Algorithmic Construction of Sets for k-Restrictions (Extended Abstract)

Noga Alon [*] Dana Moshkovitz [†] Shmuel Safra [‡]

Abstract

This work addresses *k-restriction problems*, which unify combinatorial problems of the following type: One is given a set of demands on strings in Σ^m. Each demand depends only on k symbols out of the m. The task is to construct a small set of strings, so that for every k indices and every demand, some string satisfies the demand at this specific location. Problems of this form arise in many fields in Computer Science. A few prominent examples are group testing [17] and generalized hashing [3].

The standard approach for deterministically solving such problems is via almost k-wise independence or k-wise approximations for other distributions. We offer a generic algorithmic method that yields considerably smaller constructions. To this end, we generalize a previous work of Naor, Schulman and Srinivasan [16]. Among other results, we greatly enhance the combinatorial objects in the heart of their method, called *splitters*, using the topological *Necklace Splitting Theorem* [1]. To conclude, we show how this is used in an improved inapproximability result for Set-Cover under the assumption $\mathcal{P} \neq \mathcal{NP}$.

[*] `nogaa@tau.ac.il`. School of Computer Science, Tel-Aviv University, Tel-Aviv, 69978, Israel.
[†] `danamo@tau.ac.il`. School of Computer Science, Tel-Aviv University, Tel-Aviv, 69978, Israel.
[‡] `safra@tau.ac.il`. School of Computer Science, Tel-Aviv University, Tel-Aviv, 69978, Israel.

1 Introduction

k-restriction problems may be described illustratively as follows:

A goldsmith has to prepare an object (we will focus on strings) for a hot-tempered pirate. When the pirate comes back to collect the object, he claims he had posed some arbitrary demand on some small part of it, and threatens he would cut the goldsmith's head off, if such an object is not supplied to him immediately. The goldsmith - knowing the capricious, contradicting demands of the pirate in advance - wishes to prepare a small set of objects, such that no matter what is the a posteriori requirement of the pirate, some object satisfies it. Formally,

Definition 1 (k-restriction problems) *k-restriction problems are as follows:*

1. *The input is an alphabet Σ of size $|\Sigma| = q$, a length m and a set of s possible demands $f_i : \Sigma^k \to \{0,1\}$, $1 \le i \le s$. For every $1 \le i \le s$ there exists $a \in \Sigma^k$ so that $f_i(a) = 1$.*
2. *The task is to prepare a set $A \subseteq \Sigma^m$ so that: For any choice of k indices $1 \le i_1 < \cdots < i_k \le m$, and a demand j, $1 \le j \le s$, there is some $a \in A$, such that $f_j(a(i_1) \cdots a(i_k)) = 1$.*
3. *The quality of the solution is measured by how small $|A|$ is.*

Background. Clearly, $A = \Sigma^m$ is a solution for any k-restriction problem. But in fact, a simple calculation (see section 3) shows there exist solutions of considerably lower size, as a small random subset of Σ^m is a solution with high probability. As to deterministic constructions, it is well known that such may be obtained via almost k-wise independence, or k-wise approximations for other distributions (see section 4). Nevertheless, it turns out that they induce constructions that are considerably larger than the probabilistic ones (demonstrated in section 4). There is no wonder in this, as almost k-wise independence and similar constructs have stronger properties than what is really needed here. Instead of satisfying *any* k-demand with *noticeable* probability, the requirement is to satisfy *preset* k-demands with *positive* probability.

Our Work. We address the problem of deterministically finding solutions for k-restriction problems that are superior to k-wise approximations. For every constant k, we show an efficient algorithm that finds a solution whose size is arbitrarily close to the size of the probabilistic construction. For $k = O(\frac{\log m}{\log \log m})$, we show an efficient algorithm obtaining solutions of a slightly larger size. We also present an approach for handling larger values of k. Specifically, our results imply substantial improvements for many problems that may be formulated as k-restriction problems, such as group-testing [17] or generalized hashing [3]. They also allow us to improve the inapproximability result for SET-COVER under the assumption $\mathcal{P} \neq \mathcal{NP}$.

Previous Work. Naor, Schulman and Srinivasan [16] considered a few important problems that fall into the category of k-restriction problems, such as perfect hash families and universal sets. They solved all these problems using similar methods, focusing on decomposition of the problems into smaller problems. We take a *unifying* approach and address all k-restriction problems. Moreover, two ingredients are added. Naor *et al* considered problems in which a small set of strings chosen *uniformly* is (with high probability) a solution. We consider problems in which a small set of strings chosen from some (efficiently approximatable) distribution, not necessarily the uniform, is (with high probability) a solution. Many intriguing k-restriction problems are so (few examples appear in section 2). In addition, Naor *et al* used *splitters* to split a problem into sub-problems. We put *splitters* in a much wider context, viewing their construction as a special case of the topological Necklace Splitting Theorem [1]. This realization allows us to construct *multi-way splitters* that form much finer splits, and are, hence, suitable for a larger class of applications.

Perspective. In the past decades, randomness was established as a powerful tool in Computer Science. Following, many efforts were made to fully understand its necessity. Those efforts revealed that many applications of randomness may be, in fact, *derandomized*. The reason is that frequently one can settle for objects that only *look* random to a limited observer, that is, *randomness is (many times) in the eyes of the beholder*. This understanding motivated a series of works constructing small sample spaces for different limitations on a beholder: observing only k places (k-wise independence), considering only linear tests (ϵ-bias [15]), *etc.* This work takes this notion to the extreme, when given *any* limitation on the beholder in a fairly general setting, it provides algorithmically an appropriate set.

Organization. We start by motivating our work and reviewing a small set of its applications in section 2. Then we layout the foundations, when discussing the probabilistic construction in section 3 and k-wise approximations in section 4. In section 5 we describe our results simulating the probabilistic construction deterministically. The construction is done in three stages: greediness, approximation and concatenation. In section 6, we present a *Divide-and-Conquer* approach, that builds upon the first results and may derive efficient algorithms for a wider range of parameters. This approach is based on splitters [16] and extends them using the Necklace Splitting Theorem [1]. The Set-Cover application is deferred to section 7.

2 Applications

k-restriction problems emerge in a variety of contexts in computer science: the design of algorithm (e.g the *random orientations* of [5]), complexity, cryptography, learning, *etc.* In this section we give a small selection of *k-restriction* problems, and see the improvements our results imply for them. A more complicated example appears in section 7, where an improved inapproximability result for Set-Cover is described.

