On Some Tighter Inapproximability Results, Further Improvements*

Piotr Berman† Marek Karpinski‡

Abstract

Improved inaproximability results are given, including the best up to date explicit approximation thresholds for bounded occurence satisfiability problems, like MAX-2SAT and E2-LIN-2, and problems in bounded degree graphs, like MIS, Node Cover and MAX CUT. We prove also for the first time inapproximability of the problem of Sorting by Reversals and display an explicit approximation threshold for this problem.

Key words: Approximation Algorithms, Approximation Hardness, Bounded Dependency Satisfiability, Breakpoint Graphs, Independent Set, Node Cover, MAX-CUT, Sorting by Reversals.

*A preliminary version of this paper appeared in ECCC TR98-029 (1998).

†Dept. of Computer Science, Pennsylvania State University, University Park, PA16802. Supported in part by NSF grant CCR-9700053. Email: berman@cse.psu.edu

‡Dept. of Computer Science, University of Bonn, 53117 Bonn. Supported in part by the International Computer Science Institute, Berkeley, California, by DFG grant 673/4-1, ESPRIT BR grants 7079, 21726, and EC-US 030, by DIMACS, and by the Max–Planck Research Prize. Email: marek@cs.uni-bonn.de

1 Introduction

The paper studies *explicit* approximation thresholds for bounded dependency, and bounded degree optimization problems. There was a dramatic progress recently in proving tight inapproximability results for a number of NP-hard optimization problems (cf. [H96], [H97], [TSSW96]). The goal of this paper is to develop a new method of reductions for attacking bounded instances of the NP-hard optimization problems and also other optimization problems. The method uses randomized reductions and applies to the number of problems including Maximum Independent Set in graphs of degree d (d-MIS), bounded degree Minimum Node Cover (d-Node Cover), bounded degree MAX CUT (d-MAX CUT) and bounded occurrence MAX-2SAT (d-OCC-MAX-2SAT), (cf. [PY91], [A94], [BS92], [BF94], [BF95], [AFWZ95]). This yields also the first explicit approximation lower bounds for the small degree graph problems, and the small dependency satisfiability. Independently, we apply this method to prove for the first time approximation hardness of the problem of *sorting by reversals*, MIN-SBR, motivated by molecular biology [HP95], and proven only recently to be NP-hard [C97]. Interestingly, its signed version can be computed in polynomial time [HP95], [BH96], [KST97].

The core of the new method is the use of restricted versions of the E2-LIN-2 and E3-LIN-2 problems studied in [H97]. We denote by E2-LIN-2 the problem of maximizing the number of satisfied equations from a given set of linear equations mod 2 with exactly 2 variables per equation. E3-LIN-2 is a similar problem with three variables per equation. E2-LIN-2 can be viewed as a graph problem in the following way: each variable is a node, and an equation $x \oplus y = b$ is an edge $\{x, y\}$ with label b. (Note that the special case when all edges have label 1 constitutes MAX CUT problem.)

We denote by d-OCC-E2-LIN-2 and d-OCC-E3-LIN-2 the versions of these problems where the number of occurences of each variable is bounded by d (note that in d-OCC-2-LIN-2 can be also viewed as restricted to graphs of degree d).

The rest of the paper proves the following main theorem:

Theorem 1. *For every $\epsilon > 0$, it is NP-hard to approximate*

(i) 3-OCC-E2-LIN-2 and 3-MAX CUT within factor $332/331 - \epsilon$;

(ii) 6-OCC-MAX 2SAT within factor $668/667 - \epsilon$;

(iii) 3-OCC-E3-LIN-2 within factor $62/61 - \epsilon$;

(iv) 4-MIS within factor $74/73 - \epsilon$ and 4-Node Cover within $79/78 - \epsilon$;

(v) 3-MIS within factor $140/139 - \epsilon$ and 3-Node Cover within $145/144 - \epsilon$;

(vi) MIN-SBR within factor $1237/1236 - \epsilon$.

Our proof can be easily extended to provide explicit inapproximability factors for many other optimizations problems that are related to bounded degree graphs. E. g. we get also 1.0149 lower bound for 5-MIS, 1.0138 lower bound for 5-NodeCover, and 1.0005 lower bound for 3-OCC-MAX 2SAT. We provide proof sketches in Sections 4, and 7.

The technical core of all these results is the reduction to show *(i)*, which forms structures that can be translated into many graph problems with the very small and natural gadgets. The best to our knowledge gaps between the upper and lower approximation bounds are summarized in Table 1. The upper approximation bounds are from [GW94], [BF95], [C98], and [FG95].

Problem	Approx. Upper	Approx. Lower
3-OCC-E2-LIN-2	1.1383	1.0030
3-OCC-E3-LIN-2	2	1.0163
3-MAX CUT	1.1383	1.0030
3-OCC-MAX 2SAT	1.0741	1.0005
6-OCC-MAX-2SAT	1.0741	1.0014
3-MIS	1.2	1.0071
4-MIS	1.4	1.0136
5-MIS	1.6	1.0149
3-Node Cover	1.1666	1.0069
4-Node Cover	1.2857	1.0128
5-Node Cover	1.625	1.0138
MIN-SBR	1.5	1.0008

Table 1: Gaps between known approximation bounds.

2 Sequence of reductions

We start from E2-LIN-2 problem that was most completely analyzed by Håstad [H97] who proved that it is NP-hard to approximate it within a factor $12/11 - \epsilon$. In the sequel we will use notation of this paper. In this problem we are given a (multi)set of linear equations over $\mathbf{Z}_2$ with at most two variable per equation, and we maximize the size of a consistent subset.

In this paper, we prefer to interpret it as the following graph problem. Given an undirected graph $G = \langle V, E, l \rangle$ where l is a 0/1 edge labelling function. For $S \subset V$, $Cut(S)$ is the set of edges with exactly one endpoint in S (as in the MAX CUT problem). We define $Score(S, e) \in \{0, 1\}$ as follows: $Score(S, e) = l(e)$ iff $e \in Cut(S)$. In turn, $Score(S) = \sum_{e \in E} Score(S, e)$. The objective of E2-LIN-2 is to maximize $Score(S)$.

Our first reduction will have instance transformation τ_1, and will map an instance G of E2-LIN-2 into another instance G' of the same problem that has three properties: G' is a graph of degree 3, its girth (the length of a shortest cycle) is $\Omega(\log n)$, and its set of nodes can be covered with cycles in which all edges are labeled 0. We will use τ_1(E2-LIN-2) to denote this restricted version of E2-LIN-2. The last two properties of $\tau_1(E2 - LIN - 2)$ are important in the subsequent reductions that lead to MIN SBR problem.

