A threshold of $\ln n$ for approximating set cover

(Preliminary version)

Uriel Feige*
Department of Applied Math and Computer Science
The Weizmann Institute
Rehovot 76100, Israel
feige@wisdom.weizmann.ac.il.

Abstract

We prove that $(1 - o(1)) \ln n$ is a threshold below which set cover cannot be approximated efficiently, unless NP has slightly superpolynomial time algorithms. This closes the gap (up to low order terms) between the ratio of approximation achievable by the greedy algorithm (which is $(1-o(1)) \ln n$), and previous results of Lund and Yannakakis, that showed hardness of approximation within a ratio of $(\log_2 n)/2 \simeq 0.72 \ln n$.

1 Introduction

Let S be a set of n points and $\mathcal{F} = \{S_1, S_2, \dots S_m\}$ a collection of subsets of S. *Set cover* is the problem of selecting as few as possible subsets from $\mathcal{F}$ such that every point in S is contained in at least one of the selected subsets. This problem is NP-hard, but can be approximated within a ratio of $\ln n$, where ln denotes the natural logarithm. Lund and Yannakakis [13] showed that it is hard to approximate set cover within a ratio of $(\log n)/2$, where log denotes logarithms in base 2. We extend their hardness result, and show that for any $\epsilon > 0$, it is hard to approximate set cover within a ratio of $(1-\epsilon) \ln n$. We prove our hardness result under the assumption that NP does not have $n^{O(\log \log n)}$-time deterministic algorithms. Our result is based on a reduction from a new k-prover proof system for NP, designed specifically for this purpose. (The [13] result is based on a reduction from a 2-prover proof system for NP.)

1.1 Related work

Let $H(n) = \sum_{i=1}^{n} 1/i$ denote the *harmonic function*. For any n, $H(n) \leq \ln n + 1$. Two different approaches, one based on a greedy algorithm, the other on a linear programming relaxation of set cover, both approximate set cover within a factor of $H(n)$ [11, 12]. (This was extended by Chvatal [4] to the weighted version of set cover.) Tighter analysis, with attention to low order terms, is provided by Srinivasan [18] (for the linear programming approach) and by Slavik [17] (for the greedy algorithm).

The first hardness of approximation results for set cover followed from work on probabilistically checkable proof systems, where the work of Arora *et.al.* [1], in conjunction with the hardness of set cover with respect to MAX SNP [15] implied that for some $\epsilon > 0$, it is NP-hard to approximate set cover within a ratio of $1+\epsilon$. To present subsequent hardness of approximation results, we let TIME(t) denote the class of languages that have a deterministic algorithm that runs in time t, and let ZTIME(t) denote the class of languages that have a probabilistic algorithm that runs in expected time t (with zero error).

Lund and Yannakakis [13] showed that unless $NP \subset TIME(n^{O(\text{polylog } n)})$ then set cover cannot be approximated within a ratio of $\log n/4$ [1], and that unless $NP \subset ZTIME(n^{O(\text{polylog } n)})$ then set cover cannot be approximated within a ratio of $\log n/2$. Their proof was based on a reduction from efficient two prover proof systems for NP.

Subsequent work focused on improving the efficiency of multi-prover proof systems so as to get hardness results under weaker complexity assumptions. Bellare *et.al.* [2] constructed four prover proof systems that implied that unless P=NP set cover cannot be approximated within any constant ratio, and unless $NP \subset TIME(n^{O(\log \log n)})$ then set cover cannot be approximated within a ratio of $\log n/8$. Improved analysis of two prover proof systems by Raz [16] implies that unless $NP \subset TIME(n^{O(\log \log n)})$ then set cover cannot be approximated within a ratio of $\log n/4$, and that unless $NP \subset ZTIME(n^{O(\log \log n)})$ then set cover cannot be approximated within a ratio of $\log n/2$.

Improved deterministic constructions by Naor *et.al.* [14] closed the gap (up to low order terms) between the consequences achievable under the assumption that NP is not contained in a deterministic time class, and the assumption that NP is not contained in a probabilistic time class. It follows that unless $NP \subset TIME(n^{O(\log \log n)})$ then set cover cannot be approximated within a ratio of $\log n/2$.

In our work we close the gap between the known $\ln n$ approximation ratio and the hardness result of $\log n/2$. We

---

*Incumbent of the Joseph and Celia Reskin Career Development Chair. Supported by a Yigal Alon fellowship.

