Label Cover Algorithms via the Log-Density Threshold

Jimmy Wu
jimmyjwu@stanford.edu

June 13, 2017

1 Introduction

Since the discovery of the PCP Theorem and its implications for approximation algorithms, researchers have devised a web of reductions showing the NP-hardness of approximating numerous problems. Many of these results are obtained via reduction from a problem called Label Cover, in which we are given a bipartite graph $G = (L, R, E)$, a constraint π_e on each edge e, and a label set Σ; the goal is to label the vertices so as to satisfy as many constraints as possible. Label Cover is known to be quasi-NP-hard hard to approximate to within a factor of $2^{\log^{1-\varepsilon} N}$, where $N = n|\Sigma|$. Until recently, the best known approximation upper bound was $O(N^{1/4})$. It is an open problem to close this gap, and many believe the true approximability of the problem is N^c for some $0 < c < 1$.

In this report, we summarize a recent paper of Chlamtáč, Manurangsi, Moshkovitz, and Vijayaraghavan [Chl+17] who present evidence that the correct hardness threshold is given by $c \approx 0.17$. To do this, they apply a framework called the "log-density threshold," which leverages insights from the study of distributional problems to devise algorithms for the worst-case setting. In particular, the authors give roughly $N^{0.17}$-approximation algorithms for random and semi-random variants of Label Cover, as well as a worst-case $N^{0.23}$-approximation. Furthermore, they show that the natural Lasserre or sum-of-squares relaxation for Label Cover fails to do much better.

1.1 Preliminaries

1.1.1 Label Cover

δ-Gap Label Cover (which we will simply refer to as Label Cover) is the following problem:

> Label Cover
>
> **Input:** a bipartite graph $G = (L, R, E)$ with left-degree Δ_L and right-degree Δ_R; finite label sets/alphabets Σ_L and Σ_R; for each edge $e \in E$, a constraint function/projection $\pi_e : \Sigma_L \to \Sigma_R$.
>
> A *labeling* of G is defined as a pair of functions $\phi_L : L \to \Sigma_L$ and $\phi_R : R \to \Sigma_R$ that map each vertex to a label. We say ϕ_L, ϕ_R *satisfies* an edge $e = (x, y) \in E$ if $\pi_e(\phi_L(x)) = \phi_R(y)$.
>
> **Task:** distinguish between the following cases:

- (YES/completeness) There is a *total labeling* ϕ_L, ϕ_R that satisfies every edge.
- (NO/soundness) Every labeling ϕ_L, ϕ_R, satisfies at most a δ-fraction of edges.

This formulation of Label Cover is said to have *perfect completeness*, as the YES case is fully satisfiable. When it is clear from context, we will write ϕ instead of ϕ_L or ϕ_R, and Σ instead of Σ_L or Σ_R.

Some further notation: Let $n = |L| + |R|$ be the number of vertices in G. Let $k = |\Sigma_L|$ be the size of the left alphabet, and assume that $|\Sigma_L| \geq |\Sigma_R|$. Let $N = nk$, which we think of as the size of the instance. Let β be the number such that $|L| = N^\beta$.

Label Cover is not a problem known for its practical applications. Rather, its importance stems from the many known *gap reductions* from Label Cover to other gap problems. It is a simple observation that if it is hard to decide a δ-gap problem (e.g. δ-Gap Label Cover), then it is also hard to approximate the corresponding optimization problem (e.g. "find a labeling that satisfies the most edges") to within a factor of $1/\delta$. Thus an increase in the known hardness of δ-Gap Label Cover immediately implies improved inapproximability bounds for other problems[1]. Conversely, improved approximation algorithms for Label Cover, or algorithms for deciding smaller gaps, can be viewed as "lower bounds against lower bounds."

1.1.2 Previous Bounds

A long line of work has gone into establishing inapproximability results for Label Cover. The problem is NP-hard for $\delta = 1/\log^c N$ for every $c > 0$ [DS14]. Under the stronger but still widely-accepted assumption that $\mathsf{NP} \notin \mathsf{DTIME}(n^{O(\log n)})$, it is hard for $\delta = 2^{-\log^{1-\epsilon} n}$ for every $\epsilon > 0$ [Kor01], which is just shy of any polynomial.

