The (Parallel) Approximability of Non-Boolean Satisfiability Problems and Restricted Integer Programming

Maria Serna[1⋆], Luca Trevisan[2] and Fatos Xhafa[1⋆]

[1] Departament de Llenguatges i Sistemes Informàtics
Universitat Politècnica de Catalunya
Módul C6 - Campus Nord, Jordi Girona Salgado, 1-3
08034-Barcelona, Spain
E-mail: {mjserna,fatos}@lsi.upc.es

[2] MIT Laboratory for Computer Science
545 Technology Square, Room NE43-371, Cambridge, MA 02139, USA
E-mail: luca@theory.lcs.mit.edu

Abstract. We present parallel approximation algorithms for maximization problems expressible by integer linear programs of a restricted syntactic form introduced by Barland et al. [BKT96]. One of our motivations was to show whether the approximation results in the framework of Barland et al. holds in the parallel setting. Our results are a confirmation of this, and thus we have a new common framework for both computational settings. Also, we prove almost tight non-approximability results, thus solving a main open question of Barland et al.
We obtain the results through the constraint satisfaction problem over multi-valued domains, for which we show non-approximability results and develop parallel approximation algorithms.
Our parallel approximation algorithms are based on linear programming and random rounding; they are better than previously known sequential algorithms. The non-approximability results are based on new recent progress in the fields of Probabilistically Checkable Proofs and Multi-Prover One-Round Proof Systems [Raz95, Hås97, AS97, RS97].

1 Introduction

Expressing combinatorial optimization problems as integer linear programs (ILP) has several applications. In particular, several approximation algorithms start from the linear programming relaxation of the ILP formulation, and then use randomized rounding [RT87, GW94], primal-dual methods [GW96], or more sophisticated methods [LLR95, ENRS95].

An interesting new *structural* use of Integer Linear Programming has been taken in a recent paper of Barland, Kolaitis and Thakur [BKT96], where *syntactic classes* of maximization problems are introduced. A problem belongs to

⋆ Research supported by the ESPRIT Long Term Research Project No. 20244 - ALCOM IT and the CICYT Project TIC97-1475-CE.

one such class if it can be expressed by an ILP with a certain restricted format. The approximability properties of the problem in a class are then implied by the approximability of the respective prototypical ILP. The main goal of [BKT96] was to overcome some limitations of the standard way of defining syntactic classes, namely the approach of logical definability [PY91, PR93, KT94, KT95]. The latter approach, indeed, fails to explain why problems with similar logical definability, such as Max k-dimensional Matching and Max Clique have very different approximability properties. Furthermore, using ILP, classes are defined in terms of a single parameter that determines the hardness of the problems. This parameter is either the maximum number of occurrences of any variable or the maximum size of the domain of the variables. The latter kind of restriction gives rise to a family of classes that Barland et al. call MAX FSBLIP (for Maximum Feasible Subsystem of Bounded Layered Integer Program). Letting the variables to take values in a constant, logarithmic, or polynomial range allowed them to capture syntactic maximization classes that are constant-approximable, polylog-approximable and poly-approximable, respectively. An interesting question is whether these three classes form a proper hierarchy. Barland et al. did not completely resolve this point and left improved non-approximability results as an open question.

In this paper our interest is twofold. In one hand, we use the integer programming as a framework for parallel approximability, aiming to obtain improved parallel approximation results. It is known that all the problems contained in logically defined syntactic classes that are constant-factor approximable, are also constant-factor approximable[3] in NC. This feature of logically defined syntactic classes is desirable for at least two reasons: it reduces the study of sequential and parallel approximability to the same framework, and is in accordance with the fact that almost all the constant factor approximation algorithms that are known also admit a parallel version with a comparable approximation ratio. The issue of parallel approximability is not raised in the paper of Barland et al. Our parallel results state that in the new framework of integer programming the sequential results holds as well as in the parallel setting thus, again, we have a common framework for both computational settings. Having this outcome, the second question that we consider is what are the limits of parallel, as well as sequential, approximability for these problems. We show that our approximation factors are nearly the best possible by providing some new non-approximability results (the non-approximability results will also hold for sequential algorithms.) In both cases, our main results will be expressed in terms of the multi-valued constraint satisfaction problem, and then translated, by means of reductions, in terms of the model of Barland et al.

