

Only problem 1 and 6 will be graded.

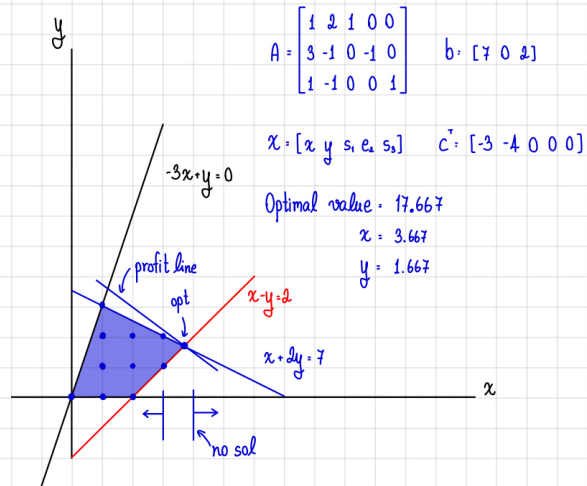
Problem 1 : Integer program

Solve the following program by using linprog function and branch and bound method:

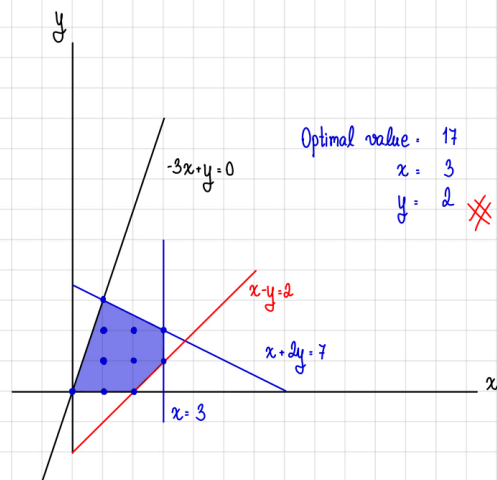
$$\text{Objective : } \max(3x + 4y)$$

$$\begin{array}{ll} & x + 2y \leq 7 \\ & 3x - y \geq 0 \\ s.t. & x - y \leq 2 \\ & x, y \in \mathbb{Z}^+ \cup \{0\} \end{array}$$

$$\begin{aligned}
 1) \quad & \max 3x + 4y \\
 \text{s.t.} \quad & x + 2y \leq 7 \\
 & 3x - y \geq 0 \\
 & x - y \leq 2 \\
 & x, y \in \mathbb{Z} \cup \{0\}
 \end{aligned}$$



add constraint
 $x \leq 3$



```

In [27]: import numpy as np
from copy import deepcopy
from scipy.optimize import linprog
c_T = np.array([-3, -4, 0, 0, 0])
A = np.array(
    [
        [1, 2, 1, 0, 0],
        [3, -1, 0, -1, 0],
        [1, -1, 0, 0, 1],
    ]
)
bound = [[0, None], [0, None], [0, None], [0, None], [0, None]] # bound for each
b = [7, 0, 2]
  
```

```
In [28]: result = linprog(c = c_T, A_eq = A, b_eq=b, bounds=bound, method='simplex')
print("optimal value is {} \n optimal souldion is {}".format(-result.fun, result

optimal value is 17.666666666666668
optimal souldion is [3.66666667 1.66666667 0.          9.33333333 0.          ]

C:\Users\BBB\AppData\Local\Temp\ipykernel_8916\3976203470.py:1: DeprecationWarnin
g: `method='simplex'` is deprecated and will be removed in SciPy 1.11.0. Please u
se one of the HiGHS solvers (e.g. `method='highs'`) in new code.
result = linprog(c = c_T, A_eq = A, b_eq=b, bounds=bound, method='simplex')
```

```
In [29]: #sub_problem 1
c_T = np.array([-3, -4, 0, 0, 0])
A = np.array(
    [
        [1, 2, 1, 0, 0],
        [3, -1, 0, -1, 0],
        [1, -1, 0, 0, 1],
    ]
)
bound = np.array([[0, 3], [0, None], [0, None], [0, None], [0, None]]) # bound f
b = [7, 0, 2]
```

```
In [30]: result = linprog(c = c_T, A_eq = A, b_eq=b, bounds=bound, method='simplex')
print("optimal value is {} \n optimal souldion is {}".format(-result.fun, result

optimal value is 17.0
optimal souldion is [3. 2. 0. 7. 1.]

C:\Users\BBB\AppData\Local\Temp\ipykernel_8916\3976203470.py:1: DeprecationWarnin
g: `method='simplex'` is deprecated and will be removed in SciPy 1.11.0. Please u
se one of the HiGHS solvers (e.g. `method='highs'`) in new code.
result = linprog(c = c_T, A_eq = A, b_eq=b, bounds=bound, method='simplex')
```

```
In [31]: def is_int(x):
    for i in x:
        if(i%1!=0):
            return False
    return True
```

```
In [32]: def solve(c_T,A,b,bound,method):
    result = linprog(c = c_T, A_eq = A, b_eq=b, bounds=bound, method='simplex')
    if(result.success):
        if(is_int(result.x)):
            return result
        else:
            bound_l=deepcopy(bound)
            bound_l[0][1]=int(result.x[0])
            bound_r=deepcopy(bound)
            bound_r[0][0]=int(result.x[0])+1
            result_l=solve(c_T,A,b,bound_l,method)
            result_r=solve(c_T,A,b,bound_r,method)
            if(result_l.fun<result_r.fun):
                return result_l
            else:
                return result_r
    else:
        return
```

```
In [33]: result=solve(c_T,A,b,bound,"simplex")
```

```
print("optimal value is {} \n optimal souldtion is {}".format(-result.fun, result
```

