DIMACS Technical Report 2001-42
November 2001

Efficient Amplifiers and Bounded Degree Optimization

by

Piotr Berman [1]
Dept. of Computer Science
and Engineering
The Pennsylvania State University

Marek Karpinski [2]
Dept. of Computer Science
University of Bonn.

[1]Research done in part while visiting Dept. of Computer Science, University of Bonn. Work partially supported by NSF grant CCR-9700053, NIH grant 9R01HG02238-12 and DFG grant Bo 56/157-1. E-mail `berman@cse.psu.edu`

[2]Research done in part while vising Dept. of Computer Science, Princeton University. Work partially supported by DFG grants, DIMACS, and IST grant 14036 (RAND-APX). E-mail `marek@cs.uni-bonn.de`

ABSTRACT

This paper studies the existence of efficient (small size) amplifiers for proving explicit inaproximability results for bounded degree and bounded occurrence combinatorial optimization problems, and gives an explicit construction for such amplifiers. We use this construction also later to improve the currently best known approximation lower bounds for bounded occurrence instances of linear equations mod 2, and for bounded degree (regular) instances of MAX-CUT. In particular we prove the approximation lower bound of 152/151 for exactly 3-occurrence E3-OCC-E2-LIN-2 problem, and MAX-CUT problem on 3-regular graphs, E3-MAX-CUT, and the approximation lower bound of 121/120 for E3-OCC-2-LIN-2 problem. As an application we are able to improve also the best known approximation lower bound for E3-OCC-MAX-E2SAT.

1 Introduction

We define Ed-OCC-Ek-LIN-2 as a problem of constructing an assignment that maximizes the number of satisfied equations for a given system of linear equations modulo 2 (hence LIN-2), where each equation has exactly k variables (hence Ek) and each variable occurs exactly d times. If we drop an E in the acronym of the problem than we have "at most d occurrences" or "at most k variables". Ed-MAX-CUT stands for the MAX-CUT problem restricted to d-regular graphs.

In [BK99] an approximation lower bound of 332/331 was proven for E3-OCC-E2-LIN-2 and 3-MAX-CUT, and in this paper we improve it to 152/151. Moreover, for E3-OCC-2-LIN-2 we obtain 121/120 lower bound, previously this problem had no better lower bound than E3-OCC-E2-LIN-2. We also obtain the approximation lower bound of 788/787 for E3-OCC-MAX-2SAT, an improvement over 2012/2011 lower bound of [BK99].

We refer to [BK99] for a "wheel-amplifier" method used to get some explicit bounded degree and bounded occurrence problems. For a survey on explicit approximation lower bounds for small degree (or small occurrence) optimization problems see [K01], and for some recent work on asymptotic relations between hardness of approximation and the bounds on a degree or the number of occurrences in optimization problems see [H00] and [T01].

2 Amplifiers

The notion of an amplifier generalizes the concept of a specific variety of expanders that are used in proving inapproximability results. This notion was introduced by Papadimitiou in [P94] (for directed graphs) and it formalizes the construction of Papadimitriou and Yannakakis of [PY91]. Although not always called that way, the amplifiers are critical in reducing more "primary" MAX-SNP complete problems to the problems with bounded degree or number of occurrences.

Definition 1. Consider an undirected graph $G = (V, E)$. We define

$Cut(U) = \{e \in E : \ e \not\subset U \text{ and } e \not\subset V - U\}$ and $cut(U) = |Cut(U)|$.

We say that G is a *strong expander* if for every $U \subset V$ we have $cut(U) \geq \min(|U|, |V - U|)$.

We say that G is an *amplifier for* $X \subset V$ if it contains no *bad sets* for X. A set $A \subset V$ is bad for X if $cut(A) < \min(||X \cap A|, |X - A|)$.

An amplifier for X is B-*regular* if each node in X has $B - 1$ neighbors and each node in $V - X$ has B neighbors.

The goal of this section is to provide a construction of small 3-regular amplifiers. In [PY91, P94] there is a description of B-regular strong expanders where $B \approx 80$. Arora and Lund [A95] (see also Ausiello *et. al.* [ACG$^+$99]) uses a result of Lubotzky *et. al.* [LPS88] to obtain a 14-regular strong expander.

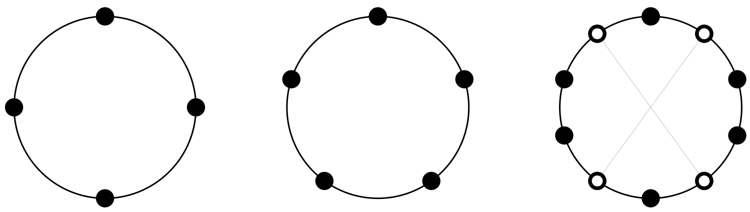

Figure 1: 3-regular amplifiers for $|X| = 4, 5, 6$, solid dots are the elements of X

.

To convert a B-regular strong expander with node set X to a 3-regular amplifier for X, most of the above papers basically do the following: they replace each node x with a connected node set V_x such that $x \in V_x$, x becomes a node of degree 2, and $cut(V_x) = B$.

This is however not always true as one can see in Fig. 2: a 3-regular strong expander is transformed into a graph that is not a 3-regular amplifier because it contains a *bad* set A such that $|A \cap X| = 5$ and $cut(A) = 4$.

