Optimal Query Bounds for Reconstructing a Hamiltonian Cycle in Complete Graphs (extended abstract)

Vladimir Grebinski and Gregory Kucherov
INRIA-Lorraine and CRIN/CNRS
Campus Scientifique
615, rue du Jardin Botanique, BP 101
54602 Villers-lès-Nancy, France
{grebinsk,kucherov}@loria.fr

Abstract

This paper studies four combinatorial search models of reconstructing a fixed unknown Hamiltonian cycle in the complete graph by means of queries about subgraphs. For each model, an efficient algorithm is proposed that matches asymptotically the information-theoretic lower bound. The problem is motivated by an application to genome physical mapping.

1. Introduction

1.1. Combinatorial Search

Combinatorial Search can be informally defined as determining an unknown object of a certain class through indirect *queries* about this object. The goal of combinatorial search is to identify the unknown object with as little cost as possible. While the cost measure may vary, it is often defined as the number of queries made by the search algorithm. We refer to [1] for a systematic presentation of the field.

Group Testing is probably the oldest and most well-known subfield of combinatorial search. In group testing, we are given a set of *items* some of which are "defective". We want to determine the defective items by making queries about subsets of items. The typical allowed form of queries is "does the subset contain a defective?", but other types of queries may be allowed (e.g. on the number of defectives in the set). In [7], Du and Hwang survey numerous group testing results. Interestingly, the first systematically treated group testing problem was motivated by efficiently finding the contaminated blood samples out of a large collection of samples, using the possibility of pouring samples together and testing the mixtures [6]. Another well-known example of combinatorial search is to identify one or more counterfeit coins in a set by weighing subsets of coins using scales of some kind.

More general search problems can be expressed in terms of graphs. The simplest instance of this class is a search for an unknown edge in a given graph by testing subgraphs induced by subsets of vertices. Another type of problem is to reconstruct an unknown graph of a given class. Here the allowed queries may be of the form "Does the edge (v_1, v_2) belong to the graph?" or more generally, "Does the subset of vertices $\{v_1, \ldots, v_k\}$ contain pairs of adjacent vertices?", "How many?", etc. It is this type of problem that we consider in this paper. Finally, a related but different type of search problems on graphs consists in checking if an unknown graph verifies some property without actually reconstructing the graph. We refer the reader to [1, 7] for an account of known results on these classes of search problems.

In combinatorial search, it is important to distinguish between adaptive (sequential) and non-adaptive (predetermined) algorithms. In adaptive algorithms, a query essentially depends on the results of queries made "so far". In the non-adaptive case, all the queries are mutually independent and can be given before any answer is known. Although non-adaptive algorithms are obviously less powerful in general, they often admit "nicer" mathematical formulations which allow to use more powerful mathematical methods. Besides, in many cases (including some cases considered in this paper) non-adaptive algorithms achieve the power of adaptiveness, that is reach the lower bound. Note also that in non-adaptive algorithms all queries can be made in parallel, which is useful in many applications. In practice, an intermediate solution is often used: the algorithm performs a

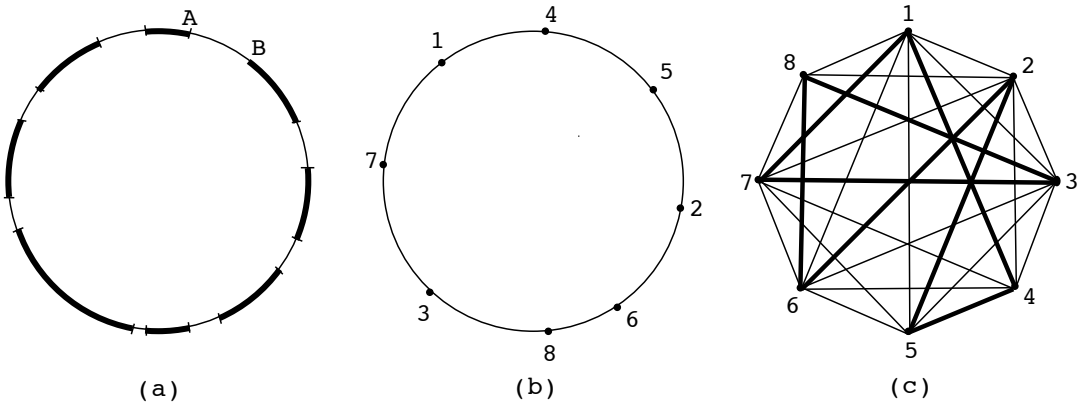

Figure 1. (a) Placement of contigs on a circular genome, (b) Placement of points on the circle (contigs are assimilated to points), (c) Hamiltonian cycle in K_8 corresponding to the order of points

preliminary "rough" selection in the non-adaptive way and then finds a final result adaptively, or vice versa. Such an approach, called *two-stage*, is used in [11] for example.

