图

根据上述示例的计算过程，总结分支定界计算机处理流程如下：

（1） 将原整数规划模型 A 整理成线性规划标准形式 B；该模型的参数有：f,A,b,Aeq,beq,lb,ub,X0,F0，其中 X0 和 F0 分别设定为模型 A 的初始可行解和目标函数值，可由人工简要计算后获得；

（2） 求解线性规划问题 B，根据问题 B 的解进行后续的分支和定界，处理准则如下：

（a） 如果无可行解，剪枝（如果不是处于递归过程中，则执行至此处，问题结束，该整数规划没有可行解）；

（b） 如果有非整数可行解，且函数值高于上界 F0，剪枝；

（c） 如果有非整数可行解，但低于上界，则继续分支，执行步骤（3）；

（d） 如果有整数可行解，且函数值高于上界，剪枝；

（e） 如果有整数可行解，且函数值低于上界，更新上界数值 F0 和可行解

X0，停止分支。

（3） 任意选择一个取值不满足整数的变量 i 进行分支，分成 $x_i \le \lfloor \tilde{x_i} \rfloor$ 和 $x_i \ge \lfloor \tilde{x_i} \rfloor + 1$，用这两个条件分别形成线性规划模型 B1 和 B2，其中 B1 模型中 ub（i）$= \lfloor \tilde{x_i} \rfloor$，模型 B2 中 lb(i)= $\lfloor \tilde{x_i} \rfloor + 1$；

（4） 采用递归函数的形式，令 B=B1，重新执行步骤（2）；

（5） 采用递归函数的形式，令 B=B2，重新执行步骤（2）；

（6） 获得最优解，解的数值分别为数值更新之后的 X0 和 F0。

4 分支定界法 Matlab 函数 jBranchBound 程序源代码

根据上述程序设计逻辑结构，设计分支定界法的 Matlab 函数源代码如下：

```
function
[xOut,fitOut,flagOut]=jBranchBound(c,A,B,Aeq,beq,vlb,vub,optXin,optF)
   global optX optVal optFlag;
   optX=optXin;optVal=optF;
   options =
optimoptions('linprog','Algorithm','interior-point-legacy','display',
'none');
   [x,fit,status]=linprog(c,A,B,Aeq,beq,vlb,vub,[],options);
   if status~=1 %no optimal solution,end the program
      %do these when the first LP cann't get a feasible solution
      %in the next loop, these value don't return back.
      xOut=x;fitOut=fit;flagOut=status;
      return;
   end
   % the following programs are based on the condition of status=1
   if max(abs(x-round(x)))>0.005 %there have real solution
      if fit>optVal
         %cann't find any real feasible solution better than optVal
         xOut=x;fitOut=fit;flagOut=-100;
         return;
      else
         %continue to branch
      end
   else% there have integer solution
      if fit>optVal
         %cann't find any integer feasible solution better than optVal
         xOut=x;fitOut=fit;flagOut=-101;
         return;
      else
         optVal=fit;optX=x; optFlag=status;
```

```
            xOut=x;fitOut=fit;flagOut=status;
            return;
        end
    end

    midX=abs(x-round(x));
    notIntV=find(midX>0.005);
    pXidx=notIntV(1);%decide the branch variable index
    tempVlb=vlb;tempVub=vub;
    %the up branch calculation
    if vub(pXidx)>=fix(x(pXidx))+1
        tempVlb(pXidx)=fix(x(pXidx))+1;
        [~,~,~]=jBranchBound(c,A,B,Aeq,beq,tempVlb,vub,optX,optVal);
    end
    %the down branch calculation
    if vlb(pXidx)<=fix(x(pXidx))
        tempVub(pXidx)=fix(x(pXidx));
        [~,~,~]=jBranchBound(c,A,B,Aeq,beq,vlb,tempVub,optX,optVal);
    end
    %get the optimal solution
    xOut=optX;fitOut=optVal;flagOut=optFlag;
end
```

