Electronic Colloquium on Computational Complexity, Report No. 34 (2001)

Polynomial Time Approximation Schemes for Dense Instances of Minimum Constraint Satisfaction

Cristina Bazgan* W. Fernandez de la Vega† Marek Karpinski‡

Abstract

It is known that large fragments of the class of *dense Minimum Constraint Satisfaction* (MIN-CSP) problems do not have *polynomial time approximation schemes* (PTASs) contrary to their Maximum Constraint Satisfaction analogs. In this paper we prove, somewhat surprisingly, that the minimum satisfaction of dense instances of kSAT-formulas, and linear equations mod 2, Ek-LIN2, do have PTASs for any k. The MIN-Ek-LIN2 problems are equivalent to the k-ary versions of the Nearest Codeword problem, the problem which is known to be exceedingly hard to approximate on general instances. The method of solution of the above problems depends on the developement of a new density sampling technique for k-uniform hypergraphs which could be of independent interest.

1 Introduction

In this paper we study approximability hardness of *dense instances* of *Minimum Constraint Satisfaction Problems* (MIN-CSP) connected to the minimum satisfiability of dense instances of kSAT-formulas, and linear equations mod 2 with exactly k variables per equation, Ek-LIN2. Somewhat surprisingly we prove existence of *polynomial time approximation schemes* (PTASs) for these two classes of problems. This should be contrasted with approximation hardness of a dual MIN-CSP problem of minimum satisfiability of dense 2DNF-formulas, the problem which is easily seen to be at least as hard to approximate as the dense Vertex Cover problem, the problem proven to be MAX-SNP-hard in [CT96],

*LAMSADE, Université Paris-Dauphine, 75775 Paris. Email:bazgan@lamsade.dauphine.fr

†LRI, Université Paris Sud, 91405 Orsay. Email:lalo@lri.fr

‡Dept. of Computer Science, University of Bonn, 53117 Bonn. Email:marek@cs.uni-bonn.de

ISSN 1433-8092

[KZ97]. It was also noticed by Luca Trevisan (personal communication) that one can easily densify arbitrary 2DNF-formulas by adding disjoint copies of original variables, and then adding all clauses having exactly one original and one copied variable, without changing the value of the optimum. In this context it is an interesting artifact that the dense and everywhere dense Maximum Constraint Satisfaction (MAX-CSP) analogs of the above problems are known to have PTASs (cf. [AKK95]). It is also not difficult to see that average-dense instances of MIN-CSP are approximation hard for the general instances.

The MIN-kSAT problems are known to be MAX-SNP-hard for all $k \geq 2$ [KKM94], and approximable within $2(1 - 1/2k)$ [BTV96]. Unlike the MIN-kSAT problems, MIN-Ek-LIN2 problems are exceedingly hard to approximate for all $k \geq 3$, they are known to be NP-hard to within a factor $n^{\Omega(1)/\log\log n}$ [ABSS93], [KST97], [DKS98], [DKRS00]. They are also easy to be seen equivalent to the k-ary versions of the Nearest Codeword problem (cf. [KST97], [BFK00]).

The special case of MIN-E2-LIN2 problem with all underlying equations being equal to 0, is equivalent to the MIN-UNCUT problem (cf. [KST97]) and known to be MAX-SNP-hard. The general MIN-E2-LIN2 is approximable to within a factor $O(\log n)$, cf. [GVY96]. It is also easily seen to be *approximation* (and *density*) preserving reducible to MIN-E3-LIN2, whereas obviously an opposite approximate reduction does not exist unless NP=P.

As mentioned before it is not difficult to see that the results of [AKK95], [F96], [FK96], [FK00], and [GGR96] on existence of PTASs for dense and average dense MAX-CSP problems cannot be applied for a large class of dense MIN-CSP problems. There were however some dense minimization problems, namely, dense BISECTION and MIN-k-CUT, identified in [AKK95] as having PTASs. Recently, the first *boolean* dense MIN-CSP problem, namely the problem of MIN EQUIVALENCE, was identified to have a PTAS [BF99]. This problem is also known as the MIN EQUIVALENCE DELETION problem, and was proven in [GVY96] to be MAX-SNP-hard, and approximable within a factor $O(\log n)$ on general instances. This problem is also clearly equivalent to the MIN-E2-LIN2 problem mentioned before. It has turned however out that the proof of the main result of [BF99] to the effect that the dense MIN-2SAT has a PTAS, based on the existence of a PTAS for dense MIN-E2-LIN2, contained an error. This was one of the starting points of this paper and the aim was to shed some light on approximation hardness of dense MIN-kSAT and dense MIN-Ek-LIN2 problems for arbitrary k.

In this paper (following [BFK00]) we design, somewhat surprisingly, the PTASs for both classes of Minimum Constraint Satisfaction, dense MIN-kSAT, and dense MIN-Ek-LIN2 problems for all $k's$.

The problems MIN-Ek-LIN2 are known to be hard to approximate for all $k \geq 3$ within a factor $n^{\Omega(1)/\log\log n}$ (cf. [KST97], [DKS98], [DKRS00]), and this hardness ratio is in fact also valid for average dense instances. Only recently a polynomial time algorithm with the first sublinear approximation ratio $O(n/logn)$ was designed for the general problem in [BK01]. Thus, the improvement in approximation ratio for the dense instances given by this paper seems to be the largest known for any NP-hard constraint satisfaction problem.

This paper extends the density sampler technique for graphs developed in [BFK00] to k-uniform hypergraphs for $k \geq 3$, as the main tool to attack the dense MIN-Ek-LIN2 problems, or equivalently, k-ary versions of the Nearest Codeword problems, and the dense MIN-EkSAT problems. The paper is organized as follows. In Section 2 we give the preliminaries and prove NP-hardness in exact setting of all the dense minimum satisfaction problems considered in this paper. Section 3 contains our main result on sampling k-uniform hypergraphs crucial for the rest of the paper. In Section 4, we design a PTAS for dense MIN-Ek-LIN2 and in Section 5 a PTAS for dense MIN-EkSAT for any k.

2 Preliminaries

We start with defining the *minimum constraint satisfaction* problems MIN-kSAT and MIN-Ek-LIN2 and give some other basic definitions.

MIN-kSat

Input: A set of m clauses $C_1, \ldots, C_m$ in boolean variables $x_1, \ldots, x_n$ with each C_j depending on at most k variables.

Output: An assignment that minimizes the number of clauses satisfied.

MIN-EkSat is the version of MIN-kSAT when each clause contains exactly k literals.

MIN-Ek-LIN2

Input: A set of m equations in n variables $x_1, x_2, ..., x_n$ over GF[2] where each equation has exactly k variables.

Output: An assignment to the variables that minimizes the number of satisfied equations.

Approximability. A *minimization problem* has a *polynomial time approximation scheme* (PTAS) if for every $\epsilon > 0$ there exists a polynomial time approximation algorihm computing for every instance x a solution y of value $m(x, y)$ such that $m(x, y) \leq (1+\epsilon)opt(x)$ for $opt(x)$ the value of an optimum solution.