In the rest of this section, we state our results and we discuss their relation with previously known ones.

In this paper, a crucial role is played by the constraint satisfaction problem over multi-valued domains. In an instance of this problem, we are given a set of

[3] An NC algorithm is an algorithm that runs in poly-logarithmic time on a parallel shared-memory machine with a polynomial number of processors. See e.g. [DSST97].

constraints of arity at most k over multi-values variables where a constraint is a boolean valued function over $\{0, 1, \ldots, d-1\}^k$ and is given a positive weight. We can think of a k-ary domain-d constraint as a set of k-tuples values (i.e. a relation over $\{0, 1, \ldots, d-1\}^k$) and say that an assignment satisfies the constraint if the corresponding values to the variables of the constraint form a k-tuple belonging to the relation. The goal is to find an assignment to the variables that maximizes the total weight of satisfied constraints. This problem is a common generalization of several known and well-studied problems. To begin with, it is a natural generalization of the boolean constraint satisfaction problem MAX kCSP, introduced by Khanna et al. [KMSV94] and then studied in [Cre95, Tre96, KSW97] (in the boolean case, the domain is $\{0, 1\}$, that is, $d = 2$.) It also generalizes Multi-Prover One-Round Proof Systems and the MAX CAPACITY REPRESENTATIVES problem (introduced by Bellare et al. [Bel93] and further considered by Barland et al.). The version over multi-valued domain has been studied in the restricted case of binary constraints [LW96] and that of "planar instances" [KM96]. In this paper we address, for the first time, the approximability of the problem in its full generality. We present a parallel approximation, based on linear programming and random rounding, that achieves an approximation factor $1/d^{k-1}$. The algorithm can be efficiently parallelized and de-randomized. Our major contribution here is the definition and the analysis of an appropriate random rounding scheme. The parallelization mimics a similar proof in [Tre96], but is not entirely straightforward. For the special case of binary constraint (k=2), our approximation guarantee is twice better than the $1/2d$-approximate algorithm of [LW96].

We also prove several non-approximability results under different complexity assumptions. Such results follow from recent advances in the fields of Probabilistically Checkable Proofs [Hås97] and of Multi-Prover One-Round Proof Systems [Raz95, RS97, AS97] and from the fact that multi-valued constraint satisfaction problems generalize both models. We use reductions from the multi-valued constraint satisfaction problem to derive negative approximation results for the rest of the problems of interest. In terms of the class FSBLIP, our result states that the classes MAX FSBLIP(2), MAX FSBLIP(log) and MAX FSBLIP(poly) form a proper hierarchy (the separation of the two last classes derives from a result of Bellare [Bel93] stating that MAX CAPACITY REPRESENTATIVES which belongs to MAX FSBLIP(poly) is not log-approximable; we separate the first two classes by proving that MAX CAPACITY REPRESENTATIVES(log) is not constant-approximable.)

We also consider the class of integer programs MAX FMIP (for Maximum Feasible Majority Integer Program) for which MAX MAJORITY SAT is a canonical problem. [BKT96] showed that this class contains only constant-approximable problems. For the general MAX FMIP problem, we present a slight improvement and simplification over their approximation result. The latter result does not depend on the constraint satisfaction problem. We also prove an almost tight non-approximability result for the problems of this class by reducing from the boolean constraint satisfaction problem.

2 Preliminaries

For an integer n, we denote by $[n]$ the set $\{0, \ldots, n-1\}$. A combinatorial optimization problem is characterized by the set of *instances*, by the finite set of *feasible solutions* associated to any instance, and by a *measure* function that associates a non-negative *cost* to any feasible solution of a given instance. We refer e.g. to [BC93] for the formal definition of NP Optimization problem.

