Algorithms with large domination ratio

Noga Alon *   Gregory Gutin †   Michael Krivelevich ‡

Abstract

Let P be an optimization problem, and let A be an approximation algorithm for P. The domination ratio $\text{domr}(A, n)$ is the maximum real q such that the solution $x(I)$ obtained by A for any instance I of P of size n is not worse than at least a fraction q of the feasible solutions of I. We describe a deterministic, polynomial time algorithm with domination ratio $1 - o(1)$ for the partition problem, and a deterministic, polynomial time algorithm with domination ratio $\Omega(1)$ for the MaxCut problem and for some far-reaching extensions of it, including Max-r-Sat, for each fixed r. The techniques combine combinatorial and probabilistic methods with tools from Harmonic Analysis.

Keywords: Combinatorial Optimization; Domination analysis; Approximation Algorithms

1 Introduction

Let P be an optimization problem, and let A be an approximation algorithm for P. The *domination ratio* $\text{domr}(A, n)$ is the maximum real q such that the solution $x(I)$ obtained by A for any instance I of P of size n is not worse than at least a fraction q of the feasible solutions of I.

Initially, domination ratios were studied only for traveling salesman problem (TSP) heuristics; for a recent survey see [16]. It turned out that some often-used heuristics such as the greedy and nearest neighbor algorithms have domination ratio $1/t_n$, where t_n is the total number of tours (i.e., $t_n = (n-1)!$ for the asymmetric TSP and $t_n = (n-1)!/2$ for the symmetric one). This means that these heuristics produce the unique worst possible tour for some assignments of weights for every $n \geq 3$. This is in line with latest computational experiments with the greedy and nearest neighbor algorithms, see, e.g., [17], where the authors came to the conclusion that the greedy algorithm might be said to be self-destructive, and that it should not be used even as a general-purpose starting tour generator.

On the other hand, there are several well-known TSP heuristics of domination ratio $\Omega(1/n)$ [14, 16, 21, 22]. The domination ratio provides a theoretical explanation of the fact that some 'small' local search neighborhoods are much more computationally effective in practice than some neighborhoods of exponential size [16], as indeed some local search algorithms with smaller neighborhoods have much larger domination ratio.

Interestingly, the well-known double minimum spanning tree heuristic (DMST) is of domination ratio $1/t_n$ ($n \geq 3$) even for the symmetric TSP with the triangle inequality [21]. Thus,

*Institute for Advanced Study, Princeton, NJ 08540, USA and Department of Mathematics, Tel Aviv University, Tel Aviv 69978, Israel. E-mail: nogaa@post.tau.ac.il. Research supported in part by a USA Israeli BSF grant and by a grant from the Israel Science Foundation.

†Department of Computer Science, Royal Holloway, University of London, Egham, Surrey TW20 0EX, UK. E-mail:gutin@cs.rhul.ac.uk. Research supported in part by a Leverhulme Trust grant.

‡Department of Mathematics, Tel Aviv University, Tel Aviv 69978, Israel. E-mail: krivelev@post.tau.ac.il. Research supported in part by a USA-Israel BSF Grant, by a grant from the Israel Science Foundation and by a Bergmann Memorial Grant.

despite the fact that its performance ratio is 2, DMST produces the unique worst tour for some assignments of weights. This justifies the study of the domination ratio alongside with the performance ratio of efficient algorithms.

The domination ratios of algorithms for some other combinatorial optimization problems have also been investigated [5, 6, 13, 14, 15, 19]. In [5], a pair of heuristics for Generalized TSP have been compared. The performance of both heuristics in computational experiments was very similar. Nevertheless, bounds for the domination ratios showed that one of the heuristics was much better than the other one in the worst case. A pair of greedy-type heuristics for the frequency assignment problem were compared in [19]. Again, bounds for the domination ratios allowed the authors of [19] to find out which of the two heuristics behaves better in the worst case.

The authors of [13] obtained a deterministic polynomial time algorithm for the minimum partition problem with domination ratio at least $1/2$. They asked which other problems admit polynomial algorithms of domination ratio $\Omega(1)$. In the present paper we describe several new results of this type. Our results include:

- A deterministic polynomial time algorithm for the minimum partition problem whose domination ratio is $1 - o(1)$, improving the $1/2$-domination ratio result of [13].
- A deterministic polynomial time algorithm for the (weighted) MaxCut problem, whose domination ratio is $\Omega(1)$, improving the $\Omega(1/n)$-domination ratio result for this problem described in [13].
- A deterministic polynomial time algorithm of domination ratio $\Omega(1)$ for the problem of maximizing the total weight of satisfied equations in a system of sparse linear equations over $GF(2)$.
- A deterministic polynomial time algorithm of domination ratio $\Omega(1)$ for the weighted Max-r-Constraint Satisfaction Problem, where r is fixed. This includes, as special cases, the Max-r-Sat problem as well as the sparse linear equations problem.

The methods combine combinatorial and probabilistic techniques with tools from Harmonic analysis and results about small sample spaces that support k-wise independent random variables. As a by-product we obtain certain consequences about linear codes with sparse parity check or generating matrices, which may be of independent interest.

In all the problems we consider it is quite easy to generate deterministically a solution whose value is at least as good as the average value of a random feasible solution. Our aim is thus either to prove that the average solution beats a constant fraction of the feasible solutions, or to supply an algorithm that improves over the average solution. The latter task has common lines with the approach of a recent paper by Håstad and Venkatesh [18]. In fact we will extend and improve some of the results of [18].

