Near-optimal approximation algorithm for simultaneous MAX-CUT

Amey Bhangale * Subhash Khot† Swastik Kopparty ‡ Sushant Sachdeva§
Devanathan Thiruvenkatachari¶

November 1, 2017

Abstract

In the simultaneous MAX-CUT problem, we are given k weighted graphs on the same set of n vertices, and the goal is to find a cut of the vertex set so that the minimum, over the k graphs, of the cut value is as large as possible. Previous work [BKS15] gave a polynomial time algorithm which achieved an approximation factor of $1/2 - o(1)$ for this problem (and an approximation factor of $1/2 + \varepsilon_k$ in the unweighted case, where $\varepsilon_k \to 0$ as $k \to \infty$).

In this work, we give a polynomial time approximation algorithm for simultaneous MAX-CUT with an approximation factor of 0.8780 (for all constant k). The natural SDP formulation for simultaneous MAX-CUT was shown to have an integrality gap of $1/2+\varepsilon_k$ in [BKS15]. In achieving the better approximation guarantee, we use a stronger Sum-of-Squares hierarchy SDP relaxation and a rounding algorithm based on Raghavendra-Tan [RT12], in addition to techniques from [BKS15].

1 Introduction

In this paper, we give near-optimal approximation algorithms for the simultaneous MAX-CUT problem. Here we are given a collection of weighted graphs $G_1, G_2, \ldots, G_k$ on the same vertex set V of size n. Our goal is to find a partition of the vertex set V into two parts, such that in *every* graph, the total weight of edges going between the two parts is large. The $k = 1$ case is the classical MAX-CUT problem, and the approximability of this problem has been extensively studied [FL92, GW95, Hås01, KKMO07, MOO05, OW08]. This paper studies the approximability of this problem for constant k.

We fix some convenient notation. Let the weighted graphs $G_1, \ldots, G_k$ be given by weight functions $\mathcal{E}_1, \ldots, \mathcal{E}_k$, which assign to each pair in $\binom{V}{2}$ a weight in $[0,1]$. We assume that for each $i \in [k]$, the total weight of all edges under $\mathcal{E}_i$ equals 1. Let $f : V \to \{0,1\}$ be a function, which we view as a partition of the vertex set. We define $\mathsf{val}(f, \mathcal{E}_i)$ to be the total weight (under $\mathcal{E}_i$) of the edges cut by the partition f. Given this setup, we can formally state the notions of approximation that we consider.

- **α-minimum approximation:** Let c be the maximum, over all partitions $f^* : V \to \{0,1\}$, of the quantity $\min_{i\in[k]} \mathsf{val}(f^*, \mathcal{E}_i)$. The goal is to output an $f : V \to \{0,1\}$ such that $\min_{i\in[k]} \mathsf{val}(f, \mathcal{E}_i) \geqslant \alpha \cdot c$.
- **α-Pareto approximation:** Let $c_1, c_2, \ldots, c_k$ be given such that there exists $f^* : V \to \{0,1\}$ with $\mathsf{val}(f^*, \mathcal{E}_i) \geqslant c_i$ for each $i \in [k]$. The goal is to output an $f : V \to \{0,1\}$ such that $\mathsf{val}(f, \mathcal{E}_i) \geqslant \alpha \cdot c_i$ for all $i \in [k]$.

For $k = 1$, there is a celebrated polynomial time $\alpha_{GW} = 0.8786\ldots$ factor (Pareto) approximation algorithm by Goemans and Williamson [GW95]. This approximation is in both the minimum and Pareto senses. Furthermore, it is Unique-Games hard to achieve a better approximation factor than this [KKMO07], and the entire polynomial time "approximation curve" is also known.

For larger (but constant) k, far less is understood. Clearly, the hardness results from the $k = 1$ case

*Department of Computer Science, Weizmann Institute of Science. Research supported in part by NSF grant CCF-1253886.

†Department of Computer Science, Courant Institute of Mathematical Sciences, New York University. Supported by the NSF Award CCF-1422159, the Simons Collaboration on Algorithms and Geometry and the Simons Investigator Award.

‡Department of Mathematics & Department of Computer Science, Rutgers University. Research supported in part by a Sloan Fellowship, NSF grants CCF-1253886 and CCF-1540634, and BSF grant 2014359.

§Department of Computer Science, University of Toronto. This work was done when the author was a research scientist at Google, Mountain View, CA, USA

¶Department of Computer Science, Courant Institute of Mathematical Sciences, New York University. Supported by same sources as Khot.

carry over, and thus it is Unique-Games hard to approximate this to a factor better than α_{GW}. [ABG06] gave a polynomial time 0.439-Pareto approximation algorithm for this problem for the case $k = 2$. Subsequently, [BKS15] gave a polynomial time $(1/2 - \varepsilon)$-Pareto approximation algorithm for this problem. For the case of unweighted graphs[1], [BKS15] showed that there is a polynomial time $(1/2 + \Omega(1/k^2))$-minimum approximation algorithm. Furthermore, [BKS15] gave a matching integrality gap of $(1/2 + O(1/k^2))$ for a natural SDP relaxation of the minimum approximation problem.

Our main result is a polynomial time 0.8780-factor Pareto approximation algorithm for simultaneous Max-Cut for arbitrary constant k.

Theorem 1.1. *For all constant k and $c > 0$, given weighted graphs $(G_i(V, \mathcal{E}_i))_{i=1}^k$ with $|V| = n$ and where all non-zero edge weights are lower bounded by $exp(-n^c)$, there is a $\mathsf{poly}(n)$ time algorithm which computes a 0.8780-factor Pareto approximation (and hence min approximation) to the simultaneous* Max-Cut *problem with k instances.*

Remark 1. *We assume that the non-zero edge weights are lower bounded by $exp(-|V|^c)$ for some constant $c > 0$. We are interested in an algorithm which runs in time polynomial in $|V|$ and hence it is natural to assume the edge weights are lower bounded by $exp(-|V|^c)$ as otherwise the bit complexity of the input will be super polynomial in $|V|$.*

Remark 2. *Our approximation ratio matches the Goemans-Williamson constant $\alpha_{GW} = 0.8786\ldots$ up to three decimal places. It might be possible to improve the approximation ratio through small modifyications our rounding procedure. However, we believe that getting the exact α_{GW}-approximation (if it exists) might require new techniques. See Remark 3 for more details.*

We give a brief overview of ideas involved in our algorithm next. The main ingredients of the algorithm are: a sum-of-squares hierarchy SDP relaxation, a generalization of the [RT12], [ABG12] approach to rounding such relaxations, and some ideas from [BKS15].

[1] We call an instance of simultaneous Max-Cut *unweighted* if for any i, all the nonzero weight edges under $\mathcal{E}_i$ have the same weight.

1.1 Overview of the algorithm We begin by considering the unweighted case; later we will discuss how to remove this restriction. One crucial observation about the unweighted case is that if there are enough edges in every graph (as a function of k), then a random cut simultaneously cuts a constant fraction of edges from each graph with high probability. Thus, we can always assume that each target value is $c_i = \Omega_k(1)$, which is a constant for a constant k.

There is a natural SDP relaxation for the simultaneous Max-Cut problem, generalizing the Goemans-Williamson SDP for the $k = 1$ case. If we solve this SDP and round the resulting vector solution using the Goemans-Williamson hyperplane rounding procedure, this gives us a distribution of partitions of the vertex set V, such that for *each* $i \in [k]$, the total weight of edges cut in instance i is at least α_{GW} times the corresponding SDP cut value. However, unlike in the $k = 1$ case, this does not guarantee the existence of a single partition of V which is achieves a large cut value for all the k instances simultaneously! This distinction between distributions of solutions which are good in expectation for each instance and single solutions that are simultaneously good for all instances is at the heart of the difficulty in designing simultaneous approximation algorithms.

One of the basic ingredients underlying mathematical programming relaxation hierarchies for combinatorial optimization problems is the idea of expanding the search space, from the discrete space of pure assignments to the continuous space of distributions over assignments. For simultaneous approximation of Max-Cut beyond a factor 1/2, this idea alone is not enough. An example from [BKS15] shows that there are cases of simultaneous Max-Cut on k-instances, for which there is a distribution of partitions of V cutting $(1 - \frac{1}{k})$-fraction of edges *in expectation* for each instance, but for which any single partition of V, there is an instance $i \in [k]$, such that at most 1/2 of the edges in instance i are cut by the partition. This is where the sum-of-squares SDP hierarchy comes in – even though it is also modeled on the idea of expanding the search space to distributions of assignments – it allows us to *condition* on partial assignments and impose a constraint that the SDP cut value is large in expectation for each instance and for every possible conditioning on a small number of variables. This is what allows us to overcome the aforementioned obstacle.

Having formulated the SDP relaxation, we now discuss the rounding procedure. The motivating observation is this: if the rounding procedure is such that for each instance the expected cut value is large, and further the cut value is concentrated around its expectation with high probability, then by a union bound, the rounding procedure will produce a cut that is simultaneously good for all instances. The rounding procedure we will use will be closely related to the Goemans-Williamson rounding (but different – it was found by computer search given various technical conditions required by the rest of the algorithm). Our algorithm now tries to improve the concentration of the cut-value produced by the rounding procedure, via a beautiful information-theoretic approach of Raghavendra-Tan [RT12]. If the cut-value for a certain instance turns out to be not concentrated under the rounding procedure, then it must be because of high correlation between many pairs of edges of that instance (more precisely, correlation between the events that the edge is cut). This in turn means that conditioning on the variables in a random edge should significantly decrease the amount of entropy of the rounded cut. Iterating this several times, and using the fact that the initial entropy is not too large, we conclude that conditioning on a small number of variables leads to good concentration for the rounding procedure. The key point is that the sum-of-squares SDP relaxation we use gives us access to a vector solution for the conditioned SDP, with the promise that the SDP cut-value (and hence the expected integral cut-value) is still large. By the concentration property and a union bound, we get a simultaneously good cut. This completes the description of the algorithm in the unweighted case.

To handle the general weighted case, we essentially need to overcome few technical obstacles. Following [BKS15], we add a preprocessing and postprocessing phase. The preprocessing phase identifies "wild" instances, i.e. those instances with an abnormally large number of high (weighted-)degree vertices (which would increase the variance of the cut value of that instance under random rounding). Then the SDP based algorithm described above is run only on the "tame" instances.

With conditioning on constantly many variables, we can only manage to bring the variance down to arbitrarily small constant. Hence, in order to use second moment method to get concentration, we would need a good lower bound on the expected value of a cut given by our rounding procedure. If the graphs are weighted then it is not necessarily true that the simultaneous cut value is large for all instances. One important property of the tame instances we used is that they have a good simultaneous Max-Cut value. We crucially use this property while formulating the SDP for tame instances.

Finally in the postprocessing phase, we find suitable assignment to the high degree vertices of the wild instances to ensure that those instance have a large cut value (without spoiling the large cut value of the tame instances that the SDP guaranteed) – this uses a new and much simpler perturbation argument compared to [BKS15].

This concludes the high-level description of the algorithm.

1.2 Note about the rounding procedure We mentioned earlier that our SDP solution after conditioning on a small number of variables is rounded by a rounding algorithm similar to the Goemans-Williamson rounding algorithm, but is different. We discuss this rounding procedure here, and compare it to the previous results that used similar rounding procedures.

For convenience, we switch the notation from 0/1 to $+1, -1$, such that any function $f : V \to \{-1, +1\}$ defines a cut in a natural way. Define the bias of a $\{+1, -1\}$ random variable x as $\mathbf{E}[x]$. The SDP solution induces a consistent local distribution on every set of variables of size at most some constant r, and we define the *SDP-bias* of a variable as the bias with respect to this local distribution. For a given rounding procedure, we define the *rounding-bias* of a variable as the bias with respect to the rounding procedure. Note that in the original hyperplane rounding of Goemans-Williamson, the rounding-bias of each vertex is 0.

In the rounding procedure for the Max-Bisection from [RT12], the rounding-bias for each variable induced by the rounding procedure is the same as the SDP-bias. Their algorithm gave a 0.85 approximation for Max-Bisection, and using the same bias function for the rounding along with the analysis of our algorithm, we can get 0.85 approximation for simultaneous Max-Cut as well (See Section 3.3.6 for more details). The approximation factor given by [RT12] was subsequently improved in [ABG12] to 0.8776, where they used new techniques to relax the restriction on the choice of the bias function. Nevertheless, the rounding procedure was still quite constrained by the need to maintain the balance of the cut, as required by the Max-Bisection problem.

In our setting, we do not need equal sized partition of the vertex set, we have more freedom in our rounding procedure with respect to the rounding-bias. It turns out that we only have to ensure that when the bias of a variable is high, the side of the cut it falls on is almost fixed (that this condition suffices heavily depends on features of our algorithm and its analysis). This helps us achieve an improved approximation factor of 0.8780. The rounding function we come up with was arrived at by computer search (along with some trial-and-error).

The approximation ratio for our rounding procedure is proved by a computer assisted prover, using techniques similar to those of [Sjo09] and [ABG12].

