On Weighted vs Unweighted Versions of Combinatorial Optimization Problems*

Pierluigi Crescenzi†, Riccardo Silvestri‡, and Luca Trevisan§

Abstract

We investigate the approximability properties of several weighted problems, by comparing them with the respective unweighted problems. For an appropriate (and very general) definition of niceness, we show that if a nice weighted problem is hard to approximate within r, then its polynomially bounded weighted version is hard to approximate within $r - o(1)$. Then we turn our attention to specific problems, and we show that the unweighted versions of Min Vertex Cover, Min Sat, Max Cut, Max DiCut, Max 2Sat, and Max Exact kSat are exactly as hard to approximate as their weighted versions. We note in passing that Min Vertex Cover is exactly as hard to approximate as Min Sat. In order to prove the reductions for Max 2Sat, Max Cut, Max DiCut, and Max E3Sat we introduce the new notion of "mixing" set and we give an explicit construction of such sets. These reductions give new non-approximability results for these problems.

1 Introduction

One of the most important results obtained in the last decade in the field of computational complexity has been the so-called *PCP-theorem* [3, 4], that is, the fact that any language in NP admits a probabilistic verifier of membership proofs using a logarithmic number of random bits and a constant number of queries. Besides being *per se* interesting, this theorem led to many negative results in the study of the approximability properties of NP-hard optimization problems. For example, it was possible to prove that, unless P=NP, the Max 3Sat problem is not approximable within a factor $1 + \epsilon$ for a given constant $\epsilon > 0$. Since the first proof of the PCP-theorem, several other refined proofs of the same result appeared in the literature in order to improve the performance of the verifier with respect to different query complexity parameters [6, 15, 7, 5, 21, 22, 23]. The last of these, due to Bellare, Goldreich and Sudan [5], and to Håstad [23], allowed to significantly improve the lower bounds on the approximability of several important optimization problems, such as Min Vertex Cover, Max Cut, and Max 2Sat. For most of these problems, however, the lower bounds hold for their weighted version only. For instance, in the case of Max 2Sat, the repetition of clauses is exploited in order to show a 1.04 lower bound: clearly, the repetition of clauses is equivalent to considering a polynomially bounded weighted version of the problem.[1]

*Research partially supported by the MURST project *Efficienza di Algoritmi e Progetto di Strutture Informative*. An extended abstract has been presented at the *4th Israel Symposium on Theory of Computing and Systems*, 1996.

†Dipartimento di Sistemi e Informatica, Università di Firenze, Email: `piluc@dsi.unifi.it`

‡Dipartimento di Matematica Pura ed Applicata, Università di L'Aquila, Email: `silvestr@univaq.it`

§Department of Computer Science, Columbia University, Email: `luca@cs.columbia.edu`

[1] It is not always clear what is the "unweighted" version of a problem. For example, for the Max Cut it is natural to consider the unweighted version as defined for simple graphs (with no multiple edges). In the unweighted version

On the other hand, many approximation algorithms for optimization problems ensure a particular performance ratio only if weights are not allowed. In order to achieve the same performance ratio in the weighted case, algorithm developers had to consider different and often more complicated techniques. For instance, the 2-approximate algorithm for unweighted Min Vertex Cover is based on a simple greedy procedure to find a maximal matching in the graph [18]. In order to obtain the first 2-approximate algorithm for the weighted version, instead, a linear programming formulation of the problem has been used by Nemhauser and Trotter [30]. In some cases, it is not known whether the same performance ratio is obtainable. For instance, while the unweighted Max Clique problem is approximable within $O(n/\log^2 n)$ where n denotes the number of vertices in the graph [8], the best approximation algorithm for the weighted version of this problem reaches a factor of $O((\log\log n)^2 n/\log^2 n)$ [20].

Finally, it is well-known that several NP-hard optimization problems turn out to be tractable whenever a polynomial bound is imposed on the weights that appear in the instance [16]. For example, the Min Partition problem can be solved in time $O(nb)$ where n denotes the number of elements and b denotes the sum of their weights: clearly, if any upper bound were imposed on the weights in advance, even a polynomial function of n, this algorithm would be a polynomial-time algorithm for the restricted problem.

The goal of this paper is to study the relative complexity of the arbitrarily weighted version, the polynomially bounded weighted version[2], and the unweighted version of an optimization problem with respect to their approximability properties. The above three considerations may suggest that these three versions have different hardness of approximation. Surprisingly, we will see that for several interesting problems the approximation threshold is exactly the same for all three versions.

1.1 Related Results

Many optimization problems have been studied both in the weighted and in the unweighted version. A list of such results is contained in the compendium of NP optimization problems by Crescenzi and Kann [10]. Bellare, Goldreich and Sudan [5] present a reduction from the Max Cut problem in multi-graphs (or, equivalently, in weighted graph with a polynomial bound on the weights) to Max Cut in simple graphs. Unlike the reduction that we present in this paper, the reduction of Bellare et al. does not preserve the existence of approximation algorithms with the *same* performance ratio.

From a more structural point of view, we recall the following results.

In [31] Papadimitriou and Yannakakis introduce the the class Max NP and study its extension to weighted problems. They also show that the very same algorithmic technique can be used to approximate both the problems in Max NP and those in the weighted version of this class. More recently, Zimand [34] studied the approximability properties of logically defined weighted problems, allowing both positive and negative weights. It turns out that the hardness of approximation dramatically increases when negative weights are admitted.

Finally, in [12] Crescenzi and Trevisan proved that any arbitrarily weighted approximable problem is reducible to a polynomially bounded weighted approximable problem by means of an approximation-preserving reduction (this result has been used by Khanna, Motwani, Sudan and Vazirani [27] to prove the APX-completeness of Max 3Sat).

of Max Sat and of constraint satisfaction problems, instead, the repetition of clauses is typically allowed. However, when Max Cut for simple graphs is reformulated as a constraint satisfaction problem there is no repetition of clauses. To resolve such an ambiguity and to obtain the strongest results, we choose to define unweighted version of constraint satisfaction problems so that clauses are not repeated.

[2] That is, the version of the problem in which weights are polynomially bounded in the size of the input.

1.2 Our Results

The first result of this paper states that, for any weighted optimization problem satisfying a particular "niceness" property, *the approximation threshold of the unbounded version and that of a polynomially bounded version are equal*. Informally, an optimization problem is nice if the feasibility of a solution does not depend on its measure. Many important optimization problems are nice: for instance, MIN VERTEX COVER and all problems in MAX NP, such as MAX CUT and MAX SAT, are nice optimization problems.

Because of this result, we will restrict our attention on polynomially bounded weighted versions of nice optimization problems. In particular, we will consider the following well-known optimization problems: MIN VERTEX COVER, MIN SAT, MAX CUT, MAX DICUT, MAX 2SAT, and MAX EXACT kSAT. For all of them, we will be able to prove that *the approximation threshold of the weighted version is exactly the same as the approximation threshold of the unweighted version*.[3] The first two results will be obtained by showing quite simple approximation preserving reductions between the two versions of the problem (we note in passing that MIN VERTEX COVER is exactly as hard to approximate as MIN SAT). On the other hand, the result for MAX CUT and MAX DICUT are based on a more sophisticated reduction technique which is mainly based on the existence of graphs with a particular "mixing" property. The existence of these graphs is a more or less direct consequence of previous results obtained in the field of hash functions. As far as we know, this is the first time that dense expander graphs are used in order to obtain non-approximability results (indeed, Papadimitriou and Yannakakis [31] use sparse expanders to prove completeness results in the class MAX SNP). In order to extend our technique to MAX EXACT kSAT and other constraint satisfaction problems we introduce the new notion of "mixing" set and we give an explicit construction of such sets (we believe that this notion is *per se* interesting and may be applied to other derandomized constructions).

The above reductions imply improved non-approximability results for the unweighted versions of the MAX CUT and the MAX DICUT problems and give the first explicit non-approximability results for the unweighted versions of the MAX E3SAT and the MAX 2SAT problems.

1.3 Preliminaries

An *optimization problem* A consists of three objects: (1) the set I of instances, (2) for any instance $x \in I$, a set $sol(x)$ of solutions, and (3) for any instance $x \in I$ and for any solution $y \in sol(x)$, a measure $\mathsf{m}(x, y)$. The goal of an optimization problem is, given an instance x, to find an optimum solution y, that is, a solution whose measure is maximum or minimum depending on whether the problem is a maximization or a minimization one (see also [9]). In the following opt will denote the function that maps an instance x into the measure of an optimum solution.

