On Some Tighter Inapproximability Results
(Extended Abstract)

Piotr Berman[1] and Marek Karpinski[2]

[1] Dept. of Computer Science, Pennsylvania State University,
University Park, PA 16802
[2] Dept. of Computer Science, University of Bonn,
53117 Bonn

Abstract. We give a number of improved inapproximability results, including the best up to date explicit approximation thresholds for bounded occurence satisfiability problems like MAX-2SAT and E2-LIN-2, and the bounded degree graph problems, 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.

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

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]). In this paper we address bounded instances of the classic NP-hard optimization problems and some related problems. The method uses randomized reductions and applies to a 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. We apply also this method to prove approximation hardness of the problem of *sorting by reversals*, MIN-SBR, the problem motivated by molecular biology [HP95] (and with a long history of related research, cf., e.g., [GP79], [CB95]), only recently proven 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

J. Wiedermann et al. (Eds.): ICALP'99, LNCS 1644, pp. 200–209, 1999.

of maximizing the number of satisfied equations from a given set of linear equations mod 2 with 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. 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 occurrences 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 paper proves the following main theorem:

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

(1) 3-OCC-E2-LIN-2 and 3-MAX CUT within factor $332/331 - \epsilon$*;*
(2) 6-OCC-MAX 2SAT within factor $668/667 - \epsilon$*;*
(3) 3-OCC-MAX 2SAT within factor $2012/2011 - \epsilon$*;*
(4) 3-OCC-E3-LIN-2 within factor $62/61 - \epsilon$*;*
(5) 4-MIS within factor $74/73 - \epsilon$ *and 4-Node Cover within* $79/78 - \epsilon$*;*
(6) 5-MIS within factor $68/67 - \epsilon$ *and 5-Node Cover within* $74/73 - \epsilon$*;*
(7) 3-MIS within factor $140/139 - \epsilon$ *and 3-Node Cover within* $145/144 - \epsilon$*;*
(8) MIN-SBR within factor $1237/1236 - \epsilon$*.*

All these results rely on the reduction to show (1), which forms structures that can be translated into many graph problems with very small and natural gadgets. The complete proofs are to be found in [BK99].

The gaps between the upper and lower approximation bounds are summarized in Table 1. The upper bounds are from [GW94], [BF95], [C98], and [FG95].

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 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 labeling function. For $S \subset V$, let χ_S be the characteristic function of S. We define $Score(S, \{u, v\}) = \chi_S(u) \oplus \chi_S(v) \oplus l(\{u, v\})$. 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 τ_1(E2-LIN-2) are important in the subsequent reductions that lead to MIN SBR problem.

To obtain other inapproximability results, we alter the reduction τ_1 in two ways. The first modification results in graphs that have all edges labeled with

Table 1. Gaps between known approximation bounds.

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

1, i.e. it reduces E2-LIN-2 to 3-MAX CUT and allows to complete the proof of (1). 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 (4).

To show (2), 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 (7) and (5) 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 labeling function, which satisfies the following two properties:

(1) for $b \in \{0, 1\}$, each connected component of $\langle V, l^{-1}(b)\rangle$ is a simple path;
(2) 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 (as done 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 BGD, 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. The parameter κ used here equals 6.

Definition 1. 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).

Consider an instance G of E2-LIN-2 with n nodes (variables) and m edges (equations). We will transform G into $\tau_1(G)$, an instance of 3-OCC-E2-LIN-2. 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 labeled 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.

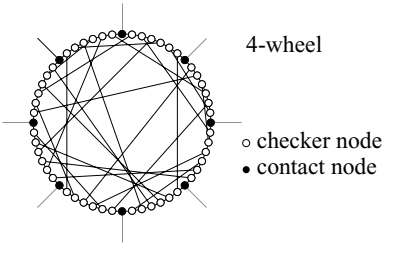

Fig. 1. A very small example of a gadget used by τ_1.

We need to elaborate this construction a bit to assure that $\tau_1(G)$ has a large girth. 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 different 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 that include edges between the wheels; this can be done using a construction of Bollobás [B78].