The simplest correction of the above error would be to replace each node x of a B-regular amplifier with a 3-regular amplifier for a set $U_x \cup \{x\}$ where $|U| = B$, and then replace an edge $\{x, y\}$ with an edge between U_x and U_y. For example, if we pick 7 random matchings for the set X then with a high probability the resulting graph is a 7-regular strong expander, and there exists an amplifier for 8 nodes that consists of 14 nodes; as a result we can construct an 3 regular amplifier with $14|X|$ nodes.

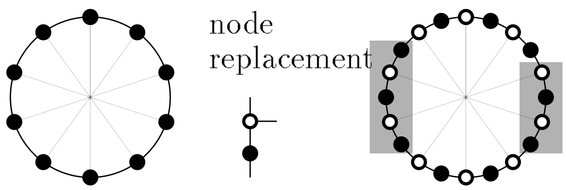

Figure 2: Incorrect amplifier for black dots obtained from a strong expander, on the right, a bad set is shaded.

However, we can show following [BK99] that an even simpler construction yields amplifiers with $7|X|$ nodes, i.e. twice smaller.

The constructions equivalent (or very similar) to our notion of amplifier can be used in lower approximability bounds of many combinatorial optimization problems with bounded degree or bounded number of occurrences, and the obtained bounds are related to the ratio $|V|/|X|$. Therefore we will use the following definition.

Definition 2.

An amplifier generator $\mathcal{G}$ is a randomized polynomial time algorithm that for a given n returns, with probability at least $1 - n^{-1}$, an amplifier $\mathcal{G}(n)$ for the set $\{1, 2, \ldots, n\}$.

A characteristic of amplifier generator $\mathcal{G}$ is a number α such that the set $\{n : |V(\mathcal{G}(n))| > \alpha n\}$ is finite.

The d-regular amplification number α_d is the largest lower bound of characteristics of d-regular amplifier generators.

For $d > 7$ we know that $\alpha_d = 1$ as there exist $(d-1)$-regular strong expanders, for $d < 3$, α_d is undefined because d-regular amplifiers do not exist. In the remaining cases the value of α_d remains an open problem. Below we prove that $\alpha_3 \leq 7$.

Definition 3. An *n-wheel* is a graph with $7 \times 2n$ nodes $W = \mathit{Contacts} \cup \mathit{Checkers}$, that contains $2n$ *contacts* and $12n$ *checkers*, and two sets of edges, C and M. C is a Hamiltonian cycle in which with consecutive contacts are separated by chains of 6 checkers, while M is a random perfect matching for the set of checkers (see Fig. 3 for an example).

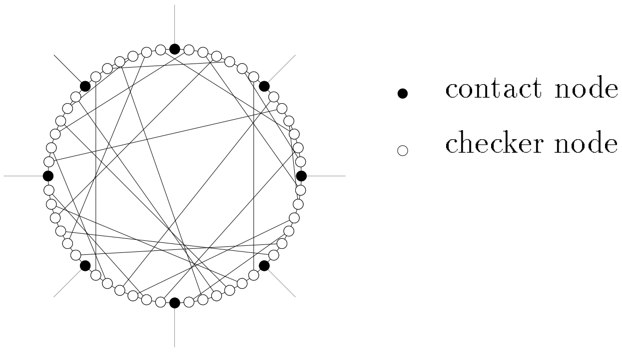

Figure 3: A 4-wheel.

In the remainder of this section we prove the following

Theorem 1. *An n-wheel forms a 3-regular amplifier for its set of $2n$ contacts with probability $1 - O(n^{-3})$.*

An n-wheel fails to be a 3-regular amplifier only if there exists a *wrong* set A, where being wrong means that $|A \cap \mathit{Contacts}\,| \leq n$ and $\mathit{cut}(A) < |A \cap \mathit{Contacts}\,|$. We need to show that a wrong set exists with probability $O(n^{-3})$.

If a set A is wrong, we say that $B = A \cap \mathit{Checkers}$ is *bad*. We need to characterize bad sets. For the remainder of this proof we convert W to a graph with set of nodes equal to *Checkers* by replacing each contact u with an edge (later called a contact edge) that connects the checkers that were adjacent to u. From now on we consider only this new graph.

Definition 4. For a set B, a_B^i is the number of contact edges that have exactly i endpoints in B;

$a_B = \min(a_B^1 + a_B^2, n)$;

$b_B = |Cut(B) \cap M|$;

$c_B = |Cut(B) \cap C|$;

B is bad if $a_B^2 \leq n$ and $b_B + c_B < a_B$.

Now it suffices to show that the probability that a bad subset of W exists is $O(n^{-3})$.

As a preliminary step, we must have some tools to estimate the probabilities in the random space consisting of perfect matchings. We will use the following definitions.

Definition 5. A set $A \subset W$ is M-closed iff $c_A = 0$, i.e. $Cut(A) \cap M = \varnothing$;

the function $\mu(m)$ denotes the number of perfect matchings in a clique with $2m$ nodes.

Lemma 2.

$$\mu(m) = \prod_{i=1}^{m}(2i-1) = \frac{(2m)!}{m!2^m}$$

Proof. By induction on m. For $m = 0$, there exists exactly one perfect matching. Now consider a clique with $2m$ nodes. A fixed node can be matched using any of the $2m - 1$ incident edges. We can complete the construction of the matching by choosing any of $\mu(m-1)$ matchings of the remaining $2m - 2$ nodes, thus $\mu(n) = (2m-1)\mu(m-1)$.