The rest of this paper is organized as follows. In Section 2 we consider the partition problem, Section 3 deals with MaxCut, Section 4 deals with the more general problem of sparse linear equations over $GF(2)$, Section 5 deals with a further generalization, the weighted constraint satisfaction problem, and the final Section 6 contains some concluding remarks and open problems, including certain results about linear codes.

2 The Minimum Partition Problem

Let B_n be the set of all n-dimensional vectors $(\epsilon_1, \epsilon_2, \ldots, \epsilon_n)$ with $\{-1, 1\}$ coordinates.

The minimum partition problem (MinPart) can be stated as follows: given n nonnegative numbers $V = \{a_1, a_2, \ldots, a_n\}$, find a vector $(\epsilon_1, \epsilon_2, \ldots, \epsilon_n) \in B_n$ such that $|\sum_{i=1}^n \epsilon_i a_i|$ is minimum.

Consider the following greedy-type algorithm $\mathcal{A}$ for MinPart. Initially sort the numbers such that $a_{\pi(1)} \geq a_{\pi(2)} \geq \cdots \geq a_{\pi(n)}$. Choose a constant $p > 0$ and fix $k = \lfloor p \log_2 n \rfloor$. Solve MinPart to optimality for $a_{\pi(1)}, a_{\pi(2)}, \ldots, a_{\pi(k)}$, i.e., find optimal values of $\epsilon_{\pi(1)}, \epsilon_{\pi(2)}, \ldots, \epsilon_{\pi(k)}$. (This can be trivially done in time $O(n^p)$.) Now for each $j > k$, if $\sum_{i=1}^{j-1} \epsilon_{\pi(i)} a_{\pi(i)} < 0$, then set $\epsilon_{\pi(j)} = 1$, and otherwise $\epsilon_{\pi(j)} = -1$.

Theorem 2.1 *The domination ratio of $\mathcal{A}$ is at least $1 - \Theta(\frac{1}{\sqrt{k}})$.*

Observe that if

$$\min |\sum_{i=1}^{k} \epsilon_{\pi(i)} a_{\pi(i)}| \geq \sum_{i=k+1}^{n} a_{\pi(i)},$$

where the minimum is taken over B_k, then $\mathcal{A}$ produces an optimal solution. Otherwise, it follows easily that the solution produced by $\mathcal{A}$ satisfies

$$|\sum_{i=1}^{n} \epsilon_{\pi(i)} a_{\pi(i)}| \leq a_{\pi(k+1)}.$$

To prove Theorem 2.1, we may thus assume that $a_1 \geq a_2 \geq \ldots \geq a_n$ and that the solution produced by the algorithm satisfies $|\sum_{i=1}^n \epsilon_i a_i| \leq a_{k+1}$. By the assumptions and the simple fact that $\binom{k}{\lfloor k/2 \rfloor}/2^k = \Theta(1/\sqrt{k})$, it suffices to prove the following:

Proposition 2.2 *The number of vectors $(\epsilon_1, \ldots, \epsilon_n) \in B_n$ for which $|\sum_{i=1}^n \epsilon_i a_i| < a_{k+1}$ is at most $\binom{k}{\lfloor k/2 \rfloor} 2^{n-k}$.*

Proof: We need the following lemma:

Lemma 2.3 *Let $a_1 \geq a_2 \geq \cdots \geq a_k$ and let (a, b) be an arbitrary open interval such that $b - a \leq 2a_k$. Then the number of vectors $(\delta_1, \ldots, \delta_k) \in B_k$ such that $\sum_{i=1}^k \delta_i a_i \in (a, b)$ is at most $\binom{k}{\lfloor k/2 \rfloor}$.*

This lemma is related to the one-dimensional Littlewood-Offord problem, and was first proved by Erdős in [10]. For completeness, we include the short and elegant proof. Let a vector $(\delta_1, \ldots, \delta_k) \in B_k$ correspond to the set $\{i : \delta_i = +1, 1 \leq i \leq k\}$. Observe that if, for a vector $(\delta_1, \ldots, \delta_k) \in B_k$, $\sum_{i=1}^k \delta_i a_i \in (a, b)$, then the change of the sign of any δ_i will move the sum from the interval (a, b). Thus, a pair of vectors for which the above sums are in (a, b) correspond to sets neither of which contains the other one. Hence, by Sperner's Lemma [23], there are at most $\binom{k}{\lfloor k/2 \rfloor}$ vectors $(\delta_1, \ldots, \delta_k) \in B_k$ such that $\sum_{i=1}^k \delta_i a_i \in (a, b)$. □

We proceed with the proof of the proposition. Fix a vector $(\epsilon_{k+1}, \ldots, \epsilon_n) \in B_{n-k}$. Denote the sum $\sum_{i=k+1}^n \epsilon_i a_i$ by S. Now $|\sum_{i=1}^n \epsilon_i a_i| < a_{k+1}$ if and only if $\sum_{i=1}^k \epsilon_i a_i$ belongs to the open interval $(-S - a_{k+1}, -S + a_{k+1})$. However, by the lemma above, there are at most $\binom{k}{\lfloor k/2 \rfloor}$ vectors $(\epsilon_1, \ldots, \epsilon_k)$ with the above property. Since we can fix $(\epsilon_{k+1}, \ldots, \epsilon_n) \in B_{n-k}$ in $|B_{n-k}| = 2^{n-k}$ ways, the assertion of the proposition follows, implying the assertion of the theorem as well. □