1.3 Other related work The simultaneous Max-Cut problem is a special case of the simultaneous approximation problem for general constraint satisfaction problems. This general problem was studied in [BKS15], where it was shown that there is a polynomial time constant factor Pareto approximation algorithm for every simultaneous CSP (with approximation factor independent of k). The algorithm there was based on understanding the structure of CSP instances whose value is highly concentrated under a random assignment to the variables, in addition to linear-programming. It was also observed that there are CSPs for which the best polynomial time approximation factor for the simultaneous version (with $k > 1$) is *different* from the best polynomial time approximation factor achievable in the standard $k = 1$ case (assuming $P \neq NP$). This makes the study of simultaneous approximation factors very interesting.

The simultaneous MAXSAT problem was studied in [GRW11], where a 1/2-Pareto approximation algorithm was given. For bounded width MAXSAT, the approximation factor was improved to $(3/4 - \varepsilon)$ in [BKS15].

It remains an open and very interesting problem to determine for which CSPs the simultaneous approximation problem for $k > 1$ is harder than the classical $k = 1$ case.

2 Preliminaries

2.1 Simultaneous Max-Cut Let V be a vertex set with $|V| = n$. We use the set $[n]$ for the vertex set V for convenience. We are given k graphs $G_1, \ldots, G_k$ on the vertex set V. Let $\mathcal{E}_\ell : [n] \times [n] \to \mathbf{R}^{\geqslant 0}$ denote the edge weights of graph G_ℓ where the edge weights are normalized such that total weight of edges in each instance is 1. As mentioned in Remark 1, we assume that all edge weights are either 0 or lower bounded by 2^{-n^c} for some $c > 0$. We'll use $\mathcal{E}_\ell$ to denote the edge set of graph G_ℓ and also the distribution of the edges based on the weights. For each instance ℓ, we are given a target cut value c_ℓ that we would like to achieve (and we know is possible).

A partition $(U, \overline{U})$ of V is said to be an α-approximation if for each instance G_ℓ,

$$\mathsf{Cut}_\ell(U, \overline{U}) \geqslant \alpha \cdot c_\ell.$$

2.2 Information Theory In this section, we define and state some facts about entropy and mutual information between random variables.

Definition 1. (Entropy) *Let X be a random variable taking values in $[q]$ then, entropy of X is defined as:*

$$H(X) := \sum_{i \in [q]} \Pr[X = i] \log \frac{1}{\Pr[X = i]}.$$

Definition 2. (Conditional Entropy) *Let X, Y be jointly distributed random variable taking values in $[q]$ then, the conditional entropy of X conditioned on Y is defined as:*

$$H(X|Y) = E_{i \in [q]} H(X|Y = i).$$

The following observations can be made about entropy of a collection of random variables.
Entropy of a collection of random variables cannot exceed the sum of their entropies.

Fact 2.1. $H(X_1, X_1, \ldots, X_n) \leqslant \sum_{i=1}^{n} H(X_i)$.

Entropy never decreases on adding more random variables to the collection.

Fact 2.2. $H(X_1, X_2|Y) \geqslant H(X_1|Y)$.

Conditioning can only decrease the entropy.

Fact 2.3. $H(X|Y) - H(X|Y, Z) \geqslant 0$.

Definition 3. (Mutual Information) *Let X, Y be jointly distributed random variable taking values in $[q]$ then, the mutual information between X and Y is defined as:*

$$I(X;Y) := \sum_{i,j \in [q]} \Pr[X = i, Y = j] \log \frac{\Pr[X = i, Y = j]}{\Pr[X = i] \Pr[Y = j]}$$

Theorem 2.1. *(Data Processing Inequality) If X, Y, W, Z are random variables such that X is fully-determined by W and Y is fully-determined by Z, then*

$$I(X, Y) \leqslant I(W, Z).$$

3 Algorithm for simultaneous weighted MAX-CUT

In this section, we give our approximation algorithm for simultaneous weighted MAX-CUT and the analysis.

3.1 Notation We use the same notation as in [BKS15], which we reproduce here. Let $\mathcal{E} = \binom{V}{2}$ be the set of all possible edges. Given an edge e and a vertex v, we say $v \in e$ if v appears in the edge e. For an edge e, let e_1, e_2 denote the endpoints of e (arbitrary order). Let $f : V \to \{0,1\}$ be an assignment. For an edge $e \in \mathcal{E}$, define $e(f)$ to be 1 if the edge e is cut by the assignment f, and define $e(f) = 0$ otherwise. Note that an assignment cuts an edge if it assigns different values to the end points. Then, we have the following expression for the cut value of the assignment:

$$\mathsf{val}(f, \mathcal{E}) \stackrel{\text{def}}{=} \sum_{e \in \mathcal{E}} \mathcal{E}(e) \cdot e(f).$$

A partial assignment $h : S \to \{0,1\}$ is an assignment to S where $S \subseteq V$. We say an edge is *active* with respect to S if at least one of the end vertices is not in S. We denote by $\mathsf{Active}(S)$ the set of all edges which are active with respect to S. For two edges $e, e' \in \mathcal{E}$, we say $e \sim_S e'$ if they share a vertex that is contained in $V \backslash S$. Note that if $e \sim_S e'$, then e, e' are both in $\mathsf{Active}(S)$, and also $e \sim_S e$, $\forall e \in \mathcal{E}$. Let $\mathsf{actdist}_S(\ell)$ denote the distribution over $\mathsf{Active}(S)$, obtained by renormalizing $\mathcal{E}_\ell$ to have total weight 1 over $\mathsf{Active}(S)$.

Define the active degree given S of a variable $v \in V \backslash S$ for instance ℓ by:

$$\mathsf{actdeg}_S(v, \ell) \stackrel{\text{def}}{=} \sum_{e \in \mathsf{Active}(S), e \ni v} \mathcal{E}_\ell(e).$$

We then define the active degree of the whole instance ℓ given S:

$$\mathsf{actdeg}_S(\ell) \stackrel{\text{def}}{=} \sum_{v \in V \backslash S} \mathsf{actdeg}_S(v, \ell).$$

Note that we count weight of an active edge in $\mathsf{actdeg}_S(\ell)$ at most twice. For a partial assignment $h : S \to \{0,1\}$, we define

$$\mathsf{val}(h, \mathcal{E}_\ell) \stackrel{\text{def}}{=} \sum_{\substack{e \in \mathcal{E} \\ e \notin \mathsf{Active}(S)}} \mathcal{E}_\ell(e) \cdot e(h)$$

which is the total weight of non-active edges cut by the partial assignment h. Thus, for an assignment $g : V \backslash S \to \{0,1\}$, to the remaining set of variables, we have the equality:

$$\mathsf{val}(h \cup g, \mathcal{E}_\ell) - \mathsf{val}(h, \mathcal{E}_\ell) = \sum_{e \in \mathsf{Active}(S)} \mathcal{E}_\ell(e) \cdot e(h \cup g).$$

3.2 Algorithm In Figure 1, we give the algorithm for Simultaneous MAX-CUT. The input to the algorithm consists of an integer $k \geqslant 1$, $\varepsilon \in (0, 1/5]$, k instances of MAX-CUT, specified by weight functions $\mathcal{E}_1, \ldots, \mathcal{E}_k$, and k target objective values $c_1, \ldots, c_k$.

3.3 Analysis of the Algorithm The algorithm broadly proceeds in 3 sections, the pre-processing step, the SDP step and the post processing step. The pre-processing step consists of identifying a small subset $S \subseteq V$ carefully. We then attempt all assignments to vertices in S by brute force iteratively and use SDP with the partial assignment followed by a rounding to assign vertices in $V \backslash S$. The post-processing step involves perturbing the assignments to the vertices in S, the need for which is explained in detail in Section 3.3.7.

In what follows, we stick to the following notation. Let $S^\star$ denote the final set S that we get at the end of Step 3. of ALG-SIM-MAXCUT. Let $f^\star : V \to \{0,1\}$ be the assignment that achieves $\mathsf{val}(f^\star, \mathcal{E}_\ell) \geqslant c_\ell$ for all $l \in [k]$ and $h^\star$ be the restriction of $f^\star$ to the set $S^\star$.

3.3.1 Pre-processing: Low and High variance instances

DEFINITION 4. (τ-SMOOTH DISTRIBUTION) *A distribution D on $\{0,1\}$ is called τ-smooth if*

$$\Pr_{x \sim D}[x = 1] \geqslant \tau, \quad \Pr_{x \sim D}[x = 0] \geqslant \tau.$$

Let $h : S \to \{0,1\}$ be an arbitrary partial assignment to the vertices in S. Let $g : V \backslash S \to \{0,1\}$ be the random assignment such that each of the marginals $g(v)$ is τ-smooth. For an instance ℓ, define the random variable

$$Y_\ell \stackrel{\text{def}}{=} \mathsf{val}(h \cup g, \mathcal{E}_\ell) - \mathsf{val}(h, \mathcal{E}_\ell) = \sum_{e \in \mathsf{Active}(S)} \mathcal{E}_\ell(e) \cdot e(h \cup g).$$

Y_ℓ measures the total active edge weight cut by the assignment in the instance ℓ.

Consider the two quantities defined in Step 3. of the algorithm. They depend only on S (and importantly, not on h), which will be useful in controlling the expectation and variance of Y_ℓ. The

Input: k instances of MAX-CUT, with weights defined by $\mathcal{E}_1, \ldots, \mathcal{E}_k$ on the set of variables V, target objective values $c_1, \ldots, c_k$, and $\varepsilon \in (0, 1/5]$.
Output: An assignment to V.
Parameters: $\delta_0 = \frac{1}{10k}$, $\varepsilon_0 = \frac{\varepsilon}{2}$, $t = \frac{2k}{\gamma} \cdot \log\left(\frac{21}{\gamma}\right)$, $\tau = \varepsilon$, $\gamma = \frac{\tau^2 \varepsilon_0^2 \delta_0}{4}$.
Pre-processing:

1. Initialize $S \leftarrow \varnothing$.
2. For each instance $\ell \in [k]$, initialize $\mathsf{count}_\ell \leftarrow 0$ and $\mathsf{flag}_\ell \leftarrow$ TRUE.
3. Repeat the following until for every $\ell \in [k]$, either $\mathsf{flag}_\ell =$ FALSE or $\mathsf{count}_\ell = t$:
   (a) For each $\ell \in [k]$, compute $\mathsf{Uvar}_\ell = \sum_{e \sim_S e'} \mathcal{E}_\ell(e)\mathcal{E}_\ell(e')$.
   (b) For each $\ell \in [k]$ compute $\mathsf{Lmean}_\ell \overset{\text{def}}{=} \tau \sum_{e \in \mathsf{Active}(S)} \mathcal{E}_\ell(e)$.
   (c) For each $\ell \in [k]$, if $\mathsf{Uvar}_\ell \geqslant \delta_0 \varepsilon_0^2 \cdot \mathsf{Lmean}_\ell^2$, then set $\mathsf{flag}_\ell =$ TRUE, else set $\mathsf{flag}_\ell =$ FALSE.
   (d) Choose any $\ell \in [k]$, such that $\mathsf{count}_\ell < t$ AND $\mathsf{flag}_\ell =$ TRUE (if any):
      i. Find $v \in V$ such that $\mathsf{actdeg}_S(v, \ell) \geqslant \gamma \cdot \mathsf{actdeg}_S(\ell)$.
      ii. Set $S \leftarrow S \cup \{v\}$. We say that v was brought into S because of instance ℓ.
      iii. Set $\mathsf{count}_\ell \leftarrow \mathsf{count}_\ell + 1$.
4. After exiting the loop:
   - Let $\mathcal{L}$ denote the set of all $\ell \in [k]$ for which flag_ℓ is set to FALSE (these will be called "low-variance" instances).
   - Let $\mathcal{H}$ denote the set of all $\ell \in [k]$ for which $\mathsf{count}_\ell = t$ (these will be called "high-variance" instances).

Main algorithm:

5. For each possible partial fixing $h : S \to \{0, 1\}$ do the following
   (a) Solve the SDP given in Figure 3 (Refer Section 3.3.3).
   (b) Follow the procedure in Figure 4 to make the solution locally independent. (Refer Section 3.3.4)
   (c) Round the solution based on the rounding procedure described in Figure 5 to get a partial assignment $g : V \backslash S \to \{0, 1\}$. (Refer Section 3.3.5)
   (d) **Post-processing step:** For every assignment $h' : S \to \{0, 1\}$, compute $\min_\ell \frac{\mathsf{val}(h' \cup g, \mathcal{E}_\ell)}{c_\ell}$ and return the assignment $h' \cup g$ that maximizes this.

