Gene trees and species trees
the gene duplication problem is fixed-parameter tractable

Gene Trees and Species Trees: The Gene-Duplication Problem is Fixed-Parameter Tractable

Ulrike Stege
Computational Biochemistry Research Group
ETH Zürich, Ch-8092 Zürich, Switzerland
stege@inf.ethz.ch

April 21, 1999

Abstract

The GENE-DUPLICATION PROBLEM is the problem of computing the optimal species tree for a given set of gene trees under the GENE-DUPLICATION MODEL (first introduced by Goodman et al.). The problem is known to be *NP*-complete. We give a fixed-parameter tractable algorithm solving the problem parameterized by the number of gene duplications necessary to rectify the gene trees with the species tree.

1 Introduction

When trying to resolve the *tree of life* one usually wants to compute the phylogenetic relationships between the organisms based on the data provided by the DNA or protein sequences of families of homologous genes. A *species tree* or *evolutionary tree* for a given set of taxa is a complete rooted binary tree built over the set of taxa representing the phylogenetic relationships between the taxa. A *gene tree* is a complete rooted binary tree formed over a family of homologous genes for the set of taxa. Gene trees for different gene families do not necessarily agree [GCMRM79, F88, BE90, P94]. The problem we consider is the determination of the (correct) species tree for a set of taxa given a set of (possibly contradictory) gene trees. Several models for attacking the problem have appeared in the literature including the famous *maximum agreement subtree* (MAST) [FM67, NA86, PC97]. The known models, which are related to agreement subtrees (see also [HJWZ95, HJWZ96] amongst others) or consensus trees (see for example [S91]), are mathematical models but biologically rather meaningless.

A biological cost model which has recently received considerable attention is the GENE DUPLICATION AND LOSS MODEL suggested in biological terms in [GCMRM79] and discussed in [P94, GMS96, MLZ98, Z97]. The basic idea here is to measure the similarity/dissimilarity between a set of gene trees by counting the number of postulated *paralogous gene duplications* and subsequent *gene losses* required to explain (in an evolutionarily meaningful way) how the gene trees could have arisen with respect to the species tree.

Gene Duplication and Loss Problem
Input: Gene trees $G_1, \ldots, G_k$
Output: Species tree S, such that the cost (gene duplications and losses) for rectifying $G_1, \ldots, G_k$ with S is minimized.

If the species tree is given the minimum cost for rectification is computable in linear time [Z97]. Otherwise, the problem is proven to be NP-complete.

Gene-Duplication Problem
Input: Gene trees $G_1, \ldots, G_k$
Output: Species tree S, such that the gene-duplication cost for rectifying $G_1, \ldots, G_k$ with S is minimized.

Note that solving the Gene-Duplication Problem might be very useful for detecing paralogous gene duplications in the databases when the gene-duplication rate is low. When gene duplication events are more frequent a helpful variant of the problem could be the Multiple-Gene-Duplication Problem [GMS96, FHS98] where the gene duplications of different gene trees are not necessarily independent events. Here we ask for the species tree which implies the minimum number of *multiple gene duplications* for a given set of gene trees. When the species tree is given the problem has been shown to be both *NP*-complete and *W*[1]-hard parameterized by the number of speciation events and restricted to two gene trees [FHS98].

In this paper we present a fixed-parameter tractable algorithm solving the Gene-Duplication Problem parameterized by the duplication cost.

2 A Short Introduction in Parameterized Complexity

The technique we use for constructing the fixed-parameter tractable algorithm presented in Section 5 (the *technique of bounded search trees*) is a fundamental one in the framework of parameterized complexity introduced in [DF95a, DF95b, DF95c, DF98]. This framework has become a routine tool to some extent, but not to the point where we feel it can be assumed to be completely familiar, and so we offer this brief summary.

The goal in parameteric complexity analysis is to identify small but still useful ranges of a *parameter* k for a hard problem, and see if for instances of size n the problem can be solved in time $f(k)n^\alpha$ for some constant α independent of the parameter. (Typically, when this is possible, α is small, just as with polynomial time complexity.) This good behavior is termed *fixed-parameter tractability* and can be viewed as a generalization of *P*-time. For surveys about fixed-parameter tractability see [DFS99a, DFS99b].