We alter the reduction τ_1 in two ways. The first modification results in graphs that have all edges labeled with 1, i.e. it reduces E2-LIN-2 to 3-MAX CUT and allows to complete the proof of *(i)*. The second modification reduces E3-LIN-2 to a very special version of 3-OCC-E3-LIN-2, which we call HYBRID, because a large majority of equations have only two variables. This reduction instantaneously leads to *(iii)*.

To show *(ii)*, we use an obvious reduction from τ_1(E2-LIN-2): an instance of E2-LIN-2 can be viewed as a set of equivalence statements, and we can replace each equivalence with a pair of implications. On the other hand, we obtain *(v)* and *(iv)* using reductions from HYBRID.

Although HYBRID problem appears to be very "efficient", we cannot use it in the chain that leads to MIN-SBR. Instead, we use another reduction, with instance translation τ_2, that leads from τ_1(E2-LIN-2) to 4-MIS. This translation replaces each node/variable with a small gadget. The resulting instances of 4-MIS can be transformed into the next problem that we consider, which we call *breakpoint graph decomposition*, BGD. This problem is related to *maximum alternating cycle decomposition*, (e.g. see Caprara, [C97]) but has a different objective function (as with another pair of related problems, Node Cover and MIS, the choice of the objective function affects approximability). An instance of BGD is a so-called breakpoint graph, i.e. an undirected graph $G = \langle V, E, l \rangle$ where l is a 0/1 edge labelling function, which satisfies the following two properties:

(i) for $b \in \{0, 1\}$, each connected component of $\langle V, l^{-1}(b) \rangle$ is a simple path;

(ii) for each $v \in V$, the degrees of v in $\langle V, l^{-1}(0) \rangle$ and in $\langle V, l^{-1}(1) \rangle$ are the same.

An alternating cycle C is a subset of E such that $\langle V, C, l|C \rangle$ has the property (ii). A decomposition of G is a partition $\mathcal{C}$ of E into alternating cycles. The objective of BGD is to minimize $cost(\mathcal{C}) = \frac{1}{2}|E| - |\mathcal{C}|$.

By changing the node-replacing gadget of τ_2 and enforcing property (i) by "brute force", we obtain reduction τ_3 that maps τ_1(E2-LIN-2) into BGD. The last reduction, π, converts a breakpoint graph G into a permutation $\pi(G)$, an instance of sorting by reversals, MIN-SBR. We use a standard reduction, i.e. the correspondence between permutations and breakpoints graphs used in the approximation algorithms for MIN-SBR (this approach was initiated by Bafna and Pevzner, [BP96]). In general, this correspondence is not approximation preserving because of so-called *hurdles* (see [BP96, HP95]). However, the permutations in $\pi(\tau_3(\tau_1(\text{E2-LIN-2})))$ do not have hurdles, and consequently for these restricted version of BGP, π is an approximation preserving reducibility with ratio 1.

3 First Reduction

To simplify the first reduction, we will describe how to compute the instance translation using a randomized poly-time algorithm. In this reduction, every node (variable) is replaced with a *wheel*, a random graph that is defined below (some parts of this definition will not be used to describe the reduction, but will be used later, in the proof of correctness). The parameter κ used here is a small constant; in this version of this paper we sketch the proof that $\kappa = 9$ is sufficiently large, in the full version we will show that $\kappa = 6$ is also sufficient.

Definition 2. An *r-wheel* is a graph with $2(\kappa+1)r$ nodes $W = \mathit{Contacts} \cup \mathit{Checkers}$, that contains $2r$ *contacts* and $2\kappa r$ *checkers*, and two sets of edges, C and M. C is a Hamiltonian cycle in which with consecutive contacts are separated by chains of κ checkers, while M is a random perfect matching for the set of checkers (see Fig. 1 for an example).
For a set of nodes $A \subset W$ let a_A be the number of contacts in A, b_A the number of contiguous fragments of of A in the cycle C (i.e. $b_A = |Cut(A) \cap C|/2$) and $c_A = |Cut(A) \cap M|$.
We say that A is *bad* iff $r \geq a_A > 2b_A + c_A$. A set B is *wrong* iff for some bad set A we have $B = A \cap \mathit{Checkers}$. A set $B \subset \mathit{Checkers}$ is *isolated* iff no edges in M connect B with $\mathit{Checkers} - B$.

Consider an instance of E2-LIN-2 with n nodes (variables) and m edges (equations). Let $k = \lceil n/2 \rceil$. A node v of degree d will be replaced with a kd-wheel W_v. All wheel edges are labelled 0 to indicate our preference for such a solution S that either $W_v \subset S$ or $W_v \cap S = \emptyset$. An edge $\{v, u\}$ with label l is replaced with $2k$ edges, each of them has label l and joins a contact of W_v with a contact of W_u. In the entire construction each contact is used exactly once, so the resulting graph is 3-regular.

We need to elaborate this construction a bit to assure a large girth of the resulting graph. First, we will assure that no short cycle is contained inside

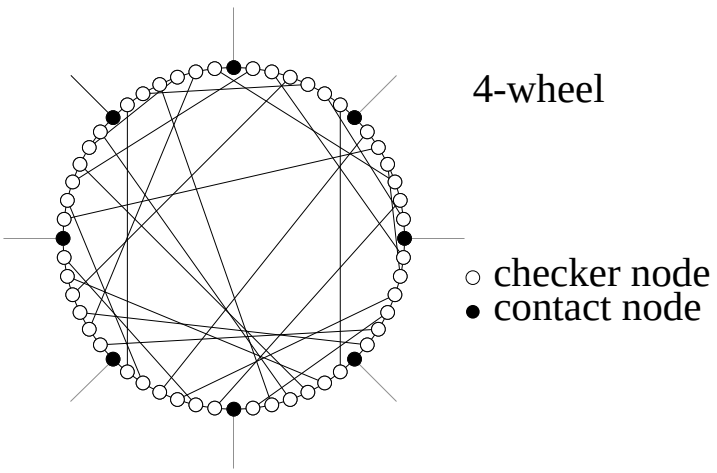

Figure 1: A small example of a gadget used by τ_1.

a wheel. We can use these properties of an r-wheel W: each cycle diferent of length lower than $2\kappa r$ must contain at least one edge of the matching M and the expected number of nodes contained in cycles of length $0.2\log_2(\kappa r)$ or less is below $(\kappa r)^{-0.8}$ fraction). Thus we can destroy cycles of length below $0.2\log_2 n$ by deleting matching edges incident to every node on such a cycle and neglect the resulting changes in *Score*.

