Randomized Rounding for Semidefinite Programs – Variations on the MAX CUT Example

Uriel Feige

Weizmann Institute, Rehovot 76100, Israel

Abstract. MAX CUT is the problem of partitioning the vertices of a graph into two sets, maximizing the number of edges joining these sets. Goemans and Williamson gave an algorithm that approximates MAX CUT within a ratio of 0.87856. Their algorithm first uses a semidefinite programming relaxation of MAX CUT that embeds the vertices of the graph on the surface of an n dimensional sphere, and then cuts the sphere in two at random.
In this survey we shall review several variations of this algorithm which offer improved approximation ratios for some special families of instances of MAX CUT, as well as for problems related to MAX CUT.

1 Introduction

This survey covers an area of current active research. Hence, there is danger, or rather hope, that the survey will become outdated in the near future. The level of presentation will be kept informal. More details can be found in the references. Results are presented in a logical order that does not always correspond to the historical order in which they were derived.

The scope of the survey is limited to MAX CUT and to strongly related other problems (such as MAX BISECTION). Many of the recent approximation algorithms based on semidefinite algorithms are not included (such as those for COLORING [10] and for 3SAT [12]). Results in which the author was involved are perhaps over-represented in this survey, but hopefully, not in bad taste.

2 The Algorithm of Goemans and Williamson

For a graph $G(V, E)$ with $|V| = n$ and $|E| = m$, MAX CUT is the problem of partitioning V into two sets, such that the number of edges connecting the two sets is maximized. This problem is NP-hard to approximate within ratios better than 16/17 [9]. Partitioning the vertices into two sets at random gives a cut whose expected number of edges is $m/2$, trivially giving an approximation algorithm with expected ratio at least 1/2. For many years, nothing substantially better was known. In a major breakthrough, Goemans and Williamson [8] gave an algorithm with approximation ratio of 0.87856. For completeness, we review their well known algorithm, which we call algorithm GW.

MAX CUT can be formulated as an integer quadratic program. With each vertex i we associate a variable $x_i \in \{-1, +1\}$, where -1 and $+1$ can be viewed as the two sides of the cut. With an edge $(i, j) \in E$ we associate the expression $\frac{1-x_i x_j}{2}$ which evaluates to 0 if its endpoints are on the same side of the cut, and to 1 if its endpoints are on different sides of the cut.

The integer quadratic program for MAX CUT:
Maximize: $\sum_{(i,j)\in E} \frac{1-x_i x_j}{2}$
Subject to: $x_i \in \{-1, +1\}$, for every $1 \le i \le n$.

This integer quadratic program is relaxed by replacing the x_i by unit vectors v_i in an n-dimensional space (the x_i can be viewed as unit vectors in a 1-dimensional space). The product $x_i x_j$ is replaced by an inner product $v_i v_j$. Geometrically, this corresponds to embedding the vertices of G on a unit n-dimensional sphere S_n, while trying to keep the images of vertices that are adjacent in G far apart on the sphere.

The geometric program for MAX CUT:
Maximize: $\sum_{(i,j)\in E} \frac{1-v_i v_j}{2}$
Subject to: $v_i \in S_n$, for every $1 \le i \le n$.

The geometric program is equivalent to a semidefinite program in which the variables y_{ij} are the inner products $v_i v_j$, and the n by n matrix Y whose i, j entry is y_{ij} is constrained to be positive semidefinite (i.e., the matrix of inner products of n vectors). The constraint $v_i \in S_n$ is equivalent to $v_i v_i = 1$, which gives the constraint $y_{ii} = 1$.

The semidefinite program for MAX CUT:
Maximize: $\sum_{(i,j)\in E} \frac{1-y_{ij}}{2}$
Subject to: $y_{ii} = 1$, for every $1 \le i \le n$,
and the matrix $Y = (y_{ij})$ is positive semidefinite.

This semidefinite program can be solved up to arbitrary precision in polynomial time, and thereafter a set of vectors v_i maximizing the geometric program (up to arbitrary precision) can be extracted from the matrix Y (for more details, see [8]).