Group Testing. The classic problem of group testing [10], which today has numerous applications in computational biology, was first presented during world war II. Among a large population of m soldiers, a fixed number d of individuals carry a fatal virus. We would like to blood test the soldiers to detect all carriers. The simple solution would be to test the blood of each soldier separately. But if we are to minimize the number of tests, we can mix the blood of several soldiers and test the mix. If the test comes out negative, then all tested soldiers are not carriers. If the test comes out positive, we know at least one of them is a carrier.

Let us formulate the problem as a *k-restriction* problem. A test may be represented by a string in $\{0,1\}^m$, indicating which of the m blood samples to take and which not. The combinatorial requirement is that no matter who are the d carriers, for every non-carrier, there should be a test that examines his blood, and none of the d infected blood samples. In other words, for every string $w \in \Sigma^k$, $k = d+1$, with d zeros and a single one, there should be a demand $f_w : \Sigma^k \to \{0,1\}$, $f_w(x) = 1$ *iff* $x = w$. Our results demonstrate the following:

Theorem 1 *For any fixed d, for every $\delta > 0$, a set of at most $(1+\delta)ed(d+1)\ln m$ group tests may be found in time polynomial in m, the size of the population.*

To the best of our knowledge, there are no known deterministic constructions of this quality (for a survey on existing results see [17]).

Perfect Hashing and Its Variants. The classic problem of $(m, q, k)-$*perfect hashing* may be described as follows: There are m possible users that should be assigned q identical resources, $q \ll m$. No two users may use the same resource simultaneously. Luckily, at any given time there are only $k \leq q$ users requesting a resource. The task is to find a small family of assignments of the m users to the q resources, so that no matter which k users are active, some assignment assigns each of them a unique resource. This is a *k-restriction* problem where Σ is the set of resources, assignments of resources to the m users are represented as vectors in Σ^m, and there is a single demand that checks uniqueness $f : \Sigma^k \to \{0,1\}$, $f(x) = 1$ *iff* x comprises k different symbols. This problem is very well known and has many applications by itself, but moreover, most of its variants are also *k-restriction* problems.

One example is (t, u)-*hash families* used for the design of parent identifying codes for digital fingerprinting [3]. For every $T \subseteq [u]$, $|T| = t$, there is a demand $f_T : \Sigma^u \to \{0,1\}$, $f_T(x) = 1$ *iff* for every $i \in T$ and $j \in [u] \setminus \{i\}$, $x(i) \neq x(j)$. [3] addressed the case of the minimal alphabet-size $q = t+1$, and concentrated on the *rate* $R = \frac{\log_q m}{n}$, where n is the size of the construction. Our results derive a better explicit construction than theirs:

Theorem 2 *For any fixed $2 \leq t < u$, for any $\delta > 0$ one can construct efficiently a (t, u)-hash family over an alphabet of size $t+1$ whose rate is at least $(1-\delta)\frac{t!(u-t)^{u-t}}{u^{u+1}\ln(t+1)}$.*

This rate resembles that of the probabilistic construction of [3], $\frac{t!(u-t)^{u-t}}{u^u(u-1)ln(t+1)} - \delta$, and improves their explicit construction whose rate is u^{-ct} for a large constant c. Importantly, our methods and the use of the *multi-way splitters* we define, provide small constructions even when u and t grow with the parameters of the problem. More details will appear in the full version.

3 Probabilistic Solution

In this section we study the quality of a random solution. For this purpose, we define the *density* ε of a *k-restriction* problem with respect to some probability distribution D over Σ^m, so for any given demand at any given position, when picking a random string from D it satisfies the demand at that position with probability at least ε.

Definition 2 (density) *Given a probability distribution, $D\colon \Sigma^m \to [0,1]$, the* density *of a k-restriction problem with respect to D is*

$$\varepsilon \doteq \min_{\substack{1 \leq i_1 < \cdots < i_k \leq m \\ 1 \leq j \leq s}} \left\{ \Pr_{a \sim D}\left[f_j(a(i_1)\cdots a(i_k)) = 1\right] \right\}$$

Note that every *k-restriction* problem has density at least q^{-k} with respect to the uniform distribution. Some problems (e.g group testing or (t, u)-hashing) have a much larger density with respect to other distributions.

A simple union bound calculation shows that if a problem has a large density (with respect to any distribution), it has a small solution:

Proposition 1 (union bound) *Every k-restriction problem with density ε with respect to some probability distribution has a solution of size at most $\frac{k \ln m + \ln s + 1}{\varepsilon}$.*

Moreover, a random construction of size of this order is – with high probability – a solution. But note that we do not know of an efficient way to verify this is truly the case, unless $k = \Theta(1)$.

4 k-wise Approximating Distributions

The purpose of this section is to present the folklore solution for *k-restriction* problems and the appropriate notions. We start by defining ways to measure distance between distributions.

Definition 3 (l_p-norm distance) *The* distance in the l_p-norm *between two probability distributions D, P over some sample space Ω is $\|D - P\|_p \doteq \left(\sum_{a\in\Omega} |D(a) - P(a)|^p\right)^{\frac{1}{p}}$. The distance in max-norm is $\|D - P\|_\infty \doteq \max_{a\in\Omega} |D(a) - P(a)|$.*

We consider distributions over Σ^m, and interested in k-wise approximations, i.e distributions that approximate a distribution well when *restricted* to any k coordinates. The *restriction* of a probability distribution D over Σ^m to indices $1 \le i_1 < \cdots < i_k \le m$ is a probability distribution $D_{i_1,\ldots,i_k}$ over Σ^k, that assigns each $a \in \Sigma^k$ the probability a string drawn randomly from D has a in indices $i_1, \ldots, i_k$, i.e, $D_{i_1,\ldots,i_k}(a) \doteq \Pr_{X\sim D}\left[X(i_1) = a(1) \wedge \cdots \wedge X(i_k) = a(k)\right]$.

Definition 4 (k-wise ε-closeness) *Two distributions D, P over Σ^m are said to be* k-wise ε-close in the l_p-norm, *if for any $1 \le i_1 < \cdots < i_k \le m$, $\|D_{i_1,\ldots,i_k} - P_{i_1,\ldots,i_k}\|_p \le \varepsilon$.*

A distribution which is k-wise 0-close (no matter in what norm!) to the uniform distribution is called *k-wise independent*, and hence the use of the term *almost k-wise independence*.

