Sub-exponential Approximation Schemes for CSPs: from Dense to Almost Sparse*

Dimitris Fotakis[1], Michael Lampis[2], and Vangelis Th. Paschos[2]

1 **School of Electrical and Computer Engineering, National Technical University of Athens, Greece**
`fotakis@cs.ntua.gr`

2 **PSL Research University, Université Paris Dauphine**
LAMSADE, CNRS UMR7243
`michail.lampis@dauphine.fr,paschos@lamsade.dauphine.fr`

Abstract

It has long been known, since the classical work of (Arora, Karger, Karpinski, JCSS 99), that Max-CUT admits a PTAS on dense graphs, and more generally, Max-k-CSP admits a PTAS on "dense" instances with $\Omega(n^k)$ constraints. In this paper we extend and generalize their exhaustive sampling approach, presenting a framework for $(1-\varepsilon)$-approximating any Max-k-CSP problem in *sub-exponential* time while significantly relaxing the denseness requirement on the input instance.

Specifically, we prove that for any constants $\delta \in (0,1]$ and $\varepsilon > 0$, we can approximate Max-k-CSP problems with $\Omega(n^{k-1+\delta})$ constraints within a factor of $(1-\varepsilon)$ in time $2^{O(n^{1-\delta}\ln n/\varepsilon^3)}$. The framework is quite general and includes classical optimization problems, such as Max-CUT, Max-DICUT, Max-k-SAT, and (with a slight extension) k-Densest Subgraph, as special cases. For Max-CUT in particular (where $k=2$), it gives an approximation scheme that runs in time sub-exponential in n even for "almost-sparse" instances (graphs with $n^{1+\delta}$ edges).

We prove that our results are essentially best possible, assuming the ETH. First, the density requirement cannot be relaxed further: there exists a constant $r < 1$ such that for all $\delta > 0$, Max-k-SAT instances with $O(n^{k-1})$ clauses cannot be approximated within a ratio better than r in time $2^{O(n^{1-\delta})}$. Second, the running time of our algorithm is almost tight *for all densities*. Even for Max-CUT there exists $r < 1$ such that for all $\delta' > \delta > 0$, Max-CUT instances with $n^{1+\delta}$ edges cannot be approximated within a ratio better than r in time $2^{n^{1-\delta'}}$.

1998 ACM Subject Classification F.1.3 Complexity Measures and Classes, F.2.2 Nonnumerical Algorithms and Problems, G.2.1 Combinatorics, G.2.2 Graph Theory

Keywords and phrases Algorithm, Complexity, Polynomial and Subexponential Approximation, Max-k-csp, Sampling, Randomized rounding

Digital Object Identifier 10.4230/LIPIcs.xxx.yyy.p

1 Introduction

The complexity of Constraint Satisfaction Problems (CSPs) has long played a central role in theoretical computer science and it quickly became evident that almost all interesting CSPs are NP-complete [29]. Thus, since approximation algorithms are one of the standard

* This research was supported by the project AlgoNow, co-financed by the European Union (European Social Fund - ESF) and Greek national funds, through the Operational Program "Education and Lifelong Learning" of the National Strategic Reference Framework (NSRF) - Research Funding Program: THALES, investing in knowledge society through the European Social Fund.

Conference title on which this volume is based on.
Editors: Billy Editor, Bill Editors; pp. 1–30
Leibniz International Proceedings in Informatics
LIPICS Schloss Dagstuhl – Leibniz-Zentrum für Informatik, Dagstuhl Publishing, Germany

tools for dealing with NP-hard problems, the question of approximating the corresponding optimization problems (Max-CSP) has attracted significant interest over the years [30]. Unfortunately, most CSPs typically resist this approach: not only are they APX-hard [24], but quite often the best polynomial-time approximation ratio we can hope to achieve for them is that guaranteed by a trivial random assignment [22]. This striking behavior is often called *approximation resistance*.

Approximation resistance and other APX-hardness results were originally formulated in the context of *polynomial-time* approximation. It would therefore seem that one conceivable way for working around such barriers could be to consider approximation algorithms running in super-polynomial time, and indeed super-polynomial approximation for NP-hard problems is a topic that has been gaining more attention in the literature recently [11, 8, 7, 12, 13, 14]. Unfortunately, the existence of quasi-linear PCPs with small soundness error, first given in the work of Moshkovitz and Raz [25], established that approximation resistance is a phenomenon that carries over even to *sub-exponential* time approximation, essentially "killing" this approach for CSPs. For instance, we now know that if, for any $\varepsilon > 0$, there exists an algorithm for Max-3-SAT with ratio $7/8 + \varepsilon$ running in time $2^{n^{1-\varepsilon}}$ this would imply the existence of a sub-exponential *exact* algorithm for 3-SAT, disproving the Exponential Time Hypothesis (ETH). It therefore seems that sub-exponential time does not improve the approximability of CSPs, or put another way, for many CSPs obtaining a very good approximation ratio requires almost as much time as solving the problem exactly.

Despite this grim overall picture, many positive approximation results for CSPs have appeared over the years, by taking advantage of the special structure of various classes of instances. One notable line of research in this vein is the work on the approximability of *dense* CSPs, initiated by Arora, Karger and Karpinski [4] and independently by de la Vega [15]. The theme of this set of results is that the problem of maximizing the number of satisfied constraints in a CSP instance with arity k (Max-k-CSP) becomes significantly easier if the instance contains $\Omega(n^k)$ constraints. More precisely, it was shown in [4] that Max-k-CSP admits a *polynomial-time approximation scheme* (PTAS) on dense instances, that is, an algorithm which for any constant $\varepsilon > 0$ can in time polynomial in n produce an assignment that satisfies $(1 - \varepsilon)$OPT constraints. Subsequent work produced a stream of positive [17, 5, 2, 10, 9, 21, 3, 20, 23] (and some negative [16, 1]) results on approximating CSPs which are in general APX-hard, showing that dense instances form an island of tractability where many optimization problems which are normally APX-hard admit a PTAS.

Our contribution: The main goal of this paper is to use the additional power afforded by sub-exponential time to extend this island of tractability as much as possible. To demonstrate the main result, consider a concrete CSP such as Max-3-SAT. As mentioned, we know that sub-exponential time does not in general help us approximate this problem: the best ratio achievable in, say, $2^{\sqrt{n}}$ time is still 7/8. On the other hand, this problem admits a PTAS on instances with $\Omega(n^3)$ clauses. This density condition is, however, rather strict, so the question we would like to answer is the following: Can we efficiently approximate a larger (and more sparse) class of instances while using sub-exponential time?

In this paper we provide a positive answer to this question, not just for Max-3-SAT, but also for any Max-k-CSP problem. Specifically, we show that for any constants $\delta \in (0, 1]$, $\varepsilon > 0$ and integer $k \geq 2$, there is an algorithm which achieves a $(1 - \varepsilon)$ approximation of Max-k-CSP instances with $\Omega(n^{k-1+\delta})$ constraints in time $2^{O(n^{1-\delta} \ln n/\varepsilon^3)}$. A notable special case of this result is for $k = 2$, where the input instance can be described as a graph. For this case, which contains classical problems such as Max-CUT, our algorithm gives an approximation scheme running in time $2^{O(\frac{n}{\Delta} \ln n/\varepsilon^3)}$ for graphs with average degree Δ. In other words, this is an approximation scheme that runs in time *sub-exponential in n* even for almost sparse instances where the average degree is $\Delta = n^\delta$ for some small $\delta > 0$. More generally, our algorithm provides a trade-off between the time available and the density of the instances we can handle. For graph problems ($k = 2$) this trade-off covers the whole spectrum from dense to almost sparse instances, while for general Max-k-CSP, it covers instances where the number of constraints ranges from $\Theta(n^k)$ to $\Theta(n^{k-1})$.

Techniques: The algorithms in this paper are an extension and generalization of the *exhaustive sampling* technique given by Arora, Karger and Karpinski [4], who introduced a framework of smooth polynomial integer programs to give a PTAS for dense Max-k-CSP. The basic idea of that work can most simply be summarized for Max-CUT. This problem can be recast as the problem of maximizing a quadratic function over n boolean variables. This is of course a hard problem, but suppose that we could somehow "guess" for each vertex how many of its neighbors belong in each side of the cut. This would make the quadratic problem linear, and thus much easier. The main intuition now is that, if the graph is dense, we can take a sample of $O(\log n)$ vertices and guess their partition in the optimal solution. Because every non-sample vertex will have "many" neighbors in this sample, we can with high confidence say that we can estimate the fraction of neighbors on each side for all vertices. The work of de la Vega [15] uses exactly this algorithm for Max-CUT, greedily deciding the vertices outside the sample. The work of [4] on the other hand pushed this idea to its logical conclusion, showing that it can be applied to degree-k polynomial optimization problems, by recursively turning them into linear programs whose coefficients are estimated from the sample. The linear programs are then relaxed to produce fractional solutions, which can be rounded back into an integer solution to the original problem.

On a very high level, the approach we follow in this paper retraces the steps of [4]: we formulate Max-k-CSP as a degree-k polynomial maximization problem; we then recursively decompose the degree-k polynomial problem into lower-degree polynomial optimization problems, estimating the coefficients by using a sample of variables for which we try all assignments; the result of this process is an integer linear program, for which we obtain a fractional solution in polynomial time; we then perform randomized rounding to obtain an integer solution that we can use for the original problem.

The first major difference between our approach and [4] is of course that we need to use a larger sample. This becomes evident if one considers Max-CUT on graphs with average degree Δ. In order to get the sampling scheme to work we must be able to guarantee that each vertex outside the sample has "many" neighbors inside the sample, so we can safely estimate how many of them end up on each side of the cut. For this, we need a sample of size at least $n \log n/\Delta$. Indeed, we use a sample of roughly this size, and exhausting all assignments to the sample is what dominates the running time of our algorithm. As we argue later, not only is the sample size we use essentially tight, but more generally the running time of our algorithm is essentially optimal (under the ETH).

Nevertheless, using a larger sample is not in itself sufficient to extend the scheme of [4] to non-dense instances. As observed in [4] "to achieve a multiplicative approximation for dense instances it suffices to achieve an additive approximation for the nonlinear integer programming problem". In other words, one of the basic ingredients of the analysis of [4] is that additive approximation errors of the order εn^k can be swept under the rug, because we know that in a dense instance the optimal solution has value $\Omega(n^k)$. This is *not* true in our case, and we are therefore forced to give a more refined analysis of the error of our scheme, independently bounding the error introduced in the first step (coefficient estimation) and the last (randomized rounding).

A further complication arises when considering Max-k-CSP for $k > 2$. The scheme of [4] recursively decomposes such dense instances into lower-order polynomials which retain the same "good" properties. This seems much harder to extend to the non-dense case, because intuitively if we start from a non-dense instance the decomposition could end up producing some dense and some sparse sub-problems. Indeed we present a scheme that approximates Max-k-CSP with $\Omega(n^{k-1+\delta})$ constraints, but does not seem to extend to instances with fewer than n^{k-1} constraints. As we will see, there seems to be a fundamental complexity-theoretic justification explaining exactly why this decomposition method cannot be extended further.

To ease presentation, we first give all the details of our scheme for the special case of Max-CUT in Section 3. We then present the full framework for approximating *smooth polynomials* in Section 4; this implies the approximation result for Max-k-SAT and more generally Max-k-CSP. We then show in Section 5 that it is possible to extend our framework to handle k-Densest Subgraph, a problem which can be expressed as the maximization of a polynomial subject to linear constraints. For this problem we obtain an approximation scheme which, given a graph with average degree $\Delta = n^\delta$ gives a $(1-\varepsilon)$ approximation in time $2^{O(n^{1-\delta/3}\ln n/\varepsilon^3)}$. Observe that this extends the result of [4] for this problem not only in terms of the density of the input instance, but also in terms of k (the result of [4] required that $k = \Omega(n)$).

Hardness: What makes the results of this paper more interesting is that we can establish that in many ways they are essentially best possible, if one assumes the ETH. In particular, there are at least two ways in which one may try to improve on these results further: one would be to improve the running time of our algorithm, while another would be to extend the algorithm to the range of densities it cannot currently handle. In Section 6 we show that both of these approaches would face significant barriers. Our starting point is the fact that (under ETH) it takes exponential time to approximate Max-CUT arbitrarily well on sparse instances, which is a consequence of the existence of quasi-linear PCPs. By manipulating such Max-CUT instances, we are able to show that for *any* average degree $\Delta = n^\delta$ with $\delta < 1$ the time needed to approximate Max-CUT arbitrarily well almost matches the performance of our algorithm. Furthermore, starting from sparse Max-CUT instances, we can produce instances of Max-k-SAT with $O(n^{k-1})$ clauses while preserving hardness of approximation. This gives a complexity-theoretic justification for our difficulties in decomposing Max-k-CSP instances with less than n^{k-1} constraints.

2 Notation and Preliminaries

An n-variate degree-d polynomial $p(\vec{x})$ is β*-smooth* [4], for some constant $\beta \geq 1$, if for every $\ell \in \{0,\dots,d\}$, the absolute value of each coefficient of each degree-ℓ monomial in the expansion of $p(\vec{x})$ is at most $\beta n^{d-\ell}$. An n-variate degree-d β-smooth polynomial $p(\vec{x})$ is δ*-bounded*, for some constant $\delta \in (0,1]$, if for every ℓ, the sum, over all degree-ℓ monomials in $p(\vec{x})$, of the absolute values of their coefficients is $O(\beta n^{d-1+\delta})$. Therefore, for any n-variate degree-d β-smooth δ-bounded polynomial $p(\vec{x})$ and any $\vec{x} \in \{0,1\}^n$, $|p(\vec{x})| = O(d\beta n^{d-1+\delta})$.

Optimization Problem. Our algorithms for Max-CUT, Max-k-SAT, and Max-k-CSP are obtained by reducing to the following problem: Given an n-variate d-degree β-smooth δ-bounded polynomial $p(\vec{x})$, we seek a binary vector $\vec{x}^* \in \{0,1\}^n$ that maximizes p.

Polynomial Decomposition and General Approach. As in [4, Lemma 3.1], our general approach is motivated by the fact that any n-variate d-degree β-smooth polynomial $p(\vec{x})$ can be naturally decomposed into a collection of n polynomials $p_j(\vec{x})$. Each of them has degree $d-1$ and at most n variables and is β-smooth.

▶ Proposition 1 ([4]). Let $p(\vec{x})$ be any n-variate degree-d β-smooth polynomial. Then, there exist a constant c and degree-$(d-1)$ β-smooth polynomials $p_j(\vec{x})$ such that $p(\vec{x}) = c + \sum_{j=1}^{n} x_j p_j(\vec{x})$.

Graph Optimization Problems. Let $G(V, E)$ be a (simple) graph with n vertices and m edges. For each vertex $i \in V$, $N(i)$ denotes i's neighborhood in G, i.e., $N(i) = \{j \in V : \{i, j\} \in E\}$. We let $\deg(i) = |N(i)|$ be the degree of i in G and $\Delta = 2|E|/n$ denote the average degree of G. We say that a graph G is *δ-almost sparse*, for some constant $\delta \in (0, 1]$, if $m = \Omega(n^{1+\delta})$ (and thus, $\Delta = \Omega(n^{\delta})$).

In Max-CUT, we seek a partitioning of the vertices of G into two sets S_0 and S_1 so that the number of edges with endpoints in S_0 and S_1 is maximized. If G has m edges, the number of edges in the optimal cut is at least $m/2$.

In k-Densest Subgraph, given an undirected graph $G(V, E)$, we seek a subset C of k vertices so that the induced subgraph $G[C]$ has a maximum number of edges.

Constraint Satisfaction Problems. An instance of (boolean) Max-k-CSP with n variables consists of m boolean constraints $f_1, \ldots, f_m$, where each $f_j : \{0, 1\}^k \to \{0, 1\}$ depends on k variables and is satisfiable, i.e., f_j evaluates to 1 for some truth assignment. We seek a truth assignment to the variables that maximizes the number of satisfied constraints. Max-k-SAT is a special case of Max-k-CSP where each constraint f_j is a disjunction of k literals. An averaging argument implies that the optimal assignment of a Max-k-CSP (resp. Max-k-SAT) instance with m constraints satisfies at least $2^{-k}m$ (resp. $(1 - 2^{-k})m$) of them. We say that an instance of Max-k-CSP is *δ-almost sparse*, for some constant $\delta \in (0, 1]$, if the number of constraints is $m = \Omega(n^{k-1+\delta})$.

