Limits of Approximation Algorithms 21 Jul, 2009

Lec. 8: Semidefinite Programming and Unique Games

Lecturer: Moses Charikar *Scribe: Alantha Newman*

8.1 Unique Games

The topic of Unique Games has generated much interest in the past few years. The *Unique Games Conjecture* was posed by Khot [Kho02]. We will discuss the associated optimization problem and the algorithmic intuition and insight into the conjecture, as well as the limits of these algorithmic techniques. Finally, we mention the amazing consequences implied for many optimization problems if the problem is really as hard as conjectured.

We now define the Unique Games problem. The input is a set of variables V and a set of k labels, L, where k is the size of the *domain*. Our goal is to compute a mapping, $\ell : V \to L$, satisfying certain constraints that we now describe. Let E denote a set of pairs of variables, $\{(u,v)\} \subset V \times V$. For each $(u,v) \in E$, there is an associated constraint represented by π_{uv}, indicating that $\ell(v)$ should be equal to $\pi_{uv}(\ell(u))$; we assume that the constraint π_{vu} is the inverse of the constraint π_{vu} i.e, $\pi_{uv} = \pi_{vu}^{-1}$. Thus, our goal is to compute the aforementioned mapping, $\ell : V \to L$, so as to *maximize the number of satisfied constraints*.

Each constraint, π_{uv}, can be viewed as a permutation on L. Note that this permutation may be different for each pair $(u,v) \in E$. For a pair $(u,v) \in E$, if v is given a particular label from L, say $\ell(v)$, then there is only one label for u that will satisfy the constraint π_{uv}. Specifically, $\ell(u)$ should equal $\pi_{vu}(\ell(v))$. Hence, the "unique" in Unique Games. The practice of calling this optimization problem a unique "game" stems from the connection of this problem to 2-prover 1-round games [FL92]. The Unique Games problem is a special case of Label Cover (discussed in other lectures in the workshop), in which each constraint forms a bijection from L to L. Having such a bijection turns out to be useful for hardness results.

8.2 Examples

We will refer to E as a set of edges, since we can view an instance of Unique Games as a graph $G = (V,E)$ in which each edge $(u,v) \in E$ is labeled with a constraint π_{uv}. We now give some specific examples of optimization problems that are special cases of Unique Games.

8.2.1 Linear Equations mod p

We are given a set of equations in the form $x_i - x_j \equiv c_{ij} \pmod{p}$. The goal is to assign each variable in $V = \{x_i\}$ a label from the set $L = [0, 1, \ldots p-1]$ so as to maximize the number of satisfied equations. Note that each constraint is a bijection.

8.2.2 Max Cut

Given an undirected graph $G = (V, E)$, the Max Cut problem is to find a bipartition of the vertices that maximizes the weight of the edges with endpoints on opposite sides of the partition.

We can represent this problem as a special case of Linear Equations $\bmod\ p$ and therefore as a special case of Unique Games. For each edge $(i, j) \in E$, we write the equation $x_i - x_j \equiv 1 \pmod 2$. Note that the domain size is two, since there are two possible labels, 0 and 1.

8.3 Satisfiable vs Almost Satisfiable Instances

If an instance of Unique Games is satisfiable, it is easy to find an assignment that satisfies all of the constraints. Can you see why? Essentially, the uniqueness property says that if you know the correct label of one variable, then you know the labels of all the neighboring variables. So we can just guess all possible labels for a variable; at some point your guess is correct and this propagates correct labels to all neighbors, and to their neighbors, and so on. This is a generalization of saying that if a graph is bipartite (e.g. all equations in the Max Cut problem are simultaneously satisfiable), then such a bipartition can be found efficiently. So when all constraints in an instance of Unique Games are satisfiable, this is an "easy" problem.

In contrast, the following problem has been conjectured to be "hard": If 99% of the constraints are satisfiable, can we satisfy 1% of the constraints? The precise form of the conjecture is known as the Unique Games Conjecture [Kho02]: For all small constants $\varepsilon, \delta > 0$, given an instance of Unique Games where $1 - \varepsilon$ of the constraints are satisfied, it is hard to satisfy a δ fraction of satisfiable constraints, for some $k > f(\varepsilon, \delta)$, where k is the size of the domain and f is some function of ε and δ.