Combinatorial search problems occur in numerous application areas, such as software engineering, industrial product testing, multi-access communication, medicine, etc. The problem we study in this paper has been motivated by a problem of molecular biology that we shortly describe below.

1.2. Biological motivation

Combinatorial search problems are often encountered in genome analysis. A typical example is screening clone libraries by hybridization probes [2]. Here groups of clones (possibly overlapping fragments of the DNA molecule), called pools, can be tested with a probe, and the aim is to determine the individual clones containing each probe through the minimal number of tests. The problem of efficiently constructing a set of pools is studied in [2, 4, 11], and [12] briefly surveys related theoretical results and mentions several other applications of combinatorial search in genome analysis.

Combinatorial search problems, including the screening problem above, occur often in the process of physical mapping. Physical mapping is the central stage in genome exploration which basically consists in reconstructing the relative positions of clones from some partial information about them (typically, information about occurrences of some short nucleotide sequences). We refer to [19] for more information about physical mapping and related mathematical problems.

The results of this paper are motivated by a particular scenario of physical mapping proposed in [18]. Its simplified description is as follows. Assume that we have identified, using some other methods, a certain number of groups of overlapping clones, called *contigs*. Contigs don't overlap and are located on the genome in some unknown order (see Figure 1(a)). The problem is to reconstruct the mutual placement of the contigs, that is their order and the lengths of the gaps between them (*physical map*). The tool for doing this is the multiplex LA PCR (*Long Accurate Polymerase Chain Reaction*) hereafter called simply *experiment* or *reaction*. Each clone is characterized by two *primers* which are short nucleotide sequences that characterize its ends (see [16]). An input to the experiment is a set of *primers* of bounded cardinality (due to technological restrictions). Whenever this set contains two primers corresponding to the adjacent ends of neighboring contigs (like primers A and B on Figure 1(a)), this is detected by the reaction and the distance between them can be determined. If there are several such pairs in the set, this fact can also be detected, and several distances can be identified. This method rises the following combinatorial question: what is the optimal strategy of conducting experiments in order to obtain the physical map using minimal number of them?

1.3. Problem formulation

In this paper we study the problem of *combinatorial search of a Hamiltonian cycle in the complete graph*:

> Let K_n be the complete graph with vertices $\{1, 2, \ldots, n\}$. Assume that some Hamiltonian cycle HC is fixed in K_n, that is not known to us. We are allowed to make *queries* about adjacency of some vertices in HC. Determine HC by making as few queries as possible.

Model	lower bound	algorithm performance	type of algorithm	
			first stage	second stage
multi-point model	$\Omega(n \log n)$	$O(n \log n)$	adaptive	
quantitative multi-point model	$\Omega(n)$	$O(n)$	adaptive	non-adaptive
k-point model	$\Omega(\frac{n^2}{k^2})$	$(1+o(1))\frac{n^2}{k^2}$	non-adaptive	adaptive
quantitative k-point model	$\Omega(\frac{n^2}{k^2})$	$(1+o(1))\frac{n^2}{k^2}$	non-adaptive	non-adaptive

Figure 2. Result summary

Obviously, the solution and its complexity will strongly depend on the type of queries we are allowed to make. Let $K_{\{a_1,\ldots a_m\}}$ denote the complete graph on the set of vertices $\{a_1,\ldots a_m\}$. We study the following four query types that lead to four different combinatorial search models:

Multi-vertex model For a set of vertices $\{a_1,\ldots,a_m\}$, is $K_{\{a_1,\ldots a_m\}} \cap HC$ empty?

Quantitative multi-vertex model For a set of vertices $\{a_1,\ldots,a_m\}$, what is the number of edges in $K_{\{a_1,\ldots a_m\}} \cap HC$?

k-vertex model Assume that a constant k is predefined. For a set of vertices $\{a_1,\ldots,a_m\}$, where $m \leq k$, is $K_{\{a_1,\ldots a_m\}} \cap HC$ empty?