Figure 1: Algorithm ALG-SIM-MAXCUT for approximating weighted simultaneous MAX-CUT

first quantity is an upper bound on $\mathsf{Var}[Y_\ell]$:

$$\mathsf{Uvar}_\ell \overset{\text{def}}{=} \sum_{e \sim_S e'} \mathcal{E}_\ell(e)\mathcal{E}_\ell(e').$$

The second quantity is a lower bound on $\mathbf{E}[Y_\ell]$:

$$\mathsf{Lmean}_\ell \overset{\text{def}}{=} \tau \cdot \sum_{e \in \mathsf{Active}(S)} \mathcal{E}_\ell(e).$$

LEMMA 3.1. *Let $S \subseteq V$ be a subset of vertices and $h : S \to \{0, 1\}$ be an arbitrary partial assignment to S. Let $Y_\ell, \mathsf{Uvar}_\ell, \mathsf{Lmean}_\ell$ be as above.*

1. *If $\mathsf{Uvar}_\ell \leqslant \delta_0 \varepsilon_0^2 \cdot \mathsf{Lmean}_\ell^2$, then $\Pr[Y_\ell < (1 - \varepsilon_0)\,\mathbf{E}[Y_\ell]] < \delta_0$.*
2. *If $\mathsf{Uvar}_\ell \geqslant \delta_0 \varepsilon_0^2 \cdot \mathsf{Lmean}_\ell^2$, then there exists $v \in V \backslash S$ such that*

$$\mathsf{actdeg}_S(v, \ell) \geqslant \frac{1}{4} \tau^2 \varepsilon_0^2 \delta_0 \cdot \mathsf{actdeg}_S(\ell).$$

We defer the formal proof to the appendix. The first part is a simple application of the Chebyshev inequality. For the second part, we use the assumption that Uvar_ℓ is large, to deduce that there exists an edge e such that the total weight of edges adjacent to the vertex/vertices in e that belong to $V \backslash S$, *i.e.*, $\sum_{e_2 \sim_S e} \mathcal{E}(e_2)$, is large. It then follows that at least one variable $v \in e$ must have large active degree given S.

The above lemma (Lemma 3.1) ensures that Step 3.(d)i in the algorithm always succeeds in finding a variable v. Next, we note that Step 3. always terminates. Indeed, whenever we find an instance $\ell \in [k]$ in Step 3.d such that $\mathsf{count}_\ell < t$ and $\mathsf{flag}_\ell = \textsc{True}$, we increment count_ℓ. This can happen only tk times before the condition $\mathsf{count}_\ell < t$ fails for all $\ell \in [k]$. Thus the loop must terminate within tk iterations.

To analyze the approximation guarantee of the algorithm, we classify instances according to how many vertices were brought into $S^\star$ because of them.

Definition 5. (Low and High variance instances) *At the completion of Step 3.d in Algorithm* Alg-Sim-MaxCUT, *if $\ell \in [k]$ satisfies $\mathsf{count}_\ell = t$, we call instance ℓ a* high variance *instance. Otherwise we call instance ℓ a* low variance *instance.*

The next two sections describes the SDPs that we formulate and solve for just the low variance instances. The claim that step 5d of the algorithm shown in Figure 1 handles the high variance instances is discussed and proved in Section 3.3.7.

3.3.2 Warmup: Basic SDP formulation for simultaneous Max-Cut.

Our algorithm involves formulating a *Lasserre Hierarchy* SDP relaxation of the residual Max-Cut problem after giving a partial assignment $h : S^\star \to \{0,1\}$. In this section, as a warmup to its analysis, we present and study the *basic* version of that SDP.

We write the $\mathsf{SDP}^\star$ for simultaneous Max-Cut problem, after the partial fixing given by pre-processing step, as in Figure 2. Let $\mathcal{L}$ denote the set of indices of the low variance instances. We have vectors $\mathbf{v}_{\mathbf{T},\alpha}$ for all T and α where T is a subset of V of size at most 2, and α is an assignment to the vertices in T.

If we consider the $\mathsf{SDP}^\star$ without the constraint (3.2), it is easy to see that this is a relaxation. Given a partition $(U, \bar{U})$ of V that achieves a simultaneous optimum, we can set vectors $\mathbf{v}_{\mathbf{T},\alpha} = \mathbf{v}_\varnothing$ if the pair (T,α) is consistent with 1_U (i.e. 1_U assigns α to T) and $\mathbf{v}_{\mathbf{T},\alpha} = 0$ otherwise. $\mathbf{v}_\varnothing$ can be viewed as a vector that denotes 1.

A part of our analysis require that for every low variance instance, the expected weighted fraction of active edges that we cut is at least a constant fraction of its active degree. An optimal SDP solution without constraint (3.2) may not guarantee this condition (for the rounding procedure we choose). Hence, we force the SDP solution to satisfy this property by adding constraint (3.2). We need to relax constraint (3.1) to make sure that there is a solution that satisfies all the constraints.

We now prove that $\mathsf{SDP}^\star$, in its present form, has feasible solutions.

Lemma 3.2. *$\mathsf{SDP}^\star(h^\star)$ shown in Figure 2 has a feasible solution.*

Proof. To show that $\mathsf{SDP}^\star$ has a feasible solution, it suffices to show that there exists an integral solution that satisfies the constraints.

Fix an optimal assignment $f^\star : V \to \{0,1\}$ to the simultaneous instance. $f^\star$ satisfies $\forall \ell \in [k]$, $\mathsf{val}(f^\star, \mathcal{E}_\ell) \geqslant c_\ell$. Consider the following random assignment: For all $v \in V \backslash S^\star$

$$r(v) = \begin{cases} f^\star(v) & \text{with probability } (1-\varepsilon) \\ \overline{f^\star(v)} & \text{otherwise} \end{cases}$$

where $\overline{f^\star(v)}$ is $f^\star(v)$ flipped. For $v \in S^\star$, set $r(v) = f^\star(v)$. Now, for any $\ell \in \mathcal{L}$, let Y_ℓ denote the random variable

$$Y_\ell = \sum_{e \in \mathsf{Active}(S^\star)} \mathcal{E}_\ell(e) \cdot e(r).$$

We have $\mathbf{E}[e(r)] \geqslant \varepsilon$, hence $\mathbf{E}[Y_\ell] \geqslant \varepsilon/2 \cdot \mathsf{actdeg}_{S^\star}(\ell)$. Also,

$$\begin{aligned}
&\mathop{\mathbf{E}}_r[\mathsf{val}(r, \mathcal{E}_\ell)] \\
&\geqslant \sum_{e \notin \mathsf{Active}(S^\star)} \mathcal{E}_\ell(e) \cdot \mathbf{E}[e(r)] \\
&\quad + \sum_{\substack{e \in \mathsf{Active}(S^\star), \\ e(f^\star)=1}} \mathcal{E}_\ell(e) \cdot \mathbf{E}[e(r)] \\
&= \sum_{e \notin \mathsf{Active}(S^\star)} \mathcal{E}_\ell(e) \cdot e(f^\star) \\
&\quad + \sum_{\substack{e \in \mathsf{Active}(S^\star), \\ e(f^\star)=1}} \mathcal{E}_\ell(e) \cdot \min((1-\varepsilon)^2 + \varepsilon^2, 1-\varepsilon) \\
&\geqslant (1-2\varepsilon) \sum_{e : e(f^\star)=1} \mathcal{E}_\ell(e) \\
&= (1-2\varepsilon)\mathsf{val}(f^\star, \mathcal{E}_\ell) \\
&\geqslant (1-2\varepsilon) c_\ell.
\end{aligned}$$

Thus, we have,

$$\sum_{e=\{i,j\}\in\mathcal{E}_\ell} \mathcal{E}_\ell(e)(\|\mathbf{v}_{\{(\mathbf{i},\mathbf{j}),(\mathbf{0},\mathbf{1})\}}\|_2^2 + \|\mathbf{v}_{\{(\mathbf{i},\mathbf{j}),(\mathbf{1},\mathbf{0})\}}\|_2^2) \geqslant (1-3\varepsilon)c_\ell \qquad \forall \ell \in [k], \tag{3.1}$$

$$\langle \mathbf{v}_{\{\mathbf{i},\mathbf{0}\}}, \mathbf{v}_{\{\mathbf{i},\mathbf{1}\}} \rangle = 0 \qquad \forall i \in [n],$$

$$\|\mathbf{v}_{\{(\mathbf{i},\mathbf{j}),(\mathbf{b_1},\mathbf{b_2})\}}\|^2 = \langle \mathbf{v}_{\{\mathbf{i},\mathbf{b_1}\}}, \mathbf{v}_{\{\mathbf{j},\mathbf{b_2}\}} \rangle \qquad \forall i,j \in [n] \text{ and } b_1, b_2 \in \{0,1\}$$

$$\|\mathbf{v}_{\{\mathbf{T},\alpha\}}\|^2 = \langle \mathbf{v}_{\{\mathbf{T},\alpha\}}, \mathbf{v}_\varnothing \rangle \qquad \forall T \subset V, |T| \leqslant 2, \alpha \in \{0,1\}^{|T|}$$

$$\mathbf{v}_{\{\mathbf{i},\mathbf{b}\}} = \mathbf{v}_\varnothing \qquad \forall i \in S^\star, b = h(i)$$

$$\|\mathbf{v}_\varnothing\|^2 = 1$$

$$\sum_{e=\{i,j\}\in\mathsf{Active}(S^\star)} \mathcal{E}_\ell(e)(\|\mathbf{v}_{\{(\mathbf{i},\mathbf{j}),(\mathbf{0},\mathbf{1})\}}\|_2^2 + \|\mathbf{v}_{\{(\mathbf{i},\mathbf{j}),(\mathbf{1},\mathbf{0})\}}\|_2^2) \geqslant \varepsilon/3 . \mathsf{actdeg}_{S^\star}(\ell) \qquad \forall \ell \in \mathcal{L} \tag{3.2}$$

Figure 2: $\mathsf{SDP}^\star(h : S^\star \to \{0,1\})$ for simultaneous MAX-CUT with partial fixing

1. $\mathbf{E}[Y_\ell] \geqslant \varepsilon/2 \cdot \mathsf{actdeg}_{S^\star}(\ell)$.
2. $\mathbf{E}_r[\mathsf{val}(r, \mathcal{E}_\ell)] \geqslant (1-2\varepsilon)c_\ell$.

Recall that the $\mathsf{SDP}^\star$ involves only the low variance instances. Also, the assignment r is ε-smooth on the set $V \backslash S^\star$. Therefore, we have concentration guarantees as given by point 1 of Lemma 3.1.

$$\Pr[Y_\ell \leqslant (1-\varepsilon_0)\,\mathbf{E}[Y_\ell]] \leqslant \delta_0$$
$$\Pr[\mathsf{val}(r, \mathcal{E}_\ell) \leqslant (1-\varepsilon_0)\,\mathbf{E}[\mathsf{val}(r,\mathcal{E}_\ell)]] \leqslant \delta_0.$$

Hence, with probability at least $1-2\delta_0$, we have $Y_\ell \geqslant (1-\varepsilon/2)\cdot \varepsilon/2 \cdot \mathsf{actdeg}_{S^\star}(\ell) \geqslant \varepsilon/3 \cdot \mathsf{actdeg}_{S^\star}(\ell)$ and $\mathsf{val}(r,\mathcal{E}_\ell) \geqslant (1-\varepsilon/2)(1-2\varepsilon)c_\ell \geqslant (1-3\varepsilon)c_\ell$.

Now we do union bound over all low variance instances, we get with a probability at least $1 - 2\cdot\delta_0\cdot k = 4/5$, all the SDP constraints are satisfied by integral solution r. Thus, there exists an *integral* solution which satisfies all $\mathsf{SDP}^\star(h^\star)$ constraints and hence is feasible.

3.3.3 Lasserre Hierarchy SDP formulation.

We now describe the r^{th}-level Lasserre SDP for the SDP in Figure 2.

The SDP formulation has vectors $\mathbf{v}_{\{\mathbf{T},\alpha\}}$ for all $T \subseteq V$ such that $|T| \leqslant r$ and $\alpha \in \{0,1\}^{|T|}$. In terms of local distribution, the SDP solution consists of *consistent local distribution* on every set T of size at most r (denoted by μ_T). The random variable corresponding to set T is denoted by X_T distributed over $\{0,1\}^{|T|}$. The vector solution and the local distribution are related as follows: Suppose T and U are subsets of V such that $|T \cup U| \leqslant r$ and the assignments $\alpha \in \{0,1\}^{|T|}$ and $\beta \in \{0,1\}^{|U|}$ are consistent on $T \cap U$ then

$$\langle \mathbf{v}_{\mathbf{T},\alpha}, \mathbf{v}_{\mathbf{U},\beta} \rangle = \Pr_{\mu_{T\cup U}}(X_T = \alpha, X_U = \beta).$$

To ensure the consistency among local distributions, we have to add the constraints 3.5 and 3.6 to the SDP in Figure 3. Here if $\alpha \in \{0,1\}^{|S|}$ is an assignment to the vertices in S, and if $S' \subset S$, $\alpha_{|S'} \in \{0,1\}^{|S'|}$ denotes the assignment α restricted to the vertices in S'. Also, if α and β are assignments to sets S and T agreeing on $S \cap T$, then we denote $\alpha \circ \beta$ an assignment to $S \cup T$. We also add the set of constraints (Equation 3.7 in Figure 3) to capture the partial assignment $h : S^\star \to \{0,1\}$ given by pre-processing.

