

Q1

1. MILP Linearization

(1)

α	β	γ	LHS	$\alpha + \beta - \gamma \leq 1$	$\alpha - \beta + \gamma \leq 1$	$-\alpha + \beta + \gamma \leq 1$	RHS	LHS=RHS?
0	0	0	T	T	T	T	T	T
0	0	1	T	T	T	T	T	T
0	1	0	T	T	T	T	T	T
0	1	1	F	T	T	F	F	T
1	0	0	T	T	T	T	T	T
1	0	1	F	T	F	T	F	T
1	1	0	F	F	T	T	F	T
1	1	1	T	T	T	T	T	T

(2)

α	β	γ	LHS	$\alpha + \beta - 1 \leq \gamma$	$\gamma \leq \alpha$	$\gamma \leq \beta$	RHS	LHS=RHS?
0	0	0	T	T	T	T	T	T
0	0	1	F	T	F	F	F	T
0	1	0	T	T	T	T	T	T
0	1	1	F	T	F	T	F	T
1	0	0	T	T	T	T	T	T
1	0	1	F	T	T	F	F	T
1	1	0	F	F	T	T	F	T
1	1	1	T	T	T	T	T	T

(3)

LHS

$$\beta = 0 \Rightarrow 0 = \gamma$$

$$\beta = 1 \Rightarrow x = \gamma$$

RHS

$$x - M \leq \gamma = 0 \leq x$$

$$0 \leq \gamma = x \leq M$$

若要使 LHS=RHS

$$\begin{cases} x - M \leq 0 \leq x \\ 0 \leq x \leq M \end{cases}$$

$$\Rightarrow 0 \leq x \leq M$$

$$\therefore x \leq 2020$$

$$\therefore M \geq 2020$$

2.

- (1) Yes, 因為將 Message M_0 與 M_1 合併, 這兩個的 period 一樣, 且 receiver 也一樣, 所以會同時傳送. 因此將兩個合併會減少傳送的 bits 數

$$\text{原本: } (8+44+3) + (16+44+3) = 118$$

$$\text{合併: } (16+44+3) = 63$$

∴ 合併效率高

- (2) No. 因為 M_0' 的 Receiver 跟 M_2 的 Sender 是一樣的, 如果 M_2 的 message 是要依據 M_0' 的結果的話, 就不能將 M_0' 跟 M_2 merge 在一起.

- (3) Yes, 因為 M_3 與 M_0' Sender 一樣, 所以可以 merge 在一起, 但是 M_3 的 period 是 100ms, 所以會多傳一次, 但只是多了一些 Bits 而已. 原本 100ms 全部 message 需要: $[(16+44+3) + (16+44+3)] \times 2$
 $+ (16+44+3) = 315$

$$\text{若將 } M_0' \text{ 與 } M_3 \text{ 合併則需: } [(32+44+3) + (16+44+3)] \times 2 \\ = 284$$

∴ 每 100ms 所傳的 bits 減少 31 bits.

Q3

此部分沿用作業一的方法計算 response time，接著用 Annealing for Priority Assignment 來挑選鄰居，隨機選取兩個 priority swap 看是否符合規則，若不符合規則，加上 1000 的 response time penalty，符合規則，若交換過的結果較佳便保留，若較差的話則有 $\exp(-c/T)$ 的機率會保留。

Source code 如下:

```
import math
import random
from copy import deepcopy
def cost(array):
    message_number = int(array[0][0])
    R_array = [0] * message_number
    tau = array[1][0]
    max_B = [0] * message_number
    #calculate block time
    for i in range(message_number):
        for j in range(i,message_number):
            if(max_B[i] < array[j+2][1]):
                max_B[i] = array[j+2][1]
    s = 0
    for i in range(message_number):
        Q = max_B[i]
        temp = 0
        #first time calculate
        for j in range(i):
            temp = temp + math.ceil((Q+tau)/array[j+2][2])*array[j+2][1]
        RHS = max_B[i] + temp
        while(RHS != Q):
            Q = RHS
            temp = 0
            for j in range(i):
                temp = temp + math.ceil((Q+tau)/array[j+2][2])*array[j+2][1]
            RHS = max_B[i] + temp
            if(RHS + array[i+2][1] > array[i+2][2]):
                return 0
        R_array[i] = round(Q + array[i+2][1],2)
```

```

        s = s + R_array[i]
#         print(R_array[i])
    return s
def main():
    f = open("Input.dat")
    array = []
    array_p = []

    for line in f:
        array.append([float(x) for x in line.split()]) #transfer input
to array
    f.close()
    T = 100
    r = 0.9995
    s = cost(array)
    s_f = s
    count = 0
    while T > 1:
        array_p = deepcopy(array)
        swap_num = []
        number = range(2,19)
        swap_num = random.sample(number,2)
        array_p[swap_num[0]],array_p[swap_num[1]] = array_p[swap_num[1]
],array_p[swap_num[0]]
        s_p = cost(array_p)
        c = s_p - s
        if s_p != 0:
            if s_p < s_f:
                s_f = s_p
            if c <= 0:
                s = s_p
                array = deepcopy(array_p)
            if c > 0:
                a = math.exp(-c/T)
                if random.uniform(0,1) < a:
                    s = s_p
                    array = deepcopy(array_p)
        else:

```

```

        c = 1000
        a = math.exp(-c/T)
        if random.uniform(0,1) < a:
            s = s_p
            array = deepcopy(array_p)

        #count = count + 1
        T = r*T
    print("Total value:",s_f)
    #print(count)
    for i in range(17):
        for j in range(17):
            if array[j+2][0] == i:
                print(j)
if __name__ == '__main__':
    main()

```

Output 結果:

```

PS C:\Users\Bill\Desktop\car\hw2> & C:/Users/Bill/AppData/Local/Programs/Python/Python37/python.exe c:/Users/Bill/Desktop/car/hw2/hw2.py
Total value: 284.28000000000002
16
4
5
2
0
6
9
1
3
7
15
14
11
13
10
8
12

```