Let A be an optimization problem. For any instance x and for any solution $y \in sol(x)$, the *performance ratio* of y with respect to x is defined as

$$R_A(x, y) = \max\left\{\frac{\mathsf{opt}(x)}{\mathsf{m}(x, y)}, \frac{\mathsf{m}(x, y)}{\mathsf{opt}(x)}\right\}.$$

Observe that the performance ratio is always a number greater than or equal to 1 and is as close to 1 as the solution is close to an optimum solution.

Let $r : \mathcal{N} \to [1, \infty)$. We say that an algorithm T for an optimization problem A is $r(n)$-*approximate* if, for any instance x of size n, the performance ratio of the feasible solution $T(x)$ with

[3] Our results do not apply directly to MAX 2SAT, but rather apply to its restricted version MAX E2SAT. However, Yannakakis [33] proves that the approximation threshold of the two problems is the same.

respect to x is at most $r(n)$. If a problem A admits an r-approximate polynomial-time algorithm for some constant $r \geq 1$, then we will say that A belongs to the *class APX*.

The *approximation threshold* of an optimization problem $A \in$ APX is a real number $r_A \geq 1$ such that, for any $\epsilon > 0$, A admits an $(r_A + \epsilon)$-approximate polynomial-time algorithm but A does not admit an $(r_A - \epsilon)$-approximate polynomial-time algorithm.

Let A and B be two optimization problems and let α be a positive constant. A is said to be *α-AP reducible* to B [11], in symbols $A \leq^{\alpha}_{\mathrm{AP}} B$, if two functions f and g exist such that:

1. For every $x \in I_A$ and for every $r > 1$, $f(x, r) \in I_B$.
2. For every $x \in I_A$, for every $r > 1$, and for every $y \in sol_B(f(x, r))$, $g(x, y, r) \in sol_A(x)$.
3. f and g are computable by two algorithms T_f and T_g, respectively, whose running time is polynomial for any fixed r.
4. For every $x \in I_A$, for every $r > 1$, and for every $y \in sol_B(f(x, r))$, $R_B(f(x, r), y) \leq r$ implies $R_A(x, g(x, y, r)) \leq 1 + \alpha(r - 1)$.

We also write $A \equiv^{\alpha}_{\mathrm{AP}} B$ when $A \leq^{\alpha}_{\mathrm{AP}} B$ and $B \leq^{\alpha}_{\mathrm{AP}} A$. Observe that if $A \leq^{\alpha}_{\mathrm{AP}} B$ for any $\alpha > 1$, then the approximation threshold of A is at most equal to the approximation threshold of B. In particular, if $A \equiv^{\alpha}_{\mathrm{AP}} B$ for any $\alpha > 1$, then the two problems have the same approximation threshold.

We denote by $\mathcal{N} = \{0, 1, 2, \ldots\}$ the set of natural numbers. For a positive integer n, we denote by $[n]$ the set $\{1, 2, \ldots, n\}$.

1.4 The Optimization Problems Studied in this Paper

The unweighted versions of the following problems can be obtained by simply imposing that $\omega(\cdot)$ is a constant function equal to 1.

We start defining *constraint satisfaction problems*. The definitions below were given implicitly by Papadimitriou and Yannakakis [31] and are explicit in the work of Khanna et al. [27]. Constraint satisfaction problems are generalizations of the standard Max Sat problem where clauses can be arbitrary applications of boolean functions to subsets of the variables.

Definition 1 *A (k-ary) constraint function is a boolean function $f : \{0, 1\}^k \rightarrow \{0, 1\}$.*

When it is applied to variables $x_1, \ldots, x_k$ (see the following definitions) the function f is thought of as imposing the constraint $f(x_1, \ldots, x_k) = 1$.

Definition 2 *A constraint family $\mathcal{F}$ is a finite collection of constraint functions. The arity of $\mathcal{F}$ is the maximum arity of the functions in $\mathcal{F}$.*

Definition 3 *A constraint C over a variable set $x_1, \ldots, x_n$ is a pair $C = (f, (i_1, \ldots, i_k))$ where $f : \{0, 1\}^k \rightarrow \{0, 1\}$ is a constraint function and $i_j \in [n]$ for $j \in [k]$. Variable x_j is said to occur in C if $j \in \{i_1, \ldots, i_k\}$. The constraint C is said to be satisfied by an assignment $a_1, \ldots, a_n$ to $x_1, \ldots, x_n$ if $C(a_1, \ldots, a_n) = f(a_{i_1}, \ldots, a_{i_k}) = 1$. We say that constraint C is from $\mathcal{F}$ if $f \in F$.*

Max Weight $\mathcal{F}$

For a function family $\mathcal{F}$, the *constraint satisfaction problem* Max Weight $\mathcal{F}$, is the optimization problem defined as follows:

INSTANCE: A collection $\phi = \{C_1, \dots, C_m\}$ of constraints from $\mathcal{F}$ over variable set $X = \{x_1, \dots, x_n\}$ and a weight function $\omega : \phi \to \mathcal{N}$.

SOLUTION: A truth assignment τ for the variables in X.

MEASURE: Total weight of the constraints satisfied by the truth assignment.

The problems MAX WEIGHT SAT, MAX WEIGHT kSAT, MAX WEIGHT EkSAT, MAX WEIGHT kCSP, are defined by specifying the set $\mathcal{F}$ in the previous definition.

- The MAX WEIGHT SAT problem is the MAX WEIGHT $\mathcal{F}$ problem where $\mathcal{F}$ is the set of boolean functions expressible as a disjunction of literals.
- The MAX WEIGHT kSAT (resp. MAX WEIGHT EkSAT) problem is the MAX WEIGHT $\mathcal{F}$ problem where $\mathcal{F}$ is the set of boolean functions expressible as a disjunction of at most (resp. exactly) k literals.
- The MAX WEIGHT kCSP problem is the MAX WEIGHT $\mathcal{F}$ problem where $\mathcal{F}$ is the set of at most k-ary boolean functions.

MIN WEIGHT VERTEX COVER

INSTANCE: A pair (G, ω) where $G = (V, E)$ is a graph and $\omega : V \to \mathcal{N}$.

SOLUTION: A vertex cover for G, i.e., a subset $V' \subseteq V$ such that, for each edge $(u, v) \in E$, at least one of u and v belongs to V'.

MEASURE: Sum of the weights of the vertices in the vertex cover.

MIN WEIGHT SAT

INSTANCE: A pair (C, ω) where C is a collection of disjunctive clauses of literals and $\omega : C \to \mathcal{N}$.

SOLUTION: A truth assignment for the variables in C.

MEASURE: Sum of the weights of the clauses satisfied by the truth assignment.

MAX WEIGHT CUT

INSTANCE: A pair (G, ω) where $G = (V, E)$ is a graph and $\omega : E \to \mathcal{N}$.

SOLUTION: A partition of V into disjoint sets V_1 and V_2.

MEASURE: Sum of the weights of the edges with one endpoint in V_1 and one endpoint in V_2.

MAX WEIGHT DICUT

INSTANCE: A pair (G, ω) where $G = (V, E)$ is a directed graph and $\omega : E \to \mathcal{N}$.

SOLUTION: A partition of V into disjoint sets V_1 and V_2.

MEASURE: Sum of the weights of the edges with the first endpoint in V_1 and the second endpoint in V_2.

Note that MAX WEIGHT CUT (resp. MAX WEIGHT DICUT) can be seen as MAX WEIGHT $\mathcal{F}$ with $\mathcal{F} = \{x_1 \neq x_2\}$ (resp. $\mathcal{F} = \{x_1 \wedge \neg x_2\}$).

2 Polynomially Bounded Weights vs. Unbounded Weights

In the following we will consider optimization problems whose goal is to find a set of objects satisfying a given property with the maximum (or minimum) weight sum. More formally, we say that an optimization problem A is a *subset problem* if:

- Any instance x of A is a triple (U, x', ω), where $U = \{u_1, \ldots, u_n\}$ is a set of objects, x' is a (possibly empty) string, and $\omega : U \to \mathcal{N}$ assigns a weight $\omega(u)$ to any object $u \in U$.
- With any solution $y \in sol(x)$ is associated a subset $S \subseteq U$, and the measure of y is equal to $\sum_{u \in S} \omega(u)$.

For example, consider the weighted version of the Max Sat problem. An instance is a triple (C, ϵ, ω) where C is the set of clauses and $\omega : C \to \mathcal{N}$ is the function that weights the clauses. A solution is a truth-assignment to the variables in C: with any solution we can then associate the subset $C' \subseteq C$ of satisfied clauses and the measure of the truth-assignment is the sum of the weights of the elements of C'. As another example, Min Weight Vertex Cover can be expressed as follows: An instance is a triple (V, E, ω) where V is the set of vertices, E is the set of edges, and $\omega : V \to \mathcal{N}$ is the function that weights the vertices. A feasible solution is a subset $V' \subseteq V$ that contains at least one endpoint for each edge in E, and the measure of V' is the sum of the weights of its elements.