Density. A family of instances of MIN-kSAT is δ-dense if for each variable, the total number of occurrences of the variable and its negation is at least δn^{k-1} in each instance. A family of instances of MIN-kSAT is *dense*, if there is a constant $\delta > 0$ such that this family is δ-dense.

An instance of MIN-kSAT is average δ-dense if the number of clauses is at least δn^k. An instance of MIN-kSAT is *average-dense* if there is a constant $\delta > 0$ such that the instance is average δ-dense.

A family of instances of MIN-Ek-LIN2 is δ-dense if for each variable x, the total number of occurrences of x is at least δn^{k-1} in each instance. A family of instances of MIN-Ek-LIN2 is *dense*, if there is a constant $\delta > 0$ such that the family is δ-dense.

DL-reductions. We call an L-reduction (cf. [PY91]) between problems P and Q *density preserving* (DL-) if it maps each dense instance of P into a dense instance of Q.

MIN-EkSAT for any $k \geq 2$ does not have PTAS on general instances [KKM94] under usual complexity theoretic assumptions but can be approximated in polynomial time within some constant factor [BTV96].

The following reduction from MIN-E2SAT can be used to prove that Dense MIN-E2SAT is NP-hard in exact setting. Given an instance F of MIN 2SAT with n variables $x_1, \ldots, x_n$ and m clauses $C_1, \ldots, C_m$, we define an instance F' of Dense MIN-2SAT as follows.
We add n new variables $y_1, \ldots, y_n$. F' will contain the clauses of F and the clauses $x_i \vee y_j$, $\bar{x}_i \vee y_j$, $1 \leq j \leq n$, $1 \leq i \leq n$. The total number of occurrences of x_i is at least $2n$ and the total number of occurrences of y_j is also at least $2n$. So, F' is a dense instance. Also, it is easy to see that $opt(F') = opt(F) + n^2$. A similar reduction shows that dense MIN-kSAT problems are NP-hard in exact setting for every $k \geq 2$.

We prove now NP-hardness (in exact setting) of Dense MIN-E2-LIN2, and in consequence also Dense MIN-Ek-LIN2 for every k. The reduction is from the general MIN-E2-LIN2 problem which is known to be MAX-SNP-hard [GVY96]. Given an instance I of MIN-E2-LIN2 on a set of variables $X = \{x_1, ..., x_n\}$ with m equations $x_i \oplus x_i = b$ with $b \in \{0, 1\}$, we construct an instance I' of Dense MIN-E2-LIN2 as follows. We extend the set of variables x by a disjoint set $Y = \{y_1, ..., y_n\}$. I' contains all equations of I, and all equations of the form $x_i \oplus y_j = 0$ and $x_i \oplus y_j = 1$ for all $1 \leq i, j \leq n$. Note that the instance I' is dense. Note also that exactly n^2 of the new added equations are satisfied independently of the values of the variables in X and Y. Thus, we have $opt(I') = opt(I) + n^2$. The similar construction can be used to prove that Dense MIN-Ek-LIN2 problems are NP-hard in exact setting for any k.

It is also not difficult to see that for the special case $k = 2$, MIN-E2-LIN2 (MIN EQUIVALENCE) is DL-reducible to MIN-E3-LIN2 (NEAREST CODEWORD). For suppose that an instance I of dense MIN-E2-LIN2 on a set of n variables $X = \{x_1, ..., x_n\}$ with m equations $x_i \oplus x_j = b$ is given. We construct an instance I' of Dense MIN-E2-LIN2 by extending the set of variables X by a disjoint set $Y = \{y_1, ..., y_n\}$, and extending the original set of m equations $x_i \oplus x_j = b$ by $mn + \binom{n}{3}$ new equations of the form $x_i \oplus x_j \oplus y_k = b$, $y_{l_1} \oplus y_{l_2} \oplus y_{l_3} = 1$. An optimum assignment for I' does have all $y's$ set to zero and defines an optimum assignment (for $x's$) for I. We have $opt(I) = opt(I')$.

Interestingly, it is also easy to show that both average-dense MIN-EkSAT and average-dense MIN-Ek-LIN2 problems are approximation hard for general instances. To see that it is enough to extend the set of variables by a new disjoint set $Y = \{y_1, ..., y_n\}$, and then add the set of all clauses $y_{i_1} \vee y_{i_2} \vee ... \vee y_{i_k}$, (respectively, equations $y_{i_1} \oplus y_{i_2} \oplus ... \oplus y_{i_k} = 1$.) The resulting instances are clearly average dense, and the optima are preserved in both cases (for all variables y_i assigned to 0).

3 Sampling k-uniform hypergraphs with bounded weights

As mentioned in Introduction, there are no approximation preserving reductions from MIN-Ek-LIN2 to MIN-E2-LIN2 for all $k \geq 3$, under usual complexity theoretic assumptions. Also, there are no known approximation and density preserving reductions from MIN-EkSAT problems to MIN-E2SAT. Therefore we prove our results by a generic method for arbitrary constant k. The straightforward generalization of our method for MIN-E3-LIN2 ([BFK00]) to higher k's does not work without leaving the structures of graphs. We need therefore a new sampling technique for k-uniform hypergraphs. This is due to the following observation. Let us consider MIN-EkSAT and let us denote by L_S the set of literals corresponding to the set of variables S. For the instances of MIN-EkSAT with "small" value of the optimum, a basic step in our method consists, for each assignment of truth values to the variables in a random sample S, in trying to set the truth value of each of the other variables so as to minimize the number of satisfied clauses within the clauses which contain this variable and $k-1$ literals from L_S.

For this scheme to be efficient, we need roughly the size of S to be $O(\log n)$ and also the number of clauses in the instance containing only literals from L_S and any fixed literal to be $\Omega(\log n/\epsilon^2\delta)$ for an accuracy requirement ϵ. This is achieved by the sampling procedures described below. Note that if we had only to sample a $(k-1)$-uniform hypergraph $\mathcal{H} = (X, \mathcal{E})$, we could use a much simpler procedure: namely pick uniformly at random elements from X^{k-1} and ask for each picked element whether or not it belongs to $\mathcal{E}$.

We need first the following inequality due to Hoeffding [H64].

Lemma 1 *Let $X_1, ..., X_m$ be independent random variables and each distributed as X_1. Let $\mu = E(X_1)$ and assume that X_1 satisfies $0 \leq X_1 \leq \Delta$. Let $S_m = \sum_{i=1}^m X_i$. Then, for every fixed $\gamma > 0$,*

$$\Pr(|S_m - \mu m| \geq \gamma\Delta m) \leq 2\exp(-2\gamma^2 m). \tag{1}$$

□

Let $k \geq 2$ be fixed. $\mathcal{H}_k$ will denote a k-uniform hypergraph with vertex set V, $|V| = n$, obtained from the complete hypergraph on V by assigning to each hyperedge $E = \{x_1, .., x_k\}$ a non-negative weight $w(x_1, ...x_k)$.

