Task Discussion
Algocoön group

Andreas Bärtschi

ETH Zürich

November 25, 2015

The problem

Find a minimum cut in directed graph G with n vertices and m edges.
No source or sink.

Precisely, compute:

$$\min_{\substack{S \\ 0<|S|<n}} \text{cap}(S, V \setminus S).$$

Approaches?

How to approach it?

1. Stoer-Wagner minimum cut.
   Undirected graphs only.
2. Heuristic: check all S with $|S| = 1$ and $|S| = n - 1$.
   Why should it work?
3. Brute force over all $S \in 2^V$.
   $\Omega(2^n)$ time, too slow even for the small testcase.
4. Maxflow – MinCut.
   No source and no sink.

Solution: Multiple maxflow calls with source/sink candidate pairs.

Solution: minimum cut via maxflow

Lemma

Fix a global minimum cut solution S.
Then for any $s \in S$ and $t \in V \setminus S$ we have the following:
- *Every s-t minimum cut is a global minimum cut.*

First solution:

- Compute u-v maxflows for all pairs $u, v \in V$. Take the smallest maxflow found.
- Correct: yes.
- Time: $O(n^2)$ flows. Solves the small testcase.

Solution: minimum cut via maxflow

Lemma

Fix a global minimum cut solution S.
Then for any $s \in S$ and $t \in V \setminus S$ we have the following:
Choose an arbitrary vertex $v_0 \in V$.
- *If $v_0 \in S$, then a v_0-t mincut is a global mincut.*
- *If $v_0 \notin S$, then a s-v_0 mincut is a global mincut.*

Second solution:

- Choose candidate pairs for u-v maxflow wisely.
  Take the smallest maxflow found among all pairs u-v.
- Correct: ?
- Time: $O(\#\text{ pairs})$ flows. Scores ? points.

Implementation

Call `push_relabel_max_flow()`. Retrieve G_f using `residual_capacity_map`. Compute S by breadth-first search.

You can call the maxflow solver consecutively on the same graph with different source-sink pairs.

```
push_relabel_max_flow(G, 0, 10);
push_relabel_max_flow(G, 1, 10);  // This is fine.
```

Make multiple calls to flow solver and remember which one had best value. Then call this "best" flow once more in the end to retrieve the residual capacity map (doesn't make much of a difference timewise). Compute the mincut.