How does f grow as a function of ε and δ? We claim that $f(\varepsilon, \delta) > 1/\delta$. This is because it can easily be shown that we can satisfy a $1/k$ fraction of the constraints: Randomly assigning a label to each variable achieves this guarantee. Thus, in words, the conjecture is that for a sufficiently large domain size, it is hard to distinguish between almost satisfiable and close to unsatisfiable instances.

8.3.1 Almost Satisfiable Instances of Max Cut

We can also consider the Max Cut problem from the viewpoint of distinguishing between almost satisfiable and close to unsatisfiable instances. However, for this problem, a conjecture as strong as that stated above for general Unique Games is clearly false. This is because we can always satisfy at least half of the equations. (See Sanjeev's lecture.) We now consider the problem of satisfying the maximum number of constraints given that a $(1 - \varepsilon)$ fraction of the constraints are satisfiable. We write the standard semidefinite programming (SDP) relaxation in which each vertex u (with a slight abuse of notation) is represented by a unit vector, u.

$$\max \quad \sum_{(u,v)\in E} \frac{1 - u \cdot v}{2}$$
$$\begin{aligned} u \cdot u &= 1 \quad \forall u \in V \\ u &\in \mathbb{R}^n \quad \forall u \in V. \end{aligned}$$

For a fixed instance of the Max Cut problem, let OPT denote the fraction of constraints satisfied by an optimal solution, and let OPT_{SDP} denote the value of the objective function of the above SDP on this instance. If $OPT \geq (1-\varepsilon)|E|$, then $OPT_{SDP} \geq (1-\varepsilon)|E|$, since $OPT_{SDP} \geq OPT$. In Lecture 1 (Sanjeev's lecture), it was shown that using the random hyperplane rounding of Goemans-Williamson [GW95], we can obtain a .878-approximation algorithm for this problem. We will now try to analyze this algorithm for the case when OPT is large, e.g. at least $(1-\varepsilon)|E|$. From a solution to the above SDP, we obtain a collection of n-dimensional unit vectors, where $n = |V|$. We choose a random hyperplane, represented by a vector $r \in N(0,1)^n$ (i.e. each coordinate is chosen according to the normal distribution with mean 0 and variance 1). Each vector $u \in V$ has either a positive or a negative dot product with the vector r, i.e $r \cdot u > 0$ or $r \cdot u < 0$. Let us now analyze what guarantee we can obtain for the algorithm in terms of ε.

As previously stated, we have the following inequality for the SDP objective function:

$$\sum_{(u,v)\in E} \frac{(1 - u \cdot v)}{2} \quad \geq \quad (1-\varepsilon)|E|.$$

Let θ'_{uv} represent the angle between vectors u and v, i.e. $\arccos(u \cdot v)$. Let θ_{uv} denote the angle $(\pi - \theta'_{uv})$. Then we can rewrite the objective function of the SDP as:

$$\sum_{(u,v)\in E} \frac{1 + \cos(\theta_{uv})}{2}.$$

Further rewriting of the objective function results in the following:

$$\begin{aligned} \sum_{(u,v)\in E} \frac{1 + \cos(\theta_{uv})}{2} &= \sum_{(u,v)\in E} 1 - \frac{1 - \cos(\theta_{uv})}{2} \\ &= |E| - \sum_{(u,v)\in E} \frac{1 - \cos(\theta_{uv})}{2} \\ &= |E| - \sum_{(u,v)\in E} \sin^2(\frac{\theta_{uv}}{2}) \\ &\geq |E| - \varepsilon|E|. \end{aligned}$$

We say that vertices u and v are "cut" if they fall on opposite sides of the bipartition after rounding.

$$\Pr[u \text{ and } v \text{ cut}] \;=\; \frac{\theta'_{uv}}{\pi} \;=\; 1 - \frac{\theta_{uv}}{\pi}.$$

The expected size of S—the number of edges cut in a solution—is:

$$\begin{aligned}
\mathrm{E}[S] &= \sum_{(u,v)\in E} 1 - \frac{\theta_{uv}}{\pi} \\
&= |E| - \sum_{(u,v)\in E} \frac{\theta_{uv}}{\pi}.
\end{aligned}$$

Assume for all $(u, v) \in E$ that $\sin^2(\frac{\theta_{uv}}{2}) = \varepsilon$. Then $\sin(\frac{\theta_{uv}}{2}) = \sqrt{\varepsilon}$. For small θ, we have that $\sin(\theta) \approx \theta$. Therefore, $\theta_{uv}/2 \approx \sqrt{\varepsilon}$.

