Noname manuscript No.
(will be inserted by the editor)

Radiation Hybrid Map Construction Problem Parameterized

Chihao Zhang · Haitao Jiang · Binhai Zhu

Abstract In this paper, we study the Radiation Hybrid Map Construction (RHMC) problem which is about reconstructing a genome from a set of gene clusters. The problem is known to be NP-complete even when all gene clusters are of size two and the corresponding problem (RHMC_2) admits efficient constant-factor approximation algorithms. In this paper, for the first time, we consider the more general case when the gene clusters can have size either two or three (RHMC_3). Let p-RHMC be a parameterized version of RHMC where the parameter is the size of solution. We present a linear kernel for p-RHMC_3 of size $22k$ that when combined with a bounded search-tree algorithm, gives an FPT algorithm running in $O(6^k k + n)$ time. For p-RHMC_2 we present a bounded search tree algorithm which runs in $O^*(2.45^k)$ time, greatly improving the previous bound using weak kernels.

1 Introduction

Radiation hybrid (Rh) mapping is an earlier technique for mapping unique DNA sequences onto chromosomes and whole genomes. The achieved map of these DNA sequences provide a basis for association studies in modern genetics. The technique has been used since 1990 for construction maps of small chromosomal regions for human and several other mammals [5, 10, 11].

In Rh mapping experiments, chromosomes of the target organism are randomly broken into small DNA fragments through gamma radiation. The underlying mechanism is that, when two markers are physically close to each other on the chromosome, the probability that these two markers are broken down by the gamma radiation is low, and so with a high probability they are

---

Department of Computer Science, Shanghai Jiao Tong University, Shanghai, 200240, China. Email: chihao.zhang@gmail.com. · School of Computer Science and Technology, Shandong University, Jinan, Shandong, 250100, China. Email: htjiang@mail.sdu.edu.cn. · Department of Computer Science, Montana State University, Bozeman, MT, 59717-3880, USA. Email: bhz@cs.montana.edu.

either co-present in or co-absent from a DNA fragment. The *radiation hybrid map construction* (RHMC) problem is to determine the most likely linear order of the markers using the observed co-occurrences. We will formally define this problem in the next section.

Traditional Rh map construction methods are mostly heuristics, and often they are only able to produce framework maps on a small portion of all the markers [6]. Slonim et al. proposed a hidden Markov model on the Rh mapping data and used a maximum-likelihood approach to compute the map [11]; Givry et al. proposed to take advantage of known sequence information for target chromosomes for building more robust maps [3]. In [2], the RHMC_2 problem (i.e., each cluster contains two genes) was shown to be NP-hard and a 2-approximation algorithm was presented. The approximation ratio was then improved to 10/7 in [1].

In this paper, we study the problem under the framework of parameterized complexity. We restrict our attention to the case when the size of each cluster is at most 3. We show that this problem is Fixed-Parameter Tractable (FPT) by presenting a $22k$ kernel for it. Moreover, on top of the kernel, we present a bounded search-tree algorithm which runs in $O(6^k k + n)$ time. Furthermore, in the case when the size of each cluster is at most 2, we give an improved FPT algorithm which runs in $O^*(2.45^k)$ time.

This paper is organized as follows. In Section 2, we give some necessary definitions regarding the problem as well as FPT algorithms. In Section 3, we present the linear kernel for RHMC_3, where each cluster has at most three genes, and then give an FPT algorithm based on the kernel. In Section 4, we present an improved FPT algorithm for RHMC_2. Finally in Section 5, we conclude the paper with several open questions.

2 Preliminaries

Radiation Hybrid Map Construction Problem. Let Σ be a set of markers, and $\mathcal{C} = \{C_i \subseteq \Sigma : 1 \leq i \leq n\}$ be a set of clusters, with $|C_i| \leq d$. The problem is to decide whether after deleting some k clusters, there is a total order $\leq$ on Σ under which for every remaining cluster C, the markers in C are consecutive.

