Hint of Assignment 4

Algorithm Design and Analysis

December 6, 2015

1 Duality

1. There are at least two methods to solve this problem. They are shown as follows.

   - METHOD 1 Set the object function to be a constant number, say, 0.
   - METHOD 2 Without losing generality, suppose the original linear-inequality feasibility problem (LIFP) is given like this

     $$\begin{array}{rcl} \sum_j a_{ij}x_j & \leqslant & b_j \\ x_j & \geqslant & 0 \end{array}$$

     , which has m inequalities in total. We can add a variable x_0 and construct an LP like this

     $$\begin{array}{rrcl} \max & -x_0 & & \\ \text{s.t.} & \sum_j a_{ij}x_j - x_0 & \leqslant & b_j \\ & x_j & \geqslant & 0 \\ & x_0 & \geqslant & 0 \end{array}$$

     If optimal object value is 0, then the original LIFP is feasible, else not.

2. There are at least two methods to solve this problem. They are shown as follows.

   - METHOD 1 Without losing generality, suppose the LP is given like this:

     $$\begin{array}{rc} \min & c^T x \\ s.t. & Ax \leqslant b \\ & x \geqslant 0 \end{array}$$

     Then its duality is

     $$\begin{array}{rc} \max & y^T b \\ s.t. & y \leqslant 0 \\ & y^T A \leqslant c^T \end{array}$$

     From strong duality, we know that optimal values of the two object functions are the same. In other words, $c^T x \geqslant OPT \geqslant y^T b$. As a result, we can write the following linear-inequality feasible problem, which can be settled with the algorithm for linear-inequality feasible problem.

$$\begin{aligned} Ax &\leqslant b \\ x &\geqslant 0 \\ y &\leqslant 0 \\ y^T A &\leqslant c^T \\ c^T x &= y^T b \end{aligned}$$

- METHOD 2 Without losing generality, suppose the LP and the corresponding DP are the ones in METHOD 1, then we can easily get their corresponding LIFPs and feasible solutions with the given algorithm. Suppose the feasible solution for LP is x_0, and the one for DP is y_0, then according to weak duality, optimal value of object function lies between $c^T x_0$ and $y_0^T b$. Then a constraint $c^T x \leqslant (c^T x_0 + y_0^T b)/2$ is added to the LIFP of LP, and check whether the new LIFP has any feasible solution. If it does, then the optimal value lies between $(c^T x_0 + y_0^T b)/2$ and $y_0^T b$, so we can add constraint $c^T x \leqslant (c^T x_0 + y_0^T b)/2$ to the LP, construct the new DP, and repeat the process until $c^T x_0$ and $y_0^T b$ are close enough. If it does not, we add constraint $y^T b \geqslant (c^T x_0 + y_0^T b)/2$ to DP, construct a new LP according to the new DP, and repeat the process until $c^T x_0$ and $y_0^T b$ are close enough.

2 Airplane Landing Problem

We use x_i to denote the landing time of airplane i, and use z to denote the smallest gap. Then

$$\begin{array}{lrl} \max & z & \\ s.t. & x_i \geqslant s_i & \text{for all } i = 1, 2, \cdots, n \\ & x_i \leqslant t_i & \text{for all } i = 1, 2, \cdots, n \\ & x_{i+1} - x_i \geqslant z & \text{for all } i = 1, 2, \cdots, n-1 \end{array}$$

3 Interval Scheduling Problem

1. This question has at least five methods. In most methods, we can suppose $x_{ij} = \begin{cases} 1, & \text{course } i \text{ uses classroom } j \\ 0, & \text{otherwise} \end{cases} \quad (i, j = 1, 2, \cdots, n).$

   - METHOD 1 Notice that if i and j are conflict iff $|S_i - S_j| + |F_i - F_j| < |F_i - S_i| + |F_j - S_j|$, and i and j are compatible iff $|S_i - S_j| + |F_i - F_j| > |F_i - S_i| + |F_j - S_j|$, because all F_is and S_is are different. Use A_{ij} to denote $e^{(|S_i - S_j| + |F_i - F_j|) - (|F_i - S_i| + |F_j - S_j|)}$, then $A_{ij} < 1$ iff i and j are conflict, $A_{ij} > 1$ iff they are compatible. Based on this, the ILP can be written as:

$$
\begin{array}{lrcll}
\max & \sum_{i=1}^{n}\sum_{j=1}^{m} x_{ij} & & & \\
\text{s.t.} & \sum_{j=1}^{m} x_{ij} & \leqslant & 1 & \text{for all } i = 1,2,\cdots,n \\
 & x_{ik} + x_{jk} & \leqslant & 1 + A_{ij} & \text{for all } i,j = 1,2,\cdots,n \text{ and } i \neq j; k = 1,2,\cdots,m \\
 & x_{ij} & \in & \{0,1\} & \text{for all } i = 1,2,\cdots,n; j = 1,2,\cdots,m
\end{array}
$$

