**이산최적화 HW#2**

**20172050 박규남**

**1.**

Let $x_i$ denote whether vertex $i$ is visited.

Let $y_{ij}$ denote the number of team members traveling from $v_i$ to $v_j$.

Let $b_{ij} = 1$ if $y_{ij} = 1$, 0 o/w

Let $v_0$ denote the depot.

Let $c_i$ denote the number of resources required at vertex $v_i$.

Let $s_i$ denote the start time at $v_i$.

$$\max \Sigma_i \sigma_i x_i$$

$$s.t.\ \Sigma_{j=1}^{n} y_{0j} = \Sigma_{i=1}^{n} y_{i0} = n,$$

$$\Sigma_{j=1}^{n} y_{ik} = \Sigma_{i=1}^{n} y_{kj}\ \forall k,$$

$$c_k x_k = \Sigma_{j=1}^{n} y_{kj}\ \forall k,$$

$$y_{ij} = n * b_{ij},$$

$$s_i + t_{ij} + a_i = s_j + M(1 - b_{ij}),$$

$$o_i = s_i\ \forall i,$$

$$s_i \le c_i\ \forall i,$$

$$s_0 \le\ T_{max},$$

$$x_i \in \{0,1\},$$

$$b_{ij} \in \{0,1\}$$

**2.**

(a)Budget is limited to \$100m. For Northwest, there are four alternatives which are exclusive: digital circuit lab, large lecture room, heat transfer lab, and computer-aided design expansion. For Southeast, there are two competing options: faculty office and manufacturing research center. Tunnel is dependent on a new manufacturing research center.

(b)

$Let\ x_i = 1$ if project $i$ is selected. 0 o/w.

$$\max 9x_1 + 2x_2 + 10x_3 + 2x_4 + 5x_5 + 8x_6 + 10x_7 + x_8$$

$$s.t.\ 48x_1 + 20.8x_2 + 32x_3 + 28x_4 + 44x_5 + 17.2x_6 + 36.8x_7 + 1.2x_8 \leq 100$$

$$x_1 + x_4 + x_5 + x_6 \leq 1$$

$$x_2 + x_7 \leq 1$$

$$x_8 \leq x_7$$

$$x_i \in \{0,1\}\ \forall i$$

(c)

```
                                ------------
Total (root+branch&cut) =    0.01 sec. (0.04 ticks)
solution: [-0.0, -0.0, 1.0, 0.0, 0.0, 1.0, 1.0, 1.0]
objective value = 29.0
```

Objective value is 29. The solution is to build Computer Vision lab, computer-aided design expansion, manufacturing research center, and tunnel.

**3.**

(a)

Let $c_i$ denote cost of work pattern $i$, $i = \{1,2,3,4,5,6,7,8\}$.

Let $x_i$ denote whether work pattern $i$ is selected.

Let $a_{ij}$ denote the availability of flight $j$ in work pattern $i$, , $j = \{1,2,3,4,5,6\}$.

$$\min \Sigma c_i x_i$$

$$s.t. \Sigma_{\mathrm{i}} x_i a_{ij} = 1 \ \forall j$$

$$x_i \in \{0,1\}$$

(b)

```
                          ------------
Total (root+branch&cut) =    0.00 sec. (0.03 ticks)
solution: [0.0, 0.0, 1.0, -0.0, 1.0, 0.0, 0.0, -0.0]
objective value = 2.84
```

Objective value is 8.6. The solution is to choose work pattern #3 and #5.

**4.**

(a)

Let $c_{ij}$ denote the cost of modifying plant $j$ to produce model $i$. If plant $j$ can't afford to produce model $i$, $c_{ij} = M$. $i = \{1,2,3,4\}$, $j = \{1,2,3,4\}$

$$\min \Sigma c_{ij} x_{ij}$$

$$s.t. \Sigma_{\mathrm{i}} x_{ij} = 1 \ \forall j$$

$$\Sigma_{\mathrm{j}} x_{ij} = 1 \ \forall i$$

$$x_{ij} \geq 0$$

(b)

If the optimal basis B has $det(B) = \pm 1$, then the linear programming relaxation solves IP.

(c)

```
                               ------------
Total (root+branch&cut) =    0.00 sec. (0.10 ticks)
solution: ['x_1_1', 'x_3_2', 'x_2_3', 'x_4_4']
objective value = 115.0
```

Objective value is 8.6. The solution is to produce model 1 in plant #1, model 2 in plant #3, model 3 in plant #2, model 4 in plant #4.

**5.**

Model #1:

Solving TSP as an assignment problem is not enough since the solution can have sub-tour. Thus, adding sub-tour elimination constraints are required to avoid it. However, there are exponentially many constraints if we naively consider it. As a bypass, we can iteratively include sub-tour constraint when it actually happens.

The procedure is described in TSP_model_1.py file attached. The result is captured below:

```
Total (root+branch&cut) =    0.02 sec. (11.94 ticks)
14th iteration
solution: [1, 5, 13, 4, 10, 2, 11, 12, 8, 9, 18, 6, 17, 15, 16, 3, 14, 7]
objective value = 231.0
```

The objective value is 231.

The optimal solution is

1→5→13→4→10→2→11→12→8→9→18→6→17→15→16→3→14→7

Model #2:

In this model, we replace subtour elimination constraints by deploying variable $u_i$, which means the number of nodes visited after visiting node $i$.

The procedure is described in TSP_model_2.py file attached. The result is captured below:

```
Total (root+branch&cut) =    0.26 sec. (91.25 ticks)
solution: [1, 5, 13, 4, 10, 2, 11, 12, 8, 9, 6, 18, 17, 15, 16, 3, 14, 7]
objective value = 231.0
```

The objective value is 231.

The optimal solution is

1→5→13→4→10→2→11→12→8→9→6→18→17→15→16→3→14→7

Model #3:

This model adopts different subtour constraints under quadratic objective statement. Here I applied QAP approach which only deals with y variables. This approach took a longest processing time compared with other models.

The procedure is described in TSP_model_3.py file attached. The result is captured below:

```
Total (root+branch&cut) =    4.55 sec. (3962.18 ticks)
objective value = 231.0
solution: ['4', '13', '5', '1', '7', '14', '3', '16', '15', '17', '18', '6', '9', '8', '12', '11', '2', '10']
```

The objective value is 231.

The optimal solution is

1→7→14→3→16→15→17→18→6→9→8→12→11→2→10→4→13→5