Explicit constructions of selectors and related combinatorial structures, with applications

Piotr Indyk
MIT
indyk@theory.lcs.mit.edu

1 Introduction

In this paper we present explicit constructions of several combinatorial objects: *selectors* [CGR00] and *selective families* [CGGPR00], *pseudo-random generators for proof systems* [ABRW00] and *fixed waking schedules* [GPP00]. As a result, we obtain almost optimal deterministic protocols for *broadcasting in unknown directed radio networks* [CGR00] and *wake-up problem* [GPP00]. We also show application of selectors (and its variants) to explicit construction of test sets for coin-weighting problems [DH00]. The parameters of our constructions come close to the best known non-constructive bounds. The constructions are achieved using a common technique, which could be of use for other problems.

Selective families and selectors. A family $S = S_1 \dots S_l$ of subsets of U, $|U| = n$, is called a k-*selector* [CGR00], if for any pair X, Y of disjoint subsets of U such that $k/2 \leq |X| \leq k$ and $|Y| = k$, there exists S_i such that $|S_i \cap X| = 1$ and $|S_i \cap Y| = 0$; in such a case we say that S_i *separates* $x \in S_i \cap X$ from X and Y. The family S is called k-*selective* [CGGPR00], if for any $X \subset U$ such that $|X| \leq k$ there exists S_i such that $|S_i \cap X| = 1$. In both cases the value l is called the *size* of S; if the size of the universe is not clear from the context, we use names "(k, n)-selector" or "(k, n)-selective family". Note that the definitions of k-selectors and k-selective families are incomparable (in the sense that none is weaker than the other), although we will see that existence of *small* k-selectors also implies existence of small k-selective families.

By using probabilistic method is easy to construct k-selectors or k-selecting families of size $O(k \log n)$. It has been shown in [CMS01] that this bound is tight for k-selective families; it is easy to see that it implies a lower bound of $\Omega(k \log n / \log k)$ for k-selectors. Unfortunately, the best constructive upper bound know up till now was $O(k^2 \log_k^2 n)$, which followed from a construction of a stronger object called *strongly selective family* or *superimposed codes*. In the latter case, we require that for any X of size at most k and $x \in U - X$ there exists S_i such that $x \in S_i$ but $X \cap S_i = \emptyset$. Since such families have to have size $\Omega(k^2 \log_k n)$ [CMS01], it is not possible to improve the size of selectors/selective families significantly using this approach.

In this paper we present an explicit construction of k-selectors of size $O(kn^{o(1)})$, which also gives a construction of k-selective families of the same size. These parameters almost match the lower bounds. The actual magnitude of the $n^{o(1)}$ factor depends on best bounds on the degree of explicitly constructible dispersers [TaS96, RRV99, TUZ01] and is $\log^{O(1)} n$. As a result, we improve the bounds for several problems in distributed networks, as described below.

Deterministic broadcasting in distributed networks. Both selective families and selectors have been used (see [CGGPR00, CGOR00, CGR00, CMS01]) for designing deterministic broadcasting protocols for directed radio networks; see [BGI87] or the aforementioned papers for the detailed description of the problem and the computational model. Here we mention that the broadcasting problem can be easily solved in deterministic $\Theta(n)$ time if the graph is *undirected*, or in time $O(n^2)$ for general directed graphs. However, the first subquadratic-time protocol (with complexity $\tilde{O}(n^{11/6})$) has been constructed only recently in [CGGPR00]; the algorithm used k-selective families. Later, the bound has been reduced to $\tilde{O}(n^{3/2})$ in [CGOR00] and then to (almost optimal) $O(n \log^2 n)$ in [CGR00]. However, the last result is non-constructive, since it uses a probabilistic construction of k-selectors. Similarly, Clementi et al [CMS01] used probabilistic construction of k-selective families to obtain a protocol with complexity $\tilde{O}(dD)$, where d is the maximum degree of the network and D is its (directed) diameter; again this result is non-constructive.

As a result of our constructions of selectors, we obtain a constructive broadcasting protocol with almost optimal complexity of $O(n^{1+o(1)})$. This also yields an explicit protocol for *gossiping* with complexity $\tilde{O}(n^{3/2})$, via the reduction to broadcasting as in [CGR00]. This improves the earlier constructive bound of $\tilde{O}(n^{7/4})$ [CGR00].

Combinatorial group testing and coin-weighting problem. Using selectors, we also show how to construct efficient solution to the coin-weighting problems defined as follows. Consider the set of functions $f : \{0,1\}^n \to \{1 \dots k\}$ of the form

$$f_S(x) = \sum_{s \in S} x_s$$

where $S \subset \{1 \dots n\}$ has cardinality at most k. We want to find a small set X of points $x^1 \dots x^r \in \{0,1\}^n$, such that given the values of f_S at those points, we can uniquely determine S; we call X an *interpolation set* for the family of functions f_S.