With these definitions and constraints, the objective is to ensure that for all $\ell \in [k]$,

$$\sum_{e=\{i,j\}\in\mathcal{E}_\ell} \mathcal{E}_\ell(e)\Pr[X_{\{i,j\}} = (0,1) \vee X_{\{i,j\}} = (1,0)] \geqslant (1-3\varepsilon)c_\ell$$

A simple way to capture this would be to write the objective of the SDP solution similar to the basic SDP formulation, as follows.

$$\sum_{e=\{i,j\}\in\mathcal{E}_\ell} \mathcal{E}_\ell(e)(\|\mathbf{v}_{\{(\mathbf{i},\mathbf{j}),(\mathbf{0},\mathbf{1})\}}\|_2^2 + \|\mathbf{v}_{\{(\mathbf{i},\mathbf{j}),(\mathbf{1},\mathbf{0})\}}\|_2^2) \geqslant (1-3\varepsilon)c_\ell$$

$$\begin{aligned}
&\sum_{e=\{i,j\}\in\mathcal{E}_\ell} \big(\, \mathcal{E}_\ell(e)(\|\mathbf{v}_{\{\mathbf{S}\cup\{\mathbf{i},\mathbf{j}\},\alpha\circ(\mathbf{0},\mathbf{1})\}}\|_2^2 + \|\mathbf{v}_{\{\mathbf{S}\cup\{\mathbf{i},\mathbf{j}\},\alpha\circ(\mathbf{1},\mathbf{0})\}}\|_2^2)\,\big) \\
&\qquad \geqslant (1-3\varepsilon)c_\ell\|\mathbf{v}_{\{\mathbf{S},\alpha\}}\|^2 \qquad \forall S\subseteq V, |S|\leqslant r-2, \alpha\in\{0,1\}^{|S|}, \forall \ell\in[k]
\end{aligned} \tag{3.3}$$

$$\begin{aligned}
&\sum_{e=\{i,j\}\in\mathsf{Active}(S^\star)} \big(\, \mathcal{E}_\ell(e)(\|\mathbf{v}_{\{\mathbf{S}\cup\{\mathbf{i},\mathbf{j}\},\alpha\circ(\mathbf{0},\mathbf{1})\}}\|_2^2 + \|\mathbf{v}_{\{\mathbf{S}\cup\{\mathbf{i},\mathbf{j}\},\alpha\circ(\mathbf{1},\mathbf{0})\}}\|_2^2)\,\big) \\
&\qquad \geqslant {}^{\varepsilon}\!/_{3}.\mathsf{actdeg}_{S^\star}(\ell)\|\mathbf{v}_{\{\mathbf{S},\alpha\}}\|^2 \qquad \forall S\subseteq V, |S|\leqslant r-2, \alpha\in\{0,1\}^{|S|}, \forall \ell\in\mathcal{L}
\end{aligned} \tag{3.4}$$

$$\langle \mathbf{v}_{\{\mathbf{S},\alpha\}}, \mathbf{v}_{\{\mathbf{T},\beta\}}\rangle = \|\mathbf{v}_{\{\mathbf{S}\cup\mathbf{T},\alpha\circ\beta\}}\|_2^2 \qquad \forall S,T\subseteq V, |S\cup T|\leqslant r, \alpha\in\{0,1\}^{|S|}, \beta\in\{0,1\}^{|T|}, \tag{3.5}$$

$$\langle \mathbf{v}_{\mathbf{S},\alpha}, \mathbf{v}_{\mathbf{T},\beta}\rangle = 0 \qquad \forall S,T\subseteq V, |S\cup T|\leqslant r, \alpha\in\{0,1\}^{|S|}, \beta\in\{0,1\}^{|T|}, \text{ s.t. } \alpha_{|S\cap T}\neq\beta_{|S\cap T} \tag{3.6}$$

$$\begin{aligned}
\|\mathbf{v}_{\{\mathbf{T},\alpha\}}\|^2 &= \langle \mathbf{v}_{\{\mathbf{T},\alpha\}}, \mathbf{v}_\varnothing\rangle && \forall T\subseteq V, |T|\leqslant r, \alpha\in\{0,1\}^{|T|} \\
\langle \mathbf{v}_{\{\mathbf{S},\alpha\}}, \mathbf{v}_{\{\mathbf{i},\mathbf{b}\}}\rangle &= \langle \mathbf{v}_{\{\mathbf{S},\alpha\}}, \mathbf{v}_\varnothing\rangle && \forall S\subseteq V, |S|\leqslant r-1, \alpha\in\{0,1\}^{|S|}, \forall i\in S^\star, b=h(i) \\
\|\mathbf{v}_\varnothing\|^2 &= 1
\end{aligned} \tag{3.7}$$

Figure 3: r-round Lasserre lift of $\mathsf{SDP}^\star(h : S^\star \to \{0,1\})$ for simultaneous MAX-CUT with partial fixing

LEMMA 3.3. *r-round Lasserre SDP shown in Figure 3 has a feasible solution.*

Proof. Note that the feasible solution provided for the basic SDP in Lemma 3.2 is integral. Therefore, we can directly conclude that the Lasserre lift of the SDP is feasible, as the same solution can be extended to the Lasserre SDP.

Assign $\mathbf{v}_{S,\alpha}$ to $\mathbf{v}_\varnothing$ if in the integral solution, the vertices in the set S were assigned to α in that order, otherwise assign $\mathbf{v}_{S,\alpha}$ to 0.

In order to make the solution locally independent, we will need to condition based on the local distribution (Refer Section 3.3.4). Therefore, we need to re-write the objective so that it is satisfied (w.r.t the conditioned local distribution) even after conditioning on at most r variables, as shown in Equation 3.3 in the SDP formulation.

Also, similar to the previous case, we need to ensure that the solution post-conditioning still cuts at least a constant fraction of the active edges, which is ensured by adding the set of constraints specified in Equation 3.4 in the SDP.

We observe that solving the SDP using ellipsoid method can result in a small additive error, and if $\mathsf{actdeg}_{S^\star}(\ell)$ is small compared to this additive error, the error would be significant. This will not cause any issues and we elaborate on this more. We can solve the SDP using ellipsoid method with an error of ε in time polynomial in n and $\log(1/\varepsilon)$. Therefore, we can take ε to be $\exp(-\mathsf{poly}(n))$ and still solve the SDP in time polynomial in n. We assumed that the non-zero edge weights are at least $\exp(-n^c)$ for some constant $c > 0$. Therefore, if the active degree is non-zero, it is at least $\exp(-n^c)$. If we take $\varepsilon = \exp(-n^{c'})$ for $c' >> c$, we can solve the SDP in time polynomial in n and get a vector solution which satisfies all the constraints upto additive error ε which is upto multiplicative factor of $(1 + o(1))$. This will not have a major effect on our analysis and hence we assume from here onward that the vector solution that we get satisfies the all the constraints exactly.

3.3.4 Obtaining independent local solution

The notion of independent solution (which is formalized below in Definition 6) that we need is different from [RT12]. Following procedure in Figure 4 is used to achieve the kind of independence we need.

DEFINITION 6. *A Lasserre solution is δ-independent if it satisfies the following condition.*

$$\forall \ell \in \mathcal{L}, \underset{a,b\sim\mathsf{actdist}_{S^\star}(\ell)}{\mathbf{E}}\left[\sum_{i,j\in\{1,2\}} I(X_{a_i};X_{b_j})\right] \leqslant \delta.$$

LEMMA 3.4. *For all $\delta > 0$, there exists $t \leqslant 2k/\delta$ and edges $e^1, e^2, \ldots, e^t \in \mathcal{E}$ such that*

$$\forall \ell \in \mathcal{L}, \underset{a,b\sim\mathsf{actdist}_{S^\star}(\ell)}{\mathbf{E}}[I(X_{a_1},X_{a_2};X_{b_1},X_{b_2}| X_{e^1_1},X_{e^1_2},\ldots,X_{e^t_1},X_{e^t_2})] \leqslant \delta \tag{3.8}$$

Proof. Consider the following potential function,

$$\phi = \sum_{\ell\in\mathcal{L}} \underset{a\in\mathsf{actdist}_{S^\star}(\ell)}{\mathbf{E}} H(X_{a_1},X_{a_2}).$$

As entropy of a bit is at most 1, clearly $\phi \leqslant 2k$. We have the following identity for each $\ell \in \mathcal{L}$ which follows from conditional entropy and linearity of expectation

$$\begin{aligned}&\underset{a,b\in\mathsf{actdist}_{S^\star}(\ell)}{\mathbf{E}}[H(X_{a_1},X_{a_2}|X_{b_1},X_{b_2})]\\ &= \underset{a\in\mathsf{actdist}_{S^\star}(\ell)}{\mathbf{E}}[H(X_{a_1},X_{a_2})] - \underset{a,b\in\mathsf{actdist}_{S^\star}(\ell)}{\mathbf{E}} I(X_{a_1},X_{a_2};X_{b_1},X_{b_2})\end{aligned}$$

This identity suggests that if for some $\ell \in \mathcal{L}$, $\mathbf{E}_{a,b\in\mathsf{actdist}_{S^\star}(\ell)} I(X_{a_1},X_{a_2};X_{b_1},X_{b_2}) > \delta$ then there exists a conditioning which reduces the potential function by at least δ. Thus, either the current conditioned solution satisfies (3.8) in which case we are done or there exists an edge b such that if we condition the SDP solution based on the value of its endpoints (b_1, b_2) according to the local distribution then the potential function decreases by at least δ. So, if we fail to achieve (3.8) then ϕ decreases by at least δ. As entropy is always non-negative and conditioning never increases entropy (Fact 2.3), this process cannot go beyond $2k/\delta$ conditioning. Thus, before at most $2k/\delta$ conditioning, we are guaranteed to achieve (3.8).

The following fact follows from the data processing inequality (Theorem 2.1).

FACT 3.1. *If X_1, X_2, Y_1 and Y_2 are random variables then for $i, j \in \{1,2\}$, we have*

$$I(X_i;Y_j) \leqslant I(X_1,X_2;Y_1,Y_2).$$

The following corollary follows from Lemma 3.4 and Fact 3.1.

COROLLARY 3.1. *For all $\delta > 0$, there exists $t \leqslant \frac{2k}{\delta}$, and edges $e^1, e^2, \ldots, e^t \in \mathcal{E}$, such that*

$$\forall \ell \in \mathcal{L}, \underset{a,b\sim\mathsf{actdist}_{S^\star}(\ell)}{\mathbf{E}}\left[\sum_{i,j\in\{1,2\}} I(X_{a_i};X_{b_j}| X_{e^1_1},X_{e^1_2},\ldots,X_{e^{t-1}_1},X_{e^{t-1}_2})\right] \leqslant 4\delta$$

LEMMA 3.5. *There exists a fixing of at most $\frac{32k}{\delta}$* variables *such that the conditioned solution is $\delta/2$ independent as well as satisfies all constraints from $\mathsf{SDP}^\star(h^\star)$. In particular, the algorithm in Figure 4 returns such a $\delta/2$ independent solution. Also, the running time is bounded by $n^{O(r)}$.*

Proof. $\delta/2$ independence follows from Corollary 3.1 for $t = \frac{16t}{\delta}$ and Fact 3.1. Also, we can verify if a given SDP solution is $\delta/2$-independent or not in time polynomial in n. We now prove the later part.

As the conditioning maintains the marginal distribution of variables and because of the the Inequality (3.3) and (3.4), the constraints about the SDP cut value as well as the fraction of active edges that are cut remain valid in the conditioned solution. Hence, from Lemma 3.2 $\mathsf{SDP}^\star(h^\star)$ remains feasible.

3.3.5 Rounding Procedure

In this section, we describe the rounding procedure for variables in $V \backslash S^\star$. The input to this procedure is 2 round Lasserre solution which is δ-independent. We use a slight variation of GW rounding procedure to round the SDP vector solution. In particular, we want to maintain the bias of heavily biased random variable in our rounding procedure.

SDP gives the vector solution $\mathbf{v_{i,0}}, \mathbf{v_{i,1}}$ for all $i \in [n]$. Let $\mu_i = 2\,\mathbf{E}[X_i] - 1$, the expectation is according to the local distribution. Define $\mathbf{v_i} = \mathbf{v_{i,1}} - \mathbf{v_{i,0}}$. These $\mathbf{v_i}$ are the unit vectors (as $\|\mathbf{v_i}\|^2 = \|\mathbf{v_{i,1}} - \mathbf{v_{i,0}}\|^2 = \|\mathbf{v_{i,1}}\|^2 + \|\mathbf{v_{i,0}}\|^2 - 2\langle v_{i0}, v_{i1}\rangle = \Pr[X_i = 0] + \Pr[X_i = 1] - 0 = 1$). Let $\mathbf{w_i}$ be component of $\mathbf{v_i}$ orthogonal to $\mathbf{v}_\varnothing$ ($\mathbf{v_i} = \mu_i \mathbf{v}_\varnothing + \mathbf{w_i}$), $\|\mathbf{w_i}\|_2 = \sqrt{1-\mu_i^2}$. Let $\overline{\mathbf{w}}_i$ be the normalized unit vector of $\mathbf{w_i}$. The rounding procedure is applied on vectors $\overline{\mathbf{w}}_i$ along with the "bias" of each variable $\langle \mathbf{v_i}, \mathbf{v}_\varnothing\rangle$. The rounding procedure is shown in Figure 5.

Input: $r+2$ round Lasserre solution of a given simultaneous MAX-CUT instance, $\delta \geqslant \frac{32k}{r}$
Output: $\frac{\delta}{2}$-*independent* 2-round Lasserre solution.

