A randomized 1.01241-approximation algorithm for maximum cut problem

Majid Zohrehbandian

Department of Mathematics, Karaj Branch, Islamic Azad University, Karaj, Iran.

zohrebandian@yahoo.com

Abstract

Maximum cut problem is a famous combinatorial problem, which its complexity has been heavily studied over the years. Among them is the efficient algorithm of Goemans and Williamson with an approximation factor roughly $1.13823 \cong \frac{1}{0.878}$ (It is most often expressed as 0.878). Their algorithm combines semidefinite programming and a rounding procedure to produce an approximate solution to the maximum cut problem. In this paper, after introducing a new semidefinite programming formulation we present an improved randomized approximation with an approximation factor roughly $1.01241 \cong \frac{1}{0.98775}$.

Keywords: Discrete Optimization, Maximum Cut Problem, Complexity Theory, NP-Complete Problems.

MSC 2010: 90C35**,** 90C60.

1. Introduction

In complexity theory, the abbreviation NP refers to "nondeterministic polynomial", where a problem is in NP if we can quickly (in polynomial time) test whether a solution is correct. P and NP-complete problems are subsets of NP Problems. We can solve P problems in polynomial time while determining whether or not it is possible to solve NP-complete problems quickly (called the P vs NP problem) is one of the principal unsolved problems in Mathematics and Computer science.

Due to the intractability of NP-complete problems, they were often addressed by using heuristic methods and approximation algorithms and as the field progressed, it became apparent that different NP-complete optimization problems have different approximation factors.

In this paper, we consider the maximum cut (Max-Cut) problem which is a famous NP-complete problem. It has been shown that it is NP-hard to approximate Max-Cut better than $1.0625 \cong \frac{1}{0.941}$, and there is a belief that $1.13823 \cong \frac{1}{0.878}$ may be the best polynomial-time approximation one can achieve if the Unique Games Conjecture (UGC) is true [2, 3]. Here, we want to show that a randomized 1.01241–approximation for Max-Cut problem can be obtained based on Goemans-Williamson approach on a new semidefinite programming (SDP) relaxation.

The rest of the paper is structured as follows. Section 2 is about the Max-Cut problem and introduces characteristics about vectors that have 120° angle to each other. In section 3, new SDP relaxation and new rounding procedure are introduced which lead to a randomized approximation algorithm for the Max-Cut problem with a performance ratio better than 1.01241. Finally, Section 4 concludes the paper.

2. Max-Cut Problem

In the mathematical discipline of graph theory, a cut in an undirected graph $G = (V, E)$, is defined as a partition of the vertices of G into two sets S and $V - S$. The size or the weight of a cut, denoted by $W(S;\ V - S)$, is the number of the edges that connect vertices of one set to the vertices of the other. Trivially, one can define the Max-Cut problem as the problem of finding a cut in G with maximum weight and it is a typical example of an NP-complete problem. Aside from its theoretical interest, the Max-Cut problem arises in many practical applications. This fact has encouraged considerable effort in finding good approximation solutions.

Despite many attempts to design approximation algorithms for Max-Cut problem, the best-known approximation ratio is 1.13823 and it is based on using SDP relaxation [1]. By assigning a unit vector $v_i \in \mathbb{R}^n$ to each vertex $i \in V$, a well known SDP formulation of the Max-Cut problem is as follows:

$$\max_{s.t.} \sum_{ij \in E} \frac{1 - v_i v_j}{2} \qquad (SDP1)$$

$$v_i v_i = 1 \qquad i \in V$$

$$v_i v_j \in \{-1, +1\} \qquad i, j \in V$$

To get the SDP1 relaxation just let $v_i v_j \in [-1, +1]$. The Goemans-Williamson algorithm [1] uses the solution of the SDP1 relaxation and cutting the optimal vectors v_i^* with a random hyperplane through zero where everything on one side of the hyperplane is in one partition, and everything on the other side of the hyperplane is in the other.

Interestingly, although the additional inequalities improve the SDP1 relaxation, they do not necessarily give rise to better approximation algorithms and so after the Goemans-Williamson rounding procedure, we are still left with a 1.13823-approximation algorithm.

Next section deals with the major contribution of the paper which is a rounding procedure, based on a new SDP formulation of the Max-Cut problem. Before going on further, we start by considering the following characteristics about vectors with angles equal to 120°, which we call 120-degree condition.

