Combinatorial search on graphs motivated by bioinformatics applications: a brief survey

Mathilde Bouvel[1], Vladimir Grebinski[2], and Gregory Kucherov[3]

[1] Département d'Informatique, Ecole Normale Supérieure de Cachan, 94235, France
[2] CompuGene Inc., Jamesburg, NJ 08831, USA
[3] INRIA/LORIA, 615, rue du Jardin Botanique, B.P. 101, 54602, Villers-lès-Nancy, France, Gregory.Kucherov@loria.fr

Abstract. The goal of this paper is to present a brief survey of a collection of methods and results from the area of combinatorial search [1,8] focusing on graph reconstruction using queries of different type. The study is motivated by applications to genome sequencing.

1 Introduction

1.1 Generic problem and bioinformatics application

Assume we have a set of labeled chemicals and some pairs of chemicals can react. Assume we have an experimental tool to detect if a reaction occurs when mixing two or several chemicals together, or a tool that allows us to count how many reacting pairs there are in the mixture. Our goal is to recover all pairs of reacting chemicals with as few experiments as possible.

One important application area for such problems is bioinformatics. For example, obtaining a whole genomic sequence is a crucial first step in the study of an organism. A common practical approach to genome sequencing is to obtain a number of short and possibly overlapping *reads* from the genomic sequence, that are then assembled into *contigs* – contiguous fragments that cover the genome with possible gaps. The problem is then to determine the relative placement of contigs on the genome, i.e. to reconstruct their original order. This step is a accomplished by testing the adjacency of contigs using a so-called *Polymerase Chain Reaction* (PCR). Nowadays, PCR is one of the most ubiquitous tools in molecular biology and can be performed very cheaply, efficiently and almost automatically (see e.g. [2]). It is based on the idea that any region of the genome can be described by a pair of *primers* that can be thought of as short nucleotide sequences bounding this region. If the primers are proximate (within several thousands of nucleotides in practice), the region that they delimit is amplified into a huge number of copies, which can be observed experimentally. Therefore, by picking primer sequences from both ends of each contig, we can reliably test if they are adjacent on the original DNA, under the assumption that the gaps between contigs are of bounded size.

While the basic PCR allows one to test one pair of primers at a time, the *multiplex PCR* presents an extension that uses several primers simultaneously

to determine amplified regions. Since several regions can be amplified simultaneously, this approach can also provide an information of *how many* pairs of primers resulted in an amplification.

In all cases, a very important question in practice is how many reactions are needed in the worst case and how quickly we can perform all of them. Ideally, we want to implement as few reactions as possible and run them in parallel. In this paper we survey some of the results related to such and similar problems.

1.2 Mathematical formulation and main definitions

If chemicals are represented as vertices of a non-oriented graph and a reaction as an edge, we come up with a problem of reconstructing an unknown graph of a given class of graphs. Note that we might also consider that a reaction is triggered by more than two chemicals, which would result in a hypergraph reconstruction problem.

The multiplex PCR problem can lead to two different mathematical formalizations. If the objects ("chemicals") we are dealing with are contigs (i.e. primers coming from both ends of a contig are always tested together), the underlying problem is to reconstruct a Hamiltonian path or a Hamiltonian cycle[4] on K_n (the complete graph with n vertices, where n is the number of *contigs*) [11]. If we are dealing with primers, we face the problem of reconstructing a matching on K_n (where n is the number of *primers*).

Graph reconstruction problem. Different kinds of combinatorial search problems on graphs have been considered in the literature (see [1]): identifying an unknown edge or vertex in a given graph, reconstructing a hidden graph of a given class, verifying a property of a hidden graph, and some others. Our interest here will be the following graph reconstruction problem:

Problem 1 *Given a class of graphs $\mathcal{G} = \cup_n \mathcal{G}_n$, where $\mathcal{G}_n$ contains all the graphs of $\mathcal{G}$ on the set of vertices $V = \{1, \ldots, n\}$, we want to reconstruct a hidden graph $G \in \mathcal{G}_n$ for a given n, making as few queries as possible. A query is a subset of V, and the answer we obtain provides us with information about the edges in the subgraph of G induced by the queried subset. This information depends on the model under consideration.*

In the particular case when only two vertices of V can be tested at a time, the query just checks if a specific edge exists in G, and the model is called a *two-vertex model*.

Boolean and quantitative models. One type of query is: "For $Q \subseteq V$, is there at least one edge in the subgraph of G induced by Q?". The possible answers being `true` or `false`, this query model is called *boolean*.