- METHOD 2 First we use the following algorithm to determine whether two courses can use the same classroom or not. The result is saved in a two-dimensional array C. The ILP can be written as:

Algorithm 1 Problem 3(1)

```
for i = 1, 2, ..., n do
    for j = i, i + 1, ..., n do
        if i == j then
            C[i][j] = C[j][i] = Compatible;
        else
            if F_i < S_j or F_j < S_i then
                C[i][j] = C[j][i] = Compatible;
            else
                C[i][j] = C[j][i] = Conflict;
return C;
```

$$
\begin{array}{lrcll}
\max & \sum_{i=1}^{n}\sum_{j=1}^{m} x_{ij} & & & \\
\text{s.t.} & \sum_{j=1}^{m} x_{ij} & \leqslant & 1 & \text{for all } i = 1,2,\cdots,n \\
 & x_{ik} + x_{jk} & \leqslant & 1 & \text{for all conflict } i \text{ and } j\text{, and all } k \\
 & x_{ij} & \in & \{0,1\} & \text{for all } i = 1,2,\cdots,n; j = 1,2,\cdots,m
\end{array}
$$

- METHOD 3 Sort $S_1, F_1, \cdots, S_n, F_n$ ascendantly and get $T_1, \cdots, T_{2n}$, which will split the whole time window into $2n-1$ time slices. Suppose Ω_k denotes the set of courses that will occupy the time slice $[T_k, T_{k+1}]$. Use x_i to denote whether course i is chosen or not, then the LP is

$$
\begin{array}{lrcll}
\max & \sum_{i=1}^{n} x_i & & & \\
\text{s.t.} & \sum_{i\in\Omega_k} x_i & \leqslant & m & k = 1,\cdots,2n-1 \\
 & x_i & \in & \{0,1\} & i = 1,2,\cdots,n
\end{array}
$$

- METHOD 4 Sort all finishing time such that $F_1 \leqslant F_2 \leqslant \cdots \leqslant F_n$. In this situation, if x_{ij} and x_{kj} both equal 1 ($i < k$), then $F_i \leqslant S_k$, so $F_i(x_{ij} + x_{kj} - 1) \leqslant S_k$. The LP can be written as

$$
\begin{array}{lrcll}
\max & \sum_{i=1}^{n}\sum_{j=1}^{m} x_{ij} & & & \\
\text{s.t.} & \sum_{j=1}^{m} x_{ij} & \leqslant & 1 & \text{for all } i = 1,2,\cdots,n \\
 & F_i x_{ij} + F_i x_{kj} & \leqslant & F_i + S_k & 1 \leqslant i < k \leqslant n, j = 1,\cdots,m \\
 & x_{ij} & \in & \{0,1\} & i = 1,2,\cdots,n; j = 1,2,\cdots,m
\end{array}
$$

- METHOD 5 Construct a directed graph whose nodes are courses and add the arc $i \to j$ if $F_i \leqslant S_j$, then add a staring node and an ending node, which connect to all nodes in the original graph with an directed arc. Then we can use a method like network flow and give an LP. We use y_{ij} to denote the network flow on arc $i \to j$, then the LP is

$$
\begin{array}{llcll}
\max & \sum_{i=1}^{n} x_i & & & \\
\text{s.t.} & y_{it} + \sum_{j=1}^{n} y_{ij} & = & y_{si} + \sum_{j=1}^{n} y_{ji} & i = 1, \cdots, n \\
 & y_{it} + \sum_{j=1}^{n} y_{ij} & \leqslant & x_i & i = 1, \cdots, n \\
 & \sum_{i=1}^{n} y_{si} & \leqslant & m & \\
 & y_{ij} & \in & \{0,1\} & i = s, 1, 2, \cdots, n; j = 1, 2, \cdots, n, t \\
 & x_i & \in \{0,1\} & i = 1, 2, \cdots, n &
\end{array}
$$

2. For the above two ILPs, we cannot prove whether solution of the relaxed LP contain only integers directly from the lemma. So I don't know the answer of this question.

4 Gas Station Placement

Similar to the problem two.