1. For all $\ell_1, \ldots, \ell_{r/2} \in \mathcal{L}$, and for all edges $e^i \in \mathsf{actdist}_{S^\star}(\ell_i)$ for all $i \in [r/2]$.
   - Let $S = \cup_{i \in [r/2]} \{e_1^i, e_2^i\}$ be the endpoints of all the edges from (1).
   - For every $\alpha \in \{0,1\}^{|S|}$ such that $\Pr[X_S = \alpha] > 0$ in the local disctibution:
     - Condition the SDP solution on the event $X_S = \alpha$.
     - Output if conditioned solution if it is $\frac{\delta}{2}$-independent.

Figure 4: Making locally independent solution

Input: δ-independent 2 round Lasserre solution, biases $\mu_i \in [-1,+1]$ and a function $f_R : [-1,1] \to [-1,1]$ which is bounded by above and below with some constant degree polynomials
Output: A partition of V.

1. Pick a random Gaussian vector $\mathbf{g}$ orthogonal to $\mathbf{v}_\varnothing$ with each co-ordinate distributed as $\mathcal{N}(0,1)$.
2. For each $i \in [n]$
   - Calculate $\xi_i = \langle \mathbf{g}, \overline{\mathbf{w}}_i \rangle$.
   - Let $r_i \leftarrow f_R(\mu_i)$
   - Set $y_i = 1$ if $\xi_i \leqslant \Phi^{-1}(r_i/2 + 1/2)$, otherwise set $y_i = -1$. (Here, Φ is the Gaussian CDF)

Figure 5: Rounding procedure

3.3.6 Analysis of the rounding procedure We use the notation $\mathsf{poly}_{<1}(x)$ to denote a "polynomial" in x with exponents as real numbers in $(0,1)$, such that $\mathsf{poly}_{<1}(x) \to 0$ as $x \to 0$.

Note that if we simply use the rounding function $f_R(x) = x$ as used in [RT12] the we get for each instance, in expectation the cut produced by the rounding procedure is at least 0.85 times the SDP value (and hence eventually 0.85 approximation for simultaneous MAX-CUT). Here, we leverage the fact that the constraints on what rounding functions are good for us are mild compared to [RT12] as explained in Section 1.2.

LEMMA 3.6. *For a fixed low variance instance, the rounding procedure described in Figure 5 gives an approximation ratio* $0.878001(1-3\varepsilon)$ *in expectation for the following* f_R*,*

$$f_R(x) = 0.79 \cdot x + 0.07 \cdot x^3 + 0.14 \cdot x^7$$

Proof. The proof of this lemma is numerical. We arrive at a informal approximate value for the bound using Matlab code (0.878001) and verify it using computer assisted techniques. The multiplicative loss of $(1-3\varepsilon)$ is because of using $\mathsf{SDP}^\star$. We elaborate on the exact constant 0.878001 that we get next. The probability p_{ij} that a given edge (i,j) is cut by the rounding procedure is a function of μ_i and μ_j, whereas its SDP contribution is a quantity $q_{ij} := 1 - \langle \mathbf{v}_i, \mathbf{v}_j \rangle / 2$. Thus to show a lower bound on approximation ratio it is sufficient to prove the same lower bound on $p_{i,j}/q_{ij}$ for all possible valid configurations of vectors. The program works in a recursive fashion, by continuously splitting the cube (all possible valid configuration) into sub-cubes. In each sub-cube, the program checks if either across all points in the region, the lower bound on α exceeds the approximation ratio we try to prove or if the upper bound on α is lower than the approximation ratio we try to prove. It proceeds with further division into smaller sub-cubes until one of the above is satisfied. If the latter is true at any point, the code returns a failure, and it returns a success if the entire region can be proved to come under the former case. The prover was adapted from [ABG12] and modified to suit our rounding procedure. For more details on the workings of the prover, refer [ABG12].

REMARK 3. *It seems possible to improve the constant 0.878001 by using a different f_R which is continuous and satisfies $f_R(1) = 1$ and $f_R(-1) = -1$ However we suspect that a serious new idea would be needed to get a α_{GW}-approximation algorithm.*

We need the following lemma from [RT12].

LEMMA 3.7. ([RT12]) *Let $\mathbf{v_i}$ and $\mathbf{v_j}$ be the unit vectors, $\mathbf{w_i}$ and $\mathbf{w_j}$ be the components of $\mathbf{v_i}$ and $\mathbf{v_j}$ that are orthogonal to $\mathbf{v}_\varnothing$. Then $|\langle \mathbf{w_i}, \mathbf{w_j} \rangle| \leqslant 2I(X_i; X_j)$.*

Above lemma along with Lemma 3.5 implies that if we sample edge $(i_1, i_2), (j_1, j_2) \sim \mathsf{actdist}_{S^\star}(\ell)$ then we have on average,

$$|\langle \mathbf{w_{i_1}}, \mathbf{w_{j_1}} \rangle| + |\langle \mathbf{w_{i_1}}, \mathbf{w_{j_2}} \rangle| + |\langle \mathbf{w_{i_2}}, \mathbf{w_{j_1}} \rangle| + |\langle \mathbf{w_{i_2}}, \mathbf{w_{j_2}} \rangle| \leqslant \delta.$$

The rounding procedure is assigning values ± 1 to variables y_i where y_i is the variable for vertex $i \in V$ and its value decides on which side of cut the vertex i is present in the final solution. Thus y_i is a random variable taking values in $\{+1, -1\}$. We now wish to prove similar guarantee as the following lemma from [RT12], which relates the mutual information between the pair of rounded variables with the inner product of the corresponding vectors w.

LEMMA 3.8. ([RT12]) *For f_R such that $f_R(x) = x$, if $|\langle \mathbf{w_i}, \mathbf{w_j} \rangle| \leqslant \delta$ then $I(y_i; y_j) \leqslant \delta^{1/3}$.*

In our case, we need that the mutual information between the events that a pair of edges are cut is small on average. Thus, our notion of local independence will be useful in proving this guarantee about mutual information.

LEMMA 3.9. *Fix f_R to be the rounding function given by Lemma 3.6. For a pair of edges (i_1, i_2) and (j_1, j_2), suppose the vectors w corresponding to their endpoints satisfy the following condition,*

$$\begin{aligned} |\langle \mathbf{w_{i_1}}, \mathbf{w_{j_1}} \rangle| + |\langle \mathbf{w_{i_1}}, \mathbf{w_{j_2}} \rangle| + \\ |\langle \mathbf{w_{i_2}}, \mathbf{w_{j_1}} \rangle| + |\langle \mathbf{w_{i_2}}, \mathbf{w_{j_2}} \rangle| \leqslant \delta \end{aligned}$$

then $I(y_{i_1} y_{i_2}; y_{j_1} y_{j_2}) \leqslant \mathsf{poly}_{<1}(\delta)$.

Proof. Since $\overline{\mathbf{w}}_i$ is a normalized vector of $\mathbf{w_i}$ and $\|\mathbf{w_i}\| = \sqrt{1 - \mu_i^2}$, we have

$$\left.\begin{aligned} \sqrt{1-\mu_{i_1}^2} \cdot \sqrt{1-\mu_{j_1}^2} \cdot |\langle \overline{\mathbf{w}}_{i_1}, \overline{\mathbf{w}}_{j_1} \rangle| \\ +\sqrt{1-\mu_{i_1}^2} \cdot \sqrt{1-\mu_{j_2}^2} \cdot |\langle \overline{\mathbf{w}}_{i_1}, \overline{\mathbf{w}}_{j_2} \rangle| \\ +\sqrt{1-\mu_{i_2}^2} \cdot \sqrt{1-\mu_{j_1}^2} \cdot |\langle \overline{\mathbf{w}}_{i_2}, \overline{\mathbf{w}}_{j_1} \rangle| \\ +\sqrt{1-\mu_{i_2}^2} \cdot \sqrt{1-\mu_{j_2}^2} \cdot |\langle \overline{\mathbf{w}}_{i_2}, \overline{\mathbf{w}}_{j_2} \rangle| \end{aligned}\right\} \leqslant \delta. \quad (3.9)$$

Since the total sum is bounded and each quantity is non-negative, at least one of the three quantities in each summand is at most $\delta^{1/3}$. We use two crucial properties of the rounding procedure:

- For the heavily biased variable according to the local distribution, the rounding procedure also keeps the rounded value heavily biased and
- If two vectors $\mathbf{w_i}$ and $\mathbf{w_j}$ are nearly orthogonal, the corresponding rounded values y_i and y_j are nearly independent.

We need following claim which we prove in Section A.

CLAIM 1. *If all these quantities $|\langle \overline{\mathbf{w}}_{i_1}, \overline{\mathbf{w}}_{j_1} \rangle|, |\langle \overline{\mathbf{w}}_{i_1}, \overline{\mathbf{w}}_{j_2} \rangle|, |\langle \overline{\mathbf{w}}_{i_2}, \overline{\mathbf{w}}_{j_1} \rangle|, |\langle \overline{\mathbf{w}}_{i_2}, \overline{\mathbf{w}}_{j_2} \rangle|$ are upper bounded by $\delta^{1/3}$, then we can upper bound $I(y_{i_1}, y_{i_2}); (y_{j_1}, y_{j_2})) \leqslant \mathsf{poly}_{<1}(\delta)$.*

We now formally prove the upper bound on $I(y_{i_1} y_{i_2}; y_{j_1} y_{j_2})$ by case analysis. We use the following upper bound which follows from data processing inequality.

$$I(y_{i_1} y_{i_2}; y_{j_1} y_{j_2}) \leqslant I(y_{i_1}, y_{i_2}); (y_{j_1}, y_{j_2})).$$

We now bound the right hand side based on following case analysis.

- Case 1: If all these quantities $|\langle \overline{\mathbf{w}}_{i_1}, \overline{\mathbf{w}}_{j_1} \rangle|, |\langle \overline{\mathbf{w}}_{i_1}, \overline{\mathbf{w}}_{j_2} \rangle|, |\langle \overline{\mathbf{w}}_{i_2}, \overline{\mathbf{w}}_{j_1} \rangle|, |\langle \overline{\mathbf{w}}_{i_2}, \overline{\mathbf{w}}_{j_2} \rangle|$ are upper bounded by $\delta^{1/3}$ then using Claim 1, we can upper bound $I(y_{i_1}, y_{i_2}); (y_{j_1}, y_{j_2})) \leqslant \mathsf{poly}_{<1}(\delta)$
- Case 2: Consider the case when both the endpoints of an edge (w.l.o.g. of (i_1, i_2)) have large bias i.e. $\sqrt{1 - \mu_{i_1}^2} \leqslant \delta^{1/3}, \sqrt{1 - \mu_{i_2}^2} \leqslant \delta^{1/3}$. It implies,

$$\min(|1 - \mu_{i_1}|, |1 + \mu_{i_1}|) \leqslant \delta^{1/3}$$
$$\min(|1 - \mu_{i_2}|, |1 + \mu_{i_2}|) \leqslant \delta^{1/3}$$

Assume both $\mu_{i_1}, \mu_{i_2} > 0$ (there cases can be handled in a similar way). Then we have, $1 - \mu_{i_1} \leqslant \delta^{1/3}$ and $1-\mu_{i_2} \leqslant \delta^{1/3}$. Since the rounding procedure maintains the bias of a variable for a heavily biased variables, up to some constant polynomial factor, we have,

$$\begin{aligned}
&I((y_{i_1}, y_{i_2}); (y_{j_1}, y_{j_2}))\\
&\quad \leqslant H(y_{i_1}, y_{i_2})\\
&\quad \leqslant H(y_{i_1}) + H(y_{i_2})\\
&\quad = O(-(1-\mathsf{poly}_{<1}(\mu_{i_1}))\log(1-\mathsf{poly}_{<1}(\mu_{i_1}))+\\
&\qquad O(-(1-\mathsf{poly}_{<1}(\mu_{i_2}))\log(1-\mathsf{poly}_{<1}(\mu_{i_2})))\\
&\quad \leqslant \mathsf{poly}_{<1}(\delta).
\end{aligned}$$

- Case 3: Consider the case when exactly two non-endpoints of an edge (w.l.o.g. of (i_1, j_i)) have large bias. This implies that $\langle \overline{\mathbf{w}}_{i_2}, \overline{\mathbf{w}}_{j_2} \rangle \leqslant \delta^{1/3}$. Using the analysis of the previous case we have $H(y_{i_1}), H(y_{j_1}) \leqslant \mathsf{poly}_{<1}(\delta)$. Mutual information can be bounded as follows:

$$\begin{aligned}
&I((y_{i_1}, y_{i_2}); (y_{j_1}, y_{j_2}))\\
&\quad \leqslant H((y_{i_1}, y_{i_2})) - H((y_{i_1}, y_{i_2})|(y_{j_1}, y_{j_2}))\\
&\quad \leqslant H(y_{i_1}) + H(y_{i_2}) - H(y_{i_2}|(y_{j_1}, y_{j_2}))
\end{aligned}$$

$$= H(y_{i_1}) + I((y_{j_1}, y_{j_2}); y_{i_2}) \tag{3.10}$$

$$= \mathsf{poly}_{<1}(\delta) + I((y_{j_1}, y_{j_2}); y_{i_2}). \tag{3.11}$$

Now,

$$\begin{aligned}
&I((y_{j_1}, y_{j_2}), y_{i_2})\\
&\quad = H((y_{j_1}, y_{j_2})) - H((y_{j_1}, y_{j_2})|y_{i_2})\\
&\quad \leqslant H(y_{j_1}) + H(y_{j_2}) - H(y_{j_2}|y_{i_2})\\
&\quad = H(y_{j_1}) + I(y_{j_2}; y_{i_2})\\
&\quad = \mathsf{poly}_{<1}(\delta) + I(y_{j_2}; y_{i_2}).
\end{aligned}$$

Therefore, we have

$$I(y_{i_1} y_{i_2}; y_{j_1} y_{j_2}) \leqslant \mathsf{poly}_{<1}(\delta) + I(y_{j_2}; y_{i_2}).$$

From Claim 1, $I(y_{j_2}; y_{i_2})$ is bounded above by $\mathsf{poly}_{<1}(\delta)$ as $\langle \overline{\mathbf{w}}_{i_2}, \overline{\mathbf{w}}_{j_2} \rangle \leqslant \delta^{1/3}$.