3 The MaxCut Problem

Let $G = (V, E)$ be an undirected graph on $|V| = n$ vertices, and let $w : E \to R^+$ be a weight function associating a nonnegative real weight $w(e)$ to each edge $e \in E$. The MaxCut problem is the problem of finding a cut $V = A \cup B$, $A \cap B = \emptyset$ of maximum possible total weight $w(A, B) = \sum_{a \in A, b \in B, ab \in E} w(a, b)$. In this section we prove that there is an absolute positive constant $c > 1/40$ such that for any graph G and any weight function w, at least a fraction c of all cuts of G have weight at most $w(E)/2$, where $w(E) = \sum_{e \in E} w(e)$ is the total weight of all edges of G. Therefore, any algorithm that finds a cut of weight at least $w(E)/2$ has domination ratio at least c.

There are many simple algorithms that produce cuts of weight at least $w(E)/2$. The simplest is probably the greedy algorithm; order the vertices arbitrarily and put each vertex in its turn either in A or in B in order to maximize in each step the total weight of crossing edges. The running time of this algorithm is obviously linear in the number of edges of G.

Another simple algorithm producing a cut of weight at least $w(E)/2$ can be obtained using a small sample space supporting n pairwise independent uniform 0/1 random variables. Since such spaces (as well as similar ones supporting k-wise independent random variables) will be applied later in the paper as well, we briefly describe their properties in the following paragraph.

By a construction described, e.g., in Section 15.2 of [4], there exists a 0/1 matrix $S^{(k)} = [s_{ij}]$ with n columns and $r = O(n^{\lfloor k/2 \rfloor})$ rows such that every $r \times k$ submatrix of $S^{(k)}$ contains an equal number of all 2^k binary k-vectors. This matrix can be constructed by a deterministic polynomial algorithm; moreover, it can be constructed by a parallel NC algorithm. The matrix is a simple explicit construction of a sample space of size r that supports n k-wise independent uniform 0/1-variables. Indeed, the points of the sample space are the rows, each of them has probability $1/r$, and the value of variable number j on the point i is s_{ij}.

In polynomial time (and poly-logarithmic parallel time), we can compute the weight of the cut corresponding to each row of the $r \times n$ matrix $S^{(2)}$ and choose the one with the maximum weight. By the submatrix property of $S^{(2)}$, every edge of G appears exactly in $r/2$ such cuts. Therefore, the total weight of the r cuts is $\frac{r}{2} w(E)$ and the cut of maximum weight has weight at least $w(E)/2$.

In fact, a more careful analysis enables us to get in this way weight at least $(1 + \Omega(\frac{1}{n}))w(E)/2$. More interesting is the fact that as a by-product of the analysis below we can get that there is an absolute positive constant c' such that at least a fraction c' of the cuts have weight that exceeds the average weight $w(E)/2$ by at least c' standard deviations. Moreover, we can find such a cut deterministically in polynomial time.

We proceed with the proofs, starting with a few probabilistic lemmas. The first lemma is standard, and appears, for example, in [2].

Lemma 3.1 *For every real random variable X with finite and positive forth moment,*

$$E(|X|) \geq \frac{E(X^2)^{3/2}}{E(X^4)^{1/2}}.$$

Proof: By Hölder's Inequality with $p = 3/2$ and $q = 3$, for every two nonnegative random variables Y, Z,

$$E(YZ) \leq (E(Y^p))^{1/p} (E(Z^q))^{1/q}.$$

The result follows by applying this inequality to $Y = |X|^{2/3}$ and $Z = |X|^{4/3}$. □

Using this lemma, we prove the following result, in which the absolute constants can, in fact, be slightly improved.

Lemma 3.2 *Let X be a real random variable and suppose that its first, second and forth moments satisfy $E(X)=0$, $E(X^2)=\sigma^2>0$ and $E(X^4)\le b\sigma^4$. Then*
(i) The probability that X is positive satisfies: $Prob(X>0)\ge \frac{1}{2^{4/3}b}$, (and a similar inequality holds, by symmetry, for the probability that X is negative).
(ii)

$$Prob(X>\frac{\sigma}{4\sqrt{b}})\ge \frac{1}{4^{4/3}b},$$

(and a similar inequality holds, by symmetry, for $Prob(X<-\frac{\sigma}{4\sqrt{b}})$.)

Proof: By Lemma 3.1, $E(|X|)\ge\frac{\sigma}{\sqrt{b}}$. Since $E(X)=0$ it follows that

$$Prob(X>0)E(X|X>0)\ge\frac{\sigma}{2\sqrt{b}}. \tag{1}$$

Put $p=Prob(X>0)$. Then, the conditional expectation $E(X|X>0)$ is at least $\frac{\sigma}{2p\sqrt{b}}$ and hence, by Jensen's Inequality, $E(X^4|X>0)\ge\frac{\sigma^4}{16p^4b^2}$, implying that

$$b\sigma^4\ge E(X^4)\ge Prob(X>0)E(X^4|X>0)\ge\frac{\sigma^4}{16p^3b^2}.$$

Therefore, $p\ge\frac{1}{2^{4/3}b}$, establishing part (i).

