On Some Tighter Inapproximability Results

Piotr Berman* Marek Karpinski†

Abstract

We prove a number of improved inaproximability results, including the best up to date explicit approximation thresholds for MIS problem of bounded degree, bounded occurrences MAX-2SAT, and bounded degree Node Cover. We prove also for the first time inapproximability of the problem of Sorting by Reversals and display an explicit approximation threshold. This last problem was proved only recently to be NP-hard, in contrast to its *signed* version which is computable in polynomial time.

*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

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 applies to the number of problems including Maximum Independent Set (d-MIS) of bounded degree, bounded degree Node Cover, and bounded occurrence MAX-2SAT (cf. [PY91], [A94], [BS92], [BF94], [BF95], [AFWZ95]). 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, it signed version can be computed in polynomial time [HP95], [BH96], [KST97].

The core of the new method is the restricted version of the E2-LIN-2 problem studied in [H97]. We denote by E2-LIN-2 the problem of maximizing the number of satisfied equations for a given number of linear equations mod 2 with exact 2 variables per equation. We denote by 3-OCC-E2-LIN-2 the E2-LIN-2 problem restricted to equations with every variable occuring in at most three equations.

Denote by k-OCC-MAX-2SAT the MAX-2SAT restricted for formulas in which no variable occurs more than k times.

The rest of the paper proves the following main theorem:

Theorem 1. *For every* $\epsilon > 0$

(i) it is NP-hard to approximate E2-LIN-2 within factor $332/331 - \epsilon$*, even if each variable occurs in at most three equations (3-OCC-E2-LIN-2);*

(ii) it is NP-hard to approximate 4-MIS within factor $556/555 - \epsilon$*;*

(iii) it is NP-hard to approximate MIN-SRB within factor $1237/1236 - \epsilon$*.*

Our proof can be easily extended to provide explicit inapproxibility constants for many other problems that are related to bounded degree graphs. E.g., we get 1676/1675 for 3-MIS, 332/331 for 5-MIS, 341/340 for NodeCover in graphs of degree 5 and 668/667 for MAX-2SAT restricted to sets of clauses in which no variable occurs more than six times (6-OCC-MAX-2SAT). We provide the proof sketches in Section 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 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-MIS	1.2	1.0005
4-MIS	1.4	1.0018
5-MIS	1.6	1.0030
MIN-SRB	1.5	1.0008
5-NodeCover	1.375	1.0029
6-OCC-MAX-2SAT	1.0741	1.0014

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 our discussion, we prefer to view it as the following graph problem. Given is 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 second reduction will have instance reduction τ_2; $\tau_1(\tau_2(G))$ is an instance of the maximum independent set with the graph of degree 4. The reduction τ_2 will replace each node of $\tau_1(G)$ with a small gadget.

The next problem we consider is a *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 independent set, 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-SRB (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 (rather than deterministic log-space). In this reduction, every node (variable) is replaced with a *wheel*, a random graphs that is defined below (some parts of this definition will not be used to describe the reduction, but later, in the proof of correctness). The parameter κ used here is a small constant; in this version of the paper we sketch the proof that $\kappa = 9$ sufficiently large, in the full version we show that $\kappa = 6$ is also sufficient.

Definition 2. An *r-wheel* is a graph with $2\kappa r$ 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 $B = A \cap Checkers$. A set $B \subset Checkers$ is *isolated* iff no edges in M connect B with $Checkers - B$.

