Optimal Reconstruction of Graphs Under the Additive Model

Vladimir Grebinski and Gregory Kucherov *

Abstract

We study the problem of combinatorial search for graphs under the additive model. The main result concerns the reconstruction of *bounded degree* graphs, i.e. graphs with the degree of all vertices bounded by a constant d. We show that such graphs can be reconstructed in $O(dn)$ non-adaptive queries, which matches the information-theoretic lower bound. The proof is based on the technique of separating matrices. Here a central result is a new upper bound for a general class of separating matrices. As a particular case, we obtain a tight upper bound for the class of d-separating matrices, which settles an open question stated by Lindström in [19]. Finally, we consider several particular classes of graphs. We show how an optimal non-adaptive solution of $O(n^2/\log n)$ queries for general graphs can be obtained. We also prove that trees with unbounded vertex degree can be reconstructed in linear number of queries by a non-adaptive algorithm.

1 Introduction and Definitions

Combinatorial Search studies problems of the following general type: determine an unknown object by means of indirect questions about this object. Perhaps the most common example of combinatorial search is the variety of problems of determining one or several counterfeit coins in a set using scales of some kind. Many of these problems still lack an optimal general solution.

Each instance of a Combinatorial Search problem has two main components: a finite *domain of objects* $\mathcal{M}$ and a *class of queries* $\mathcal{Q}$, which is a family of functions from the domain of objects to a domain $\mathcal{A}$ of *answers*. Given $\mathcal{M}$ and $\mathcal{Q}$, the combinatorial search problem is to find a sequence of queries $(q_1, q_2, \ldots, q_k)$, $q_i \in \mathcal{Q}$, such that the sequence of answers $(q_1(x), q_2(x), \ldots, q_k(x))$ uniquely identifies the object $x \in \mathcal{M}$. A method for choosing queries $(q_1, q_2, \ldots, q_k)$ is called a *(combinatorial) search algorithm*. The complexity measure of a search algorithm is the maximal number k of required queries over all $x \in \mathcal{M}$. This implies that we are concerned with *query complexity* only. Precise complexity bounds to combinatorial search problems can be rarely obtained. Instead, one is usually interested in the asymptotic complexity, when $|\mathcal{M}|$ tends to infinity.

Monographs [3, 8] present detailed accounts of numerous results on Combinatorial Search problems. Variants of these problems abound in different application domains.

*INRIA-Lorraine and CRIN/CNRS, 615, rue du Jardin Botanique, BP 101, 54602 Villers-lès-Nancy, France, e-mail: {grebinsk,kucherov}@loria.fr

For example, paper [13] deals with a problem motivated by genome analysis. Note that Combinatorial Search is closely related to Learning Theory, where the general framework is similar, except possibly that there is usually an infinity of objects and one is looking not necessarily for the object itself but for its approximation according to a given distance function.

In general, the choice of q_i in the sequence $(q_1, q_2, \ldots, q_n)$ depends on answers $(q_1(x), \ldots, q_{i-1}(x))$ obtained "so far". If this dependence exists, the algorithm is called *adaptive* (or sequential). Otherwise, when all the queries can be given before any answer is known, the algorithm is called non-adaptive (or predetermined). In this paper, we deal with non-adaptive algorithms. Although they are obviously less powerful in general, non-adaptive algorithms usually admit "nicer" mathematical formulations that allow to use more powerful mathematical methods. Besides, in many cases (including those considered in this paper) non-adaptive algorithms achieve the power of adaptiveness. Note also that in non-adaptive algorithms all queries can be made in parallel, which is useful in many applications.

For the non-adaptive case, we reformulate the combinatorial search problem as follows: find a minimal number of queries $q_1, q_2, \ldots, q_n \in \mathcal{Q}$ such that for every $x, y \in \mathcal{M}$, there is q_i, $1 \leq i \leq n$, such that $q_i(x) \neq q_i(y)$.

In contrast to the coin weighing problem where objects of $\mathcal{M}$ are just elements or subsets of elements of a given set, the objects may be of a more complex nature, such as graphs or partially ordered sets (see [2, 3]). In case of graphs, different combinatorial search problems can be raised. One may look for an unknown edge in a *given* graph by asking, for a subset of vertices, whether one of the edge's endpoints (or both) belongs to the subset. A more general problem, considered in this paper, consists of determining an unknown graph of a given class. Here again, subsets of vertices are queried, but the answer returned characterizes some property of the subgraph induced by the subset. Finally, the third type of problem is to check whether an unknown graph belongs to a given class without actually determining the graph. This problem, known as *property testing*, received much attention in connection with the study of *evasiveness* property (see [20]). Another approach to property testing, in the framework of probabilistic algorithms and approximation, was recently introduced in [11, 12].

It is clear that for the same object domain $\mathcal{M}$, different classes of queries $\mathcal{Q}$ lead to combinatorial search problems of different type and different complexity. Under the *additive model*, the domain of answers is the ring of integers $\mathbb{Z}$. This model is also called *quantitative*, as the queries $\mathcal{Q}$ are usually about some quantitative property of the object. A typical example is to identify the subset of counterfeit coins using a spring scale under the knowledge of the difference in weight between a counterfeit and authentic coin (which allows to determine the number of counterfeit coins in a subset by weighing this subset). We will come back to this example in Section 2. Some additive models of combinatorial search are studied in [15, 10, 14].

In this paper we consider the problem of *searching for a graph under the additive model* defined as follows. The domain of objects, denoted $\mathcal{G}_n$ is a class of simple graphs with n vertices labelled by natural numbers $1, 2, \ldots, n$. (A graph is simple if it does not contain loops and multiple edges.) The queries that we are allowed to make about $G \in \mathcal{G}_n$ are of the following form: For a subset $V \subseteq \{1, \ldots, n\}$ of vertices, how many edges are there in G between vertices of V? More formally, how many edges occur in the intersection $G \cap K_V$, where K_V is the complete graph with the set of vertices V?

In this paper we develop new techniques of non-adaptive additive search for a graph. Our main result concerns *bounded degree* graphs, i.e. graphs with the degree of all vertices bounded by a constant d. We prove that such a graph can be reconstructed within $O(dn)$ non-adaptive queries, which matches the information-theoretic lower bound. The key intermediate result shows that a bipartite graph can be reconstructed in $O(dn)$ non-adaptive queries provided that the degree of vertices *on one side* is bounded by d while no restriction on the other part is made (we call such graphs one-sided bounded degree bipartite graphs). We also show how optimal non-adaptive solutions of $O(n^2/\log n)$ queries for general graphs and of $O(n)$ queries for trees can be obtained.

The results show the power of the considered model, gained by the possibility of testing a *set* of vertices and *counting* the number of edges between them. For comparison, if we are allowed to query only two vertices (that is, test one edge at a time), $\Omega(n^2)$ queries are needed for many natural classes of graphs, such as trees, matchings, Hamiltonian cycles and paths, and some others (see [3]).

The paper is organized as follows. Section 2 is devoted to separating matrices – the main tool for constructing non-adaptive algorithms. In Section 2.1 we consider a general family of separating matrices, those for the class of *bounded weight* vectors, and derive an upper bound for them. In Section 2.2 we concentrate on an important subclass of *d-separating matrices*. Specializing the general result to this class, we obtain a new upper bound that matches modulo a constant factor the information-theoretic lower bound. Another subclass of *d-detecting matrices* is analyzed in Section 2.3. In Section 3 we turn to our main subject of interest – searching for graphs under the additive model. We consider bounded-degree graphs and prove that such graphs can be reconstructed in $O(dn)$ queries. Finally, in Section 4 we consider several particular classes of graphs. For some of them, that are not subclasses of degree bounded graphs, we show that our technique still applies. For others, we show that the constant factor can be improved. For the class of trees of unbounded degree, we propose an optimal construction based on the general result of Section 2.1.

2 Separating Matrices

Consider the following setting. Assume we have a set of *items* and each of them is assigned an integer value. Assume that we want to reconstruct the values by making queries about subsets of items. As noted in the introduction, this type of search problems is very common and is called *combinatorial group testing*. Note that each query can be associated to a $(0,1)$-vector q which is the incidence vector of the corresponding subset. Assume further that the result of a query is the sum of item values of the corresponding subset. This assumption typically corresponds to the *additive model* discussed in the introduction. It implies that if v is the vector of item values, the query result is the scalar product $< q, v >$. Let us now restrict ourselves to non-adaptive algorithms. Then the whole algorithm can be represented by a $(0,1)$-matrix where each row is a query vector and each column corresponds to an item. This leads us to the following notion.