[1] We use $\log n/4$ as shorthand notation. The exact result was that set cover cannot be approximated within ratio $c \log n$, for any $c < 1/4$. The same abuse of notation is used throughout this section.

STOC'96, Philadelphia PA,USA
© 1996 ACM 0-89791-785-5/96/05..$3.50

show that the upper bound is tight (up to low order terms) under the assumption that $NP \not\subset TIME(n^{O(\log\log n)})$.

There are only few NP optimization problems that are known to have a threshold of a nontrivial nature (e.g., not located at approximation ratio 1). One such example is that of the *minimum p-center* problem, for which Hsu and Nemhauser [10] showed that it is NP-hard to obtain approximation ratios below 2, whereas Hochbaum and Shmoys [9], and Dyer and Frieze [5], showed how to approximate minimum p-center within a factor of 2. Another such example is presented in [8]. Our work provides several other such examples, as the same threshold of $\ln n$ holds for all problems that are equivalent to set cover in terms of approximation ratio, such as *dominating set* (see [13] for more details).

1.2 Overview

We give a simplistic overview of the main ideas in our proof.

The ratio of $\log n/2$ proven by Lund and Yannakakis comes up from the following setting. There is a set S of m points, and a collection $\mathcal{F}$ of random subsets of S, each of size $m/2$. The set S needs to be covered by members of $\mathcal{F}$, and by *complements* of members of $\mathcal{F}$. There are two ways of covering S, the *good* way, and the *bad* way. The good way is to take a member of $\mathcal{F}$ and its complement, thus covering S by only two subsets. The bad way is not to do so, and then essentially we cover S by random and independent subsets. Since each new random subset is expected to cover only half of the yet uncovered elements, we need $\log m$ subsets in order to complete the cover. Thus the ratio between the good case and the bad case is $\log m/2$. Lund and Yannakakis showed how to reduce two prover proof systems for satisfiablity of a formula ϕ to a collection of sets as described above, such that if ϕ is satisfiable, all sets are covered by the good way, and if ϕ is not satisfiable, most sets need to be covered by the bad way.

To prove a $\ln n$ ratio, we consider a modified setting in which there is a set S of m points, and a collection $\mathcal{F}$ of random subsets of S, each of size m/k, where k is a large constant. Each random subset has $k-1$ other pairwise disjoint subsets of size m/k associated with it that together partition S into k equal parts. A good cover of S by disjoint subsets requires only k subsets. A bad cover needs roughly d random subsets (not belonging to the same partition) in order to cover S, where $(1-1/k)^d \simeq 1/m$. As k grows, d tends to $k \ln m$. The ratio between the two cases approaches $\ln m$, as desired.

To make use of the above setting, we design a k-prover proof system for satisfiability. We remark that already in [2] hardness results for set cover were proved using k-prover proof system, where $k = 4$. However, these hardness results gave poorer bounds on the ratio of approximation than those obtainable from 2-prover proof systems. The reason why we obtain stronger bounds, is that unlike previous k-prover proof systems, our proof systems have two types of acceptance criteria. *Strong consistency* is similar to "conventional" acceptance criteria, and requires that the answers of all provers be mutually consistent. *Weak consistency* differs from conventional acceptance conditions in two ways. First, it is not required that the answers of all provers are consistent. For the verifier to weakly accept, it suffices that there are only two out of the k provers whose answers are consistent. Second, even the answers of these two provers need to be consistent only in a weak sense.

In our proof system, the difference between the case that ϕ is satisfiable and the case that ϕ is not satisfiable is not only in the acceptance probability, but also in the acceptance criteria. If ϕ is satisfiable, the provers in our proof system have a strategy that is ***strongly*** consistent. If ϕ is not satisfiable, then any strategy for the provers is ***weakly*** consistent on only a small fraction of the possible questions of the verifier. The gap that we obtain for approximating set cover is due in part to the difference in acceptance probability between the cases that ϕ is satisfiable and ϕ is not satisfiable, and in part to the difference in acceptance criteria.

In Section 2 we describe our k-prover proof system. It is fortunate that we can invoke a recent theorem of Raz [16] regarding reduction of error by parallel repetition, and hence make this description largely self contained. In contrast, Lund and Yannakakis used the more complicated 2-prover proof system of Feige and Lovasz [7] (the result of [16] was not available at the time), and needed to quote its special properties without proof.

In Section 3 we describe the reduction from our k-prover proof system to set cover. In Section 4 we explain how to construct the partition systems mentioned above. In Section 5 we indicate how to get hold of the low order terms when proving hardness of approximation for set cover.