Later, we must prevent creation of short cycles when we introduce edges between the wheels; this can be done using a construction described by Bollobás [B78]. While Bollobás described how to build a graph of large girth from scratch, his construction can assure the following: given a graph of degree 3 with girth at least $0.5\log_2 n$ and two n-element disjoint sets of nodes of degree 2, each of size n, say A and B, one can increase the set of edges by a perfect bipartide matching of A and B without increasing the girth above $0.5\log_2 n$. Note that we are indeed replacing an edge of the original graph with a perfect matching with at least n edges, which allows us to use the construction of Bollobás.

The solution translation is simple. Suppose that we have a solution S for a translated instance. First we normalize S as follows: if the majority of contacts in a wheel W belong to S, we change S into $S \cup W$, otherwise we change S into $S - W$. A normalized solution S can be converted into a solution S' of the original problem in an obvious manner: a node belongs to S' iff its wheel is contained in S. Assuming that G has m edges/equations, we have $Score(S) = 2k((3\kappa+2) + Score(S'))$. Håstad [H97] proved that for E2-LIN-2 instances with $16n$ equations it is NP-hard to distinguish those that have *Score* above $(12-\epsilon)n$ and those that have *Score* below $(11+\epsilon)n$, where the positive constant ϵ can be arbitrarily small. By showing that our reduction is correct for $\kappa = 6$ we will prove

Theorem 3. *For any $\epsilon > 0$, it is NP-hard to decide whether an instance of τ_1(E2-LIN-2) $\in$ 3-OCC-E2-LIN-2 with $336n$ edges (equations) has Score above $(332-\epsilon)n$ or below $(331+\epsilon)n$.*

The latter claim uses the assumption that $Score(S)$ is not decreased by the normalization. Because the reduction uses a random matching, it actually does not have to be the case, i.e. the normalization may fail. Obviously, if the normalization fails, than one of its step, say dealing with wheel W, fails. Let us inspect closer what such a failure means. For some d, W is a kd-wheel, so it contains $2kd$ contacts. Let A be the subset of W consisting of nodes that change membership in S during the normalization step. It is easy to see that $Score(S,e)$ changes iff $e \in Cut(A)$. According to our definition, the size of $Cut(A)$ is $a_A + 2b_A + c_A$. The edges counted by $2b_A$ and c_A are inside W, so their score is changed to 1 (from 0); the edges counted by a_A are connecting the contacts in A with contacts of other wheels, pessimistically we may assume that their score changes to 0. As a result, $Score(S)$ decreases by at most $a_A - 2b_A - c_A$; the normalization step fails only if $a_A > 2b_A + c_A$, i.e. only if A is a bad subset of the wheel W. To show that our reduction preserves the approximation with a high probablility we need to show that the probablility that a wheel contains a bad subset is very low. Note that when we try to find a bad set A in a wheel, it is very easy to obtain any possible combination of the values of a_A and b_A. However, the number c_A is established by a random matching, so we need to use the fact that with a very high probability $Cut(A) \cap M$ contains many edges. We start with the following lemma.

Lemma 4. *Assume that Q is a clique, $P \subset Q$, $2q = |Q|$ and $2p = |P|$. Choose, uniformly at random, a perfect matching M for Q. Then the probability that $Cut(P) \cap M$ is empty equals*

$$\binom{q}{p}\binom{2q}{2p}^{-1} \leq 2\left(\frac{p}{2q}\right)^p \quad .$$

Proof. Let μ_r be the number of perfect matchings in a complete graph with $2r$ nodes. By an easy induction, $\mu_r = \prod_{i=1}^{r}(2i-1) = (2r)!/(2^r r!)$. The probability of our event is

$$\frac{\mu_p \mu_{q-p}}{\mu_q} = \frac{(2p)!}{2^p p!}\frac{(2(q-p))!}{2^{q-p}(q-p)!}\frac{2^q q!}{(2q)!} = \frac{(2p)!(2p-2q)!}{(2q)!}\frac{q!}{p!(q-p)!} \quad .$$

The second part of the claim follows from Stirling formula.

Consider now a bad set A. Suppose that a node $u \in A$ has two neighbors in $W - A$. It is easy to see that after removing u from A the expression $a_A - 2b_A - c_A$ increases, so A remains bad. Similarly, if $u \notin A$ has two neighbors in A we may insert u and A again remains bad. Therefore W contains a bad set only if it contains such a bad set A that neither A nor $W - A$ contains fragments of size 1.

Consider now set $B \subset Checkers$. Let B_i be the set of contacts that have exactly i neighbors in B. According to our last remark, B is wrong iff for

some $B' \subset B_1$ the set $A = B \cup B_2 \cup B'$ is bad. Clearly, whatever the choice of B', we have $a_A = |B_2| + |B'|$, $b_A = b_{B \cup B_2}$ and $c_A = c_B$. Thus if $|B_2| > r$ then B cannot be wrong, else if $|B_2| + |B_1| > r$ we can assume that $a_A = r$, and in the remaing case we can assume that $a_A = |B_2| + |B_1|$. Later we will use notation a_B, b_B and c_B to denote these reconstructed values of a_A, b_A and c_A.

The probability that W contains a bad subset can be estimated with a sum, over every $B \subset \mathit{Checkers}$, of the probability that B is wrong. Instead of computing this probability, we will estimate it, using three parameters of this set.

The first parameter of B is α, defined by the equality $a_B = \alpha r$. Because B is wrong only if $a_B \leq r$, we may assume that $\alpha \in (0, 1]$. The second parameter is β, defined by $b_B = \beta\alpha r$. Because B can be wrong only if $a_B > 2b_B$, β is a fraction in the range $(0, \frac{1}{2})$.

Before we define the third parameter, we will use the first two to count then number of ways in which B can be generated. The sets B and $\mathit{Checkers} - B$ together contain $2\beta\alpha r$ fragments which can be described by indicating, for each of them, the fist element (say, if we move in clockwise direction). This description leaves ambigous which is set B and which is $W - B$, this can be decided using the property $a_B \leq r$. Thus we can generate B in

$$\binom{2\kappa r}{2\beta\alpha r} \leq (e\kappa)^{2\beta\alpha r} \left(\frac{1}{\beta\alpha}\right)^{2\beta\alpha r} = \quad \xi$$

many ways.

