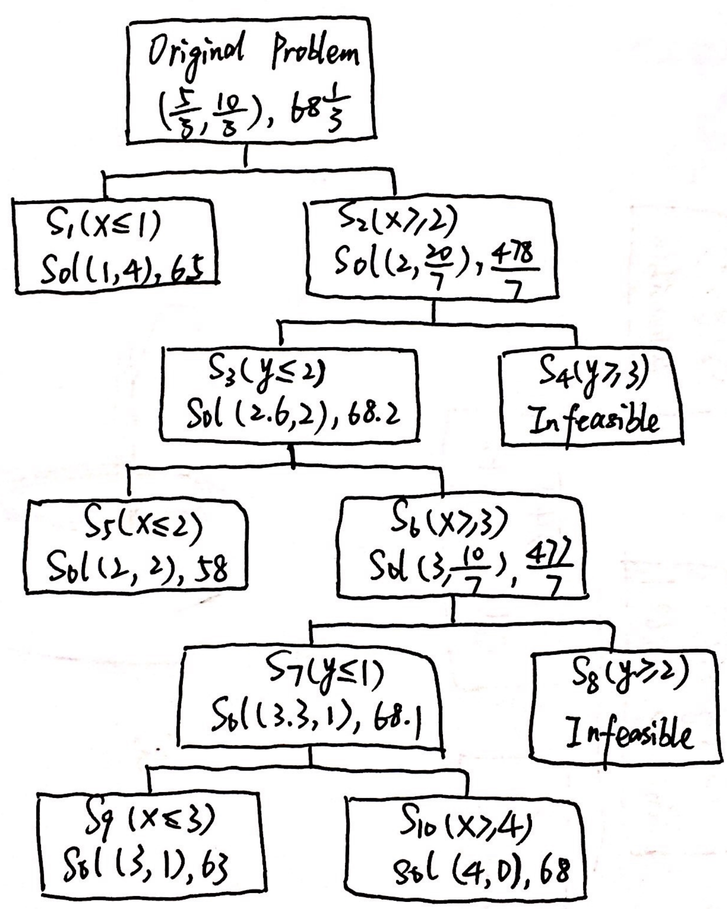
MAT3007: Optimization - Assignment 10

Ran Hu 116010078

19 December, 2018

Problem 1

The branch-and-bound tree is as follows:



The following MATLAB code shows what I did at each node.

```
A = [10 7;
    1 1];
b = [17 12];
c = [40; 5];

%% S0
cvx_begin quiet
```

```
variable x(2);
    maximize b*x
    subject to
        A*x <= c;
        x >= 0
cvx_end

x
optval = cvx_optval

%% S1
cvx_begin quiet
variable x(2);
    maximize b*x
    subject to
        A*x <= c;
        x >= 0;
        x(1) <= 1
cvx_end

x
optval = cvx_optval

%% S2
cvx_begin quiet
variable x(2);
    maximize b*x
    subject to
        A*x <= c;
        x >= 0;
        x(1) >= 2
cvx_end

x
optval = cvx_optval

%% S3
cvx_begin quiet
variable x(2);
    maximize b*x
    subject to
        A*x <= c;
        x >= 0;
        x(1) >= 2;
        x(2) <= 2
cvx_end

x
optval = cvx_optval
```

```
% S4
cvx_begin quiet
variable x(2);
    maximize b*x
    subject to
        A*x <= c;
        x(1) >= 2;
        x(2) >= 3
cvx_end

x
optval = cvx_optval

%% S5
cvx_begin quiet
variable x(2);
    maximize b*x
    subject to
        A*x <= c;
        x(1) >= 2;
        x(1) <= 2;
        x(2) <= 2;
cvx_end

x
optval = cvx_optval

%% S6
cvx_begin quiet
variable x(2);
    maximize b*x
    subject to
        A*x <= c;
        x(1) >= 3;
        x(2) <= 2;
cvx_end

x
optval = cvx_optval

%% S7
cvx_begin quiet
variable x(2);
    maximize b*x
    subject to
        A*x <= c;
        x(1) >= 3;
        x(2) <= 1;
```

```
cvx_end

x
optval = cvx_optval

%% S8
cvx_begin quiet
variable x(2);
    maximize b*x
    subject to
        A*x <= c;
        x(1) >= 3;
        x(2) <= 2;
        x(2) >= 2
cvx_end

x
optval = cvx_optval

%% S9
cvx_begin quiet
variable x(2);
    maximize b*x
    subject to
        A*x <= c;
        x(1) == 3;
        x(2) <= 1;
cvx_end

x
optval = cvx_optval

%% S10
cvx_begin quiet
variable x(2);
    maximize b*x
    subject to
        A*x <= c;
        x(1) >= 4;
        x(2) <= 1;
cvx_end

x
optval = cvx_optval
```

At each node, I add a constraint and solve the LP relaxation and get the optimal solution. If the solution is not integral, do further branching and add a constraint; if the solution is inntegral, then stop at that node.

