社団法人 情報処理学会　研究報告
IPSJ SIG Technical Report
2004－AL－93　(14)
2004／1／30

プローブ区間グラフの拡張MPQ木表現

上原 隆平†

Abstract

プローブ区間グラフは DNA の構造解析のために導入されたグラフクラスで，区間グラフの自然な一般化でもある．本稿ではプローブ区間グラフのための拡張型の $\mathcal{MPQ}$ 木構造を提案した．この木構造は可能な区間表現を自然に表現するデータ型で，$O(n^3)$ 時間で計算できる．この木構造を使うと，プローブ区間グラフの同型性が $O(n^3)$ 時間で判定できる．また，二つの未プローブな DNA 切片の間の関係を特定することができる．したがってプローブ途中の DNA に対して，次にどの切片をプローブすればよいか，という指標を効率良く計算することができる．

Extended MPQ-Trees for Probe Interval Graphs

Ryuhei UEHARA†

Abstract

Probe interval graphs are introduced to deal with the physical mapping and sequencing of DNA as a generalization of interval graphs. In this paper, we propose extended $\mathcal{MPQ}$-trees to represent the probe interval graphs. An extended $\mathcal{MPQ}$-tree is canonical, represents all possible permutations of the intervals, and can be constructed in $O(n^3)$ time. Thus we can solve the graph isomorphism problem for the graphs in $O(n^3)$ time. Using the tree, we can determine the relationship of two nonprobes. Therefore, in a sense, we can find the best nonprobe that would be probed in the next experiment.

1 Introduction

Interval graphs were introduced in the 1950's by Hajös and Benzer independently. Since then a number of interesting applications for interval graphs have been found including to model the topological structure of the DNA molecule, scheduling, and others (see [4, 10, 2] for further details). The interval graph model requires all overlap information. However, in many cases, only partial overlap data exist. The class of *probe interval graphs* is introduced by Zhang in the assembly of contigs in physical mapping of DNA, which is a problem arising in the sequencing of DNA (see [13, 15, 14, 10] for background). A probe interval graph is obtained from an interval graph by designating a subset P of vertices as *probes*, and removing the edges between pairs of vertices in the remaining set N of *nonprobes*. That is, on the model, only partial overlap information (between a probe and the others) is given. Recently, the recognition algorithms of the graph class are investigated [6, 9, 5].

A data structure called $\mathcal{PQ}$-trees was developed by Booth and Lueker to represent all possible permutations of the intervals of an interval graph [1]. Korte and Möhring simplified their algorithm by introducing $\mathcal{MPQ}$-trees [7]. An $\mathcal{MPQ}$-tree is canonical; that is, given two interval graphs are isomorphic if and only if their corresponding $\mathcal{MPQ}$-trees are isomorphic. In general, given probe interval graph, there are several affirmative interval graphs those are not isomorphic, and their interval representations are consistent to the probe interval graph. Thus there are no canonical $\mathcal{MPQ}$-trees for probe interval graphs.

In this paper, we extend $\mathcal{MPQ}$-trees to represent probe interval graphs. The extended $\mathcal{MPQ}$-tree is canonical for any probe interval graph, and the tree can be constructed in $O(n^3)$ time. There are several applications including:

†駒澤大学 自然科学教室 (Komazawa University, Natural Science Faculty), `uehara@komazawa-u.ac.jp`

1. The graph isomorphism problem for probe interval graphs can be solved in $O(n^3)$ time. From the practical point of view, it is useful in the Computational Biology community; we might determine that two given DNA sequences differ before probing all nonprobes.

From the theoretical point of view, the complexity of the graph isomorphism of probe interval graphs was not known (see [12] for related results and references). Thus the result improves the upper bound of the graph classes such that the graph isomorphism problem can be solved in polynomial time.

2. The extended $\mathcal{MPQ}$-tree gives us the information about nonprobes; two nonprobes are either (1) independent (and they cannot overlap each other), (2) overlapping, or (3) not determined without experiments. Hence, to clarify the structure of the DNA sequence, we only have to experiment on the nonprobes in the case (3). Moreover, we can find the nonprobe that has most nonprobes in the case (3) in linear time. Therefore, we can determine the "best" nonprobe to fix the structure of the DNA sequence in a sense.