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 $S - W$. We convert a normalized solution S into a solution S' of the original problem in an obvious manner: a node belongs to S' iff its wheel is contained in S. We can show that the probability that the normalization decreases the score is very low. Assuming that G has m edges/equations, we have $Score(S) = 2k((3\kappa+2)m + Score(S'))$. 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 prove

Theorem 2. *For any $\epsilon \in (0, 1/2)$, 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 $(331+\epsilon)n$.*

4 From HYBRID to *k*-MIS

We can modify τ_1 to transform E3-LIN-2 rather than E2-LIN-2. Variables (nodes) are still replaced with kd-wheels, and an equation $x \oplus y \oplus z$ is replaced with $2k$ equations, each involving one contact from each of the respective consistency wheels. HYBRID is the resulting set of instances of E3-LIN-2. By analizing this reduction, we can show (4). Furthermore, we can efficiently reduce HYBRID to k-MIS for $k = 3, 4, 5$.

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 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. Our best reduction to 5-MIS differs only ib the shape of the contact gadget. Fig. 2 and 3 show these gadgets in detail.

Solution translation is relatively simple. First we show that we can modify each solution S so that each checker and each contact gadget is pure, i.e. it contains only one kind of nodes from S (0-nodes or 1-nodes) and then we perform appropriate accounting. Note that if all gadgets are pure, than the solution translation is obvious: if it contains 0-nodes from S, the respective Boolean variable receives value 0, and similarly for 1.

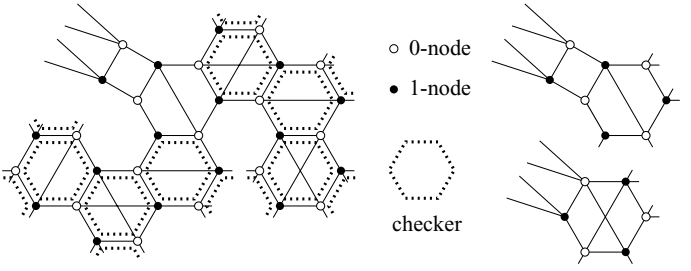

Fig. 2. Consistency wheel for 4-MIS and 5-MIS. The gadget used by 4-MIS to replace a contact node is shown in the upper right corner. The lower right corner is a gadget replacing a contact node in a reduction to 5-MIS.

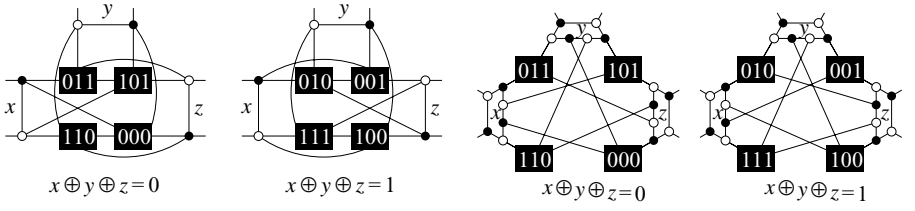

Fig. 3. Equation gadgets for 4-MIS and 5-MIS (left) and the gadgets for 3-MIS (right).

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 augmented with a *trapezoid*, 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 trapezoids 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 trapezoid and 4 special nodes, for the total of 22 nodes. The details are shown in Fig 4 and Fig. 3.

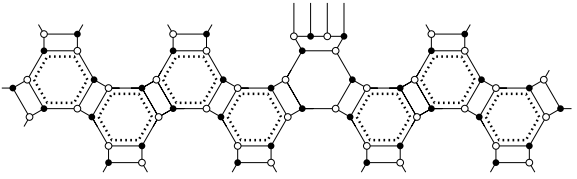

Fig. 4. Consistency wheel for 3-MIS.

The analysis of this reduction (and the two preeceding ones) allows us to prove the following theorem:

Theorem 3. *For any $\epsilon \in (0, 1/2)$, 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 $(73+\epsilon)n$.*
Further, it is NP-hard to decide if an instance of 4-MIS with $152n$ nodes has the maximum size of an independent set above $(74-\epsilon)n$ or below $(73+\epsilon)n$.
Moreover, it is NP-hard to decide if an instance of 3-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, BGD and MIN-SBR

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 5.