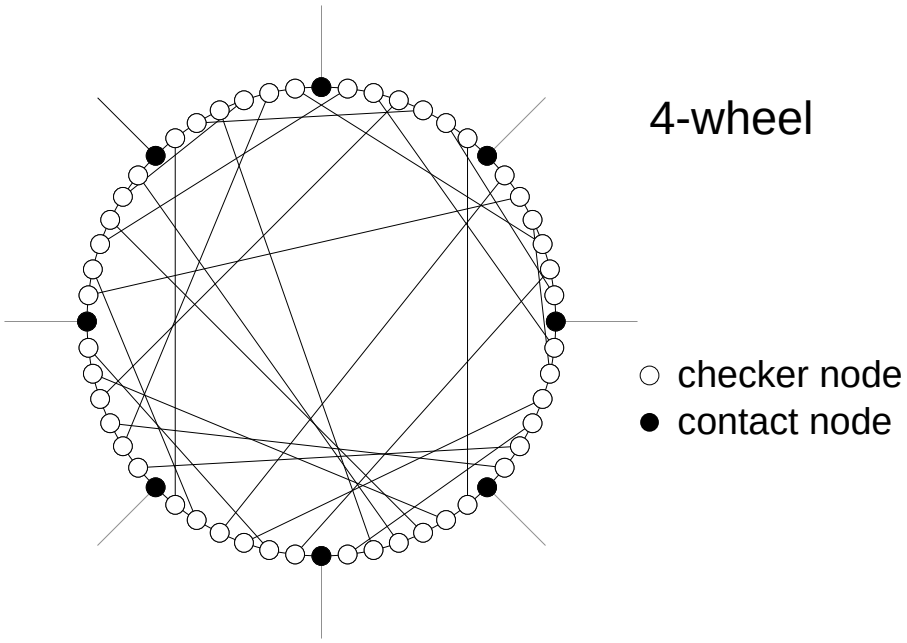


Figure 1

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 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 $(12 - \epsilon_1)n$ and those that have $Score$ below $(11 + \epsilon_2)n$, where the positive constants ϵ_1, ϵ_2 can be arbitrarily small. By showing that our reduction is correct for $\kappa = 6$ we will prove

Theorem 3. *For any positive ϵ_1, ϵ_2, it is NP-hard to decide whether an instance of τ_1(E2-LIN-2) with $336n$ edges (equations) has Score above $(332 - \epsilon_1)n$ or below $(331 + \epsilon_2)n$.*

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 establish 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*

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

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 standard estimates.

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 $B \subset$ *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 $Checkers - B$ 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

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

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 conerning 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 of length between 0 and $\kappa - 1$, so it contributes less than most $(a+1)\kappa$ 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

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

To express the latter, we define $\delta(\beta, \gamma)$ so that the size 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

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

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

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

One can quickly check that the above formula is an increasing function of α. Because 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.

4 Reductions to 4-MIS

We can reduce instances of E2-LIN-2 with 3-regular graphs to MIS instances with graphs of degree 4 (we will use 4-MIS to denote this subproblem). Consider an instance of E2-LIN-2, a 3-regular graph G with $2n$ nodes and $3n$ edges. The gadget used to replace each node of G is a 2×4 grid, partitioned into 0-nodes and 1-nodes, as shown Fig. 2a. Three pairs of nodes, each containing a 0-node and a 1-node form *contacts*. A pair of gadgets corresponding to an edge of G is connected as follows: we choose one contact pair in each of the gadgets. If the edge is labelled with 0, we identify the 0-node of one contact pair with the 0-node of the other; if the edge is labeled with 1, then we switch the order of identification. Note that the nodes of the contacts representing the edges of the main (Hamiltonian) cycle of a wheel have degree 4, and the other contacts have degree 3.

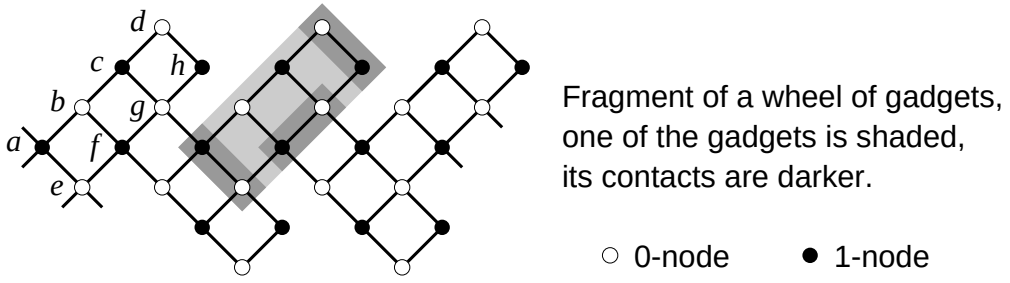


Figure 2a: a part of a 4-MIS instance

The solution translation starts from normalizing the independent set I of the new graph. After the normalization, each gadget is "pure", i.e. the intersection of I with the gadget consists of one type of nodes only. If this type is 1, we include the respective node of G in S (in terms of linear equations, we set the value of the variable to 1). It is easy to see that a normalized I contains 1 node for each node of G plus 1 node for each edge of G that scores 1. In other words, the correspondence between the score s obtained for τ_1(E2-LIN-2) with n nodes and $i = |I|$ is $i = n + s$. Moreover, the resulting 4-MIS instance has $5n$ nodes.