Using standard arithmetization techniques (see e.g., [4, Sec. 4.3]), we can reduce any instance of Max-k-CSP with n variables to an n-variate degree-k polynomial $p(\vec{x})$ so that the optimal truth assignment for Max-k-CSP corresponds to a maximizer $\vec{x}^* \in \{0, 1\}$ of $p(\vec{x})$ and the value of the optimal Max-k-CSP solution is equal to $p(\vec{x}^*)$. Since each k-tuple of variables can appear in at most 2^k different constraints, $p(\vec{x})$ is β-smooth, for $\beta \in [1, 4^k]$, and has at least m and at most $4^k m$ monomials. Moreover, if the instance of Max-k-CSP has $m = \Theta(n^{k-1+\delta})$ constraints, then $p(\vec{x})$ is δ-bounded and its maximizer $\vec{x}^*$ has $p(\vec{x}^*) = \Omega(n^{k-1+\delta})$.

Notation and Terminology. An algorithm has *approximation ratio* $\rho \in (0, 1]$ (or is *ρ-approximate*) if for all instances, the value of its solution is at least ρ times the value of the optimal solution.

For graphs with n vertices or CSPs with n variables, we say that an event E happens with high probability (or whp.), if E happens with probability at least $1 - 1/n^c$, for some constant $c \geq 1$.

For brevity and clarity, we sometimes write $\alpha \in (1 \pm \epsilon_1)\beta \pm \epsilon_2\gamma$, for some constants $\epsilon_1, \epsilon_2 > 0$, to denote that $(1 - \epsilon_1)\beta - \epsilon_2\gamma \leq \alpha \leq (1 + \epsilon_1)\beta + \epsilon_2\gamma$.

3 Approximating Max-CUT in Almost Sparse Graphs

In this section, we apply our approach to Max-CUT, which serves as a convenient example and allows us to present the intuition and the main ideas.

The Max-CUT problem in a graph $G(V, E)$ is equivalent to maximizing, over all binary vectors $\vec{x} \in \{0, 1\}^n$, the following n-variate degree-2 2-smooth polynomial

$$p(\vec{x}) = \sum_{\{i,j\} \in E} (x_i(1 - x_j) + x_j(1 - x_i))$$

Setting a variable x_i to 0 indicates that the corresponding vertex i is assigned to the left side of the cut, i.e., to S_0, and setting x_i to 1 indicates that vertex i is assigned to the right side of the cut, i.e., to S_1. We assume that G is δ-almost sparse and thus, has $m = \Omega(n^{1+\delta})$ edges and average degree $\Delta = \Omega(n^\delta)$. Moreover, if $m = \Theta(n^{1+\delta})$, $p(\vec{x})$ is δ-bounded, since for each edge $\{i, j\} \in E$, the monomial $x_i x_j$ appears with coefficient -2 in the expansion of p, and for each vertex $i \in V$, the monomial x_i appears with coefficient $\deg(i)$ in the expansion of p. Therefore, for $\ell \in \{1, 2\}$, the sum of the absolute values of the coefficients of all monomials of degree ℓ is at most $2m = O(n^{1+\delta})$.

Next, we extend and generalize the approach of [4] and show how to $(1-\varepsilon)$-approximate the optimal cut, for any constant $\varepsilon > 0$, in time $2^{O(n \ln n/(\Delta\varepsilon^3))}$ (see Theorem 4). The running time is subexponential in n, if G is δ-almost sparse.

3.1 Outline and Main Ideas

Applying Proposition 1, we can write the smooth polynomial $p(\vec{x})$ as

$$p(\vec{x}) = \sum_{j \in V} x_j (\deg(j) - p_j(\vec{x}))\,, \tag{1}$$

where $p_j(\vec{x}) = \sum_{i \in N(j)} x_i$ is a degree-1 1-smooth polynomial that indicates how many neighbors of vertex j are in S_1 in the solution corresponding to $\vec{x}$. The key observation, due to [4], is that if we have a good estimation ρ_j of the value of each p_j at the optimal solution $\vec{x}^*$, then approximate maximization of $p(\vec{x})$ can be reduced to the solution of the following Integer Linear Program:

$$\begin{aligned} & \max \sum_{j \in V} y_j (\deg(j) - \rho_j) && \text{(IP)} \\ \text{s.t.} \quad & (1-\epsilon_1)\rho_j - \epsilon_2 \Delta \le \sum_{i \in N(j)} y_i \le (1+\epsilon_1)\rho_j + \epsilon_2 \Delta && \forall j \in V \\ & y_j \in \{0, 1\} && \forall j \in V \end{aligned}$$

The constants $\epsilon_1, \epsilon_2 > 0$ and the estimations $\rho_j \ge 0$ are computed so that the optimal solution $\vec{x}^*$ is a feasible solution to (IP). We always assume wlog. that $0 \le \sum_{i \in N(j)} y_i \le \deg(j)$, i.e., we let the lhs of the j-th constraint be $\max\{(1-\epsilon_1)\rho_j - \epsilon_2\Delta, 0\}$ and the rhs be $\min\{(1+\epsilon_1)\rho_j + \epsilon_2\Delta, \deg(j)\}$. Clearly, if $\vec{x}^*$ is a feasible solution to (IP), it remains a feasible solution after this modification. We let (LP) denote the Linear Programming relaxation of (IP), where each $y_j \in [0, 1]$.

The first important observation is that for any $\epsilon_1, \epsilon_2 > 0$, we can compute estimations ρ_j, by exhaustive sampling, so that $\vec{x}^*$ is a feasible solution to (IP) with high probability (see Lemma 1). The second important observation is that the objective value of any feasible solution $\vec{y}$ to (LP) is close to $p(\vec{y})$ (see Lemma 2). Namely, for any feasible solution $\vec{y}$, $\sum_{j \in V} y_j(\deg(j) - \rho_j) \approx p(\vec{y})$.

Based on these observations, the approximation algorithm performs the following steps:

1. We guess a sequence of estimations $\rho_1, \dots, \rho_n$, by exhaustive sampling, so that $\vec{x}^*$ is a feasible solution to the resulting (IP) (see Section 3.2 for the details).
2. We formulate (IP) and find an optimal fractional solution $\vec{y}^*$ to (LP).
3. We obtain an integral solution $\vec{z}$ by applying randomized rounding to $\vec{y}^*$ (and the method of conditional probabilities, as in [28, 27]).

To see that this procedure indeed provides a good approximation to $p(\vec{x}^*)$, we observe that:

$$p(\vec{z}) \approx \sum_{j \in V} z_j(\deg(j) - \rho_j) \approx \sum_{j \in V} y^*_j(\deg(j) - \rho_j) \geq \sum_{j \in V} x^*_j(\deg(j) - \rho_j) \approx p(\vec{x}^*)\,, \qquad (2)$$

The first approximation holds because $\vec{z}$ is an (almost) feasible solution to (IP) (see Lemma 3), the second approximation holds because the objective value of $\vec{z}$ is a good approximation to the objective value of $\vec{y}^*$, due to randomized rounding, the inequality holds because $\vec{x}^*$ is a feasible solution to (LP) and the final approximation holds because $\vec{x}^*$ is a feasible solution to (IP).

In Sections 3.3 and 3.4, we make the notion of approximation precise so that $p(\vec{z}) \geq (1-\varepsilon)p(\vec{x}^*)$. As for the running time, it is dominated by the time required for the exhaustive-sampling step. Since we do not know $\vec{x}^*$, we need to run the steps (2) and (3) above for every sequence of estimations produced by exhaustive sampling. So, the outcome of the approximation scheme is the best of the integral solutions $\vec{z}$ produced in step (3) over all executions of the algorithm. In Section 3.2, we show that a sample of size $O(n \ln n/\Delta)$ suffices for the computation of estimations ρ_j so that $\vec{x}^*$ is a feasible solution to (IP) with high probability. If G is δ-almost sparse, the sample size is sublinear in n and the running time is subexponential in n.

3.2 Obtaining Estimations ρ_j by Exhaustive Sampling

To obtain good estimations ρ_j of the values $p_j(\vec{x}^*) = \sum_{i \in N(j)} x^*_i$, i.e., of the number of j's neighbors in S_1 in the optimal cut, we take a random sample $R \subseteq V$ of size $\Theta(n \ln n/\Delta)$ and try exhaustively all possible assignments of the vertices in R to S_0 and S_1. If $\Delta = \Omega(n^\delta)$, we have $2^{O(n \ln n/\Delta)} = 2^{O(n^{1-\delta} \ln n)}$ different assignments. For each assignment, described by a 0/1 vector $\vec{x}$ restricted to R, we compute an estimation $\rho_j = (n/|R|) \sum_{i \in N(j) \cap R} x_i$, for each vertex $j \in V$, and run the steps (2) and (3) of the algorithm above. Since we try all possible assignments, one of them agrees with $\vec{x}^*$ on all vertices of R. So, for this assignment, the estimations computed are $\rho_j = (n/|R|) \sum_{i \in N(j) \cap R} x^*_i$. The following shows that for these estimations, we have that $p_j(\vec{x}^*) \approx \rho_j$ with high probability.

▶ **Lemma 1.** *Let $\vec{x}$ be any binary vector. For all $\alpha_1, \alpha_2 > 0$, we let $\gamma = \Theta(1/(\alpha_1^2\alpha_2))$ and let R be a multiset of $r = \gamma n \ln n/\Delta$ vertices chosen uniformly at random with replacement from V. For any vertex j, if $\rho_j = (n/r) \sum_{i \in N(j) \cap R} x_i$ and $\hat{\rho}_j = \sum_{i \in N(j)} x_i$, with probability at least $1 - 2/n^3$,*

$$(1-\alpha_1)\hat{\rho}_j - (1-\alpha_1)\alpha_2\Delta \leq \rho_j \leq (1+\alpha_1)\hat{\rho}_j + (1+\alpha_1)\alpha_2\Delta \qquad (3)$$

We note that $\rho_j \geq 0$ and always assume that $\rho_j \leq \deg(j)$, since if ρ_j satisfies (3), $\min\{\rho_j, \deg(j)\}$ also satisfies (3). For all $\epsilon_1, \epsilon_2 > 0$, setting $\alpha_1 = \frac{\epsilon_1}{1+\epsilon_1}$ and $\alpha_2 = \epsilon_2$ in Lemma 1, and taking the union bound over all vertices, we obtain that for $\gamma = \Theta(1/(\epsilon_1^2\epsilon_2))$, with probability at least $1 - 2/n^2$, the following holds for all vertices $j \in V$:

$$(1-\epsilon_1)\rho_j - \epsilon_2\Delta \leq \hat{\rho}_j \leq (1+\epsilon_1)\rho_j + \epsilon_2\Delta \qquad (4)$$

Therefore, with probability at least $1 - 2/n^2$, the optimal cut $\vec{x}^*$ is a feasible solution to (IP) with the estimations ρ_j obtained by restricting $\vec{x}^*$ to the vertices in R.

3.3 The Cut Value of Feasible Solutions

We next show that the objective value of any feasible solution $\vec{y}$ to (LP) is close to $p(\vec{y})$. Therefore, assuming that $\vec{x}^*$ is feasible, any good approximation to (IP) is a good approximation to the optimal cut.

▶ **Lemma 2.** *Let $\rho_1, \dots, \rho_n$ be non-negative numbers and $\vec{y}$ be any feasible solution to (LP). Then,*

$$p(\vec{y}) \in \sum_{j \in V} y_j(\deg(j) - \rho_j) \pm 2(\epsilon_1 + \epsilon_2)m \tag{5}$$

Proof. Using (1) and the formulation of (LP), we obtain that:

$$\begin{aligned} p(\vec{y}) = \sum_{j \in V} y_j \left(\deg(j) - \sum_{i \in N(j)} y_i \right) &\in \sum_{j \in V} y_j \left(\deg(j) - ((1 \mp \epsilon_1)\rho_j \mp \epsilon_2 \Delta)\right) \\ &= \sum_{j \in V} y_j(\deg(j) - \rho_j) \pm \epsilon_1 \sum_{j \in V} y_j \rho_j \pm \epsilon_2 \Delta \sum_{j \in V} y_j \\ &\in \sum_{j \in V} y_j(\deg(j) - \rho_j) \pm 2(\epsilon_1 + \epsilon_2)m \end{aligned}$$

The first inclusion holds because $\vec{y}$ is feasible for (LP) and thus, $\sum_{i \in N(j)} y_i \in (1 \pm \epsilon_1)\rho_j \pm \epsilon_2 \Delta$, for all j. The third inclusion holds because

$$\sum_{j \in V} y_j \rho_j \le \sum_{j \in V} \rho_j \le \sum_{j \in V} \deg(j) = 2m \,,$$

since each ρ_j is at most $\deg(j)$, and because $\Delta \sum_{j \in V} y_j \le \Delta n = 2m$. ◀

3.4 Randomized Rounding of the Fractional Optimum

As a last step, we show how to round the fractional optimum $\vec{y}^* = (y_1^*, \dots, y_n^*)$ of (LP) to an integral solution $\vec{z} = (z_1, \dots, z_n)$ that almost satisfies the constraints of (IP).

To this end, we use randomized rounding, as in [28]. In particular, we set independently each z_j to 1, with probability y_j^*, and to 0, with probability $1 - y_j^*$. By Chernoff bounds[1], we obtain that with probability at least $1 - 2/n^8$, for each vertex j,

$$(1 - \epsilon_1)\rho_j - \epsilon_2 \Delta - 2\sqrt{\deg(j)\ln(n)} \le \sum_{i \in N(j)} z_i \le (1 + \epsilon_1)\rho_j + \epsilon_2 \Delta + 2\sqrt{\deg(j)\ln(n)} \tag{6}$$

Specifically, the inequality above follows from the Chernoff bound in footnote 1, with $k = \deg(j)$ and $t = 2\sqrt{\deg(j)\ln(n)}$, since $\mathbb{E}[\sum_{i \in N(j)} z_j] = \sum_{i \in N(j)} y_j^* \in (1 \pm \epsilon_1)\rho_j \pm \epsilon_2 \Delta$. By the union bound, (6) is satisfied with probability at least $1 - 2/n^7$ for all vertices j.

By linearity of expectation, $\mathbb{E}[\sum_{j \in V} z_j(\deg(j) - \rho_j)] = \sum_{j \in V} y_j^*(\deg(j) - \rho_j)$. Moreover, since the probability that $\vec{z}$ does not satisfy (6) for some vertex j is at most $2/n^7$ and since the objective value of (IP) is at most n^2, the expected value of a rounded solution $\vec{z}$ that satisfies (6) for all vertices j is least $\sum_{j \in V} y_j^*(\deg(j) - \rho_j) - 1$ (assuming that $n \ge 2$). Using the method of conditional expectations, as in [27], we can find in (deterministic) polynomial time an integral solution $\vec{z}$ that satisfies (6) for all vertices j and has $\sum_{j \in V} z_j(\deg(j) - \rho_j) \ge \sum_{j \in V} y_j^*(\deg(j) - \rho_j) - 1$. Next, we sometimes abuse the notation and refer to such an integral solution $\vec{z}$ (computed deterministically) as the integral solution obtained from $\vec{y}^*$ by randomized rounding.

The following is similar to Lemma 2 and shows that the objective value $p(\vec{z})$ of the rounded solution $\vec{z}$ is close to the optimal value of (LP).

[1] We use the following standard Chernoff bound (see e.g., [19, Theorem 1.1]): Let $Y_1, \dots, Y_k$ independent random variables in $[0, 1]$ and let $Y = \sum_{j=1}^{k} Y_j$. Then for all $t > 0$, $\mathbb{P}[|Y - \mathbb{E}[Y]| > t] \le 2\exp(-2t^2/k)$.