5 jBranchBound 求解整数规划问题应用算例

下面以第一、二节中例举的四个例子为例，说明运用 Matlab 分支定界函数 jBranchBound 求解的步骤和技巧。

运用 Matlab 单纯形法程序求解整数规划问题的主要步骤为：

（1）将整数规划模型转换为松弛的线性规划标准型；

（2）提取松弛的线性规划标准型模型的输入参数；

（3）运用 Matlab 分支定界程序 jBranchBound 求解；

（4）对结果进行分析和整理结论报告。

在这四个例子之后，另行列举了一个多变量的整数规划问题和一个 0-1 整数规划问题进行求解描述。

5.1 例 1 求解

目标函数 $\max \quad z = 20x_1 + 10x_2$

约束条件

$$\begin{cases} 5x_1 + 4x_2 \le 24 \\ 2x_1 + 5x_2 \le 13 \\ x_1, x_2 \ge 0 \\ x_1, x_2 \in \text{int} \end{cases}$$

Step1：将模型转化为 linprog 函数认可的标准模型

目标函数

$$\min \quad z = -20x_1 - 10x_2$$

约束条件

$$\begin{cases} 5x_1 + 4x_2 \le 24 \\ 2x_1 + 5x_2 \le 13 \\ x_1, x_2 \ge 0 \end{cases}$$

Step2：提取标准型模型的输入参数

f=[-20 -10]

B=[24 13]’

A=[5 4;2 5]

Step3：运用 Matlab 程序 jSimplex 进行求解，求解的程序源代码和结果如下：

源代码：

```
f=[-20 -10];
A=[5 4;2 5];
B=[24 13];
lb=[0 0];
ub=[inf,inf];
optX=[0,0];optVal=0;
[x,fit,exitF]=jBranchBound(f,A,B,[],[],lb,ub,optX,optVal)
```

运算结果：

```
x = 4.0000
    1.0000
fit =-90.0000
exitF =  1
```

Step4：对结果进行分析和整理结论报告

获得最优解 90。

5.2 例 2 求解

目标函数 $\max \quad z = 40x_1 + 90x_2$

约束条件

$$\begin{cases} 9x_1 + 7x_2 \le 56 \\ 7x_1 + 20x_2 \le 70 \\ x_1, x_2 \ge 0 \\ x_1, x_2 \in \text{int} \end{cases}$$

Step1：将模型转化为 linprog 函数认可的标准模型

目标函数

$$\min \quad z = -40x_1 - 90x_2$$

约束条件

$$\begin{cases} 9x_1 + 7x_2 \le 56 \\ 7x_1 + 20x_2 \le 70 \\ x_1, x_2 \ge 0 \end{cases}$$

Step2：提取标准型模型的输入参数

f=[-40 -90]

B=[56 70]’

A=[9 7;7 20]

Step3：运用 Matlab 程序 jSimplex 进行求解，求解的程序源代码和结果如下：

源代码：

```
f=[-40 -90];
A=[9 7;7 20];
B=[56 70];
lb=[0 0];
ub=[inf,inf];
optX=[0,0];optVal=0;
[x,fit,exitF]=jBranchBound(f,A,B,[],[],lb,ub,optX,optVal)
```

运算结果：

```
x = 4.0000
    2.0000
fit = -340.0000
exitF =     1
```

Step4：对结果进行分析和整理结论报告

获得了最优解 340。

5.3 例 3 求解

目标函数 $\max \quad z = 3x_1 + 2x_2$

约束条件
$$\begin{cases} 2x_1 + 3x_2 \le 14.5 \\ 4x_1 + x_2 \le 16.5 \\ x_1, x_2 \ge 0 \\ x_1, x_2 \in \text{int} \end{cases}$$

Step1：将模型转化为 linprog 函数认可的标准模型

目标函数 $\min \quad z = -3x_1 - 2x_2$

约束条件

$$\begin{cases} 2x_1 + 3x_2 \le 14.5 \\ 4x_1 + x_2 \le 16.5 \\ x_1, x_2 \ge 0 \end{cases}$$

Step2：提取标准型模型的输入参数

f=[-3 -2]

B=[14.5 16.5]’

A=[2 3;4 1]

Step3：运用 Matlab 程序 jSimplex 进行求解，求解的程序源代码和结果如下：

源代码：

```
f=[-3 -2];
A=[2 3;4 1];
B=[14.5 16.5];
lb=[0 0];
ub=[inf,inf];
optX=[0,0];optVal=-0;
[x,fit,exitF]=jBranchBound(f,A,B,[],[],lb,ub,optX,optVal)
```

