Mathematics for Decision Making: An Introduction

Lecture 18

Matthias Köppe

UC Davis, Mathematics

March 5, 2009

Augmenting Circuit Algorithm for Min Cost Flow

Augmenting Circuit Algorithm, Kantoróvich [1942]

Input: Graph $G = (V, A)$, capacities $\mathbf{u}$, excess values $\mathbf{b}$, costs $\mathbf{c}$

- Find a feasible flow $\mathbf{x}$ (max-flow, homework)
- While there exists a negative-cost directed circuit in $G(\mathbf{x})$, i.e., an **augmenting circuit** for $\mathbf{x}$ in G:
  Determine the capacity (bottleneck) of the augmenting circuit.
  Augment $\mathbf{x}$ along C by this bottleneck.

- Negative-cost directed circuits can be determined (in polynomial time) by running the Bellman–Ford algorithm, or other shortest-path algorithms. (Key: cycles in the predecessor vector.)
- Again, as we see already on simple examples, this gives us (only) a **pseudo-polynomial algorithm** for instances where it terminates.
- Choosing “most negative” augmenting circuits does not work (neither effective nor efficient)
- Choosing minimum-mean-cost (i.e., most-negative-mean-cost) circuits produces a polynomial-time algorithm, Goldberg–Tarjan [1989]
- A strongly polynomial algorithm for min-cost flow was unknown until 1985!

A dual criterion (certificate) of optimality

- What we have found is a **primal algorithm**, together with a “primal” criterion of optimality (non-existence of an augmenting circuit).
- This is in some contrast to the earlier primal algorithms we discussed, where we were aware of a “dual” criterion of optimality (existence of a certain certificate of optimality).
- For our search for more efficient algorithms, let’s try to find this missing **duality theory** first.
- Interpreting the non-existence of negative-cost directed circuits in terms of shortest path theory yields:

Theorem (Optimality Certificate Theorem)

A feasible flow $\mathbf{x}^1$ *has minimal cost if and only if there exists a* ***potential vector*** $\mathbf{y} = (y_v)_{v \in V}$ *such that for all arcs* $(v, w) \in A$*:*

$$\bar{c}_{v,w} < 0 \quad \textit{implies} \quad x_{v,w} = u_{v,w}$$

$$\bar{c}_{v,w} > 0 \quad \textit{implies} \quad x_{v,w} = 0,$$

where the ***reduced*** *costs* $\bar{c}_{v,w}$ *are defined as* $\bar{c}_{v,w} = c_{v,w} + y_v - y_w$.

A Dual Algorithm for Min-Cost Flow

- **Here's a new idea:**
  - In primal algorithms, we start with an initial feasible solution and improve it, step by step, until the (dual) optimality criterion holds.
  - Let's try instead a **dual algorithm**, where we start with a "solution" for which the (dual) **optimality criterion holds**, but which is **not feasible**; and improve it, step by step, until it becomes feasible.
- Because the (dual) optimality criterion is about the existence of a certificate, we also **maintain this certificate** during the course of the algorithm.
- Concretely, for min-cost flow:
  - Keep a **flow** $\mathbf{x} = (x_{uv})_{(u,v)\in A}$ that **satisfies the bounds** $\mathbf{0} \leq \mathbf{x} \leq \mathbf{u}$ but is allowed to **violate the flow excess conditions**;
  - keep a **potential** $\mathbf{y} = (y_v)_{v\in V}$;
  - ...such that the conditions of the Optimality Certificate Theorem hold:

$$x_{v,w} = u_{v,w} \qquad \text{for all arcs } (v,w) \in A \text{ with } \bar{c}_{v,w} < 0$$

$$x_{v,w} = 0 \qquad \text{for all arcs } (v,w) \in A \text{ with } \bar{c}_{v,w} > 0$$

- Very easy to construct an **initial pair of solutions** if all costs are non-negative: Just use $\mathbf{x} = \mathbf{0}$, $\mathbf{y} = \mathbf{0}$.
- We'll discuss the general construction later.

The Primal-Dual Algorithm

- Because both primal (flow) and dual (potential) information is maintained, we call this the primal-dual algorithm.
- The **improvement steps** of the algorithm need to push the flow towards feasibility; i.e., we wish to correct the flow balance for all vertices where $f_{\mathbf{x}}(v) \neq b_v$.
  - We call $v \in V$ an **x-source** if $f_{\mathbf{x}}(v) > b_v$.
  - We call $v \in V$ an **x-sink** if $f_{\mathbf{x}}(v) < b_v$.

  We will correct the flow balance by **sending flow from an x-source to an x-sink**.
- Again, we will be using an **x-augmenting path** (corresponding to a directed path in the auxiliary network).
- But we need to be careful to keep the optimality conditions satisfied!

Primal-Dual Algorithm

Input: Graph $G = (V, A)$, capacities $\mathbf{u}$, excess values $\mathbf{b}$, costs $\mathbf{c}$

- Construct a pair of initia solutions $\mathbf{x}$, $\mathbf{y}$.
- While $\mathbf{x}$ is not feasible:

  If there exists an $\mathbf{x}$-augmenting path P of equality arcs:

    Augment the flow $\mathbf{x}$ along P

  Otherwise:

    Find a vertex set R blocking all such paths, and change $\mathbf{y}$ at R.