The proof of part (ii) is similar. Put $\delta=Prob(X>\frac{\sigma}{4\sqrt{b}})$. Then

$$Prob(X>0)E(X|X>0)\le(1-\delta)\frac{\sigma}{4\sqrt{b}}+\delta E(X|X>\frac{\sigma}{4\sqrt{b}}),$$

implying, by (1), that

$$E(X|X>\frac{\sigma}{4\sqrt{b}})\ge\frac{\sigma}{4\sqrt{b}\delta}.$$

By Jensen's Inequality this shows that

$$E(X^4|X>\frac{\sigma}{4\sqrt{b}})\ge\frac{\sigma^4}{4^4b^2\delta^4}.$$

Therefore

$$b\sigma^4\ge E(X^4)\ge\delta E(X^4|X>\frac{\sigma}{4\sqrt{b}})\ge\frac{\sigma^4}{4^4b^2\delta^3},$$

implying the assertion of part (ii). $\square$

Returning to the MaxCut problem for a weighted graph $G=(V,E)$ on a set $V=\{1,2,\ldots,n\}$ of n vertices, with weight function $w:E\to R^+$, let $\epsilon_1,\epsilon_2,\ldots,\epsilon_n$ be independent random variables, each distributed uniformly on $\{-1,1\}$. Let X be the random variable

$$X=\sum_{ij\in E}w(i,j)\epsilon_i\epsilon_j.$$

Notice that if we associate each possible vector $(\epsilon_1,\epsilon_2,\ldots,\epsilon_n)$ with the cut determined by $A=\{i:\epsilon_i=1\}$ and $B=\{i:\epsilon_i=-1\}$, then the value of X is precisely $w(E)-2w(A,B)$. Therefore, the fraction of cuts of G whose weight is at most $w(E)/2$ is precisely the fraction of vectors ϵ_i for which the random variable X is nonnegative, that is, precisely the probability that X is nonnegative. To show that this probability is bounded away from zero we apply part (i) of Lemma 3.2. In order to do so we have to bound the moments of X as follows.

Lemma 3.3 *The random variable X defined above satisfies:*
(i) $E(X) = 0$.
(ii) $E(X^2) = \sum_{ij \in E} w^2(i,j)$.
(iii) $E(X^4) \leq 15(E(X^2))^2$.

Proof: The assertions of part (i) and of part (ii) follow easily, by linearity of expectation. To prove the assertion of part (iii) expand, first, X^4, and write it as a sum of $|E|^4$ products, where each product is a product of four terms of the form $w(i,j)\epsilon_i\epsilon_j$. By linearity of expectation, $E(X^4)$ is the sum of expectations of all these terms. Note that if some ϵ_i raised to an odd power appears in a term, then its contribution to the expectation is 0. Therefore, the only terms that have a nonzero contribution are those corresponding to collections of (not necessarily distinct) fourtuples of edges that cover each vertex an even number of times. The only fourtuples of this form are the $|E|$ fourtuples corresponding to the same edge taken four times, the $\binom{|E|}{2}$ fourtuples consisting of a pair of edges, each taken twice, and the fourtuples corresponding to four-cycles C_4 in the graph. Let $\mathcal{C}$ denote the set of all copies of C_4 in G, put $m = |E|$ and let $e_1, e_2, \ldots, e_m$ be an arbitrary enumeration of all edges of G. By the discussion above

$$E(X^4) = \sum_{i=1}^{m} w^4(e_i) + \binom{4}{2} \sum_{1 \leq i < j \leq m} w^2(e_i)w^2(e_j) + 4! \sum_{C \in \mathcal{C}} \prod_{e \in C} w(e).$$

For each 4-cycle C whose edges are e_i, e_j, e_k, e_l in this order,

$$w(e_i)w(e_j)w(e_k)w(e_l) \leq \frac{w^2(e_i)w^2(e_k) + w^2(e_j)w^2(e_l)}{2}.$$

Summing over all members of $\mathcal{C}$, for each pair of nonadjacent edges e_i, e_k, the term $w^2(e_i)w^2(e_k)$ appears at most twice in the right hand side, whereas the corresponding term for a pair of adjacent edges does not appear at all. It thus follows that

$$\sum_{C \in \mathcal{C}} \prod_{e \in C} w(e) \leq \sum_{1 \leq i < k \leq m} w^2(e_i)w^2(e_k).$$

Altogether we conclude that

$$E(X^4) \leq \sum_{i=1}^{m} w^4(e_i) + 30 \sum_{1 \leq i < j \leq m} w^2(e_i)w^2(e_j) \leq 15(E(X^2))^2,$$

completing the proof. $\Box$

Combining the last lemma with part (i) of Lemma 3.2, it follows that the probability that X is nonnegative is at least $\frac{1}{2^{4/3}15} > \frac{1}{40}$. We have thus proved the following.

Theorem 3.4 *There exists a linear time, deterministic approximation algorithm for the MaxCut problem in a weighted graph whose domination ratio exceeds* $1/40$. *There is also an NC algorithm with the same domination ratio.*

Remark: Note that the proof of Lemma 3.3 works even when the random variables ϵ_i are 8-wise independent, rather than fully independent. As there are polynomial size sample spaces that support such random variables (like the matrix $S^{(k)}$ mentioned in the beginning of this section), it follows, by Lemma 3.3 and part (ii) of Lemma 3.2 that we can also find, in any weighted graph G, a cut whose weight exceeds the average by $\frac{\sqrt{\sum_{e \in E} w^2(e)}}{8\sqrt{15}}$ in deterministic polynomial time, and in fact even in NC.