A natural extension of this model admits queries of the following form: "For $Q \subseteq V$, *how many* edges does the subgraph of G induced by Q contain?". This

[4] depending on whether the genome is linear or circular

query model is called *quantitative* (or *additive*) since the answer to a query is an integer ranging between 0 and the number of edges of a G.

In both cases, the *complexity* of a problem is defined as the minimum number of queries required to reconstruct a graph of $\mathcal{G}_n$ in the worst case. The complexity depends on n but can also made dependent on other parameters (see [4] for example).

We will be generally interested in finding upper and lower bounds on the complexity of a problem. The information theory provides a simple and powerful method to estimate the lower bound: at least $\log_d |\mathcal{G}_n|$ queries must be made in order to identify a graph from $\mathcal{G}_n$, where d is the maximal number of distinct answers provided by a query.

Adaptive and nonadaptive algorithms: Two main kinds of algorithms must be distinguished in the area of combinatorial search: in *adaptive algorithms*, every query potentially depends on the answers obtained to previous queries while in *nonadaptive algorithms*, all queries are independent of each other. A nonadaptive algorithm can be described as a family of subsets of V (queries) or as an vertex-query incidence matrix M ($M_{i,j} = 1$ iff vertex j appears in query i, $M_{i,j} = 0$ otherwise).

Nonadaptive algorithms can be seen as 1-*round algorithms*, i.e. those in which all queries can be made in parallel. From this perspective, adaptive algorithms are multi-round (have an unlimited number of rounds). Intermediate case of s-*round algorithms* composed of s successive nonadaptive stages will also be considered.

In this paper, we present a short survey of different known results on graph reconstruction. From the application perspective, our main motivation is on reconstructing Hamiltonian cycles but we also consider other graph classes such as matchings, stars, cliques, graphs with bounded vertex degree, and others. Two main query models will be considered: the boolean model (Section 2) and the quantitative model (Section 3). For each graph class, we will be interested in the complexity of reconstruction using different types of algorithms.

2 Boolean model

2.1 Hamiltonian cycles

Assume we have to reconstruct an unknown Hamiltonian cycle in the complete graph K_n. Under the boolean model, the information theory yields the lower bound $\log_2 \frac{(n-1)!}{2} = \Omega(n \log_2 n)$ as there are $\frac{(n-1)!}{2}$ Hamiltonian cycles on n vertices. The following theorem states that this bound can be reached under particular conditions.

Theorem 1 *The $\Omega(n \log_2 n)$ lower bound on the complexity of Hamiltonian cycle reconstruction can be reached by an adaptive algorithm.*

Note first that if we are restricted to the two-vertex model, any reconstruction algorithm requires $\Omega(n^2)$ queries, as shown in [1].

An adaptive algorithm reconstructing a Hamiltonian cycle H with $2n \log_2 n$ queries has been described in [11]. An interesting fact is that under the boolean model, this complexity cannot be achieved by a nonadaptive algorithm. As showed in [5], $\Omega(n^2)$ queries are necessary for a nonadaptive algorithm to reconstruct a Hamiltonian cycle. The result of [5] is actually more general, and establishes that $\Omega(n^2)$ queries are necessary for a nonadaptive algorithm to reconstruct a graph in one of the following classes: matchings, perfect matchings, graphs isomorphic to a fixed bounded degree graph with $\Omega(n)$ edges, graphs consisting in the disjoint union of a clique of size $n-3$ and a single edge.

This example illustrates the case when adaptive algorithms are strictly more powerful than nonadaptive algorithms.

2.2 Matchings

A matching is a graph such that each vertex has degree 0 or 1. As mentioned above, any nonadaptive algorithm reconstructing a matching requires a quadratic number of queries. More precisely, at least $\frac{49}{153}\binom{n}{2}$ nonadaptive queries are necessary to reconstruct a matching on K_n [5]. The authors of [5] also prove the upper bound $(\frac{1}{2} + o(1))\binom{n}{2}$ using a construction based on the Wilson theorem [22] on the decomposition of complete graphs into subgraphs isomorphic to a given graph.

As the enumeration of matchings is an open question, it is difficult to compute the exact information-theoretic lower bound. However, we can easily compute the number of perfect matchings[5] of K_n to be $\frac{n!}{2^{\lfloor \frac{n}{2} \rfloor} \cdot \lfloor \frac{n}{2} \rfloor !}$. This provides a lower bound on the number of general matchings, and implies the following information-theoretic lower bound on the reconstruction of matchings: $\log_2 \left(\frac{n!}{2^{\lfloor \frac{n}{2} \rfloor} \cdot \lfloor \frac{n}{2} \rfloor !} \right) = (1 + o(1)) \cdot (\frac{n}{2} \log_2 n)$. Even though this bound has been computed for perfect matchings only, it is possible to built an adaptive algorithm reconstructing general matchings and achieving this bound within a constant factor.