After we generated a set B, we need to estimate the probability that it is wrong. To do so, we need to make an assumption concerning its size. It is easy to see that a fragment of B that contributes, say, a, to a_B, must contain $a - 1$ complete chains of checkers, each of length κ, so it contributes at least $(a - 1)\kappa$ to the size of B. Additionally, this fragment may contain two "fringe" chains, each of length between 0 and $\kappa - 1$, so it contributes less than $(a + 1)\kappa$ to the size. After adding such inequalities together over $\beta\alpha r$ fragments we see that

$$\alpha\kappa r - \beta\alpha\kappa r \leq |B| < \alpha\kappa r + \beta\alpha\kappa r \quad ,$$

hence for some $\gamma \in [-1, 1]$ we have $|B| = (1 + \gamma\beta)\alpha\kappa r$. Note that B will become isolated if we remove the endpoints of the matching edges that connect B with $W - B$; if B is wrong, then the number of such endpoints is at most $c_B < (1 - 2\beta)\alpha r$. We can estimate the probability that B is wrong by multiplying the number of ways in which we can remove $(1-2\beta)\alpha r$ nodes (call it ρ) with the probability that the result is isolated. The former can be estimated as

$$\binom{(1 + \gamma\beta)\alpha\kappa r}{(1 - 2\beta)\alpha r} \leq (e\kappa)^{(1-2\beta)\alpha r} \left(\frac{1 + \gamma\beta}{1 - 2\beta}\right)^{(1-2\beta)\alpha r} = \quad \zeta \quad .$$

To express the latter, we define $\delta(\beta,\gamma)$ so that the size of each of our candidates for an isolated set is $2\delta(\beta,\gamma)\alpha r$, one can see that $\delta(\beta,\gamma) = [(1+\gamma\beta)\kappa-(1-2\beta)]/2$ and the probability that the candidate set is indeed isolated is below

$$\left(\frac{\delta(\beta,\gamma)\alpha}{2\kappa r}\right)^{\delta(\beta,\gamma)\alpha r} = \left(\frac{\delta(\beta,\gamma)\alpha}{2\kappa}\right)^{\delta(\beta,\gamma)\alpha r} = \quad \psi \quad .$$

We need to show $\xi\zeta\psi << 1$; it suffices to show that $(\xi\zeta\psi)^{1/(\alpha r)} < 1$. We easily can compute that

$$(\xi\zeta\psi)^{1/(\alpha r)} = e\kappa\left(\frac{1}{\beta\alpha}\right)^{2\beta}\left(\frac{1+\gamma\beta}{1-2\beta}\right)^{1-2\beta}\left(\frac{\delta(\beta,\gamma)\alpha}{2\kappa}\right)^{\delta(\beta,\gamma)} \quad .$$

One can quickly check that the above formula is an increasing function of α (because $2\beta \leq 1 < (\kappa-1)/2 < \delta(\beta,\gamma)$). Since we want to estimate it from above, we can put $\alpha = 1$. Now it remains to check that the simplified function is always smaller than 1 for $\beta \in (0, \frac{1}{2}$ and $\gamma \in [-1,1]$. Using the fact that the partial derivative is bouded, one can accomplish it by evaluating this function in a limited number of points. For $\kappa = 9$ we checked that 0.72 is an upper bound. With a more complicating argument, and more accurate estimates than Lemma 4, one can also show that $\kappa = 6$ is sufficient as well.
Remark 1. One can modify reduction τ_1 as follows. We replicate the set of equations even number of times, as before, so the number of occurences of each variable is sufficiently high. On each r-wheel the nodes are labeled with a and b, labels alternating. When we select the random matching between checkers, we choose only from perfect matchings in a full bipartite graphs formed by a-checkers and b-checkers (rather than a random perfect matching from the full graph). One can easily show that this restriction makes almost no difference in the probability calculations. Moreover, when we connect the contacts of two wheels, we do it in two ways. If the edge between the respective original variables is labeled with 0, we connect a-contacts with b-contacts, and vice versa. If this edge is labelled with 1, we connect a-contacts with a-contacts and b-contacts with b-contacts. This allows us to convert all labels in the new graph to 1, and as a result, we obtain a graph which is simultaneously an instance of E2-LIN-2 and MAX CUT (and is 3-regular). Let τ_1' be the new reduction. We obtain the following:

Theorem 5. *For any $\epsilon > 0$, it is NP-hard to decide whether an instance of τ_1'(E2-LIN-2)$\in$3-MAX CUT with $336n$ edges has Score above $(332-\epsilon)n$ or below $(331+\epsilon)n$.*

Remark 2. We can translate MAX CUT into MAX 2SAT by replacing each edge with two clauses, i.e. and edge $\{x,y\}$ is replaced with $x \vee y$, $\bar{x} \vee \bar{y}$. This reduction allows to prove Theorem 1*(ii)*.

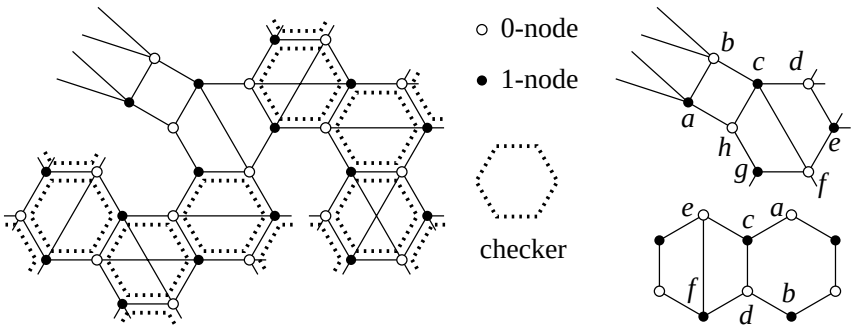

Figure 2: Consistency wheel for 4-MIS problem. The gadget used to replace a contact node is shown in the upper right corner. The lower right corner shows a way to avoid a dirty hexagon.

Remark 3. We will also use another modification. We can start from an instance of E3-LIN-2 with $2n$ equations. (Recall that Håstad has shown that it is NP-hard to distinguish instaces where $(2-\epsilon)n$ equations can be satisfied from those where we can satisfy at most $(1+\epsilon)n$.) We modify it to an instance with in which each variable occurs in at least n equations, again, by replicating the equations. Next, each variable is replaced by a r-wheel, where r is the (increased) number of occurences. The original equations are left same as before, but occurences of a variable are replaced with occurences of its contacts. Now we have a new system where each variable occurs exactly three times, and consisting of $2kn$ equations with 3 variables (replicated original equations) and $(1.5\kappa+1)6kn$ equations with 2 variables (inside the wheels). We take $\kappa = 6$, so we have $60kn$ equations inside the wheels. It will be convenient to view the resulting structure as a hypergraph that has $60kn$ normal edges and $2kn$ hyperedges (of size 3), $6kn$ contact nodes and $36kn$ checker nodes.