3. We can enumerate all possible permutations of the intervals for given probe interval graph, which is beneficial in the Computational Biology community.

Due to space limitation, all proofs and several templates are omitted and can be found in full draft available at `http://www.komazawa-u.ac.jp/ uehara/ps/MPQpig.pdf`.

2 Preliminaries

The *neighborhood* of a vertex v in a graph $G = (V, E)$ is the set $N_G(v) = \{u \in V | \{u, v\} \in E\}$, and the *degree* of a vertex v is $|N_G(v)|$ and denoted by $deg_G(v)$. For the vertex set U of V, we denote by $N_G(U)$ the set $\{v \in V | v \in N(u)$ for some $u \in U\}$. If no confusion can arise we will omit the index G. Given graph $G = (V, E)$, its cograph is defined by $\bar{E} = \{\{u, v\} | \{u, v\} \notin E\}$, and denoted by $\bar{G} = (V, \bar{E})$. A vertex set I is *independent set* if $G[I]$ contains no edges, and then the graph $\bar{G}[I]$ is said to be *clique*.

For a given graph $G = (V, E)$, a sequence of the vertices $v_0, v_1, \cdots, v_l$ is a *path*, denoted by $(v_0, v_1, \cdots, v_l)$, if $\{v_j, v_{j+1}\} \in E$ for each $0 \le j \le l - 1$. The *length* of a path is the number of edges on the path. For two vertices u and v, the *distance* of the vertices is the minimum length of the paths joining u and v. A *cycle* is a path beginning and ending with the same vertex. A cycle of length i is denoted by C_i. An edge which joins two vertices of a cycle but is not itself an edge of the cycle is a *chord* of that cycle. A graph is *chordal* if each cycle of length at least 4 has a chord. Given graph $G = (V, E)$, a vertex $v \in V$ is *simplicial* in G if $G[N(v)]$ is a clique in G. The following lemma is a folklore:

Lemma 1 Given chordal graph, all simplicial vertices can be found in linear time.

Two graphs $G = (V, E)$ and $G' = (V', E')$ are *isomorphic* iff there is a one-to-one mapping $\phi : V \to V'$ which satisfies $\{u, v\} \in E$ iff $\{\phi(u), \phi(v)\} \in E'$ for every pair of vertices u and v. Given graphs G and G', *graph isomorphism problem* is the problem to determine if G is isomorphic to G'

2.1 Interval graph representation

A graph (V, E) with $V = \{v_1, v_2, \cdots, v_n\}$ is an *interval graph* if there is a set of intervals $\mathcal{I} = \{I_{v_1}, I_{v_2}, \cdots, I_{v_n}\}$ such that $\{v_i, v_j\} \in E$ iff $I_{v_i} \cap I_{v_j} \ne \emptyset$ for each i and j with $1 \le i, j \le n$. We call the set $\mathcal{I}$ of intervals *interval representation* of the graph. For each interval I, we denote by $R(I)$ and $L(I)$ the right and left endpoints of the interval, respectively (hence we have $L(I) \le R(I)$ and $I = [L(I), R(I)]$).

A graph $G = (V, E)$ is a *probe interval graph* if V can be partitioned into subsets P and N (corresponding to the *probes* and *nonprobes*) and each $v \in V$ can be assigned to an interval I_v such that $\{u, v\} \in E$ iff both $I_u \cap I_v \ne \emptyset$ and at least one of u and v is in P. In this paper, we assume that P and N are given, and then we denote by $G = (P, N, E)$. By definition, N is an independent set, $G[P]$ is an interval graph, and $G[P \cup \{v\}]$ is also an interval graph for any $v \in N$. Let $G = (P, N, E)$ be a probe interval graph. Let E^+ be a set of edges $\{t_1, t_2\}$ with $t_1, t_2 \in N$ such that there are two probes v_1 and v_2 in P such that $\{v_1, t_1\} \in E$, $\{v_1, t_2\} \in E$, $\{v_2, t_1\} \in E$, $\{v_2, t_2\} \in E$, and $\{v_1, v_2\} \notin E$. In the case, we can know that intervals t_1 and t_2 have to overlap without experiment. Each edge in E^+ is called an *enhanced edge*, and the graph $G^+ := (P, N, E \cup E^+)$ is said to be an *enhanced probe interval graph*. It is known that a probe interval graph is weakly chordal [11], and an enhanced probe interval graph is chordal [13, 15]. For further details and references can be found in [2, 10].

