Assignment 4

Algorithm Design and Analysis

Notice:

1. **Due** Nov. 11, 2016 for graduate students at UCAS.
2. Please submit your answers in hard copy, **AND** graduate stuents at UCAS should also submit a digital version to UCAS website https://sep.ucas.ac.cn/.
3. Please choose at least two problems from problems 1-6, and choose at least one problem from problems 7-8.
4. Note that problem 1.2 and problem 6 require knowledge on duality.
5. Integer Linear Programming is different from the classic Linear Programming that some extra constraints such as

$$x_i \ is \ an \ integer, \ for \ all \ i = 1, 2, \cdots, n$$

or

$$x_i \in \{0, 1\}, \ for \ all \ i = 1, 2, \cdots, n$$

are added.

6. When you give the formulation of an LP or ILP, you should explain all mathematical symbols you are using if not appearing in the problem, and interpret the constraints if necessary.

1 Linear-inequality feasibility

Given a set of m linear inequalities on n variables $x_1, x_2, \cdots, x_n$, the **linear-inequality feasibility problem** asks if there is a setting of the variables that simultaneously satisfies each of the inequalities.

1. Show that if we have an algorithm for linear programming, we can use it to solve the linear-inequality feasibility problem. The number of variables and constraints that you use in the linear-programming problem should be polynomial in n and m.
2. Show that if we have an algorithm for the linear-inequality feasibility problem, we can use it to solve a linear-programming problem. The number of variables and linear inequalities that you use in the linear-inequality feasibility problem should be polynomial in n and m, the number of variables and constraints in the linear programming.

2 Airplane Landing Problem

With human lives at stake, an air traffic controller has to schedule the airplanes that are landing at an airport in order to avoid airplane collision. Each airplane i has a time window $[s_i, t_i]$ during which it can safely land. You must compute the exact time of landing for each airplane that respects these time windows. Furthermore, the airplane landings should be stretched out as much as possible so that the minimum time gap between successive landings is as large as possible.

For example, if the time window of landing three airplanes are [10:00-11:00], [11:20-11:40], [12:00-12:20], and they land at 10:00, 11:20, 12:20 respectively, then the smallest gap is 60 minutes, which occurs between the last two airplanes.

Given n time windows, denoted as $[s_1, t_1], [s_2, t_2], \cdots, [s_n, t_n]$ satisfying $s_1 < t_1 < s_2 < t_2 < \cdots < s_n < t_n$, you are required to give the exact landing time of each airplane, in which the smallest gap between successive landings is maximized.

Please formulate this problem as an LP, construct an instance and use GLPK or Gurobi or other similar tools to solve it.

3 Interval Scheduling Problem

A teaching building has m classrooms in total, and n courses are trying to use them. Each course i $(i = 1, 2, \cdots, n)$ only uses one classroom during time interval $[S_i, F_i)$ $(F_i > S_i > 0)$. Considering any two courses can not be carried on in a same classroom at any time, you have to select as many courses as possible and arrange them without any time collision. For simplicity, suppose $2n$ elements in the set $\{S_1, F_1, \cdots, S_n, F_n\}$ are all different.

1. Please use ILP to solve this problem, then construct an instance and use GLPK or Gurobi or other similar tools to solve it.

2. If you relax the integral constraints and change ILP to an LP (e.g. change $x \in \{0, 1\}$ to $0 \leqslant x \leqslant 1$), will solution of the LP contains only integers, regardless of values of all S_i and F_i? If it's true, prove it; if it's false, give a counter example. You can use the following lemma for help.

   Lemma If matrix A has only 0, +1 or -1 entries, and each column of A has at most one +1 entry and at most one -1 entry. In addtion, the vector b has only integral entries. Then the vertex of polytope $\{x|Ax \leqslant b, x \geqslant 0\}$ contains only integral entries.

4 Gas Station Placement

Let's consider a long, quiet country road with towns scattered very sparsely along it. Sinopec, largest oil refiner in China, wants to place gas stations along the road. Each gas station is assigned to a nearby town, and the distance between any two gas stations being as small as possible. Suppose there are n towns with distances from one endpoint of the road being $d_1, d_2, \cdots, d_n$. n gas stations are to be placed along the road, one station for one town. Besides, each station is at most r far away from its correspond town. $d_1, \cdots, d_n$ and r have been given and satisfied $d_1 < d_2 < \cdots < d_n$, $0 < r < d_1$ and $d_i + r < d_{i+1} - r$ for all i. The objective is to find the optimal placement such that the maximal distance between two successive gas stations is minimized.