Suppose that $S_0, S_1, .., S_{k-1}$ are disjoint random samples picked from V all with the same size $m = \Omega(\log n/\epsilon^2)$. Let $S = S_0 \times S_1 ... \times S_{k-1}$. We denote by $\mathcal{H}(S)$ the sub-hypergraph of $\mathcal{H}_k$ which contains the edges of $\mathcal{H}_k$ with precisely one vertex in each of $S_0, S_1, ...S_{k-1}$. We denote by $w(\mathcal{H})$, (resp. $w(\mathcal{H}(S))$), the sum of the weights of the edges of $\mathcal{H}$, (resp. of

$\mathcal{H}(S)$). Our PTAS for the instances of MIN-Ek-LIN2 with "small" value is based on the following sampling theorem.

Theorem 1. *Let $m = \Omega(\log n/\epsilon^2)$ and let $\mathcal{H}_k$ have 0,1 weights. For any fixed $\epsilon > 0$, we have*

$$\Pr\left[\ \left|w(\mathcal{H}_k(S)) - \frac{m^k w(\mathcal{H}_k)}{\binom{n}{k}}\right| \leq \epsilon m^k\right] \geq 1 - o(1/n).$$

Proof of Theorem 1. We need the following lemma.

Lemma 2. *Let a $(k+1)$-uniform hypergraph $\mathcal{H}_{k+1}$ have weights bounded above by h and suppose that S_0 is a random sample of size m picked from $V = V(\mathcal{H}_{k+1})$ and define*

$$\Sigma_o = \sum_{y \in S_o} \sum_{A \in \binom{V \setminus S_o}{k}} w(A \cup \{y\}).$$

Then, for each fixed $\epsilon > 0$, sufficiently large n and $m = o(n^{1/2})$, we have that

$$\Pr\left[\ \left|\Sigma_o - \frac{m(k+1)}{n} w(\mathcal{H}_{k+1})\right| \leq \epsilon h \binom{n}{k}\right] \geq 1 - 3e^{-2m\epsilon^2}.$$

Proof. Clearly,

$$\begin{aligned}
\Sigma_o &= \sum_{y \in S_o} \sum_{A \in \binom{V \setminus \{y\}}{k}} w(A \cup \{y\}) - \sum_{y \in S_o} \sum_{A \in \binom{V \setminus \{y\}}{k} : A \cap S_o \neq \emptyset} w(A \cup \{y\}) \\
&= \sum_{y \in S_o} W_y - O\left(m^2 \binom{n}{k-1}\right)
\end{aligned}$$

where,

$$W_y = \sum_{A \in \binom{V \setminus \{y\}}{k}} w(A \cup \{y\}).$$

Let us write

$$\Sigma'_o = \sum_{y \in S_o} W_y.$$

Thus $\Sigma_o = \Sigma'_o - O(m^2 h n^{k-1})$. We have that

$$\max_{y \in V} W_y \leq h \binom{n-1}{k}$$

and

$$\mathbf{E}(\Sigma'_o) = \frac{m(k+1)}{n} w(\mathcal{H}_{k+1}).$$

Now Σ'_o is the sum of m terms randomly chosen within the $W_y, y \in V$. Lemma 3 gives

$$\Pr\left[\ \left|\Sigma'_o - \frac{m(k+1)}{n} w(\mathcal{H}_{k+1})\right| \leq \epsilon h \binom{n-1}{k}\right] \geq 1 - 2e^{-2m\epsilon^2}\ .$$

Lemma 2 follows.

□

For simplicity, we put now $T_k = w(\mathcal{H}_k(S))$. The hypergraph $\mathcal{H}_k$ will be defined in the context.

Lemma 3. *Let ℓ and h denote natural integers. Let $m = \Omega(\log n/\epsilon^2)$. Assume that*

$$\Pr\left[\ \left|\frac{T_\ell\binom{n}{\ell}}{m^\ell} - w(\mathcal{H}_\ell)\right| \le \epsilon\binom{n}{\ell}m^h\right] \ge 1 - o(1/n),$$

for any ℓ-uniform hypergraph $\mathcal{H}_\ell$ on n vertices with maximum weight at most m^h. We have then,

$$\Pr\left[\ \left|\frac{T_{\ell+1}\binom{n}{\ell+1}}{m^{\ell+1}} - w(\mathcal{H}_\ell + 1)\right| \le \epsilon'\binom{n}{\ell}m^{h-1}\right] \ge 1 - o(1/n),$$

for any $(\ell+1)$-uniform hypergraph $\mathcal{H}_{\ell+1}$ on n vertices with maximum weight at most m^{h-1} and where ϵ' is any constant greater than ϵ.

Proof. We have $\mathbf{E}(T_{\ell+1}) = \frac{m^{\ell+1}}{\binom{n}{\ell+1}}w(\mathcal{H}_{\ell+1})$ by simple counting. Thus we have to bound only from above the fluctuations of $T_{\ell+1}$. Clearly

$$T_{\ell+1} = \sum_{x_1\in S_1,\ldots x_\ell\in S_\ell} W(x_1,\ldots x_\ell)$$

where

$$W(x_1,\ldots x_\ell) = \sum_{y\in S_o} w(y,x_1,\ldots x_\ell).$$

Thus, we can estimate $T_{\ell+1}$ by sampling the ℓ-uniform hypergraph $\mathcal{K}$ with vertex set $V(\mathcal{H}_{\ell+1})\backslash S_o$, and where the edge $\{x_1,x_2,\ldots x_\ell\}$ has weight $W(x_1,x_2,\ldots x_\ell)$. Note that $\mathcal{K}$ has maximum weight at most m^h since $\mathcal{H}_{\ell+1}$ has maximum weight at most m^{h-1}. Thus the assumption of Lemma 3 reads

$$\Pr\left[\ \left|\frac{T_{\ell+1}\binom{n}{\ell}}{m^\ell} - w(\mathcal{K})\right| \le \epsilon\binom{n}{\ell}m^h\right] \ge 1 - o(1/n).$$

Using Lemma 1, we have that $w(\mathcal{K}) = \frac{m(\ell+1)}{n}w(\mathcal{H}_{\ell+1}) \pm \epsilon m^{h-1}\binom{n}{k}$ with probability $1-o(1/n)$ and thus, after multiplication by $\frac{n}{m(\ell+1)}$,

$$\Pr\left[\ \left|\frac{T_{\ell+1}\binom{n}{\ell+1}}{m^{\ell+1}} - w(\mathcal{H}_{\ell+1})\right| \le \epsilon(m^h + m^{h-1})\binom{n}{\ell+1}\right] \ge 1 - o(1/n),$$

implying

$$\Pr\left[\ \left|\frac{T_{\ell+1}\binom{n}{\ell+1}}{m^{\ell+1}} - w(\mathcal{H}_{\ell+1})\right| \le \epsilon' m^h\binom{n}{\ell+1}\right] \ge 1 - o(1/n),$$

which is the assertion of the lemma in which we can take in fact $\epsilon' = (1+o(1))\epsilon$.

□

In order to prove Theorem 1 for any fixed value of k, we just have to apply $k-1$ times Lemma 3, the starting assumption $\ell = 1, h = k-1$ being obtained by applying Lemma 2 to the sum of a sample of size m picked from a list of n terms each bounded above by m^k. We apply Lemma 3 first for $\ell = 1, h = k-1$, then for $\ell = 2, h = k-2$, an so on until $\ell = k-1, h = 1$. This gives after scaling the assertion of Theorem 1.