Quantitative k-vertex model For a set of vertices $\{a_1,\ldots,a_m\}$, where $m \leq k$, what is the number of edges in $K_{\{a_1,\ldots a_m\}} \cap HC$?

In the multi-vertex model we ask whether at least one edge from a given set belongs to the Hamiltonian cycle. However, this set has a special structure – we ask about the edges of a complete subgraph. The multi-vertex model is strengthened in the quantitative multi-vertex model. Now we are allowed to *count* the number of edges of the Hamiltonian cycle in complete subgraphs. This is the most powerful model. The k-vertex model and quantitative k-vertex model are restriction of the multi-vertex model and quantitative multi-vertex model respectively. These models are motivated by practical constraints in biological experiments.

The k-vertex and quantitative k-vertex models are directly related to the physical mapping problem described above. The problem can be formalized as the reconstruction of the order of points, placed on the circle, by means of *queries* about the presence (or the number) of adjacent pairs in a given subset of points (see Figure 1(b); we refer to [8] for more details). Clearly, an order of n points on the circle can be uniquely associated with a Hamiltonian cycle in the complete undirected graph K_n (we assume that the direction on the circle is irrelevant). Figure 1(c) illustrates the Hamiltonian cycle corresponding to the order of Figure 1(b).

1.4. Summary of the results

In this paper we study the complexity of each of the four models above and design asymptotically optimal algorithms for all of them. The complexity bounds are summarized in Figure 2. The adaptive nature of the algorithm is indicated in the last column. For two-stage algorithms, the type of each stage is shown.

We also discuss the multiplicative constants hidden in the O-notation, that are of course important for practical applicability of the algorithms.

All the results hold for a more general problem – reconstructing an unknown graph with the degree of all vertices bounded by 2. Such a graph can be represented as a union of paths and cycles without common vertices.

2. Multi-point model

There are $(n-1)!/2$ Hamiltonian cycles in K_n. By the standard information-theoretic argument, the lower bound $\log(n-1)!/2 = \Omega(n \log n)$ [1] can be immediately obtained. Note that the same lower bound holds for the average complexity since the average length of a branch in a binary tree with $(n-1)!/2$ leaves is in $\Omega(n \log n)$.

Suppose that only two points can be tested at a time, that is each query tests whether or not an individual edge belongs to the Hamiltonian cycle. It is known (see [1, section 3.5, exercise 3.5.5]) that in this case at least $n^2/4 - n/2 - 1 = \Omega(n^2)$ queries must be made in the worst case. In our model, we are able to simultaneously ask about many edges. However, this set of edges has a special structure – it is a complete subgraph rather than any subgraph. Recall that if a subset has been detected to contain adjacent vertices, we have no information about what these vertices are. Therefore, it is not immediately clear if we can benefit from the possibility of testing many edges at once.

[1] Throughout the paper the logarithms are binary unless the base is indicated.

In this section we show that the lower bound $\Omega(n \log n)$ can be achieved. Below we propose an algorithm that matches this bound.

Let HC be a Hamiltonian cycle and assume we have already discovered some of its edges. These edges form a set of disjoint paths that will be our main data structure.

Definition 1 *Let HC be a Hamiltonian cycle in K_n. A* chain *c is a sequence of vertices $< a_1, \ldots, a_t >$, $t \geq 1$, such that $\forall j,\ 1 \leq j \leq t-1$, $(a_j, a_{j+1}) \in HC$. Note that one-vertex chains are allowed. For $c =< a_1, \ldots, a_t >$, where $t \geq 2$, define $left(c) = a_1$, $right(c) = a_t$. For $c =< a_1 >$, $left(c) = a_1$, $right(c) = \emptyset$. For a set of chains $C = \{c_1, \ldots, c_k\}$, define $left(C) = \{left(c_1), \ldots, left(c_k)\}$ and $right(C) = \{right(c_1), \ldots, right(c_k)\}$. A set of chains is* independent *if for every $c_i, c_j \in C$, the edges $(left(c_i), left(c_j))$, $(left(c_i), right(c_j))$, $(right(c_i), right(c_j))$ don't belong to HC.*

