CSC 373 Lecture 24

Message from department: please remind your students that applications for admission to our graduate program close in just one month on 8 December, and that a successful application is not something that can be put together at the last minute. Students can go to the DCS website for more information and links to the online application form (look for Information for Prospective Graduate Students). Advice for 3rd year: try for a USRA

Today

Finish up current discussion of local search

- The Max Independent Set (MIS) problem for k+1 claw free graphs.

- Perhaps start IP/LP rounding.

The MIS Problem

- Given *G = (V,E)*, the MIS problem is to find a maximum cardinality independent subset *V'* of *V*. In the weighted case, we have weights *w()* for vertices and the goal is to choose an independent set as to maximize the sum of weights.
- Even for the unweighted case, the problem is NP-hard to even approximate with a factor $n^{1-\epsilon}$ where n is the number of nodes.

MIS for special graphs

- Of course there are many graphs and graph classes for which efficient algorithms can provide optimal or good approximations.
- We have already seen that interval selection is the MIS problem for interval graphs. For the unweighted case, the EFT greedy algorithm is optimal and for the weighted case there is an optimal DP algorithm as well as an optimal "local ratio" algorithm. Informally, the latter algorithm uses the same EFT ordering of intervals but now first pushes intervals with positive "residual weight" onto a stack reducing the weight of later conflicting intervals. The stack is then popped to obtain a feasible solution.

k+1 claw free graphs

- A graph $G = (V,E)$ is k+1 claw free if it doesn't contain a $k+1$ claw, or equivalently if the neighbourhood $Nbhd(v)$ of every vertex v has at most k independent vertices.
- The intersection graphs of axis parallel translates of a rectangle are 5 claw free, the interval graphs for fixed length intervals are 3 claw free, unit disk graphs are 6 claw free, k-set packing induces a k+1 claw free graph, line graphs are 3 claw free.
- The MIS problem for 3 claw free graphs (also called claw free) can be solved in poly time (not obvious) and the MIS problem is NP hard for when $k >= 4$.
- There is a "natural" k-approximation greedy algorithm for weighted MIS (WMIS) on $k+1$ claw free graphs. Suggestions? A "simple" charging argument will prove the bound.

MIS for *k+1* claw free graphs

- We consider the following local neighbourhood: A set $S' = (S-U+T)$ is a t-neighbour of S if U has size $s < t$ and T is disjoint from S and of size at least s.
- Initialize S to some independent set
  While there is a t-neighbour S' of S such that S'
  is independent
  $S := S'$
  End While
- When $t = 2$, this is a $(k+1)/2$ approximation alg. Larger neighbourhoods can obtain $k/2$+epsilon

The analysis

- We will use a counting argument (essentially a charging argument). We let A (for alg) be a local opt and B an arbitrary indep. set (eg the OPT) . We want to show that $2|B| <= (k+1)\ |A|$.
- We can charge (match) all the nodes in the intersection of A and B to themselves. So now we will restrict attention to nodes A' and B' not in the intersection. The induced graph is now bipartite.
- Let B_i be the nodes in B' with i neighbours in A'. Let A_1 be the nodes in A' with a neighbour in B_1.

The analysis continued

- We let $b_i = |B_i|$, $a_1 = |A_1|$, $a = |A'|$, $b = |B|$.
- Using the $k+1$ clawfree property we have $b_1 + 2b_2 + 3b_3 <= k*a$

The best way to see this inequality is to count edges. Counting edges leaving A we have at most $k*a$ edges. Counting edges leaving B we have the e = sum of $i*b_i$ edges. Letting B'_i be the union of the B_i for i >=3 and letting $b'_i = |B'_i|$, we have $b_1 + 2b_2 + 3b'_3 <= e$

- By definition $b_1 >= a_1$ and since S is a local optimum $b_1 = a_1 <=a$
- Adding the inequalities, $2b = 2b_1 + 2b_2 +2b'3 <= b_1 + e = 2b_1 + 2b_2 +3b'_3 <= k*a + a_1 <=(k+1) a$

The weighted case

- The weighted case is a more involved story. A very basic local search becomes a *k*-approx (proven using a simple charging argument) matching the greedy approximation. Namely the alg tries to bring in one node not in S if its weight exceeds the total weight of its neighbouring nodes which will have to be removed.
- But another oblivious local search obtains a *2(k+1)/3* approximation if the initial solution is chosen greedily.
- There is also a non-oblivious local search that obtains a *(k+1)/2 + epsilon* approximation using the square of weights for defining a potential function. Motivation?