运算结果：

```
x = 3.0000
    2.0000
fit =  -13.0000
exitF =      1
```

Step4：对结果进行分析和整理结论报告

获得了最优解 13

5.4 例 4 求解

目标函数　　max　$z = 3x_1 + 2x_2$

约束条件

$$\begin{cases} 2x_1 + 3x_2 \le 14 \\ 2x_1 + x_2 \le 9 \\ x_1, x_2 \ge 0 \\ x_1, x_2 \in \text{int} \end{cases}$$

Step1：将模型转化为 linprog 函数认可的标准模型

目标函数　　min　$z = -3x_1 - 2x_2$

约束条件

$$\begin{cases} 2x_1 + 3x_2 \le 14 \\ 2x_1 + x_2 \le 9 \\ x_1, x_2 \ge 0 \end{cases}$$

Step2：提取标准型模型的输入参数

f=[-3 -2]

B=[14 9]'

A=[2 3;2 1]

Step3：运用 Matlab 程序 jSimplex 进行求解，求解的程序源代码和结果如下：

源代码：

```
f=[-3 -2];
A=[2 3;2 1];
B=[14 9];
lb=[0 0];
ub=[inf,inf];
optX=[0,0];optVal=-0;
[x,fit,exitF]=jBranchBound(f,A,B,[],[],lb,ub,optX,optVal)
```

运算结果：

```
x = 4.0000
    1.0000
fit =  -14.0000
exitF =      1
```

Step4：对结果进行分析和整理结论报告

获得了最优解 14。

5.5 例 5-多变量整数规划问题求解

目标函数　　max　$z=3x_1+2x_2+5x_3+8x_4+x_5$

约束条件
$$\begin{cases} 2x_1+3x_2+3x_3+3x_4+3x_5\le 140 \\ 5x_1+x_2+7x_3+x_4+x_5\le 90 \\ 7x_1+4x_2-3x_5\ge 9 \\ 5x_3+2x_4+3x_5\le 70 \\ 2x_1+3x_4+3x_5\le 109 \\ x_1,x_2\ge 0 \\ x_1,x_2\in \text{int} \end{cases}$$

Step1：将模型转化为 linprog 函数认可的标准模型

目标函数　　min　$z=-3x_1-2x_2-5x_3-8x_4-x_5$

约束条件

$$\begin{cases} 2x_1+3x_2+3x_3+3x_4+3x_5\le 140 \\ 5x_1+x_2+7x_3+x_4+x_5\le 90 \\ -7x_1-4x_2+3x_5\le -9 \\ 5x_3+2x_4+3x_5\le 70 \\ 2x_1+3x_4+3x_5\le 109 \\ x_1,x_2\ge 0 \end{cases}$$

Step2：提取标准型模型的输入参数

```
f=[-3 -2 -5 -8 -1];
A=[2 3 3 3 3;5 1 7 1 1;-7 -4 0 0 3;0 0 5 2 3;2 0 0 3 3];
B=[140 90 -9 70 109];
```

Step3：运用 Matlab 程序 jSimplex 进行求解，求解的程序源代码和结果如下：

源代码：

```
f=[-3 -2 -5 -8 -1];
A=[2 3 3 3 3;5 1 7 1 1;-7 -4 0 0 3;0 0 5 2 3;2 0 0 3 3];
B=[140 90 -9 70 109];
lb=[0 0 0 0 0];
ub=[inf,inf,inf,inf,inf];
optX=[1 1 0 0 0];optVal=-5;
[x,fit,exitF]=jBranchBound(f,A,B,[],[],lb,ub,optX,optVal)
```

运算结果：

```
x =     2.0000     10.0000      0.0000     35.0000      0.0000
fit = -306.0000
exitF =1
```

Step4：对结果进行分析和整理结论报告

获得了最优解 306，对应变量取值为[2 10 0 35 0]

5.6 例 6-多变量 0-1 整数规划问题求解

（1）0-1 规划问题 1

目标函数 $\min \quad z=4x_1+3x_2+2x_3$

约束条件

$$\begin{cases} 2x_1-5x_2+3x_3\le 4 \\ 4x_1+x_2+3x_3\ge 3 \\ x_2+x_3\ge 1 \\ x_1,x_2,x_3\in 0,1 \end{cases}$$