We can modify the last reduction in a similar manner as in Remark 1. In each chain of 6 checkers (separating two contacts) we label 3 of them with a and 3 with b; then we choose a random bipartite matching between a-checkers and b-checkers. The set of resulting instances of E3-LIN-2 will be later called HYBRID (this name refers to the fact that we have a mixture of equations with 2 and 3 variables). Observe that the reduction from E3-LIN-2 to HYBRID allows to prove Theorem 1*(iii)*.

4 From HYBRID to 4-MIS and 3-MIS

Given an instance S of HYBRID, we will form graph G of degree 4, an instance of 4-MIS. Each variable/node x of S will be replace with a gadget A_x which is an induced subgraph of G. Every gadget contains a *hexagon*, i.e. a cycle of length 6 in which nodes with labels 0 and 1 alternate. Hexagons

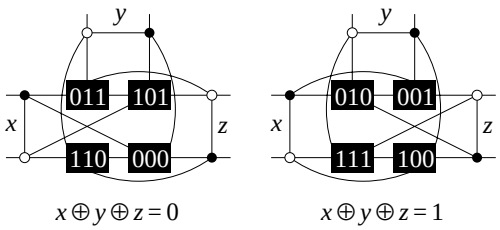

Figure 3: Equation gadgets for 4-MIS.

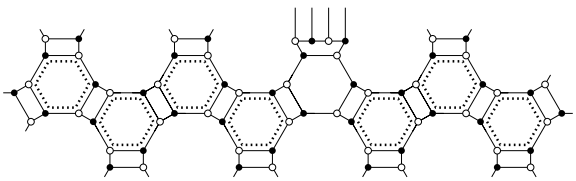

Figure 4: Consistency wheel for 3-MIS.

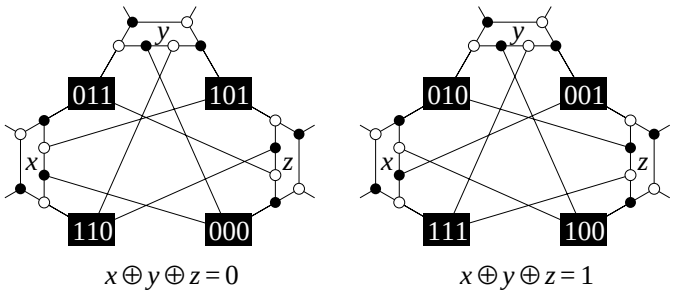

Figure 5: Equation gadgets for 3-MIS.

will have two types: a-hexagons, with 2 chords, and b-hexagons, with 1 chord.

If x and y are connected by an edge (equation with two variables), the hexagons of A_x and A_y will share a pair of adjacent edges; this edge of G corresponds to the equation/edge $x = y$. A checker gadget is simply a hexagon: 3 edges edges of equations connected by three other edges, and one or two diagonals. A contact gadget consists of a hexagon fused with a square; 3 such gadgets are connected by an equation gadget that contains 4 nodes that do not belong to gadgets of nodes/variables. Fig. 2 and 3 show these gadgets in detail.

Given an independent set (a solution) I in graph G we form a solution of S as follows. If $A_x \cap I$ consists of one type of nodes only (i.e. only 0-nodes or only 1-nodes), we assign to x the value equal to this type. In this case, we say that A_x is *pure*. If A_x is *dirty*, we must purify it without decreasing the size of I.

Suppose first that a hexagon H is dirty (a checker gadget or a part of a contact gadget). It is easy to see that H can be dirty in one way only: $H \cap I$ is a pair of nodes that forms a "missing diagonal" of H. In the lower right part of Fig. 2, we assume that $\{a, b\}$ is this pair. The construction of G assures that in this case there exists a quadrilateral (c, d, e, f) as in this figure, either because $\{e, f\}$ is a diagonal of an adgacent hexagon, or because hexagon H is a part of a contact gadget and this quadrilateral is the square included in this gadget. One must observe that in the cases we consider nodes adjacent to c and d are either adjacent to a or b (and consequently they cannot be in I) or belong to $\{e, f\}$. If $e \notin I$, we can purify H removing a from I and inserting c, if $f \notin I$, we do it by removing b and inserting d. One can see that one of these two cases must hold. Moreover, if the edge $\{c, d\}$ is shared with another gadget, we can always choose the replacement in such a way that we do not make the other gadget dirty when we purify H.

Once we made all hexagons pure, we can make every contact gadget pure as well. Suppose that the gadget from the upper right corner of Fig. 2 is dirty. There are two cases: if $a \in I$, then the hexagon (c, d, e, f, g, h) is 0-pure and we can replace a with h; the case when $b \in I$ is symmetric.

Now we can modify I so that each edge corresponding to an equation with two variables contains a node of I iff the respective equation is true. If such an edge contains a b-node ($b\{0, 1\}$), than both gadgets containing this edge must be b-pure; if both of them are b-pure, we can insert the b-node of this edge to I.

If we partition G into gadgets coresponding to equations, that a gadget A of an equation with three variables consists of 16 nodes: a square contained in a gadget of each participating (contact) variable and four special nodes corresponding to four legal combinations of variable values. Our goal is to assure that if the this equation is true, $A \cap I$ contains 7 nodes and 6 if the

equation is false. Clearly, we can place two nodes of I in each square, so $A \cap I$ always has at least 6 nodes. We consider three cases, according to the number of special nodes in $A \cap I$. If this number is 0 and the equation is false, we are done. If it is 0 and the equation is true, then we can insert the special node corresponding to the combination of the values of the three variables. If this number is 1 and the equation is true, again, we are done. If the equation is false, than one of the special node p contained in I wrongly describes one of the variable values, and so it is connected to a node q in the respective contact gadget that has the type equal to the value of this variable; clearly we can replace p with q. Now suppose that this number is 2. Because the equation gadget is very symmetric, it suffices to consider one case, e.g. that the two special nodes in I are 000 and 011. In this case the squares of y and z contain only one node of I each, thus we can replace 000 and 011 with nodes from these two squares.

To finish our reasoning, it remains to perform the accounting. We start with an HYBRID instance with $60kn$ equations with two variables and $2kn$ equations with three variables, and the difficult question whether we can satisfy at least $(62-\epsilon)kn$ equations, or at most $(61+\epsilon)kn$. Each of $2kn$ gadgets corresponding to equations with three variables contributes 6 nodes to an independent set, even if they are false. Morover, each gadget contributes a node if the respective equation is true. As a result, the new difficult question is whether the maximum independent set contains at least $(12+62-\epsilon)kn$ nodes, or at most $(12+61+\epsilon)kn$.