Thus, the expected value $\mathrm{E}[S] \geq |E|(1 - c\sqrt{\varepsilon})$ for some constant c. In other words, if we are given a Max Cut instance with objective value $(1 - \varepsilon)|E|$, we can find a solution of size $(1 - c\sqrt{\varepsilon})|E|$. In other words, an almost satisfiable instance can be given an almost satisfying assignment, although the assignment has a weaker guarantee.

8.4 General Unique Games

What happens for a large domain? How do we write an SDP for this problem? Before we had just one vector per vertex. Now for each variable, we have k values. So we have a vector for each variable and for each value that it can be assigned. First, we will write a $\{0, 1\}$ integer program for Unique Games and then we relax this to obtain an SDP relaxation.

8.4.1 Integer Program for Unique Games

Recall that L is a set of k labels. For each variable u and each label $i \in L$, let u_i be an indicator variable that is 1 if u is assigned label i and 0 otherwise. Note that the expression in the objective function is 1 exactly when a constraint π_{uv} is satisfied.

$$\begin{aligned}
\max \quad & \sum_{(u,v)\in E} \sum_{i\in L} u_i \cdot v_{\pi_{uv}(i)} \\
\sum_{i\in L} u_i &= 1 \quad \forall u \in V.
\end{aligned}$$

Now we move to a vector program. The objective function stays the same, but we can add some more equalities and inequalities to the relaxation that are valid for an integer program. Below, we write quadratic constraints since our goal is ultimately to obtain a quadratic program.

$$\begin{aligned}
\sum_{i\in L} u_i \cdot u_i &= 1 \quad \forall u \in V,\ i \in L, \\
u_i \cdot u_j &= 0 \quad \forall u \in V,\ i \neq j \in L.
\end{aligned}$$

Additionally, we can also add triangle-inequality constraints on triples of vectors, $\{u_i, v_j, w_h\}$ for $u, v, w \in V$ and $i, j, h \in L$:

$$||u_i - w_h||^2 \leq ||u_i - v_j||^2 + ||v_j - w_h||^2, \tag{8.4.1}$$

$$||u_i - v_j||^2 \geq ||u_i||^2 - ||v_j||^2. \tag{8.4.2}$$

These constraints are easy to verify for 0/1 variables, i.e. for integer solutions. Note that these constraints are not necessary for the integer program, but they make the SDP relaxation stronger.

8.4.2 Trevisan's Algorithm

We now look at an algorithm due to Trevisan [Tre08]. Recall that if we know that every constraint in a given instance is satisfiable, then we can just propagate the labels and obtain a satisfiable assignment. The algorithm that we discuss is roughly based on this idea.

How can we use a solution to the SDP relaxation to obtain a solution that satisfies many constraints? Suppose that OPT is $|E|$ and consider two vertices u and v connected by an edge. In this case, the set of k vectors corresponding to u is the same constellation of k vectors corresponding to vertex v, possibly with a different labeling. If OPT is $(1-\varepsilon)|E|$, then although these two constellations may no longer be identical, they should be "close". The correlation of the vectors corresponds to the distance, i.e. high correlation corresponds to small distance. Thus, we want to show that the vector corresponding to the label of the root vertex r is "close" to other vectors, indicating which labels to assign the other vertices.

8.4.2.1 An Algorithm for Simplified Instances

Consider the following "simplified instance". Recall that the constraint graph consists of a vertex for each variable and has an edge between two variables if there is a constraint between these two variables. Suppose the constraint graph has radius d: there exists a vertex r such that every variable is a distance at most d from vertex r. The following lemma can be proved using the ideas discussed above.

Lemma 8.1. *If every edge contributes $1-\varepsilon/8(d+1)$ to the SDP objective value, then it is possible to efficiently find an assignment satisfying a $(1-\varepsilon)$-fraction of the constraints.*

We now give the steps of the rounding algorithm.

Rounding the SDP

(i) Find root vertex, r, such that every other vertex is reachable from r by a path of length at most d.

(ii) Assign label i to r with probability $||r_i||^2$.

