Homework 2

Topics in Graph Algorithms
CS598CCI, Spring 2020
Due: 2/26/2020 10am

Instructions and Policy: Each person should write up their own solutions independently. You need to indicate the names of the people you discussed a problem with. Solutions to most of these problems can be found from one source or the other. Try to solve on your own first, and cite your sources if you do use them.

Please write clearly and concisely. Refer to known facts. You should try to convince me that you know the solution, as quickly as possible.

Do as many problems as you can. I expect you to do at least 3.

Problem 1. We saw an algorithm in lecture that (randomly) reduces the problem of detecting whether a graph G has a perfect matching to computing a determinant of a $n \times n$ matrix. Using the detection algorithm as a black box gives an inefficient algorithm to find a perfect matching. Harvey showed a nice algorithm that cleverly reduces the work needed so that the running time to find a perfect matching is $O(n^\omega)$. Read the paper and explain the algorithm and analysis in your own words.

Problem 2. Petersen's theorem states that any cubic (degree of each vertex is 3) bridgeless graph (has no edge whose removal creates two components) has a perfect matching. Prove this theorem by exhibiting a fractional solution that is feasible for the perfect matching polytope we saw in lecture.

Problem 3. In a graph $G = (V, E)$ a set of edges $S \subseteq E$ is an edge cover if each vertex v is incident to at least one edge in S. The goal is minimum edge cover problem is to find a an edge cover of smallest cardinality. If edges have non-negative weights we obtain the minimum weight edge cover problem. Note that in edge cover we seek to cover vertices by edges while in the NP-Hard vertex cover problem we seek to cover edges by vertices.

- Describe a polynomial-time reduction of minimum weight edge cover problem to minimum weight perfect matching. *Hint* Use a reduction similar to the one from max weight matching to max weight perfect matching but choose the weight of the edges between the copies in a careful fashion. You may want to try the unit weight case first.

- Prove that the following inequality system characterizes the edge cover polytope.

$$\begin{array}{rll} x\left(E[U] \cup \delta(U)\right) \geq & \frac{|U|+1}{2} & U \subseteq V; |U|\, odd \\ 0 \leq x(e) \leq 1; & & e \in E \end{array}$$

Problem 4. Consider HR (hospital resident) problem with incomplete preferences. The input is a bipartite graph $(R \cup H, E)$ and for each resident r a preference list over the hospitals in $N(r)$ (the neighbors of r in G) and for each h a preference list over the residents $N(h)$. Each hospital has a capacity u_h. A partial assignment of residents to hospitals is feasible and if no hospital is assigned more than u_h residents and the assignment respects the edges of the given graph. It is stable if there is no pair $(r, h) \in E$ such that r is not assigned to h *and* at least one of the following conditions holds:

- h is not full, and r is unassigned or $r.$ is assigned to h' and r prefers h to h'.
- h is full, r is unassigned, and there is a resident r' assigned to h such that h prefers r over r'
- h is full, r is assigned to h' and r' is assigned to h and r prefers h to h' and h prefers r to r'.

1. Prove that there is always a stable assignment for such an instance by reducing it to a standard stable marriage instance with equal number of men and women and full preference lists.
2. Describe a variant of Gale-Shapely algorithm that outputs a stable assignment and runs in $O(m + n)$ time where $m = |E|$ and $n = |R| + |H|$.
3. **Extra credit:** Prove that in every stable matching the number of residents assigned to a hospital is the same.

Problem 5. Consider the algorithm of Edmonds to find an M-augmenting path in a graph G via shrinking blossoms. Suppose G has no M-alternating X-X walk where X is the set of M-exposed nodes, then M is a maximum matching. In this case, can you algorithmically find a witness U in the Tutte-Berge formula to certify that M is a maximum matching? For this you need to work with a specific algorithm that tries to find a walk, for example the reduction to a directed path problem that we discussed in class/lecture notes (see also Schrijver's book). Note that one may find that M is a maximum matching in G after several recursive steps. Can you extend the witness U found in the recursive step to the original graph?

Problem 6. Extra credit: Design an algorithm that outputs all stable matchings of a given instance with n men and n women such that your algorithm runs in time polynomial in n and the number of stable matchings of the input instance.