Definition 1 (Max Capacity Representatives-d). For a function $d : \mathcal{Z}^+ \to \mathcal{Z}^+$, Max Capacity Representatives-$(d(n))$ problem is defined as follows:

Instance: A partition of $\{1, \ldots, n\}$ into sets $S_1, \ldots, S_m$, each of cardinality at most d; and weights $w_{i,j} \geq 0$ for any two elements belonging to different sets of the partition.
Solution: The choice of a representative in any set.
Measure: The sum of the weights $w_{i,j}$ for any i and j that are representatives in different sets of the partition.

Definition 2 (Constraint). A k-ary, domain-d constraint over $x_1, \ldots, x_n$ is a pair $(f, (i_1, \ldots, i_k))$ where $f : [d]^k \to \{0, 1\}$ and $i_j \in \{1, \ldots, n\}$ for $j = 1, \ldots, k$. A constraint $C = (f, (i_1, \ldots, i_k))$ is *satisfied* by an assignment $\mathbf{a} = a_1, \ldots, a_n$ to $x_1, \ldots, x_n$ if $C(\mathbf{a}) \stackrel{\text{def}}{=} f(a_{i_1}, \ldots, a_{i_k}) = 1$.

We say that a function $f : [d]^k \to \{0, 1\}$ is *conjunctive* if it can be expressed as a conjunction of equations, i.e. there are values $v_1, \ldots, v_k \in [d]$,

$$f(x_1, \ldots, x_k) = 1 \text{ if and only if } [x_1 = v_1] \wedge \ldots \wedge [x_k = v_k] \ .$$

When this will not cause confusion, we will sometimes blur the important difference between a constraint $(f, (i_1, \ldots, i_k))$ and the function f. For example we say that a constraint $(f, (i_1, \ldots, i_k))$ is conjunctive if function f is, and so on.

Definition 3 (Max kCSP-d and Max kConj-d). For any integer $k \geq 1$ and function $d = d(n)$, the Max kCSP-d is defined as follows:

Instance: A set $\{C_1, \ldots, C_m\}$ of domain-d constraints of arity at most k over $x_1, \ldots, x_n$, and associated non-negative weights $w_1, \ldots, w_m$.
Solution: An assignment $\mathbf{a} = (a_1, \ldots, a_n) \in [d]^n$ to the variables $x_1, \ldots, x_n$.
Measure: The total weight of satisfied constraints.

Max kConj-d is the restriction of Max kCSP-d to instances where all the constraints are conjunctive.

Definition 4 (Integer Linear Programming (ILP)). The ILP is as follows:

Instance: A matrix $A \in \mathcal{Z}^{m \times n}$ and two vectors $\mathbf{c} \in \mathcal{Z}^n$ and $\mathbf{b} \in \mathcal{Z}^m$.
Solution: A vector $\mathbf{x} \in \mathcal{Z}^n$ satisfying $A\mathbf{x} \leq \mathbf{b}$.
Measure: $\mathbf{c} \cdot \mathbf{x}$.

Note that in this formulation, the goal is to maximize the measure $\mathbf{c}\cdot\mathbf{x}$. The variables appearing (with non-zero coefficients) in the objective function are called *objective variables* and those appearing only in the linear constraints are *program variables.* The *width* of a constraint is equal to the number of its variables.

Definition 5 (Constraint Dominance). Given a linear constraint of the form $g(1-t)+\mathbf{a}\cdot\mathbf{q} \geq b$, where t is 0/1 variable, it is said that t dominates the constraint if (a) for $t = 0$ the constraint is satisfied whatever is the assignment to the rest of variables; (b) if an assignment satisfies $\mathbf{a} \cdot \mathbf{q} \geq b$, then the constraint is satisfied for any value of t.

Definition 6 (Max FSBLIP($d(n)$) **[BKT96]).** For a given function $d(n)$, the class Max FSBLIP($d(n)$) contains all the optimization problems A for which there are positive integer constants l, m, k (that only depend on A) such that every instance of A can be expressed as an ILP with the following structure:

- The program variables can take values in $\{0, 1, \dots d(n) - 1\}$.
- Each objective variable t_i occurs only in constraints of the form $(1 - t_i) + q_{i,1} + \cdots + q_{i,z} \geq 1$, where $z \in \mathbf{N}$ can be polynomial in n, and each $q_{i,j}$, $1 \leq j \leq z$ is a 0/1 program variable associated with the objective variable t_i. These constraints are referred to as objective constraints.
- Each variable $q_{i,j}$ appearing in an objective constraint occurs in at most l other constraints and dominates each of them.
- All constraints that are not objective ones have width m and are dominated by some $q_{i,j}$ associated with some objective variable t_i.
- Each objective variable t_i appears in at most k objective constraints.