Many works present *explicit* constructions of *succinct* distributions that are k-wise ε-close to different distributions in different norms. *Succinctness* refers to the size of their *support*, i.e the number of strings having a positive probability. We will use the following definition:

Definition 5 (k-wise approximation) *A distribution D over Σ^m is said to be* k-wise efficiently approximatable, *if the support of a distribution which is k-wise ε-close to it in the l_1-norm may be enumerated in time polynomial in m, ϵ^{-1} and $|\Sigma|^k$.*

Perhaps the largest family of distributions that admit an efficient k-wise approximation is that of *product distributions*. A *product distribution* over Σ^m , $|\Sigma| = q$, is specified by a matrix $\mathcal{P}^{m\times q}$, indicating the probability each of the m independent coordinates assumes each of the values in Σ. Note that the uniform distribution is a product distribution. We quote:

Lemma 1 *[11] Any product distribution on Σ^m is k-wise efficiently approximatable.*

Using the former definition we can now describe the folklore method for solving *k-restriction* problems deterministically. The underlying observation is the following:

Proposition 2 (folklore) *The support of a distribution over Σ^m which is k-wise ε-close in l_∞ (and hence in any finite norm as well) to D is a solution for any* k-restriction *problem with parameters m ,Σ, k, s and density ε with respect to D.*

The reason this holds is that no subset of weight ε with respect to some k restriction of D might vanish in such a good approximation of it.

Unfortunately, this yields solutions with a dependence on parameters that is inferior to that of proposition 1. To demonstrate this, consider the extensively studied uniform distribution over $\{0,1\}^m$. Known explicit constructions suggest distributions that are almost k-wise independent in l_∞ of support's size $O(\frac{k^2 \log^2 m}{\varepsilon^2})$ [4] or $O(\frac{k \log m}{\varepsilon^3})$ [15, 2]. The dependence is usually much worse for other distributions (e.g, see [11]).

5 Our Results

Before we turn to discuss efficient deterministic solutions, let us make one technical definition. We will sometimes consider sets of demands that are *invariant under permutations*, i.e when permutating the substrings, the possible demands remain the same. Formally, if for a demand f, we use C_f to indicate $f_i^{-1}(1)$, the set of substrings it accepts, then

Definition 6 (invariance under permutations) *A set of demands* $\{f_1, \ldots, f_s\}$ *is* invariant under permutations, *if for any permutation* $\sigma\colon [k] \to [k]$, $\{C_{f_1}, \ldots, C_{f_s}\} = \{\sigma(C_{f_1}), \ldots, \sigma(C_{f_s})\}$.

Natural problems usually have demands with this property, e.g. the problems of section 2. Moreover, problems may be added demands so they satisfy this property with only a slight change to the results.

We show algorithms providing solutions of size similar to that of proposition 1. For $k = \Theta(1)$,

Theorem 3 *Fix some efficiently approximatable probability distribution D. Given a k-restriction problem with density ε with respect to D, one can obtain a solution of size arbitrarily close to $\frac{k \ln m + \ln s}{\varepsilon}$ in time polynomial in s, m^k, q^k and ε^{-1}.*

For $k = O(\frac{\log m}{\log \log m})$,

Theorem 4 *Fix some efficiently approximatable probability distribution D. Given a k-restriction problem with density ε with respect to D and demands that are invariant under permutations, one can obtain a solution of size arbitrarily close to $\frac{2k \ln k + \ln s}{\varepsilon} \times \left|Hash_{m,k^2,k}\right|$ in time $poly(m, s, k^k, q^k, \varepsilon^{-1})$.*

When $|Hash_{m,q,k}|$ is the size of the smallest possible construction of a (m, q, k)-perfect hash family (see section 2). Note that by [2] and the connection between perfect hashing and asymptotically optimal error correcting codes (cf. [3]), $\left|Hash_{m,k^2,k}\right| \leq O(k^4 \log m)$.

For larger values of k we present in section 6, an approach that builds upon the above theorems. To prove them we use three techniques. First, we formulate *k-restriction* problems as (huge) Set-Cover problems. This derives an algorithm with two flaws: The running-time is polynomial in m^k, which is inefficient for non-constant k, and it requires an enumeration of the distribution's support (sub-section 5.1). Then we k-wise approximate the distribution to get one that preserves most of the density, and has support that may be enumerated (sub-section 5.2), thus obtaining theorem 3. Lastly, we show a technique transforming inefficient solutions, such as the one obtained when combining the first two techniques, into efficient ones, by way of concatenation (sub-section 5.3). This results in theorem 4.

5.1 Greediness

Somewhat surprisingly – bearing in mind our SET-COVER application – the first step of the construction involves finding a small solution for some SET-COVER instance (see section 7 for background). We show that any *k-restriction* problem may be formulated as a SET-COVER problem, and invoking the greedy algorithm ensures the size anticipated by probabilistic considerations. This approach is really an implementation of the *method of conditional expectations.*

Unfortunately, the reduction to SET-COVER is not polynomial in m for $k = \omega(1)$. Hence, in this case, the algorithm we present is inefficient by itself. Nonetheless, it serves as a procedure inside the final algorithm, when invoked with much smaller parameters.

First, let us adhere to the SET-COVER setting, and analyze the behavior of the greedy algorithm when all elements are covered by a significant portion of the subsets.

Proposition 3 *Let $(\mathbb{U}, \mathbb{S})$ be a* SET-COVER *instance. Let $D\colon \mathbb{S} \to [0,1]$ be some probability distribution over $\mathbb{S}$. Let $U' \subseteq \mathbb{U}$. If for every $u \in \mathbb{U}$, $\Pr_{S\sim D}[u \in S] \geq \varepsilon$, then the greedy algorithm produces a set cover of size at most $\frac{\ln|\mathbb{U}|}{\varepsilon}$.*

Proof: For a subset $S \in \mathbb{S}$ and an element $u \in \mathbb{U}$, let $X_{u,S}$ be an indicator variable for the event $'u \in S'$. By linearity of expectations,

$$\mathop{\mathbf{E}}_{S\sim D}\left[|S \cap U'|\right] = \mathop{\mathbf{E}}_{S\sim D}\left[\sum_{u\in U'} X_{u,S}\right] = \sum_{u\in U'} \mathop{\mathbf{E}}_{S\sim D}[X_{u,S}] = \sum_{u\in U'} \Pr_{S\sim D}[u \in S] \geq \varepsilon\,|U'|$$

Hence, in each iteration the greedy algorithm covers at least ε of the sub-universe. A simple calculation shows $\frac{\ln|\mathbb{U}|}{\varepsilon}$ iterations suffice. ∎

The following lemma explains how k-restriction problems reduce to SET-COVER problems and uses the above analysis to derive the algorithm.

