Max Cut (1994; 1995; Goemans, Williamson)

Alantha Newman
Max-Planck-Institut für Informatik
http://www.mpi-inf.mpg.de/alantha

1 Index Terms

Graph partitioning, Approximation algorithms.

2 Synonyms

Maximum bipartite subgraph.

3 Problem Definition

Given an undirected edge-weighted graph, $G = (V, E)$, the maximum cut problem (Max-Cut) is to find a bipartition of the vertices that maximizes the weight of the edges crossing the partition. If the edge weights are non-negative, then this problem is equivalent to finding a maximum weight subset of the edges that forms a bipartite subgraph, i.e. the maximum bipartite subgraph problem. All results discussed in this article assume non-negative edge weights. Max-Cut is one of Karp's original NP-complete problems [Kar72]. In fact, it is NP-hard to approximate to within a factor better than $\frac{16}{17}$[TSSW00, Hås01].

For nearly twenty years, the best-known approximation factor for Max-Cut was half, which can be achieved by a very simple algorithm: Form a set S by placing each vertex in S with probability half. Since each edge crosses the cut $(S, V \setminus S)$ with probability half, the expected value of this cut is half the total edge weight. This implies that for any graph, there exists a cut with value at least half of the total edge weight. In 1976, Sahni and Gonzalez presented a deterministic half-approximation algorithm for Max-Cut, which is essentially a de-randomization of the aforementioned randomized algorithm [SG76]: Iterate through the vertices and form sets S and $\bar{S}$ by placing each vertex in the set that maximizes the weight of cut $(S, \bar{S})$ thus far. After each iteration of this process, the weight of this cut will be at least half of the weight of the edges with both endpoints in $S \cup \bar{S}$.

This simple half-approximation algorithm uses the fact that for any graph with non-negative edge weights, the total edge weight of a given graph is an upper bound on the value of its maximum cut. There exist classes of graphs for which a maximum cut is arbitrarily close to half the total edge weight, i.e. graphs for which this "trivial" upper bound can be close to twice the true value of an optimal solution. An example of such a class of graphs are complete graphs on n vertices, K_n.

In order to obtain an approximation factor better than half, one must be able to compute an upper bound on the value of a maximum cut that is better, i.e. smaller, than the trivial upper bound for such classes of graphs.

3.1 Linear Programming Relaxations

For many optimization (maximization) problems, linear programming has been shown to yield better (upper) bounds on the value of an optimal solution than can be obtained via combinatorial methods. There are several well-studied linear programming relaxations for MAX-CUT. For example, a classical integer program has a variable x_e for each edge and a constraint for each odd cycle, requiring that an odd cycle C contribute at most $|C|-1$ edges to an optimal solution.

$$\begin{array}{rcl} & \max & \sum_{e \in E} w_e x_e \\ \sum_{e \in C} x_e & \leq & |C| - 1 \quad \forall \text{ odd cycles } C \\ x_e & \in & \{0, 1\}. \end{array}$$

The last constraint can be relaxed so that each x_e is required to lie between 0 and 1, but need not be integral, i.e. $0 \leq x_e \leq 1$. Although this relaxation may have exponentially many constraints, there is a polynomial-time separation oracle (equivalent to finding a minimum weight odd-cycle), and thus, the relaxation can be solved in polynomial time [GLS81]. Another classical integer program contains a variable x_{ij} for each pair of vertices. In any partition of the vertices, either zero or two edges from a 3-cycle cross the cut. This requirement is enforced in the following integer program. If edge $(i, j) \notin E$, then w_{ij} is set to 0.

$$\begin{array}{rcl} & \max & \sum_{i,j \in V} w_{ij} x_{ij} \\ x_{ij} + x_{jk} + x_{ki} & \leq & 2 \quad \forall i, j, k \in V \\ x_{ij} + x_{jk} - x_{ki} & \geq & 0 \quad \forall i, j, k \in V \\ x_{ij} & \in & \{0, 1\}. \end{array}$$

Again, the last constraint can be relaxed so that each x_{ij} is required to lie between 0 and 1. In contrast to the aforementioned cycle-constraint based linear program, this linear programming relaxation has a polynomial number of constraints.

Both of these relaxations actually have the same optimal value for any graph with non-negative edge weights [Bar93, Pol92, PT95]. (For a simplified proof of this, see [New04].) Poljak showed that the integrality gap for each of these relaxations is arbitrarily close to 2 [Pol92]. In other words, there are classes of graphs that have a maximum cut containing close to half of the edges, but for which each of the above relaxations yields an upper bound close to all the edges, i.e. no better than the trivial "all-edges" bound. In particular, graphs with a maximum cut close to half the edges and with high girth can be used to demonstrate this gap. A comprehensive look at these linear programming relaxations is contained in the survey of Poljak and Tuza [PT95].