- Case 4: Consider the only remaining case in which exactly one variable, say X_{i_1}, has a large bias i.e. $\sqrt{1-\mu_{i_1}^2} \leqslant \delta^{1/3}$. From (3.9), it implies that pairwise inner products of $\overline{\mathbf{w}}_{i_2}, \overline{\mathbf{w}}_{j_1}$ and $\overline{\mathbf{w}}_{j_2}$ are at most $\delta^{1/3}$. Hence by Claim 1, we have $I(y_{i_2}; (y_{j_1}, y_{j_2})) \leqslant \mathsf{poly}_{<1}(\delta)$. As before from (3.10),

$$\begin{aligned}
I(y_{i_1}, y_{i_2}); (y_{j_1}, y_{j_2})) &\leqslant H(y_{i_1}) + I((y_{j_1}, y_{j_2}); y_{i_2})\\
&\leqslant \mathsf{poly}_{<1}(\delta).
\end{aligned}$$

We can now upper bound the variance of a cut produced by the randomized rounding in graph $\ell \in \mathcal{L}$. Define Y_ℓ to be a random variable which is equal to the total weight of active edges cut by the rounding procedure.

$$Y_\ell = \sum_{C \in \mathsf{Active}(S^\star)} \mathcal{E}_\ell(C) e(g).$$

LEMMA 3.10. *Fix a rounding function f_R given in Lemma 3.6 and let the* SDP *solution is δ independent then*

$$\mathsf{Var}(Y_\ell) \leqslant \frac{\mathsf{poly}_{<1}(\delta)}{\varepsilon^2} \mathbf{E}[Y_\ell]^2.$$

Proof. Let $\alpha := 0.8780$. Note that by Lemma 3.6, we have for an active edge $e(i,j)$,

$$\Pr[e(i,j) \text{ is cut }] \geqslant \alpha \cdot \frac{1 - \langle \mathbf{v_i}, \mathbf{v_j} \rangle}{2}. \tag{3.12}$$

We now lower bound the expected value of Y_ℓ.

$$\begin{aligned}
\mathbf{E}[Y_\ell] &= \sum_{e \in \mathsf{Active}(S^\star)} \mathcal{E}_\ell(e) \cdot \Pr[e(i,j) \text{ is cut}]\\
&\text{(from (3.12))}\\
&\geqslant \alpha \sum_{e \in \mathsf{Active}(S^\star)} \mathcal{E}_\ell(e) \cdot \frac{1 - \langle \mathbf{v_i}, \mathbf{v_j} \rangle}{2}\\
&= \alpha \cdot \sum_{e \in \mathsf{Active}(S^\star)} \mathcal{E}_\ell(e) (\|\mathbf{v}_{\{(\mathbf{i},\mathbf{j}),(\mathbf{0},\mathbf{1})\}}\|_2^2 + \|\mathbf{v}_{\{(\mathbf{i},\mathbf{j}),(\mathbf{1},\mathbf{0})\}}\|_2^2)\\
&\text{(from (3.2))}\\
&\geqslant \alpha \cdot \varepsilon/3 \cdot \mathsf{actdeg}_{S^\star}(\ell)
\end{aligned}$$

We can now bound the variance as follows:

$$
\begin{aligned}
\mathsf{Var}(Y_\ell) &= \sum_{i,j\in\mathsf{Active}(S^\star)} \mathcal{E}_\ell(i)\mathcal{E}_\ell(j)\mathsf{Cov}\left[\frac{1-y_{i_1}y_{i_2}}{2}, \frac{1-y_{j_1}y_{j_2}}{2}\right] \\
&= \sum_{i,j\in\mathsf{Active}(S^\star)} \mathcal{E}_\ell(i)\mathcal{E}_\ell(j)\left(\frac{1}{4}\cdot\mathsf{Cov}[y_{i_1}y_{i_2}, y_{j_1}y_{j_2}]\right) \\
&\leqslant \sum_{i,j\in\mathsf{Active}(S^\star)} \mathcal{E}_\ell(i)\mathcal{E}_\ell(j)[O(\sqrt{I(y_{i_1}y_{i_2}; y_{j_1}y_{j_2})})]
\end{aligned}
$$

(from Lemma 3.9)

$$
\leqslant \sum_{i,j\in\mathsf{Active}(S^\star)} \mathcal{E}_\ell(i)\mathcal{E}_\ell(j)\mathsf{poly}_{<1}\left(\sum_{\substack{a\in\{i_1,i_2\},\\ b\in\{j_1,j_2\}}} |\langle \mathbf{w_a}, \mathbf{w_b}\rangle|\right)
$$

(from Lemma 3.7)

$$
\begin{aligned}
&\leqslant \sum_{i,j\in\mathsf{Active}(S^\star)} \mathcal{E}_\ell(i)\mathcal{E}_\ell(j)\mathsf{poly}_{<1}\left(\mathop{\mathbf{E}}_{\substack{a\sim\{i_1,i_2\},\\ b\sim\{j_1,j_2\}}}[I(X_a;X_b)]\right) \\
&\leqslant \mathsf{actdeg}_{S^\star}(\ell)^2 \\
&\qquad \times \mathop{\mathbf{E}}_{i,j\sim\mathsf{actdist}_{S^\star}(\ell)} \mathsf{poly}_{<1} \mathop{\mathbf{E}}_{\substack{a\sim\{i_1,i_2\},\\ b\sim\{j_1,j_2\}}}[I(X_a;X_b)]
\end{aligned}
$$

(from concavity of $\mathsf{poly}_{<1}$)

$$
\begin{aligned}
&\leqslant \mathsf{actdeg}_{S^\star}(\ell)^2 \\
&\qquad \times \mathsf{poly}_{<1}\left(\mathop{\mathbf{E}}_{\substack{(i_1,i_2),\\ (j_1,j_2)}\sim\mathsf{actdist}_{S^\star}(\ell)} \mathop{\mathbf{E}}_{\substack{a\sim\{i_1,i_2\},\\ b\sim\{j_1,j_2\}}}[I(X_a;X_b)]\right) \\
&\leqslant \mathsf{poly}_{<1}(\delta)\cdot\mathsf{actdeg}_{S^\star}(\ell)^2,
\end{aligned}
$$

Thus, we have

$$
\mathsf{Var}(Y_\ell) \leqslant \frac{\mathsf{poly}_{<1}(\delta)}{\varepsilon^2}\mathbf{E}[Y_\ell]^2.
$$

COROLLARY 3.2. *If we set $r := \mathsf{poly}(k, 1/\varepsilon)$ then for every low variance instance $\ell \in [k]$, with probability at least $1-1/10k$ we have $\mathsf{val}(h^\star \cup g) \geqslant (0.878001-4\varepsilon)c_\ell$.*

Proof. Choosing r a large constant (and thus δ very small), by Lemma 3.10 and application of Chebyshev's Inequality, we can deduce that with probability at least $1-1/10k$, we have $Y_\ell \geqslant (1-\varepsilon)\mathbf{E}[Y_\ell]$. Thus, with probability at least $1-1/10k$, we have,

$$
\begin{aligned}
\mathsf{val}(h^\star \cup g, \mathcal{E}_\ell) &= \mathsf{val}(h^\star, \mathcal{E}_\ell) + Y_\ell \\
&\geqslant \mathsf{val}(h^\star, \mathcal{E}_\ell) + (1-\varepsilon)\mathbf{E}[Y_\ell] \\
&\geqslant (1-\varepsilon)\cdot\mathbf{E}[\mathsf{val}(h^\star, \mathcal{E}_\ell) + Y_\ell] \\
&= (1-\varepsilon)\cdot\mathbf{E}[\mathsf{val}(h^\star \cup g, W_\ell)] \\
&\geqslant (1-\varepsilon)\cdot 0.878001\cdot(1-3\varepsilon)\cdot c_\ell \\
&\geqslant (0.878001-4\varepsilon)\cdot c_\ell,
\end{aligned}
$$

where we have used Lemma 3.6 for the lower bound $\mathbf{E}[\mathsf{val}(h^\star \cup g, W_\ell)] \geqslant 0.878001\cdot(1-3\varepsilon)c_\ell$,

3.3.7 Post-Processing

LEMMA 3.11. *For all high variance instances $\ell \in [k]$, we have*

1. *$\mathsf{actdeg}_{S^\star}(\ell) \leqslant 2(1-\gamma)^t$.*
2. *For each of the first $t/2$ variables that were brought inside $S^\star$ because of instance ℓ, the total weight of edges from $\mathcal{E}_\ell$ incident on each of that variable and totally contained inside $S^\star$ is at least $20\cdot\mathsf{actdeg}_{S^\star}(\ell)$.*

Proof. Consider any *high variance* instance $\ell \in [k]$. Initially, when $S = \varnothing$, we have $\mathsf{actdeg}_\varnothing(\mathcal{E}_\ell) \leqslant 2$ since the weight of every edge is counted at most twice, once for each of the 2 active vertices of the edge, and $\sum_{e\in\mathcal{E}} \mathcal{E}_\ell(e) = 1$. For every v, note that $\mathsf{actdeg}_{S_2}(v, \mathcal{E}_\ell) \leqslant \mathsf{actdeg}_{S_1}(v, \mathcal{E}_\ell)$ whenever $S_1 \subseteq S_2$.

Let u be one of the vertices that ends up in $S^\star$ because of instance ℓ. Let S_u denote the set $S \subseteq S^\star$ just before u was brought into $S^\star$. When u is added to S_u, we know that $\mathsf{actdeg}_{S_u}(u, \mathcal{E}_\ell) \geqslant \gamma\cdot\mathsf{actdeg}_{S_u}(\ell)$. Hence, $\mathsf{actdeg}_{S_u\cup\{u\}}(\ell) \leqslant \mathsf{actdeg}_{S_u}(\ell) - \mathsf{actdeg}_{S_u}(u, \mathcal{E}_\ell) \leqslant (1-\gamma)\cdot\mathsf{actdeg}_{S_u}(\ell)$. Since t vertices were brought into $S^\star$ because of instance ℓ, and initially $\mathsf{actdeg}_\varnothing(\ell) \leqslant 2$, we get $\mathsf{actdeg}_{S^\star}(\ell) \leqslant 2(1-\gamma)^t$.

Now, let u be one of the first $t/2$ vertices that ends up in $S^\star$ because of instance ℓ. Since at least $t/2$ vertices are brought into $S^\star$ because of instance ℓ, after u, as above, we get $\mathsf{actdeg}_{S^\star}(\ell) \leqslant (1-\gamma)^{t/2}\cdot\mathsf{actdeg}_{S_u}(\ell)$. Combining with $\mathsf{actdeg}_{S_u}(u, \mathcal{E}_\ell) \geqslant \gamma\cdot\mathsf{actdeg}_{S_u}(\ell)$, we get $\mathsf{actdeg}_{S_u}(u, \mathcal{E}_\ell) \geqslant \gamma(1-\gamma)^{-t/2}\mathsf{actdeg}_{S^\star}(\ell)$, which is at least $21\cdot\mathsf{actdeg}_{S^\star}(\ell)$, by the choice of parameters. Since any edge incident on a vertex in $V\backslash S^\star$ contributes its weight to $\mathsf{actdeg}_{S^\star}(\ell)$, the total weight of edges incident on u and totally contained inside $S^\star$ is at least $20\cdot\mathsf{actdeg}_{S^\star}(\ell)$ as required.

We now describe a procedure PERTURB (see Figure 6) which takes $h^\star : S^\star \to \{0,1\}$ and $g : V\backslash S^\star \to \{0,1\}$, and produces a new $h : S^\star \to \{0,1\}$ such that for all (low variance as well as high variance) instances $\ell \in [k]$, $\mathsf{val}(h \cup g, \mathcal{E}_\ell)$ is not much smaller than $\mathsf{val}(h^\star \cup g, \mathcal{E}_\ell)$, and furthermore, for all high variance instances $\ell \in [k]$, $\mathsf{val}(h \cup g, \mathcal{E}_\ell)$ is large. The procedure works by picking a special vertex in $S^\star$ for every high variance instance and perturbing the assignment of $h^\star$ to these special vertices. The partial assignment h is what we will be using to argue that Step 5d of the algorithm produces a good Pareto approximation. More formally, we have the following Lemma.

Input: $h^\star : S^\star \to \{0,1\}$ and $g : V \backslash S^\star \to \{0,1\}$
Output: A perturbed assignment $h : S^\star \to \{0,1\}$.