Lemma 2 (greediness) *Given the support of some probability distribution D and a* k-restriction *problem with density ε with respect to D, one can obtain a solution of size at most $\frac{k\ln m + \ln s}{\varepsilon}$ in time polynomial in $\binom{m}{k}$, s and $|supp(D)|$.*

Proof: Define an element for every possible position/demand, and a set for every support vector:

$$\mathbb{U} \doteq \{< i_1, \ldots, i_k, j > \mid 1 \leq i_1 < \cdots < i_k \leq m, 1 \leq j \leq s\}$$

$$\mathbb{S} \doteq \{\, S_a \mid a \in supp(D)\} \text{ where } S_a \doteq \{< i_1, \ldots, i_k, j >\in \mathbb{U} \mid f_j(a(i_1) \cdots a(i_k)) = 1\}$$

Note that a cover $\mathcal{C}$ for $(\mathbb{U}, \mathbb{S})$ corresponds to a solution of size $|\mathcal{C}|$. As for any $u \in \mathbb{U}$, $\Pr_{a\sim D}[u \in S_a] \geq \varepsilon$, the lemma follows from proposition 3. ∎

5.2 Approximating Distributions

It can be easily verified that a distribution D' which k-wise approximates a distribution D preserves most of the density with respect to k-restrictions.

Lemma 3 (k-wise approximation) *Let D, D' be two probability distributions over Σ^m, such that D' is k-wise ε-close to D in the l_1-norm. Then for every demand $f : \Sigma^k \to \{0,1\}$ and every k indices $1 \leq i_1 < \cdots < i_k \leq m$,*

$$\left|\Pr_{X\sim D}[f(X(i_1)\cdots X(i_k)) = 1] - \Pr_{X\sim D'}[f(X(i_1)\cdots X(i_k)) = 1]\right| \leq \varepsilon$$

Combining the greedy algorithm and this lemma, we can prove theorem 3:

Proof:[of theorem 3] First construct a distribution D' which is k-wise $\delta\varepsilon$-close in the l_1-norm to the distribution at hand, where $\delta > 0$ can be made arbitrarily small. By lemma 3, for any k indices $1 \le i_1 < \cdots < i_k \le m$ and any k-demand $f : \Sigma^k \to \{0,1\}$,

$$\Pr_{X\sim D'}[f(X(i_1)\cdots X(i_k)) = 1] \ge \Pr_{X\sim D}[f(X(i_1)\cdots X(i_k)) = 1] - \delta\varepsilon \ge \varepsilon - \delta\varepsilon = (1-\delta)\,\varepsilon$$

The corollary follows from lemma 2 with D' instead of the original distribution. ∎

5.3 Concatenation

In this sub-section we show a general scheme for transforming inefficient algorithms into more efficient algorithms via concatenation. We invoke the inefficient algorithm only on $m' \ll m$ tuples. In the case of the greedy algorithm, this reduces the $\binom{m}{k}$ factor in the running-time to the considerably smaller $\binom{m'}{k}$ factor.

It is useful to think of the output of the inefficient algorithm as a table with m' columns. Our plan is to interleave these columns in a clever way, so they compose a solution for the original problem on m tuples. This clever way is defined by an outer construction, that should be thought of as a table with m columns having entries from $[m']$. In this way, each entry references a column in the inner construction. The concatenated construction replaces each reference with the actual column. The property of the outer construction is *perfect hashing* (see section 2): for every choice of k columns out of m, there should be a row in which all k entries differ. That is, in the final construction we can find there k different columns of the inner construction. By the property of the inner construction and the invariance under permutations, they satisfy all demands. Now we can describe the concatenation algorithm for some *k-restriction* problem with demands that are invariant under permutations,

- **Input:** Solution A for the problem on m'-tuples, and (m, m', k)-perfect hash family B.
- **Output:** Solution C for the problem on m-tuples (instead of m'-tuples).
- **Process:** For every $a \in A$, and every $b \in B$, output the vector $a(b(1))\cdots a(b(m))$.

The correctness of this algorithm may be easily checked:

Lemma 4 (concatenation) *Given A and B as specified, the concatenation algorithm produces a solution C for the problem with $|C| = |A|\cdot|B|$.*

Recalling that there are small explicit constructions of (m, k^2, k)-perfect hash families (e.g of size $O(k^4 \log m)$ [2]), theorem 4 follows.

6 Divide and Conquer

We outline a Divide and Conquer approach, that may be used together with theorem 4 to obtain an efficient strategy for many *k-restriction* problems, even for $k \ge \omega(\log m/\log\log m)$:

1. Give a short list of possible divisions of $\{1,\ldots,m\}$ into b blocks, typically $b = \Theta(\log k)$.
2. Find a solution for the sub-problem with parameters $m, k/b$ (e.g using theorem 4).
3. Combine the sub-solutions.

For the purpose of dividing into blocks, [16] defined a combinatorial entity called a *splitter*. We generalize their definition, based on the topological Necklace Splitting Theorem of [1].

6.1 Necklace Splitting

The Necklace Splitting Theorem states, that if b thieves steal a necklace with beads of t types, they can cut it in only $(b-1)t$ places, in order to fairly divide it between them. Here 'fair' means that each thief gets the same number of beads of each type.

A necklace is a list of bn items, each has a type $1 \leq i \leq t$. Suppose there are ba_i, $a_i \in \mathbb{N}$, items of type $1 \leq i \leq t$. A b-splitting of a necklace is a partition of it into b parts, each consisting of a finite number of non-overlapping sub-lists of items whose union captures precisely a_i items of type i, $1 \leq i \leq t$. The size of the partition is the number of cuts that form the sub-lists.

If the items of each type appear contiguously in the list, then any b-splitting must be of size at least $(b-1)t$. The Necklace Splitting Theorem uses topological tools to show this number of cuts always suffices to produce a b-splitting. For $t = 1$, it is trivial. Let us describe the intuition behind the case of $t = b = 2$, which provides some insight to the role of topology in the proof. Call one of the types *red*. Observe a continuous analog of the necklace closed at its clasp. Consider some half necklace. If the measure of red within this half is exactly $\frac{1}{2}$, this induces a fair division, and we are done. Otherwise, assume without loss of generality that this measure is larger than $\frac{1}{2}$. When rotating the necklace $180°$, the measure of red in the observed half is hence smaller than $\frac{1}{2}$. As the change in the measure is continuous, there must be a half in which this measure is exactly $\frac{1}{2}$. In general, the proof uses a generalization of the Borsuk-Ulam theorem.

