CSG399 Problem Set 3

Jiangzhuo Chen

March 9, 2004

1 PTAS for Independent Set

An *independent set* of an undirected graph G is a subset V' of vertices such that no two vertices in V' have an edge in G. The IndependentSet problem is to find a maximum-size independent set in G. It is known that IndependentSet is NP-complete. In this problem, we investigate the approximability of IndependentSet.

Define the *product* of two graphs $G_1 = (V_1, E_1)$ and $G_2 = (V_2, E_2)$ as $G = (V, E)$, where

$$V = \{\langle v_1, v_2 \rangle : v_1 \in V_1, v_2 \in V_2\},$$

and there is an edge between vertices $\langle u_1, u_2 \rangle$ and $\langle v_1, v_2 \rangle$ in G if either $u_1 = v_1$ and $(u_2, v_2) \in E_2$ or $(u_1, v_1) \in E_1$. For a positive integer m, let G^m be defined by the recurrence relation $G^{i+1} = G^i \times G$ and $G^1 = G$.

(a) Prove that G has an independent set of size k if and only if G^m has an independent set of size k^m.

(b) Give a polynomial-time algorithm to construct an independent set of G of size $\lceil k^{1/m} \rceil$ from any independent set of G^m of size k.

(c) Using parts (a) and (b), argue that if there exists a constant c such that there is a polynomial-time c-approximation algorithm for IndependentSet, then there exists a PTAS for IndependentSet.

It is easy to see $V(G^m) = V(G)^m$. There are two ways to decide $E(G^m)$. One is described in the problem statement. The other is my "misunderstanding".

1. Vertices $(v_1, \cdots, v_m)$ and $(u_1, \cdots, u_m)$ are adjacent iff the first different pair of components are adjacent in G, i.e., $\exists i, \forall j < i, v_j = u_j, v_i \neq u_i, (v_i, u_i) \in E(G)$.

2. Vertices $(v_1, \cdots, v_m)$ and $(u_1, \cdots, u_m)$ are adjacent iff any pair of (different) components are adjacent in G, i.e., $\exists i, (v_i, u_i) \in E(G)$.

The claims are proven for both understandings, with the **same arguments**.

(a) G has an independent set of size $k \iff G^m$ has an independent set of size k^m.

$\Rightarrow$: Suppose $I = \{w_1, \cdots, w_k\}$ is an independent set of G. Then I^m is an independent set of G^m, since $\forall v_i, u_i \in I$, v_i and u_i are not adjacent in G.

$\Leftarrow$: Argue by contradiction. The arguments also provide an algorithm for part **(b)**. Suppose $I' \subset V(G)^m, |I'| = k^m$ is an independent set of G^m, but the maximum independent set of G has size at most $k-1$. Look at the first components of the vertices in I'. They only have at most $k-1$ distinct values, otherwise these $\geq k$ values form an independent set of G. So there exists $I'_1 \subseteq I'$ of size at least $k^{m-1}+1$ whose elements have the same first components. Use the same argument, it is easy to see that there exists $I'_i \subseteq I'_{i-1} \subseteq I'$ of size at least $k^{m-i}+1$ whose elements have the same first i components. Therefore, there exists $I'_m \subseteq I'$ of size at least 2 whose elements are all the same, meaning these two vertices are the same, i.e., $|I'| < k^m$, a contradiction.

- Given an independent set I' of G^m of size k, the algorithm scans the components of the elements of I' from the first to the last.

  1. $i \leftarrow 1, I'_0 \leftarrow I'$, FOUND = FALSE
  2. While not FOUND do the following:
     (a) I = the set of distinct values of the ith components of the vertices in I'_{i-1}
     (b) If $|I| \geq \lceil k^{1/m} \rceil$, FOUND = TRUE
     (c) Else find $v \in I'_{i-1}$ which has the maximum number of occurence as the ith components of the vertices in I'_{i-1}; find $I'_i \subseteq I'_{i-1}$ whose elements have v as their ith components.
     (d) i++
  3. Output I

  The proof for the $\Leftarrow$ direction of part **(a)** guarantees the correctness of the algorithm.

- Suppose A is a polytime c-approximation algorithm for IndependentSet. Suppose the algorithm defined in part **(b)** is B. Let $H(G)$ denote the optimal solution for G. Let $f(\cdot)$ be a function $f : \mathbb{R} \mapsto \mathbb{N}$. Define algorithms D as follows. On input (G, ε), construct $G^{f(\varepsilon)}$ in either way; output $B\left(A(G^{f(\varepsilon)})\right)$. We need to determine $f(\cdot)$ such that $C(G) \geq (1-\varepsilon)H(G)$. We have:

$$\begin{aligned} A(G^{f(\varepsilon)}) &\geq \frac{1}{c} H(G^{f(\varepsilon)}) \\ H(G^{f(\varepsilon)}) &= H(G)^{f(\varepsilon)} \end{aligned}$$

So

$$\begin{aligned} C(G) &= B\left(A(G^{f(\varepsilon)})\right) = \left\lceil \left(A(G^{f(\varepsilon)})\right)^{1/f(\varepsilon)} \right\rceil \\ &\geq \left\lceil \left(\frac{1}{c} H(G^{f(\varepsilon)})\right)^{1/f(\varepsilon)} \right\rceil = \left\lceil \left(\frac{1}{c}\right)^{1/f(\varepsilon)} \left(H(G)^{f(\varepsilon)}\right)^{1/f(\varepsilon)} \right\rceil \\ &\geq \left(\frac{1}{c}\right)^{1/f(\varepsilon)} H(G) \end{aligned}$$

For $C(G) \geq (1-\varepsilon)H(G)$, we only need $\left(\frac{1}{c}\right)^{1/f(\varepsilon)} \geq 1 - \varepsilon$, which can be achieved by defining

$$f(\varepsilon) = \left\lceil \frac{1}{\log_c \frac{1}{1-\varepsilon}} \right\rceil$$

.