The algorithm works in two steps. The first one is adaptive and partitions the set of vertices into $V_1 \uplus V_2$ such that no two vertices in the same V_i are adjacent in the matching. This can be done in n queries by processing vertices one-by-one. The second step can be made nonadaptive. It finds for every $v \in V_1$ the adjacent vertex to v (if it exists) in V_2 using a group testing algorithm to find one "counterfeit coin" among n (see Section 3.1). This group testing problem can be solved within $\lceil \log_2 n \rceil$ nonadaptive queries, yielding a total complexity of $(1 + o(1)) \cdot (n \log_2 n)$ for the entire algorithm. Note that the same algorithm applied to the reconstruction of perfect matchings has an optimal asymptotic complexity $(1 + o(1)) \cdot (\frac{n}{2} \log_2 n)$.

[5] a perfect matching is a graph such that the degree of all vertices except possibly one is 1.

2.3 Stars and cliques

The reconstruction of stars and cliques on n vertices has been studied in [4]. Following that paper, we define S_k to be the set of all stars with a center, k leaves and $n-k-1$ isolated vertices, and C_k to be the set of all cliques with k vertices and $n-k$ isolated vertices. $S = \cup_{k=0}^{n-1} S_k$ and $C = \cup_{k=1}^{n} C_k$ are respectively the set of all stars and all cliques on n vertices, with an arbitrary number of isolated vertices.

We now examine the information-theoretic lower bound for reconstructing stars and cliques under the boolean model. To estimate the cardinality of S, recall that a star of S_k (for $k \geq 2$) is defined by a center chosen among the n vertices and k leaves chosen among the $n-1$ remaining vertices. So $|S| = \sum_{k=2}^{n-1} n \cdot \binom{n-1}{k} + \frac{n(n-1)}{2} + 1 = n \cdot (2^{n-1} - 1) - \frac{n(n-1)}{2} + 1$. Consequently, we get the lower bound $\log_2 |S| = (1 + o(1)) \cdot n$ for the complexity of the star reconstruction problem. For cliques, it is clear that $|C| = \sum_{k=0}^{n} \binom{n}{k} = 2^n$, and the information theoretic lower bound is then $\log_2 |C| = (1 + o(1)) \cdot n$.

For both stars and cliques, the $\Omega(n)$ bound can be achieved by the following algorithm composed of two nonadaptive rounds. At the first round, find a starting vertex from which it becomes easy to reconstruct the whole graph: the center of the star or a vertex that belongs to the clique. Finding the center of the star is done through n nonadaptive queries $V \setminus \{i\}$ for $1 \leq i \leq n$. To find a vertex of the clique, we simply ask the queries $Q_i = \{1, \ldots, i\}$ for $2 \leq i \leq n$. At the second round (nonadaptive as well), finish the reconstruction by determining the neighbors of the starting vertex. Each round requires a linear number of queries.

While cliques and stars can be easily reconstructed in two nonadaptive rounds, the situation changes if we are restricted to fully nonadaptive (1-round) algorithms. To reconstruct a star of S with a nonadaptive algorithm, it is necessary, in the worst case, to query each of the $\binom{n}{2}$ pairs of vertices $\{u, v\}$ [4], i.e. the most naive algorithm turns out to be the optimal one in the worst case. In contrast, for cliques, only $\Omega(n \log n)$ nonadaptive queries are needed, and [4] showed the existence of a nonadaptive algorithm reconstructing a clique of C with $\mathcal{O}(n \log^2 n)$ queries.

3 Quantitative model

We now turn to the quantitative model, much less studied in the literature. We show that under this model, nonadaptive algorithms get all their power and often allow to achieve (or to approach) the lower bound. This is due to powerful combinatorial constructions of $(0,1)$-matrices verifying certain properties.

3.1 Hamiltonian cycles

We start again with our initial problem of reconstructing a Hamiltonian cycle on n vertices. As under the quantitative model there are $n+1$ possible answers to each query $Q \subseteq V$, the information-theoretic lower bound is $\log_{n+1} \frac{(n-1)!}{2} = (1 + o(1)) \cdot n$.

Theorem 2 *Under the quantitative model, there exists an algorithm reconstructing a Hamiltonian cycle in $O(n)$ queries.*

One such algorithm has been presented in [11] and is composed of two steps: an adaptive preparatory step followed by a nonadaptive reconstruction step[6]. We now describe this algorithm.

