杨景兰 Yang Jinglan 121090699

hw9

**Problem 1 (30pts).**
Use the branch-and-bound method to solve the following integer program.

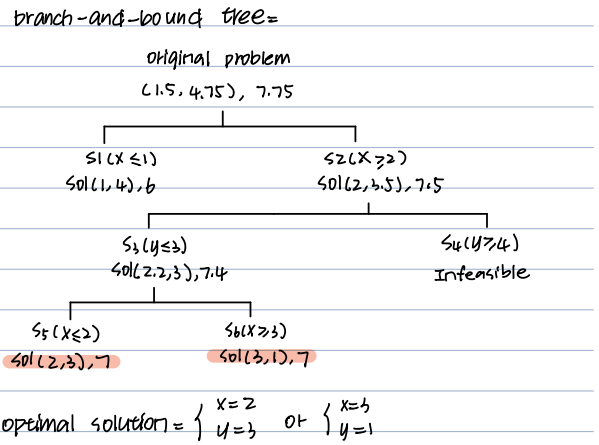
$$\begin{array}{lll} \text{maximize} & 2x+y & \\ \text{subject to} & -3x+2y & \leq 5 \\ & -x-2y & \leq -2 \\ & 5x+2y & \leq 17 \\ & x,y \in \mathbb{Z}. & \end{array}$$

You are allowed to use an LP solver to solve each of the relaxed linear program. Please specify the branch-and-bound tree and what you did at each node .

```
cvx_begin
variables x y
maximize 2*x+y
subject to
    -3*x+2*y<=5;
    -x-2*y<=-2;
    5*x+2*y<=17;
cvx_end
x,y
```

```
------------------------------------------------------------
Status: Solved
Optimal value (cvx_optval): +7.75

x =

    1.5000

y =

    4.7500
```

branch-and-bound tree:




optimal solution = $\begin{cases} x=2 \\ y=3 \end{cases}$ or $\begin{cases} x=3 \\ y=1 \end{cases}$

optimal value = 7

```
cvx_begin
variables x1 y1
maximize 2*x1+y1
subject to
    -3*x1+2*y1<=5;
    -x1-2*y1<=-2;
    5*x1+2*y1<=17;
    x1<=1;
cvx_end
x1,y1
```

```
------------------------------------------------------------
Status: Solved
Optimal value (cvx_optval): +6

x1 =

    1.0000

y1 =

    4.0000
```

```
cvx_begin
variables x2 y2
maximize 2*x2+y2
subject to
    -3*x2+2*y2<=5;
    -x2-2*y2<=-2;
    5*x2+2*y2<=17;
    x2>=2;
cvx_end
x2,y2
```

```
------------------------------------------------------------
Status: Solved
Optimal value (cvx_optval): +7.5

x2 =

    2.0000

y2 =

    3.5000
```

```
cvx_begin
variables x3 y3
maximize 2*x3+y3
subject to
    -3*x3+2*y3<=5;
    -x3-2*y3<=-2;
    5*x3+2*y3<=17;
    x3>=2;
    y3<=3;
cvx_end
x3,y3
```

```
------------------------------------------------------------
Status: Solved
Optimal value (cvx_optval): +7.4

x3 =

    2.2000

y3 =

    3.0000
```

```
cvx_begin
variables x4 y4
maximize 2*x4+y4
subject to
    -3*x4+2*y4<=5;
    -x4-2*y4<=-2;
    5*x4+2*y4<=17;
    x4>=2;
    y4>=4;
cvx_end
x4,y4
```

```
------------------------------------------------------------
Status: Infeasible
Optimal value (cvx_optval): -Inf

x4 =

    NaN

y4 =

    NaN
```

```
cvx_begin
variables x5 y5
maximize 2*x5+y5
subject to
    -3*x5+2*y5<=5;
    -x5-2*y5<=-2;
    5*x5+2*y5<=17;
    x5<=2;
    y5<=3;
cvx_end
x5,y5
```

```
------------------------------------------------------------
Status: Solved
Optimal value (cvx_optval): +7


x5 =

    2.0000


y5 =

    3.0000
```

```
cvx_begin
variables x6 y6
maximize 2*x6+y6
subject to
    -3*x6+2*y6<=5;
    -x6-2*y6<=-2;
    5*x6+2*y6<=17;
    x6>=3;
    y6<=3;
cvx_end
x6,y6
```

```
------------------------------------------------------------
Status: Solved
Optimal value (cvx_optval): +7


x6 =

    3.0000


y6 =

    1.0000
```

**Problem 2 (30pts).**
Consider a seller who sells $m$ different products. For product $j$, there are $B_j$ units in inventory. There are $n$ customers, each customer $i$ is interested in buying a bundle of the product $S_i$, where $S_i \subseteq \{1, ..., m\}$ and is willing to pay a price $v_i$ for it. For each customer, the seller can only decide to accept his entire request $S_i$ or reject him. The objective of the seller is to maximize the revenue.