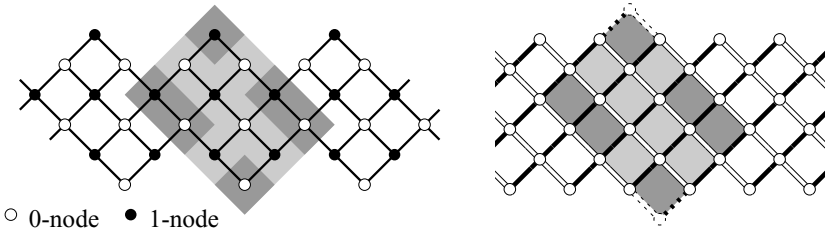

Fig. 5. A part of 4-MIS instance obtained from τ_1(E2-LIN-2) (left) and its translation within BGD instance (right).

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 proceeds differently than before, for details we refer to [BK99].

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 BGD 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 length 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 $\tau_2(\tau_1$(E2-LIN-2)). Fig. 5 shows how this replacement is applied to a cycle of gadget forming a consistency wheel. One of the gadget is shown shaded, and dark gray indicates overlaps with other gadgets; the overlap with a gadget from another wheel consists of two disconnected pieces (note that it exists if this gadget replaces a contact node). The union of ACs used in the replacements is also a disjoint union of 5 ACs (in Fig. 5 these ACs are horizontal zigzags). To apply the reasoning of the previous sections, we need to establish that no cycles of length larger than 4 have to be considered. Here, we omit this reasoning.

At this point the translation is still not correct, as the resulting graphs MUST violated property (i) of BGD: edges of one kind form a collection of cycles: in Fig. 5 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 2 this implies

Theorem 4. *For any $\epsilon \in (0, 1/2)$, it is NP-hard to decide whether an instance of BGD with $2240n$ breakpoints has the minimum cost of an alternating cycle decomposition below $(1236 + \epsilon)n$ or above $(1237 - \epsilon)n$.*

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. The number of reversals needed to sort the resulting permutation is equal to the number of black edges in G, minus the number of cycles in in the optimum cycle cover, plus the number of hurdles, plus 1 if there is a fortress. Therefore the difference between the cost of solution for G differs from that for $\pi(G)$ by the number of hurdles (possibly, plus 1). Now recall that we started from an instance of E2-LIN-2 problem with some n variables and m equations, $n < m$. Our instance of BGD has $\Theta(mn)$ nodes and edges. We can show that the number of hurdles is not larger n, than the number of consistency wheels. In a nutshell, a hurdle is a connected components of the breakpoints in so-called *interleaving graph* that satisfies certain conditions. We can show that breakpoints from each consistency wheel belong to a single connected component. Because n is much smaller than the solution size, we can conclude that Theorem 4 applies also to MIN-SBR.

6 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.

Acknowledgments

This research was partially supported by NSF grant CCR-9700053, International Computer Science Institute, Berkeley, California, DIMACS, DFG grant 673/4-1, ESPRIT BR grants 7079, 21726 and EC-US 030, and by the Max–Planck Research Prize.

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.

[BK99] P. Berman and M. Karpinski, *On Some Tighter Inapproximbility Results*, preliminary version appeared in ECCC TR98-065, the full version available under `http://theory.cs.uni-bonn.de/~marek`.

[BS92] P. Berman and G. Schnitger, *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), pp. 244–252.

[CB95] D. Cohen and M. Blum, *Improved Bounds for Sorting Burnt Pancakes*, Discrete Applied Mathematics, Vol. 61, pp. 105-125.

[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.

[GP79] W. H. Gates, and C. H. Papadimitriou, *Bounds for Sorting by Prefix Reversals*, Discrete Mathematics **27** (1979), pp. 47–57.

[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 Inapproximability 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.