For given probe interval graph G, an interval graph G' is said to be *affirmative* iff G' gives one possible interval representations for G. In general, for a probe interval graph G, there are several non-isomorphic affirmative interval graphs. For given probe interval graph $G = (P, N, E)$, the affirmative interval graph G' is also said to be *affirmative* to the corresponding enhanced probe interval graph $G^+ = (P, N, E \cup E^+)$.

2.2 $\mathcal{PQ}$-trees and $\mathcal{MPQ}$-trees

$\mathcal{PQ}$-trees were introduced by Booth and Lueker [1], and which can be used to recognize interval graphs as follows. A *$\mathcal{PQ}$-tree* is a rooted tree T with two types of internal nodes: $\mathcal{P}$ and $\mathcal{Q}$, which will be represented by circles and rectangles, respectively. The leaves of T are labeled 1-1 with the maximal cliques of the interval graph G. The *frontier* of a $\mathcal{PQ}$-tree T is the permutation of the maximal cliques obtained by the ordering of the leaves of T from left to right. $\mathcal{PQ}$-tree T and T' are *equivalent*, if one can be obtained from the other by applying the following rules a finite number of times; (1) arbitrarily permute the successor nodes of a $\mathcal{P}$-node, or (2) reverse the order of the successor nodes of a $\mathcal{Q}$-node. In [1], Booth and Lueker showed that a graph G is an interval graph iff there is a $\mathcal{PQ}$-tree T whose frontier represents a consecutive arrangement of the maximal cliques of G. They also developed an $O(|V| + |E|)$ algorithm that either constructs a $\mathcal{PQ}$-tree for G, or states that G is not an interval graph.

Lueker and Booth [8], and Colbourn and Booth [3] developed labeled $\mathcal{PQ}$-trees in which each node contains information of vertices as labels. Their labeled $\mathcal{PQ}$-trees are *canonical*; given interval graphs G_1 and G_2 are isomorphic iff corresponding labeled $\mathcal{PQ}$-trees T_1 and T_2 are isomorphic. Since we can determine if two labeled $\mathcal{PQ}$-trees T_1 and T_2 are isomorphic, the isomorphism of interval graphs can be determined in linear time.

$\mathcal{MPQ}$-trees are developed by Korte and Möhring to simplify the construction of $\mathcal{PQ}$-trees [7].The $\mathcal{MPQ}$-tree T^* assigns sets of vertices (possibly empty) to the nodes of a $\mathcal{PQ}$-tree T representing an interval graph $G = (V, E)$. A $\mathcal{P}$-node is assigned only one set, while a $\mathcal{Q}$-node has a set for each of its sons (ordered from left to right according to the ordering of the sons).

For a $\mathcal{P}$-node $\hat{P}$, this set consists of those vertices of G contained in all maximal cliques represented by the subtree or $\hat{P}$ in T, but in no other cliques[‡]. For a $\mathcal{Q}$-node $\hat{Q}$, the definition is more involved. Let $Q_1, \cdots, Q_m$ ($m \geq 3$) be the set of the sons (in consecutive order) of $\hat{Q}$, and let T_i be the subtree of T with root Q_i. We then assign a set S_i, called *section*, to $\hat{Q}$ for each Q_i. Section S_i contains all vertices that are contained in all maximal cliques of T_i and some other T_j, but not in any clique belonging to some other subtree of T that is not below $\hat{Q}$.

In [7], Korte and Möhring showed linear time algorithms that construct an $\mathcal{MPQ}$-tree for given interval graph. Although it does not shown explicitly, the $\mathcal{MPQ}$-tree is essentially the same as the labeled $\mathcal{PQ}$-tree in [3], and hence the graph isomorphism problem can be solved in linear time using the $\mathcal{MPQ}$-trees.