A canonical example that exhibits the issue is the difference between the best known algorithms for Vertex Cover and Dominating Set, taking in both cases the parameter to be the number of vertices in the set. The best known algorithm for Vertex Cover, after several rounds of improvement in the parameter function, runs in time $O((1.29175)^k k^2 + nk)$ [NR98]. Note that the exponential dependence on the parameter k in the last expression is *additive*, so that Vertex Cover is well-solved for input of any size so long as k is no more than around 60. Both problems are *NP*-complete, of course, and so the *P*-time *versus* *NP*-complete framework is unable to detect the significant qualitative difference between these two problems displayed in Table 1.

	$n = 50$	$n = 100$	$n = 150$
$k = 2$	625	2,500	5,625
$k = 3$	15,625	125,000	421,875
$k = 5$	390,625	6,250,000	31,640,625
$k = 10$	1.9×10^{12}	9.8×10^{14}	3.7×10^{16}
$k = 20$	1.8×10^{26}	9.5×10^{31}	2.1×10^{35}

Table 1: The ratio $\frac{n^{k+1}}{2^k n}$ for various values of n and k.

For parameterized complexity, the analog of *NP*-hardness is hardness for $W[1]$. The analogy is very strong, since the k-Step Halting Problem for Nondeterministic Turing Machines is complete for $W[1]$ [CCDF97], a result which is essentially a miniaturization of Cook's Theorem. Dominating Set is hard for $W[1]$ and is therefore unlikely to be fixed-parameter tractable. Indeed, one senses a "wall of brute force" (try all k-subsets) inherent complexity in this problem, much as one does for the k-Step Halting Problem, where a significant improvement on the obvious $O(n^k)$ algorithm seems fundamentally unlikely, and much as one senses a wall of brute force ("try all 2^n truth assignments") in the complexity of *NP*-complete problems such as satisfiability.

3 The Gene-Duplication Model

In this section we give a short reminding introduction about the Gene-Duplication Model which is mathematically well formalized in [Z97] and discussed in [MLZ98, FHS98].

3.1 Notation

All trees in this paper (gene trees and species trees included) are rooted, binary, and leaf labeled. Let $T = (V, E, L)$ be such a tree where V is the vertex set, E is the edge set, and $L \subseteq V$ is the leaf-label set (in short, *leafset*).

For a vertex $u \in V - L$, let T_u be the subtree of T rooted by u. The root of each tree T has a left and a right subtree, rooted by the two kids of the root *root*(T) and denoted by T_l and T_r.

We denote the leafset L of T as $L(T)$, and for a node $u \in V$ we denote the leafset of tree T_u short with $L(u)$ instead of $L(T_u)$.

For trees $T_1 = (V_1, E_1, L)$, $T_2 = (V_2, E_2, L)$, and a vertex $v \in V_1$ let $lca_{T_2}(L(v))$ be the least common ancestor of all the leaves in $L(v)$ in tree T_2.

3.2 The Model

Let $G = (V_G, E_G, L)$ be a gene tree and $S = (V_S, E_S, L')$, $L \subseteq L'$, be a species tree.

We use a function $loc_{G,S} : V_G \to V_S$ to associate each vertex in G with a vertex in S. Furthermore, we use a function $event_{G,S} : V_G \to \{dup, spec\}$ to indicate whether the event in G corresponds to a duplication or speciation event.

The function M given below maps a gene tree G into a species tree S by defining functions $loc_{G,S}$ and $event_{G,S}$. The quantity $cost(G,S) = |\{u | u \in V_G - L, event_{G,S}(u) = dup\}|$ is the minimum number of gene-duplication events necessary to rectify the gene tree G with the species tree S (cf. [MLZ98]).

$M(G,S)$: for each $u \in V_G - L$, $loc(u) = lca_S(L(u))$ and

$$event(u) = \begin{cases} spec & \text{if } loc_{G,S}(u') \neq loc_{G,S}(u), \text{for all } u' \text{ where } u' \text{ is a kid of } u \text{ in } G. \\ dup & \text{otherwise} \end{cases}$$

Furthermore for given $G_1, G_2, \ldots, G_k$, and S let $cost(G_1, G_2, \ldots, G_k, S) = \sum_{i=1}^{k} cost(G_i, S)$.

We restate the

GENE-DUPLICATION PROBLEM
Input: Set of gene trees $G_1, \ldots, G_k$.
Output: Species tree S with $cost(G_1, G_2, \ldots, G_k, S)$ is minimized?

