10. The Hardness of Approximating Set Cover

Alexander Wolff

10.1 Introduction

Imagine the personnel manager of a large company that wants to start a devision from scratch and therefore has to hire a lot of new employees. The company naturally wants to keep labour costs as low as possible. The manager knows that 100 different skills are needed to operate successfully in the new field. After placing some job ads, he gets a list of 1000 applicants each of who posesses a few of the desired skills. Whom should he hire?

This problem has long been known as SETCOVER, the problem of covering a base set (the required skills) with as few elements (the applicants' skills) of a given subset system (the list of applications) as possible. More formally:

<u>SETCOVER</u>

Instance: A base set S, $|S| = N$, and a collection $\mathcal{F} = \{S_1, \ldots, S_M\}$ of M subsets of S, where M is bounded by some polynomial in N.

Problem: Find a subset $\mathcal{C} \subseteq \mathcal{F}$ with minimum cardinality, such that every element of S is contained in at least one of the sets in $\mathcal{C}$.

Karp showed that the decision version of SETCOVER is NP-complete [Kar72], but the following greedy algorithm approximates an optimal solution.

<u>GREEDY_SET_COVER$(S, \mathcal{F})$</u>

$\mathcal{C} := \emptyset$

while $\bigcup \mathcal{C} \neq S$ **do**

  find $S_i \in \mathcal{F}$ such that the gain $|S_i \cap (S \setminus \bigcup \mathcal{C})|$ is maximal

  $\mathcal{C} := \mathcal{C} \cup \{S_i\}$

return $\mathcal{C}$

This simple algorithm always finds a cover $\mathcal{C}$ of at most $H(N) \cdot |\mathcal{C}_{opt}|$ subsets, where $\mathcal{C}_{opt}$ is a minimum cover and $H(n) = \sum_{i=1}^{n} 1/i$ is the n^{th} harmonic number. $H(n) \in \ln n + \mathcal{O}(1)$. For the proof of the greedy algorithm's approximation ratio ([Joh74], [Lov75]) of $H(N)$, see Exercise 10.1.

In this chapter I present the paper "A threshold of ln n for approximating set cover" [Fei96] by Uriel Feige. It shows under some assumption about complexity classes that there is no efficient approximation algorithm with a quality guarantee better than that of the simple greedy algorithm above, namely $\ln N$. This improves a previous result of Lund and Yannakakis [LY93] who showed under slightly stricter assumptions that there is no polynomial $(1/2-\varepsilon)\log_2 N$-approximation algorithm for SETCOVER. Note that $\log_2 N = \log_2 e \cdot \ln N \approx 1.44 \cdot \ln N$. I will use $\log N$ as shorthand notation for $\log_2 N$ from now on.

There are few NP-hard optimization problems with non-trivial approximation ratio for which an approximation algorithm matching the problem's approximation threshold is known. One example is the MIN-P-CENTER problem for which a 2-approximation algorithm exists ([HS85], [DF85]) but it is NP-hard to obtain anything better [HN79]. For the MAPLABELING problem where one is interested in placing rectilinear labels of maximum common size next to n sites, the situation is similar: There is a theoretically optimal approximation algorithm which finds a placement with labels of half the optimal size, but it is NP-hard to get any closer [FW91].

10.1.1 Previous Work

Lund and Yannakakis showed that one cannot expect to find a polynomial-time $(1/2-\varepsilon)\log N$-approximation algorithm for SETCOVER for any $\varepsilon > 0$ [LY93]. In their proof they construct a subset system whose central property is sketched as follows. Let N be the size of the base set S of this system. First chose subsets of S of size $N/2$ independently at random. Put these sets and their complements into the collection $\mathcal{F}$ of subsets of S. Then obviously there are two ways to cover S. The *good* way is to take a set and its complement, thus covering S by only two sets. The *bad* way is not to do so. In this case, S is covered by random and independent subsets. Since each set which is added to the cover is expected to contain only half of the uncovered elements of S, $\log N$ sets are needed to cover S the bad way. Thus the ratio between the good and the bad case is $\frac{1}{2}\log N$.

Lund and Yannakakis use an interactive proof system as a link between the satisfiability of a Boolean formula and the size of a cover. Such systems consist of a verifier which is connected to the input, to a random source, and to two (or more) provers. Based on the input, the verifier uses random bits to pose questions to the deterministic and all-powerful provers. Depending on their answers, the verifier accepts or rejects the input.

