Lecture 10, June 2, 2004

Uri Feige

1 Perfect graphs and the theta function of Lovasz

1.1 Perfect graphs

Let $\omega(G)$, $\alpha(G)$, $\chi(G)$, $\bar{\chi}(G)$ denote the maximum clique size, maximum independent set size, chromatic number, and clique cover number of graph G, respectively. They are related as follows (here $\bar{G}$ denotes the edge complement of graph G):

$\omega(G) = \alpha(\bar{G})$
$\chi(G) = \bar{\chi}(\bar{G})$
$\chi(G) \geq \omega(G)$
$\chi(G) \cdot \alpha(G) \geq n$

A graph G is *perfect* if $\omega(G') = \chi(G')$ for every vertex induced subgraph G' of G.

We give examples of several families of graphs that are perfect, and note in passing that the complements of these families are also perfect. In other words, also $\alpha(G') = \bar{\chi}(G')$ hold for every vertex induced subgraph.

Bipartite graphs. Every vertex induced subgraph of a bipartite graph is also bipartite. For every bipartite graph G with at least one edge, $\omega(G) = \chi(G) = 2$. Note also that $\alpha(G) = \bar{\chi}(G)$ (where the latter is the minimum number of edges needed to cover all vertices). To see this, consider a maximum matching. As we have already seen, each edge covers one vertex of minimum vertex cover and one in the maximum independent set. The rest of the vertices in the maximum independent set can be covered one by one, either by unmatched edges, or by the vertices themselves (which are cliques of size 1).

Line graphs of bipartite graphs. Let G_l be the line graph of the bipartite graph G. Then $\omega(G_l)$ is the maximum degree in G, and $\chi(G_l)$ is the chromatic index of G, which are known to be equal for bipartite graphs. We also have that $\alpha(G_l)$ is the size of the maximum matching in G, and $\bar{\chi}(G_l)$ is the size of the minimum vertex cover, which again are equal for bipartite graphs.

Interval graphs. We will not elaborate on them, as they are a special case of complements of comparability graphs.

Comparibility graphs. Given a partial order on n elements, we treat each element as a vertex, and draw an edge between vertices i and j if either $i \leq j$ or $j \leq i$. Note that an order relation is transitive. A vertex induced subgraph of a comparibility graph is a comparibility graph. A clique in a comparibility graph is a chain in the partial order. An independent set is an antichain. It is easy to show that $\omega(G) = \chi(G)$. This can be proved by induction on n: remove the maximum element from every maximum length chain; the maximum chain length decreased by one, and the set of all elements removed is an antichain.

Perhaps more interestingly, we have Dilworth's theorem.

Theorem 1 *In every partial order, the cardinality of the largest antichain is equal to the minimum number of chains that cover all elements.*

Proof: Consider an arbitrary maximal (by inclusion) chain C. If every maximum antichain A intersects C then by removing C, the cardinality of the largest antichain decreases by 1, and we can proceed by induction. So let us assume that A and C do not intersect. The maximal element in C is necessarily larger than some element in A, because otherwise we can either extend C or extend A. Likewise, the minimal element in C is necessarily smaller than some element in A. Consider all elements except for those that are smaller than some element in A. They can be covered by $|A|$ chains (by induction). The same holds for all elements except those that are larger than some element in A. Using A as the gluing points between these two sets of chains, all elements are covered. $\square$

It follows that the complement of a comparibility graph is perfect.

Chordal graphs. A graph is chordal if every cycle of length at least 4 has a chord. These graphs and their complements are perfect.

Lovasz proved the so called *weak perfect graph conjecture*, that that the complement of a perfect graph is perfect. Hence for perfect graphs, $\alpha(G') = \bar{\chi}(G')$ for every vertex induced subgraph G' of G.

A chordless odd cycle of length at least five is not a perfect graph, and neither is its complement. The strong perfect graph conjecture of Berge says that every nonperfect graph must contain one of the above as a vertex induced subgraph. The conjecture was proved by Chunovsky and Seymour. Insights from their proof helped in designing a polynomial time algorithm that tests whether a graph is perfect or not.

1.2 A linear program

Let f be a function on graphs with the following "sandwich" property: $\alpha(G) \leq f(G) \leq \bar{\chi}(G)$. If f is computable in polynomial time, it can be used to compute $\omega(G)$, $\alpha(G)$, $\chi(G)$, $\bar{\chi}(G)$ for perfect graphs.

Consider the following primal LP.

maximize $\sum_i x_i$

subject to

$\sum_{i \in C_j} x_i \leq 1$ for every clique C_j,

$x_i \geq 0$ for every vertex i.

Clearly, the above LP gives an upper bound on $\alpha(G)$. Its dual is:

minimize $\sum_j y_j$

subject to

$\sum_{\{j | \text{clique } C_j \text{contains vertex } i\}} y_j \geq 1$ for every vertex i,

$y_j \geq 0$ for every clique C_j.

Clearly, the dual gives a lower bound on $\bar{\chi}(G)$.

Hence the above LP has the desired sandwich property. Unfortunately, to solve it we seem to need a procedure for maximum weight clique in a vertex weighted graph. This problem is NP-hard. Moreover, it can be shown that for general graphs, finding the optimal value of this LP (the so called *fractional clique cover number*) is NP-hard.

For perfect graphs, both the LP and its dual necessarily have integer optimal solutions. As we shall later see, for perfect graphs the LP can be solved in polynomial time.

1.3 The theta function of Lovasz

The ϑ function of Lovasz has the sandwich property, and can be computed with arbitrary precision in polynomial time (via a semidefinite program).

There are many alternative definitions to the ϑ function. Here is a definition of the ϑ function as a minimization problem.