(iii) For each $u \in V$, assign u label j, where j is the label that minimizes the quantity $||u_j - r_i||^2$.

As mentioned earlier, the intuition for this label assignment is that u_j is the vector that is "closest" to r_i. We now prove the following key claim: For each edge (u, v), the probability that constraint π_{uv} is satisfied is at least $1-\varepsilon$. In particular, recall that edge (u, v) is mislabeled if $\ell(v) \neq \pi_{uv}(\ell(u))$. Thus, we want to show that the probability that edge (u, v) is mislabeled is at most ε.

Since r is at most a distance d from all other vertices, a BFS tree with root r has the property that each u has a path to r on the tree of distance at most d. Fix a BFS tree and consider the path from r to u: $r = u^0, u^1, u^2, \ldots, u^{t-1}, u^t = u$, where $t \leq d$. Let π_{u^1} denote the permutation π_{u^0,u^1}, and recursively define π_{u^k} as the composition of permutations $(\pi_{u^k,u^{k-1}}) \cdot (\pi_{u^{k-1}})$. Let $\pi_v = (\pi_{uv}) \cdot (\pi_u)$. We now compute the probability that vertex u is assigned label $\pi_{u(i)}$ and that vertex v is assigned label $\pi_{v(i)}$, given that r is assigned label i. Note that if both these assignments occur, then edge (u, v) is satisfied. (Since edge (u, v) may also be satisfied with another assignment, we can think of our calculation as possibly being an *underestimate* on the probability that edge (u, v) is satisfied.)

Let $A(u)$ denote the label assigned to vertex u by the rounding algorithm. We will show:

$$\Pr[A(u) = \pi_u(i)] \;\geq\; 1 - \frac{\varepsilon}{2} \quad \text{and} \quad \Pr[A(v) = \pi_v(i)] \;\geq\; 1 - \frac{\varepsilon}{2}.$$

This implies that the probability that constraint π_{uv} is satisfied is at least $1 - \varepsilon$. Now we compute the probability that $A(u) \neq \pi_u(i)$. Suppose that u_j for $j \neq \pi_u(i)$ is closer to vector r_i than $u_{\pi_u(i)}$ is. In other words, suppose:

$$||u_j - r_i||^2 \;\leq\; ||u_{\pi_u(i)} - r_i||^2. \tag{8.4.3}$$

Let B_u be the set of labels such that if r is assigned label $i \in B_u$, then u is not assigned label $\pi_u(i)$. Note that label j belongs to B_u iff inequality (8.4.3) holds for j. Thus, the probability that u is not labeled with $\pi_u(i)$ is exactly:

$$\Pr[A(u) \neq \pi_u(i)] \;=\; \sum_{i \in B_u} ||r_i||^2.$$

One can verify that if there is some label j such that inequality (8.4.3) holds, then the quantity $||r_i||^2$ is at most $2||r_i - u_{\pi_u(i)}||^2$. This proof makes use of inequalities from the SDP, (8.4.1) and (8.4.2), as well as inequality (8.4.3). (See Lemma 8.4 from [Tre08], which we include in the Appendix.) Recall that each edge in the graph (and thus each edge on the path from r to u in the BFS tree) contributes at most $1 - \varepsilon/8(d+1)$ to the objective value. By triangle inequality, this implies that $\sum_{i \in L} ||r_i - u_{\pi_u(i)}||^2 \leq \varepsilon/4$. Thus, we conclude:

$$\begin{aligned}
\Pr[A(u) \neq \pi_u(i)] &= \sum_{i \in B_u} ||r_i||^2 \\
&\leq 2 \sum_{i \in B_u} ||r_i - u_{\pi_u(i)}||^2 \\
&\leq 2 \sum_{i \in L} ||r_i - u_{\pi_u(i)}||^2 \\
&\leq \frac{\varepsilon}{2}.
\end{aligned}$$

Similarly, we conclude that $\Pr[A(v) \neq \pi_v(i)] \leq \varepsilon/2$, which implies that the probability that constraint π_{uv} is not satisfied is at most ε.

