Albert-Ludwigs-Universität, Inst. für Informatik
Prof. Dr. Fabian Kuhn
H. Ghodselahi, Y. Maus January 23, 2017

Algorithm Theory, Winter Term 2016/17
Problem Set 6 - Solution

Exercise 1: Free Vacation! (12+3 points)

Remark: This is a previous exam question.

A high school class is made an interesting offer by a reality TV show in which couples of students (one female and one male student) get a free vacation trip to an exciting location. This class consists of

- n boys $B = \{b_1, \dots, b_n\}$ and
- n girls $G = \{g_1, \dots, g_n\}$.

There are n different locations $L = \{\ell_1, \dots, \ell_n\}$ for the students to choose from. The girls are not picky about the destinations, but each girl g is only willing to partner up with an individual subset $B_g \subseteq B$ of all available boys. The boys on the other hand do not care that much about with whom they go on vacation, but they care about the location; each boy b has an individual subset $L_b \subseteq L$ of locations it is willing to visit.

a) Is it possible that everyone can go on a free vacation? Devise an algorithm that answers this question.

b) What is the time complexity of your algorithm if you assume that each girl is willing to partner up with at most $\sqrt{n}$ different boys and if you assume that each boy is willing to to visit at most $\sqrt{n}$ different locations?

Solution:

a) We first construct the flow network. We add a node s as source, a node t as sink, a set G of n nodes where each node corresponds to one girl, a set B of $2n$ nodes, that is two nodes for each boy, and finally a set L of n nodes where each node corresponds to one location. We connect s to all nodes in G. Further the node t is connected to all nodes in L. For each boy b_i we have two nodes b_i^{in} and b_i^{out}. We connect every *incoming* node with every outgoint node, that is, we introduce the edges (b_i^{in}, b_i^{out}) for $i = 1, \dots n$. Each girl g_i where $1 \leqslant i \leqslant n$ is willing to partner up with the boys in $B_{g_i} \subseteq B$. Hence for all g_i we add edges (g_i, b^{in}) for all $b \in B_{g_i}$. This induces the set of edges E_1 as you can see in Figure 1 as black box.

To be done with constructing the flow network, for every boy b_j where $1 \leqslant j \leqslant n$ who is willing to go to locations in $L_{b_j} \subseteq L$ we add the edges need to connect the node b_j^{out} with the nodes in L_{b_j}, that is, we add edges (b_j^{out}, l) for all $l \in L_{b_j}$. This induces the edges E_2 as it is shown in Figure 1 as black box.

We set the capacities of all edges to 1. We run the Ford-Fulkerson algorithm to find a maximum $s-t$-flow in the constructed network. It is possible that everyone can go on a free vacation if and only if the algorithm returns a maximum flow of value n. A quick proof for that:

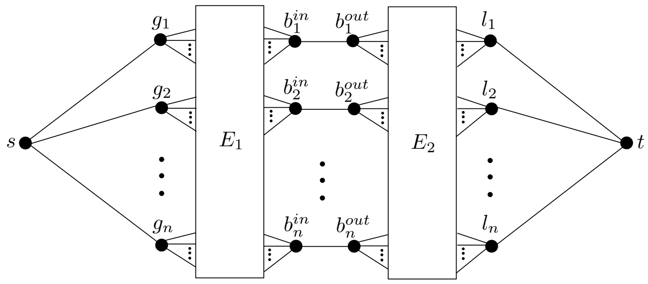

Figure 1: Network Flow

- **maximum flow with value $n \Rightarrow$ free vacation:**
  Let f be a maximum flow of value n. Then there are n (edge disjoint) augmenting paths (Menger's Theorem) (s, g, b, l, t) with $g \in G, b \in B, l \in L$. Because every *girl* node has only one incoming edge, every location node has only one outgoing edge and we split the nodes for the boys into two nodes (similar to finding vertex disjoint paths), we ensure that every girl, boy and location occur in exactly one of these paths (essential point!). Thus the paths yield a valid matching of all boys, girls and locations and everyone can go to a free vacation.
- **Free vacation for everyone $\Rightarrow$ maximum flow with value n:**
  If everyone can go on a free vacation there are matchings $\{(g, b, l)\}$ such that every girl, boy and location occurs in exactly one triple matching. This leads to n edge disjoint paths, each with bottleneck 1, from s to t (similar as above). Thus there is a maximum flow with value n.

b) Basically the running time of Ford-Fulkerson is $\mathcal{O}(|E|C)$ where $|E|$ is the total number of edges and C is maximum flow value. The total number of edges in our network flow is $2n(1 + \sqrt{n})$ since each girl is willing to partner up with at most $\sqrt{n}$ boys and every boy is willing to go to at most $\sqrt{n}$ locations. Because the cut $(\{s\}, V\backslash\{t\})$ has capacity n, the maximum possible flow which we can push from s to t is bounded by n. Thus the running time is $\mathcal{O}(n^2\sqrt{n})$.