First stage. The goal of the first stage is to reduce the problem to the reconstruction of bipartite graphs. By processing all the vertices successively, we transform the Hamiltonian cycle H into a tripartite graph, i.e. we partition the set of vertices V into 3 subsets $V_1 \uplus V_2 \uplus V_3$ such that two vertices in the same subset are not adjacent in H. As each vertex has exactly two neighbors, this transformation can be done in at most $2n$ queries. We are now dealing with the problem of reconstruction of a tripartite graph that we view as three bipartite graphs.

Second stage. The second stage reconstructs each of the three bipartite graphs in $O(n)$ nonadaptive queries. This crucial step is based on two auxiliary constructions.

First subproblem. Consider a bipartite graph $(C_1, C_2; E)$ with vertex degree bounded by a constant (2 in our case). Assume that we want to determine the *degrees* of all vertices of C_1 by querying subsets of C_1 together with the whole set C_2. This problem is equivalent to the reconstruction of an unknown vector $v = (v_1, \ldots, v_n)$ with $v_i \in \{0, \ldots, d-1\}$ ($d = 3$ in our case) by querying sums of the form $\sum_{i=1}^{n} \epsilon_i v_i$, $\epsilon_i \in \{0, 1\}$. A nonadaptive algorithm solving this problem corresponds to a $(0,1)$-matrix M of dimension $k \times n$ (k as small as possible) such that for vectors $v \in \{0, \ldots, d-1\}^n$, all products Mv are distinct. We call such matrix a *d-detecting matrix*.

The information-theoretic lower bound for k is $\log_{(d-1)n+1} d^n = (1 + o(1)) \cdot (\frac{n}{\log_d n})$.

For the particular case $d = 2$, this lower bound can be improved to $(2+o(1)) \cdot (\frac{n}{\log_2 n})$, as it was shown in [9] (another proof using Kolmogorov complexity can be found in [16]). On the other hand, it has been shown in [17,6] that this bound can be achieved. A decade later, Lindström [21] gave a tricky effective construction of a 2-detecting matrix with $(2 + o(1)) \cdot (\frac{n}{\log_2 n})$ rows using the Möbius function.

In our case, $d = 3$ and a 3-detecting matrix with $(4+o(1)) \cdot (\frac{n}{\log_2 n})$ rows can be effectively constructed as an extension of the Lindström construction. Furthermore, for an arbitrary constant d, a d-detecting matrix with $(2+o(1))(\log d \cdot \frac{n}{\log n})$ rows can be effectively constructed, and this is also a lower bound [11].

Second subproblem. Consider a bipartite graph $(C_1, C_2; E)$ and a vertex $i \in C_1$. We want to determine the vertices of C_2 adjacent to i by querying i together with subsets of C_2. In the case of Hamiltonian cycle, there are exactly two

[6] as it will follow from Section 3.4, Hamiltonian cycles can be reconstructed in $O(n)$ fully nonadaptive queries. The two-step construction presented here is for explanatory purposes.

such vertices, but to be more general, we assume that their number is bounded by a constant d. The problem can be viewed as a problem of discovering d counterfeit coins (neighbors of i) among n coins (vertices C_2) and is well-known in the area of group testing [8]. We want to solve it in a nonadaptive way (for reasons that will be clear later) using queries of type "*how many* counterfeit coins does a given subset contain?".

The case of finding one counterfeit among n can be solved by an optimal nonadaptive set of queries $Q_i = \{j|$ the i-th bit of j is $1\}$ for $1 \leq i \leq \lceil \log_2 n \rceil$. However, already for two coins the situation gets more complicated: the information-theoretic lower bound is $\log_3 \binom{n}{2} \approx 1.26 \cdot \log_2 n$ while the best known upper bound for *adaptive algorithms* is $1.44 \cdot \log_2 n$. For *nonadaptive algorithms*, the best known lower and upper bounds are respectively $\frac{5}{3} \cdot \log_2 n$ and $2 \cdot \log_2 n$ [18,20].

For the general problem of finding nonadaptively d counterfeit coins among n, we need to construct a $(0,1)$-matrix A of dimension $k \times n$ (k as small as possible) such that for vectors $v \in \{0,1\}^n$ having at most d 1's, all products Av are distinct. We call such a matrix a *d-separating matrix*. Known upper and lower bounds for the number of rows in a d-separating matrix are respectively $(4+o(1)) \cdot (\frac{d}{\log d} \log n)$ [11] and $(2+o(1)) \cdot (\frac{d}{\log d} \log n)$ [3]. Both are proved using probabilistic arguments, and thus the upper bound is non-constructive. The best known explicite nonadaptive construction uses BCH error-correcting codes and uses $O(d \log_2 n)$ queries. Note also that no better properly adaptive algorithm is known.