The value of the objective function of the geometric problem is at least that of the MAX CUT problem, as any ± 1 solution for the integer quadratic problem is also a solution of the geometric problem, with the same value for the objective function.

One approach to convert a solution of the geometric program to a feasible cut in the graph is to partition the sphere S_n into two halves by passing a hyperplane through the origin of the sphere, and labeling vertices on one half by -1 and on the other half by $+1$. The choice of hyperplane may affect the quality of solution that is obtained. Surprisingly, a random hyperplane is expected to give a cut that is not far from optimal.

Consider an arbitrary edge (i, j). Its contribution to the value of the objective function is $\frac{1-v_i vj}{2}$. The probability that it is cut by a random hyperplane is directly proportional to the angle between v_i and v_j, and can be shown to be exactly $\frac{\cos^{-1}(v_i v_j)}{\pi}$. Hence the ratio between the expected contribution of the edge (i, j) to the final cut and its contribution to the objective function of the geometric program is $\frac{2\cos^{-1}(v_i v_j)}{\pi(1-v_i v_j)}$. This ratio is minimized when the angle between v_i and v_i is $\theta \simeq 2.33$, giving a ratio of $\alpha \simeq 0.87856$. By linearity of expectation, the expected number of edges cut by the random hyperplane is at least α times the value of geometric program, giving an α approximation for MAX CUT.

We remark that a random hyperplane can be chosen algorithmically by choosing a random unit vector r, which implicitly defines the hyperplane $\{x|xr = 0\}$. See details in [8].

2.1 Outline of Survey

The algorithm of Goemans and Williamson, and variations of it, were applied to many other problems, some well known examples being MAX 2SAT [8, 3], MAX 3SAT [12], MIN COLORING [10] and MAX CLIQUE [1].

The work reported in this survey is partly motivated by the belief that algorithm GW does not exploit the full power of semidefinite programming in the context of approximating MAX CUT.

Research goal: Improve the approximation ratio of MAX CUT beyond $\alpha \simeq 0.87856$.

In the following sections, we shall survey several approaches that try to improve over algorithm GW.

In Section 3 we add constraints to the semidefinite program so as to obtain better embeddings of the graph on the sphere. In Section 4 we describe limitations of the random hyperplane rounding technique, and suggest an alternative "best" hyperplane rounding technique. In Section 5 we investigate rounding techniques that rearrange the points on the sphere prior to cutting. In Section 6 we describe approaches that rearrange the vertices after cutting.

This survey is limited to approaches that remain within the general framework of the GW algorithm.

3 Improving the Embedding

The GW algorithm starts with an embedding of the graph on a sphere. The *value* of this embedding is the value of the objective function of the geometric program. The quality of the embedding can be measured in terms of the so called *integrality ratio*: the ratio between the size of the optimal cut in the graph and the value of the geometric embedding. (We define here the integrality ratio as a number smaller than 1. For this reason we avoid the more common name *integrality gap*, which is usually defined as the inverse of our integrality ratio.) This measure of quality takes the view that we are trying to estimate the size of the maximum cut in the graph, rather than actually find this cut. We may output the value of the embedding as our estimate, and then the error in the estimation is bounded by the integrality ratio.

Goemans and Williamson show that the integrality ratio for their embedding may be essentially as low as α. As a simple example, let G be a triangle (a 3-cycle). Arranging the vertices uniformly on a circle (with angle of $2\pi/3$ between every two vectors) gives an embedding with value 9/4, whereas the maximum cut size is 2. This gives an integrality ratio of $8/9 \simeq 0.888$. For tighter examples, see [8].

To improve the value of the embedding on the sphere, one may add more constraints to the semidefinite program. In doing so, one is guided by two requirements:

1. The constraints need to be satisfied by the true optimal solution (in which the y_{ij} correspond to products of ± 1 variables).
2. The resulting program needs to be solvable in polynomial time (up to arbitrary precision).