We comment that M (cf. [Z97]) is *just* a least common ancestor mapping which elegantly clarifies the rather complicated combinatorics describing the GENE-DUPLICATION MODEL in the original papers.

4 Properties of the Gene-Duplication Model

In this section we introduce a generalized version of the GENE-DUPLICATION MODEL, the GENE-DUPLICATION MODEL FOR SPLITS. The more generalized version leads us to properties useful for attacking the GENE-DUPLICATION PROBLEM.

Definition 1 *Given a leafset L, we call $\mathcal{D} = (\mathcal{D}_l | \mathcal{D}_r)$ a* split *of L if for $\mathcal{D}' = \mathcal{D}_l, \mathcal{D}_r$*

1. *either $\mathcal{D}'$ is a split of $L \setminus (\bigcup L(\mathcal{D}'))$ or $\mathcal{D}' \subseteq L$.*
2. $\bigcup L(\mathcal{D}_l) \cap \bigcup L(\mathcal{D}_r) = \emptyset$.

Here

$$L(\mathcal{D}_i) = \begin{cases} \mathcal{D}_i & \textit{if } \mathcal{D}_i \textit{ is leafset} \\ L(\mathcal{D}_{il}) \cup L(\mathcal{D}_{ir}) & \textit{if } \mathcal{D}_i \textit{ is a split } (\mathcal{D}_i = (\mathcal{D}_{il} | \mathcal{D}_{ir})) \end{cases}.$$

Definition 2 *Given a split $\mathcal{D} = (\mathcal{D}_l|\mathcal{D}_r)$. We define*

$$\mathcal{L}(\mathcal{D}) = \begin{cases} \{\mathcal{D}\} & \textit{if } \mathcal{D} \textit{ is leafset} \\ \mathcal{L}(\mathcal{D}_l) \cup \mathcal{L}(\mathcal{D}_r) & \textit{otherwise} \end{cases} .$$

$\mathcal{L}(\mathcal{D})$ (or short $\mathcal{L}$) is called the leafset of split *$\mathcal{D}$.*

Definition 3 *Given a leafset L and a split $\mathcal{D}$ of L, we call $\mathcal{D}$* complete *if $L(\mathcal{D}) = L$ and* incomplete *otherwise.*

Note that for a given leafset L each split $\mathcal{D}$ defines a (rooted binary) tree over leafset $\mathcal{L}$ of $\mathcal{D}$. A complete split $\mathcal{D}$ with $|D| = 1$ for each $D \in \mathcal{L}(\mathcal{D})$ corresponds to a *species tree.*

Definition 4 *Suppose that for a leafset L we are given splits $\mathcal{D} = (\mathcal{D}_l|\mathcal{D}_r)$, and $\mathcal{D}' = (\mathcal{D}'_l|\mathcal{D}'_r)$ with leafsets $\mathcal{D}_l, \mathcal{D}_r\mathcal{D}'_l, \mathcal{D}'_r$. We call $\mathcal{D}'$ a* subsplit *of $\mathcal{D}$ if $\mathcal{D}_l \subseteq \mathcal{D}'_l$ and $\mathcal{D}_r \subseteq \mathcal{D}'_r$.*

Analogous to the least common ancestor in trees we define the least common ancestor in splits.

Definition 5 *Suppose we have a gene tree $G = (V, E, L)$ and a split $\mathcal{D}$ of L. For any node $u \in V$ define the least common ancestor $lca_{\mathcal{D}}(L(u))$ to be the least common ancestor in the tree defined by $\mathcal{D}$ and $\mathcal{L}$. Note that if $L(u) \subseteq D$ for an element $D \in \mathcal{L}$ then $lca_{\mathcal{D}}(L(u)) = D$.*

The cost of a gene tree and a split is defined similarily to the cost of a gene tree and a species tree and is a generalization of the definition of the Gene-Duplication Model.

