McGill University COMP360 Winter 2011 **Instructor: Phuong Nguyen**

Assignment 7 Solution

Question 1 (10pt) Consider the following variant of the Independent Set problem, called 3GIND here (G for generalized, and 3 for the maximum degree, see below). In this problem we are given an undirected graph G where the degree of each vertex is at most 3. Also, each vertex v in G is associated with a positive weight $w(v)$. The problem is to find an independent set of G of maximum total weight.

Consider the following greedy algorithm:

1. Let V be the set of all vertices of G, and S be the empty set $\varnothing$
2. while V is not empty do
3. pick a vertex v in V of maximum weight
4. add v to S
5. delete v and all its neighbors from V
6. end while
7. return S

(a)[1pt] Let S be the output of the algorithm. Show that S is an independent set.

(b)[4pt] Let T be any independent set of G. Show that for each vertex v in T:

- either v is also in S, or
- there is a vertex u in S such that $w(v) \leq w(u)$ and (v, u) is an edge of G

(c)[5pt] Show that the algorithm returns an independent set of total weight at least 1/3 times the maximum total weight of any independent set in G.

Solution

(a) Every time we choose a vertex to include in the output, its neighbors are removed from the graph, so they will no be chosen. Therefore in the output no two vertices are neighbors. So the output is an independent set.

(b) If v is also in S then there is nothing to prove. So suppose that v is not in S. We show that there is a vertex u in S such that $w(v) \leq w(u)$ and (v, u) is an edge of G. First, some neighbor of v must be in S, because otherwise during the execution of the algorithm v is never removed from the graph, so when the algorithm terminates V still contains v, contrary to the condition of the while loop on line 2.

Now let u be the first neighbor of v that is selected by the algorithm. At the time when u is chosen, it has the maximal weight among all remaining vertices, so in particular its weight is at least as large as that of v, i.e., $w(u) \geq w(v)$.

(c) We use (b) to prove this. Let T be any independent set of G. We will show that $w(S) \geq 1/3w(T)$. Associate each $v \in T$ with a vertex $f(v) \in S$ as follow:

- if $v \in S$ then $f(v) = v$;
- otherwise, let $f(v)$ be a vertex $u \in S$ given by (b) such that (v, u) is an edge of G and $w(u) \geq w(v)$.

Observe that $w(f(v)) \geq w(v)$ for all $v \in T$. So

$$\sum_{v \in T} w(v) \leq \sum_{v \in T} w(f(v))$$

Now in the sum in the RHS, each $f(v)$ is a vertex in S and appears at most three times. Therefore

$$\sum_{v \in T} w(f(v)) \leq 3 \sum_{u \in S} w(u) = w(S)$$

So the total weight of S is at least $1/3$ the total weight of T.

Question 2 (10pt) Recall that Vertex Cover is the following problem:
Input: An undirected graph G and a positive integer k.
Output: Accept if and only if G has a vertex cover of size at most k.
(A vertex cover of G is a set of vertices that contains at least one endpoint of every edge in G.)

This problem is **NP**-complete, and so is MINVC, the problem of computing the minimum size of a vertex cover. Formally, the problem MINVC is specified as follows:
Input: An undirected graph G
Output: A minimum-size vertex cover of G.

Now consider the following greedy algorithm for approximating MINVC:
On input G:

1. Let S be the empty set $\varnothing$,
2. while G contains some edge do
3. let v be the vertex in G of maximum degree
4. add v to S
5. remove v and all edges incident on it from G
6. end while
7. output S

(a)[1pt] Let S be the output of the algorithm. Show that S is a vertex cover of the given graph G.

(b)[8pt] Show that S has size at most $c \log n$ times the minimum size of a vertex cover of G for some positive constant c, where n is the number of vertices of G.

(c)[1pt] According to your argument in (b), what is the constant c?

Solution

(a) Every edge of G is considered in the loop (lines 3,4,5), and at least one of its endpoints is added to S. So S is a vertex cover.

(b) We view this problem as a special instance of the Set Cover problem as follows. The universe is the set E of edges of G. The subsets in the family $\mathcal{F}$ can be represented by the vertices of G: Each vertex v of G determines a subset S_v which consists of all edges that are incident on v.

Now each edges $e = (v, u)$ of G will appear in both S_v and S_u. Therefore the union of all subsets in $\mathcal{F}$ equals to E. The size of each S_v is precisely the degree of v in G. Thus, the algorithm given in the problem is the greedy algorithm given in class for Set Cover applied to this special case.

As a result, it achieves the approximation ratio of

$$1 + \ln(|E|) \leq \ln(n^2) = 2\ln(n) = 2\ln 2 \log(n)$$

(c) The constant c is $2\ln 2$.