Definition 2.1 *A $k \times n$ $(0,1)$-matrix M is called* separating *for a finite set of integer vectors $\mathcal{V} \subseteq \mathbb{Z}^n$ iff for every $v_1, v_2 \in \mathcal{V}$, $Mv_1 \neq Mv_2$ provided that $v_1 \neq v_2$.*

In general, our goal is to reduce the number of queries, that is to construct a separating matrix, for a given number n of columns, with possibly minimal number k of rows. Clearly, the matrix construction strongly depends on the vector set $\mathcal{V}$. In this section, using the probabilistic method [5, 9] we prove an upper bound for k for a very general class of vectors, namely the set of *bounded weight vectors*. We then show that this estimation gives optimal bounds for some more specific classes of vectors.

2.1 Separating Matrices for the Bounded Weight Vectors

Definition 2.2 *Let $n, d \in \mathbb{N}$. Consider vectors $v = (a_1, \ldots, a_n)$ with non-negative integer components a_i, and denote $weight(v) = \sum_{i=1}^{n} a_i$. The set of* d-bounded weight vectors *is the set of all vectors v with $weight(v) \leq d$.*

Our goal is to obtain an upper bound on the minimal number of rows $k = k(n, d)$ of a separating matrix for the set of d-bounded weight vectors. Note that we don't assume d to be a constant, but an arbitrary integer parameter.

By viewing the columns of a separating matrix as vectors, the problem can be interpreted in the following way. Given n, d, construct a set of n vectors $v_1, \ldots, v_n$ of minimal dimension $k(n, d)$ such that the sum of any multiset of no more than d of these vectors is distinct.

We start by giving some preliminary analysis and establishing an information-theoretic lower bound for $k(n, d)$. There are $\binom{n+i-1}{i}$ vectors of weight i, therefore

$$k(n,d) \geq \log_{(d+1)} \left(\sum_{i=1}^{d} \binom{n+i-1}{i} \right) \geq \log_{(d+1)} \binom{n+d-1}{d} = \tag{1}$$

$$\log_{(d+1)} \left(\frac{n}{n+d} \binom{n+d}{d} \right) \tag{2}$$

Applying $\binom{x}{y} \geq (\frac{x}{y})^y$, we obtain:

$$k(n,d) \geq \log_{(d+1)} \max \left(\left(\frac{n+d}{d} \right)^d, \left(\frac{n+d}{n} \right)^n \right) - \log_{(d+1)} \left(1 + \frac{n}{d} \right) = \tag{3}$$

$$= \log_{(d+1)} \left(1 + \frac{\max(n,d)}{\min(n,d)} \right)^{\min(n,d)} - \log_{(d+1)} \left(1 + \frac{n}{d} \right) \tag{4}$$

A simple calculation shows, that if $d = n^{1+\varepsilon}$, then $k(n, n^{1+\varepsilon}) = \Omega(n)$. Since a linear number of rows is trivially achieved by the identity matrix, we can assume that in all interesting cases $\log d < (1+\varepsilon) \log n$. Thus,

$$k(n,d) \geq \frac{\min(n,d) \log \left(1 + \frac{\max(n,d)}{\min(n,d)} \right)}{(1+\varepsilon) \log \min(n,d)} - \log_{(d+1)} \left(1 + \frac{n}{d} \right) \tag{5}$$

Below we are going to prove that this theoretic–information lower bound is achievable within a constant factor.

We use the probabilistic method [5, 9]. Let M be a random $k \times n$ $(0,1)$-matrix and $\mathcal{V}$ be the set of d-bounded weight vectors. If $Prob\{M \text{ is not separating for } \mathcal{V}\} < 1$, then at least one such separating matrix exists.

For two vectors $v_1, v_2 \in \mathcal{V}$ and a matrix M, we define a characteristic function $\chi(v_1, v_2, M)$:

$$\chi(v_1, v_2, M) = \begin{cases} 1 & \text{if } Mv_1 = Mv_2, \\ 0 & \text{otherwise.} \end{cases}$$

For a vector $v = (a_1, \ldots, a_n)$, let $dom(v)$ denote the set $\{i | a_i > 0\}$. Whenever M is not a separating matrix for $\mathcal{V}$, there exist vectors $v_1 = (a_1, \ldots, a_n), v_2 = (b_1, \ldots, b_n) \in \mathcal{V}$ with $Mv_1 = Mv_2$. Moreover, we can assume that v_1, v_2 verify two additional properties:

1. $dom(v_1) \cap dom(v_2) = \emptyset$. Otherwise, consider (v_1', v_2'), where $v_1' = (a_1', \ldots, a_n'), v_2' = (b_1', \ldots, b_n')$, $a_i' = a_i - min(a_i, b_i)$, $b_i' = b_i - min(a_i, b_i)$. Obviously, $weight(A') \leq weight(A)$, $weight(B') \leq weight(B)$ and $Mv_1' = Mv_2'$ when $Mv_1 = Mv_2$.
2. $weight(v_1) = weight(v_2)$. This can be insured by adding to M an additional row with all entries equal to 1. This row will not be subject to the random choice of matrix' entries. Obviously, adding one row does not affect the asymptotic bound.

A pair of vectors $v_1, v_2 \in \mathcal{V}$ satisfying the two properties above is said to be a *critical* pair. We have

$$Prob\{M \text{ is not separating for } \mathcal{V}\} = Prob\{\bigvee_{(v_1,v_2)} (\chi(v_1, v_2, M) = 1)\}, \tag{6}$$

where disjunction is for all possible critical pairs (v_1, v_2). We estimate this probability from above:

$$Prob\{\bigvee_{(v_1,v_2)} (\chi(v_1, v_2, M) = 1)\} \leq \sum_{(v_1,v_2)} Prob\{\chi(v_1, v_2, M) = 1\}. \tag{7}$$

From now on we assume the uniform distribution over $k \times n$ matrices M. The idea of obtaining an upper bound is to find the smallest k which makes the above sum smaller than 1. The first step is to obtain an upper bound for $Prob\{\chi(v_1, v_2, M) = 1\}$.

Lemma 1 *Given a critical pair* (v_1, v_2),

$$Prob\{\chi(v_1, v_2, M) = 1\} \leq \left(\frac{8}{9 \cdot |dom(v_1)|}\right)^{k/4} \cdot \left(\frac{8}{9 \cdot |dom(v_2)|}\right)^{k/4} \tag{8}$$

Proof: Let $\xi_1, \ldots, \xi_n$ be a set of independent random variables with $Prob\{\xi_i = 0\} = Prob\{\xi_i = 1\} = 1/2$. The event $Mv_1 = Mv_2$ is equivalent to k independent events corresponding to the equality in each row. Therefore,

$$Prob\{Mv_1 = Mv_2\} = Prob\{< s, v_1 >=< s, v_2 >\}^k, \tag{9}$$

where $s = (\xi_1, \ldots, \xi_n)$, and $< s, v_i >$ is the scalar product of s and v_i. Since $dom(v_1) \cap dom(v_2) = \emptyset$, then $< s, v_1 >$ and $< s, v_2 >$ are independent and

$$Prob\{< s, v_1 >=< s, v_2 >\} = \sum_i Prob\{(< s, v_1 >= i) \wedge (< s, v_2 >= i)\} = \tag{10}$$

$$= \sum_i Prob\{< s, v_1 >= i\} \cdot Prob\{< s, v_2 >= i\} \leq \tag{11}$$

$$\leq \sqrt{\sum_i Prob\{< s, v_1 >= i\}^2} \cdot \sqrt{\sum_i Prob\{< s, v_2 >= i\}^2} \tag{12}$$

The sum $\sum_i Prob\{< s, v_j >= i\}^2$, $j = 1, 2$, can be bounded from above by $\max_i Prob\{< s, v_j >= i\}$. Indeed, consider an arbitrary integer-valued random variable ξ and let $p_{max}(\xi) = \max_{i \in \mathbb{Z}} Prob\{\xi = i\}$. Then $\sum_i Prob\{\xi = i\}^2 \leq \sum_i p_{max}(\xi) Prob\{\xi = i\} = p_{max}(\xi) \sum_i Prob\{\xi = i\} = p_{max}(\xi)$. Therefore, we rewrite (12) as

$$Prob\{< s, v_1 >=< s, v_2 >\} \leq \sqrt{p_{max}(< s, v_1 >)} \cdot \sqrt{p_{max}(< s, v_2 >)} \qquad (13)$$

We now need the following technical proposition.