Definition 7 (Max FMIP **[BKT96]).** An optimization problem Π belongs to the class Max Feasible Majority IP (in short, Max FMIP) if there exist positive constants k, σ and a polynomial p such that for any instance I of Π we can find a set of linear inequalities over the integers

$$A\mathbf{x} \geq \mathbf{b}$$
$$\mathbf{x} \in \{-k, -k+1, \dots, k-1, k\}^n$$

where $b_j \leq \sigma$, the entries of A are integers of absolute value at most $p(n)$, and the optimum of I is precisely the maximum number of inequalities that are simultaneously satisfiable.

3 Reductions Among Problems

Theorem 8. *For any constant k and function $d(n)$,* Max kConj-$d(n)$ *belongs to* Max FSBLIP($d(n)$).

Proof. Our formulation is similar to that of Max Capacity Representatives given in [BKT96, Section 3]. Let $\{C_1, \dots, C_m\}$ be a set of k-ary domain-d conjunctive constraints over $x_1, \dots, x_n$, and $w_1, \dots, w_m$ be associated non-negative weights. We use two 0/1 variables t_j and f_j for any constraint, and we use a d-valued variable y_i for any variable x_i. The integer linear program is

$$\begin{array}{l} \max \sum_j w_j t_j \\ \text{s.t.} \\ \qquad \begin{array}{ll} (1-t_j)+f_j \geq 1 & \forall j=1,\ldots,m \\ d(1-f_j)+y_i \geq v & \forall j=1,\ldots,m,\ \forall[x_i=v]\in C_j \\ d(1-f_j)-y_i \geq -v & \forall j=1,\ldots,m,\ \forall[x_i=v]\in C_j \end{array} \end{array}$$

Notice that each objective variable t_j appears in a unique objective constraint, each variable f_j in an objective constraints occurs in at most $2k$ other constraints dominating each of them, and, finally, any constraints has width 2. □

Theorem 9. *If* Max kConj-d *is r-approximable (in* NC*) and $k^d = \text{poly}(n)$, then* Max kCSP-d *is r-approximable (in* NC*).*

Proof. For any constraint C_j of weight w_j, let s be the number of satisfying assignments to its variables (note that $s \leq k^d$). Then we can express C_j as the disjunction of s conjunctive constraints $K_j^1, \ldots, K_j^s$, each one enforcing one of the satisfying assignments of C_j. Observe that any (global) assignment, satisfies at most one of the K_j^i constraints and satisfies one if and only if satisfies C_j. Let us substitute C_j with the $K_j^1, \ldots, K_j^s$ constraints, and give weight w_j to all of them. We repeat the same substitution for any constraint. The new instance is equivalent to the former, in the sense that they share the same set of feasible solutions, and the cost of each solution is always the same. Observe that the substitution process can be done also in parallel for all the constraints. □

Theorem 10. Max 2Conj-d *is r-approximable (in* NC*) if and only if* Max Capacity Representatives-d *is r-approximable (in* NC*).*

Proof. It is easy to see that the two problems are equal. Without loss of generality we can assume that any set in a Max Capacity Representatives-d instance has exactly d elements (add dummy elements and give weight zero to the pairs corresponding to such elements) and that in a Max 2Conj-d instance with n variables there are all the possible $\binom{n}{2}d^2$ conjunctive constraints (add the missing constraints with weight zero). Now, the equivalence is immediate: every set S_i in Max Capacity Representatives-d corresponds to a d-valued variable $s_i = a$, $a = 0, 1, \ldots, d-1$, meaning that the representative of set S_i is a; to a pair of representatives in different sets S_i, S_j corresponds a conjunctive constraint $s_i = a \wedge s_j = b$; the weight of a constraint is that of the edge from which it was derived. Clearly, starting from an instance of Max Capacity Representatives-d we construct (in NC) an instance of Max 2Conj-d such that its feasible solutions are also feasible solutions of the same cost for Max Capacity Representatives-d and vice-versa. The theorem thus readily follows. □