The property of $\mathcal{MPQ}$-trees is summarized as follows [7, Theorem 2.1]:

Theorem 2 Let T^* be the canonical $\mathcal{MPQ}$-tree for given interval graph $G = (V, E)$. Then

(a) T^* can be obtained in $O(|V| + |E|)$ time and $O(|V|)$ space.

(b) Each maximal clique of G corresponds to a path in T^* from the root to a leaf, where each vertex $v \in V$ is as close as possible to the root.

(c) In T^*, each vertex v appears in either one leaf, one $\mathcal{P}$-node, or consecutive sections S_i, S_{i+1}, $\cdots$, S_{i+j} (with $j > 0$) in a $\mathcal{Q}$-node.

(d) The root of T^* contains all vertices belonging to all maximal cliques, while the leaves contain the simplicial vertices.

Lemma 3 Let $\hat{Q}$ be a $\mathcal{Q}$-node in the canonical $\mathcal{MPQ}$-tree. Let $S_1, \cdots, S_k$ (in this order) be the sections of $\hat{Q}$, and let U_i denote the set of vertices occurring below S_i with $1 \leq i \leq k$. Then we have the following;

(a) $S_{i-1} \cap S_i \neq \emptyset$ for $2 \leq i \leq k$,

(b) $S_1 \subseteq S_2$ and $S_k \subseteq S_{k-1}$,

‡We will use $\hat{P}, \hat{Q}$, and $\hat{N}$ for describing a $\mathcal{P}$-node, $\mathcal{Q}$-node, any node, respectively to distinguish probe set P and nonprobe set N.

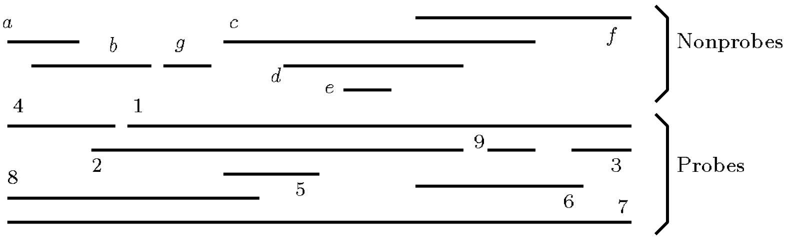

Figure 1: Given probe interval graph G

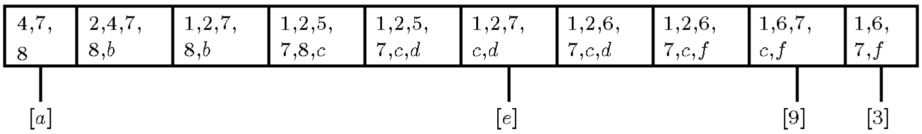

Figure 2: The $\mathcal{MPQ}$-tree of $G - g$

(c) $U_1 \neq \emptyset$ and $U_k \neq \emptyset$,

(d) $(S_i \cap S_{i+1}) \setminus S_1 \neq \emptyset$ and $(S_{i-1} \cap S_i) \setminus S_k \neq \emptyset$ for $2 \leq i \leq k-1$,

(e) $S_{i-1} \neq S_i$ with $2 \leq i \leq k-1$, and

(f) $(S_{i-1} \cup U_{i-1}) \setminus S_i \neq \emptyset$ and $(S_i \cup U_i) \setminus S_{i-1} \neq \emptyset$ for $2 \leq i \leq k$.

Given enhanced probe interval graph $G^+ = (P, N, E \cup E^+)$, let u and v be any two nonprobes with $\{u, v\} \notin E^+$. Then, we say that u *intersects* v if $I_u \cap I_v \neq \emptyset$ for all affirmative interval graphs of G^+. The nonprobes u and v are *independent* if $I_u \cap I_v = \emptyset$ for all affirmative interval graphs of G^+. Otherwise, we say that the nonprobe u *potentially intersects* v. If u potentially intersects v, we cannot determine their relation without experiments.