□

4 A PTAS for MIN-Ek-LIN2

Our techniques for designing PTASs for MIN-Ek-LIN2 and for MIN-EkSAT can be viewed as the new extensions of the technique of [AKK95]. In both cases, for each δ-dense instance of size n we run in parallel two distinct algorithms (Algorithm A and Algorithm B for MIN-Ek-LIN2, Algorithm 1 and Algorithm 2 for MIN-EkSAT), and we select the solution with the smallest value. Algorithm 1 and Algorithm A provide good approximations for the instances whose minimum value is "large" (the precise meaning of large will be specified later). These algorithms use the Smooth Integer Programming method of [AKK95]. Algorithms 2 and B provide good approximations for the instances whose optimum value is "small".

We assume now that the system of equations $\mathcal{S} = \{E_1, ..., E_m\}$ is a δ-dense instance of MIN-Ek-LIN2, on a set X of n variables $\{x_1, \dots, x_n\}$.

We run two distinct algorithms on S, Algorithm A and Algorithm B, and select the solution with the smallest value.

4.1 Algorithm A

Algorithm A formulates the problem as a Smooth Integer Program to degree k and uses a method of [AKK95]. This gives a PTAS for the instances whose optimum value is $\Omega(n^k)$. We refer to [BFK00] for an explicit construction of a smooth program for the case of $k = 3$.

4.2 Algorithm B

The algorithm B is guaranteed to give in polynomial time, as we will prove, approximation ratio $1+\epsilon$ for each fixed ϵ, whenever the optimum is at most αn^k for some fixed α, depending on ϵ and on δ.

Algorithm B

Input: Dense system $\mathcal{S}$ of linear equations in GF[2] over a set X of n variables with exactly k variables per equation.

1. Pick $k-1$ disjoint random samples $S_1, ..., S_{k-1} \subseteq X$ of size $m = \Theta\left(\log n / \epsilon^2 \delta\right)$. Let $S = \cup_{1 \le i \le k-1} S_i$.

2. For each possible assignment a, $y \to y^a$ of $\{0,1\}$ values to the variables in S do the following:

2.1. For each variable $x \notin S$ do the following:

Let $H^a_{x,0}$ and $H^a_{x,1}$ be the $(k-1)$-uniform hypergraphs with common vertex set $V(H^a_{x,0}) = V(H^a_{x,1}) = S$ and edge sets

$$\begin{aligned} E(H^a_{x,0}) &= \{\{x_1,...,x_{k-1}\} : \chi_{S_i}(x_i) = 1, 1 \le i \le k-1,\ x \oplus (\oplus_{1 \le i \le k-1} x_i) = b \in \mathcal{S} \\ &\wedge\ \oplus_{1 \le i \le k-1} x_i^a = b\} \end{aligned}$$

and

$$\begin{aligned} E(H^a_{x,1}) &= \{\{x_1,...,x_{k-1}\} : \chi_{S_i}(x_i) = 1, 1 \le i \le k-1,\ x \oplus (\oplus_{1 \le i \le k-1} x_i) = b \in \mathcal{S} \\ &\wedge\ \oplus_{1 \le i \le k-1} x_i^a = b\}. \end{aligned}$$

Let $m_0^a = |E(H^a_{x,0})|$, $m_1^a = |E(H^a_{x,1})|$.
If $m_0^a \ge \frac{2}{3}(m_0^a + m_1^a)$, then set x to 1.
If $m_1^a \ge \frac{2}{3}(m_0^a + m_1^a)$, then set x to 0.
Otherwise, set x to be *undefined.*

2.2. In this stage, we assign values to the variables which are undefined after the completion of stage 2.1. Let D^a be the set of variables assigned in stage 2.1, $U^a = S \cup D^a$ and let $V^a = X \setminus U^a$ denote the set of undefined variables. For each undefined variable y, let S_y denote the set of equations which contain y and whose $k-1$ other variables belong to U^a. Let k_0^a (resp. k_1^a) denote the number of equations in S_y satisfied by a and by setting y to 0 (resp. to 1).

If $k_0^a \le k_1^a$, then set y to 0. Else, set y to 1.

Let X^a denote the overall assignment produced at the end of this stage.

Among all the assignments X^a pick one which satisfies the minimum number of equations of $\mathcal{S}$.

Output this solution a_o.

4.3 Proof of correctness of algorithm B when the value of the instance is "small"

We assume, as we can, that a is the restriction to S of an optimal assignment $a^* \in \{0,1\}^n$. For each $y \in X$, we let y^{a^*} denote the value of y in a^*. Let $x \in X \setminus S$.

Let $H_{x,0}$ and $H_{x,1}$ be the hypergraphs with common vertex set $V(G_{x,0}) = V(G_{x,1}) = X$ and edge sets

$$\begin{aligned} E(H_{x,0}) = \{ &\ \{x_1,...,x_{k-1}\} : \chi_{S_i}(x_i) = 1, 1 \le i \le k-1,\ x \oplus (\oplus_{1 \le i \le k-1} x_i) = b \in \mathcal{S} \\ &\wedge\ \oplus_{1 \le i \le k-1} x_i^{a^*} = b\} \end{aligned}$$

and

$$E(H_{x,1}) = \{ \{x_1, ..., x_{k-1}\} : \chi_{S_i}(x_i) = 1, 1 \le i \le k-1,\ x \oplus (\oplus_{1 \le i \le k-1} x_i) = b \in \mathcal{S} \wedge \oplus_{1 \le i \le k-1} x_i^{a^*} = b\}.$$

Let $n_0^{a^*} = |E(G_{x,0})|, n_1^{a^*} = |E(G_{x,1})|$, $n^{a^*} = n_0^{a^*} + n_1^{a^*}$. Also, let $m^a = m_0^a + m_1^a$.

Lemma 4.1. *Assume that x is such that we have*

$$n_0^{a^*} \ge \frac{3(n_0^{a^*} + n_1^{a^*})}{4}.$$

Then, with probability $1 - o(1/n)$, x is assigned (correctly) to 1 in step 2.1 of Algorithm B.

Lemma 4.2. *Assume that x is such that we have*

$$n_1^{a^*} \ge \frac{3(n_0^{a^*} + n_1^{a^*})}{4}.$$

Then, with probability $1 - o(1/n)$, x is assigned (correctly) to 0 in step 2.1 of algorithm B.

Lemma 4.3. *With probability $1 - o(1/n)$, each fixed variable $y \in D^a$ is assigned to its correct value y^{a^*} by the Algorithm B.*

REMARK: The property in Lemma 4.1 holds simultaneously for all variables with probability $1 - o(1)$. The same is true for Lemmas 4.2 and for 4.3.

Proof of Lemmas 4.1 and 4.2. Let us first observe that $m_o^{a^*}$ (resp. $m_1^{a^*}$) is the number of equations in $\mathcal{S}$ containing x and which are satisfied by setting x to 0 (resp. to 1) and all other variables according to a^*. Thus if $m_o^{a^*} < m_1^{a^*}$, then we can assert that x is set to 0 in a^*. Similarly, if $m_o^{a^*} > m_1^{a^*}$, then we can assert that x is set to 1 in a^*.