Proposition 1 *Let t be a natural number, $a_1, \ldots, a_t > 0$, and $\xi_1, \ldots, \xi_t$ be independent random variables with $Prob\{\xi_i = 0\} = Prob\{\xi_i = 1\} = 1/2$. Then*

1. $2^{-t}\binom{t}{\lfloor t/2 \rfloor} \leq \sqrt{\frac{8}{9t}}$, *for all* $t \geq 1$,
2. $p_{max}(\xi_1 + \cdots + \xi_t) = 2^{-t}\binom{t}{\lfloor t/2 \rfloor}$,
3. $p_{max}(a_1\xi_1 + \cdots + a_t\xi_t) \leq p_{max}(\xi_1 + \cdots + \xi_t)$.
4. $p_{max}(a_1\xi_1 + \cdots + a_t\xi_t) = p_{max}(\xi_1 + \cdots + \xi_t)$ *iff* $a_1 = a_2 = \cdots = a_t$.

Proof:

1. For big t the inequality easily follows from Stirling formula. The constant was chosen to satisfy the inequality for *all* $t \geq 1$.
2. This is obvious since $Prob\{\xi_1 + \cdots + \xi_t = i\} = 2^{-t}\binom{t}{i}$ and $\binom{t}{\lfloor t/2 \rfloor} \geq \binom{t}{i}$, for all $i = 0, 1, \ldots, t$.
3. Let $P^{\max} = p_{max}(a_1\xi_1 + \cdots + a_t\xi_t)$. There is a value s such that $Prob\{a_1\xi_1 + \cdots + a_t\xi_t = s\} = P^{\max}$. Consider the family $\mathcal{F} = \{A | \sum_{i \in A} a_i = s\}$. Clearly, $card(\mathcal{F}) \cdot 2^{-t} = P^{\max}$. Since $a_i > 0$, $\mathcal{F}$ is a Sperner family of sets, that is there is no two sets A, B such that $A \subset B$. By Sperner's theorem [6], $card(\mathcal{F}) \leq \binom{t}{\lfloor t/2 \rfloor}$.
4. Another consequence of Sperner's theorem is that $card(\mathcal{F}) = \binom{t}{\lfloor t/2 \rfloor}$ iff $\mathcal{F}$ is the family of all subsets of equal size $\lfloor t/2 \rfloor$ or $\lfloor (t+1)/2 \rfloor$. It follows that the sum of $\lfloor t/2 \rfloor$ minimal elements among $a_1, \ldots, a_t$ is equal to the sum of $\lfloor t/2 \rfloor$ maximal elements, so they are all equal.

□

We return to the proof of Lemma 1. To bound $p_{max}(< s, v_j >)$, $j = 1, 2$, we apply the Proposition above with $t = dom(v_j)$. We then have

$$p_{max}(< s, v_j >) \leq 2^{-t}\binom{t}{\lfloor t/2 \rfloor} \leq \left(\frac{8}{9 \cdot |dom(v_j)|}\right)^{1/2}$$

By (9), (13), the Lemma follows. □

We now rewrite the right-hand side of (7) as

$$\sum_{(v_1, v_2)} Prob\{\chi(v_1, v_2, M) = 1\} = \frac{1}{2} \sum_{w=1}^{d} \sum_{\substack{v_1, v_2, \\ |weight(v_1)| = w, \\ |weight(v_2)| = w, \\ dom(v_1) \cap dom(v_2) = \emptyset}} Prob\{\chi(v_1, v_2, M) = 1\} \qquad (14)$$

Using Lemma 1, we bound the inner sum for some fixed w.

$$\sum_{\substack{v_1,v_2,\\ |weight(v_1)|=|weight(v_2)|=w,\\ dom(v_1)\cap dom(v_2)=\emptyset}} Prob\{\chi(v_1,v_2,M)=1\} \leq \tag{15}$$

$$\sum_{\substack{v_1,v_2,\\ dom(v_1)\cap dom(v_2)=\emptyset}} \left(\frac{8}{9\cdot|dom(v_1)|}\right)^{k/4}\cdot\left(\frac{8}{9\cdot|dom(v_2)|}\right)^{k/4} \leq \tag{16}$$

$$\left(\sum_{\substack{v_1,\\ weight(v_1)=w}} \left(\frac{8}{9\cdot|dom(v_1)|}\right)^{k/4}\right)^2 = \left(\sum_{s=1}^{w}\sum_{\substack{v_1,\\ weight(v_1)=w,\\ |dom(v_1)|=s}} \left(\frac{8}{9s}\right)^{k/4}\right)^2 = \tag{17}$$

$$\left(\sum_{s=1}^{w}\binom{n}{s}\binom{w-1}{s-1}\left(\frac{8}{9s}\right)^{k/4}\right)^2 \tag{18}$$

(17) has been obtained by dropping the condition $dom(v_1)\cap dom(v_2)=\emptyset$. To obtain (18), we used the fact that there are $\binom{n}{s}\binom{w-1}{s-1}$ vectors v_1 of weight w with $|dom(v_1)|=s$, which follows from simple combinatorial considerations.

We are now left with the technical problem of finding a possibly minimal k which makes (18) smaller than $\frac{2}{d}$. This will make (14) smaller than 1 and achieve our goal. Finding such k requires some calculations that are left for Appendix C. The following proposition gives the final result.

Proposition 2 *There exist absolute constants C_1, C_2, C_3 such that for all $1 \leq w \leq d$,*

$$\sum_{s=1}^{w}\binom{n}{s}\binom{w-1}{s-1}\left(\frac{8}{9s}\right)^{k/4} < \sqrt{\frac{2}{d}}$$

provided that

$$k(n,d) \leq \frac{4\min(n,d)\log\left(C_1\cdot\frac{\max(n,d)}{\min(n,d)}\right)}{\log\min(n,d)+C_2} + C_3\log d. \tag{19}$$

Putting all together, we state the main result of this section.

Theorem 1 *There exist absolute constants C_1, C_2, C_3 such that for all n, d there exists a $k\times n$ separating matrix for the set of d-bounded weight vectors with $k(n,d)$ bounded as in (19).*

Relating (19) to lower bound (5), we conclude that upper bound (19) is within the factor of $4(1+\varepsilon)$ from the information-theoretic lower bound provided that $d < n^{1+\varepsilon}$ for any $\varepsilon > 0$.

We conclude this section with two special cases of Theorem 1 that will be used in the sequel. The first one corresponds to the case when d is a constant, and the second one to the case $d = d_0 n$ for some constant $d_0 > 1$.

Corollary 1 *For a constant d, there exists a separating matrix for the set of d-bounded weight vectors of dimension n with asymptotically $4\frac{d}{\log d}\log n$ rows.*

Corollary 2 *Assume that $d = d_0 n$ for some constant $d_0 > 1$. Then there exists a separating matrix for the set of d-bounded weight vectors of dimension n with asymptotically $4 \log d_0 \frac{n}{\log n}$ rows.*

2.2 Optimal d-Separating Matrices

Here we concentrate on an important subclass of separating matrices for bounded weight vectors.

Definition 2.3 *For a constant $d \in \mathbb{N}$, a* d-separating matrix *is a separating matrix for the set of $(0,1)$-vectors containing at most d entries equal to 1.*

Equivalently, for a d-separating matrix, all the sums of any up to d columns are distinct. If a matrix has n columns, there will be $\sum_{i=0}^{d} \binom{n}{i}$ different sums of at most d columns. Since each entry of such a sum is at most d, a d-separating matrix has at least $\log_{d+1} \binom{n}{d} = (1 + o(1)) d \log_d(n)$ rows (recall that d is a constant). A better lower bound was proved by Noga Alon [4]. Using the second moment method (cf. [5]), it was shown in [4] that there exists an absolute constant c such that for every $n > d$ any d-separating matrix has at least $\frac{2d}{\log d + c} \log (n/d)$ rows.

By definition, a matrix is 1-separating iff all its columns are different. Clearly, a 1-separating matrix with n columns and $\lceil \log_2 n \rceil$ rows can be easily constructed by setting the columns to be the binary representations of numbers $1, 2, \ldots, n$. This matrix corresponds to the *non-adaptive binary search*, as it provides a non-adaptive analogue to the binary search procedure. For an arbitrary constant d, it is known that a d-separating matrix with asymptotically $d \log_2 n$ rows can be effectively constructed (see [19] and [3, exercise 2.3.5]). For $d = 2$, the lower bound $(5/3) \log_2 n$ has been proved by Lindström [18], while no better upper bound than $2 \log_2 n$ is known. This suggests that settling the multiplicative factor for the case of arbitrary constant d is difficult.