▶ **Lemma 3.** *Let $\vec{y}^*$ be the optimal solution of (LP) and let $\vec{z}$ be the integral solution obtained from $\vec{y}^*$ by randomized rounding (and the method of conditional expectations). Then,*

$$p(\vec{z}) \in \sum_{j \in V} y_j^* (\deg(j) - \rho_j) \pm 3(\epsilon_1 + \epsilon_2) m \tag{7}$$

3.5 Putting Everything Together

Therefore, for any $\varepsilon > 0$, if G is δ-almost sparse and $\Delta = n^\delta$, the algorithm described in Section 3.1, with sample size $\Theta(n \ln n / (\varepsilon^3 \Delta))$, computes estimations ρ_j such that the optimal cut $\vec{x}^*$ is a feasible solution to (IP) whp. Hence, by the analysis above, the algorithm approximates the value of the optimal cut $p(\vec{x}^*)$ within an additive term of $O(\varepsilon m)$. Specifically, setting $\epsilon_1 = \epsilon_2 = \varepsilon/16$, the value of the cut $\vec{z}$ produced by the algorithm satisfies the following with probability at least $1 - 2/n^2$:

$$p(\vec{z}) \geq \sum_{j \in V} y_j^* (\deg(j) - \rho_j) - 3\varepsilon m/8 \geq \sum_{j \in V} x_j^* (\deg(j) - \rho_j) - 3\varepsilon m/8 \geq p(\vec{x}^*) - \varepsilon m/2 \geq (1-\varepsilon) p(\vec{x}^*)$$

The first inequality follows from Lemma 3, the second inequality holds because $\vec{y}^*$ is the optimal solution to (LP) and $\vec{x}^*$ is feasible for (LP), the third inequality follows from Lemma 2 and the fourth inequality holds because the optimal cut has at least $m/2$ edges.

▶ **Theorem 4.** *Let $G(V, E)$ be a δ-almost sparse graph with n vertices. Then, for any $\varepsilon > 0$, we can compute, in time $2^{O(n^{1-\delta} \ln n / \varepsilon^3)}$ and with probability at least $1 - 2/n^2$, a cut $\vec{z}$ of G with value $p(\vec{z}) \geq (1 - \varepsilon) p(\vec{x}^*)$, where $\vec{x}^*$ is the optimal cut.*

4 Approximate Maximization of Smooth Polynomials

Generalizing the ideas applied to Max-CUT, we arrive at the main algorithmic result of the paper: an algorithm to approximately optimize β-smooth δ-bounded polynomials $p(\vec{x})$ of degree d over all binary vectors $\vec{x} \in \{0, 1\}^n$. The intuition and the main ideas are quite similar to those in Section 3, but the details are significantly more involved because we are forced to recursively decompose degree d polynomials to eventually obtain a linear program. In the Appendix, Section B, we take care of the technical details and prove the following:

▶ **Theorem 5.** *Let $p(\vec{x})$ be an n-variate degree-d β-smooth δ-bounded polynomial. Then, for any $\varepsilon > 0$, we can compute, in time $2^{O(d^7 \beta^3 n^{1-\delta} \ln n / \varepsilon^3)}$ and with probability at least $1 - 8/n^2$, a binary vector $\vec{z}$ so that $p(\vec{z}) \geq p(\vec{x}^*) - \varepsilon n^{d-1+\delta}$, where $\vec{x}^*$ is the maximizer of $p(\vec{x})$.*

Max-k-CSP: Using Theorem 5 it is a straightforward observation that for any Max-k-CSP problem (for constant k) we can obtain an algorithm which, given a Max-k-CSP instance with $\Omega(n^{k-1+\delta})$ constraints for some $\delta > 0$, for any $\varepsilon > 0$ returns an assignment that satisfies $(1 - \varepsilon)$OPT constraints in time $2^{O(n^{1-\delta} \ln n / \varepsilon^3)}$. This follows from Theorem 5 using two observations: first, the standard arithmetization of Max-k-CSP described in Section 2 produces a degree-k β-smooth δ-bounded polynomial for β depending only on k. Second, the optimal solution of such an instance satisfies at least $\Omega(n^{k-1+\delta})$ constraints, therefore the additive error given in Theorem 5 is $O(\varepsilon\text{OPT})$. This algorithm for Max-k-CSP contains as special cases algorithm for various standard problems such as Max-CUT, Max-DICUT and Max-k-SAT.

5 Approximating the k-Densest Subgraph in Almost Sparse Graphs

In this section, we present an extension of the algorithms we have presented which can be used to approximate k-Densest Subgraph in δ-almost sparse graphs. This is a problem also handled in [4], but only for the case where $k = \Omega(n)$. Smaller values of k cannot be handled by the scheme of [4] for dense graphs because when $k = o(n)$ the optimal solution has objective value much smaller than the additive error of εn^2 inherent in their scheme.

Here we obtain a sub-exponential time approximation scheme that works on graphs with $\Omega(n^{1+\delta})$ edges *for all* k by judiciously combining two approaches: when k is relatively large, we use a sampling approach similar to Max-CUT; when k is small, we can resort to the naïve algorithm that tries all $\binom{n}{k}$ possible solutions. We select (with some foresight) the threshold between the two algorithms to be $k = \Omega(n^{1-\delta/3})$, so that in the end we obtain an approximation scheme with running time of $2^{O(n^{1-\delta/3} \ln n)}$, that is, slightly slower than the approximation scheme for Max-CUT. It is clear that the brute-force algorithm achieves this running time for $k = O(n^{1-\delta/3})$, so in the remainder we focus on the case of large k.

The k-Densest Subgraph problem in a graph $G(V, E)$ is equivalent to maximizing, over all vectors $\vec{x} \in \{0,1\}^n$, the n-variate degree-2 1-smooth polynomial $p(\vec{x}) = \sum_{\{i,j\}\in E} x_i x_j$, under the linear constraint $\sum_{j\in V} x_j = k$. Setting a variable x_i to 1 indicates that the vertex i is included in the set C that induces a dense subgraph $G[C]$ of k vertices. We assume that G is δ-almost sparse i.e. $m = \Omega(n^{1+\delta})$ edges. As usual, $\vec{x}$ denotes the optimal solution.

The algorithm follows the same general approach and the same basic steps as the algorithm for Max-CUT in Section 3. In the following, we highlight only the differences.

Obtaining Estimations by Exhaustive Sampling. We first observe that if G is δ-almost sparse and $k = \Omega(n^{1-\delta/3})$, a random subset of k vertices contains $\Omega(n^{1+\delta/3})$ edges in expectation. We thus assume that the optimal solution induces at least $\Omega(n^{1+\delta/3})$ edges.

Working as in Section 3.2, we use exhaustive sampling and obtain for each vertex $j \in V$, an estimation ρ_j of j's neighbors in the optimal dense subgraph, i.e., ρ_j is an estimation of $\hat{\rho}_j = \sum_{i\in N(j)} x_i^*$. For the analysis, we apply Lemma 1 with $n^{\delta/3}$, instead of Δ, or in other words, we use a sample of size $\Theta(n^{1-\delta/3} \ln n)$. The reason is that we can only tolerate an additive error of $\varepsilon n^{1+\delta/3}$, by the lower bound on the optimal solution observed in the previous paragraph. Then, the running time due to exhaustive sampling is $2^{O(n^{1-\delta/3} \ln n)}$.

By Lemma 1 and the discussion following it in Section 3.2, we obtain that for all $\epsilon_1, \epsilon_2 > 0$, if we use a sample of the size $\Theta(n^{1-\delta/3} \ln n/(\epsilon_1^2 \epsilon_2))$, with probability at least $1 - 2/n^2$, the following holds for all estimations ρ_j and all vertices $j \in V$:

$$(1-\epsilon_1)\rho_j - \epsilon_2 n^{\delta/3} \le \hat{\rho}_j \le (1+\epsilon_1)\rho_j + \epsilon_2 n^{\delta/3} \tag{8}$$

Linearizing the Polynomial. Applying Proposition 1, we can write the polynomial $p(\vec{x})$ as $p(\vec{x}) = \sum_{j\in V} x_j p_j(\vec{x})$, where $p_j(\vec{x}) = \sum_{i\in N(j)} x_i$ is a degree-1 1-smooth polynomial that indicates how many neighbors of vertex j are in C in the solution corresponding to $\vec{x}$. Then, using the estimations ρ_j of $\sum_{i\in N(j)} x_i^*$, obtained by exhaustive sampling, we have that approximate maximization of $p(\vec{x})$ can be reduced to the solution of the following ILP:

$$\max \sum_{j\in V} y_j \rho_j \tag{IP$'$}$$

$$\text{s.t.} \quad (1-\epsilon_1)\rho_j - \epsilon_2 n^{\delta/3} \le \sum_{i\in N(j)} y_i \le (1+\epsilon_1)\rho_j + \epsilon_2 n^{\delta/3} \quad \forall j \in V$$

$$\sum_{i\in V} y_i = k$$

By (8), if the sample size is $|R| = \Theta(n^{1-\delta/3} \ln n/(\epsilon_1^2 \epsilon_2))$, with probability at least $1 - 2/n^2$, the densest subgraph $\vec{x}^*$ is a feasible solution to (IP′) with the estimations ρ_j obtained by restricting $\vec{x}^*$ to the vertices in R. In the following, we let (LP′) denote the Linear Programming relaxation of (IP′), where each $y_j \in [0, 1]$.

The Number of Edges in Feasible Solutions. We next show that the objective value of any feasible solution $\vec{y}$ to (LP′) is close to $p(\vec{y})$. Therefore, assuming that $\vec{x}^*$ is feasible, any good approximation to (IP′) is a good approximation to the densest subgraph.

▶ **Lemma 6.** *Let $\rho_1, \ldots, \rho_n$ be non-negative numbers and $\vec{y}$ be any feasible solution to (LP′). Then,*

$$p(\vec{y}) \in (1 \pm \epsilon_1) \sum_{j \in V} y_j \rho_j \pm \epsilon_2 n^{1+\delta/3} \tag{9}$$

Randomized Rounding of the Fractional Optimum. As a last step, we show how to round the fractional optimum $\vec{y}^* = (y_1^*, \ldots, y_n^*)$ of (LP′) to an integral solution $\vec{z} = (z_1, \ldots, z_n)$ that almost satisfies the constraints of (IP′). We use randomized rounding, as for Max-CUT.

▶ **Lemma 7.** *Let $\vec{y}^*$ be the optimal solution of (LP′) and let $\vec{z}$ be the integral solution obtained from $\vec{y}^*$ by randomized rounding (and the method of conditional expectations). Then,*

$$p(\vec{z}) \in (1 \pm \epsilon_1)^2 \sum_{j \in V} y_j^* \rho_j \pm 3\epsilon_2 n^{1+\delta/3} \tag{10}$$

We thus arrive to the main theorem of this section.

▶ **Theorem 8.** *Let $G(V, E)$ be a δ-almost sparse graph with n vertices. Then, for any integer $k \geq 1$ and for any $\varepsilon > 0$, we can compute, in time $2^{O(n^{1-\delta/3} \ln n/\varepsilon^3)}$ and with probability at least $1 - 2/n^2$, an induced subgraph $\vec{z}$ of G with k vertices whose number of edges satisfies $p(\vec{z}) \geq (1 - \varepsilon)p(\vec{x}^*)$, where $\vec{x}^*$ is the number of edges in the k-Densest Subgraph of G.*

6 Lower Bounds

We now give some lower bound arguments showing that the schemes we have presented are, in some senses, likely to be almost optimal. Our complexity assumption will be the ETH, which states that no algorithm can solve instances of 3-SAT of size n in time $2^{o(n)}$.

There are two natural ways in which one may hope to improve or extend the algorithms we have presented so far: relaxing the density requirement or decreasing the running time. First, recall that the algorithm we have given for Max-k-CSP works in the density range between n^k and n^{k-1}. Here, we give a reduction establishing that it's unlikely that this can be improved. Our starting point is the following (known) inapproximability result.

▶ **Theorem 9.** *There exist $c, s \in (0, 1)$ with $c > s$ such that for all $\epsilon > 0$ we have: if there exists an algorithm which, given an n-vertex 5-regular instance of* Max-*CUT, can distinguish between the case where a solution cuts at least a c fraction of the edges and the case where all solutions cut at most an s fraction of the edges in time $2^{n^{1-\epsilon}}$ then the ETH fails.*

▶ **Theorem 10.** *There exists $r > 1$ such that for all $\epsilon > 0$ and all (fixed) integers $k \geq 3$ we have the following: if there exists an algorithm which r-approximates* Max-k-SAT *on instances with $\Omega(n^{k-1})$ clauses in time $2^{n^{1-\epsilon}}$ then the ETH fails.*

Proof. We reduce a MAX-CUT instance from Theorem 9 to MAX-2-SAT: the set of variables is the set of vertices; for each edge (u, v) we include the clauses $(u \vee v)$ and $(\neg u \vee \neg v)$. The new instance has n variables and $5n$ clauses and there exist constants c, s such that either some assignment satisfies $5cn$ clauses or all assignments satisfy at most $5sn$ of them.

Fix k and add to the instance $(k-2)n$ new variables $x_{(i,j)}$, $i \in \{1, \ldots, k-2\}$, $j \in \{1, \ldots, n\}$. We perform the following transformation: for each clause $(l_1 \vee l_2)$ and for each tuple $(i_1, i_2, \ldots, i_{k-2}) \in \{1, \ldots, n\}^{k-2}$ we construct 2^{k-2} new clauses of size k. The first two literals of these clauses are l_1, l_2. The rest consist of the variables $x_{(1,i_1)}, x_{(2,i_2)}, \ldots, x_{(k,i_{k-2})}$, but in each clause a different set of variables is negated. In other words, to construct a clause of the new instance we select a clause of the original instance, one variable from each of the groups of n new variables, and a subset of these variables to be negated.

First, observe that the new instance has $5n^{k-1}2^k$ clauses and $(k-1)n$ variables, which satisfies the density conditions. Consider an assignment of the original formula. Any satisfied clause has now been replaced by $n^{k-2}2^k$ satisfied clauses, while for an unsatisfied clause any assignment to the new variables satisfies exactly $n^{k-2}(2^k - 1)$ clauses. Thus, for fixed k, there exist constants s', c' such that either a c' fraction of the clauses of the new instance is satisfiable or at most a s' fraction is. If we had an approximation algorithm with ratio better than c'/s' running in time $2^{N^{1-\epsilon}}$, where N is the number of variables of the new instance, we could use it to decide the original instance in time that would disprove the ETH. ◀

A second possible avenue for improvement may be to consider potential speedups of our algorithms. We give an almost tight answer to such questions via the following theorem.

▶ **Theorem 11.** *There exists $r > 1$ such that for all $\epsilon > 0$ we have the following: if there exists an algorithm which, for some $\Delta = o(n)$, r-approximates* MAX-*CUT on n-vertex Δ-regular graphs in time $2^{(n/\Delta)^{1-\epsilon}}$ then the ETH fails.*

Proof. Without loss of generality we prove the theorem for the case when the degree is a multiple of 10. Consider an instance $G(V, E)$ of MAX-CUT as given by Theorem 9. Let $n = |V|$ and suppose that the desired degree is $d = 10\Delta$, where Δ is a function of n. We construct a graph G' as follows: for each vertex $u \in V$ we introduce Δ new vertices $u_1, \ldots, u_\Delta$ as well as 5Δ "consistency" vertices $c^u_1, \ldots, c^u_{5\Delta}$. For every edge $(u, v) \in E$ we add all edges (u_i, v_j) for $i, j \in \{1, \ldots, \Delta\}$. Also, for every $u \in V$ we add all edges (u_i, c^u_j), for $i \in \{1, \ldots, \Delta\}$ and $j \in \{1, \ldots, 5\Delta\}$. This completes the construction.

The graph we have constructed is 10Δ-regular and is made up of $6\Delta n$ vertices. Consider an optimal cut and observe that, for a given $u \in V$ all the vertices c^u_i can be assumed to be on the same side of the cut, since they all have the same neighbors. Furthermore, for a given $u \in V$, all vertices u_i can be assumed to be on the same side of the cut, namely on the side opposite that of c^u_i, since the vertices c^u_i are a majority of the neighborhood of each u_i. With this observation it is easy to construct a one-to-one correspondence between cuts in G and locally optimal cuts in G'.

Consider a cut that cuts $c|E|$ edges of G. If we set all u_i of G' on the same side as u is in G we cut $c|E|\Delta^2$ edges of the form (u_i, v_j). Furthermore, by placing the c^u_i on the opposite side of u_i we cut $5\Delta^2|V|$ edges. Thus the max cut of G' is at least $c|E|\Delta^2 + 5\Delta^2|V|$. Using the observations on locally optimal cuts of G' we can conclude that if G' has a cut with $s|E|\Delta^2 + 5\Delta^2|V|$ edges, then G has a cut with $s|E|$ edges. Having $2|E| = 5|V|$ (since G is 5-regular) we get a constant ratio r between the size of the cut of G' in the two cases.