Combining the subproblems. The two techniques presented above (d-detecting and d-separating matrices) allow us to solve the problem of reconstruction of a bipartite graph $(C_1, C_2; E)$ with the degree of each vertex in C_1 bounded by a constant d. Using d-separating matrices, the adjacent vertices of each $i \in C_1$ can be obtained by querying i against $P_1, \ldots, P_m \subseteq C_2$, where $P_1, \ldots, P_m$ do not depend on i. For each P_j, we can determine the degree of each $i \in C_1$ in P_j by querying P_j against $S_1, \ldots, S_\ell \subseteq C_1$ using d-detecting matrices. Again, $S_1, \ldots, S_\ell$ do not depend on P_j. Thus, querying all pairs $S_k \cup P_j$ is sufficient to reconstruct the whole graph. The resulting number of queries is $(2+o(1))(\log d \frac{n}{\log n})(4+o(1))(\frac{d}{\log d} \log n) = (8+o(1))dn$.

This proves the following

Theorem 3 *A (one-sided) d-bounded degree bipartite graph can be reconstructed within $(8+o(1)) \cdot dn$* nonadaptive *queries. This matches the lower bound up to a constant factor.*

Turning back to our initial motivation (Theorem 2), a Hamiltonian cycle can be reconstructed within $2n + 3 \cdot 2 \log_2 n \cdot \frac{(4+o(n)) \cdot n}{\log_2 n} = O(n)$ queries asymptotically by a two-stage algorithm. This matches the lower bound up to a constant factor.

3.2 Matchings

As in Section 2.2, consider the lower bound $\frac{n!}{2^{\lfloor \frac{n}{2} \rfloor} \cdot \lfloor \frac{n}{2} \rfloor !}$ on the number of matchings on n vertices. Note that as the number of edges in a matching on n vertices is

at most $\lfloor \frac{n}{2} \rfloor$, the maximal number of distinct answers to a query is $\lfloor \frac{n}{2} \rfloor + 1$. Consequently, we can compute an information-theoretic lower bound on the complexity of the matching reconstruction problem under the quantitative model to be $\log_{\lfloor \frac{n}{2} \rfloor + 1} \left(\frac{n!}{2^{\lfloor \frac{n}{2} \rfloor} \cdot \lfloor \frac{n}{2} \rfloor !} \right) = (1 + o(1)) \cdot \frac{n}{2}$.

It is possible to reach this bound, up to a constant factor, by a fully nonadaptive algorithm. This will follow from Section 3.4 where we describe a general nonadaptive algorithm for reconstructing graphs of vertex degree bounded by d within $\mathcal{O}(dn)$ queries.

3.3 Stars and cliques

Recall from Section 2.3 that the number of stars and cliques on n vertices are respectively $|S| = n \cdot (2^{n-1} - 1) - \frac{n(n-1)}{2} + 1$ and $|C| = 2^n$. The information-theoretic lower bound for reconstructing stars under the quantitative model is then $\log_n \left(n \cdot (2^{n-1} - 1) - \frac{n(n-1)}{2} + 1 \right) = (1 + o(1)) \cdot (\frac{n}{\log_2 n})$ and that for cliques is $\log_{\frac{n(n-1)}{2}+1}(2^n) = (\frac{1}{2} + o(1)) \cdot (\frac{n}{\log_2 n})$.

There exist adaptive algorithms that achieve these bounds within a constant factor. Here we give only a very high-level description of them. Similar to Section 2.3, the algorithms are divided into two main steps, the first one is adaptive and the second one nonadaptive. At the first step, we find, in a logarithmic number of adaptive queries, either the center of the star, or one vertex of the clique. (This can be done using binary search.) $(2 + o(1)) \frac{n}{\log_2 n}$ nonadaptive queries are then sufficient to reconstruct the neighbors of the vertex found in the first stage, using 2-detecting matrices introduced in the first subproblem of Section 3.1 (see [19,13]). For stars, this construction applies immediately and for cliques, we need to transform each query answer from $k + k(k-1)/2$ to k which is done non-ambiguously.

3.4 Bounded degree graphs

Theorem 3 states that a (one-sided) d-bounded degree bipartite graph can be reconstructed through $O(dn)$ nonadaptive queries. We now want to use this technique to reconstruct general bounded degree graphs [13]. The idea is to consider a bipartite representation of a graph defined as follows. Given a graph $G = (V, E)$, the bipartite representation of G is $G' = (V_1, V_2; E')$, where V_1 and V_2 are two disjoint copies of V, $E \subseteq V_1 \times V_2$, and $(i, j) \in E$ implies $(i, j) \in E'$ and $(j, i) \in E'$. Note that any edge of G produces two edges in G'. Moreover, if G is d-bounded degree then G' is d-bounded degree too.