Theorem 1. Suppose that there are 3 vectors V_1, V_2, $V_3 \in \mathbb{R}^n$ which satisfy 120-degree condition; i.e. $V_1 o V_2 = V_2 o V_3 = V_3 o V_1 = 120°$, where $V_i o V_j$ denote the angle between vectors V_i and V_j. Then, these vectors are coplanar.

Proof. We know that two arbitrary vectors are always coplanar. Then, we can assume that the vectors V_2 and V_3 have been fixed on the x_1x_2 plane in coordinates $V_2 = \begin{bmatrix}\frac{-\sqrt{3}}{2} & \frac{-1}{2} & 0 & \cdots & 0\end{bmatrix}^t$ and $V_3 = \begin{bmatrix}\frac{\sqrt{3}}{2} & \frac{-1}{2} & 0 & \cdots & 0\end{bmatrix}^t$, where $V_1 = [a_1 \quad \cdots \quad a_n]^t$. Then, it is sufficient to show that $a_2 > 0, a_1 = a_3 = \quad \ldots \quad = a_n = 0$.

We have $V_1V_2 = V_1V_3 = \frac{-1}{2}\|V_1\|$. Hence, $\frac{-\sqrt{3}}{2}a_1 - \frac{1}{2}a_2 = \frac{\sqrt{3}}{2}a_1 - \frac{1}{2}a_2$, and therefore $a_1 = 0$.

Moreover, based on the length of a vector and the law of cosine on triangles we have:

$$\left\|\overrightarrow{V_1V_3}\right\|^2 = \frac{3}{4} + (a_2 + \frac{1}{2})^2 + a_3^2 + \cdots + a_n^2$$

$$\left\|\overrightarrow{V_1V_3}\right\|^2 = (a_2^2 + \cdots + a_n^2) + (1) - 2\left(\sqrt{a_2^2 + \cdots + a_n^2}\right)(1)\cos(V_1oV_3)$$

Therefore, $\cos(V_1oV_3) = \frac{-a_2}{2\left(\sqrt{a_2^2+\cdots+a_n^2}\right)} = \frac{-1}{2}$ iff $a_2 > 0, a_1 = a_3 = \quad \ldots \quad = a_n = 0$ ■

Corollary 1. Let V_1, V_2, $V_3 \in \mathbb{R}^n$ satisfy the 120-degree conditions, where $\|V_i\| = 1$ (i=1,2,3). Then we have $V_l = -\left(V_p + V_q\right) \quad 1 \le l \ne p \ne q \le 3$.

Corollary 2. Let $V_1, \ldots, V_k \in \mathbb{R}^n$ satisfy the 120-degree conditions. Then we have $k \le 3$.

3. A new randomized approximation algorithms for Max-Cut problem

We will now sketch a randomized 1.01241-approximation algorithm for Max-Cut problem, which is roughly similar to the Goemans-Williamson algorithm [1]. The main idea is to use a new SDP relaxation as follows, where we want to have a basis vector v_o^1 and two sets of vectors in which the 120-degree condition is satisfied between them as much as possible.

$$\max_{s.t.} \sum_{ij \in E} \frac{1 - v_i v_j}{3/2} \qquad (SDP2)$$

$$v_o^p v_o^p = v_i v_i = 1 \qquad i \in V, \quad p = 1,2,3$$

$$v_o^p v_o^q = v_o^1 v_i = -1/2 \qquad i \in V, \quad 1 \le p \neq q \le 3$$

$$v_o^p v_i, v_i v_j \in \{-1/2, +1\} \qquad i, j \in V, \quad p = 2,3$$

Note that, based on the 120-degree equalities, the angles $v_o^p o\, v_o^q$ and $v_o^1 o\, v_i$ are 120° ($i \in V$, $1 \le p \neq q \le 3$). Therefore, the vectors v_o^1, v_o^2, and v_o^3 are coplanar.

Definition 1. Let P_{123} denote the two-dimensional plane constructed based on the optimal vectors v_o^{1*}, v_o^{2*}, and v_o^{3*} .

Here, in an integral solution of the SDP2 relaxation, a cut set is composed of the edges $ij \in E$ that their corresponding vectors v_i and v_j satisfy the 120°-condition; i.e. v_i is picked coincide with the vector v_o^2 and v_j is picked coincide with the vector v_o^3 , or vice versa. In other words, $\forall i \in S,\ \forall j \in V - S: v_i v_j = -1/2$ and $\forall i, j \in S: v_i v_j = 1$, $\forall i, j \in V - S: v_i v_j = 1$.