Definition 6 Gene-Duplication Model for Splits *Let $G = (V_G, E_G, L)$ be a gene tree and $\mathcal{D} = (\mathcal{D}_l|\mathcal{D}_r)$ be a split of L, $\mathcal{D}_r, \mathcal{D}_l \neq \emptyset$. The function $loc_{G,\mathcal{D}} : V_G \to L(\mathcal{D})$ associates each vertex in G with a set in $L(\mathcal{D})$. The function $event_{G,\mathcal{D}} : V_G \to \{dup, unknown\}$ indicates whether the event in G caused by the split corresponds necessarily to a duplication or not.*

Furthermore we define

$M(G, \mathcal{D})$: for each $u \in V_G - L$, $loc(u) = lca_{\mathcal{D}}(L(u))$ and

$$event(u) = \begin{cases} unknown & \textit{if } (loc_{G,\mathcal{D}}(u) \in \mathcal{L}) \textit{ or } (loc_{G,\mathcal{D}}(u') \neq loc_{G,\mathcal{D}}(u), \textit{for all } u' \\ & \textit{where } u' \textit{ is a kid of } u \textit{ in } G.) \\ dup & \textit{otherwise} \end{cases}$$

Let $Dups(G, \mathcal{D}) = \{u | u \in V_G - L, event_{G,\mathcal{D}}(u) = dup\}$ and $cost(G, \mathcal{D}) = |Dups(G, \mathcal{D})|$.

Definition 7 *Suppose we have k gene trees $G_1, \ldots G_k$ and a split $\mathcal{D}$. We define the k-dimensional vector*

$$\vec{c} = [cost(G_1, \mathcal{D}), cost(G_2, \mathcal{D}), \ldots, cost(G_k, \mathcal{D})]$$

the cost vector *of* $G_1, G_2 \ldots, G_k$ *and* $\mathcal{D}$. *Furthermore define* $|\vec{c}| = \sum_{i=1}^{k} cost(G_i, \mathcal{D})$ *and let*

$$Dups(G_1, \ldots, G_k, \mathcal{D}) = \bigcup_{i=1}^{k} Dups(G_i, \mathcal{D}).$$

Definition 8 *1. Given gene trees* $G_1, G_2, \ldots, G_k$ *and a species tree* S. *We call* S optimal *if* $cost(G_1, G_2, \ldots, G_k, S_0)$ *is minimized over all possible species trees.*

2. For given gene trees $G_1, G_2, \ldots, G_k$ *and a split* $\mathcal{D}$ *we call a species tree* S_0 optimal depending on $\mathcal{D}$ *if* $cost(G_1, G_2, \ldots, G_k, S_0)$ *is minimized over all possible species trees* S *who are a subsplit of* $\mathcal{D}$.

The following two straightforward observations and Lemma 1 provide the main ingredients for the fixed-parameter tractable algorithm described in Section 5.

Observation 1 *Suppose we are given a leafset* L *and gene trees* $G_1, \ldots, G_k$, *where* $L(G_i) \subseteq L$ $(i = 1, \ldots, k)$. *Let* $\mathcal{D}$ *be a complete split of* L *with leafsets* $\mathcal{D}_l$ *and* $\mathcal{D}_r$. *Furthermore let* S *be an optimal species tree depending on* $\mathcal{D}$. *Then* $Dups(G_1, \ldots, G_k, \mathcal{D}) \subseteq Dups(G_1, \ldots, G_k, S)$ *and* $cost(G_1, \ldots, G_k, \mathcal{D})$ *is exactly the number of duplications located at the root of* S.

Observation 2 *Suppose we are given a leafset* L *and gene trees* $G_1, \ldots, G_k$, *where* $L(G_i) \subseteq L$ $(i = 1, \ldots, k)$. *Let* $\mathcal{D}$, $\mathcal{D}'$ *be complete splits of* L *with leafsets* $\mathcal{D}_l$, $\mathcal{D}_r$, $\mathcal{D}'_l$, *and* $\mathcal{D}'_r$. *Let* S *be an optimal species tree depending on* $\mathcal{D}$ *and let* S' *be the optimal species tree depending on* $\mathcal{D}'$.

If $Dups(G_1, \ldots, G_k, \mathcal{D}) \subseteq Dups(G_1, \ldots, G_k, \mathcal{D}')$ *then*

$$cost(G_1, G_2, \ldots, G_k, S) \leq cost(G_1, G_2, \ldots, G_k, S').$$

Lemma 1 *Suppose we are given a leafset* L *and gene trees* G *and* H. *Let* $L(G), L(H) \subseteq L$ *and let* $\mathcal{D}$ *be an incomplete split of* L *with leafsets* $\mathcal{D}_l$, $\mathcal{D}_r$, $\mathcal{D}_l, \mathcal{D}_r \neq \emptyset$. *Then either we can supplement the split to a complete split without increasing the cost (this subsplit we call a* completion*) or there are leaves* $a, b \in (L - L(\mathcal{D}))$ *such that each of the 4 possibilities for building a split* $\mathcal{D}'$ *of* $(L(\mathcal{D}) \cup \{a, b\})$, *with* $\mathcal{D}'$ *a subsplit of* $\mathcal{D}$, *increases the cost, that is,* $cost(G, H, \mathcal{D}') > cost(G, H, \mathcal{D})$. *We call* (a, b) *a* candidate pair.