❒

Lemma 3. *The probability that a set of* $2d$ *checkers is* M*-closed is* $p(d) = \mu(d)\mu(n-d)/\mu(n)$, *or*

$$\prod_{i=1}^{d} \frac{2i-1}{2n-2i+1}$$

Proof. Straightforward consequence of Lemma 2.

❒

Lemma 4. *If* B *is a bad set,* B *contains an* M*-closed subset of size* $2d_B$*, where* $d_B = \lceil (s_B - a_B + c_B + 1)/2 \rceil$.

Proof. An M-closed set must have even size. We obtain an M-closed set S by removing from B all endpoints of the edges from $Cut(B) \cap M$, and $|S| = s_B - b_B$. Because B is bad, $-b_B \geq -a_B + c_B$, hence $|S| \geq s_B - a_B + c_B + 1$. If S is too large, we can decrease its size by removing endpoints of some edges of M that are contained in S.

❒

Our general method of estimating the probability of a bad set existing, is to consider separately cases when a bad set B has a particular vector of parameters a_B, b_B and $s_B = |B|$. For each of them we will

a) estimate the numbers of candidates for a bad set, such that if a bad set exists, than one of the candidates must be bad as well;

b) find the number of subsets of a candidate B, each of size $2d_B$, such that if B is bad, than one of these subsets must be M-closed;

c) multiply the product of the results of a) and b) with the probability $p(d_B)$.

While discussing a candidate for a bad set, say B, we will refer to *fragments* of B, connected components of B within cycle C (note that in the modified W, the cycle C consists of checkers only). The following lemma limits the number of candidates for an M-closed subset.

We will use two ways of estimating the probability that a bad set exists. The first one is applied to sufficiently small candidates.

Lemma 5. *For $a < n/9$, the probability that there exists a bad set B with $a_B = a$ is $O(n^{-3}0.6^a)$.*

Proof. We formulate the proof for *odd* value of a, the case of even a is very similar. We use the following notation: $a = a_B$ is the number of contacts of B (incident contact edges), $f = c_B/2$ is the number of fragments of B, $s = s_B$ is the size of B and b satisfies the identity $a = 1 + 2f + b$ (note that $b \geq b_B$).

First we will the upper and lower bounds for s. Consider a fragment of B that is incident to, say, a_o contact edges. This fragment must contain $a_o - 1$ chains of 6 nodes, and portions (possibly empty) of two other such chains on its fringes. Thus it contains between $6(a_o - 1)$ and $6(a_o + 1)$ nodes. By adding sizes of all fragments, we obtain $6(a - f) \leq s \leq 6(a + f)$.

One conclusion that we can draw is that $s < 9a \leq n$. Another is that $s \geq 6(a - f) = 6(1 + 2f + b - f)) = 6f + 6b + 6$. The latter implies that B contains an M-closed set of size $2d = s - b$ where $d \geq 3f + 2b + 3$.

We can generate a candidate B as follows. First, we select f of the "left ends" of the fragments; this can be done in at most $\mathbf{C}(12n, f)$ ways, where $\mathbf{C}$ is our notation for the binomial coefficient. Next, we distribute the sizes of the fragments; because the sum of sizes is less than n, and all of them are positive, this can be done in less than $\mathbf{C}(n, f)$ ways.

Given a candidate, we can obtain an M-closed set by removing some b nodes, this can be done in less than $\mathbf{C}(n, b)$ ways. Altogether, we generate an M-close set in less than $\mathbf{C}(12n, f)\mathbf{C}(n, f)\mathbf{C}(n, b)$ many ways, i.e.

$$\prod_{i=1}^{f} \frac{12r - i + 1}{i} \prod_{i=1}^{f} \frac{r - i}{i} \prod_{i=1}^{b} \frac{r - i}{i}. \quad < \quad \prod_{i=1}^{f} \frac{12r}{i} \prod_{i=1}^{f} \frac{r}{i} \prod_{i=1}^{b} \frac{r}{i} \quad = \quad \alpha\beta\gamma.$$

By Lemma 3, each of these candidates is M-closed with probability at most

$$\prod_{i=1}^{3f+2b+3} \frac{2i - 1}{12r - (2i - 1)} \quad < \quad \prod_{i=1}^{3f+2b+3} \frac{2i - 1}{11r} \quad = \quad \delta.$$

Note that the latter product has more terms than the first three combined, and that even the largest of these terms is less than $4a/11.5n < 4a/100a < 1/25$.

We set aside the first three terms of δ to get factor $O(n^{-3})$. Then we combine the i^{th} factors of α, β and γ with the factors of δ with numbers $3i+1$, $3i+2$ and $3i+3$ respectively. We have still at least $f+b$ factors of δ left, and since we estimate them by $1/25 = 1/15 \times 3/5$, we get the following overestimate of our probability:

$$\prod_{i=1}^{f}\left(\frac{1}{15}\frac{12r}{i}\frac{6i+1}{11r}\right)\prod_{i=1}^{f}\left(\frac{3}{5}\frac{r}{i}\frac{6i+3}{11r}\right)\prod_{i=1}^{b}\frac{1}{15}\left(\frac{3}{5}\frac{r}{i}\frac{6i+5}{11r}\right) \quad < \quad \left(\frac{6}{11}\right)^{a-1}\left(\frac{1}{15}\right)^{b}.$$

Thus the probability that a set with $a_B = a$ is bad is $O(n^{-3})(6/11)^{a-1}(\sum_{i=1}^{b} 15^{-i})$.