Suppose now that we have an approximation algorithm with ratio better than r which, given an N-vertex d-regular graph runs in time $2^{(N/d)^{1-\epsilon}}$. Giving our constructed instance as input to this algorithm would allow to decide the original instance in time $2^{n^{1-\epsilon}}$. ◀

References

1 Nir Ailon and Noga Alon. Hardness of fully dense problems. *Inf. Comput.*, 205(8):1117–1129, 2007.

2 Noga Alon, Wenceslas Fernandez de la Vega, Ravi Kannan, and Marek Karpinski. Random sampling and approximation of max-csps. *J. Comput. Syst. Sci.*, 67(2):212–243, 2003.

3 Sanjeev Arora, Alan Frieze, and Haim Kaplan. A new rounding procedure for the assignment problem with applications to dense graph arrangement problems. *Mathematical Programming, Series A*, 92:1–36, 2002.

4 Sanjeev Arora, David Karger, and Marek Karpinski. Polynomial time approximation schemes for dense instances of NP-hard problems. *Journal of Computer and System Sciences*, 58:193–210, 1999.

5 Cristina Bazgan, Wenceslas Fernandez de la Vega, and Marek Karpinski. Polynomial time approximation schemes for dense instances of minimum constraint satisfaction. *Random Struct. Algorithms*, 23(1):73–91, 2003.

6 Piotr Berman and Marek Karpinski. On some tighter inapproximability results (extended abstract). In Jirí Wiedermann, Peter van Emde Boas, and Mogens Nielsen, editors, *Automata, Languages and Programming, 26th International Colloquium, ICALP'99, Prague, Czech Republic, July 11-15, 1999, Proceedings*, volume 1644 of *Lecture Notes in Computer Science*, pages 200–209. Springer, 1999.

7 Nicolas Bourgeois, Federico Della Croce, Bruno Escoffier, and Vangelis Th. Paschos. Fast algorithms for min independent dominating set. *Discrete Applied Mathematics*, 161(4-5):558–572, 2013.

8 Nicolas Bourgeois, Bruno Escoffier, and Vangelis Th. Paschos. Approximation of max independent set, min vertex cover and related problems by moderately exponential algorithms. *Discrete Applied Mathematics*, 159(17):1954–1970, 2011.

9 Jean Cardinal, Marek Karpinski, Richard Schmied, and Claus Viehmann. Approximating subdense instances of covering problems. *Electronic Notes in Discrete Mathematics*, 37:297–302, 2011.

10 Jean Cardinal, Marek Karpinski, Richard Schmied, and Claus Viehmann. Approximating vertex cover in dense hypergraphs. *J. Discrete Algorithms*, 13:67–77, 2012.

11 Parinya Chalermsook, Bundit Laekhanukit, and Danupon Nanongkai. Independent set, induced matching, and pricing: Connections and tight (subexponential time) approximation hardnesses. In *54th Annual IEEE Symposium on Foundations of Computer Science, FOCS 2013, 26-29 October, 2013, Berkeley, CA, USA*, pages 370–379. IEEE Computer Society, 2013.

12 Marek Cygan, Lukasz Kowalik, and Mateusz Wykurz. Exponential-time approximation of weighted set cover. *Inf. Process. Lett.*, 109(16):957–961, 2009.

13 Marek Cygan and Marcin Pilipczuk. Exact and approximate bandwidth. *Theor. Comput. Sci.*, 411(40-42):3701–3713, 2010.

14 Marek Cygan, Marcin Pilipczuk, and Jakub Onufry Wojtaszczyk. Capacitated domination faster than $O(2^n)$. *Inf. Process. Lett.*, 111(23-24):1099–1103, 2011.

15 Wenceslas Fernandez de la Vega. MAX-CUT has a randomized approximation scheme in dense graphs. *Random Struct. Algorithms*, 8(3):187–198, 1996.

16 Wenceslas Fernandez de la Vega and Marek Karpinski. On the approximation hardness of dense TSP and other path problems. *Inf. Process. Lett.*, 70(2):53–55, 1999.

17 Wenceslas Fernandez de la Vega and Marek Karpinski. Polynomial time approximation of dense weighted instances of MAX-CUT. *Random Struct. Algorithms*, 16(4):314–332, 2000.

18 Irit Dinur. The PCP theorem by gap amplification. In *Proc. 38th ACM Symp. on Theory of Computing*, pages 241–250, 2006.

19 Devdatt P. Dubhashi and A. Panconesi. *Concentration of Measure for the Analysis of Randomized Algorithms*. Cambridge University Press, 2009.

20 Martin E. Dyer, Alan M. Frieze, and Mark Jerrum. Approximately counting hamilton paths and cycles in dense graphs. *SIAM J. Comput.*, 27(5):1262–1272, 1998.

21 Alan M. Frieze and Ravi Kannan. The regularity lemma and approximation schemes for dense problems. In *37th Annual Symposium on Foundations of Computer Science, FOCS '96, Burlington, Vermont, USA, 14-16 October, 1996*, pages 12–20. IEEE Computer Society, 1996.

22 Johan Hastad. Some optimal inapproximability results. *J. ACM*, 48(4):798–859, 2001.

23 Tomokazu Imamura and Kazuo Iwama. Approximating vertex cover on dense graphs. In *Proceedings of the Sixteenth Annual ACM-SIAM Symposium on Discrete Algorithms, SODA 2005, Vancouver, British Columbia, Canada, January 23-25, 2005*, pages 582–589. SIAM, 2005.

24 Sanjeev Khanna, Madhu Sudan, and David P. Williamson. A complete classification of the approximability of maximization problems derived from boolean constraint satisfaction. In Frank Thomson Leighton and Peter W. Shor, editors, *Proceedings of the Twenty-Ninth Annual ACM Symposium on the Theory of Computing, El Paso, Texas, USA, May 4-6, 1997*, pages 11–20. ACM, 1997.

25 Dana Moshkovitz and Ran Raz. Two-query PCP with subconstant error. *J. ACM*, 57(5), 2010.

26 Christos H. Papadimitriou and Mihalis Yannakakis. Optimization, approximation, and complexity classes. *J. Comput. Syst. Sci.*, 43(3):425–440, 1991.

27 Prabhakar Raghavan. Probabilistic construction of deterministic algorithms: Approximating packing integer programs. *Journal of Computer and System Sciences*, 37(2):130–143, 1988.

28 Prabhakar Raghavan and Clark D. Thompson. Randomized rounding: A technique for provably good algorithms and algorithmic proofs. *Combinatorica*, 7(4):365–474, 1987.

29 Thomas J. Schaefer. The complexity of satisfiability problems. In Richard J. Lipton, Walter A. Burkhard, Walter J. Savitch, Emily P. Friedman, and Alfred V. Aho, editors, *Proceedings of the 10th Annual ACM Symposium on Theory of Computing, May 1-3, 1978, San Diego, California, USA*, pages 216–226. ACM, 1978.

30 Luca Trevisan. Inapproximability of combinatorial optimization problems. *Electronic Colloquium on Computational Complexity (ECCC)*, 2004.

A Appendix

A.1 Proof of Proposition 1

The proposition is shown in [4, Lemma 3.1]. We prove it here just for completeness. Each polynomial $p_j(\vec{x})$ is obtained from $p(\vec{x})$ if we keep only the monomials with variable x_j and pull x_j out, as a common factor. The constant c takes care of the constant term in $p(\vec{x})$. Each monomial of degree ℓ in $p(\vec{x})$ becomes a monomial of degree $\ell - 1$ in $p_j(\vec{x})$, which implies that the degree of $p_j(\vec{x})$ is $d-1$. Moreover, by the β-smoothness condition, the coefficient t of each degree-ℓ monomial in $p(\vec{x})$ has $|t| \leq \beta n^{d-\ell}$. The corresponding monomial in $p_j(\vec{x})$ has degree $\ell-1$ and the same coefficient t with $|t| \leq \beta n^{d-1-(\ell-1)}$. Therefore, if $p(\vec{x})$ is β-smooth, each $p_j(\vec{x})$ is also β-smooth. ◀

A.2 Proof of Lemma 1

Sketch. If $\hat{\rho}_j = \Omega(\Delta)$, the neighbors of j are well-represented in the random sample R whp., because $|R| = \Theta(n \ln n/\Delta)$. Therefore, $|\hat{\rho}_j - \rho_j| \leq \alpha_1 \hat{\rho}_j$ whp., by Chernoff bounds. If $\hat{\rho}_j = o(\Delta)$, the lower bound in (3) becomes trivial, since it is non-positive, while $\rho_j \geq 0$. As for the upper bound, we increase some x_i to $x'_i \in [0,1]$, so that $\hat{\rho}'_j = \alpha_2 \Delta$. Then, $\rho'_j \leq (1+\alpha_1)\hat{\rho}'_j = (1+\alpha_1)\alpha_2\Delta$ whp., by the same Chernoff bound as above. Now the upper bound of (3) follows from $\rho_j \leq \rho'_j$, which holds for any instantiation of the random sample R. The formal proof follows from Lemma 12, with $\beta = 1$, $d = 2$ and $q = 0$, and with Δ instead of n^δ. ◀

A.3 The Proof of Lemma 3

Using (6) and an argument similar to that in the proof of Lemma 2, we obtain that:

$$\begin{aligned}
p(\vec{z}) &= \sum_{j \in V} z_j \left(\deg(j) - \sum_{i \in N(j)} z_i \right) \\
&\in \sum_{j \in V} z_j \left(\deg(j) - \left((1 \mp \epsilon_1)\rho_j \mp \epsilon_2 \Delta \mp 2\sqrt{\deg(j)\ln(n)} \right) \right) \\
&= \sum_{j \in V} z_j(\deg(j) - \rho_j) \pm \epsilon_1 \sum_{j \in V} z_j \rho_j \pm \epsilon_2 \Delta \sum_{j \in V} z_j \pm 2 \sum_{j \in V} z_j \sqrt{\deg(j)\ln(n)} \\
&\in \sum_{j \in V} z_j(\deg(j) - \rho_j) \pm (3\epsilon_1 + 2\epsilon_2)m \\
&\in \sum_{j \in V} y^*_j(\deg(j) - \rho_j) \pm 3(\epsilon_1 + \epsilon_2)m
\end{aligned}$$

The first inclusion holds because $\vec{z}$ satisfies (6) for all $j \in V$. For the third inclusion, we use that $\sum_{j \in V} z_j \rho_j \leq \sum_{j \in V} \deg(j) = 2m$, that $\Delta \sum_{i \in V} z_i \leq \Delta n = 2m$ and that by Jensen's inequality,

$$2 \sum_{j \in V} z_j \sqrt{\deg(j) \ln n} \leq \sum_{j \in V} \sqrt{4 \deg(j) \ln n} \leq \sqrt{8mn \ln n} \leq \epsilon_1 m \,,$$

assuming that n and $m = \Omega(n^{1+\delta})$ are sufficiently large. For the last inclusion, we recall that $\sum_{j \in V} z_j(\deg(j) - \rho_j) \geq \sum_{j \in V} y^*_j(\deg(j) - \rho_j) - 1$ and assume that m is sufficiently large. ◀

B Approximate Maximization of Polynomials: The Proof of Theorem 5

Next, we significantly generalize the ideas applied to MAX-CUT so that we approximately optimize β-smooth δ-bounded polynomials $p(\vec{x})$ of degree d over all binary vectors $\vec{x} \in \{0,1\}^n$. The structure of this section deliberately parallels the structure of Section 3, so that the application to MAX-CUT can always serve as a reference for the intuition behind the generalization.

As in [4] (and as explained in Section 2), we exploit the fact that any n-variate degree-d β-smooth polynomial $p(\vec{x})$ can be decomposed into n degree-$(d-1)$ β-smooth polynomials $p_j(\vec{x})$ such that $p(\vec{x}) = c + \sum_{j \in N} x_j p_j(\vec{x})$ (Proposition 1). For smooth polynomials of degree $d \geq 3$, we apply Proposition 1 recursively until we end up with smooth polynomials of degree 1. Specifically, using Proposition 1, we further decompose each degree-$(d-1)$ β-smooth polynomial $p_{i_1}(\vec{x})$ into n degree-$(d-2)$ β-smooth polynomials $p_{i_1 j}(\vec{x})$ such that $p_{i_1}(\vec{x}) = c_{i_1} + \sum_{j \in N} x_j p_{i_1 j}(\vec{x})$, etc. At the basis of the recursion, at depth $d-1$, we have β-smooth polynomials $p_{i_1 \dots i_{d-1}}(\vec{x})$ of degree 1, one for each $(d-1)$-tuple of indices $(i_1, \dots, i_{d-1}) \in N^{d-1}$. These polynomials are written as

$$p_{i_1 \dots i_{d-1}}(\vec{x}) = c_{i_1 \dots i_{d-1}} + \sum_{j \in N} x_j c_{i_1 \dots i_{d-1} j} \,,$$

where $c_{i_1 \dots i_{d-1} j}$ are constants (these are the coefficients of the corresponding degree-d monomials in the expansion of $p(\vec{x})$). Due to β-smoothness, $|c_{i_1 \dots i_{d-1} j}| \leq \beta$ and $|c_{i_1 \dots i_{d-1}}| \leq \beta n$. Inductively, β-smoothness implies that each polynomial $p_{i_1 \dots i_{d-\ell}}(\vec{x})$ of degree $\ell \geq 1$ in this decomposition[2] has $|p_{i_1 \dots i_{d-\ell}}(\vec{x})| \leq (\ell+1)\beta n^\ell$ for all binary vectors $\vec{x} \in \{0,1\}^n$. Such a decomposition of $p(\vec{x})$ in β-smooth polynomials of degree $d-1, d-2, \dots, 1$ can be computed recursively in time $O(n^d)$.

B.1 Outline and General Approach

As in Section 3 (and as in [4]), we observe that if we have good estimations $\rho_{i_1 \dots i_{d-\ell}}$ of the values of each degree-ℓ polynomial $p_{i_1 \dots i_{d-\ell}}(\vec{x})$ at the optimal solution $\vec{x}^*$, for each level $\ell = 1, \dots, d-1$ of the decomposition, then approximate maximization of $p(\vec{x})$ can be reduced

[2] This decomposition can be performed in a unique way if we insist that $i_1 < i_2 < \cdots < i_{d-1}$, but this is not important for our analysis.

to the solution of the following Integer Linear Program:

$$\max \sum_{j \in N} y_j \rho_j \tag{d-IP}$$

$$\text{s.t.} \quad c_{i_1} + \sum_{j \in N} y_j \rho_{i_1 j} \in \rho_{i_1} \pm \epsilon_1 \bar{\rho}_{i_1} \pm \epsilon_2 n^{d-1+\delta} \qquad \forall i_1 \in N$$

$$c_{i_1 i_2} + \sum_{j \in N} y_j \rho_{i_1 i_2 j} \in \rho_{i_1 i_2} \pm \epsilon_1 \bar{\rho}_{i_1 i_2} \pm \epsilon_2 n^{d-2+\delta} \qquad \forall (i_1, i_2) \in N \times N$$

$$\cdots$$

$$c_{i_1 \dots i_{d-\ell}} + \sum_{j \in N} y_j \rho_{i_1 \dots i_{d-\ell} j} \in \rho_{i_1 \dots i_{d-\ell}} \pm \epsilon_1 \bar{\rho}_{i_1 \dots i_{d-\ell}} \pm \epsilon_2 n^{d-\ell+\delta} \quad \forall (i_1, \dots, i_{d-\ell}) \in N^{d-\ell}$$

$$\cdots$$

$$c_{i_1 \dots i_{d-1}} + \sum_{j \in N} y_j c_{i_1 \dots i_{d-1} j} \in \rho_{i_1 \dots i_{d-1}} \pm \epsilon_1 \bar{\rho}_{i_1 \dots i_{d-1}} \pm \epsilon_2 n^{\delta} \qquad \forall (i_1, \dots, i_{d-1}) \in N^{d-1}$$

$$y_j \in \{0, 1\} \qquad \forall j \in N$$

In (d-IP), we also use *absolute value estimations* $\bar{\rho}_{i_1 \dots i_{d-\ell}}$. For each level $\ell \geq 1$ of the decomposition of $p(\vec{x})$ and each tuple $(i_1, \dots, i_{d-\ell}) \in N^{d-\ell}$, we define the corresponding absolute value estimation as $\bar{\rho}_{i_1 \dots i_{d-\ell}} = \sum_{j \in N} |\rho_{i_1 \dots i_{d-\ell} j}|$. Namely, each absolute value estimation $\bar{\rho}_{i_1 \dots i_{d-\ell}}$ at level ℓ is the sum of the absolute values of the estimations $\rho_{i_1 \dots i_{d-\ell} j}$ at level $\ell - 1$. The reason that we use absolute value estimations and set the lhs/rhs of the constraints to $\rho_{i_1 \dots i_{d-\ell}} \pm \epsilon_1 \bar{\rho}_{i_1 \dots i_{d-\ell}}$, instead of simply to $(1 \pm \epsilon_1)\rho_{i_1 \dots i_{d-\ell}}$, is that we want to consider linear combinations of positive and negative estimations $\rho_{i_1 \dots i_{d-\ell}}$ in a uniform way.