The general result of the previous section allows to improve the upper bound by the factor of $\log_2 d$. Corollary 1 implies immediately the following result.

Theorem 2 (d-separating matrix) *For fixed d, there exists a d-separating matrix with n columns and asymptotically $4 \frac{d}{\log d} n$ rows.*

Thus, we obtain the upper bound which is within the factor two of the lower bound $(2 + o(1)) d \log_d n$ from [4]. This answers the question whether the upper bound $d \log_2 n$ can be improved, posed by Lindström in [19].

Let us note a straightforward connection between d-separating matrices and a classical problem of *counterfeit coins*. A d-separating matrix with n columns solves the following problem. *Suppose we have n coins of which at most d are counterfeit. We are allowed to ask how many counterfeit coins occur in a subset. Find an optimal non-adaptive algorithm that determines all counterfeit coins.*

2.3 d-Detecting Matrices

In this section we consider another important class of separating matrices.

Definition 2.4 *Let d be a constant. A $k \times n$ $(0,1)$-matrix, with n columns, is called* d-detecting *iff it is separating for the set of n-vectors $\{0, \ldots, d-1\}^n$.*

Let $v_1, v_2, \ldots, v_n$ be the columns of a $(0,1)$-matrix. Then this matrix is d-detecting iff all the sums $\sum_{i=1}^{n} \epsilon_i v_i$ $(\epsilon_i = 0, 1, \ldots, d-1)$ are different. Such a set of vectors is called *detecting* in [17], hence our terminology.

Given n and d, we are interested in d-detecting matrices with minimal number of rows. An information-theoretic argument gives the inequality $d^n \leq (dn)^k$, and the lower bound $\Omega(n/(1+\log_d n))$ for k.

Corollary 2 implies the asymptotic bound $4n/\log_d n$ for k. While this already meets the lower bound modulo a constant factor, an effective construction yielding a better upper bound can be given. The problem has been studied by several authors, and particularly by Bernt Lindström in a series of papers [16, 17, 19]. In [17] Lindström presents a construction of a detecting matrix, using the theory of Möbius functions. This construction gives a solution of order $2n/\log_d n$, although this was not explicitly pointed out by the author. Moreover, this bound turns out to be optimal, that will be stated in Lemma 2 below. We refer to Appendix A for a summary of main Lindström's results from [16, 17], and their application to the construction of an optimal d-detecting matrix. Here we summarize this construction in the following theorem.

Theorem 3 *For fixed d, a d-detecting matrix can be effectively constructed with n columns and asymptotically $2n/\log_d n$ rows.*

In [19] Lindström concentrates on the case $d = 2$ for which he proposes a construction of a detecting matrix based on elementary methods (the construction is also described in [3, 8]). The matrix has asymptotically $2n/\log_2 n$ rows. Lindström also proves that the construction is optimal, that is the bound $2n/\log_2 n$ is the asymptotic lower bound. Further references to the case $d = 2$ can be found in [19].

It is interesting that the construction of Theorem 3 matches the asymptotic lower bound for d-detecting matrices.

Lemma 2 *For a fixed d, any d-detecting matrix with n columns has at least $2n/\log_d n$ rows asymptotically.*

The proof is given in Appendix B. It is a generalization of the proof for $d = 2$ [19] based on the method attributed to L. Mozer.

Similarly to d-separating matrices, d-detecting matrices have a natural interpretation in terms of "generalized counterfeit coins problem". *Assume we have n coins and an unknown arbitrary number of them are false. Assume further that we know the weight α of an authentic coin, and that the weight of each false coin takes one of the values $\alpha + \delta i$ for $i = 1, \ldots, d-1$. One can think of i (the overweight of a coin) as the "measure of falsity". We are allowed to weigh subsets of coins and thus measure the overall overweight of a subset. Determine the false coins and their falsity by possibly minimal number of weighing.* It is easily seen that finding a non-adaptive solution of the generalized counterfeit coins problem is directly translated to constructing a d-detecting matrix with minimal number of rows. Note that for $d = 2$ we get the counterfeit coins problem described in Sect. 2.2 but with an arbitrary non-fixed number of false coins.

3 Reconstructing Bounded Degree Graphs

Now we turn to our main subject of interest – the problem of graph reconstruction under the additive model. Let $\mathcal{G}_n$ be a class of undirected graphs on the set V of n

vertices labelled by $\{1, 2, \ldots, n\}$. We consider simple graphs, that is graphs without loops or multiple edges.

We address the following problem. Reconstruct an unknown graph $G = (V, E) \in \mathcal{G}_n$ by means of queries of the following type: For a subset $W \subseteq V$ of vertices, how many edges are there in the intersection $G \cap K_W$, where K_W is the complete graph with the set of vertices $W \subseteq V$? In other words, we want to reconstruct G by means of the *query function* $\mu(W) = |E \cap (W \times W)|$.

Note that a slightly different graph reconstruction model was considered in [1]. In this model, two subsets $W_1, W_2 \subseteq V$ are queried, and the query yields the number of 1's in the submatrix of the adjacency matrix of G induced by rows W_1 and columns W_2. While this model is clearly stronger than ours, below we will show that such a query can be simulated by five queries μ, and therefore both models are equivalent up to a constant factor.

Using the results on separating matrices presented in Section 2, we solve this problem for an important subclass of graphs, namely the *bounded degree graphs*. These are graphs with the degree of vertices bounded by some constant d. Bounded degree graphs are quite common objects and cover such classes as matchings, cycles and paths, trees with bounded branching degree, etc. Property testing for bounded degree graphs was considered in [12]. In this section we propose an asymptotically optimal (modulo a constant factor) predetermined search algorithm for this class.

We first prove an auxiliary result. Using the results of Section 2.2 and 2.3, we prove the existence of an optimal predetermined algorithm for the class of *one-sided bounded degree bipartite graphs*. Consider a bipartite graph $G = (V, W, E)$, where $V \cup W$ is the set of vertices, $V \cap W = \emptyset$, and $E \subseteq V \times W$. For a constant d, G is called a one-sided (d-)bounded degree graph if $deg(v) \leq d$ for every vertex $v \in V$.

Assume that $|V| = |W| = n$. By assuming that every node of V has degree d, it is easy to estimate the number of such graphs from below as $\binom{n}{d}^n = \Omega((n/d)^{dn})$. Since the answer to a query has $nd + 1$ potential values, any search algorithm requires at least $\log_{dn+1}(n/d)^{dn} = dn(1 + o(1))$ queries. We now prove that this lower bound can be met, modulo a constant factor, by a non-adaptive algorithm.

Theorem 4 *For a constant d, there exists a non-adaptive search algorithm for the class of one-sided d-bounded degree bipartite graphs with n vertices on each side, that requires $8dn$ queries asymptotically.*

Proof: Consider a bipartite graph $G = (V, W, E)$, where $V \cup W$ is the set of vertices, $V \cap W = \emptyset$, $|V| = |W| = n$, and $E \subseteq V \times W$ is the set of edges. Assume that $deg(v) \leq d$ for all $v \in V$. By definition, each query is associated with a couple (V', W'), $V' \subset V$, $W' \subset W$, and has the form: how many edges of G are between vertices of V' and W'? (what is $|E \cap (V' \times W')|$?)

For a vertex $v \in V$ and a subset $W' \subseteq W$, denote by $deg_{W'}(v)$ the number of vertices of W' adjacent to v. Note that $deg_{W'}(v) \leq d$ and can be determined by one query.

Fix a vertex $v \in V$. According to Theorem 2, we can find all its adjacent vertices in W using $4d\frac{\log n}{\log d}$ queries (think of adjacent vertices as being "counterfeit", all other vertices in W being "authentic"). Each query asks about $deg_{W'}(v)$ for some subset $W' \subseteq W$. Let $W_1, W_2, \ldots, W_k \subseteq W$ (k asymptotically to $4d\frac{\log n}{\log d}$) be these subsets. Since the algorithm is predetermined, the subsets $W_1, W_2, \ldots, W_k$ are independent of

v. In other words, if for some $v \in V$ we know $deg_{W_i}(v)$ for every W_i, we can reconstruct all the adjacent vertices of v in W.

Now fix some W_i. By Theorem 3, we can find a sequence of subsets $V_1, \ldots, V_l$, with l asymptotically to $2n\frac{\log d}{\log n}$, such that the queries $< (W_i, V_j), j = 1, \ldots, l >$ allow to reconstruct $deg_{W_i}(v)$ for every $v \in V$ (think of $deg_{W_i}(v)$ as "the degree of falsity" of v).