optimal value is 17.0

optimal souldtion is [3. 2. 0. 7. 1.]

C:\Users\BBB\AppData\Local\Temp\ipykernel_8916\2725324835.py:2: DeprecationWarning: `method='simplex'` is deprecated and will be removed in SciPy 1.11.0. Please use one of the HiGHS solvers (e.g. `method='highs'`) in new code.

```
result = linprog(c = c_T, A_eq = A, b_eq=b, bounds=bound, method='simplex')
```

Problem 6 : Instraham

After several crises, Hamtaro is fed up with the manufacturing business and is now looking for new business opportunities. He finds out that opening social media platform could make a hefty sum of money. Moreover, since hamsters do not currently have a social media site, Hamtaro can monopolize the market easily. Therefore, he creates Instraham, the first social media website of hamsters, by hamsters, for hamsters.

After consulting with Koushi-kun, Hamtaro figures out that social network platforms often have the features shown in the table below. For each feature, the CPU load and storage load are shown with its associated business value score.

index	Feature name	CPU load (%)	storage load (%)	business value score
1	A	20	30	10
2	B	10	5	5
3	C	30	10	10
4	D	5	10	3
5	F	15	30	10
6	G	60	70	30
7	H	80	80	80
8	I	10	50	20
9	J	3	50	5

Feature A, and J is mandatory while the rest is optional. The objective is to maximize the business value score of the website while not overloading CPU and storage servers. His engineering friend, Taisho-kun, also suggests him that he could improve the website efficiency by performing the following operations:

- Feature compression. This method will reduce both CPU and storage load by half, but it also reduces the business value to 55% of the original value. Every feature could be compressed, but the number of compressed features in the website is limited to two.
- The usage of storage efficient algorithm. By using this method, the feature storage load is reduced by half but it also doubles the CPU load. However, only feature H, I, J

can use this method. This method could not be used concurrently with feature compression.

From this information, which features should Hamtaro develop? ~~Use Amdahl's law to find the best speedup.~~ Formulate the problem as an integer program and solve for an optimal solution.

Note : This problem is based on the blog

(<https://engineering.fb.com/2021/07/29/data-infrastructure/linear-programming/>)

...

6) Obj $\max (10x_1 + 5x_2 + 10x_3 + 3x_4 + 10x_5 + 30x_6 + 80x_7 + 20x_8 + 5x_9)$

s.t. $20x_1 + 10x_2 + 30x_3 + 5x_4 + 15x_5 + 60x_6 + 80x_7 + 10x_8 + 3x_9 \leq 100$ CPU constraint
 $30x_1 + 5x_2 + 10x_3 + 10x_4 + 30x_5 + 70x_6 + 80x_7 + 50x_8 + 50x_9 \leq 100$ Storage constraint
 $x_1, x_9 = 1$ *

Let $X = [x_1 \ x_2 \ \dots \ x_9]$ $x_i \in \{0,1\}$ #selected feature

*Operation 1
 Let $P = [p_1 \ p_2 \ \dots \ p_9]$ $p_i \in \{0,1\}$ # use operation 1 s.t. $\sum_i p_i \leq 2$ *