A subset problem A is *nice* if, for any instance (U, x', ω), if y is a feasible solution, then, for any function $\omega' : U \to \mathcal{N}$, y is a feasible solution for (U, x', ω') as well. Roughly speaking, this property says that the definition of feasible solution in A is independent of the weights. It is easy to see that, for instance, Min Weight Vertex Cover, Max Weight Cut and Max Weight $\mathcal{F}$ (for any $\mathcal{F}$) are nice.

Let A be a subset problem and p be a polynomial. We denote by A^p the restriction of A to instances x such that the sum of the weights is at most $p(|x|)$.

The following result exploits the scaling techniques used by Ibarra and Kim [24] to develop a fully polynomial-time approximation scheme for the knapsack problem. Although the proof is only an application of very standard ideas, we give it for completeness.

Theorem 4 *Let A be a nice subset problem in* APX*. A polynomial p exists such that, for any $r > 1$, if A^p is r-approximable then A is $(r + 1/n)$-approximable, where n is the size of the input.*

Proof: We assume that A is a maximization problem (the proof for minimization problems is similar). Let A be r_0-approximable, let T be an r_0-approximate polynomial-time algorithm for A, and let $t(x) = \mathsf{m}(x, T(x))$. For any instance x, we have that

$$t(x) \leq \mathsf{opt}(x) \leq r_0 t(x).$$

Let $x = (U, x', \omega)$ be an instance of size n. Note that $|U| \leq n$. We define a new *scaled down* weight function $\tilde{\omega}$, such that, for any $u \in U$,

$$\tilde{\omega}(u) = \left\lfloor \frac{\omega(u) n^2}{t(x)} \right\rfloor.$$

Note that, since A is nice, the set of feasible solutions for (U, x', ω) is equal to that of $(U, x', \tilde{\omega})$. Moreover, we may assume without loss of generality that $\omega(u) \leq r_0 t(x)$ for any $u \in U$. It follows that $\tilde{\omega}(u) \leq r_0 n^2$ for any $u \in U$. Thus, $\tilde{x} = (U, x', \tilde{\omega})$ is an instance of $A^{r_0 n^3}$.

From the definition of $\tilde{\omega}$ it follows that, for any feasible solution y,

$$\mathsf{m}(x,y) \geq \mathsf{m}(\tilde{x},y)t(x)/n^2.$$

Moreover,

$$\begin{aligned}
\mathsf{m}(x,y) &= \sum_{u\in S}\omega(u) \leq \sum_{u\in S}\frac{t(x)}{n^2}\left\lceil\frac{\omega(u)n^2}{t(x)}\right\rceil \\
&\leq (t(x)/n^2)\sum_{u\in S}(\tilde{\omega}(u)+1) \\
&\leq (t(x)/n^2)(\mathsf{m}(\tilde{x},y)+n).
\end{aligned}$$

On the other hand, we may assume that $\mathsf{m}(x,y) \geq t(x)$, otherwise we replace y with the solution computed by the approximation algorithm T. Then,

$$\begin{aligned}
\frac{\mathsf{opt}(x)}{\mathsf{m}(x,y)} &\leq \frac{(t(x)/n^2)(\mathsf{opt}(\tilde{x})+n)}{\mathsf{m}(x,y)} \\
&\leq \frac{(t(x)/n^2)\mathsf{opt}(\tilde{x})}{(t(x)/n^2)\mathsf{m}(\tilde{x},y)} + \frac{t(x)}{n\cdot\mathsf{m}(x,y)} \\
&\leq \frac{\mathsf{opt}(\tilde{x})}{\mathsf{m}(\tilde{x},y)} + \frac{1}{n}.
\end{aligned}$$

Thus if y is an r-approximate solution for $\tilde{x}$, then y is an $(r+\frac{1}{n})$-approximate solution for x. The theorem follows. $\diamond$

Corollary 5 *Let A be a nice subset problem such that $A \in$ APX. Then a polynomial p exists such that A and A^p have the same approximability threshold.*

Remark 6 *Theorem 4 is stated for* APX*problems but it is easy to see that it can be extended to any nice subset problem that admits an n^c-approximate algorithm where c is a constant and n denotes the number of objects in the instance. For example, it is possible to prove that a polynomial p exists such that if* MAX WEIGHT CLIQUE^p *is approximable within $r(n)$ then* MAX WEIGHT CLIQUE *is approximable within $r(n)+1/n$, where n denotes the number of vertices of the input graph.*

3 Minimization Problems

The MIN VERTEX COVER and the MIN SAT problems are both nice ones. In this section, we will mainly deal with the unweighted and the polynomially bounded weighted version of these two problems.

Lemma 7 *For any polynomial p,* MIN WEIGHT VERTEX COVER^p $\leq^1_{\mathrm{AP}}$ MIN VERTEX COVER.

PROOF: Let $G=(V,E)$ be a graph and $\omega : V \to \mathcal{N}$ be a polynomially bounded weight function for its vertices. Let us define the graph $G^\omega = (V^\omega, E^\omega)$ as follows. For any vertex $u \in V$ such that $\omega(u) = w$, V^ω contains w distinct vertices $u^1,\ldots,u^w$. If $(u,v) \in E$ is an edge of G, then E^ω

contains all the edges (u^i, v^j) for $i = 1, \ldots, \omega(u)$ and $j = 1, \ldots, \omega(v)$. We note that, given G and ω, G^ω can be constructed in polynomial time.

Let C be a solution for (G, ω) whose measure is c. Then there exists a solution for G^ω of size c. Indeed, we can simply consider the set $C^\omega = \{u^i \in V^\omega : u \in C \wedge i = 1, \ldots, \omega(u)\}$.

On the other hand, let C be a solution for G^ω of size c. From C, we can recover in polynomial time a solution C' for (G, ω) whose measure is at most c. To this aim, assume that a vertex $u \in V$ and two indices $i, j \in \{1, \ldots, \omega(u)\}$ exist such that $u^i \in C$ and $u^j \notin C$. Then, $C - \{u^i\}$ is still a feasible solution for G^ω. Indeed, u^i and u^j have the same neighborhood, and since $u^j \notin C$ it follows that all the vertices that are adjacent to u^i belong to C and thus there is no need of taking u^i in the cover. By an easy induction argument, it follows that there exists a subset C^ω of C such that for any $u \in U$ either all its copies belong to C^ω, or none does. C^ω can be transformed into a solution C' for (G, ω) of measure $|C^\omega| \leq c$. $\diamond$

The following two lemmas can be proved by using arguments similar to the reduction between Max Clique and zero-free bit PCP given by Bellare, Goldreich, and Sudan [5] and the reduction between PCP and Max Clique given by Feige et al. [14], respectively. They have been independently proved by Marathe and Ravi [29].

Lemma 8 Min Vertex Cover $\leq^1_{\mathrm{AP}}$ Min Sat.

Proof: Let $G = (V, E)$ be a graph. Fix an arbitrary total order $\leq$ in V (e.g. lexicographic order). We define an instance of Min Sat as follows. There is a variable $x_{u,v}$ for any edge $(u, v) \in E$; there is a clause C_u for any vertex $u \in V$. C_u is defined as

$$C_u = \bigvee_{v:(u,v)\in E \wedge u \leq v} x_{u,v} \vee \bigvee_{v:(u,v)\in E \wedge u \geq v} \neg x_{u,v} \ .$$

Given an assigment τ for ϕ, $C = \{u : \tau \text{ satisfies } C_u\}$ is a vertex cover for G and has the same measure of τ. Indeed, if $(u, v) \in E$ then $x_{u,v}$ occurs in C_u and C_v once negated and once not, so at least one of the clauses is satisfied and so either $u \in C$ or $v \in C$.

Conversely, if C is a vertex cover then we define an assigment τ as follows. For any $u \notin C$ and any $(u, v) \in E$, $\tau(x_{u,v}) = \mathsf{true}$ if $u \geq v$, and $\tau(x_{u,v}) = \mathsf{false}$ if $u \leq v$. Since C is a vertex cover, this definition is not contradictory. All the other variables (if any) are set arbitrarily. τ contradicts all the clauses C_u such that $u \notin C$, and so its measure is at most the measure of C. $\diamond$

Lemma 9 *For any polynomial p,* Min Weight Sat^p $\leq^1_{\mathrm{AP}}$ Min Weight Vertex Cover^p.

Proof: Let $(C_1, \ldots, C_m), \omega$ be an instance of Min Weight Sat^p. We define a graph $G = (V, E)$ as follows. For any clause C_j of weight $w_j = \omega(C_j)$, we have a vertex u_j of weight w_j. Two vertices u_j and u_h are connected if there is a variable x_i such that x_i occurs in C_j and $\neg x_i$ occurs in C_h.