Exercise 2: Ford Fulkerson revisited.(10 points)

Show that the below statement is correct or prove that it does not hold.

Often the Ford Fulkerson algorithm needs many augmenting paths. If the algorithm always chooses the 'correct' augmenting paths it never has to choose more than $|E|$ paths.

Solution:

Let $G = (V, E)$ be a flow network with max flow $f : E \to \mathbb{R}^+$. In the following we show the existence of at most $|E|$ augmenting paths which form the max flow f. To construct these paths we make use of the max flow f. Thus our approach is not helpful for an algorithm because it first has to know the max flow f before constructing the augmenting paths.

Construction of One Augmenting Path: Let $G(f) = (V, E_f)$ be the graph induced by f where $E_f = \{d \in E \mid f(d) > 0\}$. If $|f| = 0$ the graph $G(f)$ does not have any edges and the claim holds. If $|f| > 0$ then there is a path from s to t in $G(f)$. Pick any such path and denote it by P. Then there is some edge e on the path with $f(e) = \min\{f(d)|d \text{ is edge on } P\}$. Let the first augmenting path be P with value $f(e)$.

Iterating the Construction: Redefine the flow network by reducing all capacities of G on the path P by $f(e)$. This way one obtains a new flow network with max flow $|f| - f(e)$ which is met by a flow f' which we define as the flow f reduced by the first augmenting path. To obtain the second augmenting path we again look at the induced graph $G(f')$ and proceed as before. The crucial observation is that $G(f')$ lost edge e (and we are done if $G(f')$ does not have any edge). Thus we can only repeat this procedure at most $|E|$ times and in the end all $|E|$ augmenting paths combined form the max flow of the original flow network.

Exercise 3: Large Chromatic Number without Cliques. (15 points)

A c-coloring of a graph $G = (V, E)$ is a function $\phi : V \to \{1, \ldots, c\}$ such that any two neighboring nodes have different colors, i.e., for each $\{u, v\} \in E$ $\phi(u) \neq \phi(v)$. The chromatic number $\chi(G)$ of a graph G is the smallest integer c such that a c-coloring of G exists, e.g., the chromatic number of an k node clique is k. In the following we want to use probability theory to show that not only cliques imply large chromatic number, in particular we want to show the following:

For any k and l there is a graph with chromatic number greater than k and no cycle shorter than l.

In the following consider a (random) graph $G_{n,p}$ on n nodes. Each (possible) edge $\{u, v\}, u, v \in V$ exists with probability $p = n^{\frac{1}{2l}-1}$.

1) **(1 point)** An independent set I of a graph G is a collection of nodes such that $G[I]$ does not have any edge. The independence number $\alpha(G)$ of a graph denotes the size of the largest independent set.

   Explain why $\chi(G) \geqslant |V(G)|/\alpha(G)$ holds.

2) **(5 points)** Show that for $a = \lceil \frac{3}{p} \ln n \rceil$ we have

$$\Pr[\alpha(G) \geqslant a] \longrightarrow_{n \to \infty} 0.$$

   Hint: There are $\binom{n}{a}$ choices for the nodes of an independent set of size a. What is the probability that a specific nodes form an independent set? Also use the linearity of expectation!

3) **(5 points)** Let X be the number of cycles of length at most l. Show that its expectation $E[X]$ can be upper bounded by $\frac{n}{4}$ for large n.

   Hint: What is the probability that j specific nodes form a cycle? How many choices of nodes which can possibly form a cycle of length less than l are there? Again, use the linearity of expectation.

4) **(3 points)** From 2) and 3) we can deduce that $\Pr[X \geqslant n/2 \text{ or } \alpha(G) \geqslant a] < 1$ holds. This means that there exists a graph H with n nodes where the number of cycles with length less than l is less than $n/2$ and the independence number is smaller than a. So H has a small independence number but it might contain some short cycles.

   Explain how to modify the graph H to obtain a graph H' with no cycles of length at most l, $\alpha(H') < a$ and $|V(H')| \geqslant n/2$.

5) **(1 point)** Show that the graph H' has no cycle of length at most l and chromatic number at least k.

