ChE 597 Computational Optimization

Homework 4

Feb 11th 11:59 pm

1. Consider the following linear programming problem:

$$\text{minimize} \quad x_1 - x_2$$

Subject to the constraints:

$$\begin{aligned}
&2x_1 + 3x_2 - x_3 + x_4 \leq 0 \\
&3x_1 + x_2 + 4x_3 - 2x_4 \geq 3 \\
&-x_1 - x_2 + 2x_3 + x_4 = 6 \\
&x_1 \leq 0 \\
&x_2,\ x_3 \geq 0
\end{aligned}$$

Write down the corresponding dual problem.

2. In this question, we will show the ***max-flow*** problem and the ***min-cut*** problem are dual to each other. First, let's formulate both problems to facilitate solving a numerical problem in Pyomo.

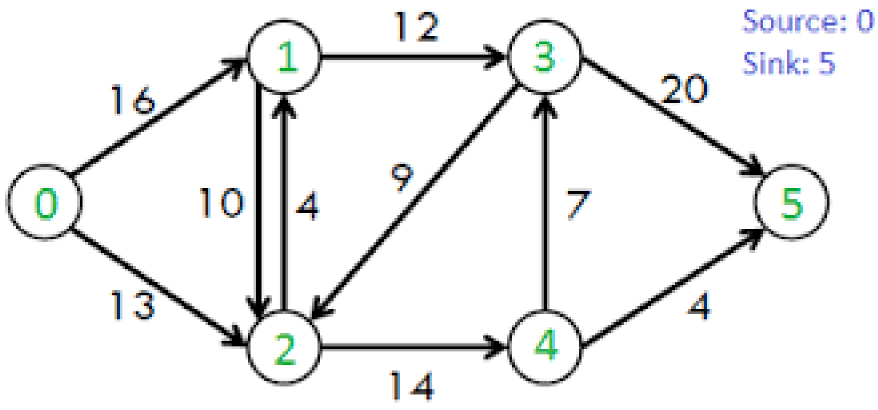

Figure 1: Directed graph (Network flow)

(a) **Formulating the *max-flow problem* as a LP problem:** Consider a directed graph $G = (V,E)$, where V denotes a set of vertices and $E \subseteq V \times V$ denotes a set of edges. Let $e = (u,v) \in E$ be an edge for vertex u to vertex v, we define $e \in \delta^+(v)$ to be the set of all outgoing edges to vertex v and $e \in \delta^-(v)$ to be the set of all incoming nodes from vertex v. Let $c(e)$, $e \in E$ be the capacity of the edge, i.e., the maximum amount of commodity that one can push through the edge. Let $f(e)$ be the flow across an edge e. $f(e)$ is non-negative as the flow in the edge cannot flow in the reverse direction i.e., towards the source. Also, the flow across each node should be conserved. In other words, the flow coming into a node has to be equal to the flow leaving the node. This conservation property does not apply to the source node (s) and the sink node (t). Formulating a linear programming problem to maximize the overall flow departing from the source node leads us to the formulation provided below:

$$
\begin{aligned}
\max \quad & \textstyle\sum_{e\in\delta^+(s)} f(e) - \sum_{e\in\delta^-(s)} f(e) \\
\text{subject to} \quad & \textstyle\sum_{e\in\delta^-(v)} f(e) = \sum_{e\in\delta^+(v)} f(e) \quad \forall v \in V \setminus \{s,t\} \qquad \text{(flow conservation)} \\
& f(e) \le c(e) \quad \forall e \in E \qquad \text{(capacity constraints)} \\
& f(e) \ge 0 \quad \forall e \in E
\end{aligned}
$$

(b) **Formulating the *min-cut problem* as an LP problem:** The minimum cut problem requires identifying the minimal set of edges that, when removed, completely separates the source and sink vertices, creating two non-overlapping partitions in the graph. In other words, given any subset S of nodes with source node $s \in S$, let T be the set of the remaining nodes. The cut (S, T) is the set of edges $e = (u,v)$ with $u \in S$ and $v \in T$. It is so-called because removing all those edges in the cut would cut the flow from s to t. Let c(e), $e \in E$ be the capacity of the edge, i.e., the maximum amount of commodity that one can push through the edge. Let x_k, $\forall\ k \in V$ be a variable corresponding to the vertices and let y_e, $\forall\ e = (u,v) \in E$ be a variable corresponding to the edges. The min-cut problem can be formulated as the following:

$$
\begin{array}{lll}
\min & \sum_{e\in E} c(e) y_e & \\
\text{s.t.} & x_u - x_v + y_e \geq 0 & \forall e = (u,v) \in E \\
& -x_s + x_t \geq 1 & \\
& x_v \in \mathbb{R} & \forall v \in V \\
& y_e \geq 0 & \forall e \in E
\end{array}
$$

(a) For the directed graph in Figure 1, solve the max-flow problem using Pyomo.

(b) For the directed graph in Figure 1, solve the min-cut problem using Pyomo. Check if the problem have the same objective as (a).

(c) Show that the max-flow problem and the min-cut problem are duals of each other.

Hint: We can write an equivalent formulation for the maximum flow problem to facilitate the ease of obtaining its dual by introducing a new variable f^* and following the steps below:

(i). Replacing f^* with "$\sum_{e\in\delta^+(s)} f(e) - \sum_{e\in\delta^-(s)} f(e)$" in the objective function.

