A Randomized Approximation Scheme for Metric MAX-CUT

W. Fernandez de la Vega
CNRS, LRI, bat 490
Universite Paris Sud
91405 Orsay Cedex, France
Email: lalo@lri.fr

Claire Kenyon
LRI, bat 490
Universite Paris Sud
91405 Orsay Cedex, France
Email: kenyon@lri.lri.fr
Web page: http://www.lri.fr/ kenyon

Abstract

Metric MAX-CUT is the problem of dividing a set of points in metric space into two parts so as to maximize the sum of the distances between points belonging to distinct parts. We show that metric MAX-CUT has a polynomial time randomized approximation scheme.

1. Introduction

1.1. Background

MAX-CUT, the problem of finding a 2-partition of the vertices of a (possibly weighted) graph which maximizes the number of edges (or sum of edge weights) accross the partition, has recently attracted a lot of attention. It has been known for a long time that this basic optimization problem is $\mathcal{NP}$-hard [8] but has a (straightforward) .5-approximation algorithm [18]. The best approximation in the general case is a recent exciting .87856-approximation algorithm due to Goemans and Williamson [9, 10], building upon previous work [4, 5, 15]. Unfortunately, there is not much room for improvement since the problem is Max-$\mathcal{SNP}$-hard [14], and hence [17] has no ϵ-approximation scheme if $\mathcal{P} \neq \mathcal{NP}$. Thus one is led to consider restricted versions of MAX-CUT. In [16, 2], polynomial time approximation schemes were presented for dense unweighted graphs, *i.e.* graphs with $\Omega(n^2)$ edges, and in [3], dense weighted instances are dealt with. Related results also appear in [13, 7].

In this paper, we focus on metric instances of MAX-CUT, when the vertices correspond to points in metric space, the graph is the complete graph, and edge $\{x, y\}$ has a weight equal to the distance between x and y. Our main result is that metric MAX-CUT has a Polynomial Time Approximation Scheme.

Metric MAX-CUT is mentioned by Bern in [6, chapter 8], in relation to clustering problems. Following his presentation, we loosely define a 2-clustering problem as given by a set X of points in some metric space and seeking the "best" partition of X into two clusters. To measure the quality of the partition, some criterion is applied to each cluster individually. Some criterions for which polynomial-time algorithms exist are the diameter [19, 11] and the variance [1, 12], but there is no polynomial-time algorithm known for minimizing the sum of pairwise distances, which is equivalent to maximizing the sum of distances between points in different clusters, i.e. to metric MAX-CUT. Thus Bern raises the question of designing an efficient approximation algorithm for metric MAX-CUT, which is answered in this paper.

1.2. The result

We now state our main theorem.

Theorem 1 *Metric MAX-CUT has a Polynomial Time Approximation Scheme (PTAS),* i.e. *for any given $\epsilon > 0$, there is a randomized algorithm which takes as input a discrete metric space given by its distance matrix, runs in time polynomial in the size of the space, and outputs a bipartition whose value is at least $(1 - \epsilon)$ times the value of the maximum cut.*

One may contrast this result with the situation for the Traveling Salesman Problem (TSP): as is the case for MAX-CUT, the TSP problem is Max-$\mathcal{SNP}$-hard[14], but even the metric version of TSP is also Max-$\mathcal{SNP}$-hard.

Unfortunately, we must here mention two weaknesses of this result. First, metric MAX-CUT is not known to be $\mathcal{NP}$-hard, so that someone may conceivably come up with an exact polynomial-time solution (the likelihood of this event is perhaps lessened by the lack of results on clustering with respect to our criterion). Second, a PTAS for MAX-CUT is not the same as a PTAS for clustering: for example, if in one dimension there are $n/2$ points at position 0 and $n/2$ points at position 1, then the maximum cut obviously has value $n^2/4$, and the value of Min-Cluster is 0. A cut (L, R) such that L contains $n(1/2 - \epsilon)$ of the points at 0 would be an ϵ-approximation to MAX-CUT, but would be very bad with respect to Min-Cluster. Thus we do not have a PTAS for Min-Cluster.