Feige and Goemans [3] analyse the effect of adding *triangle constraints* of the form $y_{ij}+y_{jk}+y_{ki} \geq -1$ and $y_{ij}-y_{jk}-y_{ki} \geq -1$, for every i, j, k. Geometrically, these constraints forbid some embeddings on the sphere. In particular, if three vectors v_i, v_j, v_k lie in the same plane (including the origin), it now must be the case that either two of them are identical, or antipodal. The 3-cycle graph no longer serves as an example for a graph with bad integrality ratio. Moreover, it can be shown that for every planar graph, the value of the geometric program is equal to that of the maximum cut.

Feige and Goemans were unable to show that the more constrained semidefinite relaxation leads to an approximation algorithm with improved approximation ratio for MAX CUT (though they were able to show this for related problems such as MAX 2SAT).

Open question: Does addition of the triangle constraints improve the integrality ratio of the geometric embedding for MAX CUT?

Feige and Goemans also discuss additional constraints that can be added. Lovasz and Schriver [13] describe a systematic way of adding constraints to semidefinite relaxations. The above open question extends to all such formulations.

4 Improving the Rounding Technique

Goemans and Williamson use the random hyperplane rounding technique. The analysis or the approximation ratio compares the expected number of edges in the final cut to the value of the geometric embedding. We remark that for most graphs that have maximum cuts well above $m/2$, the random hyperplane rounding technique will actually produce the optimal cut. This is implicit in [2, 5].

Karloff [11] studies the limitations of this approach. He considers a family of graphs in which individual graphs have the following properties:

1. The maximum cut in the graph is not unique. There are $k = \Omega(\log n)$ different maximum cuts in the graph. (The graph is very "symmetric" – it is both vertex transitive and edge transitive.)

2. The value of the geometric program (and the semidefinite program) for this graph is exactly equal to the size of the maximum cut. Hence the integrality ratio is 1.
3. The vertices can be embedded on a sphere as follows. Each vertex is a vector in $\{+1, -1\}^k$ (and normalized by $1/\sqrt{k}$), where coordinate j of vector i is ± 1 depending on the side on which vertex i lies in the jth optimal cut. It follows that the value of this embedding is equal to the size of the maximum cut.
4. The sides of each maximum cut are labeled ± 1 in such a way that for the above embedding, the angle between the vectors of any two adjacent vertices is (arbitrarily close to) θ, where θ is the worst angle for the GW rounding technique. Hence the analysis of the random hyperplane rounding technique only gives an approximation ratio of α for the above graph and associated embedding.

The embedding described above, derived as a combination of legal cuts, satisfies all constraints discussed in Section 3. Hence we are led to conclude that if one wishes to get an approximation ratio better than α, one needs a rounding technique different than that of Goemans and Williamson.

For Karloff's graph and embedding, each of the k maximum cuts can be derived by using a hyperplane whose normal is a vector in the direction of the respective coordinate. Hence a rounding technique that uses the *best* hyperplane (the one that cuts the most edges) rather than a random one would find the maximum cut.

Open question: Design examples that show a large gap between the value of the geometric embedding and the cut obtained by the best hyperplane.

The above open question can serve as an intermediate step towards analysing the integrality ratio.

Remark: The best hyperplane can be approximated in polynomial time in the following sense. The dimension of the embedding can be reduced by projecting the sphere on a random d-dimensional subspace. When d is a large enough constant, the vast majority of distances are preserved up to a small distortion, implying the same for angles. To avoid degeneracies, perturb the location of each point by a small random displacement. The value of the objective function hardly changes by this dimension reduction and perturbation (the change becomes negligible the larger d is). Now each relevant hyperplane is supported by d points, allowing us to enumerate all hyperplanes in time n^d.

5 Rotations

In some cases, it is possible to improve the results of the random hyperplane rounding technique by first rearranging the vectors v_i on the sphere. This modifies the geometric embedding, making it suboptimal with respect to the objective function. However, this suboptimal solution is easier to round with the random hyperplane technique.