(ii). Adding constraints to compensate for the objective coefficient replacement.

$$\sum_{e\in\delta^+(s)} f(e) - \sum_{e\in\delta^-(s)} f(e) - f^* = 0$$

$$\sum_{e\in\delta^+(t)} f(e) - \sum_{e\in\delta^-(t)} f(e) + f^* = 0$$

The initial constraint involves setting f^* equal to the objective function in our initial max-flow LP formulation. The subsequent constraint pertains to the sink node. Given that the flow originating from the source node must traverse the sink node, we establish an equivalence between f^* and the flow entering the sink node.

(iii). Finally, the equivalent formulation for the max-flow problem is as follows:

$$
\begin{array}{lll}
\max & f^* & \\
\text{s.t.} & \sum_{e\in\delta^+(s)} f(e) - \sum_{e\in\delta^-(s)} f(e) - f^* = 0 & \\
& \sum_{e\in\delta^+(v)} f(e) - \sum_{e\in\delta^-(v)} f(e) = 0 & \forall v \in V \backslash \{s,t\} \\
& \sum_{e\in\delta^+(t)} f(e) - \sum_{e\in\delta^-(t)} f(e) + f^* = 0 & \\
& f(e) \leq c(e) & \forall e \in E \\
& f(e) \geq 0 & \forall e \in E \\
& f^* \geq 0 &
\end{array}
$$

Now, derive the dual for the above max-flow problem formulation and show that it is equivalent to the provided minimum cut problem formulation in the question.

3. Interdiction game

Let us consider an interesting scenario. Suppose that terrorists may attack any of three sites, and the police can only patrol one site each day. To keep the terrorists guessing, the police will randomly choose a site to patrol each day. Similarly, to keep the police guessing, the terrorists will randomly choose a site to attack each day. Now, the question is with what probability should the police and terrorists choose each site to maximize their expected utility?

Let the gain to police entails an equal loss to the terrorists (zero-sum game). The table given below indicates the gain to the police for each possible outcome.

		police		
		site 1	site 2	site 3
terrorists	site 1	4	−10	−10
	site 2	−8	5	−8
	site 3	−12	−12	9

For example, if police patrol site 2 but terrorists hit site 1, the police lose 10 and the terrorists gain 10.

Let the probability of police choosing site j be x_j and terrorists choosing site i is u_i. Let the table above be the payoff (utility) matrix A. The police assume that whatever probabilities they choose, the terrorists will choose an optimal strategy based on those probabilities. Therefore, they wish to find x^* by solving the following problem:

$$\max_{\substack{x\geq 0\\ e^T x=1}} \left\{ \min_{\substack{u\geq 0\\ e^T u=1}} \left\{ u^T A x \right\} \right\}$$

The constraint $e^T x = 1$, where e is a vector of ones, ensures that the probabilities sum to 1 .

The min problem is trivial to solve, since Ax is a fixed vector. If component i of Ax is the smallest component, let $u_i = 1$ and all other u_k 's vanish. So it can be written as

$$\max_{\substack{x\geq 0\\ e^T x=1}} \left\{ \min_{i} \left\{ e_i^T A x \right\} \right\}$$

where e_i is the i th unit vector (1 in position $i, 0$ in all other positions). This problem can be written

$$\begin{array}{ll} \max & z \\ \text{s.t.} & z \leq e_i^T A x, \text{ all } i \\ & e^T x = 1 \\ & x \geq 0 \end{array}$$

where z is a scalar.

or equivalently

$$\begin{array}{ll} \max_{z,x} & z \\ \text{s.t.} & ze - Ax \leq 0 \\ & e^T x = 1 \\ & x \geq 0 \end{array}$$

Where e is a vector of ones to ensure that the probabilities sum to one.

Similarly, the terrorists would try to find u^* by solving the following problem:

$$\min_{\substack{u \geq 0 \\ e^T u = 1}} \left\{ \max_{\substack{x \geq 0 \\ e^T x = 1}} \left\{ u^T A x \right\} \right\}$$

which can be rewritten as

$$\begin{array}{ll} \min_{w,u} & w \\ \text{s.t.} & w e^T - u^T A \geq 0 \\ & u^T e = 1 \\ & u \geq 0 \end{array}$$

where w is a scalar.

(a) Calculate the optimal solution for the police problem using Pyomo.

(b) Calculate the optimal solution for the terrorist problem using Pyomo. Check if the problem have the same objective as (a).

(c) Show that the police problem and the terrorist problem are duals of each other.

Hint: it suffices to show that the dual of the terrorist problem is the police problem. Alternatively, you can also show that the dual of the police problem is the terrorist problem.

4. Let A be a symmetric square matrix. Consider the linear programming problem

$$\begin{aligned} \min \quad & c^T x \\ \text{s.t. } & Ax \geq c \\ & x \geq 0. \end{aligned}$$

Prove that if x^* satisfies $Ax^* = c$ and $x^* \geq 0$, then x^* is an optimal solution.

5. Solve the following SDP problem using the cutting plane algorithm in python.

$$\max_{X \in \mathcal{S}^4} < Q, X >$$
$$\operatorname{diag}(X) = e$$
$$X \succeq 0$$

where $Q = \begin{bmatrix} 0 & 1 & 3 & 1 \\ 1 & 0 & 0 & 2 \\ 3 & 0 & 0 & 4 \\ 1 & 2 & 4 & 0 \end{bmatrix}$

$\operatorname{diag}(X) = e$ means the diagonal entries of X equal to 1.

Hint: the quadratic constrained program in the cutting plane algorithm can be solved using Gurobi. The eigenvalue decomposition can be done using packages like numpy. The algorithm can terminate when the smallest eigenvalue of X is greater than -10^{-4}