Theorem 6. *For any $\epsilon > 0$, it is NP-hard to decide whether an instance of 4-MIS with $152n$ nodes has the maximum size of an independent set above $(74-\epsilon)n$ or below $(73+\epsilon)n$.*

Suppose now that we can reduce the size of the gadget corresponding to an equation with three variables so it consists of 10 nodes rather than 16, and it contributes 4 nodes to an independent set if the equation is false, and 5 if it is true. In this case the above accounting would show that for graphs with $140n$ nodes it is difficult to distinguish between those that have a maximum independent set with at least $(68-\epsilon)n$ nodes and thoses that have at most $(67+\epsilon)n$ nodes. We can achieve this by constructing the gadget for replacing contact nodes that has two nodes less than the one in Fig. 2. However, some nodes in this gadget have degree 5 (see Fig. 6 and hence the improved result, mentioned in Table 1, applies to 5-MIS (and, by extension, 5-Node Cover). Because in these instances only $12n$ nodes out of $140n$ have degree 5, we believe that this result should be easy to improve.

We can describe a similar reduction from HYBRID to 3-MIS. Given a HYBRID system of equations S, we form a graph G of degree 3. Again, each variable x of HYBRID is replaced with a gadget A_x; the gadget of a checker variable is a hexagon, and a gadget of a contact variable is a hexagon

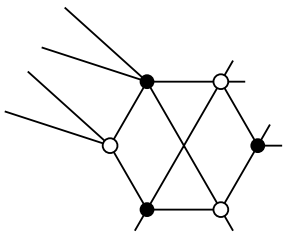

Figure 6: Contact gadget for 5-MIS.

augmented with a *trapesoid*, a cycle of 6 nodes that shares one edge with the hexagon. The hexagons used here have no chords. If two variables/nodes x, y are connected by an equation/edge, $x = y$, we connect their hexagons with a pair of edges to form a rectangle in which the edges of the hexagons and the new edges alternate. The rectangle thus formed is a gadget of this equation. If three variables are connected by an equation/hyperedge, say, $x \oplus y \oplus z = 0$, the trapesoids of A_x, A_y and A_z are connected to four special nodes of the gadget of this equation. As a result, the gadget of this equation consists of 3 trapesoid and 4 special nodes, for the total of 22 nodes. The details are shown in Fig 4 and Fig. 5.

Given a solution of the new problem, and independent set I of G, we translate it into a solution of S in the same manner as before. Again, if some variable gadget are dirty, we need to purity them, so that this translation will be well-defined. The beginning of the purification is same as before: we purify dirty hexagon using the method illustrated in the lower right corner of Fig. 2. As a result, all checker gadgets become pure. We can also insists that if a checker variable x is connected to a contact y, I contains a node in the intersection of the gadget of this equation (a rectangle) with A_x.

Now we consider a contact gadget A_x. Of H_x is the hexagon of A_x, we will say that $A_x - H_x$ is the *front piece* of A_x and we use F_x to denote it. Before we proceed, we make the following observation:

Observation. Assume that $|F_x \cap I| = i$ and that A_u and A_v are the adjacent checker gadgets. We can modify I so that A_x becomes pure, $A_u \cap I$ and $A_v \cap I$ do not change, and the size of A_x increases by $2 - i$.

Now we can return to the gadget of $x \oplus y \oplus z = 0$. Our goal is that after all stages of the purifications, each variable gadges is pure, each trapesoid contains 3 nodes of I and if the equation is satisfied (we can decide that once the the gadget variables are pure and the value translation is defined) its gadget contains the special node described by the triple of values of x, y and z; otherwise no special nodes belong to I. As a result, a satisfied equation corresponds to 10 nodes in I and an unsatisfied equation corresoponds to 10. Morever, a satisfied equation with two variables corresponds to 2 nodes in I. This will lead to the following accounting: the question whether we can satisfy at least $(62 - \epsilon)kn$ equations or at most $(61 + \epsilon)kn$, where $60kn$

equations have two variable translates into the question whether the maximum independenet set has at least $(2 \times 60 + 20 - \epsilon)kn$ elements or at most $(2 \times 60 + 19 + \epsilon)kn$. This will lead to the following theorem:

The first case that we consider during the purification of an equation gadget is when I contains all 4 of its special nodes. In this case, $|F_v \cap I| = 0$ for $v = x, y, z$; according to the Observation, we can remove 4 special nodes form I, make all the participating contact gadgets pure and increase the size of I by at least 6-4. The second case is when I contains 3 of the special nodes; then $|F_v \cap I \leq 1$ for $v = x, y, z$; now throwing away the special nodes and purifying the contact gadgets increases the size of I be at least $3 - 3$. Lastly, when I contains two special nodes, we can remove one of them and, by Observation 1, purify one of the variable gadgets and restore the size of I (one can inspect all 6 cases to prove it). Thus at the end we need to consider only cases when I contains at most one special node (from a given equation gadget).

Theorem 7. *For any $\epsilon > 0$, it is NP-hard to decide whether an instance of 4-MIS with $284n$ nodes has the maximum size of an independent set above $(140 - \epsilon)n$ or below $(139 + \epsilon)n$.*

5 From E2-LIN-2 to 4-MIS

An instance of 4-MIS can be modified to became an instance of BGD in a simple manner: each node can be replace with an alternating cycle of length 4; adjacent nodes will be replaced with a pair such cycles that have an edge (or two) in common. If we are "lucky", after the replacement we indeed obtain a breakpoint graph. Unfortunately, it is not possible to apply such transformation consistently to a graph from Fig. 3. We did not find other gadgets that can replace an equation with three variables and can later be replaced with a fragment of a breakpoint graph. Therefore we will be using a translation from τ_1(E2-LIN-2), shown in Fig 7.

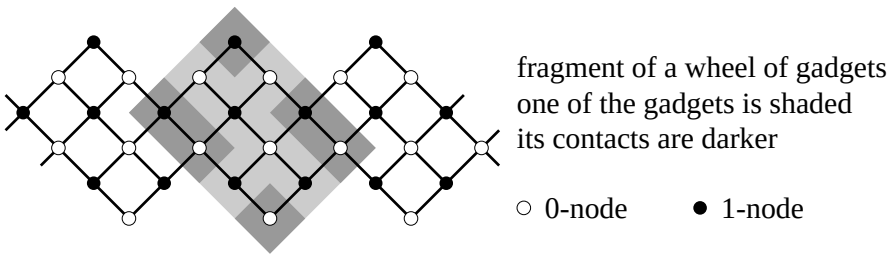

Figure 7: A part of 4-MIS instance obtained from τ_1(E2-LIN-2).

It is easy to see that the size of the resulting 4-MIS graph is $9n$, and that the correspondence between the size of the pure solution and the score in the

original τ_1(E2-LIN-2) instance is $i = 3n + s$. The "purifying" normalization has to proceed somewhat different, however. We do it in two stages. The result of the first stage is that gadgets are either pure, or contain no nodes of I in their contacts.