To normalize I, we "purify" gadgets one at the time. To describe a normalization of a gadget Γ, we assume that $\Gamma = \{a, b, \ldots, h\}$, as shown in Fig. 2a. We consider several cases. Assume first that $\{b, c\} \cap I = \emptyset$. Then $\Gamma \cap I$ contains at most 3 nodes. If only one of them (or none) is a 1-node, we change I by inserting b and removing its neighbor (if any), the gadget becomes pure and I is not smaller than before. If $\Gamma \cap I$ contains more than one 1-node, we can achieve the same effect by using c instead of b. Now we consider the case when $b \in I$ (the case of $c \in I$ is symmetric). If Γ is not pure, then $h \in I$ while $f, g \notin I$. If the neighbor of g in the adjacent gadget is not in I, we change I by replacing h with g; otherwise the neighbor of f is not in I and we can replace $\{b, e\}$ with $\{c, f\}$.

Given and instance G of τ_1(E2-LIN-2) with $2n$ nodes and $3n$ edges, our construction creates 4-MIS instance G' with $10n$ nodes, and the correspondence between the size i of an independent set in G' and s, *Score* of the corresponding solution of G is $i = 2n + s$. Together with our previous theorem this implies

Theorem 5. *For any positive ϵ_1, ϵ_2, it is NP-hard to decide whether an instance of 4-MIS with $1120n$ nodes has the maximum size of an independent set above $(556 - \epsilon_1)n$ or below $(555 + \epsilon_2)n$.*

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 in common. If we are "lucky", after the replacement we indeed obtain a breakpoint graph.

Unfortunately, if we apply such translation to a graph from Fig. 2a, we will get a graph violates part (ii) of the definition of BGD. However, this approach is succesful if we apply a somewhat larger gadget shown in Fig. 2b.

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

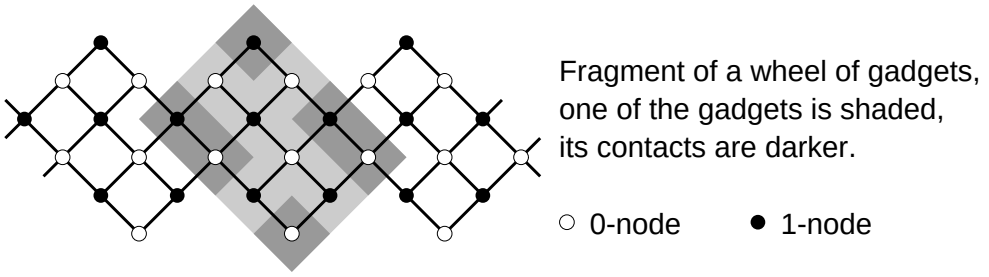


Figure 2b: a part of a 4-MIS instance made of larger gadgets

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.

5 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.

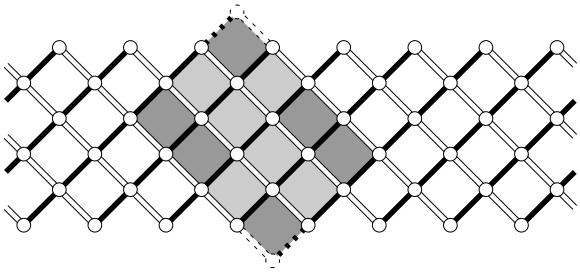
The strategy suggested by these observation is to create instances of BGP in which alternating cycles that either have 2 breakpoints, or $\Omega(\log n)$. Then the task of approximating is equivalent to the one of maximizing the size of independent set in the graph $\mathcal{G}$ of all ACs of 4; we draw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

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



Fragment of a wheel of gadgets, one of the gadgets is shaded, its contacts have darker shade, dash lines show the contact with another wheel.