Associate with each vertex a unit vector in R^n. The two vectors associated with vertices that are nonadjacent must have inner product $-1/(k-1)$. Under the above restrictions, minimize k.

Observe that given a vector arrangement with a particular value of k, we can always transform it to a vector arrangement with a larger value of k, by adding another coordinate in which all vectors have the same value, and rescaling the vectors to have norm 1.

Theorem 2 *For every graph G, $\alpha(G) \leq \vartheta(G) \leq \bar{\chi}(G)$.*

Proof: Assume that G can be covered by k disjoint cliques. With each clique associate a unit vector, and make the unit vectors the vertices of a perfect $(k-1)$ dimensional simplex. All inner products are $-1/(k-1)$. This satisfies the constraints, and gives k as an upper bound on the minimization version of $\vartheta(G)$. (An algebraic discription of these vectors rather than geometric is as follows. There are $\binom{k}{2}$ coordinates, indexed by $1 \leq i < j \leq k$. Vector i has values $1/\sqrt{k-1}$ in coordinates ij with $j > i$, values $-1/\sqrt{k-1}$ in coordinates ji with $j < i$, and 0 elsewhere. The inner products are as claimed. The dimension can be reduced to k by a change of basis, because there are only k vectors.)

Let $\{v_1, \ldots, v_k\}$ be the largest independent set in G. Associate a unit vector x_i with each vertex v_i of this independend set, and let $x = \sum_{i=1}^{k} x_i$. We have

$$0 \leq x^T x = (\sum x_i^T)(\sum x_i) = k + 2\sum_{i<j} x_i^T x_j$$

By averaging over $k(k-1)$ terms, there must be a pair i, j with $x_i^T y_j \geq -1/(k-1)$. Hence k is a lower bound on $\vartheta(G)$. $\Box$

Note that $\vartheta(G)$ might be irrational (e.g., for the pentagon we have $\vartheta(C_5) = \sqrt{5}$), and hence we should not expect to compute it exactly.

Theorem 3 *For every graph G and every $\epsilon > 0$, $\vartheta(G)$ can be computed in polynomial time up to a precision of ϵ.*

Proof: To see if $\vartheta(G) \leq k$, set up the following constraints over the vectors x_i.
$x_i^T x_i = 1$,
$x_i^T x_j = -1/(k-1)$ for every $(i,j) \notin E$.

This can equivalently be represented as the following positive semidefinite program. We associate a variable y_{ij} with every inner product $x_i^T x_j$, and set up the following constraints.

- $y_{ii} = 1$ for every i,
- $y_{ij} = -1/(k-1)$ for every $(i,j) \notin E$,
- The n by n matrix $Y = \{y_{ij}\}$ is symmetric and positive semidefinite.

By adding some small slackness to the first two sets of constraints, we can solve the SDP using the ellipsoid algorithm. If we slightly overrestrict the semidefine requirement (by requiring strictly positive definiteness), we do not significantly change the optimal value of k, and we can jump to a solution where all other constraints are satisfied with equality, while maintaining positive semidefiniteness. A Gram decomposition of the matrix Y gives the desirable vectors x_i. $\square$

In perfect graphs, we can find an independent set of size $\alpha(G)$ as follows. Solve the SDP to find the value of $\alpha(G)$. Iteratively remove a vertices from the graph, each time recomputing the value of ϑ. Those vertices that caused a drop in ϑ form a maximum independent set. Note that we used the fact that every subgraph of a perfect graph is perfect.

Note that the above procedure will for every graph either find a maximum independent set, or certify that the graph in not perfect.

We shall also need to consider cliques in vertex weighted graphs. Here vertex i has weight w_i, and one wishes to find a clique of maximum weight. In general, algorithms for max clique that work in unweighted graphs work also for weighted graphs, and this applies to the ϑ function as well. A generic reduction (that can be used when we can afford tiny modifications to the weight of cliques) is as follows. Scale all weights so that the maximum weight vertex was weight 1. Pick some sufficiently small ϵ (such as $\epsilon = 1/n^2$), and round all vertex weights to the nearest mutiple of ϵ. Now multiply all weights by $1/\epsilon$, and they become integer. Replace every vertex i of integer weight $w_i > 1$ by a clique of size w_i (every vertex in the clique will have weight 1) connected to all neighbors of vertex i. We obtain an unweighted graph with a polynomial number of vertices, and a straightforward correspondence between weighted cliques in the original graph and cliques in the new graph. It is important for our applicatoin that if the original graph was perfect, then so is the resulting graph. And indeed, this is the case, for the following reason. Every cycle of length at least 5 that includes two representatives of the same original vertex i must have a chord. The same applies also to the complement of the graph.

To find a minimum clique-cover (a clique-cover is a set of cliques that cover all vertices) of a perfect graph, use the ellipsoid algorithm to solve the primal LP of Section 1.2. At each iteration one needs to see if there is a violated clique constraint, and this can be done by finding a maximum weight clique with the help of the theta function. When an optimal solution is reached, consider the list of clique constraints that were used. The solution is optimal also with respect to these cliques alone. Hence call them the *essential* cliques, and consider a modified primal LP that has only the clique constraints that correspond to the essential cliques. Now go to the dual program of the modified primal. The essential cliques are the basic variables of the dual. Solve the dual, to obtain a "fractional" clique-cover. By LP duality, the value of the dual is equal of that of the primal, which is equal to the size of the maximum independent set (because the graph is perfect). Take the clique with largest fractional weight. It must intersect every maximum size independent set, because the other independent sets did not contribute enough fractional weight to cover all vertices of the maximum independent set. Remove this clique, and repeat, eventially obtaining a clique-cover of size equal to that of the maximum independent set.