The following algorithm solves the problem.

```
RECONSTRUCT-MULTI(n, HC)
1   C := {< 1 >}
2   for i := 2 to n do
3       C := INSERT-VERTEX(C, i)
```

The function INSERT-VERTEX(C, i) inserts vertex i into C maintaining the independent set of chains. Clearly, when all vertices have been processed, the set C consists of the Hamiltonian cycle HC.

```
INSERT-VERTEX(C, i)
1   query left(C) ∪ {i} and right(C) ∪ {i}
2   if both answers are no
3       then add the one-vertex chain < i > to C
4   if left(C) ∪ {i} yields yes and right(C) ∪ {i} yields no
5     then find in left(C) one or two vertices adjacent to i
6          if one such vertex is found
7              then append i to the corresponding chain
8          if two such vertices a'_1, a'_2 are found
9              then replace the chains < a'_1, ..., a'_t' >,
                    < a''_1, ..., a''_t'' > by the chain
                    < a'_t', ..., a'_1, i, a''_1, ..., a''_t'' >
10  if right(C) ∪ {i} yields yes and left(C) ∪ {i} yields no
11      then proceed symmetrically to the previous case
12  if both left(C) ∪ {i} and right(C) ∪ {i} yields yes
13      then find in left(C) a vertex a'_1 adjacent to i
14           find in right(C) a vertex a''_t'' adjacent to i
15           replace the chains < a'_1, ..., a'_t' >,
                < a''_1, ..., a''_t'' > by the chain
                < a''_1, ..., a''_t'', i, a'_1, ..., a'_t' >
16  return C
```

It remains to estimate the number of queries of steps 5 and 13-14. By simple binary search, step 5 can be done in $\lceil 2 \log n \rceil$ queries. Similarly, steps 13, 14 can be done in $\lceil \log n \rceil$ each. Thus, INSERT-VERTEX(C, i) makes at most $\lceil 2 + 2 \log n \rceil$ queries and the whole algorithm RECONSTRUCT-MULTI(n, HC) makes $\lceil (2 + 2 \log n)n \rceil = O(n \log n)$ queries which matches the lower bound.

3. Quantitative multi-point model

This model extends the multi-point model by the possibility of counting in a query set the number of pairs of vertices adjacent in HC. The first observation is that this feature decreases the information-theoretic lower bound. Since each query has potentially $n+1$ distinct answers, at least $\log_{n+1}(n-1)!/2 = \Omega(n)$ queries must be made by any algorithm. In this section we prove that (surprisingly enough) this bound can be achieved and propose an algorithm that matches this bound.

The algorithm has two main stages:

```
RECONSTRUCT-QUANTITATIVE(n, HC)
1   split the set of vertices {1, ..., n} into three disjoint
        subsets such that any two vertices from the same
        subset are not adjacent in HC
2   find all edges of HC between the subsets
```

The first stage is easy to accomplish in $O(n)$ queries:

```
SPLIT(n, HC)
1   initialize three empty sets S_1, S_2, S_3
2   for i := 2 to n do
3       if querying S_1 ∪ {i} yields no
4           then add i to S_1
5           elseif querying S_2 ∪ {i} yields no
6               then add i to S_2
7               else add i to S_3
```

Clearly, SPLIT(n, HC) makes at most $2n$ queries.

The second stage deals with three bipartite graphs formed by the edges of HC between the vertices of each of the three subsets. Consider such a graph. It is a bipartite graph in which the degrees of vertices have the values $\{0, 1, 2\}$. The problem now is to reconstruct this graph by querying its subgraphs, where the output of a query is the number of edges in the subgraph.

Let C_1, C_2 be the two independent vertex sets of the bipartite graph, each of size n. As the first step, consider the problem of determining the degree of each vertex in C_1 by querying different subsets of C_1 together with the whole set C_2. This can be trivially done in $O(n)$ steps by querying, for each $i \in C_1$, the set $\{i\} \cup C_2$ and getting immediately the degree of i. However, it is possible to do better.

The problem can be again reformulated as follows: Reconstruct an unknown vector $(a_1, \ldots, a_n)$, where $a_i \in \{0, 1, 2\}$, by means of querying, for a set of indices $I = \{i_1, \ldots, i_l\} \subseteq \{1, \ldots, n\}$, for the sum $\sum_{j \in I} a_j$.