We want to query the binary representation through the following queries: "Given $X \subseteq V_1$ and $Y \subseteq V_2$, how many edges are there in G' connecting vertices of X to vertices of Y ? ". We define the corresponding query function $\mu'_{G'}(X, Y) = |E' \cap (X \times Y)|$. A query $\mu'_{G'}(X, Y)$ can be expressed through quantitative queries to the initial graph G, i.e. through the query function $\mu_G(X) = |E \cap (X \times X)|$, for $X \subseteq V$. Using elementary set-theoretic considerations, it can be shown that $\mu'(X, Y) = \mu((X \setminus Y) \cup (Y \setminus X)) - 2\mu(X \setminus Y) - 2\mu(Y \setminus X) + \mu(X) + \mu(Y)$.

By Theorem 3, the binary representation G' can be reconstructed by $O(dn)$ nonadaptive queries $\mu'(X, Y)$. From the observation above, it follows that G' can be reconstructed by $O(dn)$ nonadaptive queries $\mu(X)$.

Theorem 4 *A d-bounded degree graph can be reconstructed within $O(dn)$ nonadaptive queries. This is an asymptotically tight bound.*

3.5 General graphs

Under the quantitative model, the information-theoretic lower bound for reconstructing general graphs is $\log_{1+\frac{n(n-1)}{2}} 2^{\frac{n(n-1)}{2}} = (\frac{1}{4}+o(1))\cdot\frac{n^2}{\log_2 n}$. A better lower bound $(\frac{1}{2}+o(1))\cdot\frac{n^2}{\log_2 n}$ can be obtained using lower bounds for d-detecting matrices (see Section 3.1). As it was shown in [13], this bound can be achieved up to a constant factor using again the bipartite representation of a graph introduced in the previous section.

Consider the bipartite representation $G' = (V_1, V_2; E')$ of a general graph $G = (V, E)$. For each vertex $i \in V_1$, reconstruct its adjacent vertices among $\{1, \ldots, i-1\} \subseteq V_2$ with $(2+o(1))\cdot\frac{i}{\log_2 i}$ queries of the form $\mu'(\{i\}, W)$, $W \subseteq V_2$, using 2-detecting matrices. Observe that $\mu'(\{i\}, W) = \mu(W \cup \{i\}) - \mu(W \setminus \{i\})$ which allows us to express each query $\mu'(\{v_i^1\}, W)$ through two queries to the original graph G.

The overall complexity of this method for the reconstruction of a general graph is then $\sum_{i=2}^{n}(2+o(1))\cdot\frac{i}{\log_2 i} = (2+o(1))\frac{n^2}{\log_2 n}$. This is within the factor of four from the known lower bound for nonadaptive algorithms.

Theorem 5 *A general graph can be reconstructed within $(2+o(1))\frac{n^2}{\log_2 n}$ nonadaptive queries. This matches the lower bound up to a constant factor.*

3.6 k-degenerate graphs and trees

The general technique used to reconstruct bounded degree graphs (Section 3.4) can be further extended to reconstruct more general *k-degenerate graphs*. An intuitive definition of k-degenerate graphs is as follows: G is k-degenerate if there exists a vertex v of G with vertex degree less than or equal to k such that $G \setminus \{v\}$ has the same property. More formally, a graph G is k-degenerate if vertices V can be ordered $(v_1, v_2, \cdots, v_n)$ such that $deg_{G_i}(v_i) \leq k$, where G_i is the subgraph of G induced by the vertices $\{v_i, v_{i+1}, \cdots, v_n\}$. For example, trees are 1-degenerate as there exists a leaf of vertex degree 1 and after deleting it the graph is still a tree. Another example is provided by planar graphs that are 5-degenerate: there is always a vertex of degree at most 5 and deleting it keeps the graph planar.

Let us first compute the information-theoretic lower bound for the reconstruction of k-degenerate graphs. The number of edges in a k-degenerate graph is clearly less than nk. To obtain a lower bound on the number of k-degenerate

graphs, we fix some order on vertices and count number of possibilities to connect v_{k+t} to $v_{k+t+1}, \cdots, v_n$. Since all such choices can be made independently for all $v_1, \cdots, v_{n-k}$, we have $N(n+1,k) \geq \prod_{i=k+1}^{n} \binom{i}{k} \geq \frac{(n!)^k}{k^{nk}}$. The corresponding information-theoretic lower bound is then

$$\log_{nk} N(n+1,k) \geq \frac{nk(\log n - \log k - 1)}{\log n + \log k}.$$

In the case $k \leq n^\alpha$ for some $\alpha < 1$, this bound can be simplified into $\Omega(nk)$. For n sufficiently large, we can prove that this bound is tight, meaning that there exists an algorithm that reconstructs a graph in the class of k-degenerate graphs with $\mathcal{O}(nk)$ queries.

