

《运筹学》实验报告

| | | |
|---------|--------------|-----------|
| 第 三 次实验 | 日期：10 月 29 日 | 得分： |
| 学号： | 姓名： | 专业：智能科学技术 |

一、实验目的

理解、熟悉整数规划的分支定界法的运算过程和思想

二、算法设计

理解分支定界法的数学思路，将其转换为代码进行运算。

在完成运算后，将其绘制出来。

三、核心程序代码

主要代码：运算函数

```
function
[optSolution,optValue,exists,hist]=BranchBound(c,A,b)
%% 格式要求如下（运算主体是 Matlab 自带的求解器函数 linprog）：
% min c'*x; c 是个列向量，如果为 max 将 c 自乘-1
% s.t.
% A*x<=b; 默认约束条件为\leq，若有\geq 则将对应该行的 A(i,j)和 b(i)
自乘-1

hist=[];
upper = inf;
lower = -inf;
branchStack_A(1) = {A};
branchStack_B(1) = {b};
numOfVariable = length(c);

while ~isempty(branchStack_A)
    len = length(branchStack_A);
    A = branchStack_A{len};
    b = branchStack_B{len};
    branchStack_A(len) = [];
    branchStack_B(len) = [];

    [x,opt,exist] = linprog(c,A,b);
    hist=[hist;x'];
    % 是否迭代的判断条件
```

```

if 1 == exist && (opt - upper) <= 1e-7
    for iVar = 1:numOfVariable + 1
        if iVar <= numOfVariable
            if abs(x(iVar) - round(x(iVar))) > 1e-7
                %进行分枝操作，就将子问题入栈
                temp_A = zeros(1,numOfVariable);

                temp_A(iVar) = -1;
                branchStack_A(len) = {[A;temp_A]};
                branchStack_B(len) = {[b;-(floor(x(iVar) +
1))]];

                temp_A(iVar) = 1;
                branchStack_A(len + 1) = {[A;temp_A]};
                branchStack_B(len + 1) =
{[b;floor(x(iVar))]];
                break;
            end
        else
            %是个整数解.更新 upper
            if abs(upper - opt) < 1e-7
                optSolution(:,end + 1) = x;
            else
                optSolution = x;
                upper = opt;
                optValue = opt;
            end
        end
    end
end
end
end

sizeOfSolution = size(optSolution);
if sizeOfSolution(2) == 0
    exists = 0;
else
    exists = 1;
end
end

```

主要代码：绘图函数

```

function Draw(hist,A,b)
% hist:迭代点的迭代过程，每行为一个点
% A,b:定义同 BranchBound()
x1=0:0.05:10;

```

| |
|--|
| <pre> x2=zeros(length(b),length(x1)); % 画约束条件 for i=1:length(b) x2(i,:)=(b(i)-A(i,1).*x1)./A(i,2); % 根据 Ax<=b 求解对应的 x(2) plot(x1,x2(i,:), 'LineWidth',3); axis([0 max(x1) 0 max(max(x2))]); hold on; end % 画迭代点 plot(hist(:,1),hist(:,2), 'r.-', 'LineWidth',3); hold on; for i=1:size(hist,1) text(hist(i,1),hist(i,2),int2str(i)+ ("+num2str(hist(i,1))+","+num2str(hist(i,2))+")", "FontSize",1 0)); end % 做点修饰 title("约束条件与迭代过程"); xlabel('x_1'); ylabel('x_2'); grid; hold off; end </pre> |
| <p>函数调用</p> <pre> c=[-3,-2]'; A=[2,3;1,0.5]; b=[14,4.5]'; [optSolution,optValue,exists,hist]=BranchBound(c,A,b); % 求解 Draw(hist,A,b); % 绘图 </pre> |