- Formulate this problem as an integer program.
- Consider the following example $B_1 = 1$, $B_2 = 2$, $B_3 = 3$, $S_1 = \{1, 2\}$, $v_1 = 2$, $S_2 = \{3\}$, $v_2 = 1$, $S_3 = \{1, 3\}$, $v_3 = 3$, $S_4 = \{2, 3\}$, $v_4 = 2$, $S_5 = \{2\}$, $v_5 = 2$. What is one of the optimal solution to the LP (Linear programming) and IP respectively? What is the integrality gap?

(1) Assume for each customer $i$, the seller accept request $X_i = 1$, reject request $X_i = 0$.

maximize $\sum_{i=1}^{n} v_i x_i$

subject to $x_i \in \{0, 1\}$, $\forall i \in [1, n]$

$\sum_{i=1}^{n} S_{ij} x_i \le B_j$, $\forall j \in [1, m]$

(2) maximize $2x_1 + x_2 + 3x_3 + 2x_4 + 2x_5$

subject to $x_1 + x_3 \le 1$

$x_1 + x_4 + x_5 \le 2$

$x_2 + x_3 + x_4 \le 3$

$x_i \in \{0, 1\}$, $\forall i \in [1, 5]$

For LP:

maximize $2x_1 + x_2 + 3x_3 + 2x_4 + 2x_5$

subject to $x_1 + x_3 \le 1$

$x_1 + x_4 + x_5 \le 2$

$x_2 + x_3 + x_4 \le 3$

$0 \le x_i \le 1$, $\forall i \in [1, 5]$

```
cvx_begin
variable x(5)
maximize 2*x(1)+x(2)+3*x(3)+2*x(4)+2*x(5)
subject to
    x(1)+x(3)<=1;
    x(1)+x(4)+x(5)<=2;
    x(2)+x(3)+x(4)<=3;
    x>=0;
    x<=1;
cvx_end
x
```

```
------------------------------------------------------------
Status: Solved
Optimal value (cvx_optval): +8


x =

    0.0000
    1.0000
    1.0000
    1.0000
    1.0000
```

optimal solution for LP and IP is the same: $X = [0, 1, 1, 1, 1]$

integrality gap = 0

**Problem 3 (40pts).**
Suppose we have a set of $n$ many items and a set of $m$ different knapsacks. For each item $i$ and knapsack $j$, the following information is given:

- The item $i$ has value (preference) $v_i$.
- The weight of item $i$ is $a_i$.
- The capacity of knapsack $j$ is at most $C_j$.

a) Formulate an integer program to maximize the total value of items that can be packed in the different knapsack while adhering to the capacity constraint (i.e., the total weight of items in each bag $j$ is not allowed to be larger than $C_j$).

**Hint:** You can introduce variables $x_{ij}$ to denote whether item $i$ is placed in knapsack $j$.

b) Consider the following list of items and bags:

| | 1 | 2 | 3 | 4 | 5 | 6 | 7 |
|---|---|---|---|---|---|---|---|
| Item | Laptop | T-Shirt | Swim. Trunks | Sunglasses | Apples | Opt. Book | Water |
| Value | 2 | 1 | 3 | 2 | 1 | 4 | 2 |
| Weight | 2 | 0.5 | 0.5 | 0.1 | 0.5 | 1 | 1.5 |

| Knapsack 1 | Knapsack 2 |
|---|---|
| $C_1 = 3$ | $C_2 = 2$ |

Formulate the corresponding IP in that case. What are the optimal solutions to the IP and its LP relaxation (you can use MATLAB or CVX to solve the problems)? Is there an integrality gap in this case?