Interactive proof systems can be seen as maximization problems. The aim is to find strategies for the provers which maximize the probability with which the verifier accepts the input. Lund and Yannakakis use a two-prover proof system for satisfiability to reduce a formula φ to a collection of sets as described above. They do this by indexing the subsets of this collection with all possible question-answer pairs which can be exchanged between verifier and provers. Their reduction has the property that if φ is satisfiable, the base set can be covered the good way, and if φ is not satisfiable, it must be covered the bad way. Knowing the sizes of a good and a bad cover, we could use a $(1/2-\varepsilon)\log N$-approximation algorithm for SetCover to decide the satisfiability of φ. On the one hand their reduction is in ZTIME($N^{\mathcal{O}(\text{poly}\log N)}$) (see Definition ??), on the other hand every problem in $\mathcal{NP}$ can be reduced in polynomial time to that of testing the satisfiability of a Boolean formula [Coo71]. Therefore we would have $\mathcal{NP} \subseteq$ ZTIME($N^{\mathcal{O}(\text{poly}\log N)}$), which is only a little weaker than $\mathcal{P} = \mathcal{NP}$.

10.1.2 Idea of the Proof

In his proof of the $\ln N$-threshold, Feige constructs a partition system where each partition of the base set consists of k (instead of 2) random subsets of size N/k. k is a large constant. A *good* cover now consists of all k elements of one partition, while in the *bad* case d subsets of pairwise different partitions are needed. These d subsets are added successively to the cover. Each of them contains with high probability only a $1/k$-th of the yet uncovered elements. In other words, we are looking for the smallest d such that the number of uncovered points $N(1-1/k)^d$ is less than 1. This is equivalent to $d \ln \frac{k}{k-1} \approx \ln N$. $\ln n$ grows monotonically, and its slope is $1/n$, which gives us $\frac{1}{k} < \ln \frac{k}{k-1} = \ln k - \ln(k-1) < \frac{1}{k-1}$. For large k this means that $d \approx k \ln N$. Thus the ratio between a bad and a good cover approaches $\ln N$ as desired.

To take advantage of the idea above, the author uses a multi-prover proof system with k provers for the following problem.

ε-RobE3Sat-5

Instance: A Boolean formula φ of n variables in conjunctive normal form in which every clause contains exactly three literals, every variable appears in exactly five clauses, and a variable does not appear in a clause more than once, i.e. φ is in *3CNF-5 form*. Furthermore, φ is either satisfiable or at most a $(1-\varepsilon)$-fraction of its clauses is simultaneously satisfiable.

Question: Is φ satisfiable?

To prove that there is an $\varepsilon \in (0,1)$ such that ε-RobE3Sat-5 is NP-hard is left to the reader, see Exercise 10.2 and Lemma 4.9 in Section 4.5.

k-prover proof systems have been used before for the special case $k = 4$ to show the hardness of approximating SetCover [BGLR93], but the resulting bounds were less tight than those obtained with 2-prover proof systems. Feige's main new idea is to introduce two different acceptance criteria for the verifier: The verifier of his proof system accepts "strongly" if the answers of *all* provers are mutually consistent – this corresponds to the conventional acceptance criterion. For the verifier to accept "weakly", it suffices if *two* of the k provers answer consistently, and their answers only need to be consistent in a weak sense.

The subsets of the partition system sketched above are used to build a large set system to which a Boolean formula φ can be reduced. The sets of this system are indexed by all question-answer pairs which are potentially exchanged between verifier and provers. If φ is satisfiable then it is reduced to a collection of subsets in which there is a cover of few, pairwise disjoint sets. This cover corresponds to the questions of the verifier and the mutually consistent answers of the provers which are based on a satisfying truth assignment for φ. If however only a $(1-\varepsilon)$-fraction of the clauses is satisfiable, then every cover which corresponds to a weak acceptance of the verifier has a cardinality which is greater than the cover in the satisfiable case by at least a factor of $\ln N$. Due to the complexity of the reduction we would have $\mathcal{NP} \subseteq \mathrm{DTIME}(N^{\mathcal{O}(\log\log N)})$ if for any $\varepsilon > 0$ there was a polynomial-time $(1-\varepsilon)\ln N$-approximation algorithm for SetCover. The complexity class $\mathrm{DTIME}(\cdot)$ is introduced in Definition 1.5.