Consider the problem above where $a_i \in \{0, 1\}$. For this case, a solution was proposed by Lindström [14]. Given n, a $k \times n$ $\{0, 1\}$-matrix A is called a *detecting matrix* for the set of $\{0, 1\}$-vectors of length n, if for any two such vectors v_1, v_2, $v_1 \neq v_2$ implies $Av_1 \neq Av_2$. In other terms, the sums of two different subsets of columns of A are different. Associating columns to positions and interpreting rows as incidence vectors of queries, such a matrix provides a *non-adaptive* algorithm for the vector reconstruction problem with $a_i \in \{0, 1\}$, that makes k queries.

Theorem 1 (Lindström [14]) *A detecting $k \times n$ matrix for the set of $\{0, 1\}$-vectors can be effectively constructed with $k = 2n/\log n$ asymptotically.*

Note that the algorithm provided by theorem 1 meets the information-theoretic lower bound $\Omega(n/\log n)$, which is easily obtained from the equality $(n+1)^k \geq 2^n$ relating the number of different column sums and different vectors.

For the case when $a_i \in \{0, 1, 2\}$, we obtained the following extension of the Lindström's method.

Theorem 2 *A detecting $k \times n$ matrix for the set of vectors with elements $\{0, 1, 2\}$ can be effectively constructed with $k = 4n/\log n$ asymptotically.*

The theorem is a consequence of a more general result that can be found in the full version of this paper [8].

Let us turn back to the bipartite graph problem. As the second step, consider the following problem: Given a vertex $i \in C_1$, find its adjacent vertices (at most two) in C_2 by querying subsets of C_2. The simplest way to do it (see also Section 2) is to find the two vertices in $2 \log n$ tests using binary search. However, the binary search is a strongly adaptive method, and for the reason that will become clear in a moment, we need a non-adaptive algorithm.

Note that this problem is closely related to the problem of determining two counterfeit coins in a set of n coins (see [10]). In our case, two counterfeit coins should be identified in a non-adaptive manner and querying a subset yields the number (0,1 or 2) of counterfeit coins in it ("spring scale model"). Giving such a non-adaptive algorithm amounts to constructing a $k \times n$ $\{0, 1\}$-matrix such that the sums of two different pairs of columns are all different. Let us call such a matrix a *2-separation $k \times n$ matrix*. The following result is from [13].

Theorem 3 ([13]) *A 2-separation $k \times n$ matrix can be effectively constructed with the asymptotic value of $k = 2 \log n$.*

A proof, different from that of [13], can be found in the full version [8]. Note that although Theorem 3 provides the same bound as the naive binary search method, its proof is non-trivial since the non-adaptiveness is a serious restriction here. For comparison, the optimal adaptive algorithm for finding two "defective objects" in the model with counting was proved to make $C \log n$ queries, where $1.26 \leq C \leq 1.44$ ([10]).

Now we are in position to give an efficient algorithm for the bipartite graph problem that combines the two non-adaptive algorithms above.

Consider the non-adaptive algorithm based on Theorem 3 for finding the two adjacent vertices in C_2 for a given vertex $i \in C_1$. This algorithm is simply a collection of subsets $P_1, \ldots, P_k \subseteq C_2$ such that the numbers of adjacent vertices of i in $P_1, \ldots, P_k$ identify uniquely the two adjacent vertices of i in C_2. Since P_j's don't depend on i, we will ask about each P_j for all $i \in C_1$ "at once" by applying the detecting matrix of Theorem 2.

RECONSTRUCT-BIPARTITE(C_1, C_2)
1 **for** $j := 1$ **to** k **do**
2 apply the detecting matrix to find, for each $i \in C_1$, the number of adjacent vertices in P_j

Clearly, after the whole run of RECONSTRUCT-BIPARTITE(C_1, C_2) the number of adjacent vertices of each $i \in C_1$ in each P_j will be known, and therefore the adjacent vertices themselves can be determined. We conclude that RECONSTRUCT-BIPARTITE reconstructs a bipartite graph with n vertices in each component asymptotically in $(2 \log n)(4n/\log n) = 8n = O(n)$ queries.

Turning back to algorithm RECONSTRUCT-QUANTITIVE, solving the initial Hamiltonian cycle reconstruction problem, we summarize the complexity in the following final theorem.

