Algorithm Theory, Tutorial 5

Johannes Kalmbach

University of Freiburg

johannes.kalmbach@gmail.com

January 2020

General Hints

- Contact tutor (`johannes.kalmbach@gmail.com`) for questions concerning corrections etc.
- Contact forum (`daphne.informatik.uni-freiburg.de`) for everything else
- Suggestion: Submit in groups of two (better for understandable algorithms)
- Submit readable solutions (LaTeX as pdf, CLEAN handwriting (+ good scan if necessary))
- Spend enough time on exercise sheets and writeup (you and I have to understand your submission).

Exercise 1 Matching & Vertex Cover in Bipartite Graphs

Let $G = (V, E)$ be a graph and assume that $M^* \subseteq E$ is a maximum matching and that $S^* \subseteq V$ is a minimum vertex cover (i.e., M^* is a largest possible matching and S^* a smallest possible vertex cover). We have seen in the lecture that for every graph G, it holds that $|M^*| \leq |S^*|$ because the edges in M^* have to be covered by disjoint nodes in S^*. In this exercise, we assume that G is a *bipartite graph* and our goal is to show that in this case, it always holds that $|M^*| = |S^*|$.

a) Recall that we can solve the maximum bipartite matching problem by reduction to maximum flow. Also recall that if we are given a maximum matching M^* (and thus a maximum flow of the corresponding flow network), we can find a minimum s-t cut by considering the residual graph. Describe how such a minimum cut looks like.

   Hint: Consider the set of all nodes which can be reached from an unmatched node on the left side via an alternating path.

b) Use the above description to show that any bipartite graph G has a vertex cover S^* of size $|M^*|$.

c) Show that the same thing is not true for general graphs by showing that for every $\varepsilon > 0$, there exists a graph $G = (V, E)$ for which $|S^*| \geq (2 - \varepsilon)|M^*|$.

   Hint: First try to find any graph for which $|S^*| > |M^*|$.

Describe the min-cut

- Let $B = (U \cup V, E)$ be a bipartite graph with maximum matching M^*
- In the corresponding flow network there is a source node s which is connected to all nodes in U and a target node t to which all nodes in V are connected.
- For $u \in U$ and $v \in V$, there is an edge from u to v iff u and v are adjacent in B.
- All edges have capacities 1.

- Let f be the maximum flow that corresponds to M^* and R the residual graph w.r.t. f.
- We know from the lecture that (A^*, B^*) is a minimum cut where A^* is defined as the set of nodes which can be reached from s by a path in R on which each edge has a positive capacity.
- For an $u \in U$, the edge (s, u) has positive residual capacity iff u is not matched.

- As there are no edges in B directing from V to U, we know that all edges from U to V are forward edges and all edges from V to U are backward edges.
- If (u, v) for an $v \in V$ has positive residual capacity, there is no flow through (u, v) and we know $\{u, v\} \notin M^*$.
- If (v, u') for an $u' \in U$ has positive residual capacity, there is flow through (u', v) and we know $\{u', v\} \in M^*$. Hence, A^* consists of s and all nodes which can be reached from an unmatched node on the left side via an alternating path.

- content...

1b $|\text{Maximal Matching}| = |\text{Min Vertex Cover}|$

- Define $S^* = (U \cap B^*) \cup (V \cap A^*)$.
- We need to show that $\boldsymbol{S^*}$ is a vertex cover and that $|\boldsymbol{S^*}| = |\boldsymbol{M^*}|$

$\boldsymbol{S^*}$ is a vertex cover:

- S^* covers all edges with left endpoint in B^* or right endpoint in A^*.
- Need to show that there is no edge in the graph with left endpoint in A^* and right endpoint in B^*.

- Assume $e = \{u, v\}$ with $e \in A^*$ and $v \in B^*$
- As $u \in A^*$, there is an alternating path to u.
- So if $e \notin M^*$, we could extend this path to v and therefore have $v \in A^*$, a contradiction.
- Otherwise, if $e \in M^*$, an alternating path reaching u must also contain v which implies that also $v \in A^*$, a contradiction.

$|\boldsymbol{S^*}| = |\boldsymbol{M^*}|$:

- We showed before that there is no edge between a node in $U \cap A^*$ and a node in $V \cap B^*$.
- As there is no edge directed from V to U this means that the edges going out of A^* are those from s to $U \cap B^*$ and from $V \cap A^*$ to t.
- These edges stand in a 1-1 correspondence to the nodes in S^*
- So the size of the minimum cut (A^*, B^*) equals $|S^*|$. As the size of a min-cut also equals the maximum flow which equals $|M^*|$, we obtain $|S^*| = |M^*|$.

Exercise 1 c

Show that the same thing is not true for general graphs by showing that for every $\varepsilon > 0$, there exists a graph $G = (V, E)$ for which $|S^*| \geq (2 - \varepsilon)|M^*|$.

Hint: First try to find any graph for which $|S^*| > |M^*|$.

Exercise 2 - Matching in Regular Graphs

The degree of a node in a graph is the number of its neighbors. A graph is called r-regular for an $r \in \mathbb{N}$ if all nodes have degree r.

a) Show that any regular bipartite graph has a perfect matching.

b) Show that an n-regular graph with $2n$ nodes has a matching of size at least $n/2$.

Show that any regular bipartite graph has a perfect matching.

- We are going to use Hall's Theorem, let U, V be the two subsets of nodes that only have edges between them.
- Let $U' \subset U$.
- The graph is r-regular, so U' has $r * |U'|$ "outgoing" edges.
- Since each node in $N(U')$ has r "incoming" edges, so the $r * |U'|$ edges going out of U' hit at least $|U'| * r/r = |U'|$ different nodes in $N(U')$.
- Thus $|N(U')| \geq |U'|$.
- Similarly $|N(V') \geq |V'|$ for $V' \subset V$
- This directly implies $|U| = |V|$
- Thus the graph has a bipartite matching according to Hall's Theorem

Exercise 2b

- Show that an n-regular graph with $2n$ nodes has a matching of size at least $n/2$.
- This graph has exactly .... edges
- The simple greedy approach works in this case:
- Pick any edge for the matching.
- Remove all edges that cannot be matched any more (those are at most ....)
- Repeat this procedure as long as there are still edges left
- After $n/2$ steps we have removed at most ...... $\leq |E|$ edges, so we were able to create a matching of size at least $n/2$

Exercise 3

Cover all Edges

You are given an undirected graph $G = (V, E)$, a capacity function $c : V \to \mathbb{N}$, and a subset $U \subseteq V$ of nodes. The goal is to cover every edge with the nodes in U, where every node $u \in U$ can cover up to $c(u)$ of its incident edges.
Formally, we are interested in the existence of an assignment of the edges to incident nodes in U such that each node u gets assigned at most $c(u)$ of its incident edges.

Devise an efficient algorithm to determine whether or not such an assignment exists for a given subset U and a given cost function c and state its runtime.

- Each node has a certain capacity
- Each unit of capacity can be used to cover one of the incident edges
- This kind of resource management problem can often be stated as max-flow:

Exercise 3, Graphically

Exercise 3, Formally

- Define a new Flow-Graph as $U' = \{s,t\} \dot\cup U \dot\cup E$,
  $E' = \{(s,u) : u \in U\} \cup \{(e,t) : e \in E\} \cup \{(u,e) : u, e \text{ incident in } G\}$
- Define Capacities as $c'(s,u) = c(u), c(e,t)' = 1, c'(u,e) = 1$
- The original problem is solvable if and only if the max $s - t$ flow on this Graph has value $|E|$.
- Runtime using Ford-Fulkerson is ...