Given a vertex cover C for G, we define the assigment τ as follows: if x_i (resp. $\neg x_i$) occurs in C_j and $u_j \notin C$, then $\tau(x_i) = \mathsf{false}$ (resp. $\tau(x_i) = \mathsf{true}$). This is always possible since if x_i occurs in C_j and $\neg x_i$ occurs in C_h then x_j and x_h are adjacent in G and at most one of them can be out of C. All the other variables (if any) are set arbitrarily. τ contradicts all clauses C_j such that $u_j \notin C$, and so its measure is no larger than the measure of C.

Conversely, given an assignment τ, we have that $C = \{u_j : \tau \text{ satisfies } C_j\}$ is a vertex cover for G and has the same measure of τ (see the proof of the previous lemma). $\diamond$

Remark 10 *¿From the proof of the above lemmas it is also possible to show that* Min Sat *is equivalent to its restricted version where each variable occurs exactly twice, once positively and once negatively. Note that a similar result is not likely to hold for* Max Sat.

From the above three lemmas, the main result of this section follows.

Theorem 11 *The approximability threshold of the four problems* Min (Weight) Sat *and* Min (Weight) Vertex Cover *is the same.*

Proof: From Corollary 5 it follows that a polynomial p exists such that Min Weight Sat^p and Min Sat have the same approximability threshold and Min Weight Vertex Cover^p and Min Vertex Cover have the same approximability threshold. The above three lemmas also imply that Min Weight Sat^p $\equiv^1_{\text{AP}}$ Min Weight Vertex Cover^p $\equiv^1_{\text{AP}}$ Min Vertex Cover$\equiv^1_{\text{AP}}$ Min Sat. The theorem thus follows. $\diamond$

4 The Max Weight Cut Problem

In this section we show how to reduce r-approximating the Max Weight Cut^p problem to $(r - o(1))$-approximating the simple Max Cut problem, for any polynomial p. We first give a rough sketch of the reduction, and then we present it formally. Let $(G = (V, E), \omega)$ be a weighted graph; we define a graph $\hat{G} = (\hat{V}, \hat{E})$ in the following way: for each vertex $u \in V$ there are N vertices $u^1, \ldots, u^N$ in $\hat{V}$, and for each edge $(u, v) \in E$ of weight w, $\hat{E}$ contains the set of edges $\{(u^i, v^j) | (i, j) \in S^w\}$, where S^w is a random set of w elements of $[N] \times [N]$. Any cut of cost c in G clearly yields a cut of cost c in $\hat{G}$. Consider now a cut $\hat{C}$ in $\hat{G}$. For any $u \in V$, let p_u be the ratio of vertices $u^1, \ldots, u^N$ in $\hat{C}$, and let $q_u = 1 - p_u$. We assume that the sets S^w are picked at random after $\hat{C}$ is chosen. A calculation with Chernoff bounds would show that

$$\mathsf{m}(\hat{G}, \hat{C}) \approx \sum_{(u,v)\in E} \omega(u, v)(p_u q_v + p_v q_u).$$

Now define a random cut C in G such that the probability that $u \in V$ is equal to p_u (observe that this is also known as a random rounding). Clearly, the expected measure of this cut satisfies the following equality:

$$\begin{aligned} \mathbf{E}[\mathsf{m}(G, \omega, C)] &= \sum_{(u,v)\in E} \omega(u, v)(p_u q_v + p_v q_u) \\ &\approx \mathsf{m}(\hat{G}, \hat{C}). \end{aligned}$$

We will now be more formal and we will show how to *derandomize* the reduction. We first define a kind of graphs that will play the role of the random sets S^w. These graphs are related to the mixing properties of expander graphs (see for instance [19]).

Definition 12 *Let w be any positive integer and let ϵ and δ be any two positive real numbers. A bipartite graph $G = (V_1, V_2, E)$ is said to be (n, w, δ, ϵ)-mixing if $|V_1| = |V_2| = n$, and for any two subsets $A \subseteq V_1$, $B \subseteq V_2$ with at least δn vertices, the following holds*

$$\left| Cut(A, B) - w\frac{|A||B|}{n^2} \right| \leq \epsilon w \frac{|A||B|}{n^2}$$

where $Cut(A, B)$ denotes the number of edges between a vertex in A and a vertex in B.

We note in passing that a (n, w, δ, ϵ)-mixing graph has at least $(1-\epsilon)w$ and at most $(1+\epsilon)w$ edges.

It is well known that expander graphs have good mixing properties. Indeed, the proof of the following theorem (contained in the Appendix) is mainly based on the explicit construction of expanders in order to construct mixing graphs.

Theorem 13 *For any $\epsilon, \delta > 0$, two constants c and n_0 exist such that for any $n > n_0$ and for any $w \geq cn$, an (n, w, ϵ, δ)-mixing graph can be constructed in time polynomial in n.*

We are now ready to state and prove the main result of this section.

Theorem 14 *For any $\alpha > 1$,* Max Weight Cut $\leq_{\text{AP}}^{\alpha}$ Max Cut.

Proof: Let $(G = (V, E), \omega)$ be an instance of Max Weight Cut, let $r > 1$ be fixed. Without loss of generality (see Theorem 4) we can assume that a polynomial q exists such that $\omega(e) \leq q(|V|)$ for any edge $e \in E$.

Let ϵ, δ be such that $0 < \epsilon \leq 1/2$, $0 < \delta < 1/2$, and

$$\alpha \geq \frac{1}{r-1}\left(\frac{(1+\epsilon)}{(1-\epsilon)(1-12\delta)}r - 1\right) .$$

Let c, n_0 be the constants of Theorem 13 relative to ϵ and δ. We assume that there exists an integer $N > n_0$ such that, for any edge e, $cN \leq \omega(e) \leq N^2$ (otherwise we can multiply each weight by $c^2 n_0^2 w_{\max}$, where $w_{\max}$ is the maximum weight, and then set $N = cn_0 w_{\max}$).

We construct an unweighted graph $\hat{G} = (\hat{V}, \hat{E})$ as follows. For any $cN \leq w \leq N^2$ let $M_w = ([N], [N], E_w)$ be the (N, w, ϵ, δ)-mixing graph whose existence is guaranteed by Theorem 13. We let $\hat{V}$ contain N copies of any vertex of V, and, for any edge e of weight $\omega(e)$ of G, we let $\hat{E}$ contain the edges of $M_{\omega(e)}$. More formally,

$$\hat{V} = \{v^i \ : \ v \in V \text{ and } i = 1, \ldots, N\}$$

and

$$\hat{E} = \{(u^i, v^j) \ : \ (u, v) \in E \text{ and } (i, j) \in E_{\omega(u,v)}\}.$$

In order to show that it is a $\leq_{\text{AP}}^{\alpha}$-reduction, we prove that $\mathsf{opt}(G, \omega) \leq \mathsf{opt}(\hat{G})/(1-\epsilon)$ and that from any cut in $\hat{G}$ it is possibile to construct in polynomial time a cut in G whose measure is sufficiently large (w.r.t. the measure of the cut in $\hat{G}$).

Claim 15 *For any cut C in (G, ω) there exists a cut $\hat{C}$ in $\hat{G}$ such that*

$$\mathsf{m}(\hat{G}, \hat{C}) \geq (1-\epsilon)\mathsf{m}(G, \omega, C) .$$

Proof: [Of Claim 15] Let $\hat{C} = \{u^i \ : \ u \in C \text{ and } i \in [N]\}$. Then,

$$\begin{aligned} \mathsf{m}(\hat{G}, \hat{C}) &= \sum_{(u,v) \in E, |\{u,v\} \cap C| = 1} |E_{\omega(u,v)}| \\ &\geq \sum_{(u,v) \in E, |\{u,v\} \cap C| = 1} (1-\epsilon)\omega(u, v) \\ &= (1-\epsilon)\mathsf{m}(G, \omega, C). \end{aligned}$$

◇

Notice that from the above claim it derives that $\mathsf{opt}(G, \omega) \leq \mathsf{opt}(\hat{G})/(1-\epsilon)$.

Claim 16 *Given a cut $\hat{C}$ in $\hat{G}$, we can construct in polynomial time a cut C for G such that* $\mathsf{m}(G, \omega, C) \geq \mathsf{m}(\hat{G}, \hat{C})(1 - 12\delta)/(1 + \epsilon)$.