Step1：将模型转化为 linprog 函数认可的标准模型

目标函数 $\min \quad z=4x_1+3x_2+2x_3$

约束条件

$$\begin{cases} 2x_1 - 5x_2 + 3x_3 \le 4 \\ -4x_1 - x_2 - 3x_3 \le -3 \\ -x_2 - x_3 \le -1 \\ x_1, x_2, x_3 \in 0,1 \end{cases}$$

Step2：提取标准型模型的输入参数

```
f=[4 3 2];
A=[2 -5 3;-4 -1 -3;0 -1 -1];
B=[4 -3 -1];
```

Step3：运用 Matlab 程序 jSimplex 进行求解，求解的程序源代码和结果如下：

源代码：

```
f=[4 3 2];
A=[2 -5 3;-4 -1 -3;0 -1 -1];
B=[4 -3 -1];
lb=[0 0 0];
ub=[1 1 1];
optX=[1 1 0];optVal=7;
[x,fit,exitF]=jOR.jBranchBound(f,A,B,[],[],lb,ub,optX,optVal)
```

运算结果：

```
x =      0.0000      0.0000      1.0000
fit =      2.0000
exitF =       1
```

Step4：对结果进行分析和整理结论报告

获得了最优解 2，对应变量取值为[0 0 1]

（2）0-1 规划问题 2

目标函数 $\min \quad z = 2x_1 + 5x_2 + 3x_3 + 4x_4$

约束条件

$$\begin{cases} -4x_1 + x_2 + x_3 + x_4 \ge 0 \\ -2x_1 + 4x_2 + 2x_3 + 4x_4 \ge 4 \\ x_1 + x_2 - x_3 + x_4 \ge 1 \\ x_1, x_2, x_3, x_4 \in 0,1 \end{cases}$$

Step1：将模型转化为 linprog 函数认可的标准模型

目标函数 $\min \quad z = 2x_1 + 5x_2 + 3x_3 + 4x_4$

约束条件
$$\begin{cases} -4x_1 + x_2 + x_3 + x_4 \le 0 \\ 2x_1 - 4x_2 - 2x_3 - 4x_4 \le -4 \\ -x_1 - x_2 + x_3 - x_4 \le -1 \\ x_1, x_2, x_3, x_4 \in 0,1 \end{cases}$$

Step2：提取标准型模型的输入参数

```
f=[2 5 3 4];
A=[-4 1 1 1;2 -4 -2 -4;-1 -1 1 -1];
B=[0 -4 -1];
```

Step3：运用 Matlab 程序 jSimplex 进行求解，求解的程序源代码和结果如下：

源代码：

```
f=[2 5 3 4];
A=[-4 1 1 1;2 -4 -2 -4;-1 -1 1 -1];
B=[0 -4 -1];
lb=[0 0 0 0];
ub=[1 1 1 1];
optX=[1 1 0 1];optVal=11;
[x,fit,exitF]=jOR.jBranchBound(f,A,B,[],[],lb,ub,optX,optVal)
```

运算结果：

```
x =     1.0000          0     1.0000     1.0000
fit =     9.0000
exitF =       1
```

Step4：对结果进行分析和整理结论报告

获得了最优解 9，对应变量取值为[1 1 0 1]

从上述应用算例可以看出，分支定界算法可以很好的解决整数规划问题，设计的程序可以很好的运行。

6 matlab 中函数 intlinprog 函数介绍

在 Matlab2014 版之后提供了现成函数 intlinprog 可以进行混合整数线性规划问题的求解，该函数求解需要将混合整数线性规划转换为如下的标准形式：

目标函数 min $z = \sum_{j=1}^{n} c_j x_j$

约束条件
$$\begin{cases} Ax \le b \\ A_{eq}x = b_{eq} \\ lb \le x \le ub \\ x(\text{int}\,con) \in \text{int} \end{cases}$$