The above problem has quite a long history (e.g., see [DH00], p. 216). It has been often phrased as the problem of detecting k counterfeit coins in a large set of n coins. All good coins are assumed to have same weight, and all bad coins are assumed the have same weight as well, but the weights of good and bad coins are different. It is assumed we know the two weight values a priori. The goal is to discover the bad coins by repeatedly choosing a subset of coins and measuring their weight; the number of measurements should be as small as possible. It is easy to see that if the choice of measured subsets has to be done a-priori (i.e., we cannot choose the measured subsets based on the results of previous measurements), the solution to the coin problem is equivalent to finding a small interpolation set for the functions f_S.

Since there are $2^{\Omega(k \log n)}$ different functions f_S, and each $f(x^i)$ yields only $\log k$ bits of information, we need at least $\Omega(k \log n / \log k)$ points. It has been observed in [I99] that an explicit interpolation set of size $O(k \log n)$ can be constructed from sample spaces with ϵ bias; in fact, that result holds even for the case when the sum is modulo 2. However, there was no efficient way of actually *decoding* the function f_S given its values on in the interpolation set[1]. If we insist on interpolation set which can be decoded in time polynomial in n and k, the best interpolation set was based on superimposed codes and therefore was of size quadratic in k (specifically, $\Theta(k^2 \text{polylog} n)$). We note that the latter bound holds even for the more difficult case when the arithmetic sum in the definition of f is replaced by the boolean one (i.e., OR).

In this paper we show how to construct interpolation set of size $l \log n$, given a k-selective family of size l; thus, its size is $k 2^{\log^{O(1)} \log n}$. Moreover, the function f_S can be recovered in time polynomial in the size of the interpolation set. Note that in most cases this means that the running time of the algorithm is *sublinear* in n.

Expander matrices. A binary matrix A of size $m \times n$ is called a (r, s, c)-*expander* [ABRW00] if

- the number of ones in every row is at most s, and
- for set of rows I, $|I| \le r$, we have $|\sigma_A(I)| \ge c|I|$, where $\sigma_A(I)$ is the set of all columns J such that A restricted to $I \times J$ has at most one 1 per column.

A natural interpretation of an expander matrix is to treat it as an adjacency matrix of a bipartite graph $G = (X, Y, A)$ with $|X| = m$ and $|Y| = n$. The constraints imposed on the matrix now mean that

- the left degree of the graph is at most s, and
- for any set I of left vertices, the number of neighbors of I in B connected to only one element in I is at least $c|I|$

It was shown in [ABRW00] that given an $(r, s, 2)$-expander one can construct a pseudorandom generator PRG:$\{0,1\}^n \to \{0,1\}^m$ whose "breaking" requires a resolution proof of size $\exp(\Omega(\frac{r^2}{ms}))$ (the formal definition of pseudorandom generators for proof systems is beyond the topic of this paper). They also showed existence of an $(n/\log n, \log n, 0.75 \log n)$-expander of size $n^2 \times n$. In this paper we construct an $(n/2, 2^{\log^{O(1)} \log n}, 2)$-expander of size $n^2 \times n 2^{\log^{O(1)} \log n}$, which yields essentially optimal explicit pseudorandom generators.

Fixed waking schedules. Finally, consider the following *wake-up* problem, first investigated in [GPP00]. There are n processors which can communicate in synchronous steps though a common channel. The channel has the following property: if only one processor writes to it, the information is broadcasted to all other processors. However, if two or more processor write to the channel at the same time, no information is broadcasted. Initially, all processors are *asleep*. At any point in time, one or more of the processors wake up and try to wake up the remaining processors by sending them messages through the channel. Unfortunately, at the moment of the wake-up, the local clocks of the processors are set to 0. Thus each processor can have different time, although the *increments* of all clocks of alert processors are the same. Despite of that we want the processors to perform a fixed sequence of communication trials such that at least one of these trials is successful.

In order to solve this problem, we use a combinatorial structure defined as follows. Let P be a set of

[1] By enumerating all possible functions f_S one can achieve a running time of n^k; however, we do not consider this algorithm to be efficient.

pairs of integers such that for any two different pairs $(i,j),(i',j') \in P$ we have $i \neq i'$; we call P a *shift pattern*. For any $n \times l$ binary matrix $S[i,j]$, $i \in [n], j \in [l]$, we define S^P to be a binary matrix obtained by vertical concatenation of rows $r_0 \dots r_{n-1}$ such that $r_i[j] = S[i, j-p_i]$ if $(i, p_i) \in P$ and r_i consists of zeros otherwise (the out-of-range values are assumed to be zero). For any $(i,j) \subset [n] \times [l]$, if $S[i,j] = 1$ and $S[i',j] = 0$ for all $i' \neq i$, then we say that (i,j) is *free* in S; otherwise we say it is *non-free* in S.