Figure 3: a part of a BPG instance

Theorem 6. *For any positive ϵ_1, ϵ_2, it is NP-hard to decide whether an instance of BGD with $2240n$ breakpoints has the minimum cost of an alternating cycle decomposition below $(1236 + \epsilon_1)n$ or above $(1237 - \epsilon_2)n$.*

6 Reduction to MIN-SRB

Our reduction from BGD to MIN-SRB 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) form 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 6 apllies also to MIN-SRB.

7 Bounded MIS, NodeCover and MAX-2SAT

The hardness bound of 1676/1675 for 3-MIS follows directly from the construction in Theorem 5 for 4-MIS.

To show that it is hard to approximate 5-MIS within a constant factor better than 332/331, we can reduce E2-LIN-2 to MIS, and then apply Theorem 3. An edge $\{x, y\}$ is replaced with two nodes, each with a pair of

labels; if $l(\{x,y\}) = 1$ (i.e. the edge stand for $x \neq y$), the pairs of labels are $\{x^1, y^0\}$ and $\{x^0, y^1\}$, otherwise (when the edge stands for $x = y$) this pairs are $\{x^1, y^1\}$ and $\{x^0, y^0\}$. Then we introduce an edge between nodes u and v if for some x, u has label x^1 and v has label x^0.

The solution translation is computed as follows. We start with an independent set I for a 5-MIS instance. For each variable/node x of the original instance of 3-OCC-E2-LIN-2 we define a set of nodes V_x that have label x^0 or x^1. If $V_x \cap I$ contains a node with label x^1, then the implied solution S for the original instance contains x. We further normalize the solution by inspecting each edge e of the original instance such that $Score(S, e) = 1$. Suppose that $\{x, y\}$ is such an edge, and that it has the label 0. Then $V_x \cap V_y$ consists of two nodes, with label sets $\{x^0, y^0\}$ and $\{x^0, y^0\}$. One can see that either this pair contains exactly one node of I, or one of these two nodes can be inserted; if $x, y \in S$, then we can insert the node with label set $\{x^1, y^1\}$, and if $x, y \notin S$, we can insert the other node. The case of label 1 is similar. One can see that the normalization may only increase the set S, and after the normalization, $Score(S) = |I|$. Therefore every approximation ratio which is hard for 3-OCC-E2-LIN-2 is also hard for 5-MIS.

To show that it is hard to approximate 5-NodeCover within a constant factor better than 341/340, we can use the same instance reduction. One can observe that if the original graph of 3-OCC-E2-LIN-2 had $336n$ edges, the new graph has $772n$ nodes, and it is hard to distinguish between instances with MIS larger than $(332 - \epsilon_1)n$ nodes and those with $(331 + \epsilon_2)n$; equivalently, it is hard to distinguish between instances with minimal node cover size below $(340 + \epsilon_1)n$ from those above $(341 - \epsilon_2)n$.

The reduction of 3-OCC-E2-LIN-2 to 6-OCC-MAX-2SAT with variables occuring at most six times is very simple: an equality (equivalence) is replaced with the corresponding pair of implications. One can see that for a fixed truth assignment, an equality is satisfied iff both of the corresponding implications are satisfied, otherwise exactly one implication is satisfied. Because it is difficult to decide whether in a given set of $336n$ equations we may have only $(4 + \epsilon_1)n$ unsatisfied ones, or we must have at least $(5 - \epsilon_2)n$, the same is true for the the corresponding 6-OCC-MAX-2SAT instance, thus it is difficult to distinguish between instances with score at least $(2 \cdot 336 - 4 - \epsilon_1)n$ and those with score at most $(2 \cdot 336 - 5 + \epsilon_2)n$.

8 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-SRB is the first inapproximability result for this problem. The especially huge gap between 1.5 and 1.0008 for the MIN-SRB problem reflects a great challenge for future improvements.

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-2-SAT*, 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.