We prove Lemma 4.1. The proof of Lemma 4.2. is similar to that of Lemma 4.2. Theorem 2 applied to the hypergraph $G_{x,0}$ with $d = \frac{n_0^{a^*}}{\binom{n}{k-1}}$ and the samples $S_1, ..., S_{k-1}$, gives

$$\Pr\left(m_0^a \ge (1-\epsilon)\frac{m^{k-1}}{\binom{n}{k-1}} n_0^{a*}\right) = 1 - o(1/n).$$

Let $m^a = m_0^a + m_1^a$. We apply now Theorem 2 to the union of the graphs $G_{x,0}$ and $G_{x,1}$. This gives

$$\Pr\left(m^a \le (1+\epsilon)\frac{m^{k-1}}{\binom{n}{k-1}} n^{a^*}\right) = 1 - o(1/n).$$

Substraction gives

$$\Pr\left(m_0^a - \frac{2m^a}{3} \ge \frac{m^{k-1}}{\binom{n}{k-1}}((1-\epsilon)n_o^{a^*} - (1+\epsilon)\frac{2(n_0^{a^*} + n_1^{a^*})}{3})\right) = 1 - o(1/n).$$

Using the inequality $n_0^{a^*} + n_1^{a^*} \le \frac{4n_0^{a^*}}{3}$, we obtain

$$\Pr\left(m_0^a - \frac{2m^a}{3} \ge \frac{2m^2}{n(n-1)} \frac{1-20\epsilon}{9} n_o^{a^*}\right) = 1 - o(1/n),$$

which implies

$$\Pr\left(m_0^a - \frac{2m^a}{3} \geq 0\right) = 1 - o(1/n),$$

if $\epsilon \leq 1/20$. This concludes the proof.

□

Proof of Lemma 4.3. Suppose that y is assigned to 1 in stage 2.1. The case where y is assigned to 0 is similar. We have to prove that $n_0^{a^*} \geq n_1^{a^*}$ with probability $1 - o(1/n)$ since if in an optimum solution $x_i = 1$ then $n_0^{a^*} \geq n_1^{a^*}$. Thus, Theorem 1 applied to the hypergraph $H_{x,0}$ and the samples $S_1, ..., S_{k-1}$ gives, with $\epsilon = 1/7$,

$$\Pr\left(m_0^a \leq \frac{8n_0^{a^*} m^{k-1}}{7\binom{n}{k-1}}\right) = 1 - o(1/n),$$

and so,

$$\Pr\left(n_0^{a^*} \geq \frac{7m_0^a\binom{n}{k-1}}{8m^{k-1}}\right) = 1 - o(1/n). \tag{2}$$

Theorem 1 applied to the union of the hypergraphs $H_{x,0}$ and $H_{x,1}$ with the samples $S_1, ..., S_{k-1}$ and $\epsilon = 1/9$, gives

$$\Pr\left(m^a \geq \frac{8n^{a^*} m^{k-1}}{9\binom{n}{k-1}}\right) = 1 - o(1/n),$$

and so,

$$\Pr\left(n^{a^*} \leq \frac{9m^a\binom{n}{k-1}}{4m^2}\right) = 1 - o(1/n). \tag{3}$$

Since y is assigned to 1 in stage 2.1, we have that $m_0^a \geq 2/3m^a$, implying with 2 and 3,

$$\Pr\left(\frac{n_0^{a^*}}{n^{a^*}} \geq \frac{14}{27}\right) = 1 - o(1/n).$$

Lemmas 4.3 follows.

□

The following lemma is crucial.

Lemma 5. *With probability 1-o(1), the number of variables undefined after the completion of stage 2.1 satisfies*

$$|V^a| \leq \frac{4 \text{ opt}}{\delta n^2}.$$

Proof. Assume that x is undefined. We have thus simultaneously $n_0^{a^*} < \frac{3}{4}(n_0^{a^*} + n_1^{a^*})$ and $n_1^{a^*} < \frac{3}{4}(n_0^{a^*} + n_1^{a^*})$ and so $n_1^{a^*} > \frac{1}{4}(n_0^{a^*} + n_1^{a^*})$ and $n_0^{a^*} > \frac{1}{4}(n_0^{a^*} + n_1^{a^*})$. Since x appears in at least δn^2 equations, $n_0^{a^*} + n_1^{a^*} \geq \delta n^2$. Thus,

$$opt \geq \min\{n_0^{a^*}, n_1^{a^*}\} \cdot |V^a| \geq \frac{\delta n^2}{4}|V^a|.$$

The assertion of the lemma follows. □

We can now complete the correctness proof. Let val denote the value of the solution given by our algorithm and let opt be the value of an optimum solution.

Theorem 2. *Let ϵ be fixed. If opt $\leq \alpha n^k$ where α is sufficiently small, then we have that val $\leq (1+\epsilon)$opt.*

Proof. Let us write

$$\mathrm{val} = \sum_{0 \leq i \leq k} \mathrm{val}_i$$

where val_i is the number of satisfied equations with exactly i variables in V^a.

With an obvious intended meaning, we write also

$$\mathrm{opt} = \sum_{0 \leq i \leq k} \mathrm{opt}_i$$

. We have clearly $\mathrm{val}_0 = \mathrm{opt}_0$ and $\mathrm{val}_1 \leq \mathrm{opt}_1$. Thus,

$$\begin{aligned}
\mathrm{val} &\leq \mathrm{opt} + \sum_{2 \leq i \leq k} (\mathrm{val}_i - \mathrm{opt}_i) \\
&\leq \mathrm{opt} + \sum_{2 \leq i \leq k} \mathrm{val}_i \\
&\leq \mathrm{opt} + \sum_{2 \leq i \leq k} \binom{|V^a|}{i} \binom{n}{k-i} \\
&\leq \mathrm{opt} + \sum_{2 \leq i \leq k} |V^a|^i n^{k-i} \\
&\leq \mathrm{opt} + (k-1)|V^a|^2 n^{k-2} \\
&\leq \mathrm{opt} + \frac{16(k-1)\mathrm{opt}^2}{\delta^2 n^k}
\end{aligned}$$

where we have used Lemma 5 for the last line. Thus,

$$\begin{aligned}
\mathrm{val} &\leq \mathrm{opt}\left(1 + \frac{16(k-1)\mathrm{opt}}{\delta^2 n^k}\right) \\
&\leq \mathrm{opt}(1+\epsilon)
\end{aligned}$$

if $\mathrm{opt} \leq \frac{\epsilon \delta^2 n^k}{16(k-1)}$.

□

It is known that the Algorithm A runs in polynomial time for any fixed $\epsilon > 0$ [AKK95], and the same is now easy to check for the Algorithm B on "small" instances. Thus we have, for any fixed k, a PTAS for MIN-Ek-LIN2.

5 Dense MIN-kSAT has a PTAS

In this section, we apply the technique of sampling k-uniform hypergraphs of Section 3 to obtain a PTAS for Dense MIN-EkSAT for each fixed k. As a side effect we give also PTAS for the general Dense MIN-kSAT.