Similarly to Section 3, the estimations $\rho_{i_1 \dots i_{d-\ell}}$ (and $\bar{\rho}_{i_1 \dots i_{d-\ell}}$) are computed (by exhaustive sampling) and the constants $\epsilon_1, \epsilon_2 > 0$ are calculated so that the optimal solution $\vec{x}^*$ is a feasible solution to (d-IP). In the following, we let $\vec{\rho}$ denote the sequence of estimations $\rho_{i_1 \dots i_{d-\ell}}$, for all levels ℓ and all tuples $(i_1, \dots, i_{d-\ell}) \in N^{d-\ell}$, that we use to formulate (d-IP). The absolute value estimations $\bar{\rho}_{i_1 \dots i_{d-\ell}}$ can be easily computed from $\vec{\rho}$. We let (d-LP) denote the Linear Programming relaxation of (d-IP), where each $y_j \in [0, 1]$, let $\vec{x}^*$ denote the binary vector that maximizes $p(\vec{x})$, and let $\vec{y}^* \in [0, 1]^n$ denote the fractional optimal solution of (d-LP).

As in Section 3, the approach is based on the facts that (i) for all constants $\epsilon_1, \epsilon_2 > 0$, we can compute estimations $\vec{\rho}$, by exhaustive sampling, so that $\vec{x}^*$ is a feasible solution to (d-IP) with high probability (see Lemma 12 and Lemma 13); and that (ii) the objective value of any feasible solution $\vec{y}$ to (d-LP) is close to $p(\vec{y})$ (see Lemma 14 and Lemma 15). Based on these observations, the general description of the approximation algorithm is essentially identical to the three steps described in Section 3.1 and the reasoning behind the approximation guarantee is that of (2).

B.2 Obtaining Estimations by Exhausting Sampling

We first show how to use exhaustive sampling and obtain an estimation $\rho_{i_1 \dots i_{d-\ell}}$ of the value at the optimal solution $\vec{x}^*$ of each degree-ℓ polynomial $p_{i_1 \dots i_{d-\ell}}(\vec{x})$ in the decomposition of $p(\vec{x})$.

As in Section 3.2, we take a sample R from N, uniformly at random and with replacement. The sample size is $r = \Theta(n^{1-\delta} \ln n)$. We try exhaustively all 0/1 assignments to the variables in R, which can performed in time $2^r = 2^{O(n^{1-\delta} \ln n)}$. For each assignment, described by a 0/1 vector $\vec{s}$ restricted to R, we compute the corresponding estimations recursively, as described

Algorithm 1 Recursive estimation procedure Estimate($p_{i_1 \dots i_{d-\ell}}(\vec{x}), \ell, R, \vec{s}$)

Input: n-variate degree-ℓ polynomial $p_{i_1 \dots i_{d-\ell}}(\vec{x})$, $R \subseteq N$ and a value $s_j \in \{0,1\}$ for each $j \in R$

Output: Estimation $\rho_{i_1 \dots i_{d-\ell}}$ of $p_{i_1 \dots i_{d-\ell}}(\bar{\vec{s}})$, where $\bar{\vec{s}}_R = \vec{s}$

```
if ℓ = 0 then return c_{i1...id}      /* p_{i1...id}(x) is equal to the constant c_{i1...id} */
compute decomposition p_{i1...i(d-ℓ)}(x) = c_{i1...i(d-ℓ)} + Σ_{j∈N} x_j p_{i1...i(d-ℓ)j}(x)
for all j ∈ N do
    ρ_{i1...i(d-ℓ)j} ← Estimate(p_{i1...i(d-ℓ)j}(x), ℓ − 1, R, s)
ρ_{i1...i(d-ℓ)} ← c_{i1...i(d-ℓ)} + (|N|/|R|) Σ_{j∈R} s_j ρ_{i1...i(d-ℓ)j}
return ρ_{i1...i(d-ℓ)}
```

in Algorithm 1. Specifically, for the basis level $\ell = 0$ and each d-tuple $(i_1, \dots, i_d) \in N^d$ of indices, the corresponding estimation is the coefficient $c_{i_1 \dots i_d}$ of the monomial $x_{i_1} \cdots x_{i_d}$ in the expansion of $p(\vec{x})$. For each level ℓ, $1 \le \ell \le d-1$, and each $(d-\ell)$-tuple $(i_1, \dots, i_{d-\ell}) \in N^{d-\ell}$, given the level-$(\ell-1)$ estimations $\rho_{i_1 \dots i_{d-\ell} j}$ of $p_{i_1 \dots i_{d-\ell} j}(\bar{\vec{s}})$, for all $j \in N$, we compute the level-ℓ estimation $\rho_{i_1 \dots i_{d-\ell}}$ of $p_{i_1 \dots i_{d-\ell}}(\bar{\vec{s}})$ from $\vec{s}$ as follows:

$$\rho_{i_1 \dots i_{d-\ell}} = c_{i_1 \dots i_{d-\ell}} + \frac{n}{r} \sum_{j \in R} s_j \rho_{i_1 \cdots i_{d-\ell} j} \tag{11}$$

In Algorithm 1, $\bar{\vec{s}}$ is any vector in $\{0,1\}^n$ that agrees with $\vec{s}$ on the variables of R. Given the estimations $\rho_{i_1 \dots i_{d-\ell} j}$, for all $j \in N$, we can also compute the absolute value estimations $\bar{\rho}_{i_1 \dots i_{d-\ell}}$ at level ℓ. Due to the β-smoothness property of $p(\vec{x})$, we have that $|c_{i_1 \dots i_{d-\ell}}| \le \beta n^\ell$, for all levels $\ell \ge 0$. Moreover, we assume that $0 \le \bar{\rho}_{i_1 \dots i_{d-\ell}} \le \ell \beta n^\ell$ and $|\rho_{i_1 \dots i_{d-\ell}}| \le (\ell+1)\beta n^\ell$, for all levels $\ell \ge 1$. This assumption is wlog. because due to β-smoothness, any binary vector $\vec{x}$ is feasible for (d-IP) with such values for the estimations $\rho_{i_1 \dots i_{d-\ell}}$ and the absolute value estimations $\bar{\rho}_{i_1 \dots i_{d-\ell}}$.

▶ Remark. For simplicity, we state Algorithm 1 so that it computes, from $\vec{s}$, an estimation $\rho_{i_1 \dots i_{d-\ell}}$ of the value of a given degree-ℓ polynomial $p_{i_1 \dots i_{d-\ell}}(\vec{x})$ at $\bar{\vec{s}}$. So, we need to apply Algorithm 1 $O(n^{d-1})$ times, one for each polynomial that arises in the recursive decomposition, with the same sample R and the same assignment $\vec{s}$. We can easily modify Algorithm 1 so that a single call Estimate($p(\vec{x}), d, R, \vec{s}$) computes the estimations of all the polynomials that arise in the recursive decomposition of $p(\vec{x})$. Thus, we save a factor of d on the running time. The running time of the simple version is $O(dn^d)$, while the running time of the modified version is $O(n^d)$.

B.3 Sampling Lemma

We use the next lemma to show that if $\vec{s} = \vec{x}^*_R$, the estimations $\rho_{i_1 \dots i_{d-\ell}}$ computed by Algorithm 1 are close to $c_{i_1 \dots i_{d-\ell}} + \sum_{j \in N} x^*_j \rho_{i_1 \dots i_{d-\ell} j}$ with high probability.

▶ **Lemma 12.** *Let $\vec{x}$ be any binary vector and let $(\rho_j)_{j \in N}$ be any sequence such that for some integer $q \ge 0$ and some constant $\beta \ge 1$, $\rho_j \in [0, (q+1)\beta n^q]$, for all $j \in N$. For all integers $d \ge 1$ and for all $\alpha_1, \alpha_2 > 0$, we let $\gamma = \Theta(dq\beta/(\alpha_1^2 \alpha_2))$ and let R be a multiset of $r = \gamma n^{1-\delta} \ln n$ indices chosen uniformly at random with replacement from N, where $\delta \in (0,1]$ is any constant. If $\rho = (n/r) \sum_{j \in R} \rho_j x_j$ and $\hat{\rho} = \sum_{j \in N} \rho_j x_j$, with probability at least $1 - 2/n^{d+1}$,*

$$(1-\alpha_1)\hat{\rho} - (1-\alpha_1)\alpha_2 n^{q+\delta} \le \rho \le (1+\alpha_1)\hat{\rho} + (1+\alpha_1)\alpha_2 n^{q+\delta} \tag{12}$$

Proof. To provide some intuition, we observe that if $\hat{\rho} = \Omega(n^{q+\delta})$, we have $\Omega(n^{\delta})$ values $\rho_j = \Theta(n^q)$. These values are well-represented in the random sample R, with high probability, since the size of the sample is $\Theta(n^{1-\delta} \ln n)$. Therefore, $|\hat{\rho} - \rho| \le \alpha_1 \hat{\rho}$, with high probability, by standard Chernoff bounds. If $\hat{\rho} = o(n^{q+\delta})$, the lower bound in (12) becomes trivial, since it is non-positive, while $\rho \ge 0$. As for the upper bound, we increase the coefficients ρ_j to $\rho'_j \in [0, (q+1)\beta n^q]$, so that $\hat{\rho}' = \alpha_2 n^{q+\delta}$. Then, $\rho' \le (1+\alpha_1)\hat{\rho}' = (1+\alpha_1)\alpha_2 n^{q+\delta}$, with high probability, by the same Chernoff bound as above. Now the upper bound of (12) follows from $\rho \le \rho'$, which holds for any instantiation of the random sample R.

We proceed to formalize the idea above. For simplicity of notation, we let $B = (q+1)\beta n^q$ and $a_2 = \alpha_2/((q+1)\beta)$ throughout the proof. For each sample l, $l = 1, \dots, r$, we let X_l be a random variable distributed in $[0,1]$. For each index j, if the l-th sample is j, X_l becomes ρ_j/B, if $x_j = 1$, and becomes 0, otherwise. Therefore, $\mathbb{E}[X_l] = \hat{\rho}/(Bn)$. We let $X = \sum_{l=1}^{r} X_l$. Namely, X is the sum of r independent random variables identically distributed in $[0,1]$. Using that $r = \gamma n^{1-\delta} \ln n$, we have that $\mathbb{E}[X] = \gamma \hat{\rho} \ln n/(Bn^{\delta})$ and that $\rho = BnX/r = Bn^{\delta} X/(\gamma \ln n)$.

We distinguish between the case where $\hat{\rho} \ge a_2 Bn^{\delta}$ and the case where $\hat{\rho} < a_2 Bn^{\delta}$. We start with the case where $\hat{\rho} \ge a_2 Bn^{\delta}$. Then, by Chernoff bounds[3],

$$\begin{aligned}\mathbb{P}[|X - \mathbb{E}[X]| > \alpha_1 \mathbb{E}[X]] &\le 2\exp\left(-\frac{\alpha_1^2 \gamma \hat{\rho} \ln n}{3Bn^{\delta}}\right) \\ &\le 2\exp(-\alpha_1^2 a_2 \gamma \ln n/3) \le 2/n^{d+1}\end{aligned}$$

For the second inequality, we use that $\hat{\rho} \ge a_2 Bn^{\delta}$. For the last inequality, we use that $\gamma \ge 3(d+1)/(\alpha_1^2 a_2) = 3(d+1)(q+1)\beta/(\alpha_1^2 \alpha_2)$, since $a_2 = \alpha_2/((q+1)\beta)$. Therefore, with probability at least $1 - 2/n^{d+1}$,

$$(1-\alpha_1)\frac{\gamma \hat{\rho} \ln n}{Bn^{\delta}} \le X \le (1+\alpha_1)\frac{\gamma \hat{\rho} \ln n}{Bn^{\delta}}$$

Multiplying everything by $Bn/r = Bn^{\delta}/(\gamma \ln n)$, we have that with probability at least $1 - 2/n^{d+1}$, $(1-\alpha_1)\hat{\rho} \le \rho \le (1+\alpha_1)\hat{\rho}$, which clearly implies (12).

We proceed to the case where $\hat{\rho} < a_2 Bn^{\delta}$. Then, $(1-\alpha_1)\hat{\rho} < (1-\alpha_1)a_2 Bn^{\delta} = (1-\alpha_1)\alpha_2 n^{q+\delta}$. Therefore, since $\rho \ge 0$, because $\rho_j \ge 0$, for all $j \in N$, the lower bound of (12) on ρ is trivial. For the upper bound, we show that with probability at least $1 - 1/n^{d+1}$, $\rho \le (1+\alpha_1)a_2 Bn^{\delta} = (1+\alpha_1)\alpha_2 n^{q+\delta}$. To this end, we consider a sequence $(\rho'_j)_{j \in N}$ so that $\rho_j \le \rho'_j \le (q+1)\beta n^q$, for all $j \in N$, and $\hat{\rho}' = \sum_{j \in N} \rho'_j x_j = a_2 Bn^{q+\delta}$. We can obtain such a sequence by increasing an appropriate subset of ρ_j up to $(q+1)\beta n^q$ (if $\vec{x}$ does not contain enough 1's, we may also change some x_j from 0 to 1). For the new sequence, we let $\rho' = (n/r)\sum_{j \in R} \rho'_j x_j$ and observe that $\rho \le \rho'$, for any instantiation of the random sample R. Therefore,

$$\mathbb{P}[\rho > (1+\alpha_1)\alpha_2 n^{q+\delta}] \le \mathbb{P}[\rho' > (1+\alpha_1)\hat{\rho}'],$$

where we use that $\hat{\rho}' = a_2 Bn^{\delta} = \alpha_2 n^{q+\delta}$. By the choice of $\hat{\rho}'$, we can apply the same Chernoff bound as above and obtain that $\mathbb{P}[\rho' > (1+\alpha_1)\hat{\rho}'] \le 1/n^{d+1}$. ◀

[3] We use the following bound (see e.g., [19, Theorem 1.1]): Let $Y_1, \dots, Y_k$ be independent random variables identically distributed in $[0,1]$ and let $Y = \sum_{j=1}^{k} Y_j$. Then for all $\epsilon \in (0,1)$, $\mathbb{P}[|Y - \mathbb{E}[Y]| > \epsilon\, \mathbb{E}[Y]] \le 2\exp(-\epsilon^2\, \mathbb{E}[Y]/3)$.

Lemma 12 is enough for MAX-CUT and graph optimization problems, where the estimations $\rho_{i_1 \dots i_{d-\ell} j}$ are non-negative. For arbitrary smooth polynomials however, the estimations $\rho_{i_1 \dots i_{d-\ell} j}$ may also be negative. So, we need a generalization of Lemma 12 that deals with both positive and negative estimations. To this end, given a sequence of estimations $(\rho_j)_{j \in N}$, with $\rho_j \in [-(q+1)\beta n^q, (q+1)\beta n^q]$, we let $\rho_j^+ = \max\{\rho_j, 0\}$ and $\rho_j^- = \min\{\rho_j, 0\}$, for all $j \in N$. Namely, ρ_j^+ (resp. ρ_j^-) is equal to ρ_j, if ρ_j is positive (resp. negative), and 0, otherwise. Moreover, we let

$$\rho^+ = (n/r) \sum_{j \in R} \rho_j^+ x_j \,, \quad \hat{\rho}^+ = \sum_{j \in N} \rho_j^+ x_j \,, \quad \rho^- = (n/r) \sum_{j \in R} \rho_j^- x_j \quad \text{and} \quad \hat{\rho}^- = \sum_{j \in N} \rho_j^- x_j$$

Applying Lemma 12 once for positive estimations and once for negative estimations (with the absolute values of ρ_j^-, ρ^- and $\hat{\rho}^-$, instead), we obtain that with probability at least $1 - 4/n^{d+1}$, the following inequalities hold:

$$\begin{aligned} (1-\alpha_1)\hat{\rho}^+ - (1-\alpha_1)\alpha_2 n^{q+\delta} \leq \quad & \rho^+ \quad \leq (1+\alpha_1)\hat{\rho}^+ + (1+\alpha_1)\alpha_2 n^{q+\delta} \\ (1+\alpha_1)\hat{\rho}^- - (1+\alpha_1)\alpha_2 n^{q+\delta} \leq \quad & \rho^- \quad \leq (1-\alpha_1)\hat{\rho}^- + (1-\alpha_1)\alpha_2 n^{q+\delta} \end{aligned}$$

Using that $\rho = \rho^+ + \rho^-$ and that $\hat{\rho} = \hat{\rho}^+ + \hat{\rho}^-$, we obtain the following generalization of Lemma 12.