For example, take $\Sigma = \{a, b, c, d\}$ and $C_1 = \{b, c, d\}, C_2 = \{a, c, d\}, C_3 = \{a, b\}$. If none of C_1, C_2, C_3 is deleted, there is no total order on Σ satisfying our requirement. However, if we delete C_1, the order $c \leq d \leq a \leq b$ satisfies the condition since both $C_2 = \{a, c, d\}$ and $C_3 = \{a, b\}$ appear consecutively.

We denote this problem by RHMC. For fixed d, we call it $\mathsf{RHMC_d}$. It is easy to see that RHMC_2 is equivalent to the so-called Minimum co-Path problem: Given a simple undirected graph, can we delete some k edges, resulting a graph which is a disjoint union of paths? Minimum co-Path problem can be viewed as the complement of the Hamiltonian path problem, hence it is NP-hard [2].

Fixed-Parameter Tractable Algorithm. Let (I, k) be an instance of parameterized problem. An FPT algorithm decides (I, k) in time $f(k) \cdot n^c$, where f is an arbitrary computable function that only depends on k and c is a constant.

We often use the notation $O^*(f(k))$ to suppress the polynomial term. A basic approach towards FPT algorithm is to consider the *problem kernel*. Formally, a polynomial time algorithm K is a *kernelization* if it reduces the instance (I, k) to another instance (I', k') such that (1) (I, k) is a YES instance if and only if (I', k') is a YES instance, and (2) there is a computable function h such that $|I'| \leq h(k)$. The reduced instance (I', k') is called the *kernel*, and if h is a linear function, we say that the kernel is linear. It is well known that a parameterized problem is FPT if and only if it has a kernel. In many cases, the kernelization K consists of many reduction rules which reduce the size of the input instance and can be implemented in polynomial time. For more information on parameterized complexity and algorithms, one can refer to [4, 7, 9]. The parameterized version of $\mathsf{RHMC_d}$ is defined as follows:

p-$\mathsf{RHMC_d}$ PROBLEM

Input:	A set of clusters $\mathcal{C} = \{C_i \subseteq \Sigma : 1 \leq i \leq n\}$ with $|C_i| \leq d$, and an integer $k \in \mathbb{N}$.
Parameter:	k.
Problem:	Decide whether after deleting some k clusters, there is a total order $\leq$ on Σ under which for every remaining cluster C, the markers in C are consecutive.

3 A Linear Kernel for p-$\mathsf{RHMC_3}$

Let Σ be a set of markers, and $I = (\mathcal{C} = \{C_i \subset \Sigma : 1 \leq i \leq n\}, k)$ be an instance of p-$\mathsf{RHMC_3}$ such that $|C_i| \leq 3$ for all $1 \leq i \leq n$. We can rephrase p-$\mathsf{RHMC_3}$ as a problem on graphs:

Let $G(I) = (V(I), E(I))$ be an undirected graph. We consider each marker $c \in \Sigma$ as a vertex in $G(I)$. We add edge $\{u, v\}$ to $E(I)$ if $u, v \in C_i$ for some i and $u \neq v$. Since each cluster contains at most 3 markers, it can be viewed as a subgraph (a vertex, or two vertices on an edge, or a K_3) in the $G(I)$. We call a subgraph of $G(I)$ *legal* if it corresponds to some cluster. We say two clusters C_i, C_j are neighbors if $C_i \cap C_j \neq \varnothing$, i.e. their corresponding subgraphs share at least one vertex.

Then $\mathsf{RHMC_3}$ is equivalent to deciding whether one can remove a set $S(I)$ of k clusters from I such that there exists a set of disjoint paths $T(I)$ in $G(I)$ and for each remaining legal subgraph: (1) if it is a triangle, then it contains exactly two edges covered by some path in $T(I)$; (2) if it is an edge or a vertex, it belongs to some path in $T(I)$. We call $T(I)$ the *valid* set.