The idea behind introducing the notion of weak acceptance is that it is a good trade-off between the following two opposing wishes. On the one hand, we want to show that the probability that the verifier accepts an unsatisfiable formula is low. On the other hand, we want to be able to design a simple random strategy for the provers which would make the verifier accept with high probability in the case where there were a cover of small size in the subset system. This will give us the necessary contradiction to show that such a cover cannot exist.

The proof of the main theorem, Theorem 10.9, is split as follows. In Section 10.2 we construct the k-prover proof system and show with which probability the verifier accepts a formula. In Section 10.3 we build a partition system with the following two properties. Firstly, the k sets of one partition suffice to cover the base set of the partition system. Secondly, a cover with sets of pairwise different partitions needs more sets by a logarithmic factor. Finally in Section 10.4 we show how to reduce an instance of ε-RobE3Sat-5 to an instance of SetCover. The proof system will act as a mediator between the two. It links the satisfiability of a Boolean formula φ to the size of a cover such that the ε-gap in the number of satisfiable clauses of φ is expanded to a logarithmic gap in the cover size.

10.2 The Multi-Prover Proof System

Our aim is to build a proof system whose provers have a strategy with which they can convince the verifier to ("strongly") accept an input formula if it is satisfiable. At the same time the probability that the verifier accepts ("weakly") should be low for all stategies of the provers if an ε-fraction of the clauses is not satisfiable. This will be used in the reduction to SetCover (Lemma 10.6).

We first turn our attention to a less complex system, a so-called *two-prover one-round proof system*, which Feige describes as follows:

> 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.

For a more detailed and very vivid description of two-prover one-round proof systems refer to Section 6.3. Example 6.2 corresponds to the proof system used here.

Proposition 10.1. *Under the optimal strategy of the provers, the verifier of the two-prover one-round proof system accepts a 3CNF-5 formula φ with probability at most $1 - \varepsilon/3$, where ε is the fraction of unsatisfiable clauses in φ.*

Proof. The strategy of the second prover defines an assignment to the variables of φ. The verifier selects a clause which is not satisfied by this assignment with probability ε. In this case, the first prover must set at least one of the three variables differently from this assignment in order to satisfy the clause, and an inconsistency is detected with probability at least $1/3$. ∎

With the help of Raz' Parallel Repetition Theorem [Raz95] (see Chapter 6) the error in the case of unsatisfiable formulae (which initially is $1 - \varepsilon/3$) 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. Using the terminology of Chapter 6, our "game" has value $1 - \varepsilon/3$ and answer size $s = 2^4$. Then Theorem 6.4 yields the following:

Proposition 10.2. *If the two-prover one-round proof system is repeated l times independently in parallel, then the error is $2^{-c_1 l}$, where c_1 is a constant that depends only on ε, and $c_1 > 0$ if $\varepsilon > 0$.*

But what does the *k-prover proof system* look like? First of all, we need a binary code whose task it will be to encode the structure of the questions the verifier is going to send to the provers. We need k code words of length l, each with $l/2$ ones and $l/2$ zeros. Furthermore, we require that any two code words have a Hamming distance of at least $l/3$. Such codes exist; for instance if $k < l$ and l a power of 2, then a Hadamard matrix fulfills these requirements. Quoting Feige:

> Let l be the number of clauses the verifier picks uniformly and independently at random out of the $5n/3$ clauses of the 3CNF-5 formula. Call these clauses $C_1, \dots, C_l$. From each clause, the verifier selects a single variable uniformly and independently at random. Call these variables $x_1, \dots, x_l$. (We will refer to them as the *sequence of distinguished variables.*) With each prover the verifier associates a code word. Prover P_i receives C_j for those coordinates j in its code word that are one, and x_j

for those coordinates in its code word that are zero. 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 which assign consistently to the sequence of distinguished variables. Observe that they might still assign inconsistently to those variables in a clause which were not distinguished by the random string.

Lemma 10.3. *Consider the k-prover proof system defined above and a 3CNF-5 formula φ. If φ is satisfiable, then the provers have a strategy that causes the verifier to always accept strongly. If at most a $(1-\varepsilon)$ - fraction of the clauses in φ is simultaneously satisfiable, then the verifier accepts weakly with probability at most $k^2 \cdot 2^{-c_2 l}$, where $c_2 > 0$ is a constant which depends only on ε.*

Proof. Suppose φ is satisfiable, then the provers can base their strategy on a fixed truth assignment for φ, say the lexicographically first. Obviously this satisfies all clauses, and makes sure that the answers of the provers are mutually consistent, so the verifier will accept strongly.

