CS466/666, Fall 2011: Assignment 5

Out: November 17, Due: December 1, 5pm

Remark: For this assignment, all algorithms you design must be deterministic and polynomial-time, unless specifically said otherwise.

1. **Final Scheduling** (11 marks) Let $P = p_1, \dots, p_n$ be a set of professors, and $S = s_1, \dots, s_m$ be a set of students; $m \geq n$. There is also a matrix $A = (a_{i,j})$, with $a_{i,j}$ being 1 if student s_i needs to take an oral final exam with professor p_j. But the professors are very busy: each professor p_j has only two time slots t_j^1 and t_j^2 where she is available. Students may have up to 4 final exams. Of course no student or professor may have two different final exams at the same time.

   Given an $O(mn)$ time algorithm to test whether the final exams can be scheduled.

   Remarks: (1) You may assume that any two time slots either are identical or are disjoint. (You are encouraged to think about where you need this - it's quite subtle!) (2) $O(mn)$ is not the best possible run-time (if the matrix is stored as a sparse matrix), but suffices for full credit. (3) The algorithm idea is quite simple, but there are lots of little details to be considered. A substantial part of the marks will be for attention to details and the quality of explanations.

2. **Independent set with bounded max-degree** (8 marks) Show that Independent Set is APX-hard even in graphs with maximum degree 3. Specifically, give a value ϵ and prove that there is no polynomial-time $(1-\varepsilon)$-approximation algorithm for Independent Set in graphs with maximum degree 3.

   You may use (without proof) that Max2SAT($2L$) is APX-hard. Here, Max2SAT($2L$) is the special case of Max2SAT where for any $i = 1, \dots, n$ there are at most two occurrences of x_i and at most two occurrences of $\overline{x_i}$ in the clauses. The gap-constant in this APX-hardness is $1 - \frac{1}{2013}$, i.e., there is a reduction from SAT such that if the SAT-instance is satisfiable then the Max2SAT($2L$)-instance has all clauses satisfied, and if the SAT-instance is not satisfiable then the Max2SAT($2L$)-instance has at most $(1 - \frac{1}{2013})$#clauses satisfied.

3. **Independent Set of Unit Squares** (10 marks) Let S be a set of unit squares in 2-d, i.e., every element in S is a set $[x, x+1] \times [y, y+1]$ for some $x, y \in \mathbb{R}$. The *independent set in unit squares* problem asks to find a maximum subset S' of S such that no two squares in S' intersect (not even at the boundary.)

   (a) Let $S_{2,2}$ be all those elements in S for which $0 \leq x \leq 1$ and $0 \leq y \leq 1$, i.e., any element in $S_{2,2}$ is entirely within the $[0,2] \times [0,2]$-square. Explain how to solve the independent set problem in $S_{2,2}$ in time that is linear in $|S_{2,2}|$.

(b) Let $S_{2,\infty}$ be all those elements in S for which $0 \leq x \leq 1$, i.e., any element in $S_{2,\infty}$ is entirely within a vertical strip that intersects that x-axis at $[0, 2]$. Explain how to solve the independent set problem in $S_{2,\infty}$ in time that is polynomial in $|S_{2,\infty}|$.

(c) Give a $\frac{1}{2}$-approximation algorithm for the independent set problem in S.

(d) (Bonus:) Give a PTAS for this problem.

4. **Maximum matching** (11 marks) Recall that the *Maximum matching* problem asks, given a graph $G = (V, E)$, to find a largest possible matching, i.e., a set M of edges such that no two edges in M have a common endpoint. Finding a maximum matching is polynomial, but the algorithm is fairly complicated.[1] Therefore it still makes sense to apply techniques for NP-hard problems, especially if they lead to linear-time algorithms.

   (a) Formulate Maximum Matching as an integer program. Clearly state the intended meaning of variables and explain your constraints.

   (b) Give a linear-time algorithm that computes a $\frac{1}{2}$-approximation for maximum matching.

   (c) Let G be a graph with maximum degree 3. Give an algorithm that tests whether G has a matching of size k that has run-time $O(f(k)(n+m))$, where $f(k)$ is any function that is independent of n and m.

5. **Lost cow with graduated pricing.** (0 marks. This question is NOT required for Assignment 5—we won't get to this in class on time— but put here to give you an idea of what questions one can ask regarding online problems.)

   In the lost-cow problem, our cost measure was exactly the distance travelled. In practical settings, there is a difference in cost depending on whether that region has been explored previously or not. So consider the lost-cow problem, except that it takes 1 unit cost to walk along new terrain, but c units cost ($c > 0$) to walk along previously visited terrain.

   (a) Using the doubling strategy, what is the worst-case competetive ratio in this cost-model? What is the expected competetive ratio if you start in a random direction?

   (b) Recall that a *geometric search strategy* is the strategy of exploring by $d, d^2, d^3, \ldots,$ units (always returning to the origin and alternating sides); the doubling strategy is a geometric search strategy for $d = 2$. What is the best geometric exploration strategy (depending on c) for the above cost model?

[1] Knowing the algorithm is not required and will not help to answer the questions.