Feige and Goemans [3] suggested to use rotations in cases where the sphere has a distinct "north pole" (and "south pole"). Rotating each vector somewhat towards its nearest pole prior to cutting the sphere with a random hyperplane can lead to improved solutions. The usefulness of this approach was demonstrated for problems such as MAX 2SAT, where there is a unique additional vector v_0 that is interpreted as the $+1$ direction and can serve as the north pole. It is not clear whether a similar idea can be applied for MAX CUT, due to a lack of a natural candidate direction that can serve as the north pole of the sphere.

Zwick [15] has used a notion of *outward rotations* for several problems. For MAX CUT, Zwick observes that there are two "bad" angles for which the random hyperplane fails to give an expectation above α. One is the angle θ mentioned in Section 2. The other is the trivial angle 0, for which the contribution to the value of the geometric program is 0, and so is the contribution to the cut produced by a hyperplane. Hence worst case instances for the GW algorithm may have an arbitrary mixture of both types of angles for pairs of vertices connected by edges. In the extreme case, where all angles are 0 (though this would never be the optimal geometric embedding) it is clear that a random hyperplane would not cut any edge, whereas ignoring the geometric embedding and giving the vertices ± 1 values independently at random is expected to cut roughly half the edges. This latter rounding technique is equivalent to first embedding the vertices as mutually orthogonal unit vectors, and then cutting with a random hyperplane. *Outward rotation* is a technique of averaging between the two embeddings: the optimal geometric embedding on one set of coordinates and the mutually orthogonal embedding on another set of coordinates. It can be used in order to obtain approximation ratios better than α whenever a substantial fraction of the edges have angles 0, showing that essentially the only case when the geometric embedding (perhaps) fails to have integrality ratio better than α is when all edges have angle θ.

6 Modifying the Cut

The random hyperplane rounding technique produces a cut in the graph. This cut may later be modified to produce the final cut. Below we give two representative examples.

Modifying the cut to get a feasible solution. MAX BISECTION is the problem of partitioning the vertices into two *equal* size sets while maximizing the number of edges in the cut. Rounding the geometric embedding via a random hyperplane produces a cut for which the expected number of vertices on each side is $n/2$, but the variance may be very large. Hence, this cut may not be a feasible solution to the problem. Frieze and Jerrum [7] analysed a greedy algorithm that modifies the initial cut by moving vertices from the larger side to the smaller one until both sides are of equal size. As moving vertices from one side to the other may decrease the number of edges in the cut, it is necessary to have an estimate of the expected number of vertices that need to be moved. Such an estimate can be derived if we add a constraint such as $\sum v_i v_j = 0$ to the geometric embedding, which is satisfied if exactly half the vectors are $+1$ and half of them -1. Frieze and Jerrum used this approach to obtain an approximation ratio of roughly 0.65. This was later improved by Ye [14], who used outward rotations prior to the random hyperplane cut. This has a negative effect of decreasing the expected number of edges in the initial cut, and a positive effect of decreasing the expected number of vertices that need to be moved (note that in the extreme case for outward rotation all vectors are orthogonal and then with high probability each side of the cut contains $n/2 \pm O(\sqrt{n})$ vertices). Trading off these two effects, Ye achieves an approximation ratio of 0.699.

Other problems in which a graph needs to be cut into two parts of prescribed sizes are studied in [6]. An interesting result there shows that when a graph has a *vertex cover* of size k, then one can find in polynomial time a set of k vertices that covers more than $0.8m$ edges. The analysis in [6] follows approaches similar to that of [7], and in some cases can be improved by using outward rotations.