❒

To tackle the case of larger values of a_B we will need another lemma.

Lemma 6. *In a bad set B of minimum size nodes of C_B are not incident to edges of $Cut(B) \cap C$.*

Proof. Suppose that a node of C_B is incident to an edge of $Cut(B) \cap C$. If we remove it from B, b_B remains unchanged, c_B is decreased by 1 and a_B is decreased by at most 1 (the later decrease occurs if the edge in question is a contact edge). Thus we have obtain a smaller bad set, a contradiction.

❒

The next lemma finishes the proof of our theorem.

Lemma 7. *The probability that there exists a bad set B with $a_B = a > n/9$ is $O(n^3 0.81^a)$.*

Proof. We use variables a, s, c and f as in the proof of Lemma 5. We will overestimate the number of such candidates for a bad set B with these parameters in three ways and take them minimum. The first method, counting in how many ways we can select $2f$ edges of $Cut(B) \cap C$, yields $\mathbf{C}(12n, 2f)$.

To understand the second method, imagine that we label each edge of $Cut(B) \cap C$; if this edge has its right endpoint in B, we label it $<$, and otherwise (left endpoint in B) we label it $>$. Next, we move each $<$ label to the nearest contact edge to its right, and each $>$ label to its left. Finally, we move the labels back to their original positions. The positions of the labels at the time when they are all placed on the contact edges provides a lower bound on the size of B; a fragment that is incident to a_o contact edges will have its size estimated as $6(a_o - 1)$. (Note that fragments of B that do not have incident contact edges will obtain the size estimate of -6; this is because its $<$ label is at this time positioned 6 edges to the right of its $<$ label.) Therefore the sum of distances that the labels will traverse from their positions on the contact edges to their correct positions is $s - 6(a-f) = d$. This allows us to select any

B with parameters b and d as follows: first we select the positions of $2f$ labels on $2n$ contact nodes, this can be done in $\mathbf{C}(2n+2f-1, 2f)$ ways; subsequently we distribute d "units of displacement" to $2f$ labels, this can be done in $\mathbf{C}(d+2f-1, d)$ ways. Summarizing, the second method is to compute $d = s - 6(a-f)$ and return $\mathbf{C}(2n+2f-1, b)\mathbf{C}(d+2f-1, d)$.

The third method is very similar, except that we move the labels in the opposite directions. The resulting formula is identical, except that we compute d differently: $d = 6(a+f) - s$.

Note that if we obtain negative d while using the second method, we can conclude that s is too low to be compatible with a and f, similarly, negative d in the third method implies that s is too large. If s is neither too large nor too small, we estimate the number of the candidates for a bad set using the minimum of the results of the three methods described here.

By Lemma 4, B contains an M-closed subset S of size $s + 2f - a$ (plus minus one). Moreover, by Lemma 6, we may assume that no elements of $B - S$ are adjacent in C to the complement of B, hence only $s - b$ nodes may be considered for removal.

Our goal is to show that the probability computed according to the above principles, and raised to power $1/a$, is bounded by 0.81. We achieved this goal as follows. We define the real parameters of B as follows:

- α such that $a = \alpha n$;
- β such that $2f = \beta\alpha n$; because we are looking at the parameters of bad sets, we know that $0 < \alpha \leq 1$ and $0 < \beta \leq 1$;
- ξ such that $d = \xi\beta\alpha n$, if a respective counting method (second or third) is applicable, $0 \leq \xi \leq 3$.

Using Stirling's formula, and the above estimation formula, we can compute the $1/a$ power of our probability from the parameters α, β and ξ. To consider all possible cases, we can use parameter values that are multiples of some fraction, say ϵ; then, in a subexpression that is an decreasing functions of a parameter, we use the current multiple, say $i\epsilon$, and in an subexpression that is an increasing function, we use $(i+1)\epsilon$. This covers the case of all values between $i\epsilon$ and $(i+1)\epsilon$. In our program, we used the following values for ϵ: $1/20$ for ξ, $1/100$ for β and $1/2000$ for α. The worst case was obtained for $\alpha = 1$, $\beta = 0.77$ and $\xi = 1.15$ ($d = 4.5755n$) and it equals $e^{-0.2181} = 0.8041$.

❒

It remains open whether the same approach may prove a similar result for wheels with 5 checkers between each pair of contacts. In our attempts we introduced several parameters, like the number of fragments that are not incident to any contacts. Even though we were not successful, the logarithm of the target number was estimated to be 0.03. We believe that with an improved counting method this estimate can be decreased below 0.

3 Eq-Reductions

For the purposes of this paper we introduce the following notion that is applicable to MAX-SNP hard commbinatorial optimization problems:

> $(f(n), g(n))$ gap property of problem A means that for every sufficiently small positive ε it is NP-hard to distinguish between two groups of instances of A of size n: those that have no solutions with score above $f(n) + \varepsilon n$¡ and those that have solutions with score at least $g(n) - \varepsilon n$.

Of course, this notion of gap property can be easily modified for minimization problems as well. While not formalized in exactly this fashion, gap properties were widely used in proving lower bounds on approximation ratios that can be attained by polynomial time algorithms.