Theorem 4 *Reconstructing a Hamiltonian cycle in the quantitative multi-point model can be done in $O(n)$ queries.*

Proof: Consider the algorithm RECONSTRUCT-QUANTITIVE. The first step (algorithm SPLIT) requires $2n$ queries. Note that the size of each independent set is at most $n/2$. The second step can be done by three applications of algorithm RECONSTRUCT-BIPARTITE. The overall query complexity is $2n + 3 \cdot 8n/2 = 14n = O(n)$. □

4. k-point and quantitative k-point models

In some applications, only a limited number of vertices can be tested. The biological method described in [18] re-

stricts this number to 16, as reactions with a bigger number of primers don't give reliable outputs. In terms of our mathematical model, the number of vertices that we can test is bounded by some predefined constant. This restriction cannot be captured by the methods of Sections 2 and 3, as they essentially require at some stages an unbounded number of vertices. This motivates the k-point and the quantitative k-point models, that are restrictions of the multi-point and quantitative multi-point models respectively.

First observe that each experiment with k vertices can be simulated by $\frac{k(k-1)}{2}$ experiments with 2 vertices. Since the lower bound for the 2-point model is asymptotically $n(n-2)/4$ (see Section 2), any algorithm that solves the problem in the restricted models makes at least $\frac{n(n-2)}{2k(k-1)}$ queries. Therefore, the focus of this section is to reduce the multiplicative constant in the quadratic complexity bound.

For both models, we propose an algorithm which makes $\frac{n(n-1)}{k(k-1)}(1+o(1))$ queries, which is twice more than the above lower bound. Note that this complexity is the best we could expect, since for the 2-point model, no essentially better algorithm than querying all the $n(n-1)/2$ edges is presently known.

The central idea is to cover K_n by subgraphs $G_1, \ldots, G_M$, where each G_i is a complete graph K_m, $m \leq k$, such that every edge (i,j) of K_n belongs to only one G_i. Assume that such a covering is constructed. By querying each G_i, we find at most n of them which contain edges of the Hamiltonian cycle HC. In each such G_i, we can identify the edges of HC using the technique developed in the previous sections. (Of course, $G_i \cap HC$ does not form a Hamiltonian cycle of G_i, but the results of Sections 2 and 3 still apply to such graphs as it was noted in Section 1.4.) Processing one G_i then requires $O(k \log k)$ queries for the k-point model (Section 2) and $O(k)$ for the quantitative k-point model (Section 3), and the overall complexity of the method is respectively $M + O(nk \log k)$ and $M + O(nk)$. Thus, the main problem is to minimize M, that is to cover the graph K_n by a minimal number of complete graphs K_m, $m \leq k$, such that every edge of K_n occurs in only one of them. In the rest of the Section we describe how it can be done.

The problems of arranging objects from some set into some number of (intersecting) subsets of a given size such that each object and each pair of objects occur in a specified number of subsets is a well-established area in combinatorics called *Design theory* or *Block design* (see e.g. [9, 3, 5]). However, most of these results give existence conditions for arrangements and don't give algorithms for their construction. Furthermore, the subsets are usually required to have one or several specified cardinalities. These requirements are too strong for our purpose, as we allow subgraphs of any size smaller than k and we look for an algorithm approximating the minimal number of subgraphs and not for an exact solution.

Another link that should be mentioned here is the Theorem of Rödl [17] that insures that one can find an asymptotically optimal coverage. However, we need stronger properties – the construction should be "efficient" and should guarantee that no edge is covered many times.

We present below an algorithmic solution to this problem. This solution is related to classical Design Theory results (see methods of *Affine Block Design* in [3]), but we will not discuss this relationship here. Instead, we present it in a self-contained way and focus on algorithmic aspects and complexity analysis.