Repeating this algorithm for each W_i, we can determine $deg_{W_i}(v)$ for all $v \in V$ and all W_i via $4d\frac{\log n}{\log d} \cdot 2n\frac{\log d}{\log n} = 8dn$ queries asymptotically. This allows us to reconstruct the adjacent vertices in W of each $v \in V$, that is to reconstruct the whole graph. □

The key argument of the proof is that the algorithm implied by Theorem 2 is non-adaptive, i.e. the sets W_i don't depend on vertices of V. The fact that the algorithm implied by Theorem 3 is also non-adaptive does not affect the complexity bound but insures that the resulting algorithm is completely predetermined too. Specifically, it insures that all the sets V_j are predetermined, and therefore all the queries (W_i, V_j) are mutually independent and can be made in any order.

We now use Theorem 4 to construct a separating set of queries for general bounded degree graphs. We start with computing the information-theoretic lower bound and for that we estimate from below the number of graphs with bounded degree. Instead of counting all such graphs, we will count a subclass of them, and show that their number is already sufficiently big.

Denote by $D(n, d)$ the set of labelled bipartite graphs with n vertices on each side with the degree of each vertex equal to d (d constant). This d-regular graph is a union of d disjoint matchings. Clearly, $|D(n, 0)| = 1$ and $|D(n, 1)| = n!$.

Consider a graph $G \in D(n, d)$. From G we can obtain a graph in $D(n, d+1)$ by adding a matching which doesn't intersect with G. To estimate the number of possible extensions, consider the complement bipartite graph $\bar{G}$ (an edge connecting the sides belongs to $\bar{G}$ iff it does not belong to G). It is an $(n-d)$-regular graph. Since the number of matching in a bipartite graphs is equal to the permanent of the adjacency matrix, from the Van der Waerden conjecture, proved by Egorychev and Falikman (see [7]), it follows that this graph has at least $(n-d)^n \frac{n!}{n^n}$ matchings. Obviously, none of them intersects with G.

On the other hand, consider a graph $G' \in D(n, d+1)$. The number of matchings it contains is bounded from above by $(d+1)^n$ (a better estimation is not important for our purposes). Thus, $|D(n, 0)| = 1$ and $|D(n, d)|(n-d)^n \frac{n!}{n^n} \leq |D(n, d+1)|(d+1)^n$. From this recurrence, $|D(n, d)| \geq \binom{n}{d}^n (\frac{n!}{n^n})^d \geq (n/d)^{dn}(e^{-nd}) = (\frac{n}{ed})^{nd}$. We obtain $\log_{nd+1} |D(n, d)| \geq nd(1 + o(1))$, and thus any search algorithm for $D(n, d)$ requires $\Omega(nd)$ queries. As $D(n/2, d)$ is a subclass of d-bounded degree graphs with n vertices, we conclude that at least $\Omega(\frac{nd}{2})$ queries are needed for this class.

Using theorem 4 we now show that this lower bound can be achieved, modulo a constant factor, by a predetermined algorithm.

Theorem 5 *For a constant d, there exists a predetermined search algorithm for the class of d-bounded degree graphs with n vertices, that requires $24dn$ queries asymptotically.*

Proof: Consider a graph $G = (V, E)$ with $deg(v) \leq d$ for all $v \in V$, and $|V| = n$. Recall that the nodes V are labelled by $\{1, \ldots, n\}$. We associate to G a bipartite graph

$G' = (V', V'', E')$, where $V' = V'' = \{1, \ldots, n\}$, and $(v_1, v_2) \in E'$ iff $(v_1, v_2) \in E$. Note that $deg(v) \leq d$ for every $v \in V' \cup V''$. We want to reconstruct graph G by applying Theorem 4 to graph G'. The query function for graph G' is $\mu'(X, Y) = |E' \cap (X \times Y)|$ for $X \subseteq V'$, $Y \subseteq V''$. Note that μ' coincides with the above-mentioned query function considered in [1]. We now show that a query μ' can be simulated by a constant number of queries μ.

Observe the following properties of μ':

1. $\mu'(X, Y) = \mu'(Y, X)$
2. $\mu'(X, X) = 2\mu(X)$
3. If $X \cap Y = \emptyset$ then $\mu'(X, Y) = \mu(X \cup Y) - \mu(X) - \mu(Y)$
4. If $X_1 \cap X_2 = \emptyset$ then $\mu'(X_1 \cup X_2, Y) = \mu'(X_1, Y) + \mu'(X_2, Y)$ for any $Y \in V''$.

For arbitrary $X_1, X_2 \subseteq V$, let $Y = X_1 \cap X_2$. Then

$$\mu'(X_1, X_2) = \mu'((X_1 \setminus X_2) \cup Y, (X_2 \setminus X_1) \cup Y) = \\ \mu'(X_1 \setminus X_2, X_2 \setminus X_1) + \mu'(X_1 \setminus X_2, Y) + \mu'(X_2 \setminus X_1, Y) + \mu'(Y, Y).$$

Using properties 1-4, we obtain for arbitrary $X \subset V'$, $Y \subset V''$

$$\mu'(X, Y) = \mu((X \setminus Y) \cup (Y \setminus X)) - 2\mu(X \setminus Y) - 2\mu(Y \setminus X) + \mu(X) + \mu(Y) \qquad (20)$$

Thus, one query μ' can simulated by five queries μ. By Theorem 4, graph G', and therefore G, can be reconstructed through $8nd$ queries $\mu'(W_i, V_j)$. The number of corresponding queries μ can be optimized, if queries $\mu(W_i)$, $\mu(V_j)$ are computed once. This gives us $24nd + 4d \log_d n + 2n/\log_d n = 24nd(1 + o(1))$ queries μ. □

4 Case Studies

Here we consider some particular classes of graphs. For some of them, that are not subclasses of bounded degree graphs, we show that our technique still applies. For others, we show that the multiplicative factor can be improved. Finally, for trees, we propose a new construction based on the general result of Section 2.1.

4.1 General and c-Colorable Graphs

The results above assume some knowledge about the class that the unknown graph is drawn from. What can be said when no prior information about the structure of the graph is given, i.e. all graphs are possible? The information-theoretic lower bound for this case is immediate. There are $2^{\frac{n(n-1)}{2}}$ labelled graphs with n vertices and each query can yield up to $1 + n(n-1)/2$ answers. Therefore, any algorithm should make at least $\log_{(1+n(n-1)/2)} 2^{\frac{n(n-1)}{2}} = \Omega(\frac{n^2}{4 \log_2 n})$ queries. Note that since graphs can be represented by $(0,1)$-vectors of length $n(n-1)/2$, any non-adaptive algorithm for reconstructing general graphs gives a 2-detecting matrix with $n(n-1)/2$ columns. By Lemma 2, we can then obtain a better lower bound of $2 \cdot \frac{n(n-1)}{2} / \log_2 \frac{n(n-1)}{2} = \Omega(\frac{n^2}{2 \log_2 n})$ for non-adaptive algorithms for reconstructing general graphs. This lower bound can be

achieved, within a factor of 4, using the technique of the proof of Theorem 5. Represent $G = (V, E)$ as a bipartite graph $G' = (V_1, V_2, E')$ where each side consists of a copy of vertices of V ($V_1 = V_2 = V$) and $(v_1, v_2) \in E'$ iff $(v_1, v_2) \in E$. For each vertex $v_1 \in V_1$, we can find all its adjacent vertices in V_2 (and then in G) through $\frac{2n}{\log_2 n}$ queries of the form "How many adjacent vertices does v_1 have in a subset $W \subseteq V_2$?". Every such query can be simulated by two queries to the initial graph G. Similar to the proof of Theorem 5, let μ (resp. μ') denote the query function for graph G (resp. G'). Then $\mu'(\{v_1\}, W) = \mu(W \cup \{v_1\}) - \mu(W \setminus \{v_1\})$. To reconstruct the graph, we find for every vertex i the adjacent vertices among $1, \ldots, i-1$. Then the overall complexity is $2 \cdot \sum_{i=2}^{n} \frac{2i}{\log_2 i} = \frac{2n^2}{\log_2 n}(1 + o(1))$ which is 4 times the lower bound.

Note that the same upper bound $\frac{2n^2}{\log_2 n}$ was obtained in [1]. However, their model is stronger than ours as was mentioned earlier in this Section.