For example, Håstad [H97] has shown that if $0 < \varepsilon < 0.5$ then for systems of n linear equations modulo 2 with 3 variables per equation it is NP-hard to distinguish between instances where a solution may satisfy $n - \varepsilon$ equations and instances where no solution satisfies more then $n/2 + \varepsilon$ equations. Thus the problem E3-LIN-2 has instances with even number of equations modulo 2 with 3 variables each, n the number of equations in an instance and this problem has $(n/2 + \varepsilon, n - \varepsilon)$ gap property. We will be omitting ε terms, so we can say that this problem has $(n/2, n)$ gap property.

Eq-reductions are tools to prove gap properties.

Consider two maximization problem, A and B with objective functions a and b. An Eq-reduction from A to B has 5 randomized polynomial time computable functions, τ, t, ν, ρ and r, in its description:

- instance translation τ and parameter translation t; if x is an instance of A with parameter n then $\tau(x)$ is an instance of B with parameter $t(n)$;
- solution normalization ν; if y is a solution of $\tau(X)$, then $\nu(u)$ is another solution of $\tau(X)$ such that $b(\nu(y)) \geq b(y)$;
- solution equivalence ρ and value equivalence r; let $S_P(x)$ be the set of solutions of an instance x of problem P, ρ is 1-1 onto function from $S_A(x)$ to $\nu(S_B(\tau(x)))$ such that $b(\rho(s)) = r(a(s), n)$.

Observation 8. *Assume that problem A has $(f(n), g(n))$ gap property and that there exists an Eq-reduction from A to B with the parameters described above. Then problem B has $(r(f(n), n), r(g(n), n))$ gap property.*

3.1 Reducing E3-LIN-2 to E2-LIN-2

We will describe the reduction of Håstad [H97], and later we will alter this reduction for our purposes.

Consider the following equation modulo 2: $x_0 + x_1 + x_2 + x_3 = 0$. We define the corresponding system of equations S as follows. We will use indices i, j with range 0,1,2,4;

original variables x_i, auxiliary variables a_i and 16 equations $x_i + a_j = b_{ij}$ where $b_{ij} = 1$ if $i = j$ and $b_{ij} = 0$ if $i \neq j$.

For a particular value of $x = (x_0, x_1, x_2, x_3)$ let $s(x)$ be the maximum, over different values of a, of the number of satisfied equations in S. Because of symmetries of S, $s(x)$ depends only on $|x| = \sum_{i=0}^{3} x_i$. Moreover, if we replace all variables with their negations, the set of satisfied equations does not change, thus it suffices to consider the cases when $|x| \leq 2$.

Case: $|x| = 0$, i.e. $x = (0, 0, 0, 0)$. We have to set $a_j = 0$ to satisfy equations $x_i + a_j = 0$ for $i \neq j$, which fails to satisfy $x_j + a_j = 1$. Thus $s(x) = 12$.

Case: $|x| = 1$, e.g. $x = (1, 0, 0, 0)$. We have to set $a_0 = 0$ which satisfies all equations $x_i + a_0 = b_{i0}$. The equations $x_i + a_1 = b_{i1}$ have the form $1 + a_1 = 0$, $0 + a_1 = 1$, $0 + a_1 = 0$ and $0 + a_1 = 0$, thus however we set a_1, two of them will be not satisfied. Because the same happens with a_2 and a_3, we have $s(x) = 16 - 3 \times 2 = 10$.

Case: $|x| = 2$, e.g. $x = (1, 1, 0, 0)$. We have to set $a_0 = 0$ which satisfies all equations that include a_0 except $x_1 + a_0 = 0$. One can see that each auxiliary variables is a similar situation, thus, like in case $|x| = 0$, we have $s(x) = 16 - 4 = 12$.

One can see that $s(x) = 12$ if $x_0 + x_1 + x_2 + x_3 = 0$ and otherwise $s_S(x) = 10$.

Now we can describe a reduction from E3-LIN-2 into 2-LIN-2. Consider a system E of n equations modulo 2 with 3 variables per equation. We define $\tau(E)$ by replacing, one by one, each equation in E. Given an equation $w + x + y = b$, we view it as $w + x + y + b = 0$, we create 4 new auxiliary variables and replace it with 16 equations as described above. Because one of the 4 original variables is actually a constant, we have 12 equations with 2 variables and 4 equations with 1 variable (which must be an auxiliary one), thus $t(n) = 16(n)$.

Let x be the vector of the variables of E and a be the vector of the auxiliary variables of $\tau(E)$. Given a value of (x, a) we can compute $\nu(x, a)$ by setting each a_i in such a way that a maximal number of equation is satisfied, if the two choices are equally good, we set $a_i = 0$. Because no equation involves two auxiliary variables, these value selections can be performed independently and they cannot conflict.

The solution equivalence is $\rho(x) = \nu(x, a')$, observe that $\nu(x, a')$ does not depend in a'. It is easy to see that the value equivalence is $r(k, n) = 10n + 2k$.

Value equivalence $10n + k$ translates $(n/2, n)$ gap property of E3-LIN-2 into $(10n + n, 10n + 2n) = (11n, 12n)$ gap property of 2-LIN-2; of we wish n to refer to the size of the new instance, i.e. $16n$, we got $(11/16\ n, 12/16\ n)$ gap.