In Section 3.1 we will define the notion of *good* pattern and present a linear kernel in Section 3.2.

3.1 Good patterns

To ease the presentation, we first consider the case that all clusters are of size 3, i.e. all the legal subgraphs are triangles. We denote this problem by

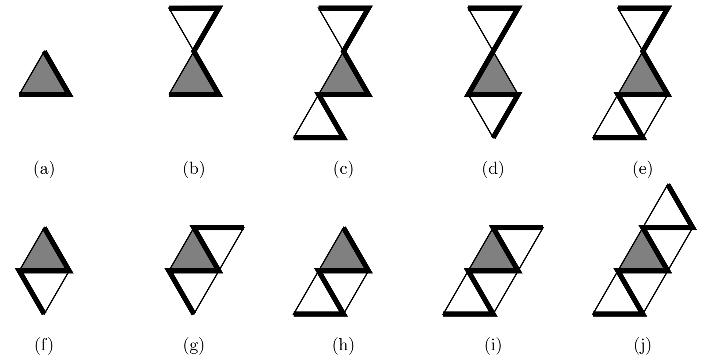

Fig. 1 10 good patterns of shaded cluster.

p-$\mathsf{RHMC_3}^*$. At the end of Section 3.2, we will come to the general case of p-$\mathsf{RHMC_3}$.

We first consider patterns across which one can draw a path. For a fixed cluster, consider all its neighbors, there are 10 such patterns as illustrated in Figure 1. (The fixed cluster is the one shaded and the thick line denote the path.)

A cluster C is *good* if its neighbors and itself form a pattern in Figure 1 where C is the shaded one; a good cluster is *free* if all of its neighbors are good. If a cluster is not good, we call it *bad*.

Given an instance I, if all clusters are good, then it is easy to solve this instance. If $G(I)$ contains more than one component, then we can consider each component independently. Hence in the following we assume that at least one cluster is bad and $G(I)$ is connected.

We immediately have the following lemma:

Lemma 1 *Let* $I = (\{C_i : 1 \leq i \leq n\}, 0)$ *be an instance of* $\mathsf{RHMC_3}^*$. *Define* $H(I) = (V_H(I), E_H(I))$, *where* $V_H(I) = \{C_i : 1 \leq i \leq n\}$ *and* $\{C_i, C_j\} \in E_H(I)$ *if and only if* $C_i \cap C_j \neq \varnothing$. *Then* I *is a YES instance if and only if (1) all the clusters are good (hence free) and (2)* $H(I)$ *contains no cycle.*

Proof The "only if" direction is straightforward, as we need to at least remove one cluster in bad pattern. Moreover, if $H(I)$ contains a cycle, we need to remove at least one cluster to break the cycle.

If all clusters are good, by recognizing all patterns in Figure 1, it is easy to verify that we can also connect two valid paths when attaching two good clusters. □

This lemma directly implies an FPT algorithm for p-$\mathsf{RHMC_3}^*$: For a bad cluster, consider a minimal bad pattern (removing any one of clusters in this pattern forms a good pattern) containing it and its neighbors. Since a good pattern contains at most 5 clusters (see (j) in Figure 1), this minimal bad pattern contains at most 6 clusters. We know at least one of these 6 clusters

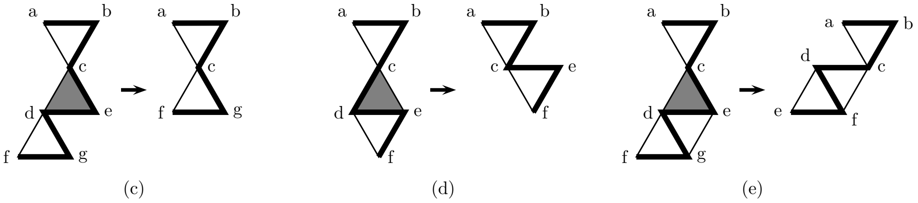

Fig. 2 Contraction of patterns (c)(d) and (e).

needs to be removed, hence it implies an $O^*(6^k)$ time FPT algorithm using the bounded search-tree method. After all the bad patterns are destroyed, we need to in addition break all the cycles.