Lemma 6. For any $k \geq 2$, Dense MIN-kSAT is DL-reducible to Dense MIN-EkSAT.

Proof. Let F be a δ-dense instance of MIN kSAT with n variables $x_1, \ldots, x_n$ and m clauses $C_1, \ldots, C_m$. We construct an instance F' of Dense MIN-EkSAT as follows: F' is built over the variables of F and a set Y of n new variables $y_1, \ldots, y_n$. For each clause of F, $\ell_1 \vee \ldots \vee \ell_t$, of length $t < k$, we put in F' the clause $\ell_1 \vee \ldots \vee \ell_t \vee y_1 \vee \ldots \vee y_{k-t}$. We also put in F' all the clauses of F of length k and all the clauses of length k with all variables in Y.

Let us justify that this is a DL-reduction.

It is easy to see that $opt(F') = opt(F)$. Now, given an optimal solution v of F', we can assume that each variable y takes the value zero in v, since otherwise we obtain a solution with a smaller value by assigning false to y. The assignment v satisfies in F $m(F, v) \leq m(F', v)$ clauses. Thus we have an L-reduction.

Since F is δ-dense the number of occurrences of the variable x_i and its negation for each $i = 1, \ldots, n$ is δn^{k-1}. Each variable y appears in F', $\Theta(n^{k-1})$ times. Thus F' is dense.

5.1 A PTAS for MIN-EkSAT

1. **Algorithm 1.** (Algorithm for the case of "large" instances)

For each fixed $k \geq 2$, we can formulate MIN-EkSAT as a degree k smooth integer program. We can then use again the approximation method of [AKK95]. Let us display such a smooth program for $k = 2$. For each clause C_i we construct a smooth polynomial P_i where

$$
\begin{aligned}
P_i &\equiv 1 - (1-x)(1-y) \text{ if } C_i = x \vee y \\
P_i &\equiv 1 - (1-x)y \text{ if } C_i = x \vee \bar{y} \\
P_i &\equiv 1 - x(1-y) \text{ if } C_i = \bar{x} \vee y \\
P_i &\equiv 1 - xy \text{ if } C_i = \bar{x} \vee \bar{y}
\end{aligned}
$$

Optimum solution of MIN-E2SAT corresponds now to the optimum solution of the following degree 2 smooth integer program:

$$
\begin{cases}
\min \sum_{j=1}^{m} P_j \\
x_i \in \{0,1\} \; 1 \leq i \leq n.
\end{cases}
$$

2. **Algorithm 2.** (Algorithm for the case of instances with a "small" value)

We need first some notation. Let F be a δ-dense instance of MIN-EkSAT, with m clauses over a set $X = \{x_1, \ldots, x_n\}$ of n variables. Let $\mathcal{S} = \{S_1, ..., S_{k-1}\}$ be a family of $k-1$ disjoint subsets of X. (Actually these sets will be random as defined in the algorithm below.) Let $S = \cup_{i=1}^{k-1} S_i$ and denote by L_i the set of literals corresponding to $S_i, 1 \leq i \leq k-1$. We denote by $C_{\mathcal{S}}$ the set of clauses of length $k-1$ obtained by picking a literal from each of the sets L_i. We write also, for a fixed assignment a of truth values to the variables in S,

$$C_{\mathcal{S},0} = \{C \in C_{\mathcal{S}} : C \text{ false under } a\}$$

and

$$C_{\mathcal{S},1} = \{C \in C_{\mathcal{S}} : C \text{ true under } a\}$$

Finally, we denote by C_1 (resp. C_0) the set of clauses of length $k-1$ which are true (resp. false) under an optimal assignment a^*.

For each variable $x \notin S$, we denote by $F_{\mathcal{S}}$ the set of clauses in F of the form $C \vee x$ or $C \vee \bar{x}$ for some clause $C \in C_{\mathcal{S}}$ and we define the numbers

- $u_1^a = |\{C \in C_{\mathcal{S},1} : C \vee x \in F_{\mathcal{S}}\}|, \quad u_1^{a^*} = |\{C \in C_1 : C \vee x \in F\}|,$
- $u_o^a = |\{C \in C_{\mathcal{S},0} : C \vee x \in F_{\mathcal{S}}\}|, \quad u_0^{a^*} = |\{C \in C_0 : C \vee x \in F\}|,$
- $v_1^a = |\{C \in C_{\mathcal{S},1} : C \vee \bar{x} \in F_{\mathcal{S}}\}|, \quad v_1^{a^*} = |\{C \in C_1 : C \vee \bar{x} \in F\}|,$
- $v_0^a = |\{C \in C_{\mathcal{S},0} : C \vee \bar{x} \in F_{\mathcal{S}}\}|, \quad v_0^{a^*} = |\{C \in C_0 : C \vee \bar{x} \in F\}|.$

Algorithm 2. (Algorithm for the case of "small" instances)

Input. A dense instance F of MIN-EkSAT over a set of variables X.

1. Pick $k-1$ random disjoint sets $S_1, ..., S_{k-1}$ each countaining $\ell = \Theta(\log n/\epsilon^2\delta)$ variables. Let $S = \cup_{i=1}^{k-1} S_i$.

2. For each possible assignment $a : S \to \{0,1\}$ of the variables of S do the following:

2.1. For each variable $x \in X \backslash S$ do the following with $u_1^a, u_0^a, ...$ as defined above:

If $u_1^a + u_0^a + v_1^a \leq (u_1^a + u_0^a + v_o^a + v_1^a)/8$, then set x to 1.

If $u_1^a + v_0^a + v_1^a \leq (u_1^a + u_0^a + v_o^a + v_1^a)/8$, then set x to 0.

Otherwise, set x to be *undefined.*

2.2. In this stage, we assign values to the variables which are undefined after the completion of stage 2.1. Let D^a be the set of variables assigned in stage 2.1, $U^a = S \cup D^a$ and let $V^a = X \setminus U^a$ denote the set of undefined variables. For each undefined variable y, let S_y denote the set of clauses which contain y or $\bar{y}$ and whose $k-1$ other literals belong to U^a. Let k_0^a (resp. k_1^a) denote the number of clauses in S_y satisfied by a and by setting y to 0 (resp. to 1).

If $k_0^a \leq k_1^a$, then set y to 0 and bias$(y) = k_1^a - k_0^a$. Else, set y to 1 and bias$(y) = k_0^a - k_1^a$.

Let a_x denote the overall assignment produced at the end of this stage.

Among all the assignments a_x pick one which satisfies the minimum number of clauses in F.
Output this solution a.

We denote by $B(F)$ the value of the solution given by the Algorithm 2, i.e., the number of clauses in F satisfied by the assignment a.

5.2 Proof of correctness of Algorithm 2 when the value of the instance is "small"

Lemma 7.1. *With probability $1 - o(1/n)$, each variable x with the property such that in an optimum assignment a^* of X which coincides with a on S we have that*

$$u_1^{a^*} + u_0^{a^*} + v_1^{a^*} \leq (u_1^{a^*} + u_0^{a^*} + v_1^{a^*} + v_o^{a^*})/8 \tag{4}$$

is assigned to 1 (as in a^*) in stage 2.1 of algorithm 2.