4 Sparse linear equations

Let L be a system of m linear equations in n variables $x_1, x_2, \ldots, x_n$ over $GF(2)$, and suppose that each equation e has a positive weight $w(e)$. The problem MaxLin is that of finding an assignment of values to the variables x_i, trying to maximize the weight of the equations satisfied.

Note that the MaxCut problem is a special case of MaxLin, where for each edge $e = ij$ we have the equation $x_i + x_j = 1$ with weight $w(e)$. As is the case with the MaxCut problem, it is easy to obtain, deterministically, in linear time, a solution whose weight is at least half the total weight; simply assign each variable in its turn the value that contributes as much as possible to the total weight of the equations determined so far. It turns out that when there is a uniform bound, say r, on the number of variables participating in each equation, the domination ratio of this algorithm is already at least some $c(r) > 0$. The proof is similar to the one in the previous section; but since the constants we get here are far worse, we included the discussion for MaxCut separately.

Our treatment here is similar to the one of Håstad and Venkatesh in [18], with one crucial change that enables us to bound the domination ratio away from zero. This also supplies, in fact, an improvement on the success probability of the algorithm of [18].

Note, first, that we may assume that the left hand sides of each pair of our equations are different, since if we have for a subset I of $\{1, 2, \ldots, n\}$ the equation $\sum_{i \in I} x_i = 0$ with weight w_0 and the equation $\sum_{i \in I} x_i = 1$ with weight w_1, we can simply replace this pair by the equation whose weight is bigger, modifying its new weight to be the difference between the two old ones.

We thus assume that all the left hand sides of the equations are pairwise distinct. Let $I_j \subset \{1, 2, \ldots, n\}$ be the set of indices of the variables participating in equation number j, and let $\delta_j \in \{0, 1\}$ be the right hand side of this equation. Let w_j denote the weight of equation number j and define

$$X = \sum_{j=1}^{m} (-1)^{\delta_j} w_j \prod_{i \in I_j} \epsilon_i,$$

where $\epsilon_i \in \{-1, 1\}$. Let W denote the total weight of all equations. Then, the total weight of the equations satisfied by putting $x_i = 0$ if $\epsilon_i = 1$ and $x_i = 1$ if $\epsilon_i = -1$ for all i is clearly $\frac{1}{2}(X(\epsilon_1, \epsilon_2 \ldots, \epsilon_n) + W)$. Therefore, in order to bound the domination ratio of the simple greedy algorithm it suffices to lower bound the probability that X is negative, when the variables ϵ_i range over all 2^n possibilities in $\{-1, 1\}$ with equal probability. This is done by applying Lemma 3.2. In order to do that we need a known extension of Khinchin's Inequality. Versions of this inequality appear in various places including [7], [24] and [11]. Here we use the following version, proved by Bourgain in [7]. The same lemma has been applied in [18] in a similar context.

Lemma 4.1 ([7]) *Let $f = f(x_1, \ldots, x_n)$ be a polynomial of degree r in n variables $x_1, \ldots, x_n$ with domain $\{-1, 1\}$. Define a random variable X by choosing a vector $(\epsilon_1, \ldots, \epsilon_n) \in \{-1, 1\}^n$ uniformly at random and setting $X = f(\epsilon_1, \ldots, \epsilon_n)$. Then, for every $p \geq 2$, there is a constant c_p such that $(E(|X|^p))^{1/p} \leq (c_p)^r (E(X^2))^{1/2}$. In particular, $c_4 \leq 2^{3/2}$.*

Plugging $b = 2^{6r}$ in Lemma 3.2, part (i), we conclude that by the above discussion, the domination ratio of the simple greedy algorithm described above is at least $\frac{1}{2^{4/3} \cdot 2^{6r}}$. Moreover, r-wise independent random variables suffice to get a solution satisfying at least half the total weight. Thus, we conclude that the following holds.

Theorem 4.2 *There exists a linear time, deterministic approximation algorithm for the MaxLin problem for a weighted set of linear equations over $GF(2)$ in which each equation contains at most r variables, whose domination ratio exceeds $\frac{1}{2^{4/3} \cdot 2^{6r}}$. There is also an NC algorithm with the same domination ratio.*

Remark: By part (ii) of Lemma 3.2 we conclude that in the above notation, if $\sigma = \sqrt{\sum_j w_j^2}$, then the probability that X exceeds $\frac{\sigma}{2^{3r+2}}$ is at least $\frac{1}{4^{4/3} \cdot 2^{6r}}$. This improves the estimate in [18] that shows that the probability X exceeds $\sigma/2^{3+3r}$ is at least $\frac{1}{2^{3+3r}m}$. The main difference is that for fixed r the probability bound obtained here is bounded away from zero. Moreover, since the analysis works whenever the random variables ϵ_i are $4r$-wise independent, we can find such an assignment for the variables in deterministic polynomial time, and in fact even in NC.

5 Constraint satisfaction problems

Let r be a fixed positive integer, and let $\mathcal{F} = \{f_1, f_2, \ldots, f_m\}$ be a collection of Boolean functions, each involving at most r of the n variables, and each having a real, positive weight $w(f_i)$. The Max-r-Constraint Satisfaction Problem (or Max-r-CSP problem, for short), is the problem of finding a truth assignment to the variables so as to maximize the total weight of the functions satisfied. Note that this includes, as a special case, the problem of sparse linear equations considered in the previous section (and hence also the MaxCut problem). Another interesting special case is the Max-r-Sat problem, in which each of the Boolean functions f_i is a clause of at most r literals. In this section we observe that our techniques can handle this more general problem without any essential changes. The first observation is that as before, it is easy to find deterministically a truth assignment in which the total weight of the satisfied equations is at least the expected value of this weight when the values of the variables are assigned at random. This can be done either by the method of conditional expectations, or by using small sample spaces with r-wise independent random variables. It thus suffices to show that for every fixed r, there is some $c(r) > 0$ so that at least a fraction $c(r)$ of all feasible assignments do not supply a better solution than the average. As before, this will be done by applying Lemma 3.2.