Does the knowledge of the graph's chromatic number $c = \chi(G)$ help? Not much, as there are at least $2^{\frac{c-1}{2c}n^2}$ such graphs (divide n vertices into c parts evenly and consider all possible edge combinations between different parts). The information-theoretic lower bound is then $\Omega(\frac{c-1}{4c} \cdot \frac{n^2}{\log n})$, and the algorithm above for the general case is again optimal up to a constant factor.

4.2 h-Edge Colorable Graphs

If the graph is known to be h-edge-colorable, the degree of vertices is bounded by h and by Theorem 5, it can be reconstructed within $O(hn)$ non-adaptive queries. Note that this is asymptotically best possible, as by Vizing's theorem (see [7]), the graphs with the edge chromatic number less than or equal to h contain all the $(h-1)$-bounded degree graphs.

4.3 Matchings in Bipartite Graphs

Matchings occur in numerous applications and we consider important to present a refinement of the general technique that can be obtained for this class. This refinement is valid for a more general class, namely the 1-bounded one-sided bipartite graphs (see Section 3). Let $G = (V, W, E)$ be a bipartite graph, where $|V| = n$, $|W| = m$ and all vertices in V have degree at most 1. According to the proof of Theorem 4, to reconstruct G we need a 1-separating matrix for m objects and 2-detecting matrix for n objects. A 1-separating matrix with m columns has $\log_2 m$ rows (see Section 2.2). As for 2-detecting matrix with n columns, $\frac{2n}{\log_2 n}(1 + o(1))$ rows are necessary and sufficient, as it was mentioned in Section 2.3. Putting together, G can be reconstructed by $\log_2(m)\frac{2n}{\log_2 n}(1 + o(1))$ non-adaptive queries. Note that the probabilistic proof of Theorem 2 is not involved here, and the queries can be constructed explicitly.

In the case of matchings in bipartite graphs we have $m = n$ which gives a non-adaptive algorithm to reconstruct a matching within $2n(1 + o(1))$ queries. This bound is asymptotically optimal within a factor of 2.

4.4 Hamiltonian Cycles and Paths

Let us consider the 2-bounded degree graphs with n vertices. Any such graph is a disjoint collection of paths and cycles. In particular, this class contains the Hamiltonian

cycles and the Hamiltonian paths. The problem of reconstructing Hamiltonian cycles under different models was considered in [13] in connection with its application to genome physical mapping.

Theorem 5 suggests a non-adaptive solution that requires $48n$ queries asymptotically. A better performance can be obtained if we sacrifice the requirement for the algorithm to be fully non-adaptive. Instead, we propose a two-stage algorithm – the first stage does an adaptive "pre-processing" and the second stage reconstructs the graph non-adaptively.

At the first stage, we sort out the vertices into three disjoint independent subsets, that is without adjacent pairs in each subset. As each vertex has at most two adjacent vertices, this sorting can be easily done in at most $2n$ queries by processing the vertices consecutively and testing each vertex against at most two of the already formed subsets.

At the second stage, we reconstruct separately each of the three bipartite 2-degree bounded graphs resulting from the first stage. Again, applying the proof of Theorem 5, we need a 2-separating and a 3-detecting matrices. As noted in Section 2.2, a 2-separating matrix with n columns and $2\log_2 n$ rows can be effectively constructed. On the other hand, by adapting the proof of Theorem 3 to the case $d = 3$, it can be shown that there exists a 3-detecting matrix with n columns and $\frac{4n}{\log_2 n}$ rows. By applying the algorithm of reconstructing bipartite graphs in such a way that the detecting matrix always acts on a smaller part (see proof of Theorem 4), we can reconstruct each bipartite graph in $2\log_2 n \cdot \frac{4n/2}{\log_2 n/2} = 4n$ queries.

Putting two stages together, this gives an algorithm with $2n + 3 \cdot 4n = 14n$ queries.

4.5 Trees

Trees are quite different from previous graph classes in that they are not bounded degree, and the technique of Section 3 does not apply. Here we use the general construction of Section 2.1.

By Cayley theorem, there are n^{n-2} labelled trees and since the number of edges inside a subset of vertices ranges in $0, \ldots, n-1$, the information-theoretic argument gives a simple lower bound $\log_n(n^{n-2}) = n-2 = \Omega(n)$. Below we prove that this bound can be reached asymptotically by a non-adaptive algorithm.

To construct a separating matrix for trees we use the following well-known facts:

- A forest with n vertices has at most $n-1$ edges,
- for a forest on vertices $\{v_1, \ldots, v_n\}$, $\sum_{i=1}^{n}(deg(v_i)) \leq 2n-2$,
- in a non-empty forest at least one vertex has degree 1.

Consider the following two matrices. M_1 is a 1-separating matrix (matrix of non-adaptive binary search, as described in Section 2.2) with n columns, augmented by an additional row, say the first one, consisting of all 1's. M_2 is a separating matrix, with n columns, for the set of $(2n-2)$-bounded weight vectors (see Section 2.1). Thus, M_1 has $\lceil \log_2 n \rceil + 1$ rows and M_2 has $O(\frac{n}{\log n})$ rows by Theorem 1.

We view the rows of both matrices as queries, that is subsets of vertices $\{1, \ldots, n\}$. Let subsets $< Q_i^1 >$ correspond to the rows of M_1 and $< Q_j^2 >$ correspond to the rows of M_2.

Consider a tree $T = (V, E)$, $V = \{1, \ldots, n\}$. As in the proof of Theorem 5, let $\mu'(W_1, W_2)$ be the number of edges of T with one endpoint in W_1 and another in W_2

(edges with both endpoints in $W_1 \cap W_2$ count twice). According to (20), query μ' can be simulated by a constant number of queries μ of our basic additive model.

Theorem 6 *T is uniquely determined by the numbers $\mu'(Q_i^1, Q_j^2)$.*

Proof: As in the proof of Theorem 5, denote by $deg_W(v)$ the number of vertices of W adjacent to v. Let $a_{i,j} = \mu'(Q_i^1, Q_j^2)$. By definition of matrix M_2, we can reconstruct, for each Q_i^1, $deg_{Q_i^1}(v)$ for every $v \in Q_i^1$. (Think of the degree of each vertex in the subtree induced by Q_i^1 as its "weight", the "weight" of the vertices $V \setminus Q_i^1$ being zero. Note that the weight of the whole vector is no more than $2n - 2$.) In particular, we know the degree of each vertex $v \in V$ in the whole tree, since $Q_1^1 = V$ (due to the additional row in M_1). Choose a vertex $v^* \in V$ of degree 1. Since we know $deg_{Q_i^1}(v^*)$ for each Q_i^1, we can determine a unique vertex $w^* \in V$ adjacent to v^* using the definition of matrix M_1. Now we "forget" the edge (v^*, w^*) by properly updating $a_{i,j}$'s. Formally, $a_{i,j}$ is decremented by 1 if $v^* \in Q_i^1, w^* \in Q_j^2$. Then $a_{i,j}$ is decremented by 1 if $v^* \in Q_j^2, w^* \in Q_i^1$. Iterating this process $n - 1$ times we end up with the empty tree which can be easily recognized, as both matrices contain a row of all 1's. By that time we determined all the edges of T. This shows that T is uniquely determined by $a_{i,j}$'s. □

Since we made $O(\log n) \cdot O(\frac{n}{\log n}) = O(n)$ queries overall, we conclude with the following theorem.

Theorem 7 *There exists a non-adaptive algorithm reconstructing a tree on n vertices within $O(n)$ queries.*

5 Remarks

An interesting open question is to give an explicit construction of separating matrices the existence of which has been proved in Theorem 1 by a probabilistic method. An effective construction of d-separating matrices with $O(d \log_d n)$ rows would be of special interest. An explicit construction could open a way to the study of computational complexity of reconstructing the unknown graph given a vector of answers. This question is another direction for future research.

References

[1] Alok Aggarwal, Don Coppersmith, and Dan Kleitman. A generalized model for understanding evasiveness. *Information Processing Letters*, 30:205–208, 1989.

[2] Martin Aigner. Search problems on graphs. *Discrete Applied Mathematics*, 14:215–230, 1986.

[3] Martin Aigner. *Combinatorial Search.* John Wiley & Sons, 1988.

[4] Noga Alon. Separating matrices. private communication, May 1997.

[5] Noga Alon and Joel H. Spencer. *The Probabilistic Method. With an appendix on open problems by Paul Erdös.* Wiley-Interscience Series in Discrete Mathematics and Optimization. New York: John Wiley & Sons, 1992.

[6] Ian Anderson. *Combinatorics of Finite Sets*. Clarendon Press, 1989.