Lemma 5 (The Necklace Splitting Theorem,[1]) *Every necklace with ba_i beads of type i, $1 \leq i \leq t$, has a b-splitting of size at most $(b-1)t$.*

Note that the proof of this theorem is not constructive. Nevertheless, this is not a drawback in our context, as we invoke it on very small parameters, and consider all possible splits.

6.2 Multi-Way Splitters

A partition of m coordinates into b blocks is a function $\pi\colon [m] \to [b]$ assigning each coordinate the block that contains it. We say π splits a subset $I \subseteq [m]$ of the coordinates, if every block $1 \leq j \leq b$ sees the same number of coordinates up to rounding, i.e $\left\lfloor \frac{|I|}{b} \right\rfloor \leq \left|\pi^{-1}(j) \cap I\right| \leq \left\lceil \frac{|I|}{b} \right\rceil$.

[16] introduced the notion of (m, k, b)-splitters, referring to list of partitions such that every set of k coordinates is split. To use the analogy of the Necklace Splitting Theorem, say the necklace is composed of m seemingly identical jewels, but only k of them are real gemstones. [16] wanted a small set of possible partitions of the necklace to b thieves, so that no matter which k jewels are real, there is a partition that splits these k jewels fairly between the thieves. We introduce a generalization of this notion, namely *multi-way* splitters. The only difference is that now jewels may be of various types, and a fair partition is one that gives each thief the same number of jewels of each type.

Definition 7 (multi-way splitter) *Let m, b be natural parameters.*

- *A* t-way splitter *is a list A of partitions $\pi\colon [m] \to [b]$, s.t. for any series of t pairwise disjoint subsets of coordinates $S_1, \ldots, S_t \subseteq \{1, \ldots, m\}$, there exists a partition in A that splits each one of $S_1, \ldots, S_t$.*

- *A* k-wise t-way splitter *is a list A of partitions $\pi\colon [m] \to [b]$, s.t. for any series of t pairwise disjoint subsets of at most k coordinates $S_1, \ldots, S_t \subseteq \{1, \ldots, m\}$, $\sum |S_i| \leq k$, there exists a partition in A that splits each one of $S_1, \ldots, S_t$.*

When there is only one type, i.e $t = 1$, our definition is equivalent to the [16] definition. When $t > 1$, we need to use the Necklace Splitting Theorem.

Lemma 6 *Let m, b be natural parameters. There exists an explicit construction of a t-way splitter that splits m indices into b blocks, and has size $b^{(b-1)t+1} \cdot \binom{m}{(b-1)t}$.*

Proof: For every $0 = i_0 < i_1 < \cdots < i_{(b-1)t} \leq i_{(b-1)t+1} = m$, and every $s \in [b]^{(b-1)t+1}$, define a partition π as follows: $\pi(j) = s_k \leftrightarrow i_{k-1} < j \leq i_k$. The lemma follows from [1]. ■

We are interested in constructions of (smaller) k-wise t-way splitters. This is a *k-restriction* problem over alphabet of size b with demands that are invariant under permutations; There is a demand for every classification of k coordinates into t types, containing all partitions that split all types. Now, applying concatenation,

Corollary 1 *A k-wise t-way splitter for splitting m coordinates into b blocks of size $\left|Hash_{m,k^2,k}\right| \cdot b^{(b-1)t+1} \cdot \binom{k^2}{(b-1)t}$ may be constructed in time polynomial in m, t^k and the size of the construction.*

Proof: Apply lemma 4 (concatenation) on lemma 6 (the exhaustive construction). ■

7 Set-Cover

Given a universe $\mathbb{U} = \{u_1, ..., u_n\}$ and a family of its subsets, $\mathbb{S} = \{S_1, ..., S_m\} \subseteq P(\mathbb{U})$, $\bigcup_{S_j \in \mathbb{S}} S_j = \mathbb{U}$, Set-Cover is the problem of finding a minimal sub-family $\mathbb{C} \subseteq \mathbb{S}$ that covers the whole universe, $\bigcup_{S_j \in \mathbb{C}} S_j = \mathbb{U}$. Set-Cover is a classic $\mathcal{NP}$-hard combinatorial optimization problem, and it is known it can be approximated to within $\ln n - \ln\ln n + \Theta(1)$[19, 13, 20].

Feige proved that for every $\varepsilon > 0$, there is no efficient approximation to within $(1-\varepsilon)\ln n$, under the strong assumption $\mathcal{NP} \not\subseteq DTIME(n^{\log\log n})$ [12]. Weaker results were known for strictly polynomial reductions. The best result of this nature was derived from the sub-constant error PCP of Raz and Safra [18]. By [7] and the derandomization of [16], their work implied that unless $\mathcal{P} = \mathcal{NP}$, there exists a constant $0 < c < 1$ so that Set-Cover cannot be efficiently approximated to within any number smaller than $c \log_2 n$. This constant is $c = \frac{1}{2(d+1)}$, where the natural $d \geq 2$ is a certain parameter of the underlying PCP, called its *dependency*, or *the number of provers*. Due to the complication of the PCP constructions, the exact value of d is vague in many results, but, certainly, to this date, there are *no* known PCP constructions that admit both a poly-logarithmic small error probability and d which is smaller than 5.

Up to this work, the inverse linear dependence of the Set-Cover inapproximability factor in d was considered inherent to the reduction of [7], which is the only existing reduction facilitating $\mathcal{NP}$-hardness results for Set-Cover. We manage to reduce it to the considerably smaller inverse dependency in $\ln d$, which improves the result no matter how small d is made, and certainly for the values of d known today.