3.2 Eigenvalue Upper Bounds

Delorme and Poljak [DP93a] presented an eigenvalue upper bound on the value of a maximum cut, which was a strengthened version of a previous eigenvalue bound considered by Mohar and Poljak [MP90]. Computing Delorme and Poljak's upper bound is equivalent to solving an eigenvalue minimization problem. They showed that their bound is computable in polynomial time with arbitrary precision. In a series of work, Delorme, Poljak and Rendl showed that this upper bound behaves "differently" from the linear-programming-based upper bounds. For example, they studied classes of sparse random graphs (e.g. $G(n,p)$ with $p = 50/n$) and showed that their upper bound is close to optimal on these graphs [DP93b]. Since graphs of this type can also be used to demonstrate an integrality gap arbitrarily close to 2 for the aforementioned linear programming relaxations, their work highlighted contrasting behavior between these two upper bounds. Further computational experiments on other classes of graphs gave more evidence that the bound was indeed stronger than previously studied bounds [PR95b, PR94]. Delorme and Poljak conjectured that the 5-cycle demonstrated the worst-case behavior for their bound: a ratio of $\frac{32}{25+5\sqrt{5}} \approx .88445$ between their bound and the optimal integral solution. However, they could not prove that their bound was strictly less than twice the value of a maximum cut in the worst case.

4 Key Result

In 1994, Goemans and Williamson presented a randomized .87856-approximation algorithm for Max-Cut [GW95]. Their breakthrough work was based on rounding a semidefinite programming relaxation and was the first use of semidefinite programming in approximation algorithms. Poljak and Rendl showed that the upper bound provided by this semidefinite relaxation is equivalent to the eigenvalue bound of Delorme and Poljak [PR95a]. Thus, Goemans and Williamson's proved that the eigenvalue bound of Delorme and Poljak is no more than 1.138 times the value of a maximum cut.

4.1 A Semidefinite Relaxation

Max-Cut can be formulated as the following quadratic integer program, which is NP-hard to solve. Each vertex $i \in V$ is represented by a variable y_i, which is assigned either 1 or -1 depending on which side of the cut it occupies.

$$\begin{aligned} \max \quad & \frac{1}{2} \sum_{(i,j) \in E} w_{ij}(1 - y_i y_j) \\ y_i \quad \in \quad & \{-1, 1\} \qquad \forall i \in V. \end{aligned}$$

Goemans and Williamson considered the following relaxation of this integer program, in which each vertex is represented by a unit vector.

$$\begin{array}{rcll} & \max & \frac{1}{2} \sum_{(i,j)\in E} w_{ij}(1 - v_i \cdot v_j) & \\ v_i \cdot v_i & = & 1 & \forall i \in V \\ v_i & \in & \mathcal{R}^n & \forall i \in V. \end{array}$$

They showed that this relaxation is equivalent to a semidefinite program. Specifically, consider the following semidefinite relaxation:

$$\begin{array}{rcll} & \max & \frac{1}{2} \sum_{(i,j)\in E} w_{ij}(1 - y_{ij}) & \\ y_{ii} & = & 1 & \forall i \in V \\ Y & & \text{positive semidefinite.} & \end{array}$$

The equivalence of these two relaxations is due to the fact that a matrix Y is positive semidefinite if and only if there is a matrix B such that $B^T B = Y$. The latter relaxation can be solved to within arbitrary precision in polynomial time via the Ellipsoid Algorithm, since it has a polynomial time separation oracle [GLS88]. Thus, a solution to the first relaxation can be obtained by finding a solution to the second relaxation and finding a matrix B such that $B^T B = Y$. If the columns of B correspond to the vectors $\{v_i\}$, then $y_{ij} = v_i \cdot v_j$, yielding a solution to the first relaxation.

4.2 Random Hyperplane Rounding

Goemans and Williamson showed how to round the semidefinite programming relaxation of Max-Cut using a new technique that has since become known as "random-hyperplane rounding" [GW95]. First obtain a solution to the first relaxation, which consists of a set of unit vectors $\{v_i\}$, one vector for each vertex. Then choose a random vector $r \in \mathcal{R}^n$ in which each coordinate of r is chosen from the standard normal distribution. Finally, set $S = \{i \mid v_i \cdot r \geq 0\}$ and output the cut $(S, V \setminus S)$.

The probability that a particular edge $(i,j) \in E$ crosses the cut is equal to the probability that the dot products $v_i \cdot r$ and $v_j \cdot r$ differ in sign. This probability is exactly equal to θ_{ij}/π, where θ_{ij} is the angle between vectors v_i and v_j. Thus, the expected weight of edges crossing the cut is equal to $\sum_{(i,j)\in E} \theta_{ij}/\pi$. How large is this compared to the objective value given by the semidefinite programming relaxation, i.e. what is the approximation ratio?