1.3. Proof techniques

We will obtain our main theorem as a consequence of the following reduction.

Theorem 2 *Approximating Metric MAX-CUT reduces to approximating Dense MAX-CUT.*

Actually, we will reduce metric MAX-CUT to an instance of ordinary MAX-CUT in which the maximum weight exceeds the average weight by at most a constant factor. It is almost immediate to check that the algorithms for dense (in the usual sense of the word) MAX-CUT work for this case with trivial modifications. We refer to [FdlVK-] for a more general definition of dense weighted instances.

In fact the main idea of the reduction is the following. Let us first see what problem is raised by a naive adaptation of dense graphs algorithms to metric MAX-CUT. The first step usually consists in taking a constant size sample of the vertices. In the dense graphs setting, all significant vertices have the same number of edges (up to a constant factor), hence contribute the same number of edges to MAX-CUT (up to a constant factor), hence a sample of constant size is sufficient to get a fairly good picture of the whole graph. In the metric setting, the situation is completely different. Outliers (points really far from the rest of the set) may contribute much more to MAX-CUT than other points. A constant size sample is bound to miss the few outliers, and examining the sample will not give good information about MAX-CUT. Thus a naive adaptation of the dense graph algorithm to metric MAX-CUT is doomed. The solution to this problem is simple: the critical observation is that the contribution of a point x to MAX-CUT is roughly proportional to the average distance from x to the rest of the set. Thus in the metric setting one should not use a uniform sample of the set of points, but a biased sample, where the probability of taking x in the sample is proportional to the average distance from x to the rest of the set. This is the key to our algorithm. In practice, we create a "graph of clones" obtained by duplicating each vertex a number of times proportional to its distance to the rest of the set, and perform a uniform sample on the graph of clones.

The analysis uses two main observations: one, that up to a small constant factor, solving metric MAX-CUT reduces to solving MAX-CUT on the (suitably weighted) graph of clones; two, that this graph is a dense instance of MAX-CUT. Both proofs rely on the triangular inequality.

1.4. Open Problems

Reasonable extensions would include trying the same approach to solve other optimization problems of a similar flavor: bisection and cutting into k parts for example.

2. Notations and Definitions

We need a few important definitions.

Throughout the paper, we denote by $d(x, y)$ the distance between two points x and y. Let X be a set of n points in a metric space. The value of a partition (or cut) (A, B) of X is the sum of the distances between points of A and points of B. Let MAXCUT(X) denote the value of an optimum cut of X.

Definition 1 *The* weight w_x *of* $x \in X$ *is defined as:* $w_x = \sum_{y \in X} d(x, y)$. *The* total weight W *of* X *is defined as:* $W = \sum_{x \in X} w_x$.

Up to scaling, we can assume that the average distance between two points drawn at random from X is equal to 1, *i.e.* that $W = 2n^2$.

Definition 2 *We define a weighted complete graph* $G' = (X', E')$*, where the* set of clones X' *is a multiset of points; each point* $x \in X$ *is cloned to create* $\lfloor w_x \rfloor$ *identical points of* X'*, and the edge between* x'*, a clone of* x*, and* y'*, a clone of* y*, has weight* $e_{x'y'} = d(x, y)/(w_x w_y)$.

3. The Algorithm

The algorithm is very simple.

Input: A discrete metric space X of size n with distance function d.

1. Scale d so that $\sum_{x \in X} \sum_{y \in X} d(x, y) = 2n^2$.
2. Construct the graph of clones $G' = (X', E')$.
3. Apply the dense weighted MAX-CUT approximation scheme from [3] to G', so as to get a cut (A', B') of X'.
4. Construct a cut (A, B) of X by doing the following for each $x \in X$:
   - Let $x' \in X'$ be any clone of x, let $a_x = \sum_{z' \in A'} e_{x'z'}$ and $b_x = \sum_{z' \in B'} e_{x'z'}$.
   - If $a_x < b_x$ then put x in A, otherwise put x in B.