Lemma 1 *Consider the complete graph K_n. Let $n \geq k^2$ and assume that the set of vertices $V = \{1, \ldots, n\}$ is divided into k disjoint subsets $S_1, \ldots, S_k$ of n/k elements each. If $n/k = p^a$ is a prime power, then $(n/k)^2$ subgraphs K_k can be effectively constructed such that every edge between S_i and S_j, $i \neq j$, occurs in exactly one of the subgraphs.*

Proof: Consider a $k \times n/k$ table A^0 where the elements of S_i are placed (in any order) in row i. Consider the Galois field $\mathbb{GF}(p^a)$ and let us view the elements of each row as distinct elements of $\mathbb{GF}(p^a)$. With each row i we associate a "speed" $v_i \in \mathbb{GF}(p^a)$ such that all speeds are different (this is possible as $k < n/k$). Now construct a sequence of $k \times n/k$ tables $A^1, A^2, \ldots, A^{n/k-1}$ according to the following formula: $A^t(i,j) = A^0(i, j + v_i * t)$, $1 \leq t \leq n/k - 1$, where $+$ and $*$ are addition and multiplication in $\mathbb{GF}(n/k)$ (i, j and t are also naturally regarded as elements of $\mathbb{GF}(n/k)$). Intuitively, at each step t each row is "rotated" by v_i. We claim that for every two elements $x \in S_{i_1}, y \in S_{i_2}$, $i_1 \neq i_2$, there is exactly one column in $A^0, A^1, \ldots, A^{n/k-1}$ containing both x and y. Indeed, since $v_{i_1} \neq v_{i_2}$, the equation $j_1 + v_{i_1} * t = j_2 + v_{i_2} * t$ has exactly one solution $t = (j_1 - j_2) * (v_{i_2} - v_{i_1})^{-1}$ in $\mathbb{GF}(n/k)$. This shows that if there are two such columns, they must belong to the same A^t. However, in each A^t there is only one column containing x in row i_1. □

The proof gives an effective procedure of constructing the subgraphs K_k. If n/k is not a prime power, we can extend the set V of vertices by dummy vertices V' such that $(|V| + |V'|)/k$ is a prime power, and apply the construction. Thus, the following corollary holds.

Corollary 1 *Under the conditions of Lemma 1, if $r = p^a$ is a prime power greater than n/k, then r^2 subgraphs K_m, $m \leq k$ can be effectively constructed such that every edge between S_i and S_j, $i \neq j$, occurs in exactly one subgraph.*

To cover the whole graph, we apply the construction recursively to each subset S_j. This leads to the following

algorithm.

COVER(K_n, k)
1 **if** $n \geq k^2$ **then**
2 find the smallest number $q \geq n$ such that q/k is a prime power
3 divide the n vertices into k disjoint subsets $S_1, \ldots, S_k$
4 by applying Corollary 1 find $(q/k)^2$ subgraphs K_m, $m \leq k$ covering each edge between distinct subsets S_i and S_j exactly once
5 **for** $j := 1$ **to** k
6 COVER($|S_j|, k$)
7 **elseif** $k < n < k^2$ **then**
8 find the smallest number $q \geq k^2$ such that q/k is a prime power
9 proceed as in the previous case
10 **elseif** $n \leq k$ **then**
11 output K_n

To estimate the total number M of subgraphs K_m, we need an estimate of the smallest prime power greater than n. From Number Theory results on the distribution of primes the asymptotic bound $nextp(n) - n \leq n^{\frac{11}{20}}$ is known (see e.g. [15]), where $nextp(n) = min\{p \text{ is prime} | p \geq n\}$. Let $M = f(n,k)$ be the total number the subgraphs constructed by COVER(n,k). Then

$$f(n,k) = \begin{cases} 1 & \text{if } n \leq k \\ nextp(k)^2 + k & \text{if } k < n < k^2 \\ nextp(\frac{n}{k})^2 + k * f(\frac{n}{k}, k) & \text{if } n \geq k^2 \end{cases}$$

As $n \to \infty$, we consider only the last case.

$$f(n,k) \leq \sum_{i=1}^{\log_k n} k^{i-1} * \left(\frac{n}{k^i} + (\frac{n}{k^i})^{11/20} \right)^2 = \\ = \frac{n(n-1)}{k(k-1)} + o(\frac{n^2}{k^2})$$

Since the overall query complexity of the method is $f(n,k) + O(nk \log k)$ for the k-point model and $f(n,k) + O(nk)$ for the quantitative k-point model, we obtain the final result.