1. Initialize $h \leftarrow h^\star$.
2. For $\ell = 1, \ldots, k$, if instance ℓ is a high variance instance case (i.e., $\mathsf{count}_\ell = t$), we pick a special variable $v_\ell \in S^\star$ associated to this instance as follows:
   (a) Let $B = \{v \in V \mid \exists \ell \in [k] \text{ with } \sum_{e \in \mathcal{E}, e \ni v} \mathcal{E}_\ell(e) \cdot e(h \cup g) \geqslant \frac{\varepsilon}{2k} \cdot \mathsf{val}(h \cup g, \mathcal{E}_\ell)\}$. Since the weight of each edge is counted at most twice, we know that $|B| \leqslant \frac{4k^2}{\varepsilon}$.
   (b) Let U be the set consisting of the first $t/2$ vertices brought into $S^\star$ because of instance ℓ.
   (c) Since ${}^t\!/_2 > |B| + k$, there exists some $u \in U$ such that $u \notin B \cup \{v_1, \ldots, v_{\ell-1}\}$. We define v_ℓ to be u.
   (d) By Lemma 3.11, the total $\mathcal{E}_\ell$ weight of edges that are incident on v_ℓ and only containing vertices from $S^\star$ is at least $20 \cdot \mathsf{actdeg}_{S^\star}(\ell)$. We update h by setting $h(v_\ell)$ to be that value from $\{0,1\}$ such that at least half of the $\mathcal{E}_\ell$ weight of these edges is satisfied.
3. Return the assignment h.

Figure 6: Procedure PERTURB for perturbing the optimal assignment

LEMMA 3.12. *For the assignment h obtained from Procedure* PERTURB *(see Figure 6), for each $\ell \in [k]$, $\mathsf{val}(h \cup g, \mathcal{E}_\ell) \geqslant (1 - {}^\varepsilon\!/_2) \cdot \mathsf{val}(h^\star \cup g, \mathcal{E}_\ell)$. Furthermore, for each* high variance *instance $\mathcal{E}_\ell$, $\mathsf{val}(h \cup g, \mathcal{E}_\ell) \geqslant 8 \cdot \mathsf{actdeg}_{S^\star}(\ell)$.*

Proof. Consider the special vertex v_ℓ that we choose for *high variance* instance $\ell \in [k]$. Since $v_\ell \notin B$, the edges incident on v_ℓ only contribute at most a ${}^\varepsilon\!/_{2k}$ fraction of the objective value in each instance. Thus, changing the assignment v_ℓ can reduce the value of any instance by at most a $\frac{\varepsilon}{2k}$ fraction of their current objective value. Also, we pick different special variables for each *high variance* instance. Hence, the total effect of these perturbations on any instance is that it reduces the objective value (given by $h^\star \cup g$) by at most $1 - (1 - \frac{\varepsilon}{2k})^k \leqslant \frac{\varepsilon}{2}$ fraction. Hence for all instances $\ell \in [k]$, $\mathsf{val}(h \cup g, \mathcal{E}_\ell) \geqslant (1 - {}^\varepsilon\!/_2) \cdot \mathsf{val}(h^\star \cup g, \mathcal{E}_\ell)$.

For a *high variance instance* $\ell \in [k]$, since $v_\ell \in U$, the vertex v_ℓ must be one of the first ${}^t\!/_2$ variables brought into $S^\star$ because of ℓ. Hence, by Lemma 3.11 the total weight of edges that are incident on v_ℓ and entirely contained inside $S^\star$ is at least $20 \cdot \mathsf{actdeg}_{S^\star}(\ell)$. Hence, there is an assignment to v_ℓ that satisfies at least at least half the weight of these MAX-CUT constraints in ℓ. At the end of the iteration when we pick an assignment to v_ℓ, we have $\mathsf{val}(h \cup g, \mathcal{E}_\ell) \geqslant 10 \cdot \mathsf{actdeg}_{S^\star}(\ell)$. Since the later perturbations do not affect value of this instance by more than ${}^\varepsilon\!/_2$ fraction, we get that for the final assignment h, $\mathsf{val}(h \cup g, \mathcal{E}_\ell) \geqslant (1 - {}^\varepsilon\!/_2) \cdot 10 \cdot \mathsf{actdeg}_{S^\star}(\ell) \geqslant 8 \cdot \mathsf{actdeg}_{S^\star}(\ell)$.

THEOREM 3.1. *Suppose we're given $\varepsilon \in (0, {}^1\!/_5]$, k simultaneous* MAX-CUT *instances $\mathcal{E}_1, \ldots, \mathcal{E}_k$ on n variables, and target objective value $c_1, \ldots, c_k$ with the guarantee that there exists an assignment $f^\star$ such that for each $\ell \in [k]$, we have $\mathsf{val}(f^\star, \mathcal{E}_\ell) \geqslant c_\ell$. Then, the algorithm* ALG-SIM-MAXCUT *runs in time $\exp({}^{k^3}\!/_{\varepsilon^2} \log({}^k\!/_{\varepsilon^2})) \cdot n^{\mathsf{poly}(k)}$, and with probability at least 0.9, outputs an assignment f such that for each $\ell \in [k]$, we have, $\mathsf{val}(f, \mathcal{E}_\ell) \geqslant (0.878001 - 5\varepsilon) \cdot c_\ell$.*

Proof. Let $\alpha := 0.878001$. By Corollary 3.2 and a union bound, with probability at least 0.9, over the choice of g, we have that for *every* low variance instance $\ell \in [k]$, $\mathsf{val}(h^\star \cup g, \mathcal{E}_\ell) \geqslant (\alpha - 4\varepsilon) \cdot c_\ell$. Henceforth we assume that the assignment g sampled in Step 5c of the algorithm is such that this event occurs. Let h be the output of the procedure PERTURB given in Figure 6 for the input $h^\star$ and g. By Lemma 3.12, h satisfies

1. For every instance $\ell \in [k]$, $\mathsf{val}(h \cup g, \mathcal{E}_\ell) \geqslant (1 - {}^\varepsilon\!/_2) \cdot \mathsf{val}(h^\star \cup g, \mathcal{E}_\ell)$.
2. For every high variance instance $\ell \in [k]$, $\mathsf{val}(h \cup g, \mathcal{E}_\ell) \geqslant 8 \cdot \mathsf{actdeg}_{S^\star}(\ell)$.

We now show that the desired Pareto approximation behavior is achieved when h is considered as the partial assignment in Step 5d of the algorithm. We analyze the guarantee for low and high variance instances separately.

For any *low variance* instance $\ell \in [k]$, from property 1 above, we have $\mathsf{val}(h \cup g, \mathcal{E}_\ell) \geqslant (1 - \varepsilon/2) \cdot \mathsf{val}(h^\star \cup g, \mathcal{E}_\ell)$. Since we know that $\mathsf{val}(h^\star \cup g, \mathcal{E}_\ell) \geqslant (\alpha - 4\varepsilon) \cdot c_\ell$, we have $\mathsf{val}(h \cup g, \mathcal{E}_\ell) \geqslant (\alpha - 5\varepsilon) \cdot c_\ell$.

For every high variance instance $\ell \in [k]$, since $h^\star = f^\star|_{S^\star}$, for any g we must have,

$$\begin{aligned}\mathsf{val}(h^\star \cup g, \mathcal{E}_\ell) &\geqslant \mathsf{val}(f^\star, \mathcal{E}_\ell) - \mathsf{actdeg}_{S^\star}(\ell)\\ &\geqslant c_\ell - \mathsf{actdeg}_{S^\star}(\ell)\end{aligned}$$

Combining this with properties 1 and 2 above, we get,

$$\begin{aligned}&\mathsf{val}(h \cup g, \mathcal{E}_\ell)\\ &\quad\geqslant (1 - \varepsilon/2) \cdot \max\{c_\ell - \mathsf{actdeg}_{S^\star}(\ell), 8 \cdot \mathsf{actdeg}_{S^\star}(\ell)\}\\ &\quad\geqslant (\alpha - \varepsilon) \cdot c_\ell.\end{aligned}$$

Thus, for all instances $\ell \in [k]$, we get $\mathsf{val}(h \cup g) \geqslant (\alpha - 5\varepsilon) \cdot c_\ell$. Since we are taking the best assignment $h \cup g$ at the end of the algorithm Alg-Sim-MaxCUT, the theorem follows.

Plugging the appropriate value of ε in Theorem 3.1 completes the proof of 0.8780-factor Pareto approximation (and hence min approximation) for simultaneous Max-Cut for arbitrary constant k.

4 Open Questions

The main open question we would like to highlight is the question of determining optimal approximability and inapproximability results for simultaneous approximation of constraint satisfaction problems (CSPs). In particular, it would be very interesting to develop techniques for showing nontrivial hardness of approximation in this context.

Acknowledgement

We would like to thank the authors of [ABG12] for making the prover code available for us. Our implementation of prover involves minor modifications of their code to suit our rounding algorithm. We also want to thank anonymous referees for helpful comments.

References

[ABG06] Eric Angel, Evripidis Bampis, and Laurent Gourvs. Approximation algorithms for the bicriteria weighted max-cut problem. *Discrete Applied Mathematics*, 154(12):1685 – 1692, 2006.

[ABG12] Per Austrin, Siavosh Benabbas, and Konstantinos Georgiou. Better balance by being biased: A 0.8776-approximation for max bisection. *CoRR*, abs/1205.0458, 2012.

[BKS15] Amey Bhangale, Swastik Kopparty, and Sushant Sachdeva. Simultaneous approximation of constraint satisfaction problems. In *International Colloquium on Automata, Languages, and Programming*, pages 193–205. Springer, 2015.

[FL92] Uriel Feige and László Lovász. Two-prover one-round proof systems: Their power and their problems (extended abstract). In *Proceedings of the Twenty-fourth Annual ACM Symposium on Theory of Computing*, STOC '92, pages 733–744, New York, NY, USA, 1992. ACM.

[GRW11] Christian Glaßer, Christian Reitwießner, and Maximilian Witek. Applications of discrepancy theory in multiobjective approximation. *CoRR*, abs/1107.0634, 2011.

[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, November 1995.

[Hås01] Johan Håstad. Some optimal inapproximability results. *J. ACM*, 48(4):798–859, July 2001.

[KKMO07] Subhash Khot, Guy Kindler, Elchanan Mossel, and Ryan O'Donnell. Optimal inapproximability results for max-cut and other 2-variable csps? *SIAM J. Comput.*, 37(1):319–357, April 2007.

[MOO05] Elchanan Mossel, Ryan O'Donnell, and Krzysztof Oleszkiewicz. Noise stability of functions with low in.uences invariance and optimality. In *Proceedings of the 46th Annual IEEE Symposium on Foundations of Computer Science*, FOCS '05, pages 21–30, Washington, DC, USA, 2005. IEEE Computer Society.

[OW08] Ryan O'Donnell and Yi Wu. An optimal sdp algorithm for max-cut, and equally optimal long code tests. In *Proceedings of the Fortieth Annual ACM Symposium on Theory of Computing*, STOC '08, pages 335–344, New York, NY, USA, 2008. ACM.

[RT12] Prasad Raghavendra and Ning Tan. Approximating csps with global cardinality constraints using sdp hierarchies. In *Proceedings of the twenty-third annual ACM-SIAM symposium on Discrete Algorithms*, pages 373–387. SIAM, 2012.

[Sjo09] Henrik Sjogren. Rigorous analysis of approximation algorithms for max 2-csp. 2009.

A Deferred Proofs

A.1 Proof of Claim 1

We need following bounds on the gaussian random variables.

Claim 2. *For all $x > 0$, $\Pr_{g \sim \mathcal{N}(0,1)}[|g| > x] \leqslant e^{-x^2/2}$.*

CLAIM 3. *For all* $1 > x > 0$, $\Pr_{g\sim\mathcal{N}(0,1)}[|g| < x] \leqslant x$.

Random process $\mathcal{P}$**:** Let $\mathbf{w}_1, \mathbf{w}_2, \mathbf{w}_3, \mathbf{w}_4 \in \mathbf{R}^4$ be unit vectors and $\mu_1, \mu_2, \mu_3, \mu_4$ be any real numbers. Consider the following random variables (y_1, y_2, y_3, y_4) where $y_i \in \{-1, +1\}$ which are sampled as follows: Pick a random vector $\mathbf{g} := (g_1, g_2, g_3, g_4) \in \mathbf{R}^4$ with each entry distributed as $\mathcal{N}(0, 1)$. Set

$$\begin{aligned} y_i &= -1 \qquad \text{if } \langle \mathbf{g}, \mathbf{w}_i\rangle \leqslant \mu_i, \\ &= +1 \qquad \text{otherwise.} \end{aligned}$$

The following lemmas gives sufficient conditions when $I(y_1, y_2; y_3, y_4)$ is *small.*

LEMMA A.1. *Suppose* $|\langle \mathbf{w}_i, \mathbf{w}_j\rangle| \leqslant \delta$ *for all* $i, j \in [4]$, $i \neq j$ *and* y_i*s are sampled according to the random process* $\mathcal{P}$*, then for all* $\mathbf{b} \in \{-1, +1\}^4$*, we have*

$$\left|\Pr[(y_1, y_2, y_3, y_4) = \mathbf{b}] - \prod_{1\leqslant i\leqslant 4} \Pr[y_i = b_i]\right| = O(\delta^{1/4}),$$

In fact, the joint distribution on any subset of variables is close to its product distribution pointwise *with an additive error of at most* $O(\delta^{1/4})$.