Lemma 7.2. *With probability $1 - o(1/n)$, each variable x with the property such that in an optimum assignment a^* of X which coincides with a on S we have that*

$$v_1^{a^*} + v_o^{a^*} + u_1^{a^*} \leq (u_1^{a^*} + u_0^{a^*} + v_1^{a^*} + v_o^{a^*})/8 \tag{5}$$

is assigned to 0 (as in a^*) in stage 2.1 of algorithm 2.

Lemma 7.3. *With probability $1 - o(1/n)$, each fixed variable x for which either 4 or 5 holds is assigned in a as in a^*.*

Note that the property in Lemma 7.1 holds simultaneously for all variables with probability $1 - o(1)$. The same is true for Lemmas 7.2 and 7.4.

Before turning to the proof of these lemmas, let us observe that $m_o = u_1^{a^*} + u_0^{a^*} + v_1^{a^*}$ is the number of clauses in F containing the variable x and which are satisfied by setting x to 0 (and the other variables according to a^*) and $m_1 = u_1^{a^*} + v_1^{a^*} + v_o^{a^*}$ is the number of clauses containing x and which are satisfied by setting x to 1. Thus, if $m_0 < m_1$, (that is if $u_o^{a^*} < v_o^{a^*}$,) then we can assert that x is set to 0 in a^*. Similarly, if $m_1 < m_0$, then we can assert that x is set to 1 in a^*. Observe also that $n_o = u_1^a + u_0^a + v_1^a$ is the number of clauses in $F_{\mathcal{S}}$ containing the variable x and which are satisfied by setting x to 0 (and the other variables according to a^*). Also $n_1 = u_1^a + v_1^a + v_o^a$ is the number of clauses in $F_{\mathcal{S}}$ containing x and which are satisfied by setting x to 1.

Proofs. We prove 7.1 and 7.3. The proof of 7.2 is similar to that of 7.1. and is omitted.

Proof of Lemma 7.1. Let x be a variable with the property that in the optimum solution $u_1^{a^*} + u_0^{a^*} + v_1^{a^*} \leq (u_1^{a^*} + u_0^{a^*} + v_1^{a^*} + \bar{u}_0^{a^*})/10$. Plainly, u_1^a, (resp. u_0^a, v_1^a, v_o^a), are obtained by sampling the hypergraphs with edge sets U_1^{a*}, (resp. $U_0^{a*}, V_1^{a*}, V_o^{a*}$), where $U_i^{a*} = \{C \in C_i : C \vee x \in F\}, i = 0, 1$, and $V_i^{a*} = \{C \in C_i : C \vee \bar{x} \in F\}, i = 0, 1$, (and vertex set the literals) in the sense of Theorems 1 and 2. We can thus apply Theorem 2 to these quantities. (Actually, for $k = 2$ we are sampling points rather than edges, but then we can use Hoeffding's inequality.) This gives for any fixed $\gamma > 0$,

$$\Pr(|u_1^a - \frac{u_1^{a^*}}{n}\ell| \le \gamma\ell) \ge 1 - n^{-\Omega(1)}. \tag{6}$$

Similarly we have,

$$\Pr(|u_o^a - \frac{u_0^{a^*}}{n}\ell| \le \gamma\ell) \ge 1 - n^{-\Omega(1)}, \tag{7}$$

$$\Pr(|v_1^a - \frac{v_1^{a^*}}{n}\ell| \le \gamma\ell) \ge 1 - n^{-\Omega(1)} \tag{8}$$

and

$$\Pr(|v_o^a - \frac{v_o^{a^*}}{n}\ell| \le \gamma\ell) \ge 1 - n^{-\Omega(1)}. \tag{9}$$

These inequalities imply clearly

$$u_1^a + u_o^a + v_1^a + v_o^a \ge \frac{\ell}{n}(u_1^{a^*} + u_0^{a^*} + v_1^{a^*} + v_o^{a^*}) - 4\gamma\ell$$

with probability $1 - n^{-\Omega(1)}$, and also

$$u_1^a + u_o^a + v_1^a \le \frac{\ell}{n}(u_1^{a^*} + u_0^{a^*} + v_1^{a^*}) + 3\gamma\ell.$$

So, again with probability $1 - n^{-\Omega(1)}$,

$$\begin{aligned} u_1^a + u_o^a + v_1^a - \frac{1}{8}(u_1^a + u_o^a + v_1^a + v_o^a) &\le -\frac{7\ell}{8n}(u_1^{a^*} + u_0^{a^*} + v_1^{a^*} + v_o^{a^*}) + \frac{7\gamma\ell}{2} \\ &\le -\frac{7\delta\ell}{8} + \frac{7\gamma\ell}{2} \end{aligned}$$

which is negative for $\gamma \le \frac{\delta}{4}$.

□

Proof of Lemma 7.3. Let us assume that x is assigned to 0 in a^* (the other case is similar). Thus we assume that that the inequaliy $u_0^{a^*} \ge v_o^{a^*}$ holds and we have to prove that the inequality

$$u_1^{a^*} + u_0^{a^*} + v_1^{a^*} \le (u_1^{a^*} + u_0^{a^*} + v_1^{a^*} + v_o^{a^*})/8$$

implies

$$u_1^a + u_0^a + v_1^a > (u_1^a + u_0^a + v_o^a + v_1^a)/8.$$

Using the inequalities (6)-(8), we have that, with probability $1 - n^{-\Omega(1)}$, for a fixed γ,

$$u_1^a + u_o^a + v_1^a \ge \frac{\ell}{n}(u_1^{a^*} + u_0^{a^*} + v_1^{a^*}) - 3\gamma\ell$$

and using the inequality (9),

$$\frac{v_o^a}{8} \le \frac{\ell}{8n}v_o^{a^*} + \frac{\gamma\ell}{8}.$$

By substraction, we get, again with probability $1 - n^{-\Omega(1)}$,

$$u_1^a + u_o^a + v_1^a - \frac{v_o^a}{8} \geq \frac{\ell}{n}(u_1^{a^*} + u_0^{a^*} + v_1^{a^*} - \frac{v_o^{a^*}}{8}) - 4\gamma\ell.$$

Since $u_0^{a^*} \geq v_o^{a^*}$ and $u_1^{a^*} + u_0^{a^*} + v_1^{a^*} + v_0^{a^*} \geq \delta n$, we have

$$u_1^{a^*} + v_1^{a^*} + u_0^{a^*} - \frac{1}{8}v_o^{a^*} > u_1^{a^*} + v_1^{a^*} + \frac{6}{8}u_0^{a^*} \geq \frac{3}{8}\delta n.$$

So, the difference

$$\frac{\ell}{n}(u_1^{a^*} + u_0^{a^*} + v_1^{a^*} - \frac{v_0^{a^*}}{8}) - 4\gamma\ell$$

is positive if $\gamma < \frac{3}{28}\delta$. Thus, with high probability if $\gamma < \frac{3}{28}\delta$, $u_1^{a^*} + u_o^{a^*} + v_1^{a^*} - \frac{v_o^{a^*}}{8} > 0$ which is what we want.