Theorem 1 *p-$\mathsf{RHMC_3}^*$ can be solved in $O(6^k n)$ time.*

3.2 Kernelization algorithm

In this section, we present a kernelization algorithm for p-$\mathsf{RHMC_3}^*$. Let I be an instance. We first define an operation on good patterns called *contraction*. Let C be a good cluster of size 3, if its pattern is among (a)(b)(f)(g)(h)(i)(j) in Figure 1, then contracting C means removing C from I. Otherwise, if C's pattern is among (c)(d)(e), then by contraction we mean removing C from I and identifying two vertices of its neighbors, as depicted in Figure 2.

Our kernelization algorithm is exhaustively applying the following rule:

Rule: If there is a free cluster C in G, then contract it.

Lemma 2 *Let $I = (\mathcal{C}, k)$ be an instance of p-$\mathsf{RHMC_3}^*$ where $\mathcal{C} = \{C_i : 1 \leq i \leq n\}$, C be a free cluster. Let $I' = (\mathcal{C}', k)$ be the instance obtained from I by contracting C. Then I is a YES instance if and only if I' is a YES instance.*

Proof We first prove the "only if" direction: Assume that I is a YES instance. If C is of pattern (a)(b)(f)(g)(h)(i)(j), contracting C is equivalent to deleting C, hence a solution set of I is also a solution set of I'. Otherwise, let $S(I)$ and $T(I)$ be the solution set and valid set of I respectively. If $C \notin S(I)$, then $S(I)$ is also a solution set of I'. If $C \in S(I)$, we can consider the case that all neighbors of C are not in $S(I)$ without loss of generality, this is because if some neighbor $C' \in S(I)$, then the analysis is the same as in the pattern that C' is removed.

If C's pattern is (c) (d) (e), let $S(I') = S(I) \setminus \{C\} \cup \{C'\}$ where C' is the cluster $\{a, b, c\}$. It is easy to see $S(I')$ is a solution set of I', $T(I')$ is obtained from $T(I)$ by removing edges in C'.

To prove the "if" part we need more effort. Assume I' is a YES instance and let $S(I')$ be one of its solution of size at most k, we show that $S(I')$ is also a solution set of I.

We know that after removing $S(I')$ in $G(I')$, we can find a valid set $T(I')$. We claim that after restoring C in $G(I')$, either

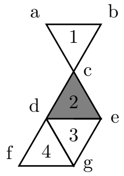

Fig. 3 One good pattern of C'.

(1) there exists a path in C that connects two paths in $T(I')$, or
(2) C already has two edges covered by some path in $T(I')$.

We first prove the claim. Since C is free, all its neighbors are good, and hence it is among the 10 patterns illustrated in Figure 1. Let C' be one of its neighbors, we justify the claim by examing the pattern of C'.

The pattern illustrated in Figure 3 encompasses all situations, where C' is cluster 2 (the one shaded). We analyse this pattern in detail and other patterns can be checked similarly.