8.4.2.2 Shift Invariant Instances

In the case of Linear Equations $\bmod\ p$, we can add more constraints to the SDP relaxation, which allow for a simplified analysis of the rounding algorithm. For any assignment of labels, we can shift each of the labels by the same fixed amount, i.e, by adding a value $k \in L$ to each label, and obtain an assignment with the same objective value. This property of a solution has been referred to as *shift invariance*. In these instances, the following are valid constraints. Note that $p = |L|$.

$$\begin{aligned} ||u_i||^2 &= \frac{1}{p} \quad u \in V,\ i \in L, \\ u_i \cdot v_j &= u_{i+k} \cdot v_{j+k} \quad u, v \in V,\ i, j, k \in L. \end{aligned}$$

In this case, we obtain a stronger version of Lemma 8.1.

Lemma 8.2. *In a shift invariant instance in which every edge contributes more than $1 - 1/2(d+1)$ to the SDP objective value, it is possible to efficiently find an assignment that satisfies all of the constraints.*

We will show that in this case, the vector r_i is closer to vector $u_{\pi_u(i)}$ than to vector u_j for any label $j \neq \pi_u(i)$. In other words, $r_i \cdot u_{\pi_u(i)} > r_i \cdot u_j$ for all $j \in L$. If each edge contributes more than $1 - 1/2(d+1)$ to the objective value, then $||r_i - u_{\pi_u(i)}||^2 < 1/p$. This implies that $r_i \cdot u_{\pi_u(i)} > 1/2p$. By triangle inequality, we have:

$$\begin{aligned} ||u_j - u_{\pi_u(i)}||^2 &\leq ||u_j - r_i||^2 + ||r_i - u_{\pi_u(i)}||^2 \\ \frac{2}{p} &\leq \frac{2}{p} - 2r_i \cdot u_j + \frac{1}{p} \quad \Rightarrow \\ r_i \cdot u_j &\leq \frac{1}{2p}. \end{aligned}$$

Assuming that vector u_j is closer to r_i than vector $u_{\pi_u(i)}$, we obtain the following contradiction:

$$\frac{1}{2p} < r_i \cdot u_{\pi_u(i)} \leq r_i \cdot u_j \leq \frac{1}{2p}.$$

Note that in the case of shift invariance, r is assigned each label from L with equal probability. Because of shift invariance, it does not actually matter which label r is assigned. Thus, we can just assign r a label i arbitrarily (we no longer need randomization) and then proceed with the rest of the SDP rounding algorithm.

8.4.2.3 Extension to General Instances

Applying this SDP rounding to general graphs may not yield such good results as in Lemmas 8.1 and 8.2, since the radius of an arbitrary graph can be large, and the objective values of the SDP relaxation would therefore have to be very high for the lemmas to be applicable. In order to apply these lemmas, we break the graph into pieces, each with a radius of no more than $O(\log n/\varepsilon)$. Doing this requires throwing out no more than an ε-fraction of the constraints. The following lemma is originally due to Leighton and Rao [LR99] and can also be found in [Tre08].

Lemma 8.3. *For a given graph $G = (V, E)$ and for all $\varepsilon > 0$, there is a polynomial time algorithm to find a subset of edges $E' \subseteq E$ such that $|E'| > (1 - \varepsilon)|E|$, and every connected connected component of E' has diameter $O(\log |E|/\varepsilon)$.*

Using this lemma, we obtain the following guarantee for general instances: Given an instance for which OPT is at least $(1 - c\varepsilon^3/\log n)|E|$, we can efficiently find a labeling satisfying a $1-\varepsilon$ fraction of the constraints. Note that c is an absolute constant. For shift invariant instances, we can satisfy $(1-\varepsilon)|E|$ of the constraints for an instance where OPT is at least $(1 - c\varepsilon^2/\log n)|E|$.

Given a graph, we remove the $\frac{\varepsilon}{3}$ fraction of constraints that contribute the least to the objective value. This leaves us with at least $(1-\varepsilon/3)|E|$ constraints that each contributes at least $1 - 3c\varepsilon^2/\log n$ (or $1 - 3c\varepsilon/\log n$ for shift invariant instances) to the objective value. We can apply Lemma 8.1 (or Lemma 8.2) with $d = \log n/\varepsilon$, satisfying at least $(1-2\varepsilon/3)|E|$ constraints (or $(1-\varepsilon/3)|E|$ constraints).