We proceed with the details. For each Boolean function f of r Boolean variables

$$x_{i_1}, x_{i_2}, \ldots, x_{i_r},$$

with a weight $w(f)$, define a random variable X_f as follows. For convenience, suppose each variable x_{i_j} attains values in $\{-1, 1\}$. Let $V \subset \{-1, 1\}^r$ denote the set of all satisfying assignments of f. Then

$$X_f(x_1, x_2, \ldots, x_n) = w(f) \sum_{v=(v_1,\ldots,v_r) \in V} \frac{\prod_{j=1}^r (1 + x_{i_j} v_j)}{2^r} \; - \; w(f) \frac{|V|}{2^r}.$$

This is clearly a random variable defined over the space $\{-1, 1\}^n$ and its value at $x = (x_1, x_2, \ldots, x_n)$ is $w(f)(1 - \frac{|V|}{2^r})$ if x satisfies f, and is $-w(f)\frac{|V|}{2^r}$ otherwise. It follows that the expectation of X_f is zero. Define now $X = \sum_{f \in \mathcal{F}} X_f$. Then the value of X at $x = (x_1, x_2, \ldots, x_n)$ is precisely the total weight of the functions satisfied by the truth assignment x, minus the average value of this total weight, computed over all possible truth assignments. It thus suffices to show that with probability at least $c(r) > 0$, the value of X is negative.

By linearity of expectation, $E(X) = 0$. Moreover, since X is a polynomial of degree at most r in $x_1, x_2, \ldots, x_n$, it follows, by Lemma 4.1, that $E(X^4) \leq 2^{6r} E(X^2)^2$. The desired result with $c(r) = \frac{1}{2^{4/3} \cdot 2^{6r}}$ now follows from Lemma 3.2, part (i). We have thus proved the following.

Theorem 5.1 *For each fixed integer $r \geq 1$ there exists a linear time, deterministic approximation algorithm for the Max-r-CSP problem for a weighted set of Boolean functions in which each function depends on at most r variables, whose domination ratio exceeds $\frac{1}{2^{4/3} \cdot 2^{6r}}$. There is also an NC algorithm with the same domination ratio.*

6 Concluding remarks

There are very simple efficient *randomized* algorithms with expected domination ratios close to 1 for all the problems considered here (and for many other ones). Simply pick randomly and independently, with repetitions, enough feasible solutions, and choose the best one. This can be done for every problem in which a random feasible solution can be generated efficiently. Clearly, if we pick t random solutions, then the probability the best one is worse than an ϵ-fraction of the solutions is at most $(1-\epsilon)^t$. For a problem of size n and any $\epsilon \geq n^{-O(1)}$ we can choose a polynomial t for which this probability will be inverse polynomially small. It is therefore unlikely to expect any strong inapproximability results for any of these problems with respect to the domination criterion, as all of them admit efficient randomized algorithms.

6.1 Unweighted graphs

It is sometimes simpler to get efficient algorithms with large domination ratio for unweighted graphs than for the weighted case. Consider, for example, the unweighted MaxCut problem. Let $G = (V, E)$ be a t-colorable graph with $|E| = m$ edges. Fix a t-coloring $c = (C_1, \ldots, C_t)$ of G. We form a random partition of V by choosing a set $I \subset \{1, \ldots, t\}$ of size $|I| = \lfloor t/2 \rfloor$ uniformly at random and by defining $A = \bigcup_{i \in I} C_i$, $B = V \setminus A$. It is easy to see that each edge $e \in E$ crosses between A and B with probability $2\binom{t-2}{\lfloor \frac{t}{2} \rfloor - 1} \Big/ \binom{t}{\lfloor \frac{t}{2} \rfloor} \geq \frac{1}{2} + \frac{1}{2t}$. Hence the expected number of edges in the cut $[A, B]$ is $m/2 + m/(2t)$. This argument (mentioned, e.g., in [1]) can be easily derandomized using standard derandomization techniques, to give a polynomial time deterministic algorithm for finding a bipartite cut of value at least $m/2 + m/(2t)$ in any graph G with m edges that can be colored (in deterministic polynomial time) in t colors.

As the greedy (first fit) algorithm outputs a coloring in which each pair of color classes is connected by an edge, we conclude that the number t of colors used by it satisfies: $\binom{t}{2} \leq m$. Substituting this estimate into the above bound for the MaxCut problem, we get a deterministic algorithm producing a cut of size $m/2 + \sqrt{m/8} + O(1)$ – the well-known result of Edwards [8], [9]. Since the average size of a cut is $m/2$ and its variance is $m/4$, it can be shown that the domination ratio of this algorithm is at least $1/3 - o(1)$. This follows from the simple fact below, whose proof is similar to (and somewhat simpler than) that of Lemma 3.2.