四、测试及结果（给出测试用例及测试结果）

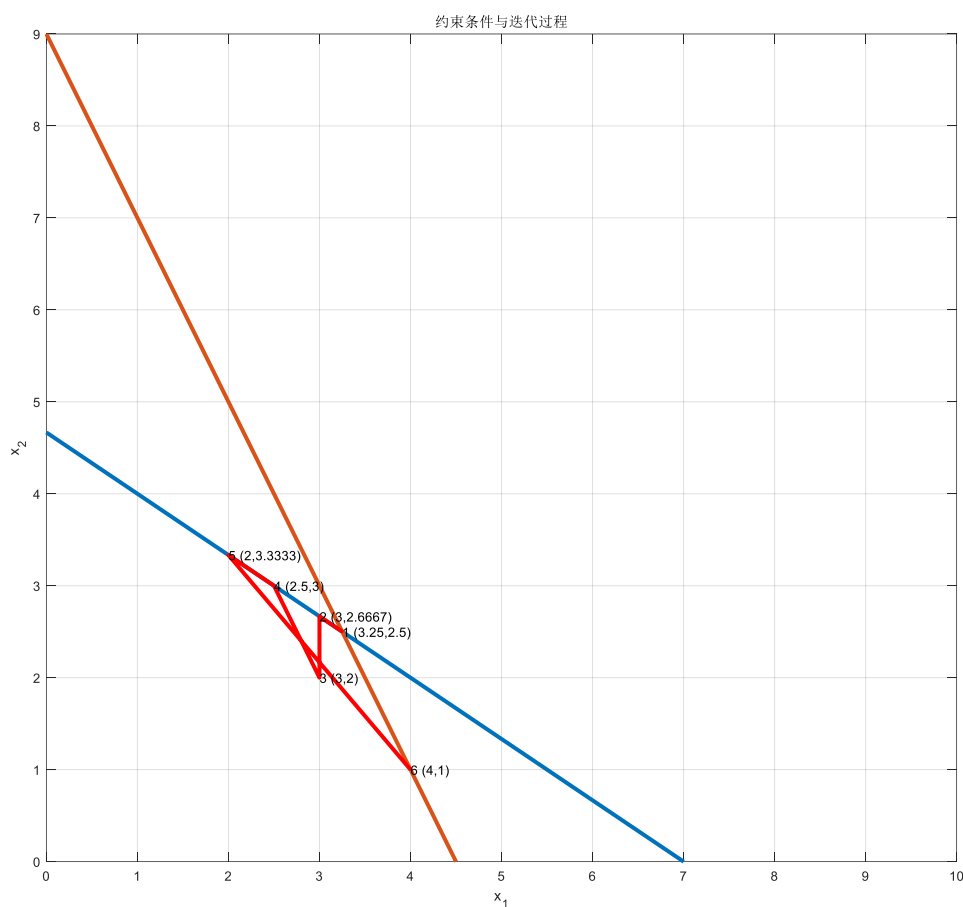
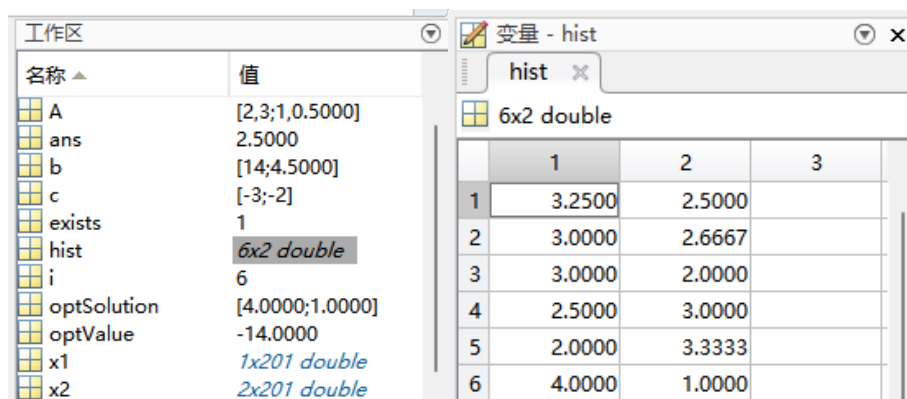
(1) 测试用例：

使用课本 P107-109 的例 1 作为测试用例

$$L_0 : \max z = 3x_1 + 2x_2$$

$$s. t. \begin{cases} 2x_1 + 3x_2 \leq 14 \\ x_1 + 0.5x_2 \leq 4.5 \end{cases}$$

(2) 运算结果：



根据结果显示，该方法找到了整数规划的最优解和最优值，由于输入为 max，系数矩阵 c 自乘-1，所以求解最优值与实际最优值互为相反数。求解完成。