If at most a $(1-\varepsilon)$-fraction of the clauses of φ is satisfiable, let γ denote the probability with which the verifier accepts φ weakly. Due to the pigeon-hole principle and the fact that the verifier reacts deterministically to the answers of the provers, there must be one among the $\binom{k}{2}$ pairs of provers with respect to which the verifier accepts at least with probability $\gamma/\binom{k}{2}$.

The code words which correspond to these two provers have a Hamming distance of at least $l/3$. They both have weight $l/2$. Obviously they have the same weight on the part in which they match. This forces them to also have the same weight on those coordinates where they are complementary. Thus, for both, the number of 0-bits equals the number of 1-bits there. It follows that the first prover has a 1-bit and therefore receives a clause on at least $l/6$ coordinates of the code word on which the second prover has a 0-bit and receives a variable. The situation on each of these coordinates is the same as in the two-prover one-round proof system introduced above: the verifier gives a clause to the first, and a variable of this clause to the second prover. It accepts if the answers are consistent. (We assume that clauses are always satisfied.) From Proposition 10.2 we know that the error probability, i.e. the probability with which the verifier accepts an unsatisfiable formula, is reduced to $2^{-c_1 l/6}$ by $l/6$ parallel repetitions of the two-prover system. The answers of the two provers on the other $5l/6$ coordinates can be fixed in a way that maximizes the acceptance probability which by averaging remains at least $\gamma/\binom{k}{2}$ and is less than $2^{-c_1 l/6}$. So $\gamma \leqslant \binom{k}{2} \cdot 2^{-c_2 l} \leqslant k^2 \cdot 2^{-c_2 l}$ for some constant $c_2 = c_1/6$ which, like c_1, depends only on ε. ∎

10.3 Construction of a Partition System

We are searching for a set system whose base set is easy to cover if we take all sets belonging to one partition, and needs roughly a logarithmic factor more sets if we take sets from pairwise different partitions. For each random string which drives the verifier, we will use a distinct copy of the upcoming partition system to be able to build a set system which models the behaviour of our proof system, see Figure 10.1.

Definition 10.4. *Given a constant $c_3 > 0$, a* partition system $B[m, L, k]$ *has the following properties.*

1. *The base set contains m* points*.*
2. *There is a collection of L distinct* partitions*, where $L < (\ln m)^{c_3}$.*
3. *Each partition partitions the base set into k disjoint subsets, where $k < \frac{\ln m}{3 \ln \ln m}$.*
4. *Any cover of the base set by subsets that appear in pairwise different partitions requires more than $(k-2)\ln m$ subsets.*

We want to show that the following randomized construction gives us a partition system which fulfills the properties above with high propability if m is sufficiently large:

> For each partition p decide independently at random in which of its k subsets to place each of the m points of the base set.

Let d denote the threshold $(k-2)\ln m$. Consider a particular choice of d subsets of pairwise different partitions and show that with high probability these do not cover the base set. The probability for a single point to be covered is $1 - (\frac{k-1}{k})^d$. This can be transformed to

$$1 - (1 - 1/k)^{k \cdot (1-2/k) \ln m} < 1 - e^{-(1+1/k)(1-2/k) \ln m} < 1 - m^{-(1-1/k)}$$

by applying the inequalities $(1 - 1/k)^k > e^{-(1+1/k)}$ for $k \geqslant 2$ and

$$(1 + 1/k)(1 - 2/k) = (1 - 1/k - 2/k^2) < (1 - 1/k).$$

The probability that all m points are covered is therefore $(1-m^{-1+1/k})^m$, which is less than $e^{-m^{1/k}}$.

How many ways are there for choosing the d subsets from our partition system? First we choose d out of L partitions, and then from each of these one out of the k subsets it consists of. This gives us $k^d\binom{L}{d} = \frac{k^d}{d!} \cdot \frac{L!}{(L-d)!}$ ways. Since $d \gg k$ the first fraction is less than 1, and the whole expression less than L^d. Substituting $L < (\ln m)^c$ and $d < k \ln m$ gives us that the number of choices is less than $L^d < (\ln m)^{ck \ln m}$.