▶ **Lemma 13** (Sampling Lemma). *Let $\vec{x} \in \{0,1\}^n$ and let $(\rho_j)_{j \in N}$ be any sequence such that for some integer $q \geq 0$ and some constant $\beta \geq 1$, $|\rho_j| \leq (q+1)\beta n^q$, for all $j \in N$. For all integers $d \geq 1$ and for all $\alpha_1, \alpha_2 > 0$, we let $\gamma = \Theta(dq\beta/(\alpha_1^2 \alpha_2))$ and let R be a multiset of $r = \gamma n^{1-\delta} \ln n$ indices chosen uniformly at random with replacement from N, where $\delta \in (0,1]$ is any constant. If $\rho = (n/r)\sum_{j \in R} \rho_j x_j$, $\hat{\rho} = \sum_{j \in N} \rho_j x_j$ and $\bar{\rho} = \sum_{j \in N} |\rho_j|$, with probability at least $1 - 4/n^{d+1}$,*

$$\hat{\rho} - \alpha_1 \bar{\rho} - 2\alpha_2 n^{q+\delta} \leq \rho \leq \hat{\rho} + \alpha_1 \bar{\rho} + 2\alpha_2 n^{q+\delta} \tag{13}$$

For all constants $\epsilon_1, \epsilon_2 > 0$ and all constants c, we use Lemma 13 with $\alpha_1 = \epsilon_1$ and $\alpha_2 = \epsilon_2/2$ and obtain that for $\gamma = \Theta(dq\beta/(\epsilon_1^2 \epsilon_2))$, with probability at least $1 - 4/n^{d+1}$, the following holds for any binary vector $\vec{x}$ and any sequence of estimations $(\rho_j)_{j \in N}$ produced by Algorithm 1 with $\vec{s} = \vec{x}_R$ (note that in Algorithm 1, the additive constant c is included in the estimation ρ when its value is computed from the estimations ρ_j).

$$\overbrace{c + \frac{n}{r} \sum_{j \in R} \rho_j x_j}^{\rho} - \epsilon_1 \overbrace{\sum_{j \in N} |\rho_j|}^{\bar{\rho}} - \epsilon_2 n^{q+\delta} \leq c + \sum_{j \in N} x_j \rho_j \leq \overbrace{c + \frac{n}{r} \sum_{j \in R} \rho_j x_j}^{\rho} + \epsilon_1 \overbrace{\sum_{j \in N} |\rho_j|}^{\bar{\rho}} + \epsilon_2 n^{q+\delta} \tag{14}$$

Now, let us consider (d-IP) with the estimations computed by Algorithm 1 with $\vec{s} = \vec{x}_R^*$ (i.e., with the optimal assignment for the variables in the random sample R). Then, using (14) and taking the union bound over all constraints, which are at most $2n^{d-1}$, we obtain that with probability at least $1 - 8/n^2$, the optimal solution $\vec{x}^*$ is a feasible solution to (d-IP). So, from now on, we condition on the high probability event that $\vec{x}^*$ is a feasible solution to (d-IP) and to (d-LP).

B.4 The Value of Feasible Solutions to (d-LP)

From now on, we focus on estimations $\vec{\rho}$ produced by Estimate($p(\vec{x}), d, R, \vec{s}$), where R is a random sample from N and $\vec{s} = \vec{x}_R^*$, and the corresponding programs (d-IP) and (d-LP).

The analysis in Section B.2 implies that $\vec{x}^*$ is a feasible solution to (d-IP) (and to (d-LP)), with high probability.

We next show that for any feasible solution $\vec{y}$ of (d-LP) and any polynomial $q(\vec{x})$ in the decomposition of $p(\vec{x})$, the value of $q(\vec{y})$ is close to the value of $c + \sum_j y_j \rho_j$ in the constraint of (d-LP) corresponding to q. Applying Lemma 14, we show below (see Lemma 15) that $p(\vec{y})$ is close to $c + \sum_{j \in N} y_j \rho_j$, i.e., to the objective value of $\vec{y}$ in (d-LP) and (d-IP), for any feasible solution $\vec{y}$.

To state and prove the following lemma, we introduce *cumulative absolute value estimations* $\bar{\tau}_{i_1 \dots i_{d-\ell}}$, defined recursively as follows: For level $\ell = 1$ and each tuple $(i_1, \dots, i_{d-1}) \in N^{d-1}$, we let $\bar{\tau}_{i_1 \dots i_{d-1}} = \bar{\rho}_{i_1 \dots i_{d-1}} = \sum_{j \in N} |c_{i_1 \dots i_{d-1} j}|$. For each level $\ell \geq 2$ of the decomposition of $p(\vec{x})$ and each tuple $(i_1, \dots, i_{d-\ell}) \in N^{d-\ell}$, we let $\bar{\tau}_{i_1 \dots i_{d-\ell}} = \bar{\rho}_{i_1 \dots i_{d-\ell}} + \sum_{j \in N} \bar{\tau}_{i_1 \dots i_{d-\ell} j}$. Namely, each cumulative absolute value estimation $\bar{\tau}_{i_1 \dots i_{d-\ell}}$ is equal to the sum of all absolute value estimations that appear below the root of the decomposition tree of $p_{i_1 \dots i_{d-\ell}}(\vec{x})$.

▶ **Lemma 14.** *Let $q(\vec{x})$ be any ℓ-degree polynomial appearing in the decomposition of $p(\vec{x})$, let $q(\vec{x}) = c + \sum_{j \in N} x_j q_j(\vec{x})$ be the decomposition of $q(\vec{x})$, let ρ and $\{\rho_j\}_{j \in N}$ be the estimations of q and $\{q_j\}_{j \in N}$ produced by Algorithm 1 and used in (d-LP), and let $\bar{\tau}$ and $\{\bar{\tau}_j\}_{j \in N}$ be the corresponding cumulative absolute value estimations. Then, for any feasible solution $\vec{y}$ of (d-LP)*

$$\rho - \epsilon_1 \bar{\tau} - \ell \epsilon_2 n^{\ell - 1 + \delta} \leq q(\vec{y}) \leq \rho + \epsilon_1 \bar{\tau} + \ell \epsilon_2 n^{\ell - 1 + \delta} \tag{15}$$

Proof. The proof is by induction on the degree ℓ. The basis, for $\ell = 1$, is trivial, because in the decomposition of $q(\vec{x})$, each $q_j(\vec{x})$ is a constant c_j. Therefore, Algorithm 1 outputs $\rho_j = c_j$ and

$$q(\vec{y}) = c + \sum_{j \in N} y_j q_j(\vec{x}) = c + \sum_{j \in N} y_j c_j \in \rho \pm \epsilon_1 \bar{\tau} \pm \epsilon_2 n^{\delta},$$

where the inclusion follows from the feasibility of $\vec{y}$ for (d-LP). We also use that at level $\ell = 1$, $\bar{\tau} = \bar{\rho}$ (i.e., cumulative absolute value estimations and absolute value estimations are identical).

We inductively assume that (15) is true for all degree-$(\ell - 1)$ polynomials $q_j(\vec{x})$ that appear in the decomposition of $q(\vec{x})$ and establish the lemma for $q(\vec{x}) = c + \sum_{j \in N} x_j q_j(\vec{x})$. We have that:

$$\begin{aligned} q(\vec{y}) = c + \sum_{j \in N} y_j q_j(\vec{y}) &\in c + \sum_{j \in N} y_j \left(\rho_j \pm \epsilon_1 \bar{\tau}_j \pm (\ell - 1)\epsilon_2 n^{\ell - 2 + \delta}\right) \\ &= \left(c + \sum_{j \in N} y_j \rho_j\right) \pm \epsilon_1 \sum_{j \in N} y_j \bar{\tau}_j \pm (\ell - 1)\epsilon_2 \sum_{j \in N} y_j n^{\ell - 2 + \delta} \\ &\in \left(\rho \pm \epsilon_1 \bar{\rho} \pm \epsilon_2 n^{\ell - 1 + \delta}\right) \pm \epsilon_1 \sum_{j \in N} \bar{\tau}_j \pm (\ell - 1)\epsilon_2 n^{\ell - 1 + \delta} \\ &\in \rho \pm \epsilon_1 \bar{\tau} \pm \ell \epsilon_2 n^{\ell - 1 + \delta} \end{aligned}$$

The first inclusion holds by the induction hypothesis. The second inclusion holds because (i) $\vec{y}$ is a feasible solution to (d-LP) and thus, $c + \sum_{j \in N} y_j \rho_j$ satisfies the corresponding constraint; (ii) $\sum_{j \in N} y_j \bar{\tau}_j \leq \sum_{j \in N} \bar{\tau}_j$; and (iii) $\sum_{j \in N} y_j \leq n$. The last inclusion holds because $\bar{\tau} = \bar{\rho} + \sum_{j \in N} \bar{\tau}_j$, by the definition of cumulative absolute value estimations. ◀

Using Lemma 14 and the notion of cumulative absolute value estimations, we next show that $p(\vec{y})$ is close to $c + \sum_{j \in N} y_j \rho_j$, for any feasible solution $\vec{y}$.

▶ **Lemma 15.** *Let $p(\vec{x}) = c + \sum_{j\in N} x_j p_j(\vec{x})$ be the decomposition of $p(\vec{x})$, let $\{\rho_j\}_{j\in N}$ be the estimations of $\{p_j\}_{j\in N}$ produced by Algorithm 1 and used in (d-LP), and let $\{\bar{\tau}_j\}_{j\in N}$ be the corresponding cumulative absolute value estimations. Then, for any feasible solution $\vec{y}$ of (d-LP)*

$$p(\vec{y}) \in c + \sum_{j\in N} y_j\rho_j \pm \epsilon_1 \sum_{j\in N} \bar{\tau}_j \pm (d-1)\epsilon_2 n^{d-1+\delta} \tag{16}$$

Proof. By Lemma 14, for any polynomial p_j, $p_j(\vec{y}) \in \rho_j \pm \epsilon_1\bar{\tau}_j \pm (d-1)\epsilon_2 n^{d-2+\delta}$. Therefore,

$$\begin{aligned} p(\vec{y}) = c + \sum_{j\in N} y_j p_j(\vec{y}) &\in c + \sum_{j\in N} y_j \left(\rho_j \pm \epsilon_1\bar{\tau}_j \pm (d-1)\epsilon_2 n^{d-2+\delta}\right) \\ &= c + \sum_{j\in N} y_j\rho_j \pm \epsilon_1 \sum_{j\in N} y_j\bar{\tau}_j \pm (d-1)\epsilon_2 \sum_{j\in N} y_j n^{d-2+\delta} \\ &\in c + \sum_{j\in N} y_j\rho_j \pm \epsilon_1 \sum_{j\in N} \bar{\tau}_j \pm (d-1)\epsilon_2 n^{d-1+\delta} \end{aligned}$$

The second inclusion holds because $y_j \in [0,1]$ and $\sum_{j\in N} y_j \le n$. ◀

B.5 Randomized Rounding of the Fractional Optimum

The last step is to round the fractional optimum $\vec{y}^* = (y_1^*, \dots, y_n^*)$ of (d-LP) to an integral solution $\vec{z} = (z_1, \dots, z_n)$ that almost satisfies the constraints of (d-IP) and has an expected objective value for (d-IP) very close to the objective value of $\vec{y}^*$.

To this end, we use randomized rounding, as in [28]. In particular, we set independently each z_j to 1, with probability y_j^*, and to 0, with probability $1 - y_j^*$. The analysis is based on the following lemma, whose proof is similar to the proof of Lemma 12.

▶ **Lemma 16.** *Let $\vec{y} \in [0,1]^n$ be any fractional vector and let $\vec{z} \in \{0,1\}^n$ be an integral vector obtained from $\vec{y}$ by randomized rounding. Also, let $(\rho_j)_{j\in N}$ be any sequence such that for some integer $q \ge 0$ and some constant $\beta \ge 1$, $\rho_j \in [0, (q+1)\beta n^q]$, for all $j \in N$. For all integers $k \ge 1$ and for all constants $\alpha, \delta > 0$ (and assuming that n is sufficiently large), if $\rho = \sum_{j\in N} \rho_j z_j$ and $\hat{\rho} = \sum_{j\in N} \rho_j y_j$, with probability at least $1 - 2/n^{k+1}$,*

$$(1-\alpha)\hat{\rho} - (1-\alpha)\alpha n^{q+\delta} \le \rho \le (1+\alpha)\hat{\rho} + (1+\alpha)\alpha n^{q+\delta} \tag{17}$$

Proof. We first note that $\mathbb{E}[\rho] = \hat{\rho}$. If $\hat{\rho} = \Omega(n^q \ln n)$, then $|\rho - \hat{\rho}| \le \alpha\hat{\rho}$, with high probability, by standard Chernoff bounds. If $\hat{\rho} = o(n^q \ln n)$, the lower bound in (17) becomes trivial, because $\rho \ge 0$ and $o(n^q \ln n) < \alpha n^{q+\delta}$, if n is sufficiently large. As for the upper bound, we increase the coefficients ρ_j to $\rho_j' \in [0, (q+1)\beta n^q]$, so that $\hat{\rho}' = \Theta(n^q \ln n)$. Then, the upper bound is shown as in the second part of the proof of Lemma 12.

We proceed to the formal proof. For simplicity of notation, we let $B = (q+1)\beta n^q$ throughout the proof. For $j = 1, \dots, n$, we let $X_j = z_j\rho_j/B$ be a random variable distributed in $[0,1]$. Each X_j independently takes the value ρ_j/B, with probability y_j, and 0, otherwise. We let $X = \sum_{j=1}^n X_j$ be the sum of these independent random variables. Then, $\mathbb{E}[X] = \hat{\rho}/B$ and $X = \sum_{j\in N} z_j\rho_j/B = \rho/B$.

As in Lemma 12, we distinguish between the case where $\hat{\rho} \ge 3(k+1)B\ln n/\alpha^2$ and the case where $\hat{\rho} < 3(k+1)B\ln n/\alpha^2$. We start with the case where $\hat{\rho} \ge 3(k+1)B\ln n/\alpha^2$. Then, by Chernoff bounds (we use the bound in footnote 3),

$$\mathbb{P}[|X - \mathbb{E}[X]| > \alpha\mathbb{E}[X]] \le 2\exp\left(-\frac{\alpha^2\hat{\rho}}{3B}\right) \le 2\exp(-(k+1)\ln n) \le 2/n^{k+1}\,,$$

where we use that $\hat{\rho} \geq 3(k+1)B\ln n/\alpha^2$. Therefore, with probability at least $1-2/n^{k+1}$,

$$(1-\alpha)\hat{\rho}/B \leq X \leq (1+\alpha)\hat{\rho}/B$$

Multiplying everything by B and using that $X = \rho/B$, we obtain that with probability at least $1-2/n^{k+1}$, $(1-\alpha)\hat{\rho} \leq \rho \leq (1+\alpha)\hat{\rho}$, which implies (17).