Proof: Let $E_1, \ldots, E_m \in Dups(G, \mathcal{D})$ be the *lowest* duplication nodes of G, i.e., for E_i there does not exist an element $X \in V_{G_{E_i}}$, $X \neq E_i$, and $X \in Dups(G, \mathcal{D})$ $(i = 1 \ldots m)$. Analogously, let $F_1, \ldots, F_l \in Dups(H, \mathcal{D})$ be the lowest duplication nodes of H.

Furthermore, let $e_1, e_2, \ldots, e_{4m}$ be the roots of all the left and the right subtrees of the kids of $E_1, \ldots, E_m$, and let $f_1, f_2, \ldots, f_{4l}$ be the roots of all the left and the right subtrees of the kids of $F_1, \ldots, F_l$.

For each e_i we define $\alpha_i = e_i \cap L(\mathcal{D})$, and for each f_j we define $\beta_j = f_j \cap L(\mathcal{D})$. Obviously, either $\alpha_i \subseteq \mathcal{D}_l$ or $\alpha_i \subseteq \mathcal{D}_r$, and either $\beta_j \subseteq \mathcal{D}_l$ or $\beta_j \subseteq \mathcal{D}_r$.

If there exists a completion of the split $\mathcal{D}$ then there is a completion $\mathcal{D}' = (\mathcal{D}'_l|\mathcal{D}'_r)$ having the following form:

$$\mathcal{D}'_l = \{a \in L(e_i)|\alpha_i \subseteq \mathcal{D}_l, 1 \leq i \leq 4m\} \cup \{a \in f_j|\beta_j \subseteq \mathcal{D}_l, 1 \leq j \leq 4l\},$$

$$\mathcal{D}'_r = \{a|a \notin \mathcal{D}'_l, a \in L(e_i), \alpha_i \subseteq \mathcal{D}_r, 1 \leq i \leq 4m\} \cup \{a|a \notin \mathcal{D}'_l, a \in L(f_j), \beta_j \subseteq \mathcal{D}_r, 1 \leq j \leq 4l\}.$$

Note that for every completion $\mathcal{D}' = (\mathcal{D}'_l|\mathcal{D}'_r)$

$$(\{a \in L(e_i)|\emptyset \neq \alpha_i \subseteq \mathcal{D}_l, 1 \leq i \leq 4m\} \cup \{a \in L(f_j)|\emptyset \neq \beta_j \subseteq \mathcal{D}_l, 1 \leq j \leq 4l\}) \subseteq \mathcal{D}'_l,$$

$$(\{a \in L(e_i)|\emptyset \neq \alpha_i \subseteq \mathcal{D}_r, 1 \leq i \leq 4m\} \cup \{a \in L(f_j)|\emptyset \neq \beta_j \subseteq \mathcal{D}_r, 1 \leq j \leq 4l\}) \subseteq \mathcal{D}'_r.$$

We show, if there is no pair of leaves wich increases the cost in all possible subsplits then $\mathcal{D}$ has a completion.

Let $a, b \in (L - L(\mathcal{D}))$. Then one of the following cases is true.

I There is an e_i with $a, b \in L(e_i)$.

II There are e_{i_1}, e_{i_2} with $a \in L(e_{i_1}), b \in L(e_{i_2})$ $(e_{i_1} \neq e_{i_2})$.

Furthermore either

A there is an f_j with $a, b \in L(f_j)$ or

B there are f_{j_1}, f_{j_2} with $a \in L(f_{j_1}), b \in L(f_{j_2})$ $(f_{j_1} \neq f_{j_2})$.

When is a pair (a, b) not a candidate pair? I.e., when does (a, b) not imply increasing the cost in all possible subsplits? We consider all cases.

IA There is an e_i with $a, b \in L(e_i)$ and there is an f_j with $a, b \in L(f_j)$.