2.3 Extended $\mathcal{MPQ}$-trees

If given graph is an interval graph, the corresponding $\mathcal{MPQ}$-tree is uniquely determined up to isomorphism. However, for a probe interval graph, this is not in the case. For example, consider a probe interval graph $G = (P, N, E)$ with $P = \{1, 2, 3, 4, 5, 6, 7, 8, 9\}$ and $N = \{a, b, c, d, e, f, g\}$ given in Figure 1. If the graph does not contain the nonprobe g, we have the canonical $\mathcal{MPQ}$-tree in Figure 2. However, the graph is a probe interval graph and we do not know if g intersects b and/or c since they are nonprobes. According to the relations between g and b and/or c, we have four possible $\mathcal{MPQ}$-trees that are affirmative to G shown in Figure 3, where X is either $\{1, 2, 7, 8\}$, $\{1, 2, 7, 8, c\}$, or $\{1, 2, 7, 8, b, c\}$. We call such a vertex g *floating leaf* (later, it will be shown that such a vertex has to be a leaf in an $\mathcal{MPQ}$-tree). For a floating leaf, there is a corresponding $\mathcal{Q}$-node (which also will be shown later). Thus we extend the notion of a $\mathcal{Q}$-node to contain the information of the floating leaf. A floating leaf appears consecutive sections of a $\mathcal{Q}$-node $\hat{Q}$ as the ordinary vertices in $\hat{Q}$. To distinguish them, we draw them over the corresponding sections; see Figure 4. Further details will be discussed in Section 3.

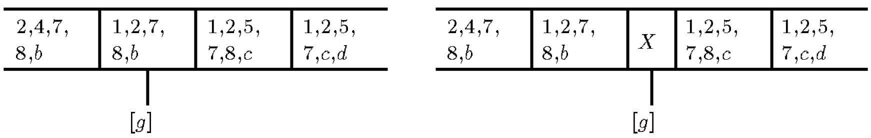

Figure 3: Four $\mathcal{MPQ}$-trees of G

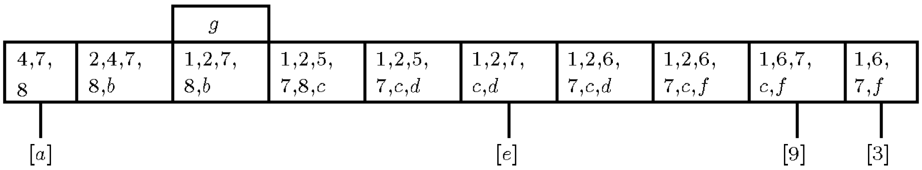

Figure 4: The extended $\mathcal{MPQ}$-tree of G

3 Construction of Extended $\mathcal{MPQ}$-tree of Probe Interval Graph

Let $G = (P, N, E)$ be a given probe interval graph, and $G^+ = (P, N, E \cup E^+)$ be the corresponding enhanced probe interval graph, where E^+ is the set of enhanced edges. Then the outline of the algorithm is as follows.

A0. Given probe interval graph $G = (P, N, E)$, compute the enhanced probe interval graph $G^+ = (P, N, E \cup E^+)$;

A1. Partition N into two subsets $N_S := \{u | u \text{ is simplicial in } G^+\}$ and $N^* := N \setminus N_S$;

A2. Construct the (ordinary) $\mathcal{MPQ}$-tree T^* of $G^* = (P, N^*, E^*)$, where E^* is the set of edges induced by $P \cup N^*$ from G^+;

A3. Embed each nonprobe v in N_S into the (extended) $\mathcal{MPQ}$-tree T^*.

For example, for the graph $G = (P, N, E)$ in Figure 1, $E^+ = \{\{c, d\}, \{c, f\}\}$, $N_S = \{a, e, g\}$, and $N^* = \{b, c, d, f\}$. The following observation is obtained by definition:

Observation 4 Let v be a nonprobe in N_S. Then for any two vertices $u_1, u_2 \in N_{G^+}(v)$, $I_{u_1} \cap I_{u_2} \neq \emptyset$.

3.1 Construction of $\mathcal{MPQ}$-tree of G^*

Let $G^* = (P, N^*, E^*)$ be the enhanced probe interval graph induced by P and N^*. The following lemma plays an important role in this subsection.

Lemma 5 Let u and v be any nonprobes in N^*. Then there is an interval representation of G^* such that $I_u \cap I_v \neq \emptyset$ if and only if $\{u, v\} \in E^+$.