Define α_{gw} as the worst-case ratio of the expected contribution of an edge to the cut, to its contribution to the objective function of the semidefinite programming relaxation. In other words: $\alpha_{gw} = \min_{0 \leq \theta \leq \pi} \frac{2}{\pi} \frac{\theta}{1-\cos\theta}$. It can be shown that $\alpha_{gw} > .87856$. Thus, the expected value of a cut is at least $\alpha_{gw} \cdot SDP_{OPT}$, resulting in an approximation ratio of at least .87856 for Max-Cut. The same analysis applies to weighted graphs with non-negative edge weights.

This algorithm was de-randomized by Mahajan and Hariharan [MH95]. Goemans and Williamson also applied their random-hyperplane rounding techniques to give improved approximation guarantees for other problems such as Max-Dicut and Max-2Sat.

4.3 Integrality Gap and Hardness

Karloff showed that there exist graphs for which the best hyperplane is only a factor α_{gw} of the maximum cut [Kar99], showing that there are graphs for which the analysis in [GW95] is tight. Since the optimal SDP value for such graphs equals the optimal value of a maximum cut, these graphs can not be used to demonstrate an integrality gap. However, Feige and Schechtman showed that there exist graphs for which the maximum cut is a α_{gw} fraction of the SDP bound [FS02], thereby establishing that the approximation guarantee of Goemans and Williamson's algorithm matches the integrality gap of their semidefinite programming relaxation. Recently, Khot, Kindler, Mossel and O'Donnell [KKMO04] showed that if the Unique Games Conjecture of Khot [Kho02] is assumed to be true, then it is NP-hard to approximate Max-Cut to within any factor larger than α_{gw}.

5 Applications

The work of Goemans and Williamson paved the way for the further use of semidefinite programming in approximation algorithms, particularly for graph partitioning problems. Methods based on the random-hyperplane technique have been successfully applied to many optimization problems that can be categorized as partition problems. A few examples are 3-Coloring [KMS98], Max-3-Cut [FJ97, GW04, dKPW04], Max-Bisection [HZ02], Correlation-Clustering [CGW03, Swa04], and Sparsest-Cut [ARV04]. Additionally, some progress has been made in extending semidefinite programming techniques outside the domain of graph partitioning to problems such as Betweenness [CS98], Bandwidth [BKRV00], and Linear Equations mod p [AEH01].

6 Cross References

Maximum Satisfiability, Bandwidth, Graph Coloring, Sparsest Cut.

References

[AEH01] Gunnar Andersson, Lars Engebretsen, and Johan Håstad. A new way to use semidefinite programming with applications to linear equations mod p. *Journal of Algorithms*, 39:162–204, 2001.

[ARV04] Sanjeev Arora, Satish Rao, and Umesh Vazirani. Expander flows, geometric embeddings and graph partitioning. In *Proceedings of the 36th Annual Symposium on the Theory of Computing (STOC)*, pages 222–231, Chicago, 2004.

[Bar93] Francisco Barahona. On cuts and matchings in planar graphs. *Mathematical Programming*, 60:53–68, 1993.

[BKRV00] Avrim Blum, Goran Konjevod, R. Ravi, and Santosh Vempala. Semi-definite relaxations for minimum bandwidth and other vertex-ordering problems. *Theoretical Computer Science*, 235:25–42, 2000.

[CGW03] Moses Charikar, Venkatesan Guruswami, and Anthony Wirth. Clustering with qualitative information. In *Proceedings of the 44th Annual IEEE Symposium on Foundations of Computer Science (FOCS)*, pages 524–533, Boston, 2003.

[CS98] Benny Chor and Madhu Sudan. A geometric approach to betweeness. *SIAM Journal on Discrete Mathematics*, 11:511–523, 1998.

[dKPW04] E. de Klerk, D.V. Pasechnik, and J.P. Warners. On approximate graph colouring and MAX-k-CUT algorithms based on the θ function. *Journal of Combinatorial Optimization*, 8(3):267–294, September 2004.

[DP93a] Charles Delorme and Svatopluk Poljak. Laplacian eigenvalues and the maximum cut problem. *Mathematical Programming*, 62:557–574, 1993.

[DP93b] Charles Delorme and Svatopluk Poljak. The performance of an eigenvalue bound in some classes of graphs. *Discrete Mathematics*, 111:145–156, 1993. Also appeared in *Proceedings of the Conference on Combinatorics*, Marseille, 1990.

[FJ97] Alan Frieze and Mark R. Jerrum. Improved approximation algorithms for MAX-k-CUT and MAX BISECTION. *Algorithmica*, 18:61–77, 1997.