Now we define the fixed schedule as follows. An $n \times l$ binary matrix S is called an *n-schedule* of length l, if for any shift pattern $P \subset [n] \times \{0, \dots \infty\}$ containing $(i, 0)$ for some i there exist a free pair in $S^P[\cdot, 0 \dots l-1]$.

It is not difficult to see that in order to solve the wake-up problem, the processors' communication pattern has to follow an n-schedule, with $S[i,j] = 1$ meaning that the i-th processor tries (if alert) to send a message during the j-th step (according to its local time). Therefore, it is of interest to construct n-schedules of short length. It was shown in [GPP00] that schedules of length $O(n \log^2 n)$ *exist*; however, the proof again used probabilistic method. They also gave a construction of an n-schedule of length $O(n^2)$.

In this paper we show how to construct an n-schedule of size $O(n^{1+\epsilon})$, for any $\epsilon > 0$. The construction is "almost" explicit, in the sense that it can be performed using a polylogarithmic number of random bits; a completely deterministic construction can be done by enumerating all random bits, which can be done in quasi-polynomial time.

1.1 Overview of techniques. The two constructions of selectors, expanders and fixed waking schedules, although different, on the high-level share the same basic idea. The technique is best explained by the example. Consider the problem of constructing a k-selective family. If we k was close to n, a trivial family $S_i = \{i\}$ would be sufficient. Therefore, it would be beneficial to "reduce" n to $O(k)$, by mapping (possibly several times) the original universe U into a smaller set U' such that the mapping in some sense preserves the identity of the elements of U. One could perform such a reduction by perfect hashing, or via balanced partitioning (cf. [NSS95]). However, such a construction would not lead to $|U'| = O(k)$, unless unacceptably large number of mappings was used. Therefore, we apply weaker mappings (constructed using dispersers), which can always merge some elements of U, but do it in a way robust enough to make sure that a k-selective family for U' can be "pulled back" into a k-selective family for U. We mention that a similar idea has been used earlier in [RR99, TUZ01].

A similar idea works for n-schedules, which we construct in a recursive fashion as follows. Assume that we have an m-schedule S with the property that not only *one* but *many* positions are free under any shift pattern. Then, if we concatenate (vertically) n/m copies of S, then we obtain a schedule with n rows with the property that for each shift pattern there are many columns which contain small (say $O(n/m)$) number of 1's. Now, if we replace each 1 by a row of a variant of an n-schedule (say S') which works correctly if at most $O(n/m)$ rows are present (i.e., only $O(n/m)$ processors are awake), then the whole construction yields an n-schedule. The desired schedule S' is obtained by mapping the n rows into the universe of small size (say $O(n/m)$) and using the rows of a schedule S'' with $O(n/m)$ rows (i.e., essentially same as S). As before, some of the mapped elements (rows) collapse, but not enough of them to create a problem. The mapping itself could be done again using disperser-like graphs (extractors, to be specific); however, in our construction we use random mappings, since we need to use some randomness to construct S'' anyway.

2 Preliminaries

The following definition is a minor modification of the standard definition of a disperser:

DEFINITION 1. *A* (n,m,d)-disperser *is a bipartite graph* $G = (A, B, E)$ *such that* $|A| = n$, $|B| = m$ *and*

- *for any* $Z \subset A$ *s.t.* $|Z| = m/2$ *the size of the set* $\Gamma_G(Z)$ *of neighbors of* Z *has size at least* $m/4$,
- *the degree of each node in* A *is equal to* d

We will also use a *p-colliding* family $G = \{g_0 \dots g_{r-1}\}$ of functions $g_j : A \to [u]$, i.e. such that for any $x, y \in A$, $x \neq y$, we have $\Pr_{g \in G}[g(x) = g(y)] \leq p$. It is well-known that for such families can be constructed with parameters $r = O(1/p \cdot \log n)$ and $u = O(1/p \cdot \log^2 n)$ (e.g., we can choose $r = O(1/p \cdot \log n)$ primes $p_0 \dots p_{r-1}$ such that $p_i = \Theta(1/p \cdot \log^2 n)$ and define $g_j(x) = x \bmod p_j$).

3 Selectors

In this section we show how to construct a k-selector.

THEOREM 3.1. *Given a* (n,k,d)*-disperser, one can explicitly construct a* (k,n)*-selector of size* $O(kd^3 \log^3 n)$.

Observe that this also enables us to construct a k-selective family of (asymptotically) same size, by taking the union of $(2^i, n)$-selectors, for $i = 1 \dots \log k$. We also mention that the bounds achieved in the above theorem can be somewhat improved by using extractors or strong extractors.

Proof. Take a disperser (A, B, E) satisfying the above conditions. Since the degree of all nodes in A is d, one can construct d functions $h_0 \dots h_{d-1}$, $h_i : A \to B$, such that $h_i(x)$ is the i-th neighbor of x (assuming arbitrary ordering). We also use a p-colliding family $G = \{g_0 \dots g_{r-1}\}$ of functions $g_j : A \to [u]$ for $p = \frac{1}{48d}$; recall that we can assume $r = O(d \log n)$ and $u = O(d \log^2 n)$.