Then (a, b) is not a candidate pair if and only if $(\emptyset \neq L(\alpha_i) \subseteq L(\mathcal{D}_l)$ and $\emptyset \neq L(\beta_j) \subseteq L(\mathcal{D}_l))$ or $(\emptyset \neq L(\alpha_i) \subseteq L(\mathcal{D}_r)$ and $\emptyset \neq L(\beta_j) \subseteq L(\mathcal{D}_r))$.

IB There is an e_i with $a, b \subseteq L(e_i)$ and there are f_{j_1}, f_{j_2} with $a \in L(f_{j_1}), b \in L(f_{j_2})$ $(f_{j_1} \neq f_{j_2})$.

Then (a, b) is not a candidate pair if and only if $\emptyset \neq L(\alpha_i), L(\beta_{j_1}), L(\beta_{j_2}) \subseteq L(\mathcal{D}_l)$ or $\emptyset \neq L(\alpha_i), L(\beta j_1), L(\beta_{j_2}) \subseteq L(\mathcal{D}_r)$.

IIA There are e_{i_1}, e_{i_2} with $a \in L(e_{i_1}), b \in L(e_{i_2})$ $(e_{i_1} \neq e_{i_2})$ and there is an f_j with $a, b \in L(f_j)$.

Then (a, b) is not a candidate pair if and only if $\emptyset \neq L(\alpha_{i_1}), L(\alpha_{i_2}), L(\beta_j) \subseteq L(\mathcal{D}_l)$ or $\emptyset \neq L(\alpha_{i_1}), L(\alpha_{i_2}), L(\beta_j) \subseteq L(\mathcal{D}_r)$.

IIB There are e_{i_1}, e_{i_2} with $a \in L(e_{i_1}), b \in L(e_{i_2})$ $(e_{i_1} \neq e_{i_2})$ and there are f_{j_1}, f_{j_2} with $a \in L(f_{j_1}), b \in L(f_{j_2})$ $(f_{j_1} \neq f_{j_2})$.

Then (a, b) is not a candidate pair if and only if $\emptyset \neq L(\alpha_{i_1}), L(\alpha_{i_2}), L(\beta_{j_1}), L(\beta_{j_2}) \subseteq L(\mathcal{D}_l)$ or $\emptyset \neq L(\alpha_{i_1}), L(\alpha_{i_2}), L(\beta_{j_1}), L(\beta_{j_2}) \subseteq L(\mathcal{D}_r)$ or $(\emptyset \neq L(\alpha_{i_1}), L(\beta_{j_1}) \subseteq L(\mathcal{D}_l)$ and $\emptyset \neq L(\alpha_{i_2}), L(\beta_{j_2}) \subseteq L(\mathcal{D}_r))$ or $(\emptyset \neq L(\alpha_{i_1}), L(\beta_{j_1}) \subseteq L(\mathcal{D}_r)$ and $\emptyset \neq L(\alpha_{i_2}), L(\beta_{j_2}) \subseteq L(\mathcal{D}_l))$.

If no pair of leaves in $L - L(\mathcal{D})$ is a candidate pair, then all of them fulfill one of the properties above. But this means there is a completion. ∎

For two gene trees G, H, and a split $\mathcal{D}$ we can compute a candidate pair or a completion in time in $O(n^2)$ time naively.

Lemma 2 *Suppose we are given a leafset L. Let $G_1, G_2, \ldots, G_k$ gene trees, $L(G_i) \subseteq L$ $(i = 1, \ldots, k)$. Let $\mathcal{D}$ be an incomplete split of L. Then*

$G_1, \ldots G_k$ have a completion if and only if each pair of $\{G_1, \ldots, G_k\}$ has a completion.

These two lemmata lead us to the following theorem.

Theorem 1 *Given leafset L and gene trees $G_1, \ldots, G_k$, where $L(G_i) \subseteq L$ $(i = 1, \ldots, k)$. Let $\mathcal{D}$ be an incomplete split of L with $\mathcal{D}_l$, $\mathcal{D}_r$ are leafsets, $\mathcal{D}_l, \mathcal{D}_r \neq \emptyset$. Then either there is a completion of $G_1, \ldots, G_k$ or there is a candidate pair (a, b), $a, b \in L - L(\mathcal{D})$.*

For k gene trees and a given split $\mathcal{D} = (\mathcal{D}_l, \mathcal{D}_r)$, $\mathcal{D}_l, \mathcal{D}_r \neq \emptyset$, we can compute a candidate pair or a completion in time $O((n \cdot k)^2)$.