On the algorithmic side, Charikar, Hajiaghayi, and Karloff gave an $O(N^{1/3})$-approximation algorithm [CHK09]. For instances of Label Cover that are guaranteed to have total labelings, Manurangsi and Moshkovitz achieved an approximation ratio of $O(N^{1/4})$ [MM17].

This leaves a gap between the upper and lower bounds. Many in the hardness of approximation community believe the true gap is in fact a polynomial:

Conjecture 1 (Projection Games Conjecture). *There is some $c > 0$ such that* Label Cover *with $\delta = \Omega(1/N^c)$ has no polynomial-time algorithm.*

1.1.3 Results

The paper we discuss takes the view of the Projection Games Conjecture, and presents evidence that the right threshold is $c = 3 - 2\sqrt{2} \approx 0.17$. To do this, they first study random and semi-random variants of Label Cover, in which the graph (and in the former, also the constraints) are drawn from some distribution. They show the following result:

Theorem 1.1 (Algorithm for Random and Semi-Random Label Cover). *For every $\varepsilon > 0$, there is an $N^{O(1/\varepsilon^2)}$-time algorithm for* Label Cover *with $\delta = N^{-\beta(1-\beta)/(2-\beta)-\varepsilon}$ on random graphs with alphabet*

[1] See the 2010 revision of [Tre04] for a survey of such results.

size k, where $N = |L|k$ and $|L| = N^{\beta}$. The algorithm is correct with high probably over the choice of G. For the worst β, $\delta = N^{-(3-3\sqrt{2})} \approx N^{-0.1716}$.

For the fully worst-case setting (with perfect completeness), we have the following:

Theorem 1.2 (Algorithm for worst-case Label Cover)**.** *For every $\varepsilon > 0$, there is a polynomial-time algorithm for* Label Cover *with $\delta = O(N^{-\frac{1}{6}(5-\sqrt{13})-\varepsilon}) \approx O(N^{-0.2325})$.*

Additionally, the authors show strong integrality gaps for the natural Lasserre/sum-of-squares relaxation for the problem:

Theorem 1.3 (Lasserre integrality gaps)**.** *For every $0 < \varepsilon < 1/8$, the integrality gap of the $N^{\Omega(\varepsilon)}$-level Lasserre SDP relaxation of* Label Cover *of size N is at least $N^{1/8-\varepsilon}$.*

2 Algorithms via Log-Density

The *log-density method* is a framework for devising approximation algorithms, used by Bhaskara et al. [Bha+10] to obtain improved upper bounds for the Densest k-Subgraph problem. In the present paper, this framework is implemented as follows:

1. (Distributional problems) Define an appropriate *random* or *semi-random* version of Label Cover, in which the task is to distinguish a random instance from a random instance with a planted solution.

2. (Witnesses) Try to devise a good approximation algorithm for this random problem by looking for *witnesses*: small subgraphs that exist in planted instances but not in fully random ones.

3. (Log-density threshold) There arises a threshold in the graph parameters called the *log-density threshold*, above which witnesses exist and algorithms are possible, and below which they do not.

4. (Lift to worst-case) Use insights gleaned from the above steps to devise an improved algorithm for the original worst-case setting.

2.1 The Random Model

We now focus on a natural distributional formulation of Label Cover:

Random Label Cover

Input: a random bipartite graph $G = (L, R, E)$ with average left-degree Δ_L and right-degree Δ_R; finite alphabets Σ_L and Σ_R; for each edge $e \in E$, a constraint function $\pi_e : \Sigma_L \to \Sigma_R$.

Task: distinguish between the following cases:

- (YES/completeness) There is a *total labeling* ϕ_L, ϕ_R that satisfies every edge. The constraints $\{\pi_e\}$ are uniformly random conditioned on this labeling.
- (NO/soundness) The constraints $\{\pi_e\}$ are uniformly random.

For simplicity, we restrict our attention to a highly regular case in which $|L| = |R| = n/2$, $\Delta := \Delta_L = \Delta_R$, and the constraints are d-to-1 functions—thus $|\Sigma_R| = k/d$.

