6332009521 ชินพงศ์ นิธิพงศธี

1) จะใช้ branch & bound method ในการแก้

Subproblem 1 : obj : max (3x+4y)

s.t. $x+2y \leq 7$

$3x-y \geq 0$

$x-y \leq 2$; $x,y \geq 0$

```
     con: array([0.00000000e+00, 0.00000000e+00, 2.22044605e-16])
     fun: -17.666666666666668
 message: 'Optimization terminated successfully.'
     nit: 4
   slack: array([], dtype=float64)
  status: 0
 success: True
       x: array([3.66666667, 1.66666667, 0.        , 9.33333333, 0.        ])

optimal value is 17.666666666666668
optimal soultion is [3.66666667 1.66666667]
```

∴ optimal value : 17.67

optimal solution : (x,y) = (3.67, 1.67)

ยังไม่เป็นจำนวนเต็ม

$x \geq 4$ | $x \leq 3$

Subproblem 2 : obj : max (3x+4y)

s.t. $x+2y \leq 7$

$3x-y \geq 0$

$x-y \leq 2$

$x \geq 4$; $x,y \geq 0$

```
     con: array([0.00000000e+00, 0.00000000e+00, 2.22044605e-16, 3.33333333e-01])
     fun: -17.666666666666668
 message: "Phase 1 of the simplex method failed to find a feasible solution. The pseudo
     nit: 4
   slack: array([], dtype=float64)
  status: 2
 success: False
       x: array([3.66666667, 1.66666667, 0.        , 9.33333333, 0.        ,
       0.        ])

optimal value is 17.666666666666668
optimal soultion is [3.66666667 1.66666667]
```

∴ Infeasible

Subproblem 3 : obj : max (3x+4y)

s.t. $x+2y \leq 7$

$3x-y \geq 0$

$x-y \leq 2$

$x \leq 3$; $x,y \geq 0$

```
     con: array([0., 0., 0., 0.])
     fun: -17.0
 message: 'Optimization terminated successfully.'
     nit: 5
   slack: array([], dtype=float64)
  status: 0
 success: True
       x: array([3., 2., 0., 7., 1., 0.])

optimal value is 17.0
optimal soultion is [3. 2.]
```

∴ optimal value : 17

optimal solution : (x,y) = (3,2)

เป็นจำนวนเต็มแล้ว

ดังนั้น optimal value : 17

optimal solution : (x,y) = (3,2) ✳

6332009521 ชินณภัทธ์ นิธิพรศิธี

6) ให้เลข 1,2,...,9 แทนอักษร A,B,...,J ตามลำดับ

ให้ n_i แทนการมีอยู่ของ feature i ที่ไม่ได้ improve เพิ่มเติม ซึ่ง $n_i \in \{0,1\}$ เมื่อ $i \in \{1,2,...,9\}$

ให้ c_i แทนการมีอยู่ของ feature i ที่ต่างกับการทำ feature compression ซึ่ง $c_i \in \{0,1\}$ เมื่อ $i \in \{1,2,...,9\}$

ให้ a_i แทนการมีอยู่ของ feature i ที่ต่างกับการทำ storage efficient algorithm ซึ่ง $a_i \in \{0,1\}$ เมื่อ $i \in \{7,8,9\}$

ให้ cpu_i แทน CPU load ของ feature i เมื่อ $i \in \{1,2,...,9\}$

ให้ sto_i แทน storage load ของ feature i เมื่อ $i \in \{1,2,...,9\}$

ให้ val_i แทน business value score ของ feature i เมื่อ $i \in \{1,2,...,9\}$

จะได้ว่า $n_1 + c_1 = 1$

$n_j + c_j \leq 1 \quad ; j \in \{2,3,...,6\}$

$n_7 + c_7 + a_7 \leq 1$

$n_8 + c_8 + a_8 \leq 1$

$n_9 + c_9 + a_9 = 1$

คิด limited number : $c_1 + c_2 + ... + c_9 \leq 2$

คิด CPU load : $\sum_{i=1}^{9} cpu_i n_i + \sum_{i=1}^{9} 0.5\, cpu_i c_i + \sum_{i=7}^{9} 2 cpu_i a_i \leq 100$

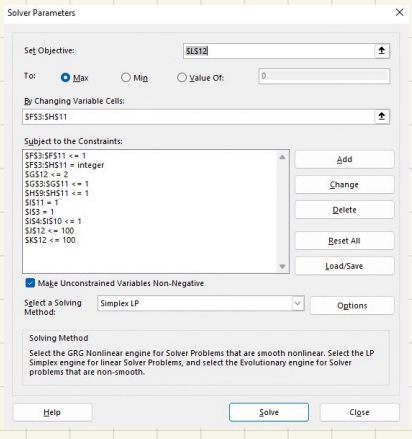
คิด storage load: $\sum_{i=1}^{9} sto_i n_i + \sum_{i=1}^{9} 0.5\, sto_i c_i + \sum_{i=7}^{9} 0.5\, sto_i a_i \leq 100$

ต้องการหา max ของ business value score : $\max \left(\sum_{i=1}^{9} val_i n_i + \sum_{i=1}^{9} 0.55\, val_i c_i + \sum_{i=7}^{9} val_i a_i \right)$

จัดทำการสร้างตารางใน excel ดังภาพ

	A	B	C	D	E	F	G	H	I	J	K	L	M
1													
2			CPU load	Storage Load	Business value score	n	c	a	n + c (+ a)	Actual CPU load	Actual storage load	Actual business value score	
3		1	20	30	10	0	0	0	0	0	0	0	
4		2	10	5	5	0	0	0	0	0	0	0	
5		3	30	10	10	0	0	0	0	0	0	0	
6		4	5	10	3	0	0	0	0	0	0	0	
7		5	15	30	10	0	0	0	0	0	0	0	
8		6	60	70	30	0	0	0	0	0	0	0	
9		7	80	80	80	0	0	0	0	0	0	0	
10		8	10	50	20	0	0	0	0	0	0	0	
11		9	3	50	5	0	0	0	0	0	0	0	
12					Total		0			0	0	0	<- Objective function
13													

และเขียน constraints ทั้งหมดใส่ solver ดังนี้



6332009521 ชินพัทธ์ นิธินธศธี

ซึ่งเมื่อทำการ solve ออกมาแล้วจะได้ผลลัพธ์ดังนี้

		CPU load	Storage Load	Business value score	n	c	a	n + c (+ a)	Actual CPU load	Actual storage load	Actual business value score	
A	1	20	30	10	0	1	0	1	10	15	5.5	
B	2	10	5	5	1	0	0	1	10	5	5	
C	3	30	10	10	1	0	0	1	30	10	10	
D	4	5	10	3	0	0	0	0	0	0	0	
F	5	15	30	10	0	0	0	0	0	0	0	
G	6	60	70	30	0	0	0	0	0	0	0	
H	7	80	80	80	0	1	0	1	40	40	44	
I	8	10	50	20	0	0	0	0	0	0	0	
J	9	3	50	5	0	0	1	1	6	25	5	
				Total		2			96	95	69.5	<- Objective function

สรุปได้ว่า optimal value = max (possible business value score) = 69.5 ※

ซึ่ง optimal solution คือ เลือกทำ feature A, B, C, H, J โดยใช้ feature compression กับ feature A, H และใช้ storage efficient algorithm กับ feature J ※