If an impure gadget contains only 4 nodes of I (or less), we replace these nodes with the (unique) independent set of size 4 with no contact nodes (i.e. contained in the light gray area of Fig. 2b). A gadget that contains 6 nodes of the independent set is already pure. If an impure gadget contains 5 nodes of I, then it must contain one of the two "central" points (note that the non-central nodes form a cycle of length 10). Suppose that this central node has label 0. Then I cannot contain neither of the 4 adjacent 1-nodes, and the remaing 7 nodes form two isolated 0-nodes and a chain of the form 0-1-0-1-0, where the final 0-1 is a contact. If the chain contains 3 nodes of I, the gadget is pure. Otherwise we can set the intersection of I with this chain to contain two 0-nodes that do not belong to the contact; afterward the gadget becomes pure.

At this point, we have "pure" gadgets, with 0 or 1 values, and at least 5 nodes of I, and "undecided" gadgets that contain only 4 nodes of I. If an undecided gadget is adjacent to two gadgets that are either 0-pure or undecided, then we can incease I by increasing the number of nodes of I to 5, all of them 0. There is also symmetric case for 1, and one of the two cases must hold.

6 Reduction to BGD

The idea of reducing MIS problem to BGD is very simple and natural. Observe that the set E of all edges forms an alternating cycle (AC for short), a disjoint union of ACs is an AC, and a difference of two ACs, one contained in another is also an AC. Thus any disjoint collection of ACs can be extended to a decomposition of AC. Consequently, the goal of BGD is to find a collection of disjoint ACs as close in size to the maximum as possible.

Second observation is that the consequences of *not finding* an AC diminish with the size of AC. Suppose that the input has n breakpoints (edges of one color), and that we neglect to find any AC's with more than k breakpoints. The increase in the cost of the solution is smaller than n/k, while the cost is at least $n/2$. Thus if $k = \Omega(\log n)$, such oversight does not affect the approximation ratio.

The strategy suggested by these observation is to create instances of BGP in which alternating cycles that either have 2 breakpoints, or $\Omega(\log n)$. Then the task of approximating is equivalent to the one of maximizing the size of independent set in the graph $\mathcal{G}$ of all ACs of 4; we draw an edge between two ACs if they share an edge.

More to the point, we need to find a difficult family of graphs of degree

4 which can be converted into breakpoint graphs by replacing each node with an alternating cycle of size 4. To this end, we can use the results of the second reduction described in the previous section. Fig. 3 shows the result of this replacement applied to the long cycles of gadgets. The union of ACs used in the replacements is also a disjoint union of 5 ACs (in Fig. 3 these ACs are horizontal zigzags). To apply the resoning of the previous sections, we need to establish that no cycles of length larger than 4 have to be considered. In the short version we only sketch this argument.

The cycles in question fall into three categories. The first kind of cycles are included in an adjacent pair of gadgets, identified on their diagonally placed corners. By an easy case analysis one can show that we can replace such cycles with a larger collection of cycles of size 4. The second kind traverses a collection of gadgets that is cycle-free (if each gadget is considered to be a node). Such a cycle has a defined interior; the union of the cycle with its interior can be easily decomposed into 4-cycles. The third and last kind traverses a cycle of gadgets. Then it must be at least as long as such a cycle, i.e. $\Omega(\log n)$.

At this point the translation is still not correct, as the resulting graphs MUST violated property (i) of BPG: edges of one kind form a collection of cycles: in Fig. 3 such edges form diagonal lines consisting of 5 edges each; such a line crosses to another strip of gadgets and then proceeds without end. However, these cycles induce cycles of gadgets, hence have length $\Omega(\log n)$, moreover, they are disjoint. Therefore we can remove all these cycles by breaking $O(n/\log n)$ contacts between the strips.

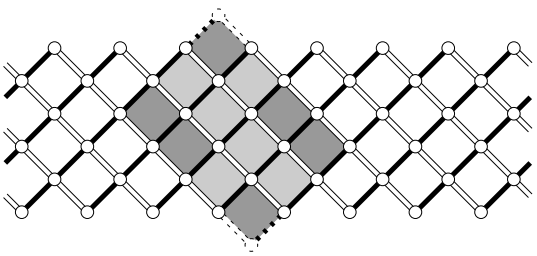

Figure 8: Gadget for breakpoint graphs.

Given and instance G of τ_1(E2-LIN-2) with $2n$ nodes and $3n$ edges, this construction creates BGD instance G' with $20n$ breakpoints (edges of one color), and the correspondence between the cost c of a cycle decomposition in G' and s, *Score* of the corresponding solution of G is $c = 20n - 3n - s$. Together with Theorem 3 this implies

Theorem 8. *For any $\epsilon > 0$, it is NP-hard to decide whether an instance of BGD with $2240n$ breakpoints has the minimum cost of an alternating cycle decomposition below $(1236 + \epsilon)n$ or above $(1237 - \epsilon)n$.*

7 Reduction of 3-MAX CUT to 3-OCC-MAX 2SAT

In order to translate an instance $G =< V, E >$ of 3-MAX CUT into a set of disjunctive two clauses, we create a separate set of 4 propositional variables for each edge $\{u, v\}$ and 4 clauses, $\sim u_0^e \vee u_1^e$, $\sim v_0^e \vee v_1^e$, $u_0^e \vee v_0^e$ and $\sim u_1^e \vee \sim v_1^e$. Morever, for each node incident to edges e, f and g we add clauses $\sim u_1^e \vee u_0^f$, $\sim u_1^f \vee u_0^g$ and $\sim u_1^g \vee u_0^e$. Thus, if $|V| = 2n$ and $|E| = 3n$ we have $12n$ propositional variables and $18n$ clauses.

To describe a solution translation, consider a valuation of propositional variables, say I. Before we translate I into a partition of V, we will normalize I without decreasing the number of satisfied clauses. We do it in three stages.

(i) We eliminate cases when for some $e = \{u, v\}$ we have $I(u_0^e) = 0$ and $I(u_1^e) = 1$. In every such situation we change $I(u_1^e)$ to 0. Afterwards all 3 clauses where u_1^e occurs are true: two of them contain $\sim u_1^e$, and the other one contains $\sim u_0^e$. Clearly, the number of true clauses could not decrease.