[7] Peter J. Cameron. *Combinatorics: Topics, Techniques, Algorithms*. Cambridge University Press, 1994.

[8] Ding-Zhu Du and Frank K. Hwang. *Combinatorial Group Testing and its applications*, volume 3 of *Series on applied mathematics*. World Scientific, 1993.

[9] Paul Erdös and Joel Spencer. *Probabilistic Methods in Combinatorics*. 1974.

[10] L. Gargano, V. Montuori, G. Setaro, and U. Vaccaro. An improved algorithm for quantitive group testing. *Discrete Applied Mathematics*, 36:299–306, 1992.

[11] Oded Goldreich, Shafi Goldwasser, and Dana Ron. Property testing and its connection to learning and approximation. In *Proceedings of the 37th Annual Symposium on Foundations of Computer Science*, pages 339–348, 1996.

[12] Oded Goldreich and Dana Ron. Property testing in bounded degree graphs. In *The 29th Annual ACM Symposium on Theory of Computing*, 1997. to appear. Full version available at `http://theory.lcs.mit.edu/~danar/papers.html`.

[13] Vladimir Grebinski and Gregory Kucherov. Optimal query bounds for reconstructing a hamiltonian cycle in complete graphs. In *Proceedings of the 5th Israeli Symposium on Theory of Computing and Systems*, pages 166–173. IEEE Press, 1997.

[14] Fred H. Hao. The optimal procedures for quantitive group testing. *Discrete Applied Mathematics*, 26:79–86, 1990.

[15] V. Koubek and J. Rajlich. Combinatorics of separation by binary matrices. *Discrete Mathematics*, 57:203–208, 1985.

[16] Bernt Lindström. On a combinatorial problem in number theory. *Canad. Math. Bull.*, 8:477–490, 1965.

[17] Bernt Lindström. On Möbius functions and a problem in combinatorial number theory. *Canad. Math. Bull.*, 14(4):513–516, 1971.

[18] Bernt Lindström. On B_2-sequences of vectors. *Journal of Number Theory*, 4:261–265, 1972.

[19] Bernt Lindström. Determining subsets by unramified experiments. In J.N. Srivastava, editor, *A Survey of Statistical Designs and Linear Models*, pages 407–418. North Holland, Amsterdam, 1975.

[20] R.L. Rivest and J. Vuillemin. On reconstructing graph properties from adjacency matrices. *Theoretical Computer Science*, 3:371–384, 1976.

Appendix A

In this section we give main ideas of an explicit construction of d-detecting matrix presented in a series of works of B. Lindström, especially in [17].

Consider the columns of a d-detecting matrix as a set of vectors. Their property can be rephrased as follows.

Definition 5.1 *A set of* $(0,1)$ *vectors* $\vec{v}_1, \vec{v}_2, \ldots, \vec{v}_n$ *is said to be* d-detecting *iff all the sums* $\sum_{i=1}^{n} \epsilon_i \vec{v}_i$ $(\epsilon_i = 0, 1, \ldots, d-1)$ *are different.* □

Given n and d, we are interested in a set of d-detecting vectors of *minimal* dimension. Below we outline the asymptotically optimal construction given by Linsdtröm in [17]. The proofs can be found in the original paper.

To introduce the Lindström's construction we recall the definition of Möbius function for partially ordered sets.

Definition 5.2 (Möbius function) *Let $(P, \leq)$ be a finite partially ordered set. The Möbius function $\mu(x, y)$ of P is defined for $x, y \in P$ as follows.*

1. $\mu(x, x) = 1$
2. *if* $x \not\leqslant y$ *then* $\mu(x, y) = 0$
3. *if* $x < y$ *then* $\mu(x, y) = -\sum_{x<z\leq y} \mu(z, y)$

For example, it is known that if P is the Boolean algebra of all subsets of a finite set then

$$\mu(x, y) = (-1)^{|y \setminus x|} \qquad \text{if } x \subset y$$

We will use this fact later. Lindström proved the following results:

Theorem 8 ([17]) Let P be a finite partially ordered set with 0 and a unique last element 1. Let $\mu(x, y)$ be the Möbius function of P. Set $m = \sum_{x\in P} |\mu(x, 1)|$. m is then an even integer. Let n be an arbitrary integer in the interval $0 \leq n \leq m/2$. Then there exists a function $f(x) \in \{0, 1\}$ on P such that

$$\sum_{0<x\leq 1} f(x)\mu(x, 1) = -n \cdot sign(\mu(0, 1)),$$

where $sign(a) = 1$ if $a \geq 0$ and $sign(a) = -1$ if $a < 0$.

Theorem 9 ([16]) Let P be a finite semilattice with Möbius function $\mu(x, y)$. Let $a, b \in P$ and $b \not\leqslant a$. Let $f(x)$ be defined for all $x \leq a \wedge b$ with values in a commutative ring with unit. Then we have

$$\sum_{x\leq b} f(x \wedge a)\mu(x, b) = 0 \tag{21}$$

Before we give the construction of a d-detecting vector set, we introduce the *detecting capacity* $h_k(x)$.

Definition 5.3 *The* detecting capacity $h_k(x)$ *is the maximum number h for which there exist integers d_i $(i = 1 \ldots h)$, $1 \leq d_i \leq x$, such that all the sums $\sum_{i=1}^{h} \epsilon_i d_i$ $(\epsilon_i = 0 \ldots k - 1)$ are distinct.*

We call a vector $(d_1, \ldots d_h)^T$ on which the maximum is reached a *detecting vector*.

The following theorem is the key-stone of the construction of optimal d-detecting vector set.

Theorem 10 ([17]) *Let $(P, \wedge)$ be a finite semilattice with $m + 1$ elements. Define a partial order on P such that $a \leq b$ iff $a = a \wedge b$. Let θ be the least element in $(P, \leq)$. Put $m_y = \sum_{x\leq y} |\mu(x, y)|$. Then there exists a d-detecting set containing $\sum_{y>\theta} h_d(m_y/2)$ vectors of dimension m.* □

The construction of the d-detecting vector set has the following stages [17]:

1. Given a semilattice $P = (x_1, x_2, \ldots, x_m)$ with m elements we consider its multiplication table, an $m \times m$ matrix $(a_{i,j})$, where $a_{i,j} = x_i \wedge x_j$.
2. Consecutively consider each column of the above matrix. Replace each entry of this column by a row-vector of dimension $h_d(m_{x_i}/2)$. Note that the column marked with x_i contains all $x_j \leq x_i$. By theorem 8, for all $k \leq m_{x_i}/2$, there is a function $f_k(x) : P \to \{0,1\}$ such that $\sum_{\theta < x \leq x_i} f_k(x)\mu(x, x_i) = -k \cdot sign(\mu(\theta, x_i))$.
3. For each column x_i, find $h_d(m_{x_i}/2)$ and a corresponding d-detecting vector $(d_1, d_2, \ldots, d_h)$. Then replace the entry at each row x_j by the row-vector $(f_{d_1}(x_i \wedge x_j), f_{d_2}(x_i \wedge x_j), \ldots, f_{d_h}(x_i \wedge x_j))$.

To prove that the columns of the obtained matrix form a d-detecting set assume that

$$\sum_{1 \leq i \leq m, 1 \leq j \leq h_d(m_{x_i}/2)} e_{i,j}\vec{v}_{i,j} = \vec{0}, \quad \text{where } e_{i,j} = -k, \ldots, 0, \ldots, k, \tag{22}$$

We prove (following [17]) that all $e_{i,j} = 0$. If it is not true, then there exists a maximal x_i in $(P, \leq)$ such that $e_{i,j} \neq 0$ for some j. Multiply the v-th row of both side of (22) by $-\mu(x_v, x_i) \cdot sign(\mu(\theta, x_i))$ and sum up all the rows. It follows from (21) that columns corresponding to $y \not\geqslant x_i$ sum up to zero. From (22) we further get

$$\sum_{j=1}^{h} e_{i,j}d_{i,j} = 0,$$

where $h = h_d(m_i/2)$. If some $e_{i,j} \neq 0$ we get a contradiction to the fact that $\{d_{i,j}\}_{j=1}^{h}$ is detecting.

Now we apply the above construction in order to obtain an upper bound of a d-detecting set of n vectors.