Proof. Assume that $0 < \delta < 1/100$ (otherwise, the lemma is trivial). Let $\mathbf{e}_i$ is a unit vector with 1 in the i^{th} coordinate. By rotational symmetry, we can assume that $\langle \mathbf{w}_i, \mathbf{e}_i\rangle \geqslant 1 - 20\delta$ for all i. We can write vector $\mathbf{w}_i = \sqrt{1-\delta_i}\mathbf{e}_i + \sqrt{\delta_i}\eta_i$ where η_i is a unit vector orthogonal to $\mathbf{e}_i$. The conditions on inner products therefore imply each $\delta_i < 40\delta$. We will prove the lemma for $\mathbf{b} = (-1, -1, -1, -1)$ (all other cases are similar). We have,

$$\begin{aligned} \Pr[y_i = -1, &\forall i \in [4]] \\ &= \Pr[\forall i, \langle \mathbf{g}, \mathbf{w}_i\rangle \leqslant \mu_i] \\ &= \Pr[\forall i, \sqrt{1-\delta_i} g_i + \sqrt{\delta_i}\langle \mathbf{g}, \eta_i\rangle \leqslant \mu_i] \end{aligned}$$

Let B be the following event,
B : There exists $1 \leqslant i \leqslant 4$, such that $|\langle \mathbf{g}, \eta_i\rangle| \geqslant 1/\delta^{1/4}$.

By union bound,

$$\begin{aligned} \Pr[B] &= \sum_i \Pr[|\langle \mathbf{g}, \eta_i\rangle| \geqslant 1/\delta^{1/4}] \\ &\leqslant 4 \cdot \Pr[|\langle \mathbf{g}, \eta_1\rangle| \geqslant 1/\delta^{1/4}] \\ &= 4 \cdot \Pr_{g\sim\mathcal{N}(0,1)}[|g| \geqslant 1/\delta^{1/4}] \\ &\leqslant 4e^{-\frac{1}{2\sqrt{\delta}}}, \end{aligned}$$

where last inequality uses Claim 2. Now,

$$\begin{aligned} \text{(A.1)} \quad \Pr[y_i = -1, &\forall 1 \leqslant i \in [4]] \\ &= \Pr[B] \cdot \Pr[y_i = -1, \forall i \in [4] | B] \\ &\quad + \Pr[\overline{B}] \cdot \Pr[y_i = -1, \forall i \in [4] | \overline{B}] \\ \text{(A.2)} \quad &\leqslant 4e^{-\frac{1}{2\sqrt{\delta}}} \cdot 1 + \Pr[y_i = -1, \forall i \in [4] | \overline{B}], \end{aligned}$$

We now estimate the probability conditioned on event $\overline{B}$.

$$\begin{aligned} \text{(A.3)} \quad \Pr[y_i = -1, &\forall i \in [4] | \overline{B}] \\ &= \Pr[\forall i, \sqrt{1-\delta_i} g_i + \sqrt{\delta_i}\langle \mathbf{g}, \eta_i\rangle \leqslant \mu_i | \overline{B}] \\ &\leqslant \Pr[\forall i, \sqrt{1-\delta_i} g_i \leqslant \mu_i + \sqrt{\delta_i} \cdot \frac{1}{\delta^{1/4}}] \\ \text{(A.4)} \quad & (g_i \text{ independent}) \\ &= \prod_i \Pr[\sqrt{1-\delta_i} g_i \leqslant \mu_i + \sqrt{\delta_i} \cdot \frac{1}{\delta^{1/4}}] \\ \text{(A.5)} \quad & (\text{from } \delta_i \leqslant 40\delta) \\ &\leqslant \prod_i \Pr[\sqrt{1-\delta_i} g_i \leqslant \mu_i + \sqrt{40}\delta^{1/4}] \\ \text{(A.6)} \quad & (\text{from } \delta_i \leqslant 1/2) \\ &\leqslant \prod_i \Pr[g_i \leqslant (1+\delta_i)(\mu_i + \sqrt{40}\delta^{1/4})] \\ \text{(A.7)} \quad & (\text{from } \delta_i \leqslant 1/2) \\ &\leqslant \prod_i \Pr[g_i \leqslant \mu_i + \delta_i\mu_i + 3/2 \cdot \sqrt{40}\delta^{1/4})] \\ &\leqslant \prod_i \Pr[g_i \leqslant (\mu_i + \delta_i\mu_i + 15\delta^{1/4})]. \end{aligned}$$

We now analyse the above probability in cases, and show the following:

$$\text{(A.8)} \quad \Pr[g_i \leqslant \mu_i + \delta_i\mu_i + 15\delta^{1/4})] \leqslant \prod_i \Pr[g_i \leqslant \mu_i] + O(\delta^{1/4})$$

Notice that

$$\begin{aligned} \text{(A.9)} \quad & \prod_i \Pr[g_i \leqslant \mu_i + c\delta^{1/4}] \\ &\leqslant \prod_i \Pr[g_i \leqslant \mu_i] + \Pr[|g_i| \leqslant c\delta^{1/4}] \\ (\text{from Claim 3}) \quad &\leqslant \left(\prod_{1\leqslant i\leqslant 4} \Pr[y_i = b_i] + c\delta^{1/4}\right) \\ \text{(A.10)} \quad &\leqslant \prod_{1\leqslant i\leqslant 4} \Pr[y_i = b_i] + O(\delta^{1/4}) \end{aligned}$$

- Case 1: $\mu_i < 0$.
  In this case, we can directly say the following.

$$\prod_i \Pr[g_i \leqslant \mu_i + \delta_i\mu_i + 15\delta^{1/4})] \leqslant \prod_i \Pr[g_i \leqslant \mu_i + 15\delta^{1/4}].$$

- Case 2: $0 \leqslant \mu_i \leqslant \frac{10}{\delta^{3/4}}$ We can say the following because $\delta_i < 40\delta$.

$$\prod_i \Pr[g_i \leqslant \mu_i + \delta_i\mu_i + 15\delta^{1/4}] \leqslant \prod_i \Pr[g_i \leqslant \mu_i + O(\delta^{1/4})]$$

- Case 3: $\mu_i > \frac{10}{\delta^{3/4}}$ In this case, since μ_i is large, we have the following from Claim 2.

$$\prod_i \Pr[g_i \leqslant \mu_i] \geqslant 1 - o(\delta^{1/4})$$

  Therefore,

$$\prod_i \Pr[g_i \leqslant \mu_i + \delta_i\mu_i + 15\delta^{1/4}] \leqslant 1 \leqslant \prod_i \Pr[g_i \leqslant \mu_i] + o(\delta^{1/4})$$

Form (A.2), (A.9) and (A.10) we get

$$\Pr[(y_1, y_2, y_3, y_4) = \mathbf{b}] - \prod_{1\leqslant i\leqslant 4} \Pr[y_i = b_i] \leqslant O(\delta^{1/4}).$$

The other direction can be shown in an analogous way.

We can now bound the Mutual information between (y_1, y_2) and (y_3, y_4) if the vectors $\mathbf{w}_i$ satisfy the condition from Lemma A.1

LEMMA A.2. *Suppose $|\langle \mathbf{w}_i, \mathbf{w}_j\rangle| \leqslant \delta$ for all $i, j \in [4]$ and $i \neq j$, then $I((y_1, y_2); (y_3, y_4)) \leqslant \mathsf{poly}_{<1}(\delta)$, where y_i are sampled according to the random process $\mathcal{P}$.*

Proof. The lemma follows from Lemma A.1 as the distribution is *close* to the product distribution.

To formally prove the lemma, first we assume that each of the random variables y_i is not heavily biased i.e. $\Pr[y_i = -1] \in [\delta^{1/100}, 1 - \delta^{1/100}]$. Using the definition of mutual information,

(A.11)

$$I((y_1, y_2); (y_3, y_4)) =$$

(A.12)

$$\sum_{\substack{b_1,b_2,b_3,b_4\\ \{-1+1\}}} \left[[\Pr[\mathbf{y} = \mathbf{b}] \cdot \log \frac{\Pr[\mathbf{y} = \mathbf{b}]}{\Pr[(y_1, y_2) = (b_1, b_2)] \cdot \Pr[(y_3, y_4) = (b_3, b_4)]} \right]$$

Form Lemma A.1, we have

$$\Pr[(y_1, y_2) = (b_1, b_2)] \geqslant \Pr[y_1 = b_1] \Pr[y_2 = b_2] - O(\delta^{1/4})$$
$$\Pr[(y_3, y_4) = (b_3, b_4)] \geqslant \Pr[y_3 = b_3] \Pr[y_4 = b_4] - O(\delta^{1/4})$$

Plugging any simplifying in (A.11), we get

$$I((y_1, y_2); (y_3, y_4)) \leqslant \sum_{b_1,b_2,b_3,b_4\{-1+1\}} \Pr[\mathbf{y} = \mathbf{b}] \cdot \log \frac{\prod_{1\leqslant i\leqslant 4} \Pr[y_i = b_i] + O(\delta^{1/4})}{\prod_{1\leqslant i\leqslant 4} \Pr[y_i = b_i] - O(\delta^{1/4})}$$

As each variable is not heavily biased, we have $\prod_{1\leqslant i\leqslant 4} \Pr[y_i = b_i] \geqslant \delta^{1/25}$ and hence the log in the above expression can be upper bounded by $\log \frac{\delta^{1/25} + O(\delta^{1/4})}{\delta^{1/25} - O(\delta^{1/4})}$ which is at most $\log(1 + O(\delta^{1/10})) \leqslant O(\delta^{1/10})$. Hence we have

$$I((y_1, y_2); (y_3, y_4)) \leqslant O(\delta^{1/10}).$$

If a variable is heavily biased, suppose say y_1 has large bias, then we can claim $I((y_1, y_2); (y_3, y_4)) \leqslant \mathsf{poly}_{<1}(\delta) + I(y_2; (y_3, y_4))$ using derivation similar to (3.11) and then proceed by upper bounding $I(y_2; (y_3, y_4))$ in a similar fashion as above.

Proof of Claim 1: The proof follows from Lemma A.2 noting the fact that the upper bound is independent of μ_i.

A.2 Proof of Lemma 3.1

Proof. Item 1 of the lemma follows from Chebyshev's inequality. We now focus on the proof of Item 2. We have

$$\mathsf{Uvar}_\ell \geqslant \delta_0\varepsilon_0^2 \cdot \mathsf{Lmean}_\ell^2$$
$$\Rightarrow \sum_{e\sim_S e'} \mathcal{E}_\ell(e)\mathcal{E}_\ell(e') \geqslant \delta_0\varepsilon_0^2 \cdot \mathsf{Lmean}_\ell^2$$

Let e_0 be an edge in $\mathsf{Active}(S)$ that maximizes $\sum_{e \sim_S e_0} \mathcal{E}_\ell(e)$. We can now upper bound the expression on the left as follows

$$\sum_{e \sim_S e'} \mathcal{E}_\ell(e)\mathcal{E}_\ell(e') \leqslant \sum_{e \sim_S e_0} \mathcal{E}_\ell(e) \cdot \sum_{e \in \mathsf{Active}(S)} \mathcal{E}_\ell(e).$$

Therefore, we have

$$\begin{aligned}
&\sum_{e \sim_S e_0} \mathcal{E}_\ell(e) \cdot \sum_{e \in \mathsf{Active}(S)} \mathcal{E}_\ell(e) \\
&\qquad\qquad \geqslant \delta_0 \varepsilon_0^2 \cdot \mathsf{Lmean}_\ell^2 \\
&\qquad\qquad \geqslant \delta_0 \varepsilon_0^2 \cdot \tau^2 \cdot \left(\sum_{e \in \mathsf{Active}(S)} \mathcal{E}_\ell(e) \right)^2 \\
&\Rightarrow \sum_{e \sim_S e_0} \mathcal{E}_\ell(e) \geqslant \delta_0 \varepsilon_0^2 \cdot \tau^2 \cdot \sum_{e \in \mathsf{Active}(S)} \mathcal{E}_\ell(e)
\end{aligned}$$

Let v be the end vertex of e_0 that has greater weight of active edges adjacent to it, $v \in V \backslash S$. We can say the following

$$\mathsf{actdeg}_S(v, \ell) \geqslant \frac{1}{2} \cdot \delta_0 \varepsilon_0^2 \cdot \tau^2 \cdot \sum_{e \in \mathsf{Active}(S)} \mathcal{E}_\ell(e).$$

From the definition of $\mathsf{actdeg}_S(\ell)$, we can say the following

$$\mathsf{actdeg}_S(\ell) \leqslant 2 \cdot \sum_{e \in \mathsf{Active}(S)} \mathcal{E}_\ell(e),$$

as each edge could contribute at most twice to the sum, once for each end vertex. This gives us the following required result.

$$\mathsf{actdeg}_S(v, \ell) \geqslant \frac{1}{4} \cdot \delta_0 \varepsilon_0^2 \cdot \tau^2 \cdot \mathsf{actdeg}_S(\ell).$$