We proceed to the case where $\hat{\rho} < 3(k+1)B\ln n/\alpha^2$. Then, assuming that n is large enough that $n^\delta/\ln n > 3(k+1)(q+1)\beta/\alpha^3$, we obtain that $(1-\alpha)\hat{\rho} < (1-\alpha)\alpha n^{q+\delta}$. Therefore, since $\rho \geq 0$, because $\rho_j \geq 0$, for all $j \in N$, the lower bound of (17) on ρ is trivial. For the upper bound, we show that with probability at least $1-1/n^{k+1}$, $\rho \leq (1+\alpha)\alpha n^{q+\delta}$. To this end, we consider a sequence $(\rho'_j)_{j\in N}$ so that $\rho_j \leq \rho'_j \leq (q+1)\beta n^q$, for all $j \in N$, and

$$\hat{\rho}' = \sum_{j\in N} \rho'_j y_j = \frac{3(k+1)B\ln n}{\alpha^2}$$

We can obtain such a sequence by increasing an appropriate subset of ρ_j up to $(q+1)\beta n^q$ (if $\sum_{j\in N} \vec{y}$ is not large enough, we may also increase some y_j up to 1). For the new sequence, we let $\rho' = \sum_{j\in R} \rho'_j z_j$ and observe that $\rho \leq \rho'$, for any instantiation of the randomized rounding (if some y_j are increased, the inequality below follows from a standard coupling argument). Therefore,

$$\mathbb{P}[\rho > (1+\alpha)\alpha n^{q+\delta}] \leq \mathbb{P}[\rho' > (1+\alpha)\hat{\rho}'] \,,$$

where we use that $\hat{\rho}' = 3(k+1)B\ln n/\alpha^2$ and that $\alpha n^\delta > 3(k+1)(q+1)\beta\ln n/\alpha^2$, which holds if n is sufficiently large. By the choice of $\hat{\rho}'$, we can apply the same Chernoff bound as above and obtain that $\mathbb{P}[\rho' > (1+\alpha)\hat{\rho}'] \leq 1/n^{k+1}$. ◀

Lemma 16 implies that if the estimations ρ_j are non-negative, the rounded solution $\vec{z}$ is almost feasible for (d-IP) with high probability. But, as in Section B.2, we need a generalization of Lemma 16 that deals with both positive and negative estimations. To this end, we work as in the proof of Lemma 13. Given a sequence of estimations $(\rho_j)_{j\in N}$, with $\rho_j \in [-(q+1)\beta n^q, (q+1)\beta n^q]$, we define $\rho_j^+ = \max\{\rho_j, 0\}$ and $\rho_j^- = \min\{\rho_j, 0\}$, for all $j \in N$. Moreover, we let $\rho^+ = \sum_{j\in N} \rho_j^+ z_j$, $\hat{\rho}^+ = \sum_{j\in N} \rho_j^+ y_j$, $\rho^- = \sum_{j\in N} \rho_j^- z_j$ and $\hat{\rho}^- = \sum_{j\in N} \rho_j^- y_j$. Applying Lemma 16, once for positive estimations and once for negative estimations (with the absolute values of ρ_j^-, ρ^- and $\hat{\rho}^-$, instead), we obtain that with probability at least $1-4/n^{k+1}$,

$$\begin{aligned} (1-\alpha)\hat{\rho}^+ - (1-\alpha)\alpha n^{q+\delta} \leq \quad & \rho^+ \quad \leq (1+\alpha)\hat{\rho}^+ + (1+\alpha)\alpha n^{q+\delta} \\ (1+\alpha)\hat{\rho}^- - (1+\alpha)\alpha n^{q+\delta} \leq \quad & \rho^- \quad \leq (1-\alpha)\hat{\rho}^- + (1-\alpha)\alpha n^{q+\delta} \end{aligned}$$

Using that $\rho = \rho^+ + \rho^-$ and that $\hat{\rho} = \hat{\rho}^+ + \hat{\rho}^-$, we obtain the following generalization of Lemma 16.

▶ **Lemma 17** (Rounding Lemma). *Let $\vec{y} \in [0,1]^n$ be any fractional vector and let $\vec{z} \in \{0,1\}^n$ be an integral vector obtained from $\vec{y}$ by randomized rounding. Also, let $(\rho_j)_{j\in N}$ be any sequence such that for some integer $q \geq 0$ and some constant $\beta \geq 1$, $|\rho_j| \leq (q+1)\beta n^q$, for all $j \in N$. For all integers $k \geq 1$ and for all constants $\alpha, \delta > 0$ (and assuming that n is sufficiently large), if $\rho = \sum_{j\in N} \rho_j z_j$, $\hat{\rho} = \sum_{j\in N} \rho_j y_j$ and $\bar{\rho} = \sum_{j\in N} |\rho_j|$, with probability at least $1-4/n^{k+1}$,*

$$\hat{\rho} - \alpha\bar{\rho} - 2\alpha n^{q+\delta} \leq \rho \leq \hat{\rho} + \alpha\bar{\rho} + 2\alpha n^{q+\delta} \tag{18}$$

For all constants $\epsilon_1, \epsilon_2 > 0$ and all constants c, we can use Lemma 17 with $\alpha = \max\{\epsilon_1, \epsilon_2/2\}$ and obtain that for all integers $k \geq 1$, with probability at least $1 - 4/n^{k+1}$, the following holds for the binary vector $\vec{z}$ obtained from a fractional vector $\vec{y}$ by randomized rounding.

$$c + \sum_{j \in N} y_j \rho_j - \epsilon_1 \overbrace{\sum_{j \in N} |\rho_j|}^{\bar{\rho}} - \epsilon_2 n^{q+\delta} \leq c + \sum_{j \in N} z_j \rho_j \leq c + \sum_{j \in N} y_j \rho_j + \epsilon_1 \overbrace{\sum_{j \in N} |\rho_j|}^{\bar{\rho}} + \epsilon_2 n^{q+\delta} \quad (19)$$

Using (19) with $k = 2(d+1)$, the fact that $\vec{y}^*$ is a feasible solution to (d-LP), and the fact that (d-LP) has at most $2n^{d-1}$ constraints, we obtain that $\vec{z}$ is an almost feasible solution to (d-IP) with high probability. Namely, with probability at least $1 - 8/n^{d+4}$, the integral vector $\vec{z}$ obtained from the fractional optimum $\vec{y}^*$ by randomized rounding satisfies the following system of inequalities for all levels $\ell \geq 1$ and all tuples $(i_1, \dots, i_{d-\ell}) \in N^{d-\ell}$ (for each level $\ell \geq 1$, we use $q = \ell - 1$, since $|\rho_{i_1 \dots i_{d-\ell} j}| \leq \ell \beta n^{\ell-1}$ for all $j \in N$).

$$c_{i_1 \dots i_{d-\ell}} + \sum_{j \in N} z_j \rho_{i_1 \dots i_{d-\ell} j} \in \rho_{i_1 \dots i_{d-\ell}} \pm 2\epsilon_1 \bar{\rho}_{i_1 \dots i_{d-\ell}} \pm 2\epsilon_2 n^{\ell-1+\delta} \quad (20)$$

Having established that $\vec{z}$ is an almost feasible solution to (d-IP), with high probability, we proceed as in Section 3.4. By linearity of expectation, $\mathbb{E}[\sum_{j \in N} z_j \rho_j] = \sum_{j \in V} y_j^* \rho_j$. Moreover, the probability that $\vec{z}$ does not satisfy (20) for some level $\ell \geq 1$ and some tuple $(i_1, \dots, i_{d-\ell}) \in N^{d-\ell}$ is at most $8/n^{d+4}$ and the objective value of (d-IP) is at most $2(d+1)\beta n^d$, because, due to the β-smoothness property of $p(\vec{x})$, $|p(\vec{x}^*)| \leq (d+1)\beta n^d$. Therefore, the expected value of a rounded solution $\vec{z}$ that satisfies the family of inequalities (20) for all levels and tuples is least $\sum_{j \in V} y_j^* \rho_j - 1$ (assuming that n is sufficiently large). Using the method of conditional expectations, as in [27], we can find in (deterministic) polynomial time an integral solution $\vec{z}$ that satisfies the family of inequalities (20) for all levels and tuples and has $c + \sum_{j \in V} z_j \rho_j \geq c - 1 + \sum_{j \in V} y_j^* \rho_j$. As in Section 3.4, we sometimes abuse the notation and refer to such an integral solution $\vec{z}$ (computed deterministically) as the integral solution obtained from $\vec{y}^*$ by randomized rounding.

The following lemmas are similar to Lemma 14 and Lemma 15. They use the notion of cumulative absolute value estimations and show that the objective value $p(\vec{z})$ of the rounded solution $\vec{z}$ is close to the optimal value of (d-LP).

▶ **Lemma 18.** *Let $\vec{y}^*$ be an optimal solution of (d-LP) and let $\vec{z}$ be the integral solution obtained from $\vec{y}^*$ by randomized rounding (and the method of conditional expectations). Then, for any level $\ell \geq 1$ in the decomposition of $p(\vec{x})$ and any tuple $(i_1, \dots, i_{d-\ell}) \in N^{d-\ell}$,*

$$p_{i_1 \dots i_{d-\ell}}(\vec{z}) \in \rho_{i_1 \dots i_{d-\ell}} \pm 2\epsilon_1 \bar{\tau}_{i_1 \dots i_{d-\ell}} \pm 2\ell \epsilon_2 n^{\ell-1+\delta} \quad (21)$$

Proof. The proof is by induction on the degree ℓ and similar to the proof of Lemma 14. The basis, for $\ell = 1$, is trivial, because in the decomposition of $p(\vec{x})$, each $p_{i_1 \dots i_d}(\vec{x})$ is a constant $c_{i_1 \dots i_d}$. Therefore, $\rho_{i_1 \dots i_d} = c_{i_1 \dots i_d}$ and

$$p_{i_1 \dots i_{d-1}}(\vec{z}) = c + \sum_{j \in N} z_j p_{i_1 \dots i_{d-1} j}(\vec{z}) = c + \sum_{j \in N} z_j c_{i_1 \dots i_{d-1} j} \in \rho_{i_1 \dots i_{d-1}} \pm 2\epsilon_1 \bar{\tau}_{i_1 \dots i_{d-1}} \pm 2\epsilon_2 n^{\delta} ,$$

where the inclusion follows from the approximate feasibility of $\vec{z}$ for (d-LP), as expressed by (20). We also use that at level $\ell = 1$, $\bar{\tau}_{i_1 \dots i_{d-1}} = \bar{\rho}_{i_1 \dots i_{d-1}}$.

We inductively assume that (21) is true for the values of all degree-$(\ell - 1)$ polynomials $p_{i_1 \dots i_{d-\ell} j}$ at $\vec{z}$ and establish the lemma for $p_{i_1 \dots i_{d-\ell}}(\vec{z}) = c_{i_1 \dots i_{d-\ell}} + \sum_{j \in N} z_j p_{i_1 \dots i_{d-\ell} j}(\vec{z})$. We have that:

$$\begin{aligned}
p_{i_1 \dots i_{d-\ell}}(\vec{z}) &= c_{i_1 \dots i_{d-\ell}} + \sum_{j \in N} z_j p_{i_1 \dots i_{d-\ell} j}(\vec{z}) \\
&\in c_{i_1 \dots i_{d-\ell}} + \sum_{j \in N} z_j \left(\rho_{i_1 \dots i_{d-\ell} j} \pm 2\epsilon_1 \bar{\tau}_{i_1 \dots i_{d-\ell} j} \pm 2(\ell-1)\epsilon_2 n^{\ell-2+\delta} \right) \\
&= \left(c_{i_1 \dots i_{d-\ell}} + \sum_{j \in N} z_j \rho_{i_1 \dots i_{d-\ell} j} \right) \pm 2\epsilon_1 \sum_{j \in N} z_j \bar{\tau}_{i_1 \dots i_{d-\ell} j} \pm 2(\ell-1)\epsilon_2 \sum_{j \in N} z_j n^{\ell-2+\delta} \\
&\in \left(\rho_{i_1 \dots i_{d-\ell}} \pm 2\epsilon_1 \bar{\rho}_{i_1 \dots i_{d-\ell}} \pm 2\epsilon_2 n^{\ell-1+\delta} \right) \pm 2\epsilon_1 \sum_{j \in N} \bar{\tau}_{i_1 \dots i_{d-\ell} j} \pm 2(\ell-1)\epsilon_2 n^{\ell-1+\delta} \\
&\in \rho_{i_1 \dots i_{d-\ell}} \pm 2\epsilon_1 \bar{\tau}_{i_1 \dots i_{d-\ell}} \pm 2\ell\epsilon_2 n^{\ell-1+\delta}
\end{aligned}$$

The first inclusion holds by the induction hypothesis. The second inclusion holds because: (i) $\vec{z}$ is an approximately feasible solution to (d-IP) and thus, $c_{i_1 \dots i_{d-\ell}} + \sum_{j \in N} z_j \rho_{i_1 \dots i_{d-\ell} j}$ satisfies (20); (ii) $\sum_{j \in N} z_j \bar{\tau}_{i_1 \dots i_{d-\ell} j} \leq \sum_{j \in N} \bar{\tau}_{i_1 \dots i_{d-\ell} j}$; and (iii) $\sum_{j \in N} z_j \leq n$. The last inclusion holds because $\bar{\tau}_{i_1 \dots i_{d-\ell}} = \bar{\rho}_{i_1 \dots i_{d-\ell}} + \sum_{j \in N} \bar{\tau}_{i_1 \dots i_{d-\ell} j}$, by the definition of cumulative absolute value estimations. ◀

▶ **Lemma 19.** *Let $\vec{y}^*$ be an optimal solution of (d-LP) and let $\vec{z}$ be the integral solution obtained from $\vec{y}^*$ by randomized rounding (and the method of conditional expectations). Then,*

$$p(\vec{z}) \in c + \sum_{j \in N} z_j \rho_j \pm 2\epsilon_1 \sum_{j \in N} \bar{\tau}_j \pm 2(d-1)\epsilon_2 n^{d-1+\delta} \tag{22}$$

Proof. By Lemma 18, for any polynomial p_j appearing in the decomposition of $p(\vec{x})$, we have that $p_j(\vec{z}) \in \rho_j \pm 2\epsilon_1 \bar{\tau}_j \pm 2(d-1)\epsilon_2 n^{d-2+\delta}$. Therefore,

$$\begin{aligned}
p(\vec{z}) = c + \sum_{j \in N} z_j p_j(\vec{z}) &\in c + \sum_{j \in N} z_j \left(\rho_j \pm 2\epsilon_1 \bar{\tau}_j \pm 2(d-1)\epsilon_2 n^{d-2+\delta} \right) \\
&= c + \sum_{j \in N} z_j \rho_j \pm 2\epsilon_1 \sum_{j \in N} z_j \bar{\tau}_j \pm 2(d-1)\epsilon_2 \sum_{j \in N} z_j n^{d-2+\delta} \\
&\in c + \sum_{j \in N} z_j \rho_j \pm 2\epsilon_1 \sum_{j \in N} \bar{\tau}_j \pm 2(d-1)\epsilon_2 n^{d-1+\delta}
\end{aligned}$$

The second inclusion holds because $z_j \in \{0, 1\}$ and $\sum_{j \in N} z_j \leq n$. ◀

B.6 Cumulative Absolute Value Estimations of δ-Bounded Polynomials

To bound the total error of the algorithm, in Section B.7, we need an upper bound on $\sum_{j \in N} \bar{\tau}_j$, i.e., on the sum of the cumulative absolute value estimations at the top level of the decomposition of a β-smooth δ-bounded polynomial $p(\vec{x})$. In this section, we show that $\sum_{j \in N} \bar{\tau}_j = O(d^2 \beta n^{d-1+\delta})$. This upper bound is an immediate consequence of an upper bound of $O(d\beta n^{d-1+\delta})$ on the sum of the absolute value estimations, for each level ℓ of the decomposition of $p(\vec{x})$.

For simplicity and clarity, we assume, in the statements of the lemmas below and in their proofs, that the hidden constant in the definition of $p(\vec{x})$ as a δ-bounded polynomial is 1. If this constant is some $\kappa \geq 1$, we should multiply the upper bounds of Lemma 20 and Lemma 21 by κ.

▶ **Lemma 20.** *Let $p(\vec{x})$ be an n-variate degree-d β-smooth δ-bounded polynomial. Also let $\rho_{i_1 \dots i_{d-\ell}}$ and $\bar{\rho}_{i_1 \dots i_{d-\ell}}$ be the estimations and absolute value estimations, for all levels $\ell \in \{1, \dots, d-1\}$ of the decomposition of $p(\vec{x})$ and all tuples $(i_1, \dots, i_{d-\ell}) \in N^{d-\ell}$, computed by Algorithm 1 and used in (d-LP) and (d-IP). Then, for each level $\ell \geq 1$, the sum of the absolute value estimations is:*

$$\sum_{(i_1,\dots,i_{d-\ell}) \in N^{d-\ell}} \bar{\rho}_{i_1 \dots i_{d-\ell}} \leq \ell \beta n^{d-1+\delta} \tag{23}$$

Proof. The proof is by induction on the level ℓ of the decomposition. For the basis, we recall that for $\ell = 1$, level-1 absolute value estimations are defined as

$$\bar{\rho}_{i_1 \dots i_{d-1}} = \sum_{j \in N} |\rho_{i_1 \dots i_{d-1} j}| = \sum_{j \in N} |c_{i_1 \dots i_{d-1} j}|$$

This holds because, in Algorithm 1, each level-0 estimation $\rho_{i_1 \dots i_{d-1} i_d}$ is equal to the coefficient $c_{i_1 \dots i_{d-1} i_d}$ of the corresponding degree-d monomial. Hence, if $p(\vec{x})$ is a degree-d β-smooth δ-bounded polynomial, we have that

$$\sum_{(i_1,\dots,i_{d-1}) \in N^{d-1}} \bar{\rho}_{i_1 \dots i_{d-1}} = \sum_{(i_1,\dots,i_{d-1},j) \in N^{d}} |c_{i_1 \dots i_{d-1} j}| \leq \beta n^{d-1+\delta} \tag{24}$$

The upper bound holds because by the definition of degree-d β-smooth δ-bounded polynomials, for each $\ell \in \{0, \dots, d\}$, the sum, over all monomials of degree $d - \ell$, of the absolute values of their coefficients is $O(\beta n^{d-1+\delta})$ (and assuming that the hidden constant is 1, at most $\beta n^{d-1+\delta}$). In (24), we use this upper bound for $\ell = 0$ and for the absolute values of the coefficients of all degree-d monomials in the expansion of $p(\vec{x})$.