Definition 2.1 (log-densities)**.** *The* log-density of the graph *is* $\log(\Delta^2)/\log|V|$. *The* log-density of the projections *is* $\log d/\log k$. *The* log-density gap *is their difference:* $2\log\Delta/\log n - \log d/\log k$.

2.1.1 A Witness-Counting Algorithm

We now discuss a way to distinguish the two cases when we have a log-density gap. The algorithm is based on looking for *witnesses*—small subgraphs of G that exist in large numbers and can be consistently labeled when G is a YES instance, but may not be numerous or satisfiable in the NO case. In particular, the kind of subgraph we look for is a special type of tree called a *caterpillar graph*:

Definition 2.2 (witness/caterpillar (informal))**.** *Given integers $r < s$, the caterpillar $\mathcal{W}_{r,s}$ is a subtree of G with $r+1$ leaves in L, $s-r$ internal vertices in L, and s internal vertices in R.*

A more detailed definition actually depends on the algorithm in use, which we omit in this report.

Suppose that there is a positive log-density gap. In particular, let $d \le k^\gamma$ and $\Delta \ge n^{\alpha/2}$, where $\gamma < \alpha$ are constants. The gap is then $\alpha - \gamma > 0$. We can find integers r, s and constants $\varepsilon_1, \varepsilon_2 > 0$ such that $\gamma + \varepsilon_1 \le r/s \le \alpha - \varepsilon_2$, then apply the following witness-based algorithm:

---

function $\textsc{Alg-Dist}_{r,s}(G, \Sigma, \{\pi_e\})$
  for each tuple $U = (u_0, u_1, \ldots, u_r) \subset L$ and assignment $(\sigma_0, \sigma_1, \ldots, \sigma_r)$ to U **do**
    Check if G has at least $\log^{2s} n$ occurrences of $\mathcal{W}_{r,s}$ with U as its leaves.
    for the first $O(\log n)$ occurrences $H \subseteq G$ of $\mathcal{W}_{r,s}$ **do**
      Check if there is a satisfying assignment to the vertices of H that is consistent with assigning $(\sigma_0, \sigma_1, \ldots, \sigma_r)$ to U.
    if all the above checks are TRUE **then return** PLANTED
  return RANDOM

Enumerating all occurrences and trying all assignments takes time at most $N^{O(s+r)}$, which is polynomial since $\mathcal{W}_{r,s}$ is of constant size.

The correctness of $\textsc{Alg-Dist}_{r,s}$ can be proved as follows:

Theorem 2.1. *Consider the* Random Label Cover *problem with parameters $\Delta = n^{\alpha/2}$ and $d = k^\gamma$. Let $\varepsilon_1, \varepsilon_2 > 0$ and integers r, s satisfy $\gamma + \varepsilon_1 \le r/s \le \alpha - \varepsilon_2$. Then $\textsc{Alg-Dist}_{r,s}$ distinguishes YES and NO instances with high probability.*

Proof sketch. Consider a random graph G with average degree $\Delta > n^{\frac{r}{2s}+\varepsilon_2}$. It can be shown by induction that with high probability, for any fixed set U of $r+1$ leaves, the number of occurrences of $\mathcal{W}_{r,s}$ in G is at least $n^{\varepsilon_2 s}$.

Suppose we are given a YES instance with total labeling ϕ^*. Clearly no matter how we fix a set of leaves U, we can satisfy every occurrence H of $\mathcal{W}_{r,s}$ supported on U, by picking the assignment consistent with ϕ^*.

On the other hand, suppose we are given a NO instance, in which $d \leq k^{r/s-\varepsilon_2}$ by assumption. Let H be an occurrence of $\mathcal{W}_{r,s}$ supported on a leaf set U. Then, for any fixed assignment $\sigma_0, \ldots, \sigma_r$ to U, we have Pr[there is a satisfying assignment for H that agrees with $\{\sigma_i\}] \leq k^{-\varepsilon_1 r}$.