Now we construct the selector S as follows. Let $f_{ij} : A \to [k] \times [u]$, for $i \in [d]$, $j \in [r]$, be a family of functions defined as

$$f_{ij}(x) = (h_i(x), g_j(x))$$

The family S contains all sets

$$S_{s,t,i,j} = f_{ij}^{-1}((s, t))$$

for $i \in [d], j \in [r], s \in [k/2], t \in [u]$.

Clearly, the size of S is $O(kd^3 \log^3 n)$. It remains to show that S is a k-selector. Fix sets X and Y. Observe that from the expansion property of G it follows that for any $Z \subset A$, $|Z| \geq k/2$ there exists h_i such that $|h_i(Z)| \geq \frac{k}{4d}$. Take such h_i for $Z = X$. We define $X_b = h_i^{-1}(b) \cap X$ and $Y_b = h_i^{-1}(b) \cap Y$, for $b \in B$. Also, let B' be the set of all b's for which $X_b \neq \emptyset$; note that $|B'| \geq \frac{k}{4d}$. Since X_b's are disjoint, we know that the average size of X_b for $b \in B'$ is at most $k/|B'| \leq 4d$. Therefore, at least $2/3 \cdot |B'|$ of such X_b's have cardinalities smaller than $12d$. The same arguments shows that there are at least $2/3 \cdot |B'|$ elements $b \in B'$ such that $|Y_b| \leq 12d$. Thus, there are at least $|B'|/3$ of elements $b \in B$ such that $|X_b| \leq 12d$ and $|Y_b| \leq 12d$. Denote the set of such b'a by B''.

For any $x \in h_i^{-1}(B'') \cap X$, we define an indicator variable $I_x^j \in \{0, 1\}$ depending on a random choice of $j \in [r]$. The value of I_x^j is equal to 1 iff

$$g_j(x) \notin g_j(X_{h_i(x)} \cup Y_{h_i(x)} - \{x\}) \tag{3.1}$$

Observe that

$$\Pr_{g \in G}[g(x) \in g(X_{h_i(x)} \cup Y_{h_i(x)} - \{x\})] \leq 24d \cdot p = 1/2$$

and therefore

$$E_{g_j \in G}[\sum_x I_x^j] \geq |B''|/2$$

Since $|B''|/2 \geq 1$, there exists a pair (x, j) such that (1) is true. Note that in such a case $f_{ij}(x) \notin f_{ij}(X \cup Y - \{x\})$, and therefore the set $S_{h_i(x), g_j(x), i, j}$ separates x from X, Y.

3.1 Application to combinatorial group testing

In this section we present an application of explicit k-selective families to combinatorial group testing. The construction is as follows. Let $S_1 \dots S_l \subset \{1 \dots n\}$ be a k-selective family. For each $i = 1 \dots l$, we create $\log n + 1$ *batches* of points x^{ij}, $j = 0 \dots \log n$. Each x^{ij} is defined by setting

$$x_s^{ij} = 1 \text{ iff } s \in S_i \text{ and } \mathrm{bin}(s)_j = 1$$

where $\mathrm{bin}(s)_j$ is the jth bit of s. The 0-th bit of every number is always assumed to be 1.

Given the values of $f_S(x^{ij})$, we can recover S as follows. Firstly, we search for a batch $x^{i_1 \cdot}$ for which f_S returns either 0 or 1. Note that this in particular implies that $|S \cap S_{i_1}| = 1$. In such a case, one can easily observe that the values $f_S(x^{i_1 1}), f_S(x^{i_1 2}), \dots$ give the binary representation of the unique $s_1 \in S \cap S_{i_1}$. Therefore, we recovered one element of S.

Consider now the set $S - \{s_1\}$. Since its cardinality is less than k, there exists (not necessarily unique) i_2 such that $|S_{i_2} \cap (S - \{s_1\})| = 1$. We can discover such an i_2 as follows. Partition the sets S_i into two groups: the first group consists of sets containing s_1, and the second one of those which do not contain s_1. If S_{i_2} belongs to the first group, we can discover it by observing that $|S_{i_2} \cap (S - \{s_1\})| = 1$ if and only if:

- $f(x^{i_2 0}) = 2$
- $f(x^{i_2 1}) - \mathrm{bin}(s_1)_1, f(x^{i_2 2})\mathrm{bin}(s_1)_2, \dots$ is a binary representation of some $s_2 \in S - \{s_1\}$.

If S_{i_2} belongs to the second group, we proceed in a similar way. In particular, we require $f(x^{i_2 0}) = 1$ and use the sequence $f(x^{i_2 1}), f(x^{i_2 2}), \dots$ as a binary representation of $s_2 \in S - \{s_1\}$. Note that the above two cases are not mutually exclusive.