For the induction step, we consider any level $\ell \geq 2$. We observe that any binary vector $\vec{x}$ satisfies the level-$(\ell-1)$ constraints of (d-LP) and (d-IP) with certainty, if for each level-$(\ell-1)$ estimation,

$$\rho_{i_1 \dots i_{d-\ell} j} \leq c_{i_1 \dots i_{d-\ell} j} + \sum_{l \in N} |\rho_{i_1 \dots i_{d-\ell} j l}| = c_{i_1 \dots i_{d-\ell} j} + \bar{\rho}_{i_1 \dots i_{d-\ell} j}$$

We also note that we can easily enforce such upper bounds on the estimations computed by Algorithm 1. Since each level-ℓ absolute value estimation is defined as $\bar{\rho}_{i_1 \dots i_{d-\ell}} = \sum_{j \in N} |\rho_{i_1 \dots i_{d-\ell} j}|$, we obtain that for any level $\ell \geq 2$,

$$\begin{aligned} \sum_{(i_1,\dots,i_{d-\ell}) \in N^{d-\ell}} \bar{\rho}_{i_1 \dots i_{d-\ell}} &\leq \sum_{(i_1,\dots,i_{d-\ell},j) \in N^{d-\ell+1}} \left(|c_{i_1 \dots i_{d-\ell} j}| + \bar{\rho}_{i_1 \dots i_{d-\ell} j} \right) \\ &\leq \beta n^{d-1+\delta} + (\ell - 1) \beta n^{d-1+\delta} = \ell \beta n^{d-1+\delta} \end{aligned}$$

For the second inequality, we use the induction hypothesis and that since $p(\vec{x})$ is β-smooth and δ-bounded, the sum, over all monomials of degree $d - \ell + 1$, of the absolute values $|c_{i_1 \dots i_{d-\ell} j}|$ of their coefficients $c_{i_1 \dots i_{d-\ell} j}$ is at most $\beta n^{d-1+\delta}$. We also use the fact that the estimations are computed over the decomposition tree of the polynomial $p(\vec{x})$. Hence, each coefficient $c_{i_1 \dots i_{d-\ell} j}$ is included only once in the sum. ◀

▶ **Lemma 21.** *Let $p(\vec{x})$ be an n-variate degree-d β-smooth δ-bounded polynomial. Also let $\bar{\tau}_{i_1 \dots i_{d-\ell}}$ be the cumulative absolute value estimations, for all levels $\ell \in \{1, \dots, d-1\}$ of the decomposition of $p(\vec{x})$ and all tuples $(i_1, \dots, i_{d-\ell}) \in N^{d-\ell}$, corresponding to the estimations $\rho_{i_1 \dots i_{d-\ell}}$ computed by Algorithm 1 and used in (d-LP) and (d-IP). Then,*

$$\sum_{j \in N} \bar{\tau}_j \leq d(d-1) \beta n^{d-1+\delta} / 2 \tag{25}$$

Proof. Using induction on the level ℓ of the decomposition and Lemma 20, we show that for each level $\ell \geq 1$, the sum of the cumulative absolute value estimations is:

$$\sum_{(i_1,\ldots,i_{d-\ell})\in N^{d-\ell}} \bar{\tau}_{i_1\ldots i_{d-\ell}} \leq (\ell+1)\ell\beta n^{d-1+\delta}/2 \tag{26}$$

The conclusion of the lemma is obtained by applying (26) for the first level of the decomposition of $p(\vec{x})$, i.e., for $\ell = d-1$.

For the basis, we recall that for $\ell = 1$, level-1 cumulative absolute value estimations are defined as $\bar{\tau}_{i_1\ldots i_{d-1}} = \bar{\rho}_{i_1\ldots i_{d-1}}$. Using Lemma 20, we obtain that:

$$\sum_{(i_1,\ldots,i_{d-1})\in N^{d-1}} \bar{\tau}_{i_1\ldots i_{d-1}} = \sum_{(i_1,\ldots,i_{d-1})\in N^{d-1}} \bar{\rho}_{i_1\ldots i_{d-1}} \leq \beta n^{d-1+\delta}$$

We recall (see also Section B.4) that for each $\ell \geq 2$, level-ℓ cumulative absolute value estimations are defined as $\bar{\tau}_{i_1\ldots i_{d-\ell}} = \bar{\rho}_{i_1\ldots i_{d-\ell}} + \sum_{j\in N} \bar{\tau}_{i_1\ldots i_{d-\ell}j}$. Summing up over all tuples $(i_1,\ldots,i_{d-\ell}) \in N^{d-\ell}$, we obtain that for any level $\ell \geq 2$,

$$\begin{aligned}
\sum_{(i_1,\ldots,i_{d-\ell})\in N^{d-\ell}} \bar{\tau}_{i_1\ldots i_{d-\ell}} &= \sum_{(i_1,\ldots,i_{d-\ell})\in N^{d-\ell}} \left(\bar{\rho}_{i_1\ldots i_{d-\ell}} + \sum_{j\in N} \bar{\tau}_{i_1\ldots i_{d-\ell}j} \right) \\
&= \sum_{(i_1,\ldots,i_{d-\ell})\in N^{d-\ell}} \bar{\rho}_{i_1\ldots i_{d-\ell}} + \sum_{(i_1,\ldots,i_{d-\ell},j)\in N^{d-\ell-1}} \bar{\tau}_{i_1\ldots i_{d-\ell}j} \\
&\leq \ell\beta n^{d-1+\delta} + \ell(\ell-1)\beta n^{d-1+\delta}/2 = (\ell+1)\ell\beta n^{d-1+\delta}/2\,,
\end{aligned}$$

where the inequality follows from Lemma 20 and from the induction hypothesis. ◀

B.7 Concluding the Proof of Theorem 5

Therefore, for any constant $\varepsilon > 0$, if $p(\vec{x})$ is an n-variate degree-d β-smooth δ-bounded polynomial, the algorithm described in the previous sections computes an integral solution $\vec{z}$ that approximately maximizes $p(\vec{x})$. Specifically, setting $\epsilon_1 = \varepsilon/(4d(d-1)\beta)$ $\epsilon_2 = \varepsilon/(8(d-1))$, $p(\vec{z})$ satisfies the following with probability at least $1 - 8/n^2$:

$$\begin{aligned}
p(\vec{z}) &\geq \left(c + \sum_{j\in N} y_j^* \rho_j \right) - \frac{\varepsilon}{2d(d-1)\beta} \sum_{j\in N} \bar{\tau}_j - \varepsilon n^{d-1+\delta}/4 \\
&\geq \left(c + \sum_{j\in N} y_j^* \rho_j \right) - \varepsilon n^{d-1+\delta}/2 \\
&\geq \left(c + \sum_{j\in N} x_j^* \rho_j \right) - \varepsilon n^{d-1+\delta}/2 \\
&\geq \left(p(\vec{x}^*) - \frac{\varepsilon}{4d(d-1)\beta} \sum_{j\in N} \bar{\tau}_j - \varepsilon n^{d-1+\delta}/8 \right) - \varepsilon n^{d-1+\delta}/2 \\
&\geq p(\vec{x}^*) - \varepsilon n^{d-1+\delta}
\end{aligned}$$

The first inequality follows from Lemma 19. The second inequality follows from the hypothesis that $p(\vec{x})$ is β-smooth and δ-bounded. Then Lemma 21 implies that $\sum_{j\in N} \bar{\tau}_j \leq \frac{d(d-1)}{2}\beta n^{d-1+\delta}$. As in Section B.6, we assume that the constant hidden in the definition of

$p(\vec{x})$ as a δ-bounded polynomial is 1. If this constant is some $\kappa \geq 1$, we should also divide ϵ_1 by κ. The third inequality holds because $\vec{y}^*$ is an optimal solution to (d-LP) and $\vec{x}^*$ is a feasible solution to (d-LP). The fourth inequality follows from Lemma 15. For the last inequality, we again use Lemma 21. This concludes the proof of Theorem 5.

C Missing Proofs for k-Densest Subgraph

C.1 Proof of Lemma 6

Using the decomposition of $p(\vec{y})$ and the formulation of (LP′), we obtain that:

$$\begin{aligned} p(\vec{y}) = \sum_{j \in V} y_j \sum_{i \in N(j)} y_i &\in \sum_{j \in V} y_j \left((1 \pm \epsilon_1)\rho_j \pm \epsilon_2 n^{\delta/3}\right) \\ &= (1 \pm \epsilon_1) \sum_{j \in V} y_j \rho_j \pm \epsilon_2 n^{\delta/3} \sum_{j \in V} y_j \\ &\in (1 \pm \epsilon_1) \sum_{j \in V} y_j \rho_j \pm \epsilon_2 n^{1+\delta/3} \end{aligned}$$

The first inclusion holds because $\vec{y}$ is feasible for (LP′) and thus, $\sum_{i \in N(j)} y_i \in (1 \pm \epsilon_1)\rho_j \pm \epsilon_2 n^{\delta/3}$, for all j. The second inclusion holds because $\sum_{j \in V} y_j \leq n$. ◀

C.2 Proof of Lemma 7

We obtain that with probability at least $1 - 2/n^8$,

$$k - 2\sqrt{n \ln(n)} \leq \sum_{j \in V} z_i \leq k + 2\sqrt{n \ln(n)} \tag{27}$$

Specifically, the inequality above follows from the Chernoff bound in footnote 1, with $t = 2\sqrt{n \ln(n)}$, since $\mathbb{E}[\sum_{i \in N(j)} z_j] = k$. Moreover, applying Lemma 16 with $q = 0$, $\beta = 1$, $k = 7$, $\delta/3$ (instead of δ) and $\alpha = \max\{\epsilon_1, \epsilon_2/2\}$, and using that $\vec{y}^*$ is a feasible solution to (LP′) and that $\epsilon_1 \in (0, 1)$, we obtain that with probability at least $1 - 2/n^8$, for each vertex j,

$$(1 - \epsilon_1)^2 \rho_j - 2\epsilon_2 n^{\delta/3} \leq \sum_{i \in N(j)} z_i \leq (1 + \epsilon_1)^2 \rho_j + 2\epsilon_2 n^{\delta/3} \tag{28}$$

By the union bound, the integral solution $\vec{z}$ obtained from $\vec{y}^*$ by randomized rounding satisfies (27) and (28), for all vertices j, with probability at least $1 - 3/n^7$.

By linearity of expectation, $\mathbb{E}[\sum_{j \in V} z_j \rho_j] = \sum_{j \in V} y_j^* \rho_j$. Moreover, since the probability that $\vec{z}$ does not satisfy either (27) or (28), for some vertex j, is at most $3/n^7$, and since the objective value of (IP′) is at most n^2, the expected value of a rounded solution $\vec{z}$ that (27) and (28), for all vertices j, is least $\sum_{j \in V} y_j^* \rho_j - 1$ (assuming that $n \geq 2$). As in Max-Cut, such an integral solution $\vec{z}$ can be found in (deterministic) polynomial time using the method of conditional expectations (see [27]).

Using the decomposition of $p(\vec{y})$ and an argument similar to that in the proof of Lemma 6, we obtain that:

$$\begin{aligned} p(\vec{z}) = \sum_{j\in V} z_j \sum_{i\in N(j)} z_i \quad &\in \sum_{j\in V} z_j \left((1\pm\epsilon_1)^2\rho_j \pm 2\epsilon_2 n^{\delta/3}\right) \\ &= (1\pm\epsilon_1)^2 \sum_{j\in V} z_j\rho_j \pm 2\epsilon_2 n^{\delta/3}\sum_{j\in V} z_j \\ &\in (1\pm\epsilon_1)^2 \sum_{j\in V} z_j\rho_j \pm 2\epsilon_2 n^{1+\delta/3} \\ &\in (1\pm\epsilon_1)^2 \sum_{j\in V} y_j^*\rho_j \pm 3\epsilon_2 n^{1+\delta/3} \end{aligned}$$

The first inclusion holds because $\vec{z}$ satisfies (28) for all $j \in V$. For the second inclusion, we use that $\sum_{j\in V} z_j \le n$. For the last inclusion, we recall that $\sum_{j\in V} z_j\rho_j \ge \sum_{j\in V} y_j^*\rho_j - 1$ and assume that n is sufficiently large. ◀

C.3 Proof of Theorem 8

For $\varepsilon > 0$, if G is δ-almost sparse and $k = \Omega(n^{1-\delta/3})$, the algorithm described computes estimations ρ_j such that the densest subgraph $\vec{x}^*$ is a feasible solution to (IP$'$) whp. Hence, by the analysis given, the algorithm computes a slightly infeasible solution approximating the number of edges in the densest subgraph with k vertices within a multiplicative factor of $(1-\epsilon_1)^2$ and an additive error of $\epsilon_2 n^{1+\delta/3}$. Setting $\epsilon_1 = \epsilon_2 = \varepsilon/8$, the number of edges in the subgraph induced by $\vec{z}$ satisfies the following with probability at least $1-2/n^2$:

$$\begin{aligned} p(\vec{z}) \;&\ge\; (1-\epsilon_1)^2 \sum_{j\in V} y_j^*\rho_j - 3\epsilon_2 n^{1+\delta/3} \ge (1-\epsilon_1)^2 \sum_{j\in V} x_j^*\rho_j - 3\epsilon_2 n^{1+\delta/3} \\ &\ge\; p(\vec{x}^*) - \varepsilon n^{1+\delta/3} \ge (1-\varepsilon)p(\vec{x}^*) \end{aligned}$$

The first inequality follows from Lemma 7, the second inequality holds because $\vec{y}^*$ is the optimal solution to (LP) and $\vec{x}^*$ is feasible for (LP), the third inequality follows from Lemma 6 and the fourth inequality holds because the optimal cut has at least $\Omega(n^{1+\delta/3})$ edges.

This solution is infeasible by at most $2\sqrt{n\ln n} = o(k)$ vertices and can become feasible by adding or removing at most so many vertices and $O(n^{1/2+\delta})$ edges. ◀

D Missing Hardness Proofs

Theorem 9. This inapproximability result follows from the construction of quasi-linear size PCPs given, for example, in [18]. In particular, we use as starting point a result explicitly formulated in [25] as follows: "Solving 3-SAT on inputs of size N can be reduced to distinguishing between the case that a 3CNF formula of size $N^{1+o(1)}$ is satisfiable and the case that only $\frac{7}{8}+o(1)$ fraction of its clauses are satisfiable".

Take an arbitrary 3-SAT instance of size N, which according to the ETH cannot be solved in time $2^{o(N)}$. By applying the aforementioned PCP construction we obtain a 3CNF formula of size $N^{1+o(1)}$ which is either satisfiable or far from satisfiable. Using standard constructions ([26, 6]) we can reduce this formula to a 5-regular graph $G(V,E)$ which will be a Max-CUT instance (we use degree 5 here for concreteness, any reasonable constant would do). We have that $|V|$ is only a constant factor apart from the size of the 3CNF formula. At the same time, there exist constants c, s such that, if the formula was satisfiable G has a cut of $c|E|$ edges, while if the formula was far from satisfiable G has no cut with more than $s|E|$ edges. If there exists an algorithm that can distinguish between these two cases in time $2^{|V|^{1-\epsilon}}$ the whole procedure would run in $2^{N^{1-\epsilon+o(1)}}$ and would allow us to decide if the original formula was satisfiable. ◀