But as we stated before, $\mathcal{W}_{r,s}$ has many occurrences in G. For a fixed assignment $\sigma_0, \ldots, \sigma_r$ to U, the probability that this assignment can be extended to one that satisfies at least $\Omega(\log n)$ other occurrences is at most $k^{-\varepsilon_1 r \log n}$. Thus by inspecting $O(\log n)$ occurrences, our algorithm is likely is catch an inconsistency. Finally, union bounding over all k^{r+1} assignments to U completes the argument. □

2.1.2 Simple Algorithms and Deriving the Threshold

What can be done if the log-density gap is non-positive? Note that Label Cover has the following simple approximation algorithms:

- By finding a perfect matching and labeling the vertices so as to satisfy those edges, we satisfy a $1/\Delta$-fraction of the edges.
- By labeling the vertices in R randomly and then labeling the L vertices greedily, we satisfy a d/k-fraction.

Thus we can always bet a $\min(\Delta, k/d)$-approximation. Now suppose there is no log-density gap, so $2 \log \Delta / \log n \leq \log d / \log k$. If $d = k^\gamma$, then by balancing the approximation ratios of the two simple algorithms, calculations show that the worst setting of γ is $\gamma = \frac{2-2\beta}{2-\beta}$, at which point (and after some more calculations) we get an approximation ratio of N^c, where $c = \beta(1-\beta)/(2-\beta)$. The largest this can be over all $0 \leq \beta \leq 1$ is $c = 3 - 2\sqrt{2}$. Naturally, we call this the *log-density threshold* for Label Cover.

2.2 Necessity of the Log-Density Gap

We just showed that if there is a positive log-density gap, then simple algorithms suffice to well-approximate Random Label Cover. We now explain why this gap is essentially necessary for local algorithms.

Consider any algorithm that works by looking for occurrences $H \subseteq G$ of a constant-size subgraph $\mathcal{W}$ ($\textsc{Alg-Dist}_{r,s}$ is an example). If we are given a YES instance, then clearly every H can be simultaneously satisfied. We thus desire the following properties:

1. When G is a random degree-Δ graph, $\mathcal{W}$ appears in G.
2. For NO instances, with high probability every assignment to each occurrence H of $\mathcal{W}$ fails to satisfy all constraints.

Suppose $\mathcal{W}$ contains a vertices, a_L of which are in L and a_R of which are in R, and b edges. Then to satisfy property (1) in expectation, we need:

$$\mathbb{E}[\#\text{ of occurrences of }\mathcal{W}] \approx \left(\frac{n}{2}\right)^a \left(\frac{\Delta}{n}\right)^b > 1 \tag{2.1}$$

Now, fix one occurrence H. For each edge $e = (u, v) \in E$, u has k labels and v has k/d labels, so each label assignment to u and v satisfies π_e with probability d/k. Thus satisfying property (2) requires

$$\mathbb{E}[\text{\# of satisfying assignments for } H] = k^{a_L} \left(\frac{k}{d}\right)^{a_R} \left(\frac{d}{k}\right)^{b} = k^{a} \left(\frac{d^{b-a_R}}{k^b}\right) < 1 \tag{2.2}$$

Combining these requirements leads to

$$\frac{b-a}{b} < \frac{\log \Delta}{\log n} \qquad \text{and} \qquad \frac{b-a}{b-a_R} > \frac{\log d}{\log k} \tag{2.3}$$

All of these calculations have been in expectation. An additional requirement on which we will not elaborate is that for the distinguisher to succeed *with high probability*, it turns out that we need every vertex in $\mathcal{W}$ to have degree at least 2, implying that $a_R \leq b/2$. This gives

$$\frac{b-a}{b} < \frac{\log \Delta}{\log n} \qquad \text{and} \qquad \frac{b-a}{b/2} > \frac{\log d}{\log k} \tag{2.4}$$

from which we conclude that

$$\frac{\log(\Delta^2)}{\log n} > \frac{\log d}{\log k} \tag{2.5}$$

is necessary for $\mathcal{W}$ to serve as a witness. Thus the log-density threshold, as we have defined it, corresponds precisely to a phase transition phenomenon at which witness-based algorithms succeed.