Remark: All subquestions in this exercise can be solved independently from each other (by using the results of the other questions as black box).

If you have difficulties with this exercise please use the forum or ask your tutors to get help.

Solution:

We first fix the parameters k and l and then do the following steps to find a graph which has chromatic number larger than k and does not have cycles shorter than l. Note that k and l cannot be a function of the number of nodes as n is chosen sufficiently large in many the following steps where the *sufficiently large* depends on k and l.

1) Every color class of a valid coloring forms an independent set. Thus no color class can contain more than $\alpha(G)$ nodes which implies that there have to be at least $\frac{|V(G)|}{\alpha(G)}$ color classes.

2) The probability that a given set of a nodes forms an independent set is $(1-p)^{\binom{a}{2}}$. By a union bound we obtain

$$
\begin{aligned}
\Pr[\alpha(G) \geqslant a] &= \Pr[\exists W \subseteq V, W \text{ independent set}, |W| \geqslant a] && (1)\\
&= \Pr[\exists W \subseteq V, W \text{ independent set}, |W| = a] && (2)\\
&\leqslant \sum_{W \subseteq V, |W|=a} \Pr[W \text{ is an independent set}] && (3)\\
&\leqslant \binom{n}{a}(1-p)^{\binom{a}{2}} && (4)\\
&\leqslant n^a e^{-pa(a-1)/2} && (5)\\
&\leqslant n^a n^{-3(a-1)/2} \longrightarrow_{n\to\infty} 0. && (6)
\end{aligned}
$$

3) If we choose j specific nodes the probability that they form a cycle is p^j. The number of potential cycles of length j is certainly at most n^j. With the linearity of expectation we obtain.

$$
E[X] \leqslant \sum_{j=3}^{l} n^j p^j = \sum_{j=3}^{l} n^{\frac{1}{2l}j} \overset{*}{\leqslant} \frac{n^{\frac{l}{2l}}}{1-n^{-\frac{1}{2l}}} = \frac{n^{\frac{1}{2}}}{1-n^{-\frac{1}{2l}}}. \tag{7}
$$

To show inequality $(*)$ we used the geometric series formula twice and then reduced the fraction by $n^{\frac{1}{2l}}$. For n large enough this is smaller than $\frac{n}{4}$ (here we get a dependence of n on l).

4) The graph H has at most $n/2$ cycles of length at most l and independence number $\alpha(H) < a$. We obtain H' by removing one node from each of these cycles. Removing a node from a graph can not increase the independence number. Then the graph H' has at least $n/2$ nodes, no cycles shorter than l and independence number $\alpha(H') < a$.

5) The graph H' has the following chromatic number.

$$
\chi(H') \geqslant \frac{|V(H')|}{\alpha(H')} \geqslant \frac{n/2}{3n^{1-\frac{1}{2l}} \ln n} = \frac{n^{\frac{1}{2l}}}{6 \ln n}. \tag{8}
$$

If we chose n sufficiently large we obtain $\chi(H') > k$ (here we get that n depends on k).

The above proof was a probabilistic proof which shows that such graphs exist. However, it is very hard to actually construct any of these graphs.

Bonus Question: Special Promotion at Christmas! (10* points)

To increase its Christmas sales a small kiosk has a special promotion: If a customer buys two articles whose prices add up to a value which ends with $11, 33, 55, 77$ or 99 cents, he will receive a voucher, worth the corresponding cent value.

Devise an algorithm which computes an optimal strategy for buying a given collection of goods (here only the price of a good matters).

Solution:

Assume that you want to buy n items which are given by the set X. We reduce the problem to an instance of maximum weighted bipartite matching. We create a bipartite graph $G = (X_1 \cup X_2, E)$ where one side of the nodes is formed by all nodes which have an odd cent value, i.e.,

$$X_1 = \{x \in X | \text{the cent value of } x \text{ is odd}\}, \tag{9}$$

and the other side is formed by all nodes which have an even cent value, i.e.,

$$X_2 = \{x \in X | \text{the cent value of } x \text{ is even}\}. \tag{10}$$

We add an edge between $x \in X_1$ and $y \in X_2$ with value $t \in \{11, 33, 55, 77, 99\}$ if and only if the cent values of x and y add up to t.

Now, finding a maximum bipartite matching in this graph is equivalent to grouping X into groups of two elements w.r.t. maximizing the value of vouchers one obtains. (one can be more specific here, but we do not expect it in this bonus question)

The maximum weighted bipartite matching was solved in the lecture which we use as a black box.