其中有部分变量 x 是整数。

6.1 intlinprog 函数调用格式

intlinprog 函数的调用格式可以是如下形式：

x=linprog(f,intcon,A,b)

x=linprog(f, intcon,A,b,Aeq,beq)

x=linprog(f, intcon,A,b,Aeq,beq,lb,ub)

x=linprog(f, intcon,A,b,Aeq,beq,lb,ub,x0)

x=linprog(f, intcon,A,b,Aeq,beq,lb,ub,x0,options)

[x,fval]=linprog(…)

[x, fval, exitflag]=linprog(…)

[x, fval, exitflag, output]=linprog(…)

[x, fval, exitflag, output, lambda]=linprog(…)

可以看出 intlinprog 函数的调用格式同 linprog 函数的调用格式基本相同，只是输入变量中多了一个参数 intcon，用于存储整数变量的序号向量，例如当一个混合整数线性规划问题中有变量 x_1, x_2, x_8 为整数变量，而其他变量为实数变量时，intcon 中存储数据形式为[1 2 8]。

6.2 运用 intlinprog 函数进行示例求解

（1）示例 1 求解源代码

```
f=[-20 -10];
A=[5 4;2 5];
B=[24 13];
lb=[0 0];
ub=[inf,inf];
intcon=[1 2];
[x,fit,exitF]=intlinprog(f,intcon,A,B,[],[],lb,ub)
```

（2）示例 2 求解源代码

```
f=[-40 -90];
A=[9 7;7 20];
B=[56 70];
lb=[0 0];
ub=[inf,inf];
intcon=[1 2];
[x,fit,exitF]=intlinprog(f,intcon,A,B,[],[],lb,ub)
```

（3）示例 3 求解源代码

```
f=[-3 -2];
A=[2 3;4 1];
B=[14.5 16.5];
lb=[0 0];
ub=[inf,inf];
```

```
intcon=[1 2];
[x,fit,exitF]=intlinprog(f,intcon,A,B,[],[],lb,ub)
```

（4）示例 4 求解源代码

```
f=[-3 -2];
A=[2 3;2 1];
B=[14 9];
lb=[0 0];
ub=[inf,inf];
intcon=[1 2];
[x,fit,exitF]=intlinprog(f,intcon,A,B,[],[],lb,ub)
```

（5）示例 5 求解源代码

```
f=[-3 -2 -5 -8 -1];
A=[2 3 3 3 3;5 1 7 1 1;-7 -4 0 0 3;0 0 5 2 3;2 0 0 3 3];
B=[140 90 -9 70 109];
lb=[0 0 0 0 0];
ub=[inf,inf,inf,inf,inf];
intcon=[1 2 3 4 5];
[x,fit,exitF]=intlinprog(f,intcon,A,B,[],[],lb,ub)
```

（6）示例 6（0-1 规划问题 1）求解源代码

```
f=[4 3 2];
A=[2 -5 3;-4 -1 -3;0 -1 -1];
B=[4 -3 -1];
lb=[0 0 0];
ub=[1 1 1];
intcon=[1 2 3];
[x,fit,exitF]=intlinprog(f,intcon,A,B,[],[],lb,ub)
```

7 小结

经过了多天的断续工作，终于理解了分支定界法计算流程和实现方法，并完成了本文档的编辑工作。个人的感受和体会有：

（1）分支定界法用手工计算容易理解，但是进行计算机编程比较麻烦，主要是（针对本人非计算机专业出身来说）计算流程中递归过程的运行逻辑和程序实现耗费了挺多时间的。

（2）该程序实现过程使用了2~3天的时间方才编程结束，而且不确定是否有不完善的地方，不过从几个算例中可以看出，至少目前为止该程序能够很好的执行并获得最优解。

（3）当然实际应用中能够运用intlinprog函数时最好matlab自带的函数，对计算结果更有信息一点。

（4）充分理解了线性规划问题和模型的结构，以前虽然对线性规划问题定义中目标函数和约束条件必须是决策变量的线性函数这个特征也是了解的，但是通过

计算机求解过程的实现，更深刻理解了这个特征，因为不是线性函数，没办法设定输入系数矩阵啊。
（5）本来可以早点结束这个算法文档的编辑，只是中间收到了一本小说的吸引，中断了近3天。自我检讨：尽量不要搜索新的小说，因为一旦发现好的小说，就会耗费几天时间啊。

2017-9-13