Theorem 11. Max kConj-2 *can be expressed as a* Max FMIP *problem with $p(n) = 1$, $k' = 2$ and $\sigma = k$.*

Proof. Let φ be an instance of MAX kCONJ-2. We have a variable $y_i \in \{-1, 0, 1\}$ for any variable x_i of φ. For any constraint C_j, let P_j (resp. N_j) be the set of indices of variables that are assigned to 1 (resp. 0) in C_j. Let k_j be the arity of C_j. Then C_j is expressible as

$$\bigwedge_{i \in P_j} [x_i = 1] \wedge \bigwedge_{i \in N_j} [x_i = 0] \ .$$

We translate C_j into the constraint $\sum_{i \in P_j} y_i + \sum_{i \in N_j} -y_i \geq k_j$. Under the understanding that $\{-1, 1\}$ assignments to y_i should be mapped to $\{0, 1\}$ assignments for x_i (i.e. $x_i = (1 + y_i)/2$), the two constraints are equivalent. We repeat the translation for any constraint, and the theorem thus follows. ☐

4 Positive Results: Algorithms

We now consider a linear programming relaxation of MAX kCONJ-d. We have a variable z_j for any constraint C_j, with the intended meaning that $z_j = 1$ when C_j is satisfied and $z_j = 0$ otherwise. We also have a variable $t_{i,v}$ for any variable x_i and any value $v \in [d]$, meaning that $t_{i,v} = 1$ if $x_i = v$ and $t_{i,v} = 0$ otherwise.

$$\begin{array}{lll} \max \sum_j w_j z_j & & \\ \text{s.t.} & & \\ & z_j \leq t_{i,v} & \forall i, v, [x_i = v] \in C_j \\ & \sum_{v \in [d]} t_{i,v} = 1 & \\ & 0 \leq t_{i,v} \leq 1 & \forall i \in [n], \forall v \in [d] \end{array} \qquad \text{(CONJ)}$$

Lemma 12. *The linear program* (CONJ) *is* $(1 - o(1))$*-approximable in* NC.

Proof. Generalization of a result of [Tre96]. The proof is omitted from this extended abstract. ☐

Lemma 13 (Random Rounding for MAX kCSP-d**).** *Let* $(\mathbf{z}, \mathbf{t})$ *be a feasible solution for* (CONJ)*. Consider the random assignment obtained by setting, for any* i*,* v

$$\mathbf{Pr}[x_i = v] = (k-1)/dk + t_{i,v}/k \ .$$

Then such an assignment has an average cost at least $\frac{1}{d^{k-1}} \sum_j w_j z_j$*. The analysis only assumes that the distribution is* k*-wise independent.*

Proof. It is sufficient to prove that any constraint C_j is satisfied with probability at least $\frac{1}{d^{k-1}} z_j$; the lemma will then follow by the linearity of expectation. Observe that if the atom $[x_i = v]$ occurs in C_j then $z_j \leq t_{i,v}$. Then

$$\mathbf{Pr}[C_j \text{ is satisfied}] \geq \left(\frac{k-1}{dk} + \frac{1}{k} z_j \right)^k \geq \frac{1}{d^{k-1}} z_j \ . \qquad (1)$$

For the last inequality, we consider the function

$$f(z) = \frac{\left(\frac{k-1}{dk} + \frac{1}{k}z\right)^k}{z}$$

in the interval $0 \leq z \leq 1$, compute its first derivative, and show that f has a minimum in $z = 1/d$, that is $f(z) \geq f(1/d) = 1/d^{k-1}$, $\forall z, 0 \leq z \leq 1$. In the first inequality of Eq. (1) we have assumed that the random variables induced by the clause C_j are independent. □

Remark. The above analysis is tight and establishes that the integrality gap of (CONJ) is d^{k-1}. The bound is achieved e.g. by the instance consisting of clauses $C_1, C_2, \ldots, C_{d^k}$ that are all possible size k (domain-d) conjunctions of $\{x_1, \ldots, x_k\}$.