Remark, The system $\tau(E)$ consists of equations that have 1 or 2 variables. We can define a similar reduction where we introduce a new variable z, and we first replace each equation $w + x + y = b$ with $w + x + y + z = b$ and then replace the new equation with a system of 16 equations as described above. We will use $\tau'(E)$ to denote the resulting system of equations with 2 variables each. This was the original reduction of Håstad [H97].

3.2 Hardness of E3-OCC-2-LIN-2 and E3-OCC-E2-LIN-2

The results of this sections follow from the existance of Eq-reductions that are described in the following lemma.

Lemma 9. *There exists an Eq-reduction R from E3-LIN-2 to E3-OCC-2-LIN-2 with value equivalence function* $119n+2k$ *and an Eq-reduction R' from E3-LIN-2 to E3-OCC-E2-LIN-2 with value equivalence* $150n+2k$.

Proof. Given a system of equations E we describe the instance transformation in seven steps.

(i) Replicate each equation n times (may be less).

(ii) For R' only: add z to each equation.

(iii) For each equation, form a four copies of each variable original that occurs and 4 auxiliary variables, each with 4 copies. Create 16 equations.

(iv) For a variable x that has m occurrences, create a 3-regular amplifier with $2m$ contacts. Every node in this amplifier is a variable, and each edge is an equation of the form $x_i + x_j = 0$.

(v) Connect m disjoint pairs of contact of the amplifier of x with chorded cycles made of 8 variables, as shown in Fig. 3.2, the edges inside the chorded cycles and that connect the cycles with the amplifiers again have the form $x_i + x_j = 0$. Four nodes of a such chorded cycle that still have only two neighbors form *a group* of copies of x.

(vi) For each equation of E form 4 auxiliary groups of variable, connect each group into a 2-regular strong expander (which happens to be a simple cycle).

(vii) Replace each equation of E with 16 equations as in τ (for R) or in τ' (for R'). In these equations, each variable occurs 4 times, replace each occurrence with a copy from the same group.

The solution normalization is described in four stages.

(i) In each amplifier/expander make all values equal to the value that is the majority among the contacts, this cannot decrease the number of satisfied equation by the very definition of an amplifier.

(ii) Consider a chorded cycle in which not all values are equal. (a) Suppose that out of 6 edges/equations that contact this cycle, at least 4 can be made true with the same value. Then we can convert entire cycle to this value, we will cease to satisfy at most 2 or the contacting equations and we will start satisfying at least 2 equations inside the cycle. (b) Suppose then that at most 5 of the contact equations are true. We can

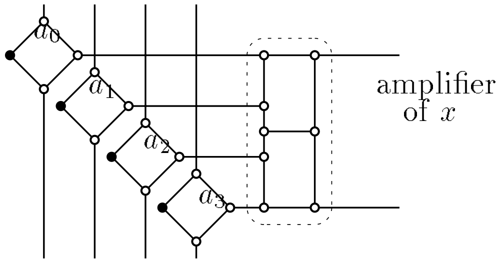

Figure 4: Part of the gadget replacing an equation with 3 variables. The other two variables also have their chorded cycles of 8 variables. Empty circles indicate variables, solid circles indicate equations with just 1 variable, edges indicate equations. We can add variable z to the original equation to eliminate the equations with 1 variable only

convert the cycle to a single value that satisfies at leat 3 of them, and thus we gain at least two equations inside the chorded cycle and loose at most two contact ones. (c) All contact equations are true, but three of them are made true with 0s, and three with 1s. If we convert the entire cycle to the value of the adjacent big amplifier we gain three equations inside the cycle. The latter follows from the fact that the chord does not separates zeros from ones the values in each amplifier are all equal.

(iii) Now each cycle and each amplifier is consistent. We normalize the values in the cycles/amplifiers of auxiliary variables as in the normalization of τ, to maximize the number of satisfied equations.

(iv) Suppose that a chorded cycle of an occurrence of a variable is inconsistent with the amplifier of this variable. We convert this cycle to be consistent, and renormalize the auxiliary variables. We gain 2 equations that form the contact of the cycle with the variable amplifier, and we loose at most 2 equations (among 16 equations that replaced an equation with 3 variables, we satisfy 10 or 12, so we could drop by at most 2).

The solution equivalence is simple: the value of x is given to all variables in its amplifier and in the chorded cycles of its occurrences, once this is done for every original variables, we compute the values of the auxiliary variables to maximize the number of satisfied equations.

It remains to calculate the value equivalence.

We started with E that had n^2 equations and $3n^2$ variable occurrences. In reduction R', we add z to each equations, which makes $4n^2$ variable occurrences.

For each equation, we made 16 equations, of which 12 are satisfied if the equation was satisfied, and otherwise only 10.

In these 16 equations, we have 16 occurrences of auxiliary variable that are connected into simple cycles, thus creating 16 satisfied equations.

An occurrence of an original variable has a chorded cycle with 9 equations, 2 equations connecting it with its amplifier. A wheel amplifier has 10 equations for each contact, so this occurrence needs 20. The total number of equations for an occurrence is $9+2+20=31$.

In Eq-reduction R, for each original equation we created $16+16+4+3\times 31+16=125$ equations. In a normalized solution that satisfies the original equation we satisfy $12+16+3\times 31=121$, and otherwise we satisfy two equations less. Thus the value equivalence is $r(k,n)=119n^n+2kn$.