Proof: [Of Claim 16] We assume that $\hat{C}$ is a local optimum w.r.t. moving vertices in and out $\hat{C}$. Thus, at least half of all the edges of $\hat{G}$ are cut by $\hat{C}$. Since $\hat{G}$ has at least $\sum_{e \in E}(1 - \epsilon)\omega(e)$ and $\epsilon \leq 1/2$, it holds that

$$\mathsf{m}(\hat{G}, \hat{C}) \geq 1/2 \sum_{e \in E}(1 - \epsilon)\omega(e) \geq 1/4 \sum_{e \in E} \omega(e).$$

We first transform $\hat{C}$, by arbitrarily moving vertices, into a new cut $\hat{C}'$ such that, for any $u \in V$, at least δN copies of u belong to $\hat{C}'$ and at least δN copies belong to $\hat{V} - \hat{C}'$. Clearly $\mathsf{m}(\hat{G}, \hat{C}')$ can be smaller than $\mathsf{m}(\hat{G}, \hat{C})$. However, since $M_{\omega(u,v)}$ has the mixing property, the number of edges of $M_{\omega(u,v)}$ that are cut by $\hat{C}$ and that are not cut by $\hat{C}'$ is at most $2(1 + \epsilon)\delta\omega(u, v)$. It follows that

$$\begin{aligned}
\mathsf{m}(\hat{G}, \hat{C}') &\geq \mathsf{m}(\hat{G}, \hat{C}) - 2 \sum_{e \in E}(1 + \epsilon)\delta\omega(e) \\
&\geq \mathsf{m}(\hat{G}, \hat{C})(1 - 12\delta) \quad (\text{since } \epsilon \leq 1/2 \text{ and } \mathsf{m}(\hat{G}, \hat{C}) \geq 1/4 \textstyle\sum_{e \in E} \omega(e))
\end{aligned}$$

Let $p_u = |\{u^i : u^i \in \hat{C}'\}|/N$. An easy application of a folklore method allows us to show that it is possible to construct in polynomial time a cut C such that

$$\mathsf{m}(G, \omega, C) \geq \sum_{(u,v) \in E} \omega(u, v)(p_u(1 - p_v) + p_v(1 - p_u)).$$

For the sake of completeness we sketch the proof: consider the *random* cut where each vertex u is chosen independently and with probability p_u. The expected measure of such a cut is clearly equal to $\sum_{(u,v) \in E} \omega(u, v)(p_u(1 - p_v) + (1 - p_u)p_v)$. Using the method of conditional probabilities (see e.g. [2]), a cut with measure not smaller than this expectation can be constructed in polynomial time.

Consider now the measure of $\hat{C}'$: for any edge $(u, v) \in E$ we have to consider the edges connecting the Np_u copies of u in $\hat{C}'$ with the $N - Np_v$ copies of v not in $\hat{C}'$, plus the edges connecting the $N - Np_u$ copies of u not in $\hat{C}'$ with the Np_v copies of v in $\hat{C}'$. Recalling that such edges are chosen from a mixing graph, we have that

$$\begin{aligned}
\mathsf{m}(\hat{G}, \hat{C}') &\leq \sum_{(u,v) \in E} \left[\frac{(1 + \epsilon)\omega(u, v)}{N^2}(Np_u(N - Np_v)) + \frac{(1 + \epsilon)\omega(u, v)}{N^2}(Np_v(N - Np_u)) \right] \\
&\leq (1 + \epsilon)\mathsf{m}(G, \omega, C).
\end{aligned}$$

◇

From the above claims it follows that given a solution $\hat{C}$ for $\hat{G}$ whose performance ratio is at most r, we can construct in polynomial time a solution C whose performance ratio is

$$\begin{aligned}
\frac{\mathsf{opt}(G, \omega)}{\mathsf{m}(G, \omega, C)} &\leq \frac{\mathsf{opt}(\hat{G})/(1 - \epsilon)}{\mathsf{m}(\hat{G}, \hat{C})(1 - 12\delta)/(1 + \epsilon)} \quad (\text{from Claims 15 and 16}) \\
&\leq \frac{1 + \epsilon}{(1 - 12\delta)(1 - \epsilon)} r \leq 1 + \alpha(r - 1) \quad (\text{from the initial assumption on } \alpha).
\end{aligned}$$

We have thus shown that Max Weight Cut $\leq^{\alpha}_{\mathrm{AP}}$ Max Cut. ◇

Remark 17 *A similar argument can be applied to* Max DiCut *thus showing that* Max Weight DiCut $\leq^{\alpha}_{\mathrm{AP}}$ Max DiCut*, for any* $\alpha > 1$*.*

From the results of [5, 32, 23] we have that, for any $\gamma > 0$ it is NP-hard to approximate Max Weight Cut (resp. Max DiCut) within $17/16-\gamma$ (resp. $13/12-\gamma$). From the previous theorem we can extend these results to the unweighted versions.

Corollary 18 *For any* $\gamma > 0$*,* Max Cut *is not* $(17/16-\gamma)$*-approximable and* Max DiCut *is not* $(13/12-\gamma)$*-approximable unless* $\mathrm{P} = \mathrm{NP}$*.*

This improves over the non-approximability result by Bellare, Goldreich and Sudan (see [5], page 57).

5 Constraint Satisfaction Problems

In this section we prove that, virtually, every (unweighted) constraint satisfaction problem has the same approximation threshold of its weighted version. This applies to Max E2Sat, Max E3Sat, and Max 2CSP. Since our result is only stated for constraint satisfaction problems without unary constraints, it does not directly apply to Max 2Sat. However, it is clear that the approximation threshold of Max 2Sat is not greater than that of Max E2Sat.

Similar to the previous section, our result is a consequence of reductions from the weighted version to the unweighted version of constraint satisfaction problems. The basic structure of the reductions is essentially the same of Theorem 14. However, to cope with constraints of arity greater than 2, we need to introduce the notion of *mixing set* of tuples, a generalization of the mixing property of bipartite graphs.

Definition 19 *Let* $n > 0$*,* $k > 0$ *be integers; a set* $S \subseteq [n]^k$ *is* k*-ary* (n, w, ϵ, δ)*-mixing if for any* k *sets* $A_1, \ldots, A_k \subseteq [n]$*, each with at least* δn *elements, the following holds:*

$$\left| |S \cap A_1 \times \cdots \times A_k| - w\frac{|A_1|\cdots|A_k|}{n^k} \right| \leq \epsilon w \frac{|A_1|\cdots|A_k|}{n^k}.$$

For any fixed k and for sufficiently large values of n and w, k-ary mixing sets are efficiently constructible.

Theorem 20 *For any integer* $k \geq 2$ *and for any two rationals* $\epsilon, \delta > 0$*, two constants* n_0 *and* c *exist such that, for any* $n \geq n_0$ *and for any* w *with* $cn \leq w \leq n^k$*, a* k*-ary* (n, w, ϵ, δ)*-mixing set is constructible in time polynomial in* n*.*

Proof: We first prove the theorem assuming that $k = 2^h$ is a power of two. We proceed by induction on h. For $h = 1$ the theorem follows from Theorem 13. Assume that the theorem holds for $k = 2^h$ and consider the case $2k = 2^{h+1}$. Let $\epsilon, \delta > 0$ be arbitrary positive constants, let $\bar{\epsilon}, \bar{\delta}$ be greater than zero and such that

$$\frac{(1+\bar{\epsilon})^3}{(1-\bar{\epsilon})^2} - 1 \leq \epsilon \ , \ 1 - \frac{(1-\bar{\epsilon})^3}{(1+\bar{\epsilon})^2} \leq \epsilon \text{ and } \bar{\delta} \leq \frac{(1-\bar{\epsilon})}{1+\bar{\epsilon}}\delta^k.$$

The inductive hypothesis ensures the existence of constants c, n_0 such that for any $n, m > n_0$ and any $w > cm$, a 2^h-ary $(n, cn, \bar{\epsilon}, \delta)$-mixing set and a binary $(m, w, \bar{\epsilon}, \bar{\delta})$-mixing set exist. Let

$n \geq n_0$ and $w \geq c^2(1+\bar{\epsilon})n$ be integers, let S be a k-ary $(n, cn, \bar{\epsilon}, \delta)$-mixing set with m elements and let B be a binary $(m, w, \bar{\epsilon}, \bar{\delta})$-mixing set. Note that B exists since

$$w \geq c^2(1+\bar{\epsilon})n \geq cm.$$

We can fix an arbitrary (say, lexicographic) order among the tuples of S, that is a bijection between S and $[m]$. Using this bijection, we can see the pairs of B as $2k$-tuples over $[n]$. Under this mapping, we are going to show that B is a $2k$-ary (n, w, ϵ, δ)-mixing set.