Theorem 5 (Set-Cover Hardness) *For any $\eta \geq 1$, Set-Cover cannot be efficiently approximated to within any number smaller than $c \ln n$, for $c = \frac{1}{(1+1/\eta)(\ln \eta d + 1 + 1/\eta d)}$, unless $\mathcal{P} = \mathcal{NP}$.*

7.1 How Do We Improve?

In the heart of all existing Set-Cover reductions lie consistency gadgets. Those gadgets are Set-Cover instances that covering them in a *legal* way is *cheap*, while covering them in an *illegal* way is *very expensive*. It was noted in the past [14] that (m, k)-*universal sets* result in gadgets of this type. Universal sets are sets of binary strings, so that every configuration of length k appears at any given location in some string. In other words, this is a k-restriction problem with $\Sigma = \{0,1\}$, and a demand f_w for every $w \in \{0,1\}^k$ accepting only it, i.e $f_w(x) = 1$ *iff* $x = w$. The reason that universal sets induce appropriate gadgets is that every string $a \in \{0,1\}^m$ in the set may be interpreted as an element in a universe with $2m$ subsets: for every $1 \le i \le m$, if $a(i) = 1$, the corresponding element is in the *even subset* $2i$. Otherwise it is in the *odd subset* $2i - 1$. Taking any two subsets $2i, 2i-1$ covers the entire universe, while it takes more than k subsets without taking such two, since smaller covers must miss some element.

After carefully examining the current construction, we found an asymmetry in the composition of these gadgets that was overlooked in the past, but may be exploited: subsets of one type (say, the *even subsets*) are worthier for an adversary wishing to cover the composed construction, than subsets of the other type (the *odd subsets*). Hence a more adequate gadget is designed by a slight modification of universal sets. Instead of looking for every $\{0,1\}^k$ configuration, we would look for every *biased* configuration, i.e let 0 and 1 have different weights. This will allow us to handle longer configurations, without enlarging the size of the set considerably.

7.2 The New Consistency Gadget

Let us describe the new consistency gadget formally. Fix some parameter $0 < p < 1$ (as to indicate the advantage an even subset has over an odd subset). Set $w_0 = -\ln p$ and $w_1 = p(1+p)$. Given a threshold l (which will dominate the inapproximability factor obtained), our requirement is that every string t over binary alphabet with $w_0 \cdot \sharp_0(t) + w_1 \cdot \sharp_1(t) \le l$ (where $\sharp_c(t)$ is the number of occurrences of c in t) appears in every possible location in one of the strings. This is clearly a *k-restriction* problem for $k \le l/p$ with demands that are invariant under permutations (we omit the details). Moreover, the weights were chosen so its density with respect to a product distribution over $\{0,1\}^m$ assuming 1 in each coordinate with probability p, is at least e^{-l}. Thus, theorem 4 provides an efficient algorithmic construction for $l = O(\log m / \log\log m)$. We may use the Divide and Conquer approach proposed in section 6 to find efficiently a construction for l which is logarithmic in m. The idea is that a k-wise 2-way splitter preserves the biased structure of the demands in the sub-problems.

To make this formal, given a partition $\pi : [m] \to [b]$, let us define $\bar{\pi} \colon [m] \to [b] \times [m]$ assigning a coordinate its (block,ordinal within the block) pair, i.e $\bar{\pi}(i) = (\pi(i), |\{j \le i \mid \pi(j) = \pi(i)\}|)$.
Algorithm *Divide-and-Conquer*
Input: parameters $l \le m$.

1. Construct a k-wise 2-way splitter that splits m coordinates into $b \doteq \lceil \log l \rceil$ blocks.
2. Find a solution R for the re-scaled problem, with parameters m and $\lceil \frac{l}{b} \rceil$.
3. For every splitter partition π, for every b strings, $r_1, \ldots, r_b \in R$, output a string x composed of $r_1, \ldots, r_b$ interleaved according to π: for $1 \le i \le m$, $\bar{\pi}(i) = (k,j) \to x(i) = r_k(j)$.

The following lemma follows from the correctness of the algorithm. The proof is deferred to the full version. Its use in the improved Set-Cover hardness proof is detailed in appendix A.

Lemma 7 (Set-Cover gadget) *Fix some p. Given parameters m and l a consistency gadget for* Set-Cover *of the predefined property, whose universe is of size $e^l l^{O(\log l)} \log^{O(\log l)} m$, may be constructed in time polynomial in m and this size.*

Acknowledgements

We would like to thank Ran Raz for help with the adjustment of the [7] reduction and to Tal Moran for proofreading and discussions.

References

[1] N. Alon. Splitting necklaces. *Advances in Mathematics*, 63(3):247–253, 1987.

[2] N. Alon, J. Bruck, J. Naor, M. Naor, and R. Roth. Construction of asymptotically good, low-rate error-correcting codes through pseudo-random graphs. *IEEE Transactions on Information Theory*, 38:509–516, 1992.

[3] N. Alon, G. Cohen, M. Krivelevich, and S. Litsyn. Generalized hashing and parent-identifying codes. *J. Combinatorial Theory, Ser. A.* to appear.

[4] N. Alon, O. Goldreich, J. Håstad, and R. Peralta. Simple constructions of almost k-wise independent random variables. In *Proc. 31st IEEE Symp. on Foundations of Computer Science*, volume 2, pages 544–553, 1990.

[5] N. Alon, R. Yuster, and U. Zwick. Color-coding. *Electronic Colloquium on Computational Complexity (ECCC)*, 1(009), 1994. Full paper appears in J.ACM 42:4, July 1995, 844-856.

[6] S. Arora and S. Safra. Probabilistic checking of proofs: a new characterization of NP. *Journal of the ACM*, 45(1):70–122, 1998.

[7] M. Bellare, S. Goldwasser, C. Lund, and A. Russell. Efficient multi-prover interactive proofs with applications to approximation problems. In *Proc. 25th ACM Symp. on Theory of Computing*, pages 113–131, 1993.

[8] S. A. Cook. The complexity of theorem-proving procedures. In *Proc. 3rd ACM Symp. on Theory of Computing*, pages 151–158, 1971.

[9] I. Dinur, E. Fischer, G. Kindler, R. Raz, and S. Safra. PCP characterizations of NP: Towards a polynomially-small error-probability. In *Proc. 31st ACM Symp. on Theory of Computing*, pages 29–40, 1999.

[10] D. Z. Du and F. K. Hwang. *Combinatorial Group Testing and its Applications*, volume 12 of *Series on Applied Mathematics*. World Scientific, New York, second ed. edition, 2000.

[11] G. Even, O. Goldreich, M. Luby, N. Nisan, and B. Velickovic. Efficient approximation of product distributions. *Random Structures and Algorithms*, 13(1):1–16, 1998.

[12] U. Feige. A threshold of $\ln n$ for approximating set cover. In *Proc. 28th ACM Symp. on Theory of Computing*, pages 314–318, 1996.

