CMPUT 675 - Assignment #4
Due Nov 14 by 1pm

Fall 2016, University of Alberta

Pages: 3

This one is a bit shorter. Enjoy your reading week :)

Exercise 1)

Marks: 3

We proved the following in the lectures. Let $\mathbf{A} \in \mathbb{Z}^{m \times n}$, $\mathbf{b} \in \mathbb{Z}^m$ and $\Gamma \geq 0$. Consider the polytope

$$\mathcal{P} = \{\mathbf{x} \in \mathbb{R}^n : \mathbf{A} \cdot \mathbf{x} \leq \mathbf{b} \text{ and } 0 \leq \mathbf{x}_j \leq \Gamma \text{ for each } 1 \leq j \leq n\}.$$

Then we can decide if $\mathcal{P} = \emptyset$ in time that is polynomial in n, m and the bit complexity of both Γ and the values in $\mathbf{A}, \mathbf{b}$. We used this to find an optimal solution to a linear program with a finite optimum in polynomial time (in the size of the input of the linear program).

Your job is to carefully describe how to determine if a linear program is unbounded (i.e. has feasible solutions of arbitrarily large size) in polynomial time, thus completing the entire algorithm for solving linear programs in polynomial time. You may use the above statement as a black box.

Do not worry about making the running time "independent" of m via separation oracles. You can rely on m in your final running time (though it is avoidable).

Exercise 2)

Marks: 5

Let $\mathbf{A} \in \mathbb{R}^{m \times n}, \mathbf{b} \in \mathbb{R}^m$.

- Show that in any extreme point $\overline{\mathbf{y}}$ of the polyhedron $\{\mathbf{y} \in \mathbb{R}^m_{\geq 0} : \mathbf{A}^T \cdot \mathbf{y} \geq \mathbf{0}, \mathbf{b}^T \cdot \mathbf{y} = -1\}$ we have that $\overline{\mathbf{y}}_i \neq 0$ for at most $n+1$ indices $1 \leq i \leq m$.
  Hint: look at the matrix of tight constraints.
- Show that if $\mathcal{P} = \emptyset$ where $\mathcal{P} = \{\mathbf{x} \in \mathbb{R}^n_{\geq 0} : \mathbf{A} \cdot \mathbf{x} \leq \mathbf{b}\}$ then in fact there is a collection of $m' \leq n+1$ constraints $\mathbf{A}' \cdot \mathbf{x} \leq \mathbf{b}'$ from the system $\mathbf{A} \cdot \mathbf{x} \leq \mathbf{b}$ such that $\mathcal{Q} = \emptyset$ where $\mathcal{Q} = \{\mathbf{x} \in \mathbb{R}^n_{\geq 0} : \mathbf{A}' \cdot \mathbf{x} \leq \mathbf{b}'\}$.

Ultimately, this means in the step where we slightly relax the constraints by some tiny $\epsilon > 0$ when solving for feasibility via the Ellipsoid method, it suffices to let ϵ depend on the maximum bit complexity of coefficients in the constraints and n (but not on m).

Exercise 3)

Marks: 6

Let $\mathcal{G} = (V; E)$ be a directed graph with edge costs $c : E \to \mathbb{R}_{\geq 0}$ and let $s, t \in V$ be distinct vertices. In the SHORTEST PATH problem, we have to find an $s - t$ path $P \subseteq E$ with minimum cost $c(P)$.

The following LP provides a lower bound on the optimum solution cost.

$$\begin{aligned} \text{minimize :} \quad & \sum_{e \in E} c_e \cdot \mathbf{x}_e \\ \text{subject to :} \quad & \mathbf{x}(\delta^{out}(S)) \;\geq\; 1 \qquad \text{for all } \{s\} \subseteq S \subseteq V - \{t\} \\ & \mathbf{x} \;\geq\; 0. \end{aligned}$$

Your goal is to show the optimum solution value of this LP in fact equals the cheapest $s - t$ path cost by using a primal-dual algorithm.

- Write the dual of this LP.
- Design a primal-dual algorithm for solving the shortest path problem based on this LP.

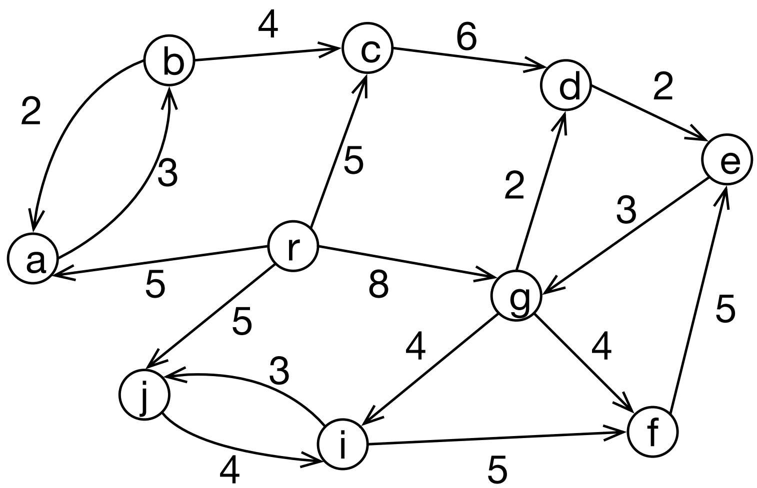
  Hint:

  - Initialize $S \leftarrow \{s\}$. Raise the dual variable for S. Expand S as dual constraints go tight.
  - Once S contains t, discard edges that were "purchased" but not needed in the $s-t$ path.
  - Use complementary slackness.

  Another Hint: One can view Dijkstra's algorithm for finding a minimum-cost $s - t$ path as a primal-dual procedure.

- Finally, write a linear program with polynomially many constraints and variables (polynomial in the size of $\mathcal{G}$) whose optimum solution value equals the cheapest $s - t$ path cost.

  You can use any approach you like (i.e. you may or may not want to stick to ideas from the above LP, any approach will do, I can think of approaches either way). Prove why the optimum value of your linear program equals the cheapest $s - t$ path cost.



Exercise 4)

Marks: 4

Compute a minimum-cost arborescence (rooted at r) in the above directed graph. In each iteration:

- Describe the set of edges F and the cycles of F.
- Describe the new graph where each vertex is labelled by the set of original nodes that were contracted to that vertex.
- Describe the new costs c' of the remaining edges after contracting the cycles.

If there is ever a tie for the cheapest edge entering a vertex, break ties however you want.

Once you finally reach an iteration where F is acyclic, step back through the graphs that you contracted to show the arborescences you buy in each such graph.

Also give a dual solution $\mathbf{y}$ witnessing the fact that your arborescence is a minimum-cost arborescence. You do **not** have to track the construction of $\mathbf{y}$ throughout the entire algorithm (though it might be helpful). Of course, you can just give the values for $\mathbf{y}_S$ that are nonzero.