Let $A_1, \ldots, A_{2k} \subseteq [n]$ be any family of $2k$ sets such that, for any $i = 1, \ldots, 2k$ $|A_i| \geq \delta n$. We define two sets T and U such that

$$T = S \cap (A_1 \times \cdots \times A_k) \text{ and } U = S \cap (A_{k+1} \times \cdots \times A_{2k}).$$

We first note that, from the fact that S is k-ary $(n, cn, \bar{\epsilon}, \delta)$-mixing set with m elements, it follows that

$$(1-\bar{\epsilon})cn \leq m \leq (1+\bar{\epsilon})cn,$$

that

$$(1-\bar{\epsilon})cn\frac{|A_1|\cdots|A_k|}{n^k} \leq |T| \leq (1+\bar{\epsilon})cn\frac{|A_1|\cdots|A_k|}{n^k},$$

and that

$$(1-\bar{\epsilon})cn\frac{|A_{k+1}|\cdots|A_{2k}|}{n^k} \leq |U| \leq (1+\bar{\epsilon})cn\frac{|A_{k+1}|\cdots|A_{2k}|}{n^k}.$$

We observe that $|T|, |U| \geq \bar{\delta}m$ and thus, from the mixing property of B, it follows that

$$(1-\bar{\epsilon})w\frac{|T||U|}{m^2} \leq |B \cap (T \times U)| \leq (1+\bar{\epsilon})w\frac{|T||U|}{m^2}.$$

Putting all the pieces together we get

$$\begin{aligned} \frac{(1-\bar{\epsilon})^3}{(1+\bar{\epsilon})^2}w\frac{|A_1|\cdots|A_{2k}|}{n^{2k}} &\leq |B \cap (A_1 \times \cdots \times A_{2k})| \\ &\leq \frac{(1+\bar{\epsilon})^3}{(1-\bar{\epsilon})^2}w\frac{|A_1|\cdots|A_{2k}|}{n^{2k}}. \end{aligned}$$

To conclude the proof, we note that from a k-ary (n, w, ϵ, δ)-mixing set S (where $w \leq n^{k-1}$) we can obtain a $(k-1)$-ary (n, w, ϵ, δ)-mixing set S' as follows

$$S' = \{(a_1, \ldots, a_{k-1}) : \exists a_{k+1}.(a_1, \ldots, a_{k-1}, a_k) \in S\}.$$

$\diamond$

Remark 21 *A simpler construction is possible for k-ary (n, w, ϵ, δ)-mixing sets if $w \geq cn^{k/2}$, where c is a constant depending on ϵ and δ. Assuming for simplicity that k is even, the idea is to consider a bipartite expander $G = ([n]^{k/2}, [n]^{k/2}, E)$ of degree c and with $n^{k/2}$ vertices on each side. If $((a_1, \ldots, a_{k/2}), (b_1, \ldots, b_{k/2})) \in E$ is an edge of G, then we let the k-tuple $(a_1, \ldots, a_{k/2}, b_1, \ldots, b_{k/2})$ be an element of our mixing set. Indeed, one can see the proof of Theorem 20 as an extension of this idea, where the two sets $k/2$-tuples of size $[n]^{k/2}$ are recursively replaced by smaller mixing $(k/2)$-ary sets.*

We are now in a position to state and prove our main result.

Theorem 22 *Let $\mathcal{F}$ be a constraint family without unary constraints. Then, for any $\alpha > 1$, it is the case that* Max Weight $\mathcal{F} \leq_{\mathrm{AP}}^{\alpha}$ Max $\mathcal{F}$.

Proof: The proof has the same structure of the proof of Theorem 14. In virtue of Theorem 4, we can restrict ourselves to Max Weight $\mathcal{F}^p$, for a suitable polynomial p. Let h be the largest arity of a constraint in $\mathcal{F}$. Let $r > 1$ be fixed. Let $\epsilon, \delta \in (0, 1/2)$ be such that

$$\alpha \geq \frac{1}{r-1}\left(\frac{1+\epsilon+2^h(1+\epsilon-(1-\epsilon)(1-\delta)^h)}{1-\epsilon}r - 1\right) .$$

Notice that such ϵ and δ exist, since the right-hand member of the above inequality tends to 1 when ϵ and δ tend to 0. Let c, n_0 be constants such that Theorem 20 holds for any $k = 2, \ldots, h$. Let $\phi = (\{C_1, \ldots, C_m\}, \omega)$ be an instance of Max Weight $\mathcal{F}^p$ over variable set $X = \{x_1, \ldots, x_n\}$. Let $w_j = \omega(C_j)$. We assume without loss of generality that an integer $N > n_0$ exists such that for all $i \in [m]$ it holds $cN \leq w_i \leq N^2$ (otherwise we can multiply each weight by $c^2 n_0^2 w_{\max}$, where $w_{\max}$ is the maximum weight, and then set $N = cn_0 w_{\max}$). For any $cN \leq w \leq N^2$ and any $k \in [h]$ we let S_k^w be a k-ary (N, w, ϵ, δ)-mixing set. We define an instance $\hat{\phi}$ of Max $\mathcal{F}$ as follows. The variable set of $\hat{\phi}$ has N copies of any variable of X:

$$\hat{X} = \{x_i^j : i \in [n] \ , \ j \in [N]\} .$$

For a k-ary constraint $C_j = f(x_{i_1}, \ldots, x_{i_k})$ (with $f \in \mathcal{F}$) of weight w_j we define a constraint set $\mathrm{red}(C_j)$ as follows:

$$\mathrm{red}(C_j) = \{f(x_{i_1}^{a_1}, \ldots, x_{i_k}^{a_k}) : (a_1, \ldots, a_k) \in S_k^{w_j}\} .$$

$\hat{\phi}$ is defined as the union of the $\mathrm{red}(C_j)$ sets

$$\hat{\phi} = \bigcup_{j \in [m]} \mathrm{red}(C_j) .$$

We now claim that any assignment for ϕ can be transformed into an assignment for $\hat{\phi}$ with roughly the same measure and vice versa.

Claim 23 *For any assignment τ for ϕ, an assignment $\hat{\tau}$ for $\hat{\phi}$ exists such that*

$$\mathsf{m}(\hat{\phi}, \hat{\tau}) \geq (1-\epsilon)\mathsf{m}(\phi, \tau) .$$

Proof: [Of Claim 23] We define $\hat{\tau}$ as follows: for any variable $x_i^j \in \hat{X}$, $\hat{\tau}(x_i^j) = \tau(x_i)$. We then have

$$\mathsf{m}(\hat{\phi}, \hat{\tau}) = \sum_j |\{C \in \mathrm{red}(C_j) : \hat{\tau} \text{ satisfies } C\}| = \sum_{j : \tau \text{ satisfies } C_j} |\mathrm{red}(C_j)| \geq (1-\epsilon)\mathsf{m}(\phi, \tau) .$$

Since $S_k^{w_j}$ is a k-ary $(N, w_j, \epsilon, \delta)$-mixing set, it holds $|\mathrm{red}(C_j)| = |S_k^{w_j}| \geq (1-\epsilon)w_j$. $\diamond$

Claim 24 *For any assignment $\hat{\tau}$ for $\hat{\phi}$, an assignment τ for ϕ can be computed in polynomial time such that*

$$\mathsf{m}(\phi, \tau) \geq \frac{1}{(1+\epsilon)(1+2^h(1-(1-\delta)^h))}\mathsf{m}(\hat{\phi}, \hat{\tau}) .$$

Proof: [Of Claim 24] Let $w_{\text{tot}} = \sum_j w_j$. To begin with, we transform $\hat{\tau}$ into an assignment $\hat{\tau}'$ such that for any variable $x_i \in X$ at least δN and at most $(1-\delta)N$ copies of x_i are set to true by $\hat{\tau}'$. To do that, for any $x_i \in X$ that does not fulfill the requested condition, we switch the value of some of its copies. Note that, for any $x_i \in X$, the value of at most δN of its copies is switched. The difference between the measure of $\hat{\tau}$ and that of $\hat{\tau}'$ is bounded by the number of constraints where a switched variable occur. For each $j \in [m]$ we can prove (using the definition of mixing set and of $\text{red}(C_j)$) that the number of constraints in $\text{red}(C_j)$ that contain a switched variable is at most $(1+\epsilon)w_j - (1-\epsilon)(1-\delta)^k w_j$, where k is the arity of C_j.