After relaxing the constraints $v_o^p v_i, v_i v_j \in \{-1/2, +1\}$ as $v_o^p v_i, v_i v_j \in [-1/2, +1]$ for $i, j \in V$ and $p = 2,3$, we can solve the SDP2 relaxation. Let v_o^{p*}, v_i^* $i \in V$, $p = 1,2,3$ is an optimal solution of it, and v_i' $i \in V$ are the projection of the vectors v_i^* $i \in V$ onto the two-dimensional plane P_{123}; See Figure 1.

Theorem 2. Let $v_i = v'_i + v''_i$, where v'_i is the projection of vector v_i onto the plane P_{123} and v''_i is the projection of v_i onto the normal vector of that plane. Then, the vector v_i' is placed between two vectors v_o^{2*} and v_o^{3*}; See Figure 1.

Proof. $\frac{-1}{2} = v_o^{1*} v_i = v_o^{1*} v'_i + v_o^{1*} v''_i = v_o^{1*} v'_i = \|v'_i\| \cos(v_o^{1*} o v'_i) \ge \cos(v_o^{1*} o v'_i)$. In other words, the angle between v_o^{1*} and v'_i is equal or greater than 120° and vector v_i' $i \in V$ is placed between two vectors v_o^{2*} and v_o^{3*} ∎

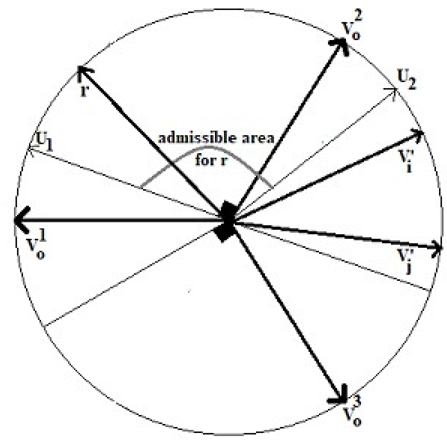

Figure 1. The two-dimensional plane constructed by the vectors v_o^{p*} $p = 1,2,3$ and the projection of the optimal vectors v_i^* onto it.

We would now like to round the optimal solution of the SDP2 relaxation to obtain a 1.01241-approximation solution for Max-Cut problem. To do this, it is sufficient to pick a random hyperplane $rv = 0$ passing through the origin which cuts two vectors v_o^{2*} and v_o^{3*}. We can also pick the normal to that hyperplane to be a random vector r. Moreover, we will show that it is sufficient to introduce such a normal vector r on the plane P_{123} and between the vectors u_1 and u_2; See Figure 1. Then, we can partition the v_i vectors according to which side of the hyperplane $rv = 0$ they lie.

Theorem 3. Let r be a vector on the plane P_{123}. Then, the probability that two vectors v_i and v_j are separated by the hyperplane $rv = 0$ equals to the probability that two vectors v'_i and v'_j are separated by that hyperplane.

Proof. Let $v_i = v'_i + v"_i$, where v'_i is the projection of vector v_i onto the plane P_{123} and $v"_i$ is the projection of v_i onto the normal vector of that plane. Then, $rv_i = rv'_i + rv"_i = rv'_i$. In other words, v_i and v_j are placed on the opposite side of the hyperplane $rv = 0$ if and only if v'_i and v'_j are placed on the opposite side of that hyperplane. Hence, the probability of splitting the vectors v_i and v_j by the hyperplane $rv = 0$ is equal to the probability of splitting the vectors v'_i and v'_j by that hyperplane ■

It is obvious that by the construction of normal vectors r on the plane P_{123} (or more limited between the vectors u_1 and u_2) and introducing a hyperplane $rv = 0$, we can cover all of vectors v'_i and as a result, we can cover all of vectors v_i. Then, we can pick a random vector r, inside the angle $u_1 o u_2$ and iterate through all the vertices to put $i \in S$ if $rv_i \geq 0$ and $i \in V - S$ otherwise. To do this, we introduce u_1 as a positive linear combination of the vectors v_o^{1*} and v_o^{2*} and u_2 as a positive linear combination of the vectors v_o^{2*} and v_o^{3*} where $u_1 o v_o^{1*} = u_2 o v_o^{2*} = 30°$, $u_1 o v_o^{2*} = u_2 o v_o^{3*} = 90°$. Then, we produce two positive-valued random variables α and β to introduce a (random) vector r as $r = \alpha u_1 + \beta u_2$.