Theorem 14. *For any $d = d(n)$ and $k = k(n)$ such that $d^k = n^{O(1)}$, there is an* NC *$(1/d^{k-1} - o(1))$-approximate algorithm for* MAX kCSP-d*. In particular, there is a $(1/d - o(1))$-approximate* NC *algorithm for* MAX CAPACITY REPRESENTATIVES-d*.*

4.1 The MAX FMIP Problems

A prototypical problem in MAX FMIP is MAX MAJORITY SAT, which is the variation of MAX SAT where a clause is satisfied if at least half the literals (rather than at least one) are satisfied. Baralnd et al. [BKT96] showed that this class contains only constant-approximable problems (using, once more, the syntactic structure of integer programs) and gave a structural explanation of this result.

It is easy to find a 2-approximate solution for MAX MAJORITY SAT. Any clause is either satisfied by the assignment $x_i = 0$, $\forall i$, or by the assignment $x_i = 1$, $\forall i$. Thus one of the two assignments satisfies at least half the clauses.[4]

For the general MAX FMIP problem, we present a slight improvement and simplification over the approximation result of Barland et al. [BKT96].

Theorem 15. *Given an instance of a* MAX FMIP *problem, the random assignment where each variable is set to $-k$ or to k with probability $1/2$ independently at random satisfies each constraint with probability at least $1/2^{1+\lceil \sigma/k \rceil}$, provided that the constraint is satisfiable.*

Proof. Consider a constraint $\sum_i a_i x_i \geq b$. If the constraint is satisfiable, then $\sum_i |a_i| k \geq b$. Since the a_i are integers, there must be a set J of at most $\lceil b/k \rceil$ indices such that $\sum_{i \in J} |a_i| k \geq b$. Under the uniform distribution, with probability at least $1/2^{|J|} \geq 1/2^{\lceil b/k \rceil}$ we will have $\sum_{i \in J} a_i x_i \geq b$. It is also easy to see that, by symmetry, with probability at least $1/2$ we have $\sum_{i \notin J} a_i x_i \geq 0$.
The theorem thus follows since for the whole set of constraints, $b_j \leq \sigma$, $\forall j$. □

The above theorem can be derandomized in NC through the techniques of Karger and Kholler [KK94].

[4] This nice idea is due to Michel Goemans.

5 Negative Results: Hardness of Approximation

We first define Probabilistically Checkable Proof Systems and Multi-Prover One-Round Proof Systems. Our notation merges the notations of [BGLR93] and [BGS96]. For an integer d, we denote by $[d]^*$ the set of all strings over $[d]$.

Definition 16 (Verifier). A verifier V for a language L is a randomized polynomial time oracle Turing machine. V receives in input a string x and has oracle access to a string π that is an alleged proof that $x \in L$.

Definition 17 (PCP and MIP). Let $c, s, r, q, d : \mathcal{Z}^+ \rightarrow \mathcal{Z}^+$ such that $0 \leq s(n) < c(n) \leq 1$ for any n; we say that a language L belongs to $\mathsf{PCP}_{c,s}[r, q, d]$ if there exists a verifier V such that

1. For any input string x and oracle proof $\pi \in [d(n)]^*$, V queries at most $q(n)$ entries of π and uses at most $O(r(n))$ random bits;
2. For any $x \in L$, there exists a $\pi \in [d(n)]^*$ such that the probability that V accepts x with oracle π is at least $c(n)$;
3. For any $x \notin L$, for any $\pi \in [d(n)]^*$, the probability that V accepts x with oracle π is at most $s(n)$.

The class $\mathsf{MIP}_{c,s}[r, q, d]$ is similar, with the only difference that π is presented as a sequence of q strings $\pi_1, \ldots, \pi_q$, where $\pi_i \in [d]^*$, and V has the further restriction that it can read at most one entry of any π_i.

From the above definition it follows that $\mathsf{MIP}_{c,s}[r, q, d] \subseteq \mathsf{PCP}_{c,s}[r, q, d]$ for any choice of the parameters. The following result is folklore.