□

Lemma 8. *With probability* $1 - o(1)$*, the number of undefined variables satisfies*

$$|V^a| \leq \frac{8opt(F)}{\delta n}$$

.

Proof. If the variable x is undefined after stage 2.1 of the Algorithm 2, then from Lemma 4.2 with high probability we have

$$u_1^{a^*} + u_0^{a^*} + v_1^{a^*} \geq (u_1^{a^*} + u_0^{a^*} + v_1^{a^*} + v_o^{a^*})/8,$$

and

$$v_1^{a^*} + v_o^{a^*} + u_1^{a^*} \geq (u_1^{a^*} + u_0^{a^*} + v_1^{a^*} + v_o^{a^*})/8.$$

Since $u_1^{a^*} + u_0^{a^*} + v_1^{a^*} + v_o^{a^*} \geq \delta n$, the optimum value opt$(F)$ satisfies

$$\begin{aligned} opt(F) &\geq \min\{u_1^{a^*} + u_0^{a^*} + v_1^{a^*}, v_1^{a^*} + v_o^{a^*} + u_1^{a^*}\}|V^a| \\ &\geq |V^a|\delta n/8. \end{aligned}$$

Theorem 3. *If* $opt(F) < \alpha n^2$ *then with high probability* $B(F) \leq (1+\varepsilon)opt(F)$ *where* $\varepsilon = \frac{64\alpha}{\delta^2}$.

Proof. The proof of Theorem 3 via Lemma 8 is similar to the proof of Theorem 2 via Lemma 5. Therefore it is omitted here.

□

In order to prove now that the Algorithms 1 and 2 give a PTAS for MIN-kSAT, it only remains to observe that both algorithms run in polynomial time for each fixed ϵ.

Acknowledgements

The authors thank Madhu Sudan, Mario Szegedy, and Luca Trevisan for stimulating remarks and discussions.

References

[ABSS93] S. Arora, L. Babai, J. Stern and Z. Sweedyk, *The hardness of approximate optima in lattices, codes, and systems of linear equations*, Proc. of 34th IEEE FOCS, 1993, 724–733.

[AKK95] S. Arora, D. Karger and M. Karpinski, *Polynomial time approximation schemes for dense instances of NP-hard problems*, Proc. of 27th ACM STOC, 1995, 284–293; the full paper appeared in Journal of Computer and System Sciences 58(1999), 193–210.

[BK01] P. Berman and M. Karpinski, *Approximating Minimum Satisfiability of Linear Equations*, ECCC Technical Reports, TR 01-025, 2001.

[BF99] C. Bazgan and W. Fernandez de la Vega, *A Polynomial Time Approximation Scheme for Dense* MIN 2SAT, Proc. Fundamentals of Computation Theory, LNCS 1684, Springer, 1999, 91–99.

[BFK00] C. Bazgan, W. Fernandez de la Vega and M. Karpinski, *Approximability of Dense Instances of Nearest Codeword Problem,* ECCC Technical Report, TR00-91, 2000.

[BTV96] D. Bertsimas, C-P. Teo and R. Vohra, *On dependent randomized rounding algorithms*, Conference on Integer Programming and Combinatorial Optimization, LNCS 1084, Springer, 1996, 330–344.

[CT96] A.E.F. Clementi and L. Trevisan, *Improved non-approximability results for vertex cover with density constraints*, Proc. of 2nd Conference on Computing and Combinatorics, COCOON'96, Springer, 1996, 333–342.

[DKS98] I. Dinur, G. Kindler and S. Safra, *Approximating CVP to Within Almost Polynomial Factors is NP-Hard*, Proc. of 39th IEEE FOCS, 1998, 99–109.

[DKRS00] I. Dinur, G. Kindler, R. Raz and S. Safra, *An Improved Lower Bound for Approximating CVP*, 2000, submitted.

[F96] W. Fernandez de la Vega, *Max-Cut has a Randomized Approximation Scheme in Dense Graphs*, Random Structures and Algorithms, 8(3) (1996), 187–198.

[FK99] W. Fernandez de la Vega and M. Karpinski, *On approximation hardness of dense TSP and other path problems*, Information Processing Letters 70 (1999), 53–55.

[FK00] W. Fernandez de la Vega and M. Karpinski, *Polynomial Time Approximation of Dense Weighted Instances of MAX-CUT*, Random Structures and Algorithms **8** (2000), 314–332.

[FK96] A. Frieze and R. Kannan, *The Regularity Lemma and Approximation Schemes for Dense Problems*, Proc. of 37th IEEE FOCS, (1996), 12–20.

[GVY96] N. Garg, V. Vazirani and M. Yannakakis, *Approximate Max-Flow Min-(Multi)Cut Theorems and Their Applications*, SIAM J. on Computing **25** (1996), 235–251. Preliminary version in Proc. 25th ACMSTOC (1993).

[GGR96] O. Goldreich, S. Goldwasser and D. Ron, *Property Testing and its Connection to Learning and Approximation*, Proc. of 37th IEEE FOCS, 1996, 339–348. The full paper has appeared in Journal of the ACM, 45 (4) (1998), 653–750.

[H97] J. Hastad, *Some Optimal Inapproximability Results*, Proc. of 29th ACM STOC, 1997, 1–10.

[H64] W. Hoeffding, *Probability Inequalities for Sums of Bounded Random Variables*, Journal of the American Statistical Association, 58(301), 1964, 13–30.

[K97] M. Karpinski, *Polynomial Time Approximation Schemes for Some Dense Instances of NP-Hard Optimization Problems*, Randomization and Approximation Techniques in Computer Science, LNCS 1269, Springer, 1997, 1–14. The full paper will appears in Algorithmica (2001).

[KZ97] M. Karpinski and A. Zelikovsky, *Approximating Dense Cases of Covering Problems*, ECCC Technical Report TR 97-004, 1997, appeared also in DIMACS Series in Discrete Mathematics and Theoretical Computer Science, vol. 40, 1998, 169–178.

[KSW97] S. Khanna, M. Sudan and D. Williamson, *A Complete Classification of the Approximability of Maximization Problems Derived from Boolean Constraint Satisfaction*, Proc. of 29th ACM STOC, 1997, 11–20.

[KST97] S. Khanna, M. Sudan and L. Trevisan, *Constraint Satisfaction: the Approximability of Minimization Problems*, Proc. of 12th IEEE Computational Complexity, 1997, 282–296.

[KKM94] R. Kohli, R. Krishnamurti and P. Mirchandani, *The Minimum Satisfiability Problem*, SIAM Journal on Discrete Mathematics 7(1994), 275–283.

[PY91] C. Papadimitriou and M. Yannakakis, *Optimization, Approximation and Complexity Classes*, Journal of Computer and System Science 43 (1991), 425–440.

[TGSW96] L. Trevisan, G. Sorkin, M. Sudan, and D. Williamson, *Gadgets, Approximation and Linear Programming*, Proceedings of the 37th IEEE Symposium on the foundations of Computer Science, 617-626, 1996.

ECCC ISSN 1433-8092
http://www.eccc.uni-trier.de/eccc
ftp://ftp.eccc.uni-trier.de/pub/eccc
ftpmail@ftp.eccc.uni-trier.de, subject 'help eccc'