a. If C is cluster 1, after contracting C, the path in $T(I')$ must be $P = cedgf$. Thus after restoring C, we can extend P with path cba or cab (depending on which one of a and b is identified with c while contracting).
b. If C is cluster 3, after contracting C, the path in $T(I')$ across this pattern may be $P_1 = abcedgf$, $P_2 = bacedgf$, $P_3 = abcedfg$ or $P_4 = bacedfg$. Restoring C does not affect P_1 and P_2. If it is the case of P_3 (resp. P_4), we can safely replace the path by $abcedgf$ (resp. $bacedgf$). This is because we only have two ways to force the existence of edge df in some path:
    (1) We have some cluster $\{d, f, x\}$ (x is not among $\{a, b, c, d, e, f, g\}$, but this cluster shares a vertex with cluster 2 and hence it is impossible in this pattern.
    (2) We have some cluster $\{g, x, y\}$ (x, y are not among $\{a, b, c, d, e, f, g\}$), but in this situation, cluster 3 is no longer good.
c. If C is cluster 4, after contracting C, the path in $T(I')$ across this pattern may be $P_1 = abcedg$, $P_2 = bacedg$, $P_3 = abcdeg$ or $P_4 = bacdeg$. Restoring C does not affect P_1 and P_2. If it is the case of P_3 (resp. P_3), we can safely replace the path by $abcedg$ (resp. $bacedg$). The reason is similar to case b above.

By the claim, if (1) happens, restoring C only extends some path in $T(I')$; if (2) happens, restoring C does not affect $T(I')$. Then we know $S(I')$ is also a solution set of I. □

Now we come to the general case of p-RHMC_3, i.e. some clusters may be of size 2.

We first generalize the operation of contraction in the following way: Let I be an instance and C be a 2-sized cluster. If there is another 3-sized cluster C' such that $C \subset C'$, then contracting C is equivalent to removing C. Similarly, for a 3-sized cluster C, if there is another 2-sized cluster C' such that $C' \subset C$,

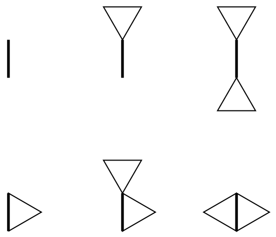

Fig. 4 Good patterns of 2-sized clusters such that all the neighbors are triangles.

then contracting C is equivalent to removing C. Otherwise, for a 2-sized cluster C, the operation is equivalent to contracting the edge in $G(I)$ and for a 3-sized cluster, the definition of contraction is the same as the case in p-$\mathsf{RHMC_3}^*$.

Secondly, in $\mathsf{RHMC_3}$, some new patterns are introduced, both for 2-sized clusters and 3-sized clusters. Figure 4 illustrates good patterns of a 2-sized cluster whose neighbors are triangles.

The analysis of free 2-sized clusters is similar to 3-sized ones. Consider the "only if" direction in the proof of Lemma 2, let C be a 2-sized cluster, it holds that

(1) if contracting C is equivalent to removing C, then $S(I) \setminus \{C\}$ is a solution set of I';
(2) otherwise, $S(I) \setminus \{C\} \cup \{C'\}$ is a solution set of I' where C' is the cluster incident to C, see Figure 4.

Now we consider the influence of 2-sized clusters to the "if" direction. It introduces a new way to force the existence of some edge in case (b) and (c). Taking (b) for instance, if edge *df* is some 2-sized cluster, then we know that we must keep it in the valid set. However, it is easy to check that, in this case, cluster *df* is not good as we cannot draw a path across its pattern. The similar argument holds for (c).

Therefore both Theorem 1 and Lemma 2 can be generalized to p-$\mathsf{RHMC_3}$.

Theorem 2 *p-$\mathsf{RHMC_3}$ can be solved in $O(6^k n)$ time.*

Lemma 3 *Let $I = (\{C_i : 1 \leq i \leq n\}, k)$ be an instance of p-$\mathsf{RHMC_3}$ and C a free cluster, I' be the instance obtained from I by contracting C. Then I is a YES instance if and only if I' is a YES instance.*

Lemma 4 *Let $I' = (\{C_i : 1 \leq i \leq n'\}, k)$ be the reduced instance of p-$\mathsf{RHMC_3}$ after exhaustively applying the reduction rule. If I' is a YES instance, then $n' \leq 22k$.*

Proof Since I' is a YES instance, there exists a set of clusters $S(I')$ with $|S(I')| \leq k$ such that after removing $S(I')$, all the clusters left are free. Let $\mathcal{L}$ be these free clusters left and $C \in S(I')$ be a cluster. Consider the set $\mathcal{L} \cup \{C\}$, some free cluster may become non-free. But this number is bounded by some

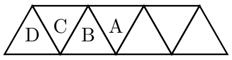

Fig. 5 A subgraph that every cluster is good.