Summing over all $j \in [m]$ we derive that

$$\mathsf{m}(\hat{\phi}, \hat{\tau}) - \mathsf{m}(\hat{\phi}, \hat{\tau}') \leq w_{\text{tot}}(1 + \epsilon - (1-\epsilon)(1-\delta)^h) \ .$$

For any variable $x_i \in X$, let

$$p_i = \frac{|\{j : \hat{\tau}'(x_i^j) = \mathsf{true}\}|}{N}$$

be the fraction of copies of x_i whose value is set to true by $\hat{\tau}'$. We define τ_R as a *random assignment* (more formally, a distribution over all the assignments) of X in the following way: $\mathbf{Pr}[\tau_R(x_i) = \mathsf{true}] = p_i$. A way of looking at this random assignment is the following: for any $i \in [n]$, a random value $j \in [N]$ is chosen uniformly at random and then the value $\tau_R(x_i)$ is set equal to $\hat{\tau}'(x_i^j)$. For a k-ary constraint $C_j = f(x_{i_1}, \ldots, x_{i_k})$ of ϕ of weight w_j, let $SAT_j \subseteq \{\mathsf{true}, \mathsf{false}\}^k$ be the set of *satisfying assignments* to the variables of C_j. From the mixing property of the set $S_k^{w_j}$ used to define $\text{red}(C_j)$ it is easy to derive

$$\begin{aligned}
\mathbf{Pr}[\tau_R \text{ satisfies } C_j] &= \sum_{(b_1,\ldots,b_k)\in SAT_j} \mathbf{Pr}[\tau_R(x_{i_1}) = b_1, \ldots, \tau_R(x_{i_k}) = b_k] \\
&= \sum_{(b_1,\ldots,b_k)\in SAT_j} \frac{|\{(a_1,\ldots,a_k) \in [N]^k : \hat{\tau}'(x_{i_1}^{a_1}) = b_1, \ldots, \hat{\tau}'(x_{i_k}^{a_k}) = b_k\}|}{N^k} \\
&\geq \sum_{(b_1,\ldots,b_k)\in SAT_j} \frac{|\{(a_1,\ldots,a_k) \in S_k^w : \hat{\tau}'(x_{i_1}^{a_1}) = b_1, \ldots, \hat{\tau}'(x_{i_k}^{a_k}) = b_k\}|}{(1+\epsilon)w_j} \\
&\geq \frac{|\{C \in \text{red}(C_j) : \hat{\tau}' \text{ satisfies } C\}|}{(1+\epsilon)w_j} \ .
\end{aligned}$$

And thus we have that

$$\mathbf{E}[\mathsf{m}(\phi, \tau_R)] \geq \mathsf{m}(\hat{\phi}, \hat{\tau}')/(1+\epsilon) \ .$$

Using the method of conditional expectation (see [2]), given a probability distribution over the variable of ϕ of average measure m_R it is possible to find in polynomial time an assignment of measure at least m_R. Note that the random assignment such that each variable is true or false with probability $1/2$ has avergae measure at least $2^{-h} w_{\text{tot}}$. It follows that we can find in polynomial time an assignment τ whose measure is at least the average measure of τ_R and at least $2^{-h} w_{\text{tot}}$. We now compare the measure of τ with that of $\hat{\tau}$

$$\mathsf{m}(\phi, \tau) \geq \frac{1}{1+\epsilon}(\mathsf{m}(\hat{\phi}, \hat{\tau}) - w_{\text{tot}}(1+\epsilon-(1-\epsilon)(1-\delta)^h)) \geq \frac{1}{1+\epsilon}(\mathsf{m}(\hat{\phi}, \hat{\tau}) - 2^h \mathsf{m}(\phi, \tau)(1+\epsilon-(1-\epsilon)(1-\delta)^h))$$

$$\mathsf{m}(\phi,\tau) \geq \frac{1}{(1+\epsilon+2^h(1+\epsilon-(1-\epsilon)(1-\delta)^h))}\mathsf{m}(\hat{\phi},\hat{\tau})$$

$\diamond$

From the above claims we have that given a solution $\hat{\tau}$ for $\hat{\phi}$ whose performance ratio is at most r, we can construct in polynomial time a solution τ for ϕ whose performance ratio is at most

$$\begin{aligned}\frac{\mathsf{opt}(\phi)}{\mathsf{m}(\phi,\tau)} &\leq \frac{\mathsf{opt}(\hat{\phi})/(1-\epsilon)}{\mathsf{m}(\hat{\phi},\hat{\tau})/(1+\epsilon+2^h(1+\epsilon-(1-\epsilon)(1-\delta)^h))} \\ &\leq \frac{(1+\epsilon+2^h(1+\epsilon-(1-\epsilon)(1-\delta)^h))}{1-\epsilon} r \leq 1+\alpha(r-1) \ .\end{aligned}$$

We have thus shown that Max Weight $\mathcal{F} \leq^{\alpha}_{\mathrm{AP}}$ Max $\mathcal{F}$. $\diamond$

The following non-approximability results are proved in [5, 32, 23].

Theorem 25 *The following holds for any $\gamma > 0$ (unless $\mathrm{P} = \mathrm{NP}$):*

- Max Weight E2Sat *is not* $(22/21-\gamma)$*-approximable;*
- Max Weight E3Sat *is not* $(8/7-\gamma)$*-approximable;*
- Max Weight 2CSP *is not* $(10/9-\gamma)$*-approximable.*

From Theorem 22, all the previous results also hold for the unweighted versions of the mentioned problems. No explicit non-approximability result was known for the unweighted versions before.

6 Conclusions

We studied the approximability properties of several weighted problems, by comparing them with the respective unweighted versions. For a very general and natural class of weighted problems, we showed that if a problem in the class is hard to approximate within r, then its polynomially bounded weighted version is hard to approximate within $r - o(1)$. Then we considered specific problems, and we showed that the unweighted versions of Min Vertex Cover, Min Sat, Max Cut, Max 2Sat, and Max EkSat are exactly as hard to approximate as their weighted versions. The reductions for Max Cut implied an improved non-approximability result for the problem, and the reductions for Max E3Sat and Max 2Sat gave the first explicit non-approximability results for these problems.

Our results hold for any constraint satisfaction problem without unary constraints, such as the d-Regular Hypergraph Trasversal studied by Alimonti [1] and by Kann, Lagergren and Panconesi [26] and the Max k Cut (also known as Max k-Colorable Subgraph) studied by Papadimitriou and Yannakakis [31] and by Kann, Khanna, Lagergren and Panconesi [25].

We believe that our reduction can be extended to the general Max Sat problem. This should require the construction of k-ary (n, w, ϵ, δ)-mixing sets with $k = n^{\Omega(1)}$ and a 1-AP reduction from the general Max Sat problem to the restriction of Max Sat to formulas without unary constraints. The latter step seems quite difficult.

Acknowledgements

We wish to thank Viggo Kann for having read a preliminary version of this paper and for suggesting some useful modifications.

References

[1] P. Alimonti Non-oblivious local search for graphs and hypergraphs coloring problems. In *Proc. 21st Workshop on Graph-Theoretic Concepts in Computer Science*, 167–180, LNCS 1017, Springer-Verlag, 1995.

[2] N. Alon and J. Spencer. *The Probabilistic Method*. Wiley Interscience, 1992.

[3] S. Arora, C. Lund, R. Motwani, M. Sudan, and M. Szegedy. Proof verification and hardness of approximation problems. In *Proc. 33rd IEEE Symposium on Foundations of Computer Science*, 14–23, 1992.

[4] S. Arora and S. Safra. Probabilistic checking of proofs: a new characterization of NP. In *Proc. 33rd IEEE Symposium on Foundations of Computer Science*, 2–13, 1992.

[5] M. Bellare, O. Goldreich, and M. Sudan. Free bits, PCP's and non-approximability. *SIAM Journal on Computing*, 27:804–915, 1998.

[6] M. Bellare, S. Goldwasser, C. Lund, and A. Russell. Efficient probabilistically checkable proofs and applications to approximation. In *Proc. 25th ACM Symposium on Theory of Computing*, 294–304, 1993.

[7] M. Bellare and M. Sudan. Improved non-approximability results, In *Proc. 26th ACM Symposium on Theory of Computing*, 184–193, 1994.

[8] R. Boppana and M.M. Halldórsson. Approximating maximum independent sets by excluding subgraphs, *Bit* 32:180–196, 1992.

[9] D.P. Bovet and P. Crescenzi. *Introduction to the Theory of Complexity*. Prentice-Hall, 1994.

[10] P. Crescenzi and V. Kann. A compendium of NP optimization problems. Latest version available on-line at http://www.nada.kth.se/∼viggo/problemlist/compendium.html.

[11] P. Crescenzi, V. Kann, R. Silvestri, and L. Trevisan. Structure in approximation classes. In *Proc. 1st Computing and Combinatorics Conference*, 539–548, LNCS 959, Springer-Verlag, 1995.

[12] P. Crescenzi and L. Trevisan. On approximation scheme preserving reducibility and its applications. In *Proc. 14th Conference on Foundations of Software Technology and Theoretical Computer Science*, 330–341, LNCS 880, Springer-Verlag, 1994.