The definition of (enhanced) probe interval graphs and Lemma 5 imply the main theorem in this section:

Theorem 6 The enhanced probe interval graph $G^* = (P, N^*, E^*)$ is an interval graph.

Corollary 7 The canonical $\mathcal{MPQ}$-tree T^* of G^* can be computed in linear time.

Hereafter we call the graph $G^* = (P, N^*, E^*)$ *the backbone interval graph* of $G^+ = (P, N, E \cup E^+)$. In the canonical $\mathcal{MPQ}$-tree T^*, for each pair of nonprobes u and v, their corresponding intervals intersect iff $\{u, v\} \in E^+$. This implies the following observation.

Observation 8 The canonical $\mathcal{MPQ}$-tree T^* gives us the possible interval representations of G^* such that two nonprobes in N^* do not intersect as possible as they can.

For example, for the graph $G = (P, N, E)$ in Figure 1, the canonical $\mathcal{MPQ}$-tree of the backbone interval graph $G^* = (P, N^*, E^*)$ is described in Figure 5. In the $\mathcal{MPQ}$-tree, $I_d \cap I_f = \emptyset$.

3.2 Embedding of Nonprobes in N_S

Lemma 9 For each nonprobe v in N_S, all vertices in $N(v)$ are probes.

Lemma 10 For any probe interval graph G, there is an affirmative interval graph G' such that every nonprobe v in N_S of G is also simplicial in G'.

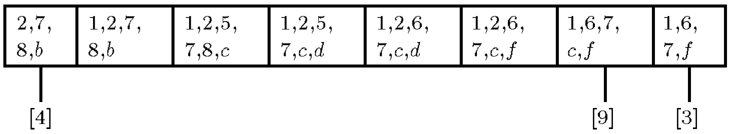

Figure 5: The canonical $\mathcal{MPQ}$-tree T^* of G^*

By Lemma 10 and Theorem 2(d), we have the following corollary.

Corollary 11 For any probe interval graph G, there is an affirmative interval graph G' such that every nonprobe v in N_S of G is in a leaf of the $\mathcal{MPQ}$-tree of G'.

Our embedding is a natural extension of what due to Korte and Möhring [7]. Each node $\hat{N}$ (including $\mathcal{Q}$-node) of the current tree T^* and each section S of a $\mathcal{Q}$-node is labeled according to how the nonprobe v in N_S is related to the probes in $\hat{N}$ or S. Nonprobes in $\hat{N}$ or S are ignored. The label is ∞, 1, or 0 if v is adjacent to all, some, or no probe from $\hat{N}$, or S, respectively. Empty sets (or the sets containing only nonprobes) obtain the label 0. Labels 1 and ∞ are called *positive* labels.

Lemma 12 For a nonprobe v in N_S, all nodes with a positive label are contained in a unique path of T^*.

Let P' be the unique minimal path in T^* containing all nodes with positive label. Let P be a path from the root of the $\mathcal{MPQ}$-tree T^* to a leaf containing P' (a leaf is chosen in any way). Let $\hat{N}_*$ be the lowest node in P with positive label. (That is, $\hat{N}_*$ is the node of the largest depth in P'.) If P contains nonempty $\mathcal{P}$-nodes or sections above $\hat{N}_*$ with label 0 or 1, let $\hat{N}^*$ be the highest such $\mathcal{P}$-node or $\mathcal{Q}$-node containing the section. Otherwise put $\hat{N}_* = \hat{N}^*$.

When $\hat{N}_* \neq \hat{N}^*$, we have the following lemma:

Lemma 13 We assume that $\hat{N}_* \neq \hat{N}^*$. Let $\hat{Q}$ be any $\mathcal{Q}$-node with sections $S_1, \cdots, S_k$ in this order between $\hat{N}_*$ and $\hat{N}^*$. If $\hat{Q}$ is not $\hat{N}^*$, all neighbors of v in $\hat{Q}$ appear in either S_1 or S_k.