Please formulate this problem as an LP.

5 Stable Matching Problem

n men $(m_1, m_2, \cdots, m_n)$ and n women $(w_1, w_2, \cdots, w_n)$, where each person has ranked all members of the opposite gender, have to make pairs. You need to give a stable matching of the men and women such that there is no unstable pair. Please choose one of the two following known conditions, formulate the problem as an ILP (*hint*: Problem 1.1 in this assignment), construct an instance and use GLPK or Gurobi or other similar tools to solve it.

1. You have known that for every two possible pairs (man m_i and woman w_j, man m_k and woman w_l), whether they are stable or not. If they are stable, then $S_{i,j,k,l} = 1$; if not, $S_{i,j,k,l} = 0$. $(i, j, k, l \in \{1, 2, \cdots, n\})$
2. You have known that for every man m_i, whether m_i likes woman w_j more than w_k. If he does, then $p_{i,j,k} = 1$; if not, $p_{i,j,k} = 0$. Similarly, if woman w_i likes man m_j more than m_k, then $q_{i,j,k} = 1$, else $q_{i,j,k} = 0$. $(i, j, k \in \{1, 2, \cdots, n\})$

6 Duality

Please write the dual problem of the MultiCommodityFlow problem in *Lec8.pdf*, and give an explanation of the dual variables.

Please also construct an instance, and try to solve both primal and dual problem using GLPK or Gurobi or other similar tools.

7 Simplex Algorithm

Please implement simplex algorithm or dual simplex algorithm with your favorate language, and make comparison with GLPK or Gurobi or other similar tools.

8 LP And Iterative Algorithm

There are n real numbers $x_1, x_2, \cdots, x_n$, and $x_1 = 0$. We want to choose $n-1$ values for $x_2, \cdots, x_n$ such that $m = \Theta(n)$ ($\lim_{n\to\infty} \frac{\Theta(n)}{n}$ is a constant) constraints, each looking like $x_i - x_j = d_{ij}$ $(i \neq j)$, can be satisfied. Each x_i and x_j pair ($i, j \in \{1, 2, \cdots, n\}$ and $i \neq j$) will appear in the same constraint at most only once. However, we cannot satisfy all these m constraints simultaneously with high probability. For example, the constraints

$$x_i - x_j = 1$$
$$x_i - x_k = 1$$
$$x_j - x_k = 1$$

cannot be satisfied simultaneously. As a result, we have to relax the limitations to $|x_i - x_j - d_{ij}| \leqslant \varepsilon_{ij}$ ($\varepsilon_{ij} \geqslant 0$) and try to minimize $\sum \varepsilon_{ij}$.

1. Formulate this problem as an LP.

2. Another algorithm to solve this problem is shown below. Implement this algorithm with your favorate language.

Algorithm 1 Problem 8

Construct an undirected graph $G = (V, E)$ with n isolated vertices $v_1, \cdots, v_n$;
for each constraint $x_i - x_j = d_{ij}$ **do**
    Connect v_i and v_j with an edge e_{ij};
Set $x_1^{(0)} = 0$;
BFS the graph from v_1, and set initial values of other vertices according to constraints on edges (e.g. when you are visiting unvisited vertex v_j from v_i along the edge e_{ij} with constraint $x_i - x_j = d_{ij}$, set $x_j^{(0)} = x_i^{(0)} - d_{ij}$);
Set $t = 0$;
repeat
    for $i = 1 \to n$ **do**
        Clear set S;
        for each vertex x_j which is connected to x_i by an edge with constraint $x_i - x_j = d_{ij}$ **do**
            Put $x_i^{(t+1)} = x_j^{(t)} + d_{ij}$ to set S;
        Sort elements in set S and set $\lfloor \frac{|S|+1}{2} \rfloor$-th smallest element as $x_i^{(t+1)}$;
    $t++$;
until convergence;
Output values of $0, x_2^{(t+1)} - x_1^{(t+1)}, \cdots, x_n^{(t+1)} - x_1^{(t+1)}$ as $x_1, x_2, \cdots, x_n$.

3. Construct an instance, then use GLPK or Gurobi or other similar tools to solve your LP, use the algorithm you implement previously to solve the problem, and compare the results and running time.