Theorem 18. *If* Max kCSP-$(d(n))$ *is* $\rho(n)$*-approximable, then, for any* $c(n)$ *and* $s(n)$ *such that* $s(n)/c(n) < \rho\left(n^{O(1)}2^{O(r(n))}\right)$*, it holds*

$$\mathsf{PCP}_{c(n),s(n)}[r(n), k(n), d(n)] \subseteq \mathsf{DTIME}\left(2^{O(r(n)+k(n)\log d(n))}\right) .$$

Theorem 19. *The following statements hold (n is the size of the input):*

(1) *A constant $c > 0$ exists such that, for any constant $d \geq 2$, it is* NP*-hard to approximate* Max 2CSP-d *within* $1/d^c$*. Furthermore, for any $\varepsilon > 0$, it is infeasible to approximate* Max 2CSP-$(\log n)$ *within* $2^{\log^{1-\varepsilon} n}$ *unless* $\mathsf{NP} \subseteq \mathsf{DTIME}\left(n^{\log^{O(1/\varepsilon)} n}\right)$.

(2) *For any constant d, for any $k \geq 3$, for any $\varepsilon > 0$, it is* NP*-hard to approximate* Max kCSP-d *within* $1/d^{\lfloor k/3 \rfloor} + \varepsilon$.

(3) *Constants k and c exist such that it is* NP*-hard to approximate* Max kCSP-$(\log n)$ *within* $1/\log n^c$.

(4) *For any $k \geq 5$, any $\varepsilon > 0$, it is* NP*-hard to approximate* Max kCSP *within* $2^{\log^{1/3-\varepsilon} n}$.

(5) *For any $\varepsilon > 0$, a constant $k = O(1/\varepsilon)$ exists such that it is* NP*-hard to approximate* Max kCSP *within* $2^{\log^{1-\varepsilon} n}$.

(6) *For any* $\varepsilon > 0$*,* Max FMIP *problems are hard to approximate within* $1/2^{\lfloor \sigma/3 \rfloor} + \varepsilon$*.*

Proof. (Sketch) For **(1)**, Raz [Raz95] has shown that a constant $c' > 0$ exists such that, for any $k : \mathcal{Z}^+ \to \mathcal{Z}^+$, $\mathsf{NP} \subseteq \mathsf{MIP}_{1,2^{-ck(n)}}[k(n)\log n, 2, 3^{k(n)}]$. The first part of the claim follows by setting $k(n) = \lfloor \log_3 d(n) \rfloor$; the second part by setting $k(n) = \log^{O(1/\varepsilon)}(n)$. Next, for **(2)**, Håstad [Hås97] has shown that for any $\varepsilon > 0$, for any fixed prime p, $\mathsf{NP} = \mathsf{PCP}_{1-\varepsilon,1/p+\varepsilon}[\log, 3, p]$. The claim follows by choosing $p = k/3$. Further, **(3)**, **(4)** and **(5)** are re-statements of the results of Raz and Safra [RS97], and Arora and Sudan [AS97] using Theorem 18. Finally, **(6)** follows from the hardness of Max kCSP-2 and from Theorem 11. □

Barland et al. asked in [BKT96] whether the problem Max Capacity Representatives($\log n$) is constant-approximable. Part **(1)** of Theorem 19 and Theorem 10 imply a negative answer to such question. Finally, it is worth to mention the almost tight non-approximability result for the problems of class Max FMIP.

Acknowledgment

We thank an anonymous referee for useful comments.

References

[AS97] Arora, S., and Sudan, M. Improved low degree testing and its applications. In *Proc. of 29th ACM STOC* (1997) 485–495

[BC93] Bovet, D.P. and Crescenzi, P. *Introduction to the Theory of Complexity.* Prentice Hall (1993)

[Bel93] Bellare, M. Interactive proofs and approximation: Reductions from two provers in one round. In *Proc. of 2nd IEEE ISTCS* (1993)

[BGLR93] Bellare, M., Goldwasser, S., Lund, C., and Russell, A. Efficient probabilistically checkable proofs and applications to approximation. In *Proc. of the 25th ACM STOC* (1993) 294–304

