Mathematics for Decision Making: An Introduction

Lecture 15

Matthias Köppe

UC Davis, Mathematics

February 24, 2009

The Ford–Fulkerson Algorithm

Ford–Fulkerson Maximum Flow Algorithm

Input: A digraph $G = (V, A)$ with arc capacities $\mathbf{u}$, vertices r and s.
Output: A maximum flow $\mathbf{x}$ and a set $R \subseteq V$ inducing a minimum cut $\delta(R)$.

- Set $\mathbf{x} := \mathbf{0}$.
- While we find a directed r-s path P in the auxiliary graph $G(\mathbf{x})$:

  Determine the **x-width** of P:

  $$\varepsilon := \min\left\{ \min\{\, u_{a,b} - x_{a,b} : (a,b) \text{ forward in } P \,\}, \min\{\, x_{a,b} : (a,b) \text{ reverse in } P \,\} \right\}$$

  Augment $\mathbf{x}$ along P by ε.
- Set R to the set of vertices that can be reached by paths from r in $G(\mathbf{x})$.

The Ford–Fulkerson Algorithm: Termination, Efficiency

Theorem (Termination of the Algorithm)

If **u** *is integral and there is a maximum flow (of value K), then the Ford–Fulkerson Maximum Flow Algorithm terminates after at most K augmentations.*

Proof.

Each of the augmentations increases the flow value by an integer amount. □

- This also establishes that the Ford–Fulkerson Algorithm is a **pseudo-polynomial algorithm** (for inputs with integer data that have a maximum flow).
  (By the Max-Flow Min-Cut Theorem, the flow value is the same as some cut capacity, so it is at most $\sum u_{ab}$, a quantity that is polynomial in the given data.)
- Examples that really take K augmentations (with a specific choice of a sequence of augmenting paths) can be easily constructed.
- Moreover, if there is no maximum flow, the procedure might fail to terminate.
- So, we are **not completely happy** with this basic algorithm.
- A scaling approach (with data $\lfloor \mathbf{u}/2^k \rfloor$, for k decreasing to 0) leads to a polynomial algorithm; we omit the details.
- Even better, it turns out that a **specific choice** of **x**-augmenting paths (which is currently unspecified) will lead to a strongly polynomial algorithm.

A Strongly Polynomial Time Variant

Theorem (Dinits [1970], Edmonds–Karp [1972])

If each augmentation is along a ***shortest*** *(i.e., minimum number of arcs)* ***augmenting path****, then the algorithm terminates after at most $nm = |V| \cdot |A|$ augmentations.*

- To prepare the proof, consider an augmentation along a (shortest) augmenting path $P = (v_0, \ldots, v_k)$ of length k, leading from flow $\mathbf{x}$ to flow $\mathbf{x}'$.
- Denote by $d_{\mathbf{x}}(v, w)$ the least number of arcs in a directed path from v to w in the auxiliary digraph $G(\mathbf{x})$; we set $d_{\mathbf{x}}(v, w) = +\infty$ if no such directed path exists.
- Since subpaths of shortest paths are shortest, we have $d_{\mathbf{x}}(r, v_i) = i$ and $d_{\mathbf{x}}(v_i, s) = k - i$.

A Strongly Polynomial Time Variant, II

Lemma

Shortest-augmenting-path augmentations ***never decrease the length of shortest directed paths in the auxiliary digraph*** *from the source r to any node v and from any node v to the sink s:*

$$d_{\mathbf{x}'}(r,v) \geq d_{\mathbf{x}}(r,v) \quad \textit{and} \quad d_{\mathbf{x}'}(v,s) \geq d_{\mathbf{x}}(v,s).$$

In particular, they never decrease the length of a shortest augmenting path:

$$d_{\mathbf{x}'}(r,s) \geq d_{\mathbf{x}}(r,s)$$

This lemma implies that shortest-augmenting-path augmentations proceed in **stages**, during which augmenting paths of **constant length** are used:

- Augmentations along paths of length 1 (possibly none)
- Augmentations along paths of length 2 (possibly none)

  ⋮

- Augmentations along paths of length $n-1$ (possibly none).

It now suffices to bound the number of augmentations of each stage in a strongly polynomial way.

A Strongly Polynomial Time Variant, III

Let $\tilde{A}(\mathbf{x})$ be the set of arcs $(a, b) \in A$ that appear in a shortest $\mathbf{x}$-augmenting path.

Lemma

If a shortest-augmenting-path augmentation does not increase the length of a shortest augmenting path, i.e., $d_{\mathbf{x}'}(r, s) = d_{\mathbf{x}}(r, s)$, then $\tilde{A}(\mathbf{x}')$ is a proper subset of $\tilde{A}(\mathbf{x})$.

Proof of the theorem.

From the second lemma, in each stage, there are at most $m = |A|$ augmentations per stage.
From the first lemma, there are at most $n - 1$ stages.
So, in total at most nm augmentations. □

An Application of Max-Flow Min-Cut: Bipartite Matching

- In the pen plotter problem, we came across a **matching problem**.
- As a reminder, a **matching** of an undirected graph $G = (V, E)$ is a set M of edges such that every vertex $v \in V$ is incident with at most one edge $e \in M$. In other words, the edges of a matching have no end in common.
- An important special case concerns **bipartite graphs** $G = (P \cup Q, E)$, i.e., graphs where every edge has its ends in different parts:

$$E \subseteq \left\{ \{p, q\} : p \in P, q \in Q \right\}.$$

- The **maximum bipartite matching problem** (or **marriage problem**) asks for a matching of maximum size in a given bipartite graph G.
- By introducing an artificial source r (with arcs of capacity 1 to all nodes in P) and a sink s (with arcs of capacity 1 from all nodes in P), and directing the edges to become arcs from $p \in P$ to $q \in Q$ (of capacity ∞), we can reduce the problem to a maximum flow problem.
- So the Ford–Fulkerson algorithm and the max-flow min-cut theorem immediately translate to results for the maximum bipartite matching problem.

An Application of Max-Flow Min-Cut: Bipartite Matching, II

In fact, the max-flow min-cut theorem translates to another classic result of combinatorial duality.

- A **cover** of a graph $G = (V, E)$ is a set $C \subseteq V$ of vertices such that every edge has at least one end in C.
- It is easy to see that matchings and covers are in weak duality:
  - Let $M \subseteq E$ be any matching, $C \subseteq V$ be any cover.
  - Then every edge $\{a, b\} \in M$ has at least one end in C (because C is a cover).
  - Because the edges of the matching M have no end in common,

$$|M| \leq |C|.$$

- But also strong duality holds:

Theorem (Kőnig's Theorem, 1931)

For a bipartite graph G,

$$\max\{\, |M| : M \textit{ is a matching} \,\} = \min\{\, |C| : C \textit{ is a cover} \,\}.$$

(This is false for non-bipartite graphs.)