COMP 5711: Advanced Algorithm
2014 Fall Semester
Midterm Exam Solutions

Problem 1 (15pts)

(a) $k = O(\log n/\log\log n)$.
Proof: $k^k = n^{O(1)} \Leftrightarrow k\log k = O(\log n)$. When $k = O(\log n/\log\log n)$, this certainly holds. Meanwhile, we can't have $k = \omega(\log n/\log\log n)$, since this would lead to $k\log k = \omega(\log n/\log\log n)\cdot\log\log n = \omega(\log n)$.

(b) If $k = n$, then FPT does not impose any running time constraint on the algorithm, as long as it terminates. So FPT equals to the class of all decidable problems.

(c) False. Suppose the optimal vertex cover is C^*. A 2-approximation algorithm for vertex cover returns a C such that $|C| \le 2|C^*|$, while a 2-approximation for independent set should return an I such that $|I| \ge \frac{1}{2}(|V| - |C^*|)$. Approximating $|C^*|$ does not mean you also approximate $|V| - |C^*|$.

Problem 2 (20pts) We number the vertices as $1, 2, \ldots, n$. The ILP is

$$\begin{aligned}
\text{min} \quad & \sum_{i=n}^{n} x_i \\
\text{subject to} \quad & \sum_{j=1}^{n} a_{ij}x_j \ge 1, i = 1, \ldots, n \\
& x_i \in \{0,1\}, i = 1, \ldots, n,
\end{aligned}$$

where

$$a_{ij} = \begin{cases} 1, & i = j \text{ or } e_{ij} \in E \\ 0, & \text{otherwise.} \end{cases}$$

To obtain a linear program, we relax the last constraint to the following

$$0 \le x_i \le 1, i = 1, \ldots, n.$$

From the optimal fractional solution of the LP, we round every $x_i \ge \frac{1}{d+1}$ to 1 and 0 otherwise. It is easy to see that the rounded solution forms a dominating set. Consider any vertex v_i. In the LP problem we have $\sum_{j=1}^{n} a_{ij}x_j \ge 1$, so at least one neighbor of x_i (or itself) will have a fractional solution larger or equal to $\frac{1}{d+1}$. Meanwhile, the rounding increases the objective function by at most a factor of $d+1$, so it is a $(d+1)$-approximation.

Problem 3 (20pts)

(a) Yes. Let S be the solution found by our algorithm and S^* the optimal solution, then

$$\sum_{i\in S} v_i \ge \sum_{i\in S} \bar{v}_i \ge \sum_{i\in S^*} \bar{v}_i \ge \sum_{i\in S^*}(v_i - \theta) \ge \sum_{i\in S^*} v_i - n\theta \ge (1-\epsilon)\sum_{i\in S^*} v_i,$$

where $\theta = \epsilon v_{max}/n$. The running time is still $O(\frac{n^3}{\epsilon})$.

(b) No. The algorithm actually still provides a $(1+\epsilon)$-approximation, but the running time is not polynomial. The key is that rounding to the multiples of θ reduces the values to a small (polynomial in n and $1/\epsilon$) integer domain, but exponential rounding does not. It only reduces the number of *distinct* values.

Problem 4 (15pts) First, since no edge can be added to a locally optimal matching, that means every edge has at least one endpoint matched. If we pick both endpoints of all the matched edges, then all edges must be covered.

Let S be a locally optimal solution, and S^* the optimal. Note that S consists of $|S|/2$ edges of the matching and these edges are disconnected. Each of these edges must be covered by a different vertex, so $|S^*| \geq |S|/2$, i.e., $|S| \leq 2|S^*|$.

Problem 5 (10pts) (a) Nash Equilibrium: $s \to u \to t$, $s \to v \to t$, price: $5+2+2+5=14$; Social Optimum: $s \to u \to t$, $s \to v \to t$, price:14;

(b) Nash Equilibrium: $s \to v \to u \to t$, $s \to v \to u \to t$, price: $2*2*2+2*2*2=16$; Social Optimum: $s \to u \to t$, $s \to v \to t$, price:14.

Problem 6 (20pts) Let O be the of elements covered by the optimal solution, and C_i be the set of elements covered by the greedy algorithm after the i-th iteration. We will show by induction that $|O|-|C_i| \leq (1-1/k)^i|O|$. This will imply that after k iterations, $|O|-|C_k| \leq (1-1/k)^k|O| \leq 1/e \cdot |O|$, which means that the greedy algorithm provides a $(1-1/e)$-approximation.

The base case $i=0$ is trivial. Suppose $|O|-|C_i| \leq (1-1/k)^i|O|$ for some i, we will show $|O|-|C_{i+1}| \leq (1-1/k)^{i+1}|O|$. Let A_{i+1} be the set of newly covered elements in the i-th iteration of the greedy algorithm, and we have $|C_{i+1}| = |C_i| + |A_{i+1}|$. We know that O is covered by k sets, so $O - C_i$ is covered by at most k sets. Thus, there must exist a set that covers at least a fraction of $1/k$ of $O - C_i$. The greedy algorithm will cover at least this many elements, so $|A_{i+1}| \geq \frac{1}{k}|O - C_i| \geq \frac{1}{k}(|O| - |C_i|)$. Now we have

$$\begin{aligned}
|O| - |C_{i+1}| = |O| - |C_i| - |A_{i+1}| \leq |O| - |C_i| - \frac{1}{k}(|O| - |C_i|) &= \left(1 - \frac{1}{k}\right)(|O| - |C_i|) \\
&\leq \left(1 - \frac{1}{k}\right)\left(1 - \frac{1}{k}\right)^i |O| \qquad \text{(induction hypothesis)} \\
&= \left(1 - \frac{1}{k}\right)^{i+1} |O|.
\end{aligned}$$