In any case, we obtained another element of S, namely s_2. This procedure can be iterated futher to obtain the remaining elements of S. In particular, to obtain the u-th element of S, we split the sets S_i into (at most l) groups indexed by $S_i \cap \{s_1 \dots s_{u-1}$. Then we discover the i_u from the the group indexed by a set A by requiring $f(x^{i_u 0}) = |A| + 1$, and taking s_u to be the number represented by the sequence

$$f(x^{i_2 1}) - \sum_{a \in A} \mathrm{bin}(a)_1, f(x^{i_2 2}) - \sum_{a \in A} \mathrm{bin}(a)_2, \dots.$$

4 Expander matrices and explicit pseudorandom generators hard for resolution

In this section we show an explicit construction of *expander matrices*. The construction is similar to the construction of selectors. To start, we need an sequence of graphs $H_i = (X, Z_i, E_i)$, $i = 0 \ldots \log n$, where each H_i is a $(n^2, n/2^i, d)$-disperser (we assume all Z_i's are disjoint). Then form a graph $H = (X, Z, E')$, where $Z = \cup_i Z_i$ and $E' = \cup E_i$. It is easy to see that the left degree of G is equal to $d' = d \log n$, and for any set $I \subset X$, $|I| \leq n/2$, we have $|\Gamma(I)| \geq |I|/8$.

Equipped with the graph H, we construct the expander G as follows. Let F be a $1/t$-colliding family of functions $g_i : X \to [u]$, for $t = 32d'$; note that we can take $u = d2^{\log^{O(1)} \log n}$. The graph $G = (X, Y, E)$ is defined over X and the set $Y = Z \times [u]$, with the edges of the form $< x, (y, f(y)) >$, for all $< x, y > \in E'$ and $f \in F$. Note that $|Y| = n2^{\log^{O(1)} \log n}$.

In the following we show that the resulting graph G is an $(n/2, d' \cdot |F|, 2)$-expander. To see this, consider any $I \subset X$, $|I| \leq n/2$. By the previous arguments we know $|\Gamma(I)| \geq |I|/8$. Since there are at most $d'|I|$ edges leaving I, there are at least $|I|/16$ elements of $\Gamma(I)$ which are adjacent to at most $16d'$ elements of I. Denote that set by J. Since F is $1/t$-colliding, it follows that for at least $1/2$ fraction of the edges of the form $< x, (y, f(y)) >$, $x \in I, y \in J$, there is no other edge $< x, z > \in E$ such that $z = (y, f(y))$. Thus, the number of such edges is at least

$$1/2 \cdot |J| \cdot |F| \geq |I|/32 \cdot |F| \geq 2|I|$$

for large enough F.

References

[ABRW00] M. Alekhnovich, E. Ben-Sasson, A. Razborov and A. Wigderson, "Pseudorandom Generators in Propositional Proof Complexity", FOCS'00.

[AS00] N. Alon, J. Spencer, "Probabilistic method", 2nd edition, 2000.

[BGI87] R. Bar-Yehuda, O. Goldreich and A. Itai, "On the Time-Complexity of Broadcast in Radio Networks: An Exponential Gap Between Determinism and Randomization", PODC'87, p. 98-108.

[CMS01] A. E. F. Clementi, A. Monti and R. Silvestri, "Selective Families, Superimposed Codes, and Broadcasting on Unknown Radio Networks", SODA '01.

[CGOR00] , B.Chlebus, L.Gasieniec, A.Ostlin and M.Robson, "Deterministic Radio Broadcasting", ICALP'00.

[CGGPR00] , B.Chlebus, L.Gasieniec, A.Gibbons, A.Pelc and W.Rytter, "Deterministic broadcasting in unknown radio networks", SODA'00.

[CGR00] M. Chrobak, L.Gasieniec, W. Rytter, "Fast Broadcasting and Gossiping in Radio Networks", FOCS'00.

[DH00] D.Z. Du and F. K. Hwang, "Combinatorial group testing and its applications", World Scientific, 2000.

[DP01] G. De Marco and A. Pelc, "Fast distributed coloring with $O(\Delta)$ colors", SODA'01.

[GPP00] L. Gasieniec, A.Pelc, D.Peleg, "The Wakeup Problem in Synchronous Broadcast Systems", PODC'00.

[I99] "Interpolation of Symmetric Functions and a New Type of Combinatorial Design",STOC'99.

[Lin92] N. Linial, "Locality in distributed graph algorithms", SICOMP 21 (1992), pp. 193-201.

[NSS95] M. Naor, L. J. Schulman and A. Srinivasan, "Splitters and Near-Optimal Derandomization", FOCS'95, p. 182-191.

[PS92] A. Panconesi and A. Srinivasan, "Improved distributed algorithms for coloring and network decomposition problems", STOC'92.