2.3 From Random to Worst-Case

Finally we discuss the strategy of [Chl+17] for lifting the techniques for Random Label Cover to worst-case Label Cover. Due to space constraints, we will limit ourselves to a high-level outline.

The first order of business is to go beyond the "distinguish planted vs. random" setup and obtain an algorithm for finding a good labeling. We saw that when there is no log-density advantage (i.e. $2 \log \Delta / \log n \leq \log d / \log k$), the simple algorithms give the desired approximation ratio. When there is a log-density gap however, we have only shown how to decide which distribution the instance is from. Towards getting an approximation for this regime, the authors prove the following:

Lemma 2.1 (alphabet reduction (informal))**.** *If $2 \log \Delta / \log n \geq \log d / \log k + \epsilon$, then in polynomial time we can find a sparsified alphabet Σ_u for each vertex $u \in V$ of size roughly $|\Sigma|^{1-\epsilon}$.*

This sparsification procedure is guided by the witnesses, which as we argued, must exist when there is a positive log-density gap. After one or more alphabet reductions, the log-density gap will be eliminated and the simple algorithms once again give us the desired approximation[2].

[2]In the precise statement of this lemma, we actually can only find a *set* of such alphabet reductions, at least one of which induces a Label Cover instance that is still fully satisfiable. We must then recurse on all of them, though since the recursion depth is only a constant, the algorithm is still polynomial-time.

With some modifications, this alphabet reduction enables a similar witness-based algorithm for a Semi-Random Label Cover variant, in which G is still random, but the projections $\{\pi_e\}$ are arbitrary. At this point, Theorem 1.1 has been proved.

Finally, the authors show an improved algorithm for worst-case Label Cover. The main effort is in establishing the following improvement lemma:

Lemma 2.2. *If there is a polynomial-time $\tilde{O}(N^\alpha)$-approximation algorithm for* Label Cover *for some $0 < \alpha \leq 1$, then there is also a polynomial-time $\tilde{O}(N^{\frac{1}{5-3\alpha}})$-approximation algorithm.*

We omit the highly technical proof of this lemma. The main obstacle is the following: in the random and semi-random scenarios, the alphabet reduction procedure requires that G is a good expander (as random graphs are with high probability). Worst-case graphs may not be; however, if the alphabet reduction step fails, we learn that G is not a good expander. This in turn allows us to partition the graph into dense components; we then approximate the instance induced by each component and merge their solutions. Since the components are relatively dense, the cuts between them are sparse, so not too many constraints are violated by merging.

With the above lemma in hand, we can prove the main theorem:

Proof of Theorem 1.2. Consider a series of algorithms, where the first algorithm is the $O(N^{1/4})$-approximation algorithm of [MM17]. In general the i-th algorithm is obtained by applying Lemma 2.2 to the $(i-1)$-th, and has ratio $\tilde{O}(N^{\alpha_i})$ where $a_i = \frac{1}{5-3\alpha_{i-1}}$. The sequence $\{\alpha_i\}_{i=1,2,\ldots}$ converges to $\frac{1}{6}(5-\sqrt{13})$ as required. □

3 Integrality Gaps

Here we discuss the integrality gap for Label Cover stated in Theorem 1.3. As a bibliographic note, this material is not present in the conference version of [Chl+17], though it is presented in detail in Pasin Manurangsi's MEng thesis [Man15].

3.1 Lasserre Relaxations for Label Cover and Max K-CSP

We assume that $\Sigma_L = \Sigma_R = [k]$. For each $S \subseteq V$, an assignment $\alpha \in [k]^S$ is viewed as a mapping from S to $[k]$. If $S' \subseteq S$, then $\alpha(S') \in [k]^{S'}$ is the assignment α restricted to S'.