Theorem 5 *Reconstructing a Hamiltonian cycle in the k-point model can be done in $\frac{n(n-1)}{k(k-1)}(1 + o(1))$ queries.*

5. Concluding remarks

We have studied four combinatorial search models for reconstructing an unknown Hamiltonian cycle in the complete graph. For the multi-point model, a simple algorithm has been proposed which makes at most $2n \log n$ queries while $n \log n$ queries is the information-theoretic lower bound. The Quantitative multi-point model introduces the possibility of counting the number of pairs of adjacent vertices in a set. For this model, an algorithm of linear complexity has been described which is asymptotically the best possible. Finally, we considered the k-point and quantitative k-point models that take into account an important practical restriction – a bound on the size of tested subgraphs. For these models, the algorithms with complexity $\frac{n(n-1)}{k(k-1)}(1 + o(1))$ have been proposed, which is the best we could expect. The key of the method is Lemma 1 that gives an efficient algorithm for covering a complete graph K_n by "small" complete graphs K_m, $m \leq k$, such that every edge of K_n occurs in exactly one small graph. All the results stay valid for the problem of reconstructing an unknown graph with the degree of all vertices bounded by 2.

The algorithm for the k-point model has been implemented in MAPLE and some "realistic" computational experiments have been made (up to $n \approx 2000$). The results showed, in addition, that the algorithm adapts well to some other practical constraints arising in particular in the physical mapping application described in Section 1.2.

Acknowledgements We are greatly indebted to Alexei Sorokin and Alexandre Bolotin from the Laboratoire de Génétique Microbienne of the Institute National de la Recherche Agronomique at Jouy-en-Josas(France) for comments and helpful discussions about the physical mapping problem. We thank Rakesh Verma for reading and commenting on the manuscript.

References

[1] M. Aigner. *Combinatorial Search*. John Wiley & Sons, 1988.
[2] E. Barillot, B. Lacroix, and D. Cohen. Theoretical analysis of library screening using a n-dimentional pooling strategy. *Nucleic Acids Res.*, 19:6241–6247, 1991.
[3] T. Beth, D. Jungnickel, and H. Lenz. *Design Theory*. Cambridge University Press, 1986.
[4] W. Bruno, E. Knill, D. Balding, D. Bruce, N. Doggett, W. Sawhill, R. Stallings, C. Whittaker, and D. Torney. Efficient pooling designs for library screening. *Genomics*, 26:21–30, 1995.
[5] J. H. Dinitz and D. R. Stinson, editors. *Contemporary Design Theory*. John Wiley & Sons, 1992.
[6] R. Dorfman. The detection of defective members of large population. *Ann. Math. Statist.*, 14:436–440, 1943.
[7] D.-Z. Du and F. K. Hwang. *Combinatorial Group Testing and its applications*, volume 3 of *Series on applied mathematics*. WorldScientific, 1993.
[8] V. Grebinski and G. Kucherov. Reconstructing a hamiltonian circuit by querying the graph: Application to DNA physical mapping. IR 96-R-123, Centre de Recherche en Informatique de Nancy, 1996.

[9] M. Hall. *Combinatorial Theory*. Blaisdell Publishing Company, 1967.

[10] F. Hwang. A tale of two coins. *American Mathematical Monthly*, 94:121–129, Feb. 1987.

[11] E. Knill. Lower bounds for identifying subset members with subset queries. In *Proc. 6th ACM-SIAM Symp. on Discrete Algorithms (SODA)*, pages 369–377, 1995.

[12] E. Knill and S. Muthukrishnan. Group testing problems in experimental molecular biology. Technical Report LAUR-95-1503, Los Alamos National Laboratory, March 1995.

[13] B. Lindström. Determination of two vectors from the sum. *Journal of Combinatorial Theory*, A6:402–407, 1969.

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

[15] C. Mozzochi. On the difference between consecutive primes. *J. of Number Th.*, 24:181–187, 1986.

[16] E. Port, F. Sun, D. Martin, and M. Waterman. Genomic mapping by end-characterized random clones: A mathematical analysis. *Genomics*, 26:84–100, 1995.

[17] V. Rödl. On a packing and covering problem. *European Journal of Combinatorics*, 5:69–78, 1985.

[18] A. Sorokin, A. Lapidus, V. Capuano, N. Galleron, P. Pujic, and S. D. Ehrlich. A new approach using multiplex long accurate PCR and yeast artificial chromosomes for bacterial chromosome mapping and sequencing. *Genome Research*, 6:448–453, 1996.

[19] M. S. Waterman. *Introduction to Computational Molecular Biology. Maps, sequences et genomes*. Chapman & Hill, 1995.