Output: Cut (A, B) of X.

4. The Analysis

We will need the following lemma.

Lemma 1

$$\forall\, x \in X,\;\; W \leq 2nw_x.$$

Proof: By the triangular inequality, $d(y,z) \leq d(y,x) + d(x,z)$. Summing over all $y, z \in X$ gives the statement of the lemma. $\square$

Since we have scaled the problem to instances where $W = 2n^2$, the weight of any x is at least n and the lemma below follows immediately.

Lemma 2

$$\forall\, x \in X,\;\; \lfloor w_x \rfloor \geq w_x(1 - 1/n).$$

The theorem below reduces MAX-CUT on X to MAX-CUT on its graph of clones.

Lemma 3

$$\mathit{MAXCUT}(X)(1 - \frac{1}{n})^2 \leq \mathit{MAXCUT}(X').$$

Proof: Consider an optimal cut (L, R) of X, and let (L', R') be the induced cut of X' (where L' contains all the clones of elements of L and R' contains all the clones of elements of R). We have:

$$\begin{aligned}
\text{Value}(L', R') &= \sum_{x' \in L', y' \in R'} e_{x'y'} \\
&= \sum_{x \in L, y \in R} \lfloor w_x \rfloor \lfloor w_y \rfloor \frac{d(x,y)}{w_x w_y} \\
&\geq \sum_{x \in L, y \in R} d(x,y)(1 - \frac{1}{n})^2 \\
&= \text{MAXCUT}(X)(1 - \frac{1}{n})^2,
\end{aligned}$$

where the inequality follows from Lemma 2. Since the optimum cut of X' is at least as good as (L', R'), this proves the theorem. $\square$

Lemma 4 *If (A', B') is any cut of X', and (A, B) is the cut of X obtained from (A', B') by proceeding as in step 4 of the algorithm, then the value of (A, B) is greater than or equal to the value of (A', B').*

Proof: Let A'_x denote the set of clones of x which belong to A' and B'_x denote the set of clones of x which belong to B'. Using the notations of the algorithm and assuming that we are in the case $a_x < b_x$ (that is, when the algorithm places x in A), we have:

$$\text{Value}(A' \cup B'_x, B' \setminus B'_x) = |B'_x|(b_x - a_x) \geq 0.$$

Thus the cut $(A' \cup B'_x, B' \setminus B'_x)$ of X' is at least as good as (A', B') and has all clones of x on the same side. Repeating this for all $x \in X$, we obtain a cut (A'', B'') of X' which is at least as good as (A', B') and which has all clones of elements of A in A'' and all clones of elements of B in B''. Then:

$$\begin{aligned}
\text{Value}(A', B') &\leq \text{Value}(A'', B'') \\
&= \sum_{x' \in A'', y' \in B''} e_{x'y'} \\
&= \sum_{x \in A, y \in B} \lfloor w_x \rfloor \lfloor w_y \rfloor \frac{d(x,y)}{w_x w_y} \\
&\leq \text{Value}(A, B),
\end{aligned}$$

which concludes the proof of the theorem. $\square$

All that remains now is to show that (X', E') is a dense weighted graph.

Lemma 5

$$\max_{x', y' \in X'} (e_{x'y'}) \leq \frac{4}{(1 - 1/n)^2} \frac{\sum_{x', y' \in X'} e_{x',y'}}{|X'|^2}.$$

Proof: First note that

$$|X'| = \sum_{x \in X} \lfloor w_x \rfloor \leq W.$$

Thus the average value of a random entry of the adjacency matrix E' is

$$\begin{aligned}
&\frac{\sum_{x,y \in X} \lfloor w_x \rfloor \lfloor w_y \rfloor d(x,y)/(w_x w_y)}{|X'|^2} \\
&\geq \frac{(1 - 1/n)^2 W}{W^2} \\
&= \frac{(1 - 1/n)^2}{W}.
\end{aligned}$$

where we have used Lemma 2.