$\max (10x_1 + 5x_2 + 10x_3 + 3x_4 + 10x_5 + 30x_6 + 80x_7 + 20x_8 + 5x_9 - 0.45(10p_1 + 5p_2 + 10p_3 + 3p_4 + 10p_5 + 30p_6 + 80p_7 + 20p_8 + 5p_9))$
 s.t. $20x_1 + 10x_2 + 30x_3 + 5x_4 + 15x_5 + 60x_6 + 80x_7 + 10x_8 + 3x_9 - 0.5(20p_1 + 10p_2 + 30p_3 + 5p_4 + 15p_5 + 60p_6 + 80p_7 + 10p_8 + 3p_9) \leq 100$
 $30x_1 + 5x_2 + 10x_3 + 10x_4 + 30x_5 + 70x_6 + 80x_7 + 50x_8 + 50x_9 - 0.5(30p_1 + 5p_2 + 10p_3 + 10p_4 + 30p_5 + 70p_6 + 80p_7 + 50p_8 + 50p_9) \leq 100$
 $x_i \geq p_i$ # cannot perform this operation on unselected feature *

*Operation 2
 Let $Q = [q_1 \ q_2 \ q_3]$ $q_i \in \{0,1\}$ # use operation 2

$\max (10x_1 + 5x_2 + 10x_3 + 3x_4 + 10x_5 + 30x_6 + 80x_7 + 20x_8 + 5x_9 - 0.45(10p_1 + 5p_2 + 10p_3 + 3p_4 + 10p_5 + 30p_6 + 80p_7 + 20p_8 + 5p_9))$
 s.t. $20x_1 + 10x_2 + 30x_3 + 5x_4 + 15x_5 + 60x_6 + 80x_7 + 10x_8 + 3x_9 - 0.5(20p_1 + 10p_2 + 30p_3 + 5p_4 + 15p_5 + 60p_6 + 80p_7 + 10p_8 + 3p_9) + 80q_1 + 10q_2 + 3q_3 \leq 100$
 $30x_1 + 5x_2 + 10x_3 + 10x_4 + 30x_5 + 70x_6 + 80x_7 + 50x_8 + 50x_9 - 0.5(30p_1 + 5p_2 + 10p_3 + 10p_4 + 30p_5 + 70p_6 + 80p_7 + 50p_8 + 50p_9) - 0.5(80q_1 + 50q_2 + 50q_3) \leq 100$
 $p_7 + q_1 \leq 1$
 $p_8 + q_2 \leq 1$
 $p_9 + q_3 \leq 1$
 $x_i \geq q_i$

conclude

$\max (10x_1 + 5x_2 + 10x_3 + 3x_4 + 10x_5 + 30x_6 + 80x_7 + 20x_8 + 5x_9 - 0.45(10p_1 + 5p_2 + 10p_3 + 3p_4 + 10p_5 + 30p_6 + 80p_7 + 20p_8 + 5p_9))$
 s.t. $20x_1 + 10x_2 + 30x_3 + 5x_4 + 15x_5 + 60x_6 + 80x_7 + 10x_8 + 3x_9 - 0.5(20p_1 + 10p_2 + 30p_3 + 5p_4 + 15p_5 + 60p_6 + 80p_7 + 10p_8 + 3p_9) + 80q_1 + 10q_2 + 3q_3 \leq 100$
 $30x_1 + 5x_2 + 10x_3 + 10x_4 + 30x_5 + 70x_6 + 80x_7 + 50x_8 + 50x_9 - 0.5(30p_1 + 5p_2 + 10p_3 + 10p_4 + 30p_5 + 70p_6 + 80p_7 + 50p_8 + 50p_9) - 0.5(80q_1 + 50q_2 + 50q_3) \leq 100$
 $x_1, x_9 = 1$
 $\sum_i p_i \leq 2$
 $x_i - p_i \geq 0$ for $i: 1 \rightarrow 9$
 $p_7 + q_1 \leq 1$
 $p_8 + q_2 \leq 1$
 $p_9 + q_3 \leq 1$
 $x_i - q_i \geq 0$ for $i: 7 \rightarrow 9$