Theorem 6 *k-degenerate graphs on n vertices can be reconstructed by a nonadaptive algorithm using $\mathcal{O}(nk)$ queries, and this bound is tight.*

As in the case of bounded degree graphs (Section 3.4), the algorithm uses the bipartite representation of k-degenerate graphs and the same general technique of reconstructing bipartite graphs. While the bipartite representation here is not of bounded vertex degree, the sum of degrees of all vertices from one side is bounded by nk. Therefore, instead of using d-detecting matrices (first subproblem in Section 3.1), we consider matrices that solve a more general combinatorial search problem, namely the reconstruction of *d-bounded weight vectors* which are vectors with the sum of entries bounded by d. Formally, define the class of d-bounded weight vectors by $\Lambda(n,d) = \{(v_1, \ldots, v_n) | v_i \in \mathbb{N} \text{ and } \sum_{i=1}^{n} v_i \leq d\}$. A nonadaptive algorithm reconstructing d-bounded weight vectors is specified by an object-query incidence matrix M such that $M \cdot v_1 \neq M \cdot v_2$ for all $v_1, v_2 \in \Lambda(n,d)$, $v_1 \neq v_2$. It has been shown in [10] that there exists such a matrix with the number of rows $k(n,d) \leq \frac{4min(n,d)\log\left(C_1 \frac{max(n,d)}{min(n,d)}\right)}{\log min(n,d) + C_2} + C_3 \log d$, for some constants C_1, C_2 and C_3.

Consider now the bipartite representation $G' = (V_1, V_2; E')$ of a k-degenerate graph G. Assume we are given two families $\{Q_j\}_{j=1}^{m}$ and $\{P_i\}_{i=1}^{l}$ that solve the d-bounded weight vector reconstruction problem for $d = k$ and $d = 2nk$ respectively. From the bound on $k(n,d)$ above, it follows that $m = \mathcal{O}(k\frac{\log n}{\log k})$ and $l = \mathcal{O}(n\frac{\log k}{\log n})$ when $n \to \infty$. It can be shown that the set of queries $\{\mu'(P_i, Q_j)\}_{i=1,\ldots,l}^{j=1,\ldots,m}$ reconstructs k-degenerate graphs. The proof, given in [10], combines the ideas of Section 3.1 with an iterative procedure of computing the answers of queries $\mu'(P_i, Q_j)$ that would be obtained after deleting all edges incident to a vertex of degree at most k (by definition of k-degenerate graphs, such a vertex always exists).

The overall complexity of the algorithm is $m \cdot l = \mathcal{O}(nk)$, which proves Theorem 6.

4 Conclusions and open problems

Through examples of Hamiltonian cycles, matchings, stars and cliques, the quantitative model has been shown to be more powerful than the boolean model. The

following table illustrates this difference and provides lower and upper bounds (for adaptive and nonadaptive algorithms) for the two-vertex, boolean and quantitative models, for the case of Hamiltonian cycle that has been our main applicative motivation.

	lower bound	adaptive	nonadaptive
two-vertex model	$\Omega(n^2)$	$O(n^2)$	$O(n^2)$
boolean model	$\Omega(n \log n)$	$O(n \log n)$	$\Omega(n^2)$
quantitative model	$\Omega(n)$	$O(n)$	$O(n)$

Another important conclusion is that nonadaptive algorithms fully benefit from the quantitative model, and vice versa. Not only the quantitative model allows faster reconstruction algorithms, but also these algorithms can be made nonadaptive, or "almost nonadaptive" (having an important nonadaptive component). Interestingly, under the quantitative model, nonadaptive algorithms often reach the asymptotic lower bound and no properly adaptive algorithm is known to outperform nonadaptive algorithms. This contrasts with the boolean model, where nonadaptive algorithms are usually strictly less powerful than adaptive ones.

The power of nonadaptive algorithms under the quantitative model is due to powerful combinatorial constructions of d-detecting and d-separating matrices (Section 3.4) and their generalizations (Section 3.6).

As far as open questions are concerned, we would like to mention two of them here. One concerns an important technical point: the upper bound for d-separating matrices (Section 3.4). The tight upper bound $O(\frac{d}{\log d} \log n)$ has been proved by a probabilistic nonconstructive argument, and finding an *effective* construction of d-separating matrices with $O(\frac{d}{\log d} \log n)$ rows remains an important open question. Another question is of more general nature: how far can we go with optimal nonadaptive reconstruction under the quantitative model? For example, can we reconstruct in $O(dn)$ queries any graph with $O(dn)$ edges?