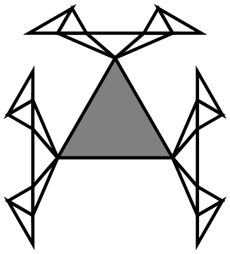

Fig. 6 The shaded cluster affects 21 clusters.

constant because C can only touch a constant number of clusters which form good pattern, and since these clusters are good before adding C, they can only have a constant number of neighbors.

The extreme case is constructed as follow: First consider the subgraph depicted in Figure 5, every cluster is good: (A) and (B) is of pattern (j), (C) is of pattern (i) and (D) is of pattern (h). Symmetrically, the three clusters to the right of (A) are also good.

However, if we add a bad cluster that share one vertex with (A) then all these seven clusters become non-free. Figure 6 shows the extreme case in which adding the shaded cluster can change at most 21 clusters (i.e., three copies of the subgraph in Figure 5) from free to non-free ones. Therefore, the total number of clusters in I' does not exceed $22k$. $\square$

Theorem 3 p-RHMC_3 *has a kernel of size* $22k$.

Remark 1 The proof of Lemma 2 implies that there is a solution set which contains no contracted cluster. Thus the $22k$ kernel directly implies a 22-approximation algorithm for RHMC_3, i.e. choose all clusters corresponding to ones in the kernel as solution.

Combining Theorem 2 and Theorem 3, we have

Corollary 1 p-RHMC_3 *can be solved in time* $O(k \cdot 6^k + n)$.

4 An FPT Algorithm for p-RHMC_2

In [8], a $5k$ weak kernel (loosely speaking, parameterized search space) is constructed for p-RHMC_2. That immediately implies an FPT algorithm which runs in $O^*(\binom{5k}{k}) = O^*(2^{3.61k})$ time. Here we present an FPT algorithm which runs in $O^*(2.45^k)$ time, using the well-known bounded search-tree method.

First notice that RHMC_2 problem is equivalent to Minimum co-Path Set problem.

Given a simple undirected graph G, a *co-path set* is a set S of edges in G whose removal leaves a graph in which every connected component is a path. And the problem is to decide whether there exists a co-path set of size k.

For an instance of RHMC_2, we let $V(G)$ be the set of markers Σ, and $\{u,v\} \in E(G)$ if there is some cluster $C_i = \{u,v\}$. It is then easy to verify these two problems are equivalent.

Hence in the following, we describe the algorithm in term of Minimum co-Path Set problem.

We start with a simple lemma.

Lemma 5 *If there are two edges $e_1 = (u,v)$ and $e_2 = (v,w)$ in G, where $d(u) = d(w) = 3$ and $d(v) = 2$, then there exists an optimal solution for RHMC_2 which does not delete e_1 and e_2.*

Proof Let $N(u) = \{v,u',u''\}$ and $N(w) = \{v,w',w''\}$. It suffices to prove that if e_1 or e_2 (or both) is deleted in some optimal solution S, then we can replace e_1 with one edge from (u,u') and (u,u'') or replace e_2 with one of (w,w') and (w,w'') (or both) to obtain another optimal solution which is at least as good as S. Suppose that e_1 is deleted in some optimal solution S for RHMC_2, then after all the edges in S are deleted, v is the end of some path P. We consider three cases (see Figure 7).

(1) If $u' \in P$ and $u'' \notin P$ (or vice versa), which means (u,u') (resp. (u,u'')) is also deleted for S, then replace e_1 with this deleted edge (u,u') (resp. (u,u'')). Clearly, P is replaced by a new path of the same length.

