Network flows - Fast(er) Algorithm

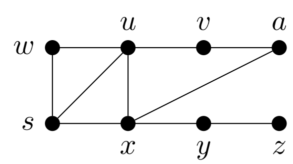
Edmonds-Karp Algorithm
Input: Network (G, u, s, t).
Output: an s-t-flow f of maximum value

1. $f(e) = 0$ for all $e \in E(G)$
2. while f-augmenting path in G_f exists:
3. find shortest f-augmenting path P in G_f
4. compute $\gamma := \min_{e \in E(P)} u_f(e)$
5. augment f along P by γ (as much as possible)

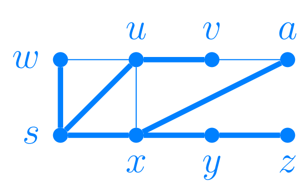
Note that the shortest path can be implemented by
BFS (Breath First Search) algorithm:
Input: Graph G, $s \in V(G)$.
Output: spanning tree T of shortest paths to s

1. $R = \{s\}$, $Q = (s)$, $T = (V, \emptyset)$.
2. while Q is not empty:
3. remove the first entry in Q, denote it by u.
4. $\forall uv \in E(G)$, if $v \notin R$
5. add v at the end of Q; add v to R; add uv to T

1: Try BFS algorithm on the following graph.



Solution: TODO: Table recording the run.



2: What is running time of BFS?

Solution: $O(m)$. Every edge is touched at most twice.

Lemma Let $f_1; f_2; \ldots$ be a sequence of flows such that f_{i+1} results from f_i by augmenting along P_i, where P_i is a shortest f_i-augmenting path. Then
(a) $|E(P_k)| \leq |E(P_{k+1})|$ for all k.
(b) $|E(P_k)| + 2 \leq |E(P_l)|$ for all $k < l$ such that $P_k \cup P_l$ contains a pair of reverse edges.

3: Prove (a). Hint: Consider edges X of P_k and P_{k+1} (with multiplicity) together (and erase reverse edges). Show that $|P_k|$ is at most half of the number of edges in X.

Solution: Notice X contains two edge disjoint paths since the outdegree of s is 2, indegree of t is 2 and all other vertices are balanced. Notice that any path in X was a candidate for P_k. Then

$$2|P_k| \leq |X| \leq |P_k| + |P_{k+1}|$$

4: Prove (b). Fix k and consider the smallest $l > k$ such that P_l uses a reverse edge of P_k. Use that there was a reverse edge and adapt the previous argument.

Solution: Same as previous there was a reverse edge, so we can substract 2.

$$2|P_k| \leq |X| \leq |P_k| + |P_{k+1}| - 2$$

5: How many augmentations are needed in Edmonds-Karp Algorithm? What is the resulting running time?

Solution: The length of the shortest path is at most n. In every augmenting path, at least one edge is being saturated. Every edge (or its reverse) is the saturated once in at most $\frac{n}{2}$ distances. Hence together, there are at most $\frac{mn}{2}$ iterations.

Every iteration takes one BFS, which takes $O(m)$. Hence the running time is $O(\frac{m^2n}{2})$.

Network flows as linear programs

6: Formulate the maximum flow problem for network (G, u, s, t) as a linear program (P). (Hint: Similar to shortest path.) Assume $G = (V, E)$.

Solution:

$$(P)\begin{cases} \text{maximize} & \sum_{ut} f_{ut} - \sum_{tw} f_{tw} \\ \text{subject to} & \sum_{uv} f_{uv} - \sum_{vw} f_{vw} = 0 \text{ for all } v \in V \setminus \{s,t\} \\ & f_e \leq u(e) \text{ for all } e \in E \\ & 0 \leq f_e \text{ for all } e \in E \end{cases}$$

7: Write the dual (D) to (P). Use dual variables y_v, where $v \in V \setminus \{s,t\}$ for $\sum_{uv} f_{uv} - \sum_{vw} f_{vw} = 0$, and z_e such that $e \in E$ for $f_e \leq u(e)$.

Solution:

$$(D)\begin{cases} \text{minimize} & \sum_{e\in E} u(e) z_e \\ \text{subject to} & -y_v + y_w + z_{vw} \geq 0 \text{ for all } vw \in E,\ v, w \in V\setminus\{s,t\} \\ & y_w + z_{sw} \geq 0 \text{ for all } sw \in E \\ & -y_v + z_{vs} \geq 0 \text{ for all } vs \in E \\ & -y_v + z_{vt} \geq 1 \text{ for all } vt \in E \\ & y_w + z_{tw} \geq -1 \text{ for all } tw \in E \\ & z_e \geq 0 \text{ for all } e \in E. \end{cases}$$

8: Add two artificial variables $y_s = 0$ and $y_t = -1$. Then the constraints all unify to the form $-y_v + y_w + z_{vw} \geq 0$ for all $vw \in E$. Write the new program (D').

Solution:

$$(D')\begin{cases} \text{minimize} & \sum_{e\in E} u(e)z_e \\ \text{subject to} & -y_v + y_w + z_{vw} \geq 0 \text{ for all } vw \in E \\ & y_s = 0; y_t = -1 \\ & z_e \geq 0 \text{ for all } e \in E. \end{cases}$$

Interpretation: every edge gives a bound how much of a decrease can occur. Use the following figure to try to find a feasible solution (assign $z_e = 0$ and see why it is not a feasible solution.)

9: Recall that every s-t-flow can be decomposed into weighted s-t-paths. Try to interpret (D') using s-t paths.

Solution:

$$(D')\begin{cases} \text{minimize} & \sum_{e\in E} u(e)z_e \\ \text{subject to} & \sum_{e\in P} z_e \geq 1 \text{ for every } s\text{-}t - \text{path } P \\ & z_e \geq 0 \text{ for all } e \in E. \end{cases}$$

If z_e is 0,1, it gives that every path must have some edge on it, where $z_e = 1$ is an edge in a cut.

10: Use minimum s-t-cut to find an optimal solution to (D').

Solution: Put edges of the cut $z_e = 1$, others 0 and check it indeed works.