Now, take an arbitrary entry of E'. Using the triangular inequality, we can bound it as follows:

$$\begin{aligned}
\frac{d(x,y)}{w_x w_y} &\leq \frac{\frac{1}{n}\sum_z (d(x,z) + d(z,y))}{w_x w_y} \\
&= \frac{1}{n} \frac{w_x + w_y}{w_x w_y} \\
&= \frac{1}{n w_y} + \frac{1}{n w_x} \\
&\leq \frac{4}{W},
\end{aligned}$$

where the last inequality follows from Lemma 1.

This implies the theorem. $\square$

Acknowledgement We wish to thank Marek Karpinski and Richard Kenyon for fruitful discussions, and the second author wishes to thank Oded Schramm for babysitting on May 4.

References

[1] E. Boros and P. Hammer. On clustering problems with connected optima in euclidean spaces. *Discrete Mathematics*, 75:81–88, 1989.

[2] W. F. de la Vega. Max-cut has a randomized approximation scheme in dense graphs. *Random Structures and Algorithms*, 8(3):187–198, 1996.

[3] W. F. de la Vega and M. Karpinski. Polynomial time approximation of dense weighted instances of max-cut. *preprint*.

[4] C. Delorme and S. Poljak. Combinatorial properties and the complexity of a max-cut approximation. *European Journal of Combinatorics*, 14:313–333, 1993.

[5] C. Delorme and S. Poljak. Laplacian eigenvalues and the maximum cut problem. *Mathematical Programming*, 62:557–574, 1993.

[6] D. S. H. (editor).

[7] A. Frieze and R. Kannan. Quick approximations to matrices and applications. *Manuscript*, 1997.

[8] M. Garey and D. Johnson. Computers and intractability, a guide to the theory of np-completeness. 1979.

[9] M. X. Goemans and D. P. Williamson. .878- approximation algorithms for max-cut and max-2sat. *Proc. 26th Symposium on the Theory of Computing (STOC)*, pages 422–431, May 1994.

[10] M. X. Goemans and D. P. Williamson. Improved approximation algorithms for maximum cut and satisfiability problems using semidefinite programming. *Journal of the ACM*, 42(6):1115–1145, November 1995.

[11] H. Hershberger and S. Suri. Finding tailored partitions. *Journal of Algorithms*, 12:431–463, 1991.

[12] N. K. M. Inaba and H. Imai. Applications of weighted voronoi diagrams and randomization of variances-based k-clustering. *Proceedings of the 10th ACM Symposium on Computational Geometry*, pages 332–339, 1994.

[13] S. G. O. Goldreich and D. Ron. Property testing and its connection to learning and approximation. *Proceedings of 37th Foundations of Computer Science (FOCS)*, November 1996.

[14] C. Papadimitriou and M. Yannakakis. Optimization, approximation and complexity classes. *Journal of Computer and System Sciences*, 43:425–440, 1991.

[15] S. Poljak and F. Rendl. Solving the max-cut problem using eigenvalues. *Report 91735-OR, Institute fur Diskrete Mathematik, University of Bonn*, 1991.

[16] D. K. S. Arora and P. T. M. Karpinski. Approximation schemes for dense instances of np-hard problems. *Proc. 27th ACM Symposium on the Theory of Computing (STOC)*, pages 284–293, May 1995.

[17] R. M. M. S. S. Arora, C. Lund and M. Szegedy. Proof verification and hardness of approximation problems. *Proc.* 33^{rd} *IEEE Foundations of Computer Science (FOCS)*, pages 24–35.

[18] S. Sahni and T. Gonzalez. P-complete approximation problems. *Journal of the ACM*, 23:555–565, 1976.

[19] G. R. V. Capoyleas and G. Woeginger. Geometric clusterings. *Journal of Algorithms*, 12:341–356, 1991.