(2) If $u' \in P$ and $u'' \in P$, then we can assume that $u \in P$ (otherwise, we can add e_1 back to P obtain a longer path whose endpoint is u). Consequently, the path is either in the form $\langle v,\cdots,u',u,u'',\cdots\rangle$ or $\langle v,\cdots,u'',u,u',\cdots,\rangle$. We can replace e_1 with (u,u') in the former case or replace e_1 with (u,u'') in the latter case, to obtain a new path with the same size as P.

(3) If $u' \notin P$ and $u'' \notin P$, then we can replace e_1 with either (u,u') or (u,u'') to have a new solution of the same size as that of S. The argument for e_2 is similar hence omitted.

$\square$

Corollary 2 *If there is a path $P = \langle u,v_1,\cdots,v_k,w\rangle$ in G with $d(u) = d(w) = 3$, and $d(v_i) = 2$ for all $1 \le i \le k$, then there is an optimal solution for RHMC_2 in which all the edges along the path P are reserved (i.e., not deleted).*

Lemma 6 *Given a vertex v, if $N(v) = \{v_1,v_2,v_3\}$ and $d(v_1) = d(v_2) = d(v_3) = 2$, then there exists an optimal solution for RHMC_2 which deletes either (v,v_1) or (v,v_2).*

Proof Assume to the contrary that the optimal solution does not delete (v,v_1) and (v,v_2), instead it deletes (v,v_3). Then due to the fact that $d(v_3) = 2$ before

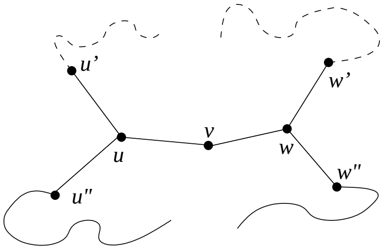

Fig. 7 Illustration for the proof of Lemma 5.

the deletion, v_3 is the endpoint of some path P. We have two cases. (1) If $v_1 \in P$ and $v_2 \in P$, then the path is either in the form $\langle v_3, \cdots, v_1, v, v_2, \cdots, \rangle$ or in the form $\langle v_3, \cdots, v_2, v, v_1, \cdots, \rangle$, and we can replace (v, v_3) with (v, v_1) in the former case or replace (v, v_3) with (v, v_2) in the latter case. (2) If $v_1 \notin P$ and $v_2 \notin P$, we can replace (v, v_3) with either (v, v_1) or (v, v_2). In both cases we obtain an alternative optimal solution for RHMC_2. □

```
Algorithm Bounded Search Co-path Set
Input: Graph $G$, integer $k$
Output: A Co-path set $S$ of size $k$
1    While there exists a vertex $v$ such that $d = d(v) \geq 4$, choose all but two of its incident edges and put them in $S$.
2    While there exists a vertex $v$ satisfying Lemma 6, choose one of its two incident edges and put it in $S$ following Lemma 6.
3    While there exists vertices $u, v, w$ satisfying Lemma 5, reserve edges $(u, v)$ and $(v, w)$ following Lemma 5.
4    For a path $P = \langle u, v, w \rangle$, where $N(v) = \{u, w, a\}$, $N(u) = \{v, b, c\}$, and $N(w) = \{v, g, h\}$
     4.1    add $(u, v)$ and $(w, g)$ to $S$;
     4.2    or add $(u, v)$ and $(w, h)$ to $S$;
     4.3    or add $(u, b)$ and $(v, w)$ to $S$;
     4.4    or add $(u, b)$ and $(v, a)$ to $S$;
     4.5    or add $(u, c)$ and $(v, w)$ to $S$;
     4.6    or add $(u, c)$ and $(v, a)$ to $S$.
5    Repeat Steps 2,3,4 until every vertex has degree less than 3.
6    Choose an arbitrary edge from each cycle and put it in $S$.
7    Return $S$.
```