In Eq-reduction R' we have need to add 31 satisfied equations, thus we produced $156n^2$ equations and the value equivalence is $r(k,n)=150n^2+2kn$.

❐

We conclude that $(n/2,n)$ gap property of E2-LIN-2 implies $(120/125n, 121/125n)$ gap property of E3-OCC-2-LIN-2. and $(151/156n, 152/156n)$ gap property of E3-OCC-2-LIN-2.

By using the same approach as in [BK99], we can extend the result for E3-OCC-E2-LIN-2 to an identical result for 3-MAX-CUT. Thus we can formulate this conclusion as follows.

Theorem 10. *For every $\varepsilon \in (0,1/302)$, it is NP hard to approximate E3-OCC-E2-LIN-2 and E3-MAX-CUT to within a factor $152/151-\varepsilon$ and to approximate E3-OCC-2-LIN-2 to within a factor of $121/120-\varepsilon$.*

3.3 Hardness of E3-OCC-MAX-2SAT

Theorem 11. *For every $\varepsilon \in (0,1/787)$, it is NP hard to approximate E3-OCC-MAX-2SAT to within a factor $788/787-\varepsilon$*

Proof. We will use a modification of Eq-reduction from E3-LIN-2 to E3-OCC-E2-LIN-2. The description of this Eq-reduction is simpler if we represent it as a composition of two reductions by introducing a special problem 1-E2-LIN-2-IM. In this problem we maximize the number of satisfied Boolean constraints in a given mixed set of constraints being linear equations mod 2 and implications, where each variable occurs in exactly 3 constraints and each constraint depends on exactly 2 variables; the following restriction is crucial: each variable occurs once in equation modulo 2, once as a left-hand-side of an implication and once as a righ-hand-side.

Lemma 12. *There exists an Eq reduction from 1-E2-LIN-2-IM to E3-OCC-MAX-2SAT with value equivalence $n+k$.*

Proof. We replace, one by one, each equation. Suppose that we have equation $x_1+y_1=b$ and implications $x_0\to x_1\to x_2$ and $y_0\to y_1\to y_2$. We describe the case of $b=1$ in detail, the case $b=0$ is similar.

The replacement clauses are $x_0\to x_1\to x_1^a\to x_2$, $y_0\to y_1\to y_1^a\to y_2$, $x_1\vee y_1$ and $\neg x_a^a\vee\neg y_1^a$ (in the case of $b=0$, the latter two clauses are $x_1^a\to y_1$ and $y_1^a\to x_1$). Here, the superscript a indicates the auxiliary copies.

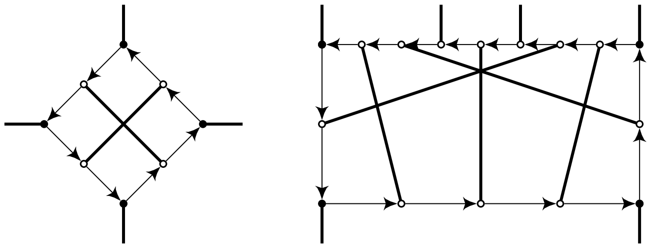

Figure 5: The implication/equation gadgets for a square of an auxiliary variable (on the right) and for a chorded cycles (on the left). Black dots indicate the variable/nodes that are taking part in the external equations, arrow indicate implications and thick lines indicate internal equalities.

The size translation is the following: 2/3 n clauses are implications and they are unchanged, and 1/3 n clauses are equations, and they are replaced with 4/3 n clauses, thus we get a system with $2n$ clauses.

The solution normalization also proceeds step by step (in reverse order). Without changing the values of other variable we will assure that $x_1 = x_1^a$ and $y_1 = y_1^a$. If $x_0 \neq x_2$ then we can set $x_1 = x_1^a$ to be either 0 or 1 without decreasing the number of satisfied implications in the chain of $x's$. If $x_0 = x_2$, we set $x_0 = x_1 = x_1^a$, and all implications in the chain are satisfied. We do the same for $y's$. If either x_1 or y_1 has freedom of choice, we can assure $x_1 \neq y_1$ and both $x_1 \vee y_1$ and $\neg x_a^a \vee \neg y_1^a$ are satisfied. If neither x_1 nor y_1 is free to choose, exactly one of the last two clauses is unsatisfied, and all other clauses are. It is easy to see that however we would alter the values of x's and y's, we would fail to satisfy one of the implications, so such alteration cannot be superior.

The solution equivalence is obvious—make each x^a equal to x. It is also ease to check that if we start with a solution that satisfied k clauses, we get a solution that satisfies $n + k$. ❒

Now consider an instance of E3-OCC-E2-LIN-2 that was obtained as $\tau'(S)$ in the proof of Lemma 9, where S is a system of n equations. For every equation of S with 3 variables, $\tau(S)$ contains

(i) 16 equations that correspond to the equations used by Håstad;

(ii) 4 4-tuples of equations that correspond to the auxiliary variables in the 16-tuple above;

(iii) 4 chorded cycles that contain 9 equations each;

(iv) 4 times 2 equations that connect a chorded cycle to an amplifier;

(v) 4 times 20 equations in the amplifier that correspond to occurrences of the original variables (and the variable that is used as zero).