Now, let $(S; V - S)$ be the cut obtained by the above rounding scheme. It is easy to see that the probability that edge $ij \in E$ is in the cut is $\frac{\theta_{ij}}{\frac{2\pi}{3}} = \frac{Arccos(v_i v_j)}{\frac{2\pi}{3}}$. Then, if X denoted the number of edges crossing the cut, the expected weight of the cut is $\mathbb{E}[X] = \sum_{ij \in E} \Pr\,(ij \in cut)$, which we want to compare against optimal values of SDP2 relaxation and Max-Cut problem. Hence, we have $\mathbb{E}[X] = \sum_{ij \in E} \frac{\theta_{ij}}{\frac{2\pi}{3}} = \sum_{ij \in E} \frac{Arccos(v_i v_j)}{\frac{2\pi}{3}} \geq \rho \sum_{ij \in E} \frac{1 - v_i v_j}{3/2} = \rho z_{SDP2}^* \geq \rho z_{Max-C}^*$, which concludes that $\rho^* = 0.98775 \leq \frac{9}{4} \frac{\sum_{ij \in E} Arccos(v_i v_j)}{\pi \sum_{ij \in E}(1 - v_i v_j)}$. This immediately implies that the proposed randomized algorithm will achieve an approximation ratio of $1.01241 \cong \frac{1}{0.98775}$.

Therefore, we have a randomized 1.01241-approximation algorithm for Max-Cut problem. The statement of the solution idea can be accomplished by the following algorithm.

Algorithm (Input: Graph $G = (V, E)$; Output: Vertices partitioning to S and $V - S$)

Step 1. After relaxing the constraints $v_o^p v_i, v_i v_j \in \{-1/2, +1\}$ as $-1/2 \leq v_o^p v_i, v_i v_j \leq +1$, let v_o^{p*}, v_i^* $i \in V$, $p = 1,2,3$ is an optimal solution of the SDP2 relaxation.

Step 2. Let u_1 be a linear combination of the vectors v_o^{1*} and v_o^{2*}, and u_2 be a linear combination of the vectors v_o^{2*} and v_o^{3*}, where $u_1 o v_o^{1*} = u_2 o v_o^{2*} = 30°$, $u_1 o v_o^{2*} = u_2 o v_o^{3*} = 90°$.

Step 3. Introduce two positive-valued random variables α and β, to produce a (random) vector r as $r = \alpha u_1 + \beta u_2$. Let $S = \{i | rv_i \geq 0\}$ and $V - S = \{i | rv_i < 0\}$.

Since any individual run of the algorithm might not produce a value of $0.98775 z_{Max-C}^*$, in practice, one can repeats Step 3 several times and picks the best cut that was generated.

Theorem 4. (Håstad (2001)) If there is an r-approximation algorithm for Max-Cut, where $r < \frac{17}{16} = 1.0625$, then P = NP.

Corollary 3. Due to proposing a randomized 1.01241–approximation algorithm for Max-Cut problem, we can conclude that P=NP.

4. Conclusions

One of the open problems about the Max-Cut problem is the possibility of introducing a randomized approximation algorithm within any constant factor smaller than $\frac{17}{16}$. Here, we introduced a randomized 1.01241–approximation algorithm for Max-Cut problem. In this manner, this approximation of Max-Cut problem to within a factor of $r = 1.01241 < \frac{17}{16}$ implies that P=NP.

However, many results in complexity theory and computational optimization assume solidly based on the hypotheses P≠NP. But, now that we know P=NP, we should make fundamental modifications in many of the results discussed in literature.

References

1. M.X. Goemans and D.P. Williamson (1995) "Improved approximation algorithms for maximum cut and satisfiability problems using semidefinite programming" *Journal of the ACM*, 42(6), 1115−1145.
2. J. Håstad (2001) "Some optimal inapproximability results" *Journal of the ACM*, 48(4), 798−859.
3. S. Khot**,** G. Kindler, E. Mossel and R. O'Donnell (2007) "Optimal Inapproximability Results for Max-Cut and Other 2-Variable CSPs?" *SIAM Journal on Computing*, 37(1), 319−357.