[13] L. Lovász. On the ratio of optimal integral and fractional covers. *Discrete Mathematics*, 13:383–390, 1975.

[14] C. Lund and M. Yannakakis. On the hardness of approximating minimization problems. *Journal of the ACM*, 41(5):960–981, 1994.

[15] J. Naor and M. Naor. Small-bias probability spaces: Efficient constructions and applications. *SIAM J. Comput*, 22(4):838–856, 1993.

[16] M. Naor, L. J. Schulman, and A. Srinivasan. Splitters and near-optimal derandomization. In *Proc. 36th IEEE Symp. on Foundations of Computer Science*, pages 182–191, 1995.

[17] H. Q. Ngo and D. Z. Du. *Discrete mathematical problems with medical applications*, volume 55 of *DIMACS Ser. Discrete Math. Theoret. Comput. Sci.*, chapter A Survey on Combinatorial Group Testing Algorithms with Applications to DNA Library Screening, pages 171–182. Amer. Math. Soc., 2000.

[18] R. Raz and S. Safra. A sub-constant error-probability PCP characterization of NP. In *Proc. 29th ACM Symp. on Theory of Computing*, pages 475–484, 1997.

[19] P. Slavik. Improved approximations of packing and covering problems. In *Proc. 27th ACM Symp. on Theory of Computing*, pages 268–276, 1995.

[20] A. Srinivasan. Improved approximations guarantees for packing and covering integer programs. *SIAM J. Comput.*, 29(2):648–670, 1999. Preliminary version in Proc. STOC 95.

A Improved Hardness Result for Set-Cover

In this section we prove the improved hardness result for SET-COVER, under the assumption $\mathcal{P} \neq \mathcal{NP}$. Our reduction generalizes the reduction of [7], and may be used with any PCP with poly-logarithmic small error-probability, which has some constant dependency d.

A.1 Preliminaries - Probabilistically Checkable Proofs

According to the traditional definition [8], $\mathcal{NP}$ is the class of all languages having a polynomially checkable membership proof. Rather surprisingly, $\mathcal{NP}$ may be equivalently characterized as the class of all languages having a *Probabilistically Checkable Proof* (PCP) [6]. Such proofs may be checked, with high probability of being correct, by inspecting only a constant number of their components, chosen at random. A PCP characterization of $\mathcal{NP}$ is derived by showing some sort of *PCP problem* – described shortly – is $\mathcal{NP}$-hard.

A PCP instance is a triplet $(X, \mathbb{F}, \Phi)$. $X = \{x_1, ..., x_n\}$ is a set of formal variables that range over a domain $\mathbb{F}$. $\Phi = \{\varphi_1, ..., \varphi_m\}$ is a collection of *local-tests*, $m = poly(n)$. Each local test is a Boolean function on $d = O(1)$ variables. Φ is *uniform*, i.e each variable appears in the same number of tests. We consider multi-assignments $\hat{A}\colon X \to P(\mathbb{F})$, assigning each variable a set of values from $\mathbb{F}$. We say $\hat{A}$ satisfies a test φ over variables $x_{i_1}, \ldots, x_{i_d}$, if there exist $a_1 \in \hat{A}(x_{i_1}), \ldots, a_d \in \hat{A}(x_{i_d})$ such that $\varphi(a_1, \ldots, a_d) = true$. A multi-assignment $\hat{A}$ is said to be (t, γ)-*good*, if there exists a considerable fraction of tests $\Phi' \subseteq \Phi$, $|\Phi'| \geq \gamma |\Phi|$, so that for every $\varphi \in \Phi'$, $\hat{A}$ satisfies φ and the variables φ depends on, $x_{i_1}, \ldots, x_{i_d}$, have few assignments, $0 < |\hat{A}(x_{i_j})| \leq t$. The following follows from [9] using a simple probabilistic argument:

Lemma 8 *[9] For any $0 < \beta, \gamma < 1$, given a PCP instance $(X, \mathbb{F}, \Phi)$ with $|\mathbb{F}| \leq 2^{log^{1-\beta} n}$, it is $\mathcal{NP}$-hard to distinguish between the case there exists a multi-assignment which is $(1,1)$-good, and the case every multi-assignment is* not *(t, γ)-good, as long as $\gamma > 2t^d / |\mathbb{F}|$.*

A.2 The Construction

Fix a PCP instance $(X, \mathbb{F}, \Phi)$ as in lemma 8, and let us construct a corresponding Set-Cover instance $(\mathbb{U}, \mathbb{S})$. In a top-down overview, our construction is composed of some number, D, of duplicates. Each duplicate has a subset for every variable and an assignment to it, as well as a sub-universe for every test. The subset covers some portion of every sub-universe associated with a test observing the subset's variable. For each test and a satisfying assignment for it, an additional subset covers a portion of the test's sub-universe in each duplicate. The appropriate portion is the complement of the union of portions covered by subsets associated with compatible assignments to the test's variables. This makes sure a $(1,1)$-good multi-assignment corresponds to a small cover. The setting of the portions, determined by the consistency gadget, promises that a moderately large cover corresponds to a rather good multi-assignment. The duplications mechanism makes sure the number of parts each subset covers is of the same order, $\Theta(|\Phi| / |X|)$.

Fix an arbitrary $\eta \geq 1$ and let $D = \left\lfloor \frac{|\Phi|}{\eta|X|} \right\rfloor$. Set $m = d \cdot |\mathbb{F}|$ and $p = 1/\eta d$. Denote the Set-Cover instance of lemma 7 above for p, m and l, which will be defined at will, by $(B, \{C_1, ..., C_m\})$. We associate every index $1 \leq j \leq d$ and element of the field $a \in \mathbb{F}$ with a one of the $m = d \cdot |\mathbb{F}|$ sets, denoted $C_{j,a}$. Construct a Set-Cover instance $(\mathbb{U}, \mathbb{S})$ as follows. Let the universe be $\mathbb{U} \doteq [D] \times \Phi \times B$. For each index $1 \leq i \leq D$ let $\mathbb{U}_i \doteq \{i\} \times \Phi \times B$ denote the i-th duplicate. For each duplicate $1 \leq i \leq D$, a variable $x \in X$ and an assignment to that variable $a \in \mathbb{F}$, add a *variable set* covering the appropriate portion of every relevant sub-universe,

$$S(i, x, a) = \bigcup_{x=\varphi[j]} \{i\} \times \{\varphi\} \times C_{j,a}$$

For each test $\varphi \in \Phi$ and a *satisfying* assignment to its variables $a_1, ..., a_d \in \mathbb{F}$, add a *test set* covering the appropriate complements in all duplicates,

$$S(\varphi, a_1, ..., a_d) = \bigcup_{1 \leq i \leq D} \{i\} \times \{\varphi\} \times (C_{1,a_1} \cup ... \cup C_{d,a_d})^c$$