But what is the probability that any of these choices covers the whole base set? Obviously it is less than $(\ln m)^{ck \ln m} \cdot e^{-m^{1/k}}$. If we can show that this converges to zero for sufficiently large m, then we have reached our goal. We can do this by applying the natural logarithm and then showing that $(ck \ln m \ln \ln m - m^{1/k})$ tends to $-\infty$. We know that $k < \frac{\ln m}{3 \ln \ln m}$, so we are certainly on the safe side if $(\frac{c}{3} \ln^2 m - m^{\frac{3 \ln \ln m}{\ln m}})$ diverges. If we apply the monotonically growing natural logarithm now to both (positive) summands, we get $\ln \frac{c}{3} + 2 \ln \ln m$ and $3 \ln \ln m$. Now we see that the second summand will dominate the first in the long run, so the sum will diverge, and the probability for a cover will converge to zero, which proves that the probabilistic construction works.

Proposition 10.5. *The construction of a partition system $B[m, L, k]$ is in* $\mathrm{ZTIME}(m^{\mathcal{O}(\log m)})$.

Proof. The randomized construction described above requires time polynomial in m. Exhaustively checking that it has Property 4 can be done in time roughly $k^d\binom{L}{d}$ – choose d out of L partitions and in each partition one out of k subsets. As we have seen above, this is less than $(\ln m)^{ck \ln m}$. Substituting $k < \frac{\ln m}{3 \ln \ln m}$ gives us $(\ln m)^{c \ln^2 m / 3 \ln \ln m}$ which is in $m^{\mathcal{O}(\log m)}$ since (applying the natural logarithm to both expressions) $\frac{c}{3} \ln^2 m$ is in $\mathcal{O}(\log m) \cdot \ln m$. The expected number of times the randomized construction needs to be tried until it succeeds is less than $2 = \sum_{i=1}^{\infty} \frac{i}{2^i}$ since it succeeds at least with probability $1/2$ for m sufficiently large. ∎

The author states that the randomized construction can be replaced by a deterministic construction using techniques developed in [NSS95].

10.4 Reduction to Set Cover

Finally we have to show how to reduce an instance of ε-RobE3Sat-5 to an instance of SetCover. The reduction must expand the ε-gap in the number of unsatisfiable clauses of a 3CNF-5 formula to a logarithmic gap in the size of a cover for the base set. By supplying us with the indices of the subsets, the proof system serves as a link between the two instances.

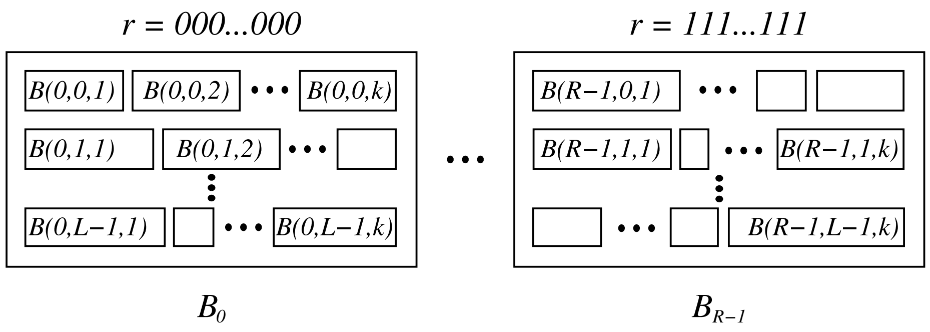

Fig. 10.1. The building blocks of the subset system.

Recall that the verifier has to select l clauses from the 3CNF-5 formula, and to distinguish a variable from each of these clauses at random. In order to do so, the verifier needs $l\cdot(\log 5n/3+\log 3) = l \log 5n$ random bits. Let $R = 2^{l \log 5n} = (5n)^l$ be the number of 0-1-strings of this length. With each such string r, we associate a *distinct* partition system $B_r = B[m, L, k]$, where $L = 2^l$ and $m = n^{\Theta(l)}$. So the base set S of our set system will have $N = mR$ points. As we know, each random string distinguishes a sequence of l variables. Each of the $L = 2^l$ partitions in B_r is labeled with an l-bit string which corresponds to a truth assignment to the sequence of distinguished variables. Each subset in a partition is labeled by the index of a unique prover P_i. We use $B(r, j, i)$ to denote the i-th subset of partion j in partition system B_r, see Figure 10.1.