8.5 Improving the Approximation Ratio

Algorithms with improved approximation guarantees for Unique Games have been presented in [GT06, CMM06]. The latter work gives an algorithm with the following guarantee: Given an instance of Unique Games with a domain size k for which OPT is at least $(1-\varepsilon)|E|$, the algorithm produces a solution that satisfies at least $\max\{1 - \sqrt{\varepsilon \log k},\ k^{-\varepsilon/(2-\varepsilon)}\}$ fraction of the constraints. Furthermore, it has been shown that the existence of an efficient algorithm that can distinguish between instances in which $(1-\varepsilon)|E|$ constraints can be satisfied and those at which less than $k^{-\varepsilon/2}$ constraints can be satisfiable, would disprove the Unique Games Conjecture [KKMO07]. Moreover, it is sufficient to refute the conjecture if this algorithm works only for the special case of Linear Equations mod p. Thus, focusing on shift invariant instances is a reasonable approach.

Additionally, the Unique Games problem has been studied for cases in which the constraint graph is an expander; in an instance in which OPT is at least $(1-\varepsilon)|E|$, one can efficiently find a solution satisfying at least $1 - O(\frac{\varepsilon}{\lambda})$ fraction of the constraints, where λ is a function of the expansion of the graph [AKK+08, MM09].

8.6 Consequences

The interest in the Unique Games Conjecture has grown due to the many strong, negative consequences that have been proved for various optimization problems. Assuming the Unique Games Conjecture, it has been shown that the Goemans-Williamson algorithm for Max Cut (presented in Sanjeev's lecture) achieves the optimal approximation ratio [KKMO07]. More surprisingly, there are many other NP-complete optimization problems for which the best-known approximation guarantees are obtained via extremely simple algorithms. Nevertheless, no one has been able to find algorithms with improved approximation guarantees, even when resorting to sophisticated techniques such as linear and semidefinite programming. Such optimization problems include the Minimum Vertex Cover problem and the Maximum Acyclic Subgraph problem, for which the best-known approximation factors are 1/2 and 2, respectively. If the Unique Games Conjecture is true, then these approximation ratios are tight [KR08, GMR08]. This phenomena has been investigated for several other optimization problems as well. A recent result shows that for a whole class of constraint satisfaction problems, which can be modeled using a particular integer program, the integrality gap of a particular SDP relaxation is exactly equal to its approximability threshold under the Unique Games Conjecture [Rag08].

References

[AKK+08] Sanjeev Arora, Subhash Khot, Alexandra Kolla, David Steurer, Madhur Tulsiani, and Nisheeth K. Vishnoi. *Unique games on expanding constraint graphs are easy: extended abstract*. In *Proc. 40th ACM Symp. on Theory of Computing (STOC)*, pages 21–28. ACM, 2008. doi:10.1145/1374376.1374380.

[CMM06] Moses Charikar, Konstantin Makarychev, and Yury Makarychev. *Near-optimal algorithms for unique games*. In *Proc. 38th ACM Symp. on Theory of Computing (STOC)*, pages 205–214. ACM, 2006. doi:10.1145/1132516.1132547.

[FL92] Uriel Feige and László Lovász. *Two-prover one-round proof systems: Their power and their problems (extended abstract)*. In *Proc. 24th ACM Symp. on Theory of Computing (STOC)*, pages 733–744. ACM, 1992. doi:10.1145/129712.129783.

[GMR08] Venkatesan Guruswami, Rajsekar Manokaran, and Prasad Raghavendra. *Beating the random ordering is hard: Inapproximability of maximum acyclic subgraph*. In *Proc. 49th IEEE Symp. on Foundations of Comp. Science (FOCS)*, pages 573–582. IEEE, 2008. doi:10.1109/FOCS.2008.51.

[GT06] Anupam Gupta and Kunal Talwar. *Approximating unique games*. In *Proc. 17th Annual ACM-SIAM Symposium on Discrete Algorithms (SODA)*, pages 99–106. SIAM, 2006. doi:10.1145/1109557.1109569.

[GW95] Michel X. Goemans and David P. Williamson. *Improved approximation algorithms for maximum cut and satisfiability problems using semidefinite programming*. J. ACM, 42(6):1115–1145, 1995. (Preliminary version in *26th STOC*, 1994). doi:10.1145/227683.227684.

[Kho02] Subhash Khot. *On the power of unique 2-prover 1-round games*. In *Proc. 34th ACM Symp. on Theory of Computing (STOC)*, pages 767–775. ACM, 2002. doi:10.1145/509907.510017.