A.3 Correctness

Claim 1 (Completeness) *If the tests system Φ has a $(1,1)$-good multi-assignment ρ, then $(\mathbb{U}, \mathbb{S})$ has a set cover of cardinality $\left(1 + \frac{1}{\eta}\right) |\Phi|$.*

Proof: Let $\mathcal{C}$ contain sets of two types: variables sets and tests sets,

$$\mathcal{C} \doteq \bigcup_{1 \leq i \leq D, x \in X} S(i, x, \rho(x)) \cup \bigcup_{\varphi(x_{i_1}, ..., x_{i_d}) \in \Phi} S(\varphi, \rho(x_{i_1}), ..., \rho(x_{i_d}))$$

Note that $\mathcal{C}$ is a legal cover of the specified size. ■

Lemma 9 *If for some $1 \leq i \leq D$, $\mathbb{U}_i$ can be covered by at most $\delta l\,|\Phi|$ tests sets and $\eta\delta l\,|X|$ variable sets, then Φ has a $(\eta dl, 1-\delta c)$-good multi-assignment for Φ, where $c \doteq \ln \eta d + 1 + 1/\eta d$.*

Proof: Observe only $\mathbb{U}_i$, and let $\mathcal{C}$ indicate the given cover. For every $\varphi \in \Phi$, denote by V_φ the collection of variable sets that participate in covering it, and denote by T_φ the collection of such tests sets. Let $\mathcal{L}$ denote the tests $\varphi \in \Phi$ with low cost, $\mathcal{L} \doteq \{\varphi \in \Phi \mid f_p(|T_\varphi|, |V_\varphi|) \leq l\}$, where $f_p(x, y) = w_0 x + w_1 y$. Observe the multi-assignment $\hat{A}\colon X \to P(\mathbb{F})$ assigning each $x \in X$,

$\hat{A}(x) = \{a \in \mathbb{F} \mid S(i,x,a) \in \mathcal{C}\}$. According to lemma 7, for every $\varphi(x_{i_1},...,x_{i_d}) \in \mathcal{L}$, there must exist $a_1,...,a_d \in \mathbb{F}$, such that $\varphi(a_1,...,a_d) = \mathit{true}$ and for every $1 \le j \le d$, $S(i,x_{i_j},a_j) \in \mathcal{C}$. Therefore, For every $1 \le j \le d$, $0 < |\hat{A}(x_{i_j})| \le \eta dl$. $\mathcal{L}$ contains a significant portion of the tests, $|\mathcal{L}| \ge |\Phi| \cdot (1-\delta c)$, as $\sum_{\varphi\in\Phi-\mathcal{L}} f_p(|T_\varphi|,|V_\varphi|) \ge (|\Phi| - |\mathcal{L}|) \cdot l$, and

$$\begin{aligned}
\sum_{\varphi\in\Phi} f_p(|T_\varphi|,|V_\varphi|) &= \sum_{\varphi\in\Phi}\left(\frac{1}{\eta d}(1+1/\eta d)\,|V_\varphi| + \ln\eta d\,|T_\varphi|\right)\\
&= \frac{1}{\eta d}(1+1/\eta d)\sum_{\varphi\in\Phi}|V_\varphi| + \ln\eta d\sum_{\varphi\in\Phi}|T_\Phi|\\
&< \frac{1}{\eta d}(1+1/\eta d)\cdot\frac{d\,|\Phi|}{|X|}\eta\delta l\,|X| + \ln\eta d\cdot\delta l\,|\Phi|\\
&= \delta l\,|\Phi|\left(1+\frac{1}{\eta d}+\ln\eta d\right)
\end{aligned}$$

Hence, the multi-assignment $\hat{A}$ is $(\eta dl, 1-\delta c)$-good. ■

Claim 2 (Soundness) *If no assignment for Φ is $(\eta dl,\gamma)$-good, then the cardinality of the minimal set cover of $(\mathbb{U},\mathbb{S})$ is at least $\frac{l}{c}|\Phi|\cdot(1-\gamma)$, where $c \doteq \ln\eta d + 1 + 1/\eta d$.*

Proof: Assume, by way of contradiction, that our SET-COVER instance may be covered by less than $\frac{l}{c}|\Phi|\cdot(1-\gamma)$ sets. By simple averaging, there exists a duplicate $1 \le i \le D$ whose universe $\mathbb{U}_i$ is covered by less than $\frac{\eta l}{c}|X|\cdot(1-\gamma)$ variable sets and $\frac{l}{c}|\Phi|\cdot(1-\gamma)$ tests sets. Lemma 9 implies there exists a multi-assignment for Φ which is $(\eta dl,\gamma)$-good. This contradicts the premise of this claim. ■

Now we set the parameter l of the construction as to optimize the result and prove our improved hardness of approximation result for SET-COVER:

Proof:[of theorem 5] Fix some $\varepsilon > 0$. Let us confine ourselves to values $l = O(\log|\Phi|)$, so by corollary 7, the construction is efficient. Also, for $\gamma \doteq 2\,|\mathbb{F}|^{-1}(\eta dl)^d$, it holds $0 < \gamma < 1$. By lemma 8, deciding if a PCP instance has either $(1,1)$-good assignments, or no $(\eta dl,\gamma)$-good assignment at all, is $\mathcal{NP}$-hard. For a PCP instance $(X,\mathbb{F},\Phi)$, choose l, so that $l \ge \frac{1-\varepsilon}{1-\gamma}\ln|\mathbb{U}| \ge \frac{1-\varepsilon}{1-\gamma}(l + 4\ln|\Phi|)$, e.g, $l = \frac{4(1-\varepsilon)\ln|\Phi|}{1-\gamma+\varepsilon}$. By claims 1 and 2, the PCP problem reduces to deciding whether the constructed SET-COVER instance is coverable by $(1+1/\eta)\,|\Phi|$ sets or by at least $(1-\varepsilon)\,|\Phi|\,\frac{\ln|\mathbb{U}|}{c}$ for the constant c of claim 2. The theorem follows. ■