2 A multiprover proof system

Our result is based on a reduction from a multiprover proof system. In this section we explain the construction of the proof system. Readers not familier with multiprover proof systems are referred to [3] (or other references in our paper).

Consider the problem of **MAX 3SAT-5**.

Input: A CNF formula with n variables and $5n/3$ clauses, in which every clause contains exactly three literals (a literal is a Boolean variable in either positive or negated form), every variable appears in exactly five clauses, and a variable does not appear in a clause more than once.

Output: The maximum number of clauses that can be satisfied simultaneously by some assignment to the variables.

Proposition 1 *It is MAX-SNP hard to approximate MAX 3SAT-5: for some $\epsilon > 0$, it is NP-hard to distinguish between satisfiable 3CNF-5 formulas, and 3CNF-5 formulas in which at most an $(1-\epsilon)$-fraction of the clauses can be satisfied simultaneously.*

Proof: By reduction from MAX 3SAT-B (which is NP-hard to approximate, see [15] and [1]).

Consider any variable x that appears b times, where $b \geq 2$ (w.l.o.g., each variable appears at least once). Replace each occurrence of x by a fresh variable x_i, and add the $2b$ clauses $(x_i \vee \bar{x}_{i+1})$, $(\bar{x}_i \vee x_{i+1})$, for $0 \leq i \leq b-1$, where $i+1$ is computed mod b. Now each variable appears exactly 5 times. For clauses that are shorter than three, add a fresh dummy literal $\bar{y}$, and add the following clauses with additional dummy variables z_1 and z_2: $(y \vee z_1 \vee z_2)$, $(y \vee \bar{z}_1 \vee z_2)$, $(y \vee z_1 \vee \bar{z}_2)$, and $(y \vee \bar{z}_1 \vee \bar{z}_2)$. Add a constant number of additional dummy variables w_i so that the total number of variables (of the types x_i, y_i, z_i, and w_i) is devisable by 3, and add dummy 3-CNF clauses that contain distinct dummy variables z_i and w_i in positive form, until each dummy variable occurs exactly five times.

The length of the original 3SAT-B formula increases only by a constant multiplicative factor. The number of unsatisfiable clauses does not change. □

We recall how one-round two-prover proof systems for 3SAT-5 can be constructed. The verifier selects a clause at random, sends it to the first prover, and sends the name of a random variable in this clause to the second prover. Each prover is requested to reply with an assignment to the variables that it received (without seeing the question and answer of the other prover). The verifier accepts if the answer of the first prover satisfies the clause, and is consistent with the answer of the second prover.

Proposition 2 *Under the optimal strategy of the provers, the verifier accepts the 3CNF-5 formula ϕ with probability $(1 - \epsilon/3)$, where ϵ is the fraction of unsatisfiable clauses in ϕ.*

As conjectured by many and proven by Raz [16] the error (which initialy is $1 - \epsilon/3$) can be decreased by repeating this proof system many times in parallel, and accepting if the two provers are consistent on each of the repeated copies independently.

Proposition 3 *If the proof system is repeated ℓ times independently in parallel, then the error is $2^{-c\ell}$, where c is a constant that depends only on ϵ, and $c > 0$ if $\epsilon > 0$.*

Proof: Follows from [16], using the fact that the answer size of the original proof system is constant. □

We construct now a k prover proof system for MAX 3SAT-5. For this, we consider a binary code that contains k code words, each of length ℓ and weight $\ell/2$, and Hamming distance at least $\ell/3$ between any two code words. For our main result we shall choose $\ell = \Theta(\log\log n)$ and k an arbitrarily large constant. In this case, assuming w.l.o.g. that ℓ is an exact power of 2 and that $k < \ell$, the rows of a Hadamard matrix give a code with the desired properties (in fact, with Hamming distance $\ell/2$). For refined results, such as determining the low order terms in the hardness of approximation result for set cover, it may be useful to choose $k > \ell$, and use some other standard code instead of the Hadamard matrix one.

The verifier selects ℓ clauses uniformly and independently at random. Call these clauses $C_1, \ldots, C_\ell$. From each clause, the verifier selects a single variable uniformly and independently at random. Call these variables $x_1, \ldots, x_\ell$. With each prover the verifier associates a code word. Prover P_i receives C_j for those coordinates in its code word that have the bit 1, and x_j for those coordinates in its code word that have the bit 0. Each prover replies with an assignment to all variables that it received. For coordinates in which the prover receives a variable, it replies with an assignment to that variable. For coordinates in which the prover receives a clause, it must reply with an assignment to the three variables that satisfies the clause (any other reply given on such a coordinate is automatically interpreted by the verifier as if it was some canonical reply that satisfies the clause). Hence in this k-prover proof system, the answers of the provers are guaranteed to satisfy all clauses, and only the question of consistency among provers arises.

