Strongly polynomial algorithms and generalized flows
Problem set 2

Summer School on Combinatorial Optimization
Hausdorff Center for Mathematics
August 2018

Exercise 2.1 Show that the Edmonds-Karp-Dinits algorithm for maximum flow can be interpreted as a special case of the Goldberg-Tarjan algorithm for minimum-cost flow. That is, given a max flow instance, modify the graph to a min-cost flow instance where running the GT algorithm corresponds to the EKD algorithm in the original graph.

Exercise 2.2 Prove Lemma 2.2 in the lecture notes.

Exercise 2.3 Consider a minimum-cost flow instance with n nodes and m arcs in the form

$$\begin{aligned} \min\ & c^\top f \\ \nabla f_i &= b_i \quad \forall i \in V \\ f &\geq 0 \end{aligned} \tag{1}$$

Let

$$\mathcal{L} = \{x \in \mathbb{R}^m : \nabla x_i = 0 \quad \forall i \in V\}$$

denote the set of circulations in the graph. Note that $\mathcal{L}$ is a linear subspace in $\mathbb{R}^m$; we let $\mathcal{L}^\perp$ denote the orthogonal complement of $\mathcal{L}$.

(a) Given an arbitrary potential $y : V \to \mathbb{R}$, we can define the cost function $d : E \to \mathbb{R}$ by $d_{ij} := y_i - y_j$. Show that $d \in \mathcal{L}^\perp$, and conversely, that every cost function in $\mathcal{L}^\top$ can be written in such a form.

(b) Show that if $c \in \mathcal{L}^\top$, then every feasible solution to (1) is optimal.

(c) Given a cost function $c \in \mathcal{L}$, recall the notation $\|c\|_\infty = \max_{e \in E} |c_e|$. For an arbitrary $\pi : V \to \mathbb{R}$, show that for the modified cost function c^π,
$$\|c^\pi\|_\infty \geq \|c\|_\infty / m^2.$$

(d) You are now ready to read
É. Tardos. A strongly polynomial minimum cost circulation algorithm. *Combinatorica*, 5(3):247–255, 1985.

Exercise 2.4 Show that if we have an algorithm for solving any LP *feasibility problem* in strongly polynomial time, then we can use this to solve any LP *optimization problem* in strongly polynomial time.

Exercise 2.5 Show that if we have an algorithm for solving an optimization LP in strongly polynomial time that has at most three nonzero entries in every column of the constraint matrix, then we can use this to solve any LP in strongly polynomial time.