[KKMO07] Subhash Khot, Guy Kindler, Elchanan Mossel, and Ryan O'Donnell. *Optimal inapproximability results for MAX-CUT and other 2-variable CSPs?* SIAM J. Computing, 37(1):319–357, 2007. (Preliminary version in *45th FOCS*, 2004). eccc:TR05-101, doi:10.1137/S0097539705447372.

[KR08] Subhash Khot and Oded Regev. *Vertex cover might be hard to approximate to within 2-ε*. J. Computer and System Sciences, 74(3):335–349, 2008. (Preliminary Version in *18th IEEE Conference on Computational Complexity*, 2003). doi:10.1016/j.jcss.2007.06.019.

[LR99] Frank Thomson Leighton and Satish Rao. *Multicommodity max-flow min-cut theorems and their use in designing approximation algorithms*. J. ACM, 46(6):787–832, 1999. doi:10.1145/331524.331526.

[MM09] Konstantin Makarychev and Yury Makarychev. *How to play unique games on expanders*, 2009. arXiv:0903.0367.

[Rag08] Prasad Raghavendra. *Optimal algorithms and inapproximability results for every CSP?* In *Proc. 40th ACM Symp. on Theory of Computing (STOC)*, pages 245–254. ACM, 2008. doi:10.1145/1374376.1374414.

[Tre08] Luca Trevisan. *Approximation algorithms for unique games*. Theory of Computing, 4(1):111–128, 2008. (Preliminary version in *46th FOCS*, 2005). eccc:TR05-034, doi:10.4086/toc.2008.v004a005.

8.7 Appendix

We include following lemma from [Tre08] and its proof:

Lemma 8.4. *Let* $\mathbf{r}, \mathbf{u}, \mathbf{v}$ *be vectors such that: (i)* $\mathbf{u} \cdot \mathbf{v} = 0$*, (ii)* $||\mathbf{r} - \mathbf{u}||^2 \geq ||\mathbf{r} - \mathbf{v}||^2$*, and (iii) the vectors* $\mathbf{r}, \mathbf{u}, \mathbf{v}$ *satisfy the triangle inequality constraints from the SDP. Then* $||\mathbf{r} - \mathbf{u}||^2 \geq \frac{1}{2}||\mathbf{r}||^2$.

Proof. There are three cases:

1. If $||\mathbf{u}||^2 \leq \frac{1}{2}||\mathbf{r}||^2$, then by (8.4.2), we have:

$$||\mathbf{r} - \mathbf{u}||^2 \geq ||\mathbf{r}||^2 - ||\mathbf{u}||^2 \geq \frac{1}{2}||\mathbf{r}||^2.$$

2. If $||\mathbf{v}||^2 \leq \frac{1}{2}||\mathbf{r}||^2$, then by (8.4.1), and subsequently (8.4.2), we have:

$$||\mathbf{r} - \mathbf{u}||^2 \geq ||\mathbf{r} - \mathbf{v}||^2 \geq ||\mathbf{r}||^2 - ||\mathbf{v}||^2 \geq \frac{1}{2}||\mathbf{r}||^2.$$

3. If $||\mathbf{u}||^2, ||\mathbf{v}||^2 \geq \frac{1}{2}||\mathbf{r}||^2$, then from (8.4.1) and assumption (ii), we have:

$$||\mathbf{v} - \mathbf{u}||^2 \leq ||\mathbf{v} - \mathbf{r}||^2 + ||\mathbf{r} - \mathbf{u}||^2 \leq 2||\mathbf{r} - \mathbf{u}||^2.$$

   By Pythagoras theorem and by orthogonality of $\mathbf{u}$ and $\mathbf{v}$ (assumption (i)), we have:

$$||\mathbf{v} - \mathbf{u}||^2 = ||\mathbf{v}||^2 + ||\mathbf{u}||^2.$$

   Finally, we have:

$$||\mathbf{r} - \mathbf{u}||^2 \geq \frac{1}{2}||\mathbf{v} - \mathbf{u}||^2 = \frac{1}{2}||\mathbf{v}||^2 + \frac{1}{2}||\mathbf{u}||^2 \geq \frac{1}{2}||\mathbf{r}||^2.$$

□