5 Stable Matching Problem

Suppose $x_{ij} = \begin{cases} 1, & \text{man } i \text{ and woman } j \text{ get married} \\ 0, & \text{otherwise} \end{cases} \quad (i, j = 1, 2, \cdots, n).$

1. The ILP is as follows:

$$
\begin{array}{llcll}
\min & 0 & & & \\
\text{s.t.} & \sum_{i=1}^{n} x_{ij} & = & 1 & \text{for all } j = 1, 2, \cdots, n \\
 & \sum_{j=1}^{n} x_{ij} & = & 1 & \text{for all } i = 1, 2, \cdots, n \\
 & x_{ij} + x_{kl} & \leqslant & S_{i,j,k,l} + 1 & \text{for all } i, j, k, l = 1, 2, \cdots, n, i \neq k, j \neq l \\
 & x_{ij} & \in & \{0,1\} & \text{for all } i, j = 1, 2, \cdots, n
\end{array}
$$

The third constraint can be replaced by $x_{ij} + (1 - S_{i,j,k,l})x_{kl} \leqslant 1$.

2. If m_l likes w_k more than w_j, and w_k likes m_l more than m_i, then m_i and w_j will never become the wrecker if m_l and w_k get married (but we are not sure whether m_l and w_k will get married since other strong wreckers might exist). In other words, if $p_{l,k,j} = 1$ and $q_{k,l,i} = 1$, then $x_{ik} = 0$ and $x_{lj} = 0$. Based on this, the ILP is as follows:

$$
\begin{array}{llcll}
\min & 0 & & & \\
\text{s.t.} & \sum_{i=1}^{n} x_{ij} & = & 1 & \text{for all } j = 1, 2, \cdots, n \\
 & \sum_{j=1}^{n} x_{ij} & = & 1 & \text{for all } i = 1, 2, \cdots, n \\
 & x_{ik} + x_{lj} & \leqslant & 3 - p_{l,k,j} - q_{k,l,i} & \text{for all } i, j, k, l = 1, 2, \cdots, n, k \neq j, l \neq i \\
 & x_{ij} & \in & \{0,1\} & \text{for all } i, j = 1, 2, \cdots, n
\end{array}
$$

The third and fourth constraints can be replaced by $x_{ij} + x_{kl} \leqslant 2 - p_{ilj} q_{jki}$.

6 Duality

For simplicity, we can assume that (u, v) denotes the arc $u \to v$. Then the primal can be rewritten and corrected as:

$$
\begin{array}{lrcll}
\max / \min & & & 0 & \\
\text{s.t.} & \sum_{i=1}^{k} f_i(u,v) & \leqslant & c(u,v) & \text{for each } (u,v) \\
& \sum_{v,(u,v)\in E} f_i(u,v) - \sum_{v,(v,u)\in E} f_i(v,u) & = & 0 & \text{for each } i \text{ and } u \in V \setminus \{s_i, t_i\} \\
& \sum_{v,(s_i,v)\in E} f_i(s_i,v) - \sum_{v,(v,s_i)\in E} f_i(v,s_i) & = & d_i & \text{for each } i \\
& f_i(u,v) & \geqslant & 0 & \text{for each } i, (u,v)
\end{array}
$$

If we use x_{uv} to denote the first constraints, y_{iu} the second and third constraints, then the duality is:

$$
\begin{array}{lrcll}
\min & c(u,v)x_{uv} + d_i y_{is_i} & & & \\
\text{s.t.} & x_{uv} + y_{iu} - y_{iv} & \geqslant & 0 & \text{for all } i \text{ and } u \neq t_i, v \neq t_i \\
& x_{ut_i} + y_{iu} & \geqslant & 0 & \text{for all } i, (u, t_i) \\
& x_{t_i v} - y_{iv} & \geqslant & 0 & \text{for all } i, (t_i, v) \\
& x_{uv} & \geqslant & 0 & \text{for all } (u,v)
\end{array}
$$

or

$$
\begin{array}{lrcll}
\max & c(u,v)x_{uv} + d_i y_{is_i} & & & \\
\text{s.t.} & x_{uv} + y_{iu} - y_{iv} & \leqslant & 0 & \text{for all } i \text{ and } u \neq t_i, v \neq t_i \\
& x_{ut_i} + y_{iu} & \leqslant & 0 & \text{for all } i, (u, t_i) \\
& x_{t_i v} - y_{iv} & \leqslant & 0 & \text{for all } i, (t_i, v) \\
& x_{uv} & \leqslant & 0 & \text{for all } (u,v)
\end{array}
$$