[PS96] A. Panconesi and A. Srinivasan, "On the complexity of distributed network decomposition", *Journal of Algorithms* 20 (1996), pp. 356-374.

[RR99] R. Raz and O. Reingold, "On Recycling the Randomness of States in Space Bounded Computation", STOC'99.

[TUZ01] A. Ta-Shma, C. Umans and D. Zuckerman, "Lossless Condensers, Unbalanced Expanders, and Extractors", STOC'01.

[RRV99] R. Raz, O. Reingold and S. Vadhan, "Extracting all the Randomness and Reducing the Error in Trevisan's Extractors", STOC'99.

[RVW00] O. Reingold, S. Vadhan and A. Wigderson, "Entropy Waves, The Zig-Zag Graph Product, and New Constant-Degree Expanders and Extractors", FOCS'00.

[TaS96] A. Ta-Shma, "On Extracting Randomness From Weak Random Source", STOC'96.

A Fixed waking schedules

In this section we show how to construct a short n-schedule. The construction is done in a hierarchical fashion. More specifically, we show how to compose (a variant of) an n-schedule of length l with (a variant of) an n'-schedule of length l' to obtain an $n \cdot n'$-schedule of length close to $l \cdot l'$. This composition could be done using explicitly constructible expanding graphs using technique similar to the one in the previous section. However, it turns out that better result can be obtained by using probabilistic arguments, since we need to perform quasi-polynomial (i.e., $2^{\log^{O(1)} n}$) time exhaustive search at the base of our recursion anyway. At the end, the composition theorem allows us to obtain an n-schedule of small length deterministically, starting from a schedule of size $\log^{O(1)} n$ (which can be found in quasi-polynomial time).

In order to make the composition work, we need a stronger and more general definitions of a schedule (in fact, we need two of them). Firstly, we need to generalize the notion of a shift pattern. The (generalized) shift pattern P of *period* l consists of pairs of integers such that if $(i,j),(i,j') \in P$, then $j-j'=kl$ for some integer k. We say that l is a *strong* period of P if P has period l and for any $(i,j) \in P$ we also have $(i,j+kl) \in P$, for any $k \geq 0$.

For any $n \times l$ binary matrix S and a shift pattern P, we define $S \oplus P$ to be a binary matrix such that $(S \oplus P)[i,j]=1$ iff there exists $(i,j') \in P$ such that $S[i,j-j']=1$. Note that if P has period l, then there can be only one such pair (i,j'); we say that this pair is associated with the pair (i,j).

Also, the number of ones in a matrix S is called its *density*. If the number of ones in all rows of S is the same, we call this number *row density*.

DEFINITION 2. *An $n \times l$ binary matrix S is called an* (n,k,α,ϵ)-schedule *of length l, if for any shift pattern P with period l, $|P| \leq k$, such that $|P'|/|P| \geq \alpha$ for $P' = P \cap ([n] \times [l])$, we have that the number of pairs free in $S \oplus P$ associated with a pair from P', divided by the density of $S \oplus P'$, is at least $1-\epsilon$ (we require that the density of $S \oplus P'$ is non-zero).*

The second definition is as follows. For any shift pattern P, we use $P(t)$ to denote the number of pairs $(i,j) \in P$ such that $i \leq t$.

DEFINITION 3. *An $n \times l$ binary matrix S is called an* (n,δ)-schedule *of length l, if for any shift pattern P with strong period l such that $P(0)>0$ and $P(l-1)/P(0)) < 2$, if we define $A=(S \oplus P)[\cdot, 0 \dots l-1]$, then the ratio of the number of free pairs in A to the density of A is at least $1-\delta$ (we require that the density of A is non-zero).*

We observe that if S is a (n,δ) for $\delta<1$ of length l then by concatenating horizontally $\log_2 n+1$ copies S we obtain an n-schedule (of length $O(l \log_2 n)$). To see why the concatenated matrix (say, S') is an n-schedule, consider any (non-generalized) shift pattern P and observe that for each $i=1 \dots \log_2 n+1$ it induces a (generalized) shift pattern with strong period l over the i-th copy of S in S'; also $P(0) \geq 1$. Note that there are at most $\log_2 n$ integers $i \geq 0$ such that $P(l(i+1)-1)/P(li) \geq 2$. Therefore, there exists $i \geq \log_2 n$ such that $P(l(i+1)-1)/P(li)) < 2$, in which case there exists a free pair in $(S^P)[\cdot, li \dots l(i+1)-1]$ (in fact, there are many of such pairs).

Firstly, we show that short (n,k,α,ϵ)-schedules exist.

LEMMA A.1. *There exist a (n,k,α,ϵ)-schedule S of length $l=O(\frac{k \log n}{\epsilon^2 \alpha})$. It can be found deterministically in time $2^{O(kl \log n)}$.*

Proof. Select S by setting in each row $S[i,\cdot]$ exactly a fraction of $p=\frac{\epsilon}{2k}$ of position to 1; the positions are chosen at random without replacement. With a constant probability such S is a desired schedule; the proof is similar to the proof of Theorem 3.5 in [GPP00].