The verifier *strongly accepts* if on each coordinate the answers of all provers are mutually consistent. The verifier *weakly accepts* if there is a pair of provers that are *weakly consistent* in the following sense: on each coordinates j, their assignments to the respective variable x_j agree. (There is slack in our definition of weak consistency, in the sense that it can be relaxed further without affecting the main results of our paper, provided that Lemma 4 holds.)

Lemma 4 *On input a 3CNF-5 formula ϕ. If ϕ is satisfiable, then the provers have a strategy that causes the verifier to always strongly accept. If at most a $(1-\epsilon)$-fraction of the clauses in ϕ are simultaneously satisfiable, then the verifier weakly accepts with probability at most $k^2 \cdot 2^{-c\ell}$, where $c > 0$ is a constant that depends only on ϵ.*

Proof: If ϕ is satisfiable, then the provers can base their answers on a canonical satisfying assignment (e.g., on the lexicographically first such assignment).

We sketch the proof for the soundness condition. Assume that the verifier weakly accepts with probability at least δ. Then with respect to two of the provers, the verifier accepts with probability at least δ/k^2. By the property of the code, there are at least $\ell/6$ coordinates on which the first of these provers receives a clause, and the other prover receives a variable in this clause. Fix the question pairs in the other $5\ell/6$ coordinates in a way that maximizes the acceptance probability, which by averaging remains at least δ/k^2. Now omit the questions on these $5\ell/6$ coordinates (the provers can reconstruct them anyway). It follows that the two provers have a strategy that succeeds with probability at least δ/k^2 on $\ell/6$ parallel repetitions of the proof system. The proof now follows from Proposition 3, when ℓ is sufficiently large. □

3 The reduction to set cover

Our reduction extends that of Lund and Yannakakis [13].

Let $B(m, L, k)$ be a *partition system* with the following properties.

1. There are m points.
2. There is a collection of L distinct *partitions*.
3. Each *partition* partitions the m points into k disjoint subsets.
4. Any cover of the m points by subsets that appear in pairwise different partitions requires at least $(1 - o(1))k \ln m$ subsets.

In Section 4 we explain how to construct such partition systems with parameters as needed in our reduction from the k-prover proof system for 3SAT-5 to set cover.

The verifier uses a random string of length $(\log 5n/3 + \log 3)\ell = \ell \log 5n$. Let $R = (5n)^\ell$ denote the number of possible random strings for the verifier. With each such random string r, we associate a distinct partition system $B_r(m, L, k)$, where $L = 2^\ell$, and $m = n^{\Theta(\ell)}$. (Altogether there are $N = mR$ points in our set cover problem.) Each partition is labeled by an ℓ bit string p, that corresponds to an assignment to the variables $x_1, \ldots, x_\ell$ asked by the verifier using the random string r. Each subset in a partition is labeled by a unique prover i. With each question-answer pair (q, a) of prover P_i, where $1 \leq i \leq k$, we associate a set $S_{(q,a)}$ as follows. For each r that is consistent with prover P_i being asked question q, for each p that is consistent (coordinatewise) with answer a, set $S_{(q,a)}$ contains the points of the respective subset i.

Let Q denote the number of possible different questions that a prover may receive. A question to a single prover includes $\ell/2$ variables, for which there are $n^{\ell/2}$ possibilities (with repetition), and $\ell/2$ clauses, for which there are $(5n/3)^{\ell/2}$ possibilities. Hence $Q = n^{\ell/2} \cdot (5n/3)^{\ell/2}$. Observe that this number is the same for all provers.

Lemma 5 *If ϕ is satisfiable, then the above set of $N = mR$ points can be covered by kQ subsets. If only a $(1-\epsilon)$ fraction of the clauses in ϕ are simultaneously satisfiable, the above set requires $(1 - f(k))kQ \ln m$ subsets in order to be covered, where $f(k) \to 0$ as $k \to \infty$.*

Proof: If ϕ is satisfiable, consider a satisfying assignment A for ϕ, and let each prover answer consistently with this satisfying assignment. Then for any r, the partition system $B_r(m, L, k)$ is completely covered by the partition p that agrees with A. The number of subsets used is k times the number of possible questions to a single prover.