Modifying the cut to get improved approximation ratios. Given a cut in a graph, a *misplaced* vertex is one that is on the same side as most of its neighbors. The number of edges cut can be increased by having misplaced vertices change sides. This local heuristic was used by Feige, Karpinski and Langberg [4] to obtain an approximation ratio significantly better than $\alpha \simeq 0.87856$ for MAX CUT on graphs with maximum degree 3 (the current version claims an approximation ratio of 0.914). Recall that the integrality ratio of the geometric embedding is as bad as α only if all edges have angle θ. Assume such a geometric embedding, and moreover, assume that the triangle constraints mentioned in Section 3 are satisfied. The basic observation in [4] is that in this case, if we consider an arbitrary vertex and two of its neighbors, there is constant probability that all three vertices end up on the same side of a random hyperplane. Such a vertex of degree at most 3 is necessarily misplaced. This gives $\Omega(n)$ expected misplaced vertices, and $\Omega(n)$ edges added to the cut by moving misplaced vertices. As the total number of edges is at most $3n/2$, this gives a significant improvement in the approximation ratio.

7 Conclusions

The algorithm of Goemans of Williamson for MAX CUT uses semidefinite programming to embed the vertices of the graph on a sphere, and then uses the geometry of the embedding to find a good cut in the graph. A similar approach was used for many other problems, some of which are mentioned in this survey. For almost all of these problems, the approximation ratio achieved by the rounding technique (e.g., via a random hyperplane) does not match the integrality ratio of the known negative examples. This indicates that there is still much room for research on the use of semidefinite programs in approximation algorithms.

Acknowledgements

Part of this work is supported by a Minerva grant, project number 8354 at the Weizmann Institute. This survey was written while the author was visiting Compaq Systems Research Center, Palo Alto, California.

References

1. Noga Alon and Nabil Kahale. "Approximating the independence number via the ϑ-function". *Math. Programming*.
2. Ravi Boppana. "Eigenvalues and graph bisection: an average-case analysis. In *Proceedings of the 28th Annual IEEE Symposium on Foundations of Computer Science*, 1997, 280–285.
3. Uriel Feige and Michel Goemans. "Approximating the value of two prover proof systems, with applications to MAX 2-SAT and MAX DICUT". In *Proceedings of third Israel Symposium on Theory of Computing and Systems*, 1995, 182–189.
4. Uriel Feige, Marek Karpinski and Michael Langberg. "MAX CUT on graphs of degree at most 3". *Manuscript*, 1999.
5. Uriel Feige and Joe Kilian. "Heuristics for semirandom graph models". *Manuscript*, May 1999. A preliminary version appeared in *Proceedings of the 39th Annual IEEE Symposium on Foundations of Computer Science*, 1998, 674–683.
6. Uriel Feige and Michael Langberg. "Approximation algorithms for maximization problems arising in graph partitioning". *Manuscript*, 1999.
7. Alan Frieze and Mark Jerrum. "Improved approximation algorithms for MAX k-CUT and MAX Bisection". *Algorithmica*, 18, 67–81, 1997.
8. Michel Goemans and David Williamson. "Improved approximation algorithms for maximum cut and satisfiability problems using semidefinite programming". *Journal of the ACM*, 42, 1115-1145, 1995.
9. Johan Hastad. "Some optimal inapproximability results". In *Proceedings of the 29th Annual ACM Symposium on Theory of Computing*, 1997, 1-10.
10. David Karger, Rajeev Motwani and Madhu Sudan. "Approximate graph coloring by semidefinite programming". *Journal of the ACM*, 45, 246–265, 1998.
11. Howard Karloff. "How good is the Goemans-Williamson MAX CUT algorithm?" In *Proceedings of the 28th Annual ACM Symposium on Theory of Computing*, 1996, 427–434.
12. Howard Karloff and Uri Zwick. "A 7/8 approximation algorithm for MAX 3SAT?" In *Proceedings of the 38th Annual IEEE Symposium on Foundations of Computer Science*, 1997, 406–415.
13. Laszlo Lovasz and Alexander Schrijver. "Cones of Matrices and set-functions and 0-1 optimization". *SIAM J. Optimization*, 1(2), 166-190, 1991.
14. Yinyu Ye. "A .699-approximation algorithm for Max-Bisection". *Manuscript*, March 1999.
15. Uri Zwick. "Outward rotations: a tool for rounding solutions of semidefinite programming relaxations, with applications to MAX CUT and other problems". In *Proceedings of the 31st Annual ACM Symposium on Theory of Computing*, 1999, 679–687.