[13] H. Davenport. *Multiplicative Number Theory*, Second Edition. Springer-Verlag, 1980.

[14] U. Feige, S. Goldwasser, L. Lovasz, S. Safra, and M. Szegedy. Interactive proofs and the hardness of approximating cliques, *Journal of the ACM*, 43(2):268–292, 1996.

[15] U. Feige and J. Kilian. Two prover protocols - low error at affordable rates, In *Proc. ACM 26th Symposium on Theory of Computing*, 172–183, 1994.

[16] M.R. Garey and D.S. Johnson. Strong NP-completeness results: motivation, examples, and implications. *Journal of the ACM*, 25:499–508, 1978.

[17] M.R. Garey and D.S. Johnson. *Computers and Intractability: a Guide to the Theory of NP-Completeness.* Freeman, 1979.

[18] F. Gavril. Cited in [17], page 134.

[19] O. Goldreich and A. Wigderson. Tiny families of mixing functions. In *Proc. 26th ACM Symposium on Theory of Computing*, 574–583, 1994.

[20] M.M. Halldórsson. Approximations via partitioning, Technical Report IS-RR-95-0003F, JAIST, March 1995.

[21] J. Håstad. Testing the long code and hardness for clique. In *Proc. 28th ACM Symposium on Theory of Computing*, 11–19, 1996.

[22] J. Håstad. Clique is hard to approximate within $n^{1-\epsilon}$. In *Proc. 37th IEEE Symposium on Foundations of Computer Science*, 627–636, 1996.

[23] J. Håstad. Some optimal inapproximability results. In *Proc. 29th ACM Symposium on Theory of Computing*, 1–10, 1997.

[24] O.H. Ibarra and C.E. Kim. Fast approximation for the knapsack and sum of subset problems, *Journal of the ACM*, 22:463–468, 1975.

[25] V. Kann, S. Khanna, J. Lagergren, and A. Panconesi. On the hardness of approximating Max k-Cut and its dual. *Chicago Journal of Theoretical Computer Science*, 2, 1997.

[26] V. Kann, J. Lagergren, and A. Panconesi. Approximability of maximum splitting of k-sets and some other APX-complete problems. *Information Processing Letters*, 58:105–110, 1996.

[27] S. Khanna, R. Motwani, M. Sudan, and U. Vazirani. On syntactic versus computational views of approximability. In *Proc. 35th IEEE Symposium on Foundations of Computer Science*, 819–830, 1994.

[28] A. Lubotzky, R. Phillips, and P. Sarnak. Ramanujan graphs. *Combinatorica*, 8(3):261–277, 1988.

[29] M.V. Marathe and S.S. Ravi. On approximation algorithms for the minimum satisfiability problem. *Information Processing Letters*, 58(1):23–29, 1996.

[30] G.L. Nemhauser and L.E. Trotter. Vertex packing: structural properties and algorithms. *Mathematical Programming*, 8:232–248, 1975.

[31] C. H. Papadimitriou and M. Yannakakis. Optimization, approximation, and complexity classes. *Journal of Computer and System Sciences*, 43:425–440, 1991. Preliminary version in *Proc. of STOC'88.*

[32] L. Trevisan, G. Sorkin, M. Sudan, and D.P. Williamson. Gadgets, approximation, and linear programming. In *Proc. 37th IEEE Symposium on Foundations of Computer Science*, 617–626, 1996.

[33] M. Yannakakis On the approximation of maximum satisfiability. *Journal of Algorithms*, 17:475–502, 1994.

[34] M. Zimand. Weighted NP optimization problems: logical definability and approximation properties. In *Proc. 10th IEEE Conference on Structure in Complexity Theory*, 12–28, 1995.

Appendix

A.1 Proof of Theorem 13

We first recall the following folklore result about the mixing properties of expander graphs.

Lemma 26 (see for instance [19]) *Let $G = (V, E)$ be a d-regular graph, let λ be an upper bound on the absolute values of all the eigenvalues, save the biggest one, of the adjacency matrix of G, let A and B be any two disjoint sets of vertices, then*

$$\left| Cut(A, B) - \frac{d}{|V|}|A||B| \right| \leq \lambda \sqrt{|A||B|}.$$

As is well known, the largest eigenvalue of the adjacency matrix of a d-regular graph is equal to d, thus we need graphs with second largest eigenvalue much smaller than d. Ramanujan graphs meet our requirement.

Theorem 27 ([28], Theorem 4.4) *Let p and q be primes such that $p \equiv q \equiv 1 \pmod 4$ and $q > p$. Then there exists a graph $Y^{p,q}$ that is $p+1$-regular, has $q+1$ vertices, and such that the second largest eigenvalue of its adjacency matrix is at most $2\sqrt{p}$.*

Dirichlet's stronger version of the Prime Number Theorem helps understanding the density of primes of the form $4k+1$. Let $\pi_{b,c}(n)$ be the number of primes $p \leq n$ such that $p \equiv c \pmod b$, and let $\phi(n)$ be the Euler function, then the following theorem holds.

Theorem 28 *If b and c are co-prime, then*

$$\lim_{n\to\infty} \frac{\pi_{b,c}(n)}{n/\ln n} = \frac{1}{\phi(b)}.$$

The above theorem is due to Dirichlet, Hadamard, and de la Vallèe Poussin, and a proof can be found in [13]. Let now $l_{b,c}(n)$ be the smallest prime $p \geq n$ such that $p \equiv c \pmod b$. From the above Theorem, it is an easy exercise to show that

$$\lim_{n\to\infty} \frac{l_{4,1}(n)}{n} = 1.$$

Said another way, for any ϵ, a constant n_0 exists such that for any $n \geq n_0$ a prime $n \leq p \leq n(1+\epsilon)$ exists such that $p \equiv 1 \pmod 4$.

We are now able to prove Theorem 13. Let n_0 be such that for any $m \geq 2n_0 - 1$ a prime $p \equiv 1 \pmod 4$ exists such that $m \leq p \leq m(1+\epsilon/2)$. Let $c = \max\{n_0, 32/(\epsilon^2\delta^2)\}$.

Consider any $n \geq n_0$ and any w $cn \leq w \leq n^2$. If $w > n^2/(1+\epsilon/2)$ then the bipartite graph $\mathrm{K}_{n,n}$ is clearly (n, w, ϵ, δ)-mixing. Otherwise, the above conditions imply the existence of two primes $p \equiv q \equiv 1 \pmod 4$ such that

$$2n - 1 \leq q \leq (2n-1)(1+\epsilon/2) \leq 2n(1+\epsilon/2) - 1,$$

$$2w/n - 1 \leq p \leq (2w/n - 1)(1+\epsilon/2) \leq 2w/n(1+\epsilon/2) - 1.$$

That is,

$$\frac{w}{n^2(1+\epsilon/2)} \leq \frac{p+1}{q+1} \leq \frac{w(1+\epsilon/2)}{n^2}$$

and thus

$$\left|\frac{p+1}{q+1} - \frac{w}{n^2}\right| \leq \frac{\epsilon}{2}\frac{w}{n^2}. \tag{1}$$

Consider now the graph $Y^{p,q}$, let V_1, V_2 be any two disjoint sets of n vertices of $Y^{p,q}$. Let E be the set of edges of $Y^{p,q}$ that have an endpoint in V_1 and an endpoint in V_2, we shall prove that the bipartite graph $G = (V_1, V_2, E)$ is (n, w, ϵ, δ)-mixing. Let $A \subseteq V_1$ and $B \subseteq V_2$ be any two sets of at least δn vertices, let $a = |A|$, $b = |B|$. Applying Lemma 26 we have that

$$\left|Cut(A,B) - \frac{p+1}{q+1}ab\right| \leq 2\sqrt{pab}$$

and, combining with (1),

$$\begin{aligned}
\left|Cut(A,B) - \frac{w}{n^2}ab\right| &\leq \frac{\epsilon}{2}\frac{w}{n^2}ab + 2\sqrt{pab} \\
&\leq \frac{\epsilon}{2}\frac{w}{n^2}ab + 2\sqrt{\frac{2w}{n}ab} \\
&\leq \frac{w}{n^2}ab\left(\frac{\epsilon}{2} + 2\sqrt{\frac{2n^3}{wab}}\right) \\
&\leq \frac{w}{n^2}ab\left(\frac{\epsilon}{2} + 2\sqrt{\frac{2n^3}{cn\delta^2 n^2}}\right) \\
&\leq \frac{w}{n^2}ab\left(\frac{\epsilon}{2} + \frac{\epsilon}{2}\right).
\end{aligned}$$