Given two sets S_1, S_2 and two assignments α_1, α_2, the assignment induced by α_1 and α_2 on $S_1 \cup S_2$ is $\alpha_1 \circ \alpha_2 \in [k]^{S_1 \cup S_2}$ given by

$$\alpha_1 \circ \alpha_2(j) = \begin{cases} \alpha_1(j), & j \in S_1 \\ \alpha_2(j), & j \notin S_1 \end{cases}$$

The r-th level Lasserre SDP for Label Cover contains a vector $U_{(S,\alpha)}$ for each $S \subseteq V$ such that $|S| \leq r$, and each $\alpha \in [k]^S$. A useful interpretation is that $\left\|U_{(S,\alpha)}\right\|^2$ is roughly the probability that α is the assignment chosen for set S. Formally, the relaxation is the following:

$$\text{maximize} \quad \sum_{(u,v)\in E} \sum_{\sigma\in[k]} \left\| U_{(\{u,v\},\{u\to\sigma, v\to\pi_{u,v}(\sigma)\})} \right\|^2 \tag{3.1}$$

$$\text{subject to} \quad \left\| U_{(\emptyset,\emptyset)} \right\| = 1 \tag{3.2}$$

$$\langle U_{(S_1,\alpha_1)}, U_{(S_1,\alpha_1)} \rangle = 0 \qquad \forall S_1, S_2, \alpha_1, \alpha_2 : \alpha_1(S_1) \neq \alpha_2(S_2) \tag{3.3}$$

$$\langle U_{(S_1,\alpha_1)}, U_{(S_2,\alpha_2)} \rangle = \langle U_{(S_3,\alpha_3)}, U_{(S_4,\alpha_4)} \rangle \qquad \forall S_1, S_2, S_3, S_4, \alpha_1, \alpha_2, \alpha_3, \alpha_4 : \alpha_1 \circ \alpha_2 = \alpha_3 \circ \alpha_4 \tag{3.4}$$

$$\langle U_{(S_1,\alpha_1)}, U_{(S_1,\alpha_1)} \rangle \geq 0 \qquad \forall S_1, S_2, \alpha_1, \alpha_2 \tag{3.5}$$

$$\sum_{\sigma\in[k]} \left\| U_{(\{v\},\sigma)} \right\|^2 = 1 \qquad \forall v \in V \tag{3.6}$$

Here, $\{u \to \sigma, v \to \pi_{u,v}(\sigma)\}$ denotes the function that maps u to σ and v to $\pi_{(u,v)}(\sigma)$. Note also that all sets S_i above are of size at most r.

[Chl+17] prove their integrality gaps for this Label Cover SDP by reducing from Max K-CSP, which is known to have large gaps [Tul09].

> Max K-CSP
>
> **Input:** A set of variables $\chi = \{x_1, \ldots, x_n\}$; a set of constraints $\mathcal{C} = \{C_1, \ldots, C_m\}$ where each C_i is a mapping $C_i : [q]^{T_i} \to \{0, 1\}$ for some $T_i \subseteq \chi$ of size K.
>
> **Output:** An assignment $\varphi : \chi \to [q]$ that maximizes the number of satisfied constraints $\sum_{i\in[m]} C_i(\varphi|_{T_i})$.

The r-th level Lasserre relaxation for Max K-CSP is very similar to the one for Label Cover:

$$\text{maximize} \quad \sum_{i\in[m]} \sum_{\alpha\in[q]^{T_i}} C_i(\alpha) \left\| V_{(T_i,\alpha)} \right\|^2 \tag{3.7}$$

$$\text{subject to} \quad \left\| V_{(\emptyset,\emptyset)} \right\| = 1 \tag{3.8}$$

$$\langle V_{(S_1,\alpha_1)}, V_{(S_1,\alpha_1)} \rangle = 0 \qquad \forall S_1, S_2, \alpha_1, \alpha_2 : \alpha_1(S_1) \neq \alpha_2(S_2) \tag{3.9}$$

$$\langle V_{(S_1,\alpha_1)}, V_{(S_2,\alpha_2)} \rangle = \langle U_{(S_3,\alpha_3)}, U_{(S_4,\alpha_4)} \rangle \qquad \forall S_1, S_2, S_3, S_4, \alpha_1, \alpha_2, \alpha_3, \alpha_4 : \alpha_1 \circ \alpha_2 = \alpha_3 \circ \alpha_4 \tag{3.10}$$

$$\langle V_{(S_1,\alpha_1)}, V_{(S_1,\alpha_1)} \rangle \geq 0 \qquad \forall S_1, S_2, \alpha_1, \alpha_2 \tag{3.11}$$

$$\sum_{j\in[q]} \left\| V_{(\{i\},j)} \right\|^2 = 1 \qquad \forall v \in V \tag{3.12}$$