Since the definition of (n,k,α,ϵ)-schedules involves only kl positions at a time, instead of using fully independent random variables $S[i,j]$ it is sufficient to use variables which are $O(kl)$-wise independent. It is known [AS00] that there exist $O(kl)$-wise independent probability spaces for the random variable $S[\cdot,\cdot]$ of size $(nk)^{O(kl)}$, thus by enumerating all elements of such spaces we can find a schedule with desired properties in $(nk)^{O(kl)}$ time. Testing if a given S is a schedule with desired properties can be clearly done in less than $2^{O(kl \log n)}$ time.

Now we proceed with the composition lemma.

LEMMA A.2. *Given an (n,δ)-schedule S_1 of length l_1 with row density $g_1 \geq 1/\epsilon$ and an $(n\lambda,k,\epsilon,\epsilon)$-schedule S_2 of length l_2 with row density g_2, where $k=\lambda(1+1/\epsilon)$ and $\epsilon<1/2$, one can construct an $(n\lambda,\delta+O(\epsilon))$-schedule S of length $O(l_1 \cdot l_2/\epsilon)$.*

Proof. We construct S as follows. Firstly, we concatenate S_1 vertically λ times to obtain matrix S_1', then we concatenate S_1' horizontally d times to obtain M, where $d=1/\epsilon$. Finally we replace each bit of M by l_2 bits in the following way:

- if $S_1'[i,j]=1$, we replace it by the i-th row of S_2
- otherwise, we replace it by a sequence of l_2 zeros

Clearly, the resulting matrix S is a $n\lambda \times l$, for $l=dl_1l_2$ matrix. It remains to show that it has the desired schedule properties. To this end, consider any shift pattern P with strong period l such that $P(0)>0$ and $P(l-1)<2P(0)$. Let $P_{/l_2}=\{(i,\lfloor j/l_2 \rfloor)\}:(i,j) \in P\}$. Observe that $P_{/l_2}(dl_1-1) < 2P_{/l_2}(0)$. Consider the matrix $F=M^{P_{/l_2}}[\cdot,0 \dots l_1d-1]$. Let c_j be the number of 1s in the j-th column of $M^{P_{/l_2}}$. For $j=0 \dots l_1d-2$ we say that a pair (i,j) for which $F[i,j]=1$ is *good* if

- $c_{j-1}+c_j+c_{j+1} \leq k$, and
- $c_j \geq \epsilon(c_{j-1}+c_j+c_{j+1})$

Note that no pair (i,l_1d-1) can be good by definition.

In the following we will show that the fraction of good pairs is high. Before that, however, we will need to make sure that F contains a high fraction of free pairs (note that since $P(0)>0$, there is at least one position set to 1 in F).

LEMMA A.3. *The fraction of free pairs in F is at least $1-\delta-O(1/d)$.*

Proof. We define

- $B_{ij} = F[in \dots (i+1)n-1, jl_1 \dots (j+1)l_1 - 1]$
- b_{ij}: the number of non-free ones in B_{ij}
- s_{ij}: the density of B_{ij}
- $P_i = [P_{/l_2} \cap \{il_1 \dots (i+1)l_1 - 1\} \times \{-\infty \dots \infty\}]$, $p_i = P_i(0)$, $q_i = P(dl_1 - 1)$

We classify the blocks B_{ij} into two classes. If $P_i((j+1)l_1 - 1) < 2P_i(jl_1)$, then we say that B_{ij} is *flat*. If the above does not hold and $P_i((j+1)l_1-1) > 0$, we say that B_{ij} is *steep*. Otherwise (i.e., when $P_i((j+1)l_1 - 1) = P_i(jl_1) = 0$), we say B_{ij} is *empty*. In the following we disregard empty blocks, since they have no impact on the correctness of the lemma.

We observe that if B_{ij} is flat, then from the definition of S_1 we know that $b_{ij} \le \delta s_{ij}$, so the total fraction of non-free ones in flat blocks is at most δ. Therefore, it is sufficient to bound the number of of non-free ones incurred in steep blocks (by $O(1/\epsilon)$). Note that if a block is steep, then it could be that $b_{ij} = s_{ij}$.

To bound the latter quantity, recall that the row density of S_1 is g_1. Therefore, we have $g_1 P_i((j+1)l_1 - 1) \le s_{ij} \le g_1 P_i((j+1)l_1 - 1)$. For any i, let j_i be the largest integer such that B_{ij_i} is a steep block. Observe that for any i, the sum of densities of all steep blocks among B_{ij}, $j = 0 \dots d-1$, can be upper bounded by $2g_1 P_i((j_1+1)l_1 - 1)$, which is at most $2g_1 q_i$.