The sets of our set system will consist of a number of such partition subsets $B(r, j, i)$. A random string r and the index i of a prover tell the verifier unequivocally (via the prover's code word, see Page 247) for which of the $l/2$ coordinates encoded by r it has to ask the prover for an assignment to all variables of a clause, and for which of the $l/2$ coordinates it only has to ask for an assignment to a single variable. Thus the question q which a verifier might ask a prover is a function π of r and i. π extracts the random bits which are relevant for prover P_i from r according to code word C_i, in other words $q = \pi(r, i)$. Feige uses the notation $(q, i) \in r$ to say that "on random string r prover P_i receives question q". Now we can describe how the proof system is reduced to a family $\mathcal{F}$ of subsets of S. With each question-answer pair $[q, a]$ which is potentially exchanged between the verifier and prover P_i we associate the subset

$$S_{(q,a,i)} = \dot{\bigcup_{(q,i)\in r}} B(r, a_r, i)$$

where a_r is a projection of the prover's answer a to the sequence of distinguished variables. For each clause for which the prover is supposed to answer with a truth assignment to all of its three variables, this projection simply ignores the assignments to the two variables which were not distinguished. Note that the prover's index i in $S_{(q,a,i)}$ is redundant since it can be deduced from the question q and the code words, but used nevertheless for clarity.

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

The following key lemma exploits the fact that the reduction above produces subset families which depend strongly on the answers of the provers. The idea is that if a formula φ is satisfiable, then there is a strategy for the provers to answer consistently, and the subsets corresponding to these answers fit nicely together. They will even be disjoint, hence few are needed to cover the base set. If however φ is not satisfiable, we will use a probabilistic argument to show that approximately by a logarithmic factor more sets are needed for a cover in the subset system to which φ is reduced in this case.

Lemma 10.6. *If φ is satisfiable, then the base set S of $N = mR$ points can be covered by kQ subsets. If only a $(1-\varepsilon)$-fraction of the clauses in φ is simultaneously satisfiable, S requires more than $(k-4)Q\ln m$ subsets in order to be covered. Thus the ratio between the two cases is at least $(1-4/k)\ln m$.*

Proof. Suppose there is a satisfying truth assignment for φ. Fix a strategy for the provers which is consistent with it. Let $a_1, \ldots, a_k$ be the provers' answers to the verifier's questions $q_1 = \pi(r,1), \ldots, q_k = \pi(r,k)$ with the notation used above. Then for any random string r (i.e. no matter what the verifier asks), these answers are consistent and map to the same assignment a_r for the sequence of distinguished variables. Hence the sets $S_{(q_1,a_1,1)}, \ldots, S_{(q_k,a_k,k)}$ contain $B(r,a_r,1), \ldots, B(r,a_r,k)$ and therefore cover B_r. Since this holds for all random strings, the collection of all subsets $S_{(q,a,i)}$, where a is the answer given by prover P_i under the strategy above, covers the whole base set S. The number of sets used is the product of k, the number of provers, and Q, the number of possible questions to a single prover. These kQ subsets are disjoint.

Now consider the case that at most a $(1-\varepsilon)$-fraction of the clauses in φ is simultaneously satisfiable. Lemma 10.3 tells us that the provers have no strategy which succeeds in convincing the verifier to accept with probability greater than $k^2 \cdot 2^{-c_2 l}$. Assume we had a cover $\mathcal{C} \subseteq \mathcal{F}$ of size $(k-4)Q\ln m$ (i.e. just below the logarithmic threshold). If we can find a strategy for the provers based on this cover which succeeds with a probability that is greater than Lemma 10.3 allows, then we have the contradiction needed.

We map our cover $\mathcal{C}$ back into a proof system. With each question q to a prover P_i, we associate a weight $w_{q,i}$ which equals the number of answers a such that $S_{(q,a,i)} \in \mathcal{C}$. Hence $\sum_{q,i} w_{q,i} = |\mathcal{C}|$. Now with each random string r we associate the weight $w_r = \sum_{(q,i)\in r} w_{q,i}$. This weight equals the number of subsets that participate in covering B_r since

$$w_r = \sum_{(q,i)\in r} w_{q,i} = \sum_{(q,i)\in r} \left|\{a : S_{(q,a,i)} \in \mathcal{C}\}\right| = \left|\{S_{(q,a,i)} \in \mathcal{C} : (q,i) \in r\}\right|.$$

We say that a random string r is *good* if $w_r < (k-2)\ln m$.