(ii) We eliminate cases when for some $e = \{u, v\}$ we have $I(u_0^e) \neq I(u_1^e) = 1$. Because we performed 1), this means $I(u_0^e) = 1$ and $I(u_1^e) = 0$. Consider $I(v_0^e)$, if it is 1, then we change $I(u_0^e)$ to 0. It results in $\sim u_0^e \vee u_1^e$ *becoming* true, $u_0^e \vee v_0^e$ and *remaining* true, so the number of true clauses cannot decrease. On the other hand, if $I(v_0^e) = 0$, then because of 1) we have $I(v_1^e) = 0$. In this case we change $I(u_1^e)$ to 1.

(iii) We eliminate cases when for some u, e, f, i, j we have $I(u_i^e) \neq I(u_j^f)$. In such a situation, pairs of the form u_0^e, u_1^e have equal values of I, thus among 3 such pairs there must be exactly one minority pair, say the one that cooresponds to edge e. We convert this pair to the majority value; as a result we gain one clause in the ring of implications of u and loose at most one clause in the gadget of e.

After the normalization, every 6-tuple of propositional variables that corresponds to a node u of G has the same valuation, which we may denote $I(u)$. We define C as the set of those nodes that have $I(u) = 0$. It is easy to see that $CUT(C) = k$ iff for our set of $18n$ clauses, I satisfies $15n - k$ (6 clauses for every of $2n$ nodes, 1 clause for each of $3n$ edges and one extra clause for every edge in $CUT(C)$).

By applying this reduction together with Theorem 5 we can show that for any $\epsilon > 0$ it is NP hard to decide whether an instance of 3-OCC-MAX 2SAT with $2016n$ clauses has a truth assignment that satisfies at least $(2012 - \epsilon)n$ clauses, or it can be at most $(2011 + \epsilon)n$.

8 Reduction to MIN-SBR

Our reduction from BGD to MIN-SBR is straightforward, in particular we can use the procedure GET-PERMUTATION of Caprara [C97, p.77] to obtain permutation $\pi(G)$ from a given breakpoint graph G. It is easy to show that if G is the result of reduction $\tau_4 \circ \tau_1$ applied to E2-LIN-2, then π has $o(n)$ *hurdles*. The basic reason is that all ACs of length 4 that may belong to a normalized solution (decomposition into ACs) for a single connected component in the *interleaving graph* (cf. [BP96, HP95]), because the number of longer cycles in a cover is $O(n/\log n)$, this implies that the total number of connected components of the interleaving graph is $O(n/\log n)$. Because hurdles are defined as connected components with a special property, we can conclude that there are $O(n/\log n) = o(n)$. As a result, the number of reversals needed to sort π is exactly equal (modulo lower order terms) to the minimum cost of a decomposition of G into alternating cycles. Therefore Theorem 8 apllies also to MIN-SBR.

9 Further Research and Open Problems

It would very interesting to improve still huge gaps between approximation upper and lower bounds for bounded approximation problems of Table 1. The lower bound of 1.0008 for MIN-SBR is the first inapproximability result for this problem. The especially huge gap between 1.5 and 1.0008 for the MIN-SBR problem reflects a great challenge for future improvements. A similar gap exists also for 3-OCC-MAX 2SAT.

Acknowledgements

We thank Johan Håstad and Luca Trevisan for stimulating remarks on the preliminary version of this paper.

References

[AFWZ95] N. Alon, U. Feige, A. Wigderson and D. Zuckerman, *Derandomized Graph Products*, Computational Complexity **5** (1995), pp. 60–75.

[A94] S. Arora, *Probabilistic Checking of Proofs and Hardness of Approximation Problems*, Ph. D. Thesis, UC Berkeley, 1994;
available as TR94-476 at `ftp://ftp.cs.princeton.edu`

[ALMSS92] S. Arora, C. Lund, R. Motwani, M. Sudan and M. Szegedy, *Proof Verification and Hardness of Approximation Problems*, Proc. 33rd IEEE FOCS (1992), pp. 14–23.

[BP96] V. Bafna and P. Pevzner, *Genome rearrangements and sorting by reversals*, SIAM J. on Computing **25** (1996), pp. 272–289.

[BF95] P. Berman and T. Fujito, *Approximating Independent Sets in Degree 3 Graphs*, Proc. 4th Workshop on Algorithms and Data Structures, LNCS Vol. 955, Springer-Verlag, 1995, pp. 449–460.

[BF94] P. Berman and M. Fürer, *Approximating Maximum Independent Set in Bounded Degree Graphs*, Proc. 5th ACM-SIAM SODA (1994), pp. 365–371.

[BH96] P. Berman and S. Hannenhali, *Fast Sorting by Reversals*, Proc. 7th Symp. on Combinatorial Pattern Matching, 1996, pp. 168–185.

[BS92] P. Berman and G. Schnittger, *On the Complexity of Approximating the Independent Set Problem*, Information and Computation **96** (1992), pp. 77–94.

[B78] B. Bollobás, *Extremal Graph Theory*, 1978, Academic Press.

[C97] A. Caprara, *Sorting by reversals is difficult*, Proc. 1st ACM RECOMB (Int. Conf. on Computational Molecular Biology), 1997, pp. 75–83.

[C98] D.A. Christie, *A 3/2-Approximation Algorithm for Sorting by Reversals*, Proc. 9th ACM-SIAM SODA (1998).

[CK97] P. Crescenzi and V. Kann, *A Compendium of NP Optimization Problems*, Manuscript, 1997;
available at `http://www.nada.kth.se/theory/problemlist.html`

[FG95] U. Feige and M. Goemans, *Approximating the Value of Two Prover Proof Systems with Applications to MAX-2SAT and MAX-DICUT*, Proc. 3rd Israel Symp. on Theory of Computing and Systems, 1995, pp. 182–189.

[GW94] M. Goemans and D. Williamson, *.878-Approximation Algorithms for MAX CUT and MAX 2SAT*, Proc. 26th ACM STOC (1994), pp. 422–431.

[H96] J. Håstad, *Clique is Hard to Approximate within* $n^{1-\epsilon}$, Proc. 37th IEEE FOCS (1996), pp. 627–636.

[H97] J. Håstad, *Some optimal Inapproximality results*, Proc. 29th ACM STOC, 1997, pp. 1–10.

[HP95] S. Hannenhali and P. Pevzner, *Transforming Cabbage into Turnip (Polynomial time algorithm for sorting by reversals)*, Proc. 27th ACM STOC (1995), pp. 178–187.

[KST97] H. Kaplan, R. Shamir and R.E. Tarjan, *Faster and simpler algorithm for sorting signed permutations by reversals*, Proc. 8th ACM-SIAM SODA, 1997, pp. 344–351.

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

[TSSW96] L. Trevisan, G. Sorkin, M. Sudan and D. Williamson, *Gadgets, Approximation and Linear Programming*, Proc. 37th IEEE FOCS (1996), pp. 617–626.