Theorem 4 *Algorithm* Bounded Search Co-path Set *computes a co-path set in* $O^*(6^{k/2}) \approx O^*(2.45^k)$ *time.*

Proof Step 1 has a recurrence relation

$$f(k) = \binom{d}{d-2} f(k - (d-2)),\ d \geq 4.$$

Step 2 has a recurrence relation

$$f(k) = 2f(k-1).$$

Step 3 has a recurrence relation

$$f(k) = 4f(k-2).$$

Step 4 branches on whether (u, v) is deleted or not. If (u, v) is deleted, then (v, w) is reserved and either (w, g) or (w, h) is also deleted; if (u, v) is reserved then at least one of (u, b) and (u, c), as well as at least one of (v, w) and (v, a) are deleted. So step 4 has a recurrence relation

$$f(k) = 6f(k-2).$$

$f(k)$ achieves its maximum value when $d = 4$ or $f(k) = 6f(k-2)$, so we have $f(k) \leq O^*((6)^{k/2}) \approx O^*(2.45^k)$. $\square$

5 Concluding Remarks

In this paper, we studied the Radiation Hybrid Map Construction problem using parameterized algorithms. For p-RHMC_3, where each gene cluster contains at most three genes, we showed an FPT algorithm based on a linear kernel of it. For p-RHMC_2, we presented a bounded search-tree algorithm which runs in $O^*(2.45^k)$ time, greatly improving the previous bound using weak kernels. An important open question is whether one can extend these methods to handle p-RHMC_d, where each gene cluster contains at most d genes. Furthermore, does the generalized version p-RHMC remain FPT? Does it have a (small) kernel?

Acknowledgments

This research is partially supported by NSF of China under project 60928006, 60970011 and 61202014, and by the Opening Fund of Top Key Discipline of Computer Software and Theory in Zhejiang Provincial Colleges at Zhejiang Normal University.

References

1. Z-Z. Chen, G. Lin, and L. Wang. An approximation algorithm for the minimum co-path set problem. *Algorithmica*, 60(4):969–986, 2011.
2. Y. Cheng, Z. Cai, R. Goebel, G. Lin, and B. Zhu. The radiation hybrid map construction problem: recognition, hardness, and approximation algorithms. *Unpublished Manuscript*, 2008.
3. S. De Givry, M. Bouchez, P. Chabrier, D. Milan, and T. Schiex. Carh ta Gene: multipopulation integrated genetic and radiation hybrid mapping. *Bioinformatics*, 21(8):1703, 2005.
4. R.G. Downey and M.R. Fellows. *Parameterized complexity.* Springer New York, 1999.
5. D.R.Cox, M.Burmeister, E.R.Price, S.Kim, and R.M.Myers. Radiation hybrid mapping: a somatic cell genetic method for constructing high resolution maps of mammalian chromosomes. *Science*, 250(4978):245–250, 1990.
6. T. Faraut, S. De Givry, P. Chabrier, T. Derrien, F. Galibert, C. Hitte, and T. Schiex. A comparative genome approach to marker ordering. *Bioinformatics*, 23(2):e50–e56, 2007.
7. J. Flum and M. Grohe. *Parameterized complexity theory.* Springer-Verlag New York, 2006.

8. H. Jiang, C. Zhang, and B. Zhu. Weak kernels. *Electronic Colloquium on Computational Complexity*, ECCC Report TR10-005, October 2010.
9. R. Niedermeier. *Invitation to fixed-parameter algorithms.* Oxford University Press, USA, 2006.
10. C.W. Richard et al. A radiation hybrid map of the proximal long arm of human chromosome 11 containing the multiple endocrine neoplasia type 1 (MEN-1) and bcl-1 disease loci. *American Journal of Human Genetics*, 49(6):1189–1196, 1991.
11. D. Slonim, L. Kruglyak, L. Stein, and E. Lander. Building human genome maps with radiation hybrids. *Journal of Computational Biology*, 4(4):487–504, 1997.