[FS02] Uriel Feige and Gideon Schechtman. On the optimality of the random hyperplane rounding technique for MAX-CUT. *Random Structures and Algorithms*, 20(3):403–440, 2002.

[GLS81] Martin Grötschel, László Lovász, and Alexander Schrijver. The ellipsoid method and its consequences in combinatorial optimization. *Combinatorica*, 1:169–197, 1981.

[GLS88] Martin Grötschel, László Lovász, and Alexander Schrijver. *Geometric Algorithms and Combinatorial Optimization*. Springer-Verlag, Berlin, 1988.

[GW95] Michel X. Goemans and David P. Williamson. Improved approximation algorithms for maximum cut and satisfiability problems using semidefinite programming. *Journal of the ACM*, 42:1115–1145, 1995.

[GW04] Michel X. Goemans and David P. Williamson. Approximation algorithms for MAX-3-CUT and other problems via complex semidefinite programming. *STOC 2001 Special Issue of Journal of Computer and System Sciences*, 68:442–470, 2004.

[Hås01] Johan Håstad. Some optimal inapproximability results. *Journal of the ACM*, 48:798–869, 2001.

[HZ02] Eran Halperin and Uri Zwick. A unified framework for obtaining improved approximation algorithms for maximum graph bisection problems. *Random Structures and Algorithms*, 20(3):382–402, 2002.

[Kar72] Richard M. Karp. Reducibility Among Combinatorial Problems. In *Complexity of Computer Computations*, pages 85–104. Plenum Press, New York, 1972.

[Kar99] Howard J. Karloff. How good is the Goemans-Williamson MAX CUT algorithm? *SIAM Journal on Computing*, 29(1):336–350, 1999.

[Kho02] Subhash Khot. On the power of unique 2-prover 1-round games. In *Proceedings of the 34th Annual Symposium on the Theory of Computing (STOC)*, pages 767–775, Montreal, 2002.

[KKMO04] Subhash Khot, Guy Kindler, Elchanan Mossel, and Ryan O'Donnell. Optimal inapproximability results for MAX CUT and other 2-variable CSPs? In *Proceedings of the 45th Annual IEEE Symposium on Foundations of Computer Science (FOCS)*, pages 146–154, Rome, 2004.

[KMS98] David R. Karger, Rajeev Motwani, and Madhu Sudan. Improved graph coloring via semidefinite programming. *Journal of the ACM*, 45(2):246–265, 1998.

[MH95] Rajeev Mahajan and Ramesh Hariharan. Derandomizing semidefinite programming based approximation algorithms. In *Proceedings of the 36th Annual IEEE Symposium on Foundations of Computer Science (FOCS)*, pages 162–169, Milwaukee, 1995.

[MP90] Bojan Mohar and Svatopluk Poljak. Eigenvalues and the max-cut problem. *Czechoslovak Mathematical Journal*, 40(115):343–352, 1990.

[New04] Alantha Newman. A note on polyhedral relaxations for the maximum cut problem. Unpublished manuscript, 2004.

[Pol92] Svatopluk Poljak. Polyhedral and eigenvalue approximations of the max-cut problem. *Sets, Graphs and Numbers, Colloqiua Mathematica Societatis Janos Bolyai*, 60:569–581, 1992.

[PR94] Svatopluk Poljak and Franz Rendl. Node and edge relaxations of the max-cut problem. *Computing*, 52:123–137, 1994.

[PR95a] Svatopluk Poljak and Franz Rendl. Nonpolyhedral relaxations of graph-bisection problems. *SIAM Journal on Optimization*, 5:467–487, 1995.

[PR95b] Svatopluk Poljak and Franz Rendl. Solving the max-cut using eigenvalues. *Discrete Applied Mathematics*, 62(1–3):249–278, September 1995.

[PT95] Svatopluk Poljak and Zsolt Tuza. Maximum cuts and large bipartite subgraphs. *DIMACS Series in Discrete Mathematics and Theoretical Computer Science*, 20:181–244, 1995.

[SG76] Sartaj Sahni and Teofilo Gonzalez. P-complete approximation problems. *Journal of the ACM*, 23(3):555–565, 1976.

[Swa04] Chaitanya Swamy. Correlation clustering: maximizing agreements via semidefinite programming. In *Proceedings of 15th Annual ACM-SIAM Symposium on Discrete Algorithms (SODA)*, pages 526–527, New Orleans, 2004.

[TSSW00] Luca Trevisan, G.B. Sorkin, M. Sudan, and D.P. Williamson. Gadgets, approximation, and linear programming. *SIAM Journal on Computing*, 29(6):2074–2097, 2000.