3.2 Reduction from Max K-CSP to Label Cover

Though the analysis of the gap is tedious, the reduction is natural, and proceeds from *random* instances of Max K-CSP. Roughly, in a random CSP instance, constraints over K variables are sampled uniformly at random subject to some requirements imposed by a linear code C. The reduction goes as follows:

Given a random Max K-CSP(C) instance $\Phi = (\chi, \mathcal{C})$, construct a Label Cover instance $(L, R, E, \Sigma_L, \Sigma_R, \{\pi_e\})$ as follows:

- Place the constraints on the left and variables on the right; i.e. $L = \mathcal{C}$ and $R = \chi$.
- Identify Σ_L with the set of codewords in C; i.e. $\Sigma_L = [|C|]$. For each constraint vertex $C_i \in L$, each element of Σ_L can be viewed as a locally satisfying assignment α such that $C_i(\alpha) = 1$.
- Let there be an edge $(C_i, x_j) \in L \times R$ if and only if $x_j \in T_i$. For each such edge (C_i, x_j), let the projection $\pi_{(C_i,x_j)} : \Sigma_L \to \Sigma_R$ be given by $\pi_{(C_i,x_j)}(\alpha) = \alpha(x_j)$.

[Chl+17] then argue the following:

Lemma 3.1 (completeness)**.** *If the* Max K-CSP(C) *instance admits a perfect vector solution to the r-level Lasserre relaxation, then the* Label Cover *instance obtained via the above reduction admits a perfect vector solution for its respective r/K-level Lasserre relaxation.*

Note that there is some loss in the required level of the relaxation. Conversely we have:

Lemma 3.2 (soundness (informal))**.** *With probability $1 - o(1)$, the optimal solution to the* Label Cover *instance produced by the reduction satisfies at most $O(\log q/q)$ edges.*

Thus the Lasserre optimum is far from the true optimum and Theorem 1.3 can be deduced by appropriate calculations using the known gap for Max K-CSP [Tul09].

References

[Bha+10] A. Bhaskara, M. Charikar, E. Chlamtac, U. Feige, and A. Vijayaraghavan. "Detecting high log-densities: an O (n 1/4) approximation for densest k-subgraph". In: *Proceedings of the forty-second ACM symposium on Theory of computing*. ACM. 2010, pp. 201–210.

[CHK09] M. Charikar, M. Hajiaghayi, and H. Karloff. "Improved approximation algorithms for label cover problems". In: *European Symposium on Algorithms*. Springer. 2009, pp. 23–34.

[Chl+17] E. Chlamtáč, P. Manurangsi, D. Moshkovitz, and A. Vijayaraghavan. "Approximation algorithms for label cover and the log-density threshold". In: *Proceedings of the Twenty-Eighth Annual ACM-SIAM Symposium on Discrete Algorithms*. SIAM. 2017, pp. 900–919.

[DS14] I. Dinur and D. Steurer. "Analytical approach to parallel repetition". In: *Proceedings of the 46th Annual ACM Symposium on Theory of Computing*. ACM. 2014, pp. 624–633.

[Kor01] G. Kortsarz. "On the hardness of approximating spanners". In: *Algorithmica* 30.3 (2001), pp. 432–450.

[Man15] P. Manurangsi. "On approximating projection games". PhD thesis. Massachusetts Institute of Technology, 2015.

[MM17] P. Manurangsi and D. Moshkovitz. "Improved approximation algorithms for projection games". In: *Algorithmica* 77.2 (2017), pp. 555–594.

[Tre04] L. Trevisan. "Inapproximability of combinatorial optimization problems". In: *The Computing Research Repository* (2004).

[Tul09] M. Tulsiani. "CSP gaps and reductions in the Lasserre hierarchy". In: *Proceedings of the forty-first annual ACM symposium on Theory of computing*. ACM. 2009, pp. 303–312.