Proposition 10.7. *Given a cover $\mathcal{C}$ of size $(k-4)Q\ln m$, the fraction of good r is at least $2/k$.*

Proof. Let $G = \{r : w_r \geqslant (k-2)\ln m\}$ be the set of *bad* random strings. Assume $|G| > (1-2/k)R$ where R is the number of possible random strings. Then

$$\sum_r w_r \geqslant \sum_{r\in G} w_r \geqslant (1-2/k)R \cdot (1-2/k)k\ln m > (1-4/k)kR\ln m$$

On the other hand, using the definition of w_r, we have

$$\sum_r w_r = \sum_r \sum_{(q,i)\in r} w_{q,i} = \sum_{q,i} \frac{R}{Q} w_{q,i} = \frac{R}{Q} \cdot |\mathcal{C}|$$

Recall that Q denotes the number of possible different questions that a prover may receive. Then R/Q is the number of random strings which cause the verifier to send a particular question q to a particular prover. This explains the second equality. Putting the two lines above together, we get $|\mathcal{C}| > (k-4)Q\ln m$ which is a contradiction. ∎

Proposition 10.8. *Given a cover $\mathcal{C}$ of size $(k-4)Q\ln m$. Then there is a strategy for the provers such that the verifier accepts φ with probability at least $8/(k^3 \ln^2 m)$.*

Proof. We will show that the following simple random strategy which is based on the assumed cover $\mathcal{C}$ succeeds with the desired probability.

> On receiving question q, P_i selects an answer a uniformly at random from the set of answers that satisfy $S_{(q,a,i)} \in \mathcal{C}$.

For a fixed random string r there is a one-to-one correspondence between the sets $B(r,p,i)$ which participate in the cover of B_r, and sets $S_{(q,a,i)}$ that belong to $\mathcal{C}$: on the one hand, it is clear that a set $S_{(q,a,i)}$ contains at most one $B(r,p,i)$ from partition system B_r, namely that with $p = a_r$. (Recall that a_r is the projection of the answer a of prover P_i to an assignment a_r for the sequence of variables distinguished by r.) On the other hand, random string r forces the verifier to pose prover P_i a unique question q.

It is enough to focus on good r, and to compute a lower bound for the probability that the verifier accepts if it receives a random string of this type. Recall Property 4 of our partition system which states that a cover of the m points of B_r by subsets that appear in pairwise different partitions requires more than $(k-2)\ln m$ subsets. Since we know that a good r is covered by fewer subsets, $\mathcal{C}$ must have used two (different) subsets $B(r,p,i)$ and $B(r,p,j)$ from the same partition p. Let $S_{(q_i,a_i,i)}$ and $S_{(q_j,a_j,j)}$ be the two sets in $\mathcal{C}$ which the above mentioned one-to-one correspondence yields.

Consider what happens when the random source supplies the verifier with a good random string r. The verifier poses question q_i to P_i and q_j to P_j. Let $A_i = \{a | S_{(q_i,a,i)} \in \mathcal{C}\}$ be the set of answers prover P_i can choose from. Define A_j analogously. Using the random strategy described above, P_i will pick a_i with probability $1/|A_i|$, P_j analogously. $|A_i|+|A_j| = w_{q_i,i}+w_{q_j,j} \leqslant w_r < k\ln m$ since r is good. So the joint probability for the provers to answer with a_i and a_j is at least $1/(|A_i|\cdot|A_j|) \geqslant (2/(|A_i|+|A_j|))^2 > 4/(k\ln m)^2$. The first inequality holds because xy is maximized by $x = y = z/2$ under the constraint that $x+y=z$. Since these answers are weakly consistent, i.e. they map to the same a_r, the verifier will accept weakly. Given that the fraction of good r is at least $2/k$ the verifier will accept with probability at least $8/(k^3\ln^2 m)$. ∎

To complete the proof of Lemma 10.6 we must show that the provers' random strategy succeeds with a probability that is greater than Lemma 10.3 allows, i.e. $8/(k^3\ln^2 m) > k^2\cdot 2^{-c_2 l}$. This is true if $8\cdot 2^{c_2 l} > k^5\ln^2 m$. Since Definition 10.4 requires $k < \frac{\ln m}{3\ln\ln m}$, we are on the safe side if $8\cdot 2^{c_2 l} > \frac{\ln^7 m}{(3\ln\ln m)^5}$, but even more so, if $2^{c_2 l} > \ln^7 m$. Setting $l = \frac{8}{c_2}\log\log n$, and $m = (5n)^{c_4 l}$, we receive $\log^8 n > \frac{8c_4}{c_2}\log\log n\cdot\ln^7 n$, which is certainly true for sufficiently large n. c_4 is a constant which will be fixed in the proof of the main theorem. ∎

Combining three results — the NP-hardness of ε-RobE3Sat-5, the complexity of constructing a partition system (Proposition 10.5), and the connection between the number of satisfiable clauses in a 3CNF-5 formula and the size of a cover with sets from such a partition system (Lemma 10.6) — we obtain the main theorem.