[BGS96] Bellare, M., Goldreich, O., and Sudan, M. Free bits, PCP's and non-approximability – towards tight results (4th version). Technical Report TR95-24, ECCC (1996) Preliminary version in *Proc. of FOCS'95.*

[BKT96] Barland, I., Kolaitis, P.G., and Thakur, M.N. Integer programming as a framework for optimization and approximability. In *Proc. of 11th IEEE CCC* (1996) 249–259

[Cre95] Creignou, N. A dichotomy theorem for maximum generalized satisfiability problems. *J. of Comp. and Sys. Sci.*, (1995) 51(3):511–522

[DSST97] Díaz, J., Serna, M., Spirakis, P., and Torán, J. *Paradigms for fast parallel approximability.* Camb. Univ. Press (1997)

[ENRS95] Even, G., Naor, J., Rao, S., and Schieber, B. Divide-and-conquer approximation algorithms via spreading metrics. In *Proc. of 36th IEEE FOCS* (1995) 62–71

[GW94] Goemans, M.X., and Williamson, D.P. New 3/4-approximation algorithms for the maximum satisfiability problem. *SIAM J. of Disc. Math.* (1994) 7:656–666

[GW96] Goemans, M.X., and Williamson, D.P. The primal-dual method for approximation algorithms and its application to the network design problems. In *Approximation Algorithms for NP-hard Problems.* PWS Pub. (1996)

[Hås97] Håstad, J. Some optimal inapproximability results. In *Proc. of 29th ACM STOC* (1997) 1–10

[KK94] Karger, D.R. and Koller, D. (De)randomized construction of small sample spaces in $\mathcal{NC}$. In *Proc. of 35th IEEE FOCS* (1994) 252–263

[KM96] Khanna, S., and Motwani, R. Towards a syntactic characterization of PTAS. In *Proc. of 28th ACM STOC* (1996) 329–337

[KMSV94] Khanna, S., Motwani, R., Sudan, M., and Vazirani, U. On syntactic versus computational views of approximability. In *Proc. of 35th IEEE FOCS* (1994) 819–830

[KSW97] Khanna, S., Sudan, M., and Williamson, D.P. A complete classification of the approximability of maximization problems derived from boolean constraint satisfaction. In *Proc. of 29th ACM STOC* (1997) 11–20

[KT94] Kolaitis, P.G. and Thakur, M.N. Logical definability of NP optimization problems. *Inf. and Comp.* (1994) 115(2):321–353

[KT95] Kolaitis, P.G. and Thakur, M.N. Approximation properties of NP minimization classes. *J. of Comp. and Syst. Sci.* (1995) 50:391–411

[LLR95] Linial, N., London, E., and Rabinovich, Y. The geometry of graphs and some of its algorithmic applications. *Combinatorica* (1995) 15(2):215–245

[LN93] Luby, M., and Nisan, N. A parallel approximation algorithm for positive linear programming. In *Proc. of 25th ACM STOC* (1993) 448–457

[LW96] Lau, H.C., and Watanabe, O. Randomized approximation of the constraint satisfaction problem. In *Proc. of 5th SWAT*, LNCS 1097, Springer-Verlag (1996) 76–87

[PR93] Panconesi, A., and Ranjan, D. Quantifiers and approximations. *Theoret. Comp. Sci.* (1993) 107:145–163

[PY91] Papadimitriou, C.H., and Yannakakis, M. Optimization, approximation, and complexity classes. *J. of Comp. and Syst. Sci.* (1991) 43:425–440

[Raz95] R. Raz. A parallel repetition theorem. In *Proc. of 27th ACM STOC* (1995) 447–456

[RS97] Raz, R., and Safra, S. A sub-constant error-probability low-degree test, and a sub-constant error-probability PCP characterization of NP. In *Proc. of 29th ACM STOC* (1997) 475–484

[RT87] Raghavan, P., and Thompson, C.D. Randomized rounding: a technique for provably good algorithms and algorithmic proofs. *Combinatorica* (1987) 7:365–374

[Tre96] Trevisan, L. Positive linear programming, parallel approximation, and PCP's. In *Proc. of 4th ESA*, LNCS 1136, Springer-Verlag (1996) 62–75