On the other hand, we know that that the total density of F is lower bounded by $g_1 d P_{/l}(0) = g_1 d \sum_i p_i$. Also, by the assumption, we know that $\sum_i q_i < 2 \sum_i p_i$. Therefore, the sum of densities of all steep blocks can be bounded by

$$2g_1 \sum_i q_i < 4g_1 \sum_i p_i$$

which is at most $4/d$ of the density of F. Therefore, the total fraction of free ones in F is at least $1 - \delta - 4\epsilon$.

LEMMA A.4. *The ratio of good pairs in F to the density of the latter matrix is at least $1 - \delta - C\epsilon$, for a fixed constant C independent from ϵ and δ.*

Proof. We know that fraction of free pairs in F is at least $1 - \delta - O(\epsilon)$. Consider first the columns c_j for $j < l_1 d - 1$ (i.e., all columns of F except for the last one). Clearly, all free pairs contribute at most λ to each c_j. Therefore, if $c_{j-1} + c_j + c_{j+1}$ exceeds $k = \lambda(1 + 1/\epsilon)$, it must be due to the contribution of at least λ/ϵ non-free pairs. Since the fraction of non-free pairs is at most δ and each such pair contributes to at most 3 columns, the fraction of pairs satisfying the first condition is at least $1 - \delta - 3\delta = 1 - 4\delta$. It is easy to verify that the second condition removes only $O(\epsilon)$ fraction of free pairs. Finally, the number of ones in the column of F is at most $P_{/l_2}(l_1 d - 1)$, which is at most $2P_{l_2}(0)$. Since the density of F is at least $g_1 P(0)$, it follows that the last column contains at most a fraction $O(1/g_1) = O(\epsilon)$ of all ones in F.

From our construction we know that each 1 of $M^{P_{/l}}$ corresponds to a row of S_2 shifted by at most l_2. Fix a column j in $M^{P_{/l}}$ and consider the set G_j of good pairs from that column. Clearly, an S_2 row corresponding to a good pair from G_j can overlap (in S) with at most $k-1$ other rows. Moreover, the ratio of the number of such rows to the total number of overlapping rows is at least ϵ. By properties of S_2 we conclude that the fraction of free ones (in S^P) in the rows corresponding to G_j is at least $1 - \epsilon$. By summing over all js we conclude that S is a $(n\lambda, \delta + C\epsilon)$-schedule.

The above composition lemma suggests the following way of constructing schedules:

- construct a $(n_1, \cdot)$-schedule S_1
- construct a $(n_1 n_2, Dn_2, \cdot, \cdot)$-schedule S_2, for a parameter $D = D(\epsilon)$
- compose S_1 and S_2 into a $(n_1 n_2, \cdot)$-schedule using Lemma A.2.

The schedule S_1 can be constructed recursively using the same procedure. The only missing link is the construction of S_2, since Lemma A.2 does not provide us with a way to construct $(n, k, \cdot, \cdot)$-schedules. In the preliminary version of this paper we showed that $(k, k, \cdot, \cdot)$-schedules can be easily extended to $(n, k, \cdot, \cdot)$-schedules by using expander graphs, and thus S_2 could be also constructed recursively. It turns however, that one can get a schedule with better parameters by keeping S_2 fixed and recursing only on S_1. Therefore we only describe the latter construction.

Let $\lambda = \log^c n$, $\epsilon = \frac{1}{a \log n}$; this implies $k = O(\log^{c+1} n)$. By Lemma A.1 there exists (and can be constructed in time quasi-polynomial in n) an $(n, k, \epsilon, \epsilon)$-schedule S_2 of length $l_2 = O(\frac{k \log k \log n}{\epsilon^3}) = \log^{c+5+o(1)} n$. We will construct a sequence of $O(\log n)$ schedules $S^1 \dots$ starting from a $(\log^{O(1)} n)$-size (λ, ϵ)-schedule S_1. The schedule S^{i+1} is obtained by using Lemma A.2 for S^i and S_2. We stop when the number of rows in $S^i = S^{i^*}$ exceeds n. By Lemma A.2 S^{i^*} is an $(n, \frac{O(i^*)}{a \log n})$-schedule, which is an n-schedule for a suitable constant a.

Let $n(i)$ be the number of rows of S^i and $l(i)$ be the length of S^i. Clearly, $n(i) = \lambda^i = \log^{ci} n$ and

$l(i) = (l_2/\epsilon)^i = \log^{(c+6+o(1))i} n$. Therefore, $l(i) = (n(i))^{1+6/c+o(1)}$. Since $n(i^*) \leq n\lambda = n\log^c$, we conclude that $l(i^*) = n^{1+6/c+o(1)}$.

In this way, we proved the following theorem.

THEOREM A.1. *An n-schedule of length $n^{1+\epsilon}$ can be constructed in deterministic quasi-polynomial time for any constant $\epsilon > 0$.*