Lecture 12

Uriel Feige

May 22, 2008

1 Max cut in planar graphs

After adding the triangle constraints to the GW SDP, the value of the SDP is equal to the value of the maximum cut on planar graphs. We shall not prove this fact. Instead, we shall explain an "easy" way of seeing that max-cut can be solved in polynomial time on planar graphs.

Let G be a weighted planar graph on which we wish to solve the max-cut problem. We may assume without loss of generality that it is connected (otherwise, solve the problem on each connected component separately). We may also assume that no cut contains only one edge (otherwise, solve the problem separately on the two parts of the graph, and arrange the sides such that this edge is cut). Consider an arbitrary planar embedding of the graph. The dual graph H has the faces of this embedding as its vertices, and every edge separating two faces in G becomes an edge connecting the respective vertices in H. By our above assumptions, H does not have self loops, but it might have parallel edges (iff G has a cut with only two edges). Observe that every cut in G corresponds in a natural way to a collection of cycles in H. Likewise, every collection of cycles in H (or equivalently, by the connection to Euler tours, any collection of edges such that degree of every vertex is even) corresponds to a cut in G. (To see the two coloring of the vertices of G, or equivalently of the faces of H, implied by the cycles, color a face as *even* or *odd* depending on the number of edges of the Eulerian subgraph that need to be crossed in order to escape to infinity.) Hence in the special case of planar graphs, the max cut problem in G is reduced to finding a maximal Eulerian subgraph of the dual graph H. Even though H is planar as well, we shall not need this fact in the rest of the discussion.

Let us digress a bit to recall the *chinese postman problem.* Given a graph H, one is to find a closed tour of minimum length that crosses every edge at least one. (Every edge is a street, and the postman needs to deliver mail to the residents of all streets.) Despite the similarity with the NP-hard travelling salesperson problem (in which every vertex needs to be visited at least once), the chinese postman problem can be solved in polynomial time. This will follow from our discussion below, and by noting that the minimum solution for the chinese postman problem is precisely the Euler tour that results by taking the maximum Eulerian subgraph plus two copies of each one of the remaining edges.

We now return to the maximum Eulerian subgraph problem. It can be formulated as a minimum T-join problem. Let T be the set of nodes (terminals) of odd degree in H. A T-join is a subgraph in which every vertex of T has odd degree and every other vertex has even degree. Clearly, removing the minimum T-join from H leaves one with the maximum Eulerian subgraph.

Observe that the size of T is even (since the sum of degrees in a graph is even). We now claim that every minimum T-join must be the union of $|T|/2$ edge disjoint paths, where each such path connects two distinct vertices in T via a shortest path. To prove the claim, consider at arbitrary T-join. By repeatedly removing cycles if they exist, what remains is a collection of $|T|/2$ paths (because there are $|T|$ vertices of odd degree). If any of these paths is not a shortest path between its respective endpoints, then it can be replaced by a shorter path, resulting in a T-join of smaller weight.

To find the minimum weight T-join, we can set up a weighted matching problem on the vertices of T, where the weight of the edge (u, v) is the length of the shortest path between u and v in H. Since shortest path can be solved in polynomial time, the reduction is polynomial time. Now we may use the (nonobvious) fact that maximum weight matching can be solved in polynomial time to conclude that max cut can be solved in polynomial time on planar graphs.

2 Coloring 3-colorable graphs

Recall that every graph of maximum degree Δ can be colored by $\Delta + 1$ colors. Karger, Motwani and Sudan [1] show that if Δ is large and the graph happens to be colorable with much fewer colors (say, 3), then there is a polynomial time algorithm that colors it with $o(\Delta)$ colors. Their paper is very readable, and here we sketch only some of the details as specialized to 3-colorable graphs.

Recall the SDP for the theta function as presented at an earlier lecture. We show how it can be used to obtain approximation algorithms for coloring.

For a 3-colorable graph with n vertices and m edges, using the SDP we obtain a unit vector solution in which for every edge (i, j), $\langle v_i, v_j \rangle = -1/2$.

To round it, we pick a random vector r as follows. Each coordinate is distributed according to the normal distribution $\phi(x) = \frac{1}{\sqrt{2\pi}} e^{-x^2/2}$ with mean 0 and variance 1. This is known to give a spherically symmetric distribution where each direction is also distributed according to the normal distribution (and orthogonal directions have orthogonal distributions).

Let $N(x) = \int_x^\infty \phi(y) d(y)$ denote the tail of the normal distribution. It is known that

$$\phi(x)(\frac{1}{x} - \frac{1}{x^3}) \leq N(x) \leq \phi(x)\frac{1}{x}$$

Let c be a value to be optimized later. The rounding technique first selects a set S of vertices that contains all vertices i with $\langle v_i, r \rangle \geq c$.

We shall use heavily the linearity of expectation. The expected number of vertices in S is $nN(c)$. To compute the expected number of edges in S, let (i,j) be an edge. We need $\langle v_i, r\rangle \geq c$ and $\langle v_j, r\rangle \geq c$ implying $\langle v_i + v_j, r\rangle \geq 2c$. Note that $v_i + v_j$ is also a unit vector because $(v_i + v_i)^2 = 1 + 2(-\frac{1}{2}) + 1 = 1$. Hence the expected number of edges is $mN(2c)$. We shall pick c such that $nN(c) - mN(2c) \geq nN(c)/2$. (Then, in expectation removing one endpoint of every edge is S, at least half the expected vertices remain and form an independent set.) Let $d = 2m/n$ denote the average degree. Hence we need $n\phi(c)(\frac{1}{c} - \frac{1}{c^3}) \geq dn\phi(2c)\frac{1}{2c}$. When d is a sufficiently large constant, this simplifies to a sufficient condition of $e^{3c^2/2} \geq d$. At this point the expected number of vertices in S is $\tilde{\Omega}(n/d^{1/3})$ (where the $\tilde{\Omega}$ notation hides some logarithmic terms that we do not care to optimize here). By repeatedly removing independent sets in this manner, we obtain a coloring with $\tilde{O}(\Delta^{1/3})$ colors. (The need to replace the average degree d by the maximum degree Δ is because d is not preserved as vertices are removed from the graph. Alternatively, rather than Δ, one can take the maximum average degree of a vertex induced subgraph, which can be computed in polynomial time using flow techniques.)

There is no better approximation known for 3-coloring as a function of Δ. As a function of n, there have been various improvements to this approximation ratio. For example, by using the observation by Wigderson that if a 3-colorable graph has a vertex of degree Δ one can find among its neighbors an independent set of size $\Delta/2$, one can color 3-colorable graph by $n^{1/4}$ colors. Further improvements are known.

Adapting the KMS analysis to k-colorable graphs gives a coloring using $\tilde{O}(\Delta^{1-\frac{2}{k}})$ colors.

References

[1] David R. Karger, Rajeev Motwani, Madhu Sudan: Approximate Graph Coloring by Semidefinite Programming. J. ACM 45(2): 246-265 (1998).