1. Consider the Boolean lattice of subsets of n-element set. So that if $x \subset y$ then $\mu(x, y) = (-1)^{|y \setminus x|}$. It follows that $m_x = 2^{|x|}$.
2. Consider a prefix of the sequence $(1, d, d^2, \ldots, d^i, \ldots)$. Clearly, this is a d-detecting vector.
3. Now we have $h_d(m) = \lfloor \log_d m \rfloor$, and the detecting capacity of element x is

$$h_d(2^{|x|}/2) = \left\lfloor \log_d 2^{|x|-1} \right\rfloor = \left\lfloor \frac{|x| - 1}{\log_2(d)} \right\rfloor \geq \frac{|x|}{\log_2(d)} - 2.$$

4. This gives a matrix with

$$\sum_{x \subset [n]} \left(\frac{|x|}{\log_2 d} - 2 \right) = \sum_{i=0}^{n} \binom{n}{i} \left(\frac{i}{\log_2 d} - 2 \right) = \frac{2^n n}{2 \log_2 d} - 2^{n+1}$$

columns.

It follows that there exists a d-detecting set of approximately $\frac{2^n n}{2 \log_2 d}$ vectors of dimension 2^n. We conclude that for large n, there exists a d-detecting set of n vectors of dimension $2 \log(d) \frac{n}{\log n}$ asymptotically.

Appendix B

Here we give a proof of Lemma 2 that an optimal d-detecting matrix has at least $2\log d\frac{n}{\log n}$ rows asymptotically. The proof is a modification of L.Moser's method described in [19, p. 415].

Let the $\{\vec{v}_1, \vec{v}_2, \ldots, \vec{v}_n\}$ be the columns of a d-detecting matrix. We show that these vectors must have dimension at least $2\log d\frac{n}{\log n}$ asymptotically. Let m be the dimension of $\vec{v}_i$'s. The idea of the proof is to show that at least half of the vectors of the set $\{\epsilon_1\vec{v}_1 + \epsilon_2\vec{v}_2 + \cdots + \epsilon_n\vec{v}_n | \epsilon_i = 0, \ldots, d-1\}$ belongs to an m-dimensional sphere of a "small" radius. This gives an estimation for m.

Consider the uniform probabilistic space $\{(\epsilon_1, \epsilon_2, \ldots, \epsilon_n) | \epsilon_i = 0, \ldots, d-1\} \cong [0, \ldots, d-1]^n$. The random variable $\xi = \epsilon_1$ has the expectation $E(\xi) = (d-1)/2$ and the variance $\sigma^2 = Var(\xi) = \frac{(d+2)d}{12}$. Denote by v_i^j the value of j-th coordinate of $\vec{v}_i$. Then the random variable $\varsigma_k = \epsilon_1 v_1^k + \cdots + \epsilon_n v_n^k$ $(k = 1 \ldots m)$ is a sum of independent random variables ϵ_i with coefficients $0, 1$. It means that

$$Var(\varsigma_k) \leq \sum_{i=1}^{n} Var(\epsilon_i) = n\sigma^2.$$

By definition of variance and the linearity of expectation, we have:

$$E\left(\sum_{k=1}^{m}(\varsigma_k - \bar{\varsigma}_k)^2\right) = \sum_{k=1}^{m} E(\varsigma_k - \bar{\varsigma}_k)^2 \leq m \cdot n\sigma^2.$$

It follows from the Chebyshev inequality,

$$Prob\left(\sum_{k=1}^{m}(\varsigma_k - \bar{\varsigma}_k)^2 \leq 2mn\sigma^2\right) \geq 1/2.$$

Since $(\epsilon_1, \ldots, \epsilon_n) \to \epsilon_1\vec{v}_1 + \cdots + \epsilon_n\vec{v}_n = (\varsigma_1, \ldots, \varsigma_m)^T$ is a bijection, at least a half of these sums belong to a sphere with center $(\bar{\varsigma}_1, \ldots, \bar{\varsigma}_m)$ and radius $r = (2mn\sigma^2)^{1/2}$. By the volume argument, the number of integer-valued points in a sphere with radius r is less than $(c/m)^{m/2} r^m$ for a constant c. Therefore, $1/2 \cdot d^n \leq (c/m \cdot r^2)^{m/2} = (c/m \cdot 2mn\sigma^2)^{m/2} = (2cn\sigma^2)^{m/2}$. It follows that $m \geq 2 \cdot \frac{n\log d - \log(2)}{\log n + \log(2 \cdot c \cdot (d+2) \cdot d/12)} = 2\log d\frac{n}{\log(n)} + o(\frac{n}{\log n})$. $\square$

6 Appendix C

Here we prove Proposition 2 from Section 2.1.

Proposition 2 There exist absolute constants C_1, C_2, C_3 such that for all $1 \leq w \leq d$,

$$\sum_{s=1}^{w} \binom{n}{s}\binom{w-1}{s-1}\left(\frac{8}{9s}\right)^{k/4} < \sqrt{\frac{2}{d}} \tag{23}$$

provided that

$$k(n,d) \leq \frac{4\min(n,d)\log(C_1 \cdot \frac{\max(n,d)}{\min(n,d)})}{\log\min(n,d) + C_2} + C_3\log d. \tag{24}$$

Sum (23) is increasing on w, so it suffices to consider only the case $w = d$. We further increase the sum by estimating $\binom{w-1}{s-1} \leq \binom{w}{s}$. Thus, we have

$$\sum_{s=1}^{w} \binom{n}{s}\binom{w-1}{s-1}\left(\frac{8}{9s}\right)^{k/4} \leq \sum_{s=1}^{d} \binom{n}{s}\binom{d}{s}\left(\frac{8}{9s}\right)^{k/4} \tag{25}$$

The binomial coefficients impose that $s \leq n$ and $s \leq d$. Define $a = \max(n, d)$, $b = \min(n, d)$. We have to find k such that

$$\sum_{s=1}^{b} \binom{a}{s}\binom{b}{s}\left(\frac{8}{9s}\right)^{k/4} < \sqrt{\frac{2}{d}}.$$

Since $b \leq d$, the inequality can be further strengthened:

$$\forall s,\ 1 \leq s \leq b \quad \binom{a}{s}\binom{b}{s}\left(\frac{8}{9s}\right)^{k/4} < \frac{\sqrt{2}}{d\sqrt{d}}$$

In other words,

$$\forall s,\ 1 \leq s \leq b \quad k \geq 4\log_{9s/8}\left(\binom{a}{s}\binom{b}{s}\frac{d\sqrt{d}}{\sqrt{2}}\right)$$

It follows that any $k \geq 4\max_s \log_{9s/8}\left(\binom{a}{s}\binom{b}{s}\frac{d\sqrt{d}}{\sqrt{2}}\right)$ will be sufficient. This maximum can be estimated from above as

$$\max_s \log_{9s/8}\left(\binom{a}{s}\binom{b}{s}\frac{d\sqrt{d}}{\sqrt{2}}\right) \leq \max_{1\leq s\leq b} \log_{9s/8}\binom{a}{s} + \max_{1\leq s\leq b} \log_{9s/8}\binom{b}{s} + \log_{9/8}\frac{d\sqrt{d}}{\sqrt{2}}.$$

Note that $\max_{1\leq s\leq b} \log_{9s/8}\binom{b}{s}$ is reached at $s^* \leq b/2$, as this function is decreasing when $s > b/2$. Applying it together with inequality $\binom{x}{y} \leq (\frac{ex}{y})^y$ we have:

$$\max_{1\leq s\leq b} \log_{9s/8}\binom{b}{s} + \max_{1\leq s\leq b} \log_{9s/8}\binom{a}{s} \leq \max_{1\leq s\leq b/2} \log_{9s/8}\left(\frac{eb}{s}\right)^s + \max_{1\leq s\leq \min(b,a/2)} \log_{9s/8}\left(\frac{ea}{s}\right)^s$$

A simple analysis of the derivative shows that maximum is reached at $s = b/2$ and $s = \min(b, a/2)$ respectively. Finally, we obtain

$$\frac{k}{4} = \begin{cases} \frac{b(\log(2)+1)}{2(2\log 3-4\log 2+\log b)} + \frac{a(\log(2)+1)}{2(2\log 3-4\log 2+\log a)} + \frac{3\log d-\log 2}{2(2\log 3-3\log 2)} & \text{if } b \leq a \leq 2b \\ \frac{b(\log(2)+1)}{2(2\log 3-4\log 2+\log b)} + \frac{b(\log a-\log b+1)}{\log b+2\log 3-3\log 2} + \frac{3\log d-\log 2}{2(2\log 3-3\log 2)} & \text{if } a \geq 2b \end{cases}$$

Two cases can be merged:

$$k \leq 4\frac{b(\log\frac{a}{b} + \frac{3}{2}(\log 2 + 1))}{2\log 3 - 4\log 2 + \log b} + 52\log d$$

□