Theorem 10.9. *[Fei96]*
If there is some $\varepsilon > 0$ such that a polynomial time algorithm can approximate SetCover *within $(1-\varepsilon)\ln N$, then $\mathcal{NP} \subseteq \mathrm{DTIME}(N^{\mathcal{O}(\log\log N)})$.*

Proof. Assume that there is such an algorithm A. Reduce an instance of an arbitrary problem $X \in \mathcal{NP}$ in polynomial time to an instance of the NP-hard problem ε-RobE3Sat-5, i.e. to a 3CNF-5 formula φ with n variables which is either satisfiable or has at most a $(1-\varepsilon)$-fraction of clauses which are simultaneously satisfiable.

According to the proof of Lemma 10.6, let $l = \frac{8}{c_2} \log\log n$ and $m = (5n)^{c_4 l}$ with $c_4 = 2/\varepsilon$. As required by Definition 10.4, let $L = 2^l < (\ln m)^{c_3}$ for some constant c_3 and choose a $k < \frac{\ln m}{3 \ln\ln m}$. Make sure that $k \geqslant 8/\varepsilon$. In order to satisfy $\frac{\ln m}{3\ln\ln m} > 8/\varepsilon$, n must be sufficiently large, namely $\frac{2l \ln 5n}{(5n)^{l/12}} < \varepsilon$.

Now reduce φ with the help of partition systems $B[m, L, k]$ to a collection $\mathcal{C}$ of sets $S_{(q,a,i)}$ as above.

Recall that $R = (5n)^l = m^{\varepsilon/2}$ and $Q = n^{l/2} \cdot (5n/3)^{l/2}$. Therefore m, R, and Q are bounded by $n^{\mathcal{O}(\log\log n)}$ and the deterministic construction of partition systems [NSS95] is in DTIME$(n^{\mathcal{O}(\log\log n)}) \subseteq$ DTIME$(N^{\mathcal{O}(\log\log N)})$ where $N = mR = m^{1+\varepsilon/2}$.

Apply algorithm A to the instance of SetCover to which φ was reduced. If φ is satisfiable, then there is a cover of size kQ according to Lemma 10.6. Therefore A must find a cover $\mathcal{C}_A$ of size at most $kQ \cdot (1-\varepsilon)\ln N$ in the satisfiable case. With $\ln N = (1+\varepsilon/2)\ln m$ and $\varepsilon/2 \geqslant 4/k$ the size of A's solution can be upper-bounded as follows:

$$|\mathcal{C}_A| \leqslant kQ(1-\varepsilon)(1+\varepsilon/2)\ln m < kQ(1-\varepsilon/2)\ln m \leqslant kQ(1-4/k)\ln m$$

On the other hand Lemma 10.6 says that if at most a $(1-\varepsilon)$-fraction of the clauses in φ is simultaneously satisfiable, S requires more than $(k-4)Q\ln m$ subsets in order to be covered. Thus we could use A to decide deterministically an arbitrary NP-problem in time proportional to $N^{\mathcal{O}(\log\log N)}$. ■

Exercises

Exercise 10.1. Proof that the greedy algorithm presented in Section 10.1 always finds a cover $\mathcal{C}_{gr} \subseteq \mathcal{F}$ of the base set S such that

$$|\mathcal{C}_{gr}| \leqslant (1 + \frac{1}{2} + \ldots + \frac{1}{\Delta}) \cdot |\mathcal{C}_{opt}|$$

where $|\mathcal{C}_{opt}|$ is the size of a minimum cover and Δ the size of the largest subset of S in $\mathcal{F}$ ([Joh74], [Lov75]). Show that this ratio cannot be improved.

Hint: Group the subsets which the greedy algorithm selects into phases such that all subsets of one phase cover the same number of so far uncovered elements of the base set.

Exercise 10.2. Show that there is a $\delta \in (0, 1)$ such that ε-RobE3Sat-5, as defined in Section 10.1.2, is NP-hard for all $\varepsilon \leqslant \delta$.

Hint: Use that it is NP-hard to approximate Max3Sat-b (see Lemma 4.9 or [ALM$^+$92]). The number of occurences of each variable in a formula of type 3CNF-b is bounded from above by a universal constant b.