(a) Assume $x_{ij}=1$ denote item $i$ in knapsack $j$, $x_{ij}=0$ denote item $i$ not in knapsack $j$.

maximize $\sum_{i=1}^{n} v_i x_{ij}$

subject to $\sum_{i=1}^{n} a_i x_{ij} \le C_j$, $\forall j \in [1,m]$

$x_{ij} \in \{0,1\}$, $\forall i \in [1,n]$, $\forall j \in [1,m]$

$\sum_{j=1}^{m} x_{ij} \le 1$, $\forall i \in [1,n]$

(b) maximize $2(x_{11}+x_{12}) + (x_{21}+x_{22}) + 3(x_{31}+x_{32}) + 2(x_{41}+x_{42}) + (x_{51}+x_{52}) + 4(x_{61}+x_{62}) + 2(x_{71}+x_{72})$

subject to $2x_{11} + 0.5x_{21} + 0.5x_{31} + 0.1x_{41} + 0.5x_{51} + x_{61} + 1.5x_{71} \le 3$

$2x_{12} + 0.5x_{22} + 0.5x_{32} + 0.1x_{42} + 0.5x_{52} + x_{62} + 1.5x_{72} \le 2$

$x_{ij} \in \{0,1\}$, $\forall i \in [1,n]$, $\forall j \in [1,m]$

$x_{i1} + x_{i2} \le 1$, $\forall i \in [1,7]$

```
cvx_begin
variable x(7,2)
maximize 2*sum(x(1,:))+sum(x(2,:))+3*sum(x(3,:))+2*sum(x(4,:))+sum(x(5,:))+4*sum(x(6,:))+2*sum(x(7,:))
subject to
    2*x(1,1)+0.5*x(2,1)+0.5*x(3,1)+0.1*x(4,1)+0.5*x(5,1)+x(6,1)+1.5*x(7,1)<=3;
    2*x(1,2)+0.5*x(2,2)+0.5*x(3,2)+0.1*x(4,2)+0.5*x(5,2)+x(6,2)+1.5*x(7,2)<=2;
    for i=1:7
        x(i,1)+x(i,2)<=1;
    end
    x>=0;
    x<=1;
cvx_end
x
```

```
------------------------------------------------------------
Status: Solved
Optimal value (cvx_optval): +13.9

x =

    0.3319    0.1181
    0.5350    0.4650
    0.5317    0.4683
    0.5062    0.4938
    0.5350    0.4650
    0.5656    0.4344
    0.6127    0.3873
```

→ optimal solution to LP relaxation

```
from pulp import *

prob=LpProblem("Problem",LpMaximize)

x11=LpVariable("var1",0,1,LpInteger)
x12=LpVariable("var2",0,1,LpInteger)
x21=LpVariable("var3",0,1,LpInteger)
x22=LpVariable("var4",0,1,LpInteger)
x31=LpVariable("var5",0,1,LpInteger)
x32=LpVariable("var6",0,1,LpInteger)
x41=LpVariable("var7",0,1,LpInteger)
x42=LpVariable("var8",0,1,LpInteger)
x51=LpVariable("var9",0,1,LpInteger)
x52=LpVariable("var10",0,1,LpInteger)
x61=LpVariable("var11",0,1,LpInteger)
x62=LpVariable("var12",0,1,LpInteger)
x71=LpVariable("var13",0,1,LpInteger)
x72=LpVariable("var14",0,1,LpInteger)

prob+=2*(x11+x12)+(x21+x22)+3*(x31+x32)+2*(x41+x42)+(x51+x52)+4*(x61+x62)+2*(x71+x72)


prob+=2*x11+0.5*x21+0.5*x31+0.1*x41+0.5*x51+x61+1.5*x71<=3
prob+=2*x12+0.5*x22+0.5*x32+0.1*x42+0.5*x52+x62+1.5*x72<=2
prob+=x11+x12<=1
prob+=x21+x22<=1
prob+=x31+x32<=1
prob+=x41+x42<=1
prob+=x51+x52<=1
prob+=x61+x62<=1
prob+=x71+x72<=1
prob.writeLP("hw9P3.lp")

prob.solve()

print("Status: ", LpStatus[prob.status])
for v in prob.variables():
    print(v.name,'=',v.varValue)
```

```
Objective value:                13.00000000
Enumerated nodes:               0
Total iterations:               0
Time (CPU seconds):             0.00
Time (Wallclock seconds):       0.00

Option for printingOptions changed from normal to all
Total time (CPU seconds):       0.00   (Wallclock seconds):       0.00

Status:  Optimal
var1 = 0.0
var10 = 0.0
var11 = 0.0
var12 = 1.0
var13 = 1.0
var14 = 0.0
var2 = 0.0
var3 = 1.0
var4 = 0.0
var5 = 0.0
var6 = 1.0
var7 = 0.0
var8 = 1.0
var9 = 1.0
```

→ optimal solution for IP

Integrality gap = 13.9 - 13 = 0.9