In [34]: `import pulp`

`prob = pulp.LpProblem("Binary_Problem", pulp.LpMaximize)`

```
#selected feature
```

```
x1 = pulp.LpVariable('x1', lowBound=0, cat='Binary')
x2 = pulp.LpVariable('x2', lowBound=0, cat='Binary')
x3 = pulp.LpVariable('x3', lowBound=0, cat='Binary')
x4 = pulp.LpVariable('x4', lowBound=0, cat='Binary')
x5 = pulp.LpVariable('x5', lowBound=0, cat='Binary')
x6 = pulp.LpVariable('x6', lowBound=0, cat='Binary')
x7 = pulp.LpVariable('x7', lowBound=0, cat='Binary')
x8 = pulp.LpVariable('x8', lowBound=0, cat='Binary')
x9 = pulp.LpVariable('x9', lowBound=0, cat='Binary')
```

```
#use oper1
```

```
p1 = pulp.LpVariable('p1', lowBound=0, cat='Binary')
p2 = pulp.LpVariable('p2', lowBound=0, cat='Binary')
p3 = pulp.LpVariable('p3', lowBound=0, cat='Binary')
p4 = pulp.LpVariable('p4', lowBound=0, cat='Binary')
p5 = pulp.LpVariable('p5', lowBound=0, cat='Binary')
p6 = pulp.LpVariable('p6', lowBound=0, cat='Binary')
p7 = pulp.LpVariable('p7', lowBound=0, cat='Binary')
p8 = pulp.LpVariable('p8', lowBound=0, cat='Binary')
p9 = pulp.LpVariable('p9', lowBound=0, cat='Binary')
```

```
#use oper2
```

```
q7 = pulp.LpVariable('q7', lowBound=0, cat='Binary')
q8 = pulp.LpVariable('q8', lowBound=0, cat='Binary')
q9 = pulp.LpVariable('q9', lowBound=0, cat='Binary')
```

```
#obj
```

```
prob += 10*x1+5*x2+10*x3+3*x4+10*x5+30*x6+80*x7+20*x8+5*x9-0.45*(10*p1+5*p2+10*p
```

```
#st
```

```
prob += 20*x1+10*x2+30*x3+5*x4+15*x5+60*x6+80*x7+10*x8+3*x9-0.5*(20*p1+10*p2+30*
prob += 30*x1+5*x2+10*x3+10*x4+30*x5+70*x6+80*x7+50*x8+50*x9-0.5*(30*p1+5*p2+10*
prob += x1==1
prob += x9==1
prob += p1+p2+p3+p4+p5+p6+p7+p8+p9<=2
prob += x1-p1>=0
prob += x2-p2>=0
prob += x3-p3>=0
prob += x4-p4>=0
prob += x5-p5>=0
prob += x6-p6>=0
prob += x7-p7>=0
prob += x8-p8>=0
prob += x9-p9>=0
prob += p7+q7<=1
prob += p8+q8<=1
prob += p9+q9<=1
prob += x7-q7>=0
prob += x8-q8>=0
prob += x9-q9>=0
```

```
prob.solve()
```

```
print(f"Optimal value of x1: {x1.varValue}")
print(f"Optimal value of x2: {x2.varValue}")
print(f"Optimal value of x3: {x3.varValue}")
print(f"Optimal value of x4: {x4.varValue}")
print(f"Optimal value of x5: {x5.varValue}")
```

```

print(f"Optimal value of x6: {x6.varValue}")
print(f"Optimal value of x7: {x7.varValue}")
print(f"Optimal value of x8: {x8.varValue}")
print(f"Optimal value of x9: {x9.varValue}")

print(f"Optimal value of p1: {p1.varValue}")
print(f"Optimal value of p2: {p2.varValue}")
print(f"Optimal value of p3: {p3.varValue}")
print(f"Optimal value of p4: {p4.varValue}")
print(f"Optimal value of p5: {p5.varValue}")
print(f"Optimal value of p6: {p6.varValue}")
print(f"Optimal value of p7: {p7.varValue}")
print(f"Optimal value of p8: {p8.varValue}")
print(f"Optimal value of p9: {p9.varValue}")

print(f"Optimal value of q7: {q7.varValue}")
print(f"Optimal value of q8: {q8.varValue}")
print(f"Optimal value of q9: {q9.varValue}")

print("objective value",pulp.value(prob.objective))
print("total cpu usage",pulp.value(20*x1+10*x2+30*x3+5*x4+15*x5+60*x6+80*x7+10*x8+
print("total storage usage",pulp.value(30*x1+5*x2+10*x3+10*x4+30*x5+70*x6+80*x7+

```

```

Optimal value of x1: 1.0
Optimal value of x2: 1.0
Optimal value of x3: 1.0
Optimal value of x4: 0.0
Optimal value of x5: 0.0
Optimal value of x6: 0.0
Optimal value of x7: 1.0
Optimal value of x8: 0.0
Optimal value of x9: 1.0
Optimal value of p1: 1.0
Optimal value of p2: 0.0
Optimal value of p3: 0.0
Optimal value of p4: 0.0
Optimal value of p5: 0.0
Optimal value of p6: 0.0
Optimal value of p7: 1.0
Optimal value of p8: 0.0
Optimal value of p9: 0.0
Optimal value of q7: 0.0
Optimal value of q8: 0.0
Optimal value of q9: 1.0
objective value 69.5
total cpu usage 96.0
total storage usage 95.0

```

ANS : ใช้feature A,B,C,H,J โดยcompress A,H และ เปลี่ยนalgo J