Note that Lemmas 12 and 13 correspond to [7, Lemma 4.1]. However, Lemma 13 does not hold at the node N^*. We are now ready to use the same bottom-up strategy from $\hat{N}_*$ to $\hat{N}^*$ as in [7]. In [7], the ordering of vertices are determined by LexBFS. In our algorithm, the step A3 consists of the following substeps;

A3.1. while there is a nonprobe v such that $N_* \neq N^*$ for v, embed v into T^*;

A3.2. while there is a nonprobe v such that $N_* = N^*$ for v and v is not a floating leaf, embed v into T^*;

A3.3. embed each nonprobe v (such that $N_* = N^*$ and v is a floating leaf) into T^*.

In the embedding, we have the following assertions:

Assertion 14 (1) Each nonprobe in N_S has no intersection with unnecessary nonprobes,

(2) each leaf contains at most one nonprobe from N_S, and

(3) each nonprobe in N_S is in a leaf.

All templates for embedding are omitted due to space limitation.

Example 15 For the graph $G = (P, N, E)$ in Figure 1 with its backbone interval graph in Figure 5, the extended $\mathcal{MPQ}$-tree $\tilde{T}$ is shown in Figure 4. Note that we can know that e intersects both of c and d with neither experiments nor enhanced edges. We also note that I_a and I_b could have intersection, but they are standardized according to Assertion 14(1).

3.3 Analysis of Algorithm

We first show the correctness. Since the correctness of steps A0, A1, and A2 are immediately, we concentrate on step A3. First, the templates cover all formally distinct cases. All templates for the case $\hat{N}_* = \hat{N}^*$ with the help-templates H1 and H2 in [7] are easily shown to be correct. Thus we consider the case $\hat{N}_* \neq \hat{N}^*$.

Theorem 16 When $\hat{N}_* \neq \hat{N}^*$, v is not a floating leaf.

We have the following corollary which corresponds to Corollary 4.4 in [7]:

Corollary 17 When $\hat{N}_* \neq \hat{N}^*$, all nodes properly between $\hat{N}_*$ and $\hat{N}^*$ on the path P will become inner sections of a $\mathcal{Q}$-node after embedding of v.

Theorem 18 The resulting extended $\mathcal{MPQ}$-tree is canonical up to isomorphism.

We next show the time complexity.

Theorem 19 For given probe interval graph $G = (P, N, E)$, let $\tilde{T}$ be the canonical extended $\mathcal{MPQ}$-tree, and $G^+ = (P, N, E \cup E^+)$ be the corresponding enhanced interval graph. Let $\tilde{E}$ be the set of edges $\{v_1, v_2\}$ joining nonprobes v_1 and v_2 which is given by $\tilde{T}$; more precisely, we regard $\tilde{T}$ as an ordinary $\mathcal{MPQ}$-tree, and the graph $\tilde{G} = (P \cup N, E \cup E^+ \cup \tilde{E})$ is the interval graph given by $\tilde{T}$. Then $\tilde{T}$ can be computed in $O((|P| + |N|)(|E| + |E^+| + \left|\tilde{E}\right|))$ time and $O(|P| + |N|^2 + |E|)$ space.

Corollary 20 The graph isomorphism problem for the class of (enhanced) probe interval graphs is solvable in $O(n^3)$ time, where n is the number of vertices.

4 Applications

We show two applications of the canonical extended $\mathcal{MPQ}$-trees for probe interval graph. Given canonical extended $\mathcal{MPQ}$-tree $\tilde{T}$, using a standard depth first search technique, we can compute in linear time if each subtree in $\tilde{T}$ contains only nonprobes. Thus, hereafter, we assume that each section S_i knows if its subtree contains only nonprobes or not.

4.1 Relationship between nonprobes

First we consider the following problem:

Input: An enhanced probe interval graph $G^+ = (P, N, E \cup E^+)$ and the canonical extended $\mathcal{MPQ}$-tree $\tilde{T}$;

Output: Mapping f from each pair of nonprobes u, v with $\{u, v\} \notin E^+$ to "intersecting", "potentially intersecting", or "independent";

We denote by E_i and E_p the set of the pairs of intersecting nonprobes, and the set of the pairs of potentially intersecting nonprobes, respectively. That is, each pair of nonprobes u, v is either in E^+, E_i, E_p, or otherwise, they are independent.