If only a $(1 - \epsilon)$-fraction of the clauses in ϕ are simultaneously satisfiable, then by Lemma 4 any strategy of the provers (weakly) succeeds with probability at most $k^2 \cdot 2^{-c\ell}$. Assume a cover of size $(1 - \delta)kQ \ln m$, where $\delta > 0$, and derive a contradiction.

Let $\mathcal{C}$ be a collection of sets that covers S, where $|\mathcal{C}| = (1-\delta)kQ \ln m$. With each question q to a prover P_i associate a weight w_q which equals the number of answers a such that $S_{(q,a)} \in \mathcal{C}$. Hence $\sum w_q = |\mathcal{C}|$. With each random string r associate a weight $w_r = \sum_{q \in r} w_q$, where the notation $q \in r$ means that on random string r, one of the provers receives question q. This weight is equal to the number of subsets that participate in covering the m points of $B_r(m, L, k)$. Call r *good* if $w_r < (1 - \delta/2)k \ln m$, and *bad* otherwise.

Lemma 6 *The fraction of good r is at least $\delta/2$.*

Proof: Assume otherwise. Then $\sum_r w_r \geq (1 - \delta/2)^2 kR \ln m > (1 - \delta)kR \ln m$, where R denotes the number of possible random strings of the verifier. On the other hand,

$$\sum_r w_r = \sum_r \sum_{q \in r} w_q = \sum_q \frac{R}{Q} w_q = \frac{R}{Q}|\mathcal{C}|$$

Hence $|\mathcal{C}| > (1 - \delta)kQ \ln m$. Contradiction □

Observe that by property 4 of partition systems, for good r the cover $\mathcal{C}$ must have used two subsets from the same partition p in the cover of $B_r(m, L, k)$. The two provers associated with these subsets give answers consistent with p, and hence are weakly consistent.

Lemma 7 *There exist k provers such that the verifier accepts ϕ with probability at least $2\delta/(k \ln m)^2$.*

Proof: On question q addressed to prover P_i, the prover selects an answer a uniformly at random from the set of answers that satisfy $S_{(q,a)} \in \mathcal{C}$. On good r, there are only $k \ln m$ answers to choose from, and for some pair of answers, the associated subsets belong to the same partition. The probability that this pair is chosen is at least $(k \ln m/2)^{-2}$. Observe that if this pair is chosen then the verifier accepts. The proof now follows from Lemma 6. □

To complete the proof of Lemma 5, observe that $2\delta/(k \ln m)^2 > k^2 \cdot 2^{-c\ell}$, for sufficiently large ℓ (made possible by $\ell = \Theta(\log \log n)$ and $m = n^{\Theta(\ell)}$) □

Recall that $N = mR = m(5n)^\ell$ denotes the number of points in the above set cover problem. If we set $m \geq (5n)^{\ell/\delta}$ we obtain that the set cover problem cannot be approximated within a factor of $(1 - 2\delta) \ln N$, for any δ.

4 Construction of partition systems

We present a randomized construction that with high probability gives a partition system $B(m, L, k)$.

Let d denote $(1 - 2\delta)k \ln m$, for an arbitrary small $\delta > 0$. It is the minimal number of subsets, each belonging to a different partition, that cover the set B. For each point in the set B, for each partition p, decide independently at random in which subset of the partition to place the point.

Consider now a particular choice of d subsets no two of which belong to the same partition. Then the probability for a point to be covered by at least one of the d subsets is $1 - (\frac{k-1}{k})^d$. If there are m points, the probability that all m points are covered by the same d subsets is $(1 - (\frac{k-1}{k})^d)^m$. There are $k^d \binom{L}{d} < L^d$ ways of choosing the subsets (the inequality holds since $d >> k$). Hence the probability that some set of d subsets covers all points is at most $L^d(1 - (\frac{k-1}{k})^d)^m$. We need this probability to be smaller than 1/2, as this gives a probabilistic construction.

For any $\delta > 0$ and sufficiently large k, $(\frac{k-1}{k})^d > e^{-d/k(1-\delta)}$. Hence $(1 - (\frac{k-1}{k})^d) < 1 - e^{-d/k(1-\delta)}$. Now substitute $d = (1 - 2\delta)k \ln m$, and obtain,

$$(1 - (\frac{k-1}{k})^d)^m < (1 - e^{-(1-\delta)\ln m})^m$$

$$< e^{-me^{-(1-\delta)\ln m}} < e^{-m^\delta}$$

Hence indeed, $L^d(1 - (\frac{k-1}{k})^d)^m < 1/2$, for $m^\delta > d \ln L > k\ell \ln m/2$.