Proposition 6.1 *Let X be a real random variable satisfying $E(X) = 0$, $E(X^2) = \sigma^2$ (> 0). Then for every real $a > 0$, $Prob(X \geq a\sigma) \leq \frac{1}{a^2+1}$.*

Proof: Put $p = Prob(X \geq a\sigma)$. Since $E(X) = 0$ it follows that $(1-p)E(|X||X < 0) \geq pa\sigma$, implying that $E(|X||X < 0) \geq \frac{pa\sigma}{1-p}$. By Jensen's Inequality, $E(X^2|X < 0) \geq \frac{p^2 a^2 \sigma^2}{(1-p)^2}$ and hence

$$\sigma^2 = E(X^2) \geq \frac{p^2 a^2 \sigma^2}{(1-p)} + p a^2 \sigma^2.$$

Therefore $p \leq \frac{1}{a^2+1}$, as needed. $\square$

It is not difficult to show that the assertion of the proposition above is tight for every value of $a > 0$. In our case, $a = \frac{1}{\sqrt{2}} + o(1)$, implying that the domination ratio of the above described MaxCut algorithm for unweighted graphs is at least $1/3 - o(1)$.

For many classes of graphs the trivial upper bound on the constructive chromatic number $\binom{t}{2} \leq m$ can be improved to $t = o(\sqrt{m})$, resulting in bipartite cuts of size $m/2 + \omega(1)\sqrt{m}$ and thus in algorithms with domination ratio $1 - o(1)$, that follows from Chebyshev's Inequality. This is the case, for example, for H-free graphs, for any fixed graph H (see [3] for more details), or for graphs with m edges and with clique number bounded by $m^{o(1)}$, as follows by standard Ramsey-type arguments. Note also that for almost all graphs on n vertices, the greedy algorithm finds a coloring in $O(n/\log n)$ colors [12], while the number of edges is quadratic in n. Thus we get an almost sure $1 - o(1)$ domination ratio algorithm here as well.

The authors of [20] show how to find, in polynomial time, given a tournament on n vertices, an ordering of the vertices consistent with at least $\frac{1}{2}\binom{n}{2} + \Omega(n^{3/2})$ arcs (that is, an acyclic subdigraph with that many arcs). Here, too, it is not difficult to show, using Chebyshev's Inequality, that the domination ratio of this algorithm (for the problem of finding an ordering with the maximum possible number of consistent arcs) is $1 - o(1)$. We omit the details.

6.2 Sparse linear binary codes

Some of the arguments here supply interesting results for linear binary codes. These are stated and proved below.

Proposition 6.2 *Let $A = (a_{ij})$ be a $k \times n$ generating matrix of a binary linear code $\mathcal{C}$ of length n and dimension k. Suppose the Hamming weight of each column of A is at most r, and no two columns are identical. Then the covering radius of the code is at most $n/2 - \sqrt{n}/2^{3r+3}$. Moreover, for every binary word $v = (v_1, v_2, \ldots, v_n)$ of length n, at least $\frac{2^k}{4^{4/3} \cdot 2^{6r}}$ of the codewords of $\mathcal{C}$ are within distance $n/2 - \sqrt{n}/2^{3r+3}$ from v.*

Proof: This is a simple consequence of the last remark in Section 4. Simply apply that remark to the linear equations $\sum_{i=1}^{k} a_{ij} x_j = v_j, \quad 1 \leq j \leq n.$ $\square$

The dual binary Hamming code shows that the assertion of the theorem fails without the assumption that the weights of all columns are bounded. The code with $n = \binom{k}{2}$ in which the columns of A are all possible binary k-vectors of Hamming weight 2 shows that the above estimate is essentially tight, as the maximum weight of a codeword is $\lfloor k^2/4 \rfloor = n/2 + \sqrt{n}/\sqrt{8} + \Theta(1)$.

Proposition 6.3 *For every integer $p > 2$ there is an absolute constant $c_p > 0$ such that the following holds. Let $H = (h_{ij})$ be an $(n-k) \times n$ parity check matrix of a binary linear code $\mathcal{C}$ of length n and dimension k. Suppose the Hamming weight of each column of A is at most r, and no two columns are identical. Then the number of codewords of Hamming weight p in $\mathcal{C}$ is at most $\frac{c_p^{rp}}{p!} n^{p/2}$.*

Proof: Let I_j be the set of all indices i such that $h_{ij} = 1$, and consider

$$X(\epsilon_1, \epsilon_2, \ldots, \epsilon_{n-k}) = \sum_{j=1}^{n} \prod_{i \in I_j} \epsilon_i.$$

When ϵ_i's take random, independent values in $\{-1,1\}^{n-k}$, X becomes a random variable. By linearity of expectation, $E(X^2)=n$. Therefore, by Lemma 4.1, $E(X^p) \le (c_p^{rp})n^{p/2}$. However, by linearity of expectation $E(X^p)$ is exactly the number of ordered p-tuples of columns of H whose sum modulo 2 is the zero vector, and each codeword of weight p of $\mathcal{C}$ is counted in this way $p!$ times. $\Box$

The binary Hamming code shows that the conclusion is false without the assumption that the columns are of bounded Hamming weight. The matrix H in which $n = \binom{n-k}{2}$, where the columns are all distinct binary vectors of Hamming weight 2 shows that the result is essentially tight, since the complete graph with n edges has $c(p)n^{p/2}$ subgraphs with p edges and all degrees even, for every fixed p.