Theorem 21 The sets E_i and E_p can be computed in $O(|E| + |E^+| + |E_i| + |E_p|)$ time for given enhanced probe interval graph $G^+ = (P, N, E \cup E^+)$ and the canonical extended $\mathcal{MPQ}$-tree $\tilde{T}$.

By Theorem 21, we can find the "best" nonprobe to fix the structure of the DNA sequence:

Corollary 22 For given enhanced probe interval graph $G^+ = (P, N, E \cup E^+)$ and the canonical extended $\mathcal{MPQ}$-tree $\tilde{T}$, we can find the nonprobe v that has the most potentially intersecting nonprobes in $O(|E| + |E^+| + |E_i| + |E_p|)$ time.

4.2 Enumeration of all affirmative interval representations

We next consider the following problem:

Input: A probe interval graph $G = (P, N, E)$ and the canonical extended $\mathcal{MPQ}$-tree $\tilde{T}$;

Output: All affirmative interval graphs.

Theorem 23 For given enhanced probe interval graph $G = (P, N, E)$ and the canonical extended $\mathcal{MPQ}$-tree $\tilde{T}$, all affirmative interval graphs can be enumerated in polynomial time of $|P| + |N| + |M|$, where M is the number of the affirmative interval graphs.

References

[1] K.S. Booth and G.S. Lueker. Testing for the Consecutive Ones Property, Interval Graphs, and Graph Planarity Using PQ-Tree Algorithms. *Journal of Computer and System Sciences*, 13:335–379, 1976.

[2] A. Brandstädt, V.B. Le, and J.P. Spinrad. *Graph Classes: A Survey.* SIAM, 1999.

[3] C.J. Colbourn and K.S. Booth. Linear Time Automorphism Algorithms for Trees, Interval Graphs, and Planar Graphs. *SIAM Journal on Computing*, 10(1):203–225, 1981.

[4] M.C. Golumbic. *Algorithmic Graph Theory and Perfect Graphs.* Academic Press, 1980.

[5] J.L. Johnson, R.M. McConnell, and J.P. Spinrad. Linear Time Recognition of Probe Interval Graphs. in preparation, 2002.

[6] J.L. Johnson and J.P. Spinrad. A Polynomial Time Recognition Algorithm for Probe Interval Graphs. In *Proc. 12th Ann. ACM-SIAM Symp. on Discrete Algorithms*, pages 477–486. ACM, 2001.

[7] N. Korte and R.H. Möhring. An Incremental Linear-Time Algorithm for Recognizing Interval Graphs. *SIAM Journal on Computing*, 18(1):68–81, 1989.

[8] G.S. Lueker and K.S. Booth. A Linear Time Algorithm for Deciding Interval Graph Isomorphism. *Journal of the ACM*, 26(2):183–195, 1979.

[9] R.M. McConnell and J.P. Spinrad. Construction of Probe Interval Models. In *Proc. 13th Ann. ACM-SIAM Symp. on Discrete Algorithms*, pages 866–875. ACM, 2002.

[10] T.A. McKee and F.R. McMorris. *Topics in Intersection Graph Theory.* SIAM, 1999.

[11] F.R. McMorris, C. Wang, and P. Zhang. On Probe Interval Graphs. *Discrete Applied Mathematics*, 88:315–324, 1998.

[12] T. Nagoya, R. Uehara, and S. Toda. Completeness of Graph Isomorphism Problem for Bipartite Graph Classes. In *IEICE Technical Report*, volume COMP2001-93, pages 1–5, 3/12 2002.

[13] P. Zhang. Probe Interval Graphs and Its Applications to Physical Mapping of DNA. manuscript, 1994.

[14] P. Zhang. Probe Interval Graph and Its Applications to Physical Mapping of DNA. RECOMB 2000, Poster Session; available at `http://recomb2000.ims.u-tokyo.ac.jp/Posters/list-posters.html`, 2000.

[15] P. Zhang. United States Patent. Method of Mapping DNA Fragments. [Online] Available `http://www.cc.columbia.edu/cu/cie/techlists/patents/5667970.htm`, July 3 2000.