To conclude, we get back to the applicative side of our study and mention that many other bioinformatics applications give rise to combinatorial search problems. Such applications include screening clone libraries [15], the FISH (*Fluorescent In Situ Hybridization*) method for chromosome identification [12], determination of exon-intron boundaries in genes [7], probe selection for DNA chips [14], and others. Thus, those applications provide a rich source for new interesting developments of combinatorial search methods in future.

References

1. M. Aigner. *Combinatorial Search*. John Wiley and Sons, 1988.
2. B. Alberts, A. Johnson, J. Lewis, M. Raff, K. Roberts, and P. Walter. *Molecular Biology of the Cell*. Garland Science, 1994.
3. N. Alon. Separating matrices. Private communication, May 1997.
4. N. Alon and V. Asodi. Learning a hidden subgraph. In *Automata, Languages and Programming: 31st International Colloquium, ICALP 2004, Turku, Finland,*

July 12-16, 2004. Proceedings, volume 3142 of *Lecture Notes in Computer Science*, pages 110–121. Springer, 2004.

5. N. Alon, R. Beigel, S. Kasif, S. Rudich, and B. Sudakov. Learning a hidden matching. In *Proceedings of the 43rd Annual IEEE Symposium on Foundations of Computer Science, FOCS 2002, Vancouver, BC, Canada, 16–19 November 2002*, pages 197–206. IEEE Computer Society Press, 2002.
6. D.G. Cantor and W.H. Mills. Determination of a subset from certain combinatorial properties. *Can. J. Math*, 18:42–48, 1966.
7. F. Cicalese, P. Damaschke, and U. Vaccaro. Optimal group testing algorithms with interval queries and their application to splice site detection. In *Proc. of the Int. Workshop on Bioinformatics Research and Applications (IWBRA 2005)*, volume 3515 of *Lectures Notes in Computer Science*, pages 1029–1037. Springer, 2005.
8. D. Du and F. Hwang. *Combinatorial group testing and its applications*, volume 3. Series on applied Mathematics, 1993.
9. P. Erdös and A. Rényi. Asymmetric graphs. *Acta Math. Acad. Sci. Hung. Acad. Sci.*, 14:295–315, 1963.
10. V. Grebinski. On the power of additive combinatorial search model. In *Proc. of Computing and Combinatorics, 4th Annual International Conference, COCOON'98, Taipei, Taiwan, August 12-14, 1998*, volume 1449 of *Lecture Notes in Computer Science*, pages 194–203. Springer, 1998.
11. V. Grebinski and G. Kucherov. Reconstructing a hamiltonian cycle by querying the graph: Application to DNA physical mapping. *Discrete Applied Mathematics*, 88:147–165, 1998.
12. V. Grebinski and G. Kucherov. Reconstructing set partitions. In *Proceedings of the 10th Annual ACM-SIAM Symposium on Discrete Algorithms, SODA'99 (Baltimore, Maryland, January 17-19, 1999)*, pages 915–916. ACM, SIAM, 1999.
13. V. Grebinski and G. Kucherov. Optimal reconstruction of graphs under the additive model. *Algorithmica*, 28:104–124, 2000.
14. G.W Klau, S. Rahmann, A. Schliep, M. Vingron, and K. Reinert. Optimal robust non-unique probe selection using integer linear programming. *Bioinformatics*, 20 (suppl. 1):i186–i193, 2004.
15. E. Knill and S. Muthukrishnan. Group testing problems in experimental molecular biology. Technical Report LAUR-95-1503, Los Alamos National Laboratory, March 1995.
16. M. Li and P. M. B. Vitányi. Kolmogorov complexity arguments in combinatorics. *J. Comb. Theory Series A*, 66(2):226–236, 1994.
17. B. Lindström. On a combinatorial problem in number theory. *Canad. Math. Bull*, 8:477–490, 1965.
18. B. Lindström. Determination of two vectors from the sum. *J. Comb. Theory*, 6:402–407, 1969.
19. B. Lindström. On Möbius functions and a problem in combinatorial number theory. *Canad. Math. Bull.*, 14(4):513–516, 1971.
20. B. Lindström. On b_2 sequences of vectors. *Journal of Number Theory*, 4:261–265, 1972.
21. B. Lindström. Determining subsets by unramified experiments. In editor J.N. Srivastava, editor, *A Survey of Statistical Designs and Linear Models*, pages 407–418. North Holland, Amsterdam, 1975.
22. R. M. Wilson. Decomposition of complete graphs into subgraphs isomorphic to a given graph. In *Congressus Numerantium XV*, pages 647–659, 1975.