6.3 Open problems

- A natural variant of the MaxCut algorithm based on small sample spaces with pairwise (or 4-wise) independent random variables, is to consider all cuts given by a sample space with k-wise independent random variables, and take the best one. It seems plausible that as k grows to infinity, the domination ratio of this algorithm tends to 1. Similar algorithms can be considered for sparse linear equations and for the Max-r-CSP problem, and it will be interesting to decide if by taking the best point in a space supporting k-wise independent random variables for k bigger than what's needed to get the average, we can obtain significantly better domination ratios.

- It was proved in [14] that every Asymmetric TSP (ATSP) algorithm, which always produces a tour of weight not worse than the average weight of a tour, is of domination ratio at least $1/(n-1)$ for all $n \neq 6$. This cannot be significantly improved due to the following simple example. Consider a complete digraph on n vertices in which all arcs apart from two arcs xy and xz are of weight 0. The weight of xy is 1 and that of xz is n. The average weight of a tour is $(n+1)/(n-1)$. Thus, a tour T through xy is of weight smaller than the average, yet T dominates only $2(n-2)!$ tours.

  Since it is not difficult to find, efficiently and deterministically, a tour of weight at most the average, it follows that there are several polynomial time ATSP algorithms, see, e.g., [14, 21], of domination ratio $\Omega(1/n)$. It will be interesting to improve this ratio to $\Omega(1)$.

Acknowledgment We would like to thank Avi Wigderson for helpful comments. Part of the work was carried out while the second author was visiting Tel Aviv University in December 2002 - January 2003.

References

[1] N. Alon, Bipartite subgraphs, Combinatorica 16 (1996), 301-311.

[2] N. Alon, Voting paradoxes and digraphs realizations, Advances in Applied Math. 29 (2002), 126-135.

[3] N. Alon, B. Bollobás, M. Krivelevich and B. Sudakov, Maximum cuts and judicious partitions in graphs without short cycles, J. Combinatorial Theory, Ser. B 88 (2003), 329-346.

[4] N. Alon and J.H. Spencer, *The Probabilistic Method*, 2nd edition, Wiley, New York, 2000.

[5] D. Ben-Arieh, G. Gutin, M. Penn, A. Yeo and A. Zverovitch, Transformations of Generalized ATSP into ATSP: experimental and theoretical study. Oper. Res. Lett. 41 (2003), 2581-2596.

[6] D. Berend and S.S. Skiena, Combinatorial dominance guarantees for heuristic algorithms. Manuscript, 2002.

[7] J. Bourgain, Walsh subspaces of ℓ_p product spaces, In: Seminaire D'Analyse Fonctionnelle, Ecole Polytechnique, Centre De Mathematiques, 1979-80, pp. 4.1–4.9.

[8] C. S. Edwards, Some extremal properties of bipartite subgraphs, Canad. J. Math. 3 (1973), 475-485.

[9] C. S. Edwards, An improved lower bound for the number of edges in a largest bipartite subgraph, Proc. 2^{nd} Czechoslovak Symposium on Graph Theory, Prague, (1975), 167-181.

[10] P. Erdős, On a lemma of Littlewood and Offord, Bull. Amer. Math. Soc. 51 (1945), 898–902.

[11] E. Friedgut and V. Rödl, Proof of a hypercontractive estimate via entropy, Israel J. Math. 125 (2001), 369–380.

[12] G. Grimmett and C. McDiarmid, On colouring random graphs, Math. Proc. Cam. Phil. Soc. 77 (1975), 313–324.

[13] G. Gutin, A. Vainshtein and A. Yeo, Domination analysis of combinatorial optimization problems. To appear in Discrete Appl. Math.

[14] G. Gutin and A. Yeo, Polynomial approximation algorithms for the TSP and the QAP with a factorial domination number. Discrete Appl. Math. 119 (2002), 107–116.

[15] G. Gutin and A. Yeo, Anti-matroids. Oper. Res. Lett. 30 (2002), 97–99.

[16] G. Gutin, A. Yeo and A. Zverovitch, Exponential Neighborhoods and Domination Analysis for the TSP. In *The Traveling Salesman Problem and its Variations* (G. Gutin and A.P. Punnen, eds.), Kluwer, Dordrecht, 2002.

[17] D.S. Johnson, G. Gutin, L. McGeoch, A. Yeo, X. Zhang, and A. Zverovitch, Experimental Analysis of Heuristics for ATSP. In *The Traveling Salesman Problem and its Variations* (G. Gutin and A. Punnen, eds.), Kluwer, Dordrecht, 2002.

[18] J. Håstad and S. Venkatesh, On the advantage over a random assignment, STOC 2002, 43–52.

[19] A.E. Koller and S.D. Noble, Domination analysis of greedy heuristics for the frequency assignment problem. To appear in Discrete Math.

[20] S. Poljak, V. Rödl and J. Spencer, Tournament ranking with expected profit in polynomial time, SIAM J. Discrete Math. 1 (1988), 372–376.

[21] A.P. Punnen, F. Margot and S.N. Kabadi, TSP heuristics: domination analysis and complexity, Algorithmica 35 (2003), 111–127.

[22] A.P. Punnen and S. Kabadi, Domination analysis of some heuristics for the asymmetric traveling salesman problem, Discrete Appl. Math. 119 (2002), 117–128.

[23] E. Sperner, Ein Satz über Untermegen einer endlichen Menge, Math. Zeitschr. 27 (1928), 544–548.

[24] M. Talagrand, On Russo's approximate $0-1$-law, Ann. Prob. 22 (1994), 1376–1387.