This randomized construction can be replaced by a deterministic construction using techniques developed in [14].

5 Refinements

We have shown, that for any ϵ, set cover cannot be approximated efficiently within a ratio of $(1 - \epsilon) \ln n$, unless $NP \subset TIME(n^{O(\log \log n)})$. Can the same threshold be proven under the weakest possible assumption, that $P \neq NP$? This remains an open question. The best that is known under the $P \neq NP$ assumption is that set cover cannot be approximated within any constant ratio. Some of the difficulties involved in extending this result are discussed in [6].

In our hardness of approximation result, ϵ need not be constant. It may be a decreasing function of n. To make ϵ as small as possible, the following changes need to be made:

1. k (the number of provers) needs to be large. The larger k is, the closer our ratio of approximation is to $\ln m$. Choose k exponential in ℓ.

2. m (the number of points in the partition system) needs to be as large as possible relative to R (which is the number of partition systems), as then $\ln m$ approximates $\ln N$ better. For this end, it is useful to strengthen the $P \neq NP$ assumption as much as possible, e.g., to 3SAT-5 $\not\subset ZTIME(2^{n^\eta})$, for some

$0 < \eta < 1$. (If we could decrease R from $n^{O(\log \log n)}$ to $O(n^{\epsilon})$, without increasing the error of the k-prover proof system, then $\ln m$ would approximate $\ln N$ even better.)

Details of the low order terms are omitted from this preliminary version.

Acknowledgements

I thank Moni Naor, Leonard Shulman, and Aravind Srinivasan for a preview of [14].

References

[1] S. Arora, C. Lund, R. Motwani, M. Sudan, M. Szegedy. "Proof verification and hardness of approximation problems". *33rd FOCS, 14–23, 1992.*

[2] M. Bellare, S. Goldwasser, C. Lund, A. Russell. "Efficient probabilistically checkable proofs and applications to approximation". *Proc. of 25th STOC, 1993, 294–304.*

[3] M. Ben-Or, S. Goldwasser, J. Kilian, A. Wigderson. "Multi-prover interactive proofs: how to remove intractability assumptions". *Proc. 20th STOC, 1988, 113–131.*

[4] V. Chvatal. "A greedy heuristic for the set covering problem". *Math. Oper. Res., 4, 1979, 233–235.*

[5] M. Dyer, A. Frieze. "A simple heuristic for the p-center problem". *Oper. Res. Lett., 3, 1985, 285–288.*

[6] U. Feige, J. Kilian. "Impossibility results for recycling random bits in two prover proof systems". *Proc. of 27th Annual ACM Symposium on the Theory of Computing,* 457–468, 1995.

[7] U Feige, L Lovasz. "Two prover one round proof systems: their power and their problems". *Proc. 24th STOC, 1992, 733–744.*

[8] J. Hastad, S. Phillips, S. Safra. "A well characterized approximation problem". *Proc. 2nd Israel Symp. on Theory of Computing and Systems, 261–265, 1993.*

[9] D. Hochbaum, D. Shmoys. "A best possible approximation algorithm for the k-center problem". *Math. Oper. Res., 10, 1985, 180–184.*

[10] W. Hsu, G. Nemhauser. "Easy and hard bottleneck location problems". *Discrete Applied Math., 1, 1979, 209–216*

[11] D. Johnson. "Approximation algorithms for combinatorial problems" *J. Comput. System Sci. 9. 1974, 256–278.*

[12] L. Lovasz. "On the ratio of the optimal integral and fractional covers". *Discrete Mathematics 13 (1975) 383–390*

[13] C. Lund, M. Yannakakis. "On the hardness of approximating minimization problems". *JACM 41(5), 1994, 960–981*

[14] M. Naor, L. Schulman, A. Srinivasan. "Splitters and near-optimal derandomization". *Proc. of 36th Annual Symposium of Foundations of Computer Science,* 182–191, 1995.

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

[16] R. Raz. "A parallel repetition theorem". *Proc. of 27th Annual ACM Symposium on the Theory of Computing,* 447–456, 1995.

[17] P. Slavik. "A tight analysis of the greedy algorithm for set cover". *Proc. of 28th Annual ACM Symposium on Theory of Computing,* 1996.

[18] A. Srinivasan. "Improved approximations of packing and covering problems". *Proc. of 27th Annual ACM Symposium on the Theory of Computing,* 268–276, 1995.