The optimal solution is $x = 4,\ y = 0$, the optimal value is 68.

Problem 2

The optimization problem is:

$$\begin{aligned} \text{minimize} \quad & \sum_{j=1}^{n} I_j \\ \text{subject to} \quad & \sum_{i=1}^{n} a_i x_{ij} \leq V \times I_j \qquad \forall j \\ & \sum_{j=1}^{n} x_{ij} = 1 \qquad \forall i \\ & I_{ij},\ x_{ij} \in \{0,\ 1\} \end{aligned}$$

The MATLAB code for IP is:

```
a = [4; 4; 5; 7]
b = [1 1 1 1]

cvx_solver gurobi;
cvx_begin quiet
integer variables I1(4) x1(4,4);
    minimize sum(I1)
    subject to
        (a'*x1) <= 10*I1'
        b*x1' == 1
        1 >= I1 >= 0
        1 >= x1 >= 0
cvx_end

I1
x1
opt_val = cvx_optval
```

The optimal solution to IP is $I_1 = 0,\ I_2 = 1,\ I_3 = 1,\ I_4 = 1$.
$x_{11} = 0,\ x_{12} = 0,\ x_{13} = 0,\ x_{14} = 1$; $x_{21} = 0,\ x_{22} = 0,\ x_{23} = 0,\ x_{24} = 1$;
$x_{31} = 0,\ x_{32} = 0,\ x_{33} = 1,\ x_{34} = 0$; $x_{41} = 0,\ x_{42} = 1,\ x_{43} = 0,\ x_{44} = 0$. The optimal value to IP is 3 .

The MATLAB code for LP relaxation is:

```
cvx_begin quiet
variables I(4) x(4,4);
    minimize sum(I)
    subject to
        (a'*x) <= 10*I'
        b*x' == 1
        1 >= I >= 0
        1 >= x >= 0
cvx_end

I
x
optval = cvx_optval
```

The optimal solution to LP is $I_1 = 0.5,\ I_2 = 0.5,\ I_3 = 0.5,\ I_4 = 0.5$.
$x_{11} = 0.25,\ x_{12} = 0.25,\ x_{13} = 0.25,\ x_{14} = 0.25$;
$x_{21} = 0.25,\ x_{22} = 0.25,\ x_{23} = 0.25,\ x_{24} = 0.25$;
$x_{31} = 0.25,\ x_{32} = 0.25,\ x_{33} = 0.25,\ x_{34} = 0.25$;
$x_{41} = 0.25,\ x_{42} = 0.25,\ x_{43} = 0.25,\ x_{44} = 0.25$. The optimal value to LP is 2 .

$$v^{IP} - v^{LP} = 3 - 2 = 1$$

The integrality gap is 1 .

Problem 3

Let I_i be the decision of whether the seller accept (1) the request S_i or reject it (0), x_{ij} be the decision whether seller i wants to buy product j or not, $x_{ij} \in \{0,\ 1\}$.

The optimization problem is:

$$\begin{aligned} \text{maximize} \quad & \sum_{i}^{n} \mathrm{v_i I_i} \\ \text{subject to} \quad & \sum_{i}^{n} \mathrm{x_{ij} I_i} \le \mathrm{B_j} \qquad \forall \mathrm{j} \\ & \mathrm{I_i} \in \{0,\ 1\} \quad \forall \mathrm{i} \end{aligned}$$

The MATLAB code for LP is:

```
v = [2; 1; 3; 2; 2]
B = [1 2 3]
S = [1, 1, 0;
    0, 0, 1;
    1, 0, 1;
    0, 1, 1;
    0, 1, 0]
```

```
cvx_begin quiet
variables I(5);
    maximize (v'*I)
    subject to
        I'*S <= B
        0 <= I <= 1
cvx_end

I
optval = cvx_optval
```

The optimal solution to IP is $I_1 = 0,\ I_2 = 1,\ I_3 = 1,\ I_4 = 1,\ I_5 = 1$. The optimal value to IP is 8 .

The MATLAB code for IP is:

```
cvx_solver gurobi;
cvx_begin quiet
integer variables I(5);
    maximize (v'*I)
    subject to
        I'*S <= B
        0 <= I <= 1
cvx_end

I
optval = cvx_optval
```

The optimal solution to IP is $I_1 = 0,\ I_2 = 1,\ I_3 = 1,\ I_4 = 1,\ I_5 = 1$. The optimal value to IP is 8 .

$$v^{LP} - v^{IP} = 8 - 8 = 0$$

The intergrality gap is 0 .