We modify this plan as follows:

- each 4-tuple (a square) of an auxiliary variable is replaced with a cycle of 8 implications that has two equalities as chords (see Fig. 3.3, left);
- each chorded cycle is replaced with a cycle of 18 implications that has five equalities as chords (see Fig. 3.3, right);
- each node that does not belong to a square or a chorded cycle is replaced with a cycle of three implications.

Let us compute the modified parameter translation. We started with a system S of n equations, $\tau'(S)$ has $156n$ equations, then 4 times we eliminated 2 equations from a square and 4 equations from a chorded cycle, thus we eliminated $24n$ equations to get $132n$ equations. Because the new system of clauses has two implications for every equation, the total size is $396n$. The goal of the solution normalization is to assure that all implications are satisfied, i.e. that each implication cycle has one value only. Our approach is to consider the cycles one at the time, and if needed, modify the values on the cycle, but without modifying any variable values outside.

First, we make sure that all equations involving variables of the cycle are satisfied. When we change a cycle variable to do so, we can decrease the number of satisfied implications by at most 1, and we increase the number of satisfied equations.

Next, we convert all values on the cycle to the majority among contacts (variables that participate in the external equations) or to 0 if the value split is even. We need to show that the the minority among the contacts is not larger than the number of contiguous groups to which the minority belongs that the minority forms (the latter is the number of violated implications).

We start with cycles of length 3. The correctness is obvious because the minority, if any, has only one contact. Once we are done with these cycles, we use the properties of the amplifiers to make all values in each amplifier equal.

It is easy to see that a necessary condition for a minority of contacts to be bad — to belong to fewer contiguous groups than its size — we must have a contiguous group that contains at least 2 contacts of the minority. This reduces the analysis of a square to one case only: two adjacent corners and the non-contact that separates them have the same value; because the equality of that noncontact is satisfied, we have the minority value between the two elements of the majority, thus at least two contiguous groups.

When we normalize a chorded cycle, the contacts that participate in the equalities with the amplifier must have the same value, say 0, because the amplifier was normalized. If the entire bottom row has value 0, then the other contacts are pairwise separated by the variables with value 0, so we cannot have a bad set of contacts. This shows that if the minority is formed just by the two bottom contacts, there must be at least two minority groups. Thus we may assume that the bottom contacts are in the majority. Note that in the remaining case the minority must have at least two groups: one that contains their contacts, and one in the bottom row. So the minority can be bad only if it has 3 contacts and only one group outside the bottom row. However, in this case we have the third minority group: the variable between the bottom and the majority element at the top.

Now we can finish the normalization in the same manner as in reduction R'. One can see that the value equivalence is similar as in reduction R', except that we have fewer satisfied equalities in a normalized solution, fewer by $24n$, and we have more satisfied implications (there were none before), more by $264n$, so instead of $150n + 2k$ we got $390n + 2k$.

Our actual reduction composes this reduction with the one in Lemma 12, so we need to compose the two value equivalences, i.e. apply $n + k$ to $390n + 2k$. The new k, the number of satisfied clauses, is $390n + k$, and the new n, the number of all clauses, is $396n$. Therefore the overall value equivalence is $786n + 2k$, which translates $(1/2\ n, n)$ gap into $(787n, 788n)$ gap, or, after normalization, $(787/792\ n, 788/792\ n)$ gap.

❐

4 Open Problems

The 3-regular amplifiers were not studied extensively yet, and little is known about the least possible size of a 3-regular amplifier, i.e. the exact value of α_3. Any improvement below 7 would instantaneously improve lower bounds for numerous combinatorial optimization problems. The same question applies for d-regular amplifiers for $3 < d < 7$.

Acknowledgements

We thank Sanjeev Arora and Lars Engebretsen for stimulating remarks and discussions on the subject of this paper.

References

[A95] S. Arora, C. Lund, *Hardness of Approximations*, in *Approximation Algorithms*, D. S. Hochbaum (ed.), PWS Publishing, Boston 1997, 399-446.

[ACG+99] G. Ausiello, P. Crescenzi, G. Gambiosi, V. Kann, A. Marchetti-Spaccamela, M. Protasi, *Complexity and Approximation*, Springer, 1999.

[BK99] P. Berman and M. Karpinski, *On some tighter inapproximability results*, Proc. 26th ICALP, LNCS 1644, Springer, 1999, 200-209.

[H97] J. Håstad, *Some optimal inapproximability results*, Proc. 29th ACM STOC, 1-10, 1997.

[H00] J. Håstad, *On bounded occurrence constraint satisfaction*, Information Processing Letters **74**:1-6, 2000.

[K01] M. Karpinski, *Approximating bounded degree instances of NP-hard problems,* Proc. 13th Symp. on Fundamentals of Computation Theory, LNCS 2138, Springer, 2001, 24-34.

[LPS88] A. Lubotzky, R. Phillips and P. Sarnak, *Ramanujan graphs*, Combinatorica, 8:261:277, 1988.

[PY91] C. H. Papadimitriou and M. Yannakakis, *Optimization, approximation and complexity classes*, JCSS **43**, 1991, 425–440.

[P94] C. H. Papadimitriou, *Computational Complexity*, Addison-Wesley, New York, 1994, 315-319.

[T01] L. Trevisan, *Non-approximability results for optimization problems on bounded degree instances*, Proc. 33rd ACM STOC, 2001.