5 A Fixed-Parameter Tractable Algorithm

The properties described in the section above invite us to follow the idea of building a bounded search tree for the following parameterized version of the Gene-Duplication Problem. (For readers not familiar with the technique of bounded search trees we refer to [DFS99a, DF98].)

Gene-Duplication Problem (Parameterized Version)
Input: Gene trees $G_1, \ldots, G_k$ over leafset L, positive integer $\mathcal{C}$.
Parameter: $\mathcal{C}$
Question: Does there exist a species tree S with $cost(G_1, \ldots, G_k, S) \leq \mathcal{C}$?

The main idea is described as follows. Let $|L| = n$. We first build all possible complete splits $\mathcal{D} = (\mathcal{D}_l | \mathcal{D}_r)$ of L costing no more than $\mathcal{C}$ gene duplications and with leafsets $\mathcal{D}_l, \mathcal{D}_r$. I.e., we keep all the complete splits $\mathcal{D}$ of L causing not more than $\mathcal{C}$ gene duplications at the root of any possible species tree resulting from $\mathcal{D}$ (cf. Observation 1). Then, recursively, we refine $\mathcal{D}_l$ and $\mathcal{D}_r$ such that $\mathcal{D}_l$ and $\mathcal{D}_r$ are complete splits of $L(\mathcal{D}_l)$ and $L(\mathcal{D}_r)$ and $\mathcal{D}' = (\mathcal{D}_l | \mathcal{D}_r)$ does not cost more than $\mathcal{C}$ gene duplications. This will be continued until either we know there is no solution of cost at most $\mathcal{C}$ or there is a split $\mathcal{D}$ of L defining a species tree for L. Each node of the search tree consists of a (complete or incomplete) split of L, a set of input forests, and $\mathcal{C}$. The search tree is thus organized as a tree of height at most $\mathcal{C}$.

Let $cost'(G, \mathcal{D}, M)$ and $Dups'(G, \mathcal{D}, M)$ denote the variants of *cost* and *Dups* when there is a set $M \neq \emptyset$ attached to the split $\mathcal{D}$. Here we assume that at least one element of M belongs to $\mathcal{D}_r$ after completing the split.

Then $cost'(G, \mathcal{D}, M) = |Dups'(G, \mathcal{D}, M)|$ where

$$Dups'(G, \mathcal{D}, M) = \begin{cases} Dups(G, \mathcal{D}) & \text{if } M = \emptyset \text{ or } \mathcal{D}_r \cap M \neq \emptyset \\ \bigcap_{e \in M} Dups(G, (\mathcal{D}_l | \mathcal{D}_r \cup \{e\})) & \text{otherwise} \end{cases} .$$

Note that this variant of the cost function is also computable in polynomial time, namely in time $O(n^2)$.

Computing a completion or a candidate pair for a split $\mathcal{D}$ if $M \neq \emptyset$ is possible in time $O(n^3 \cdot k^2)$.

The algorithm below computes the kids of a node in the search tree. The search tree is elaborated recursively using this algorithm to explore all possibilities of cost at most $\mathcal{C}$. The algorithm prunes the search tree if the cost exeeds $\mathcal{C}$. Otherwise it continues branching until a complete split of L is computed. We recurse on $\mathcal{D}_l$ and $\mathcal{D}_r$ for each of the completed splits $\mathcal{D} = (\mathcal{D}_l | \mathcal{D}_r)$.

Step 1 Create the root $\mathcal{D} = (A| \ )$ of the search tree:
Pick any leaf $A \in \bigcap_{i=1}^{k} L(G_i)$. For each $G \in \{G_1, \ldots, G_k\}$ let G_l be the subtree of the root of G with $A \in L(G_l)$. If there is a completion of $\mathcal{D}$ for $G_1, \ldots, G_k$ then all gene trees agree in their leafsets of the left and right subtrees. In this case $\mathcal{D} = (L(G_l)|L(G_r))$ for any G. Stop with answer "No". Otherwise attach a set $M = \emptyset$ to $\mathcal{D}$, let $G = G_1$, compute a candidate pair (a, b), and branch in the following way: $\mathcal{D}_1 = (\mathcal{D}_l \cup \{a, b\} | \mathcal{D}_r)$ and $M = L(G_l)$, $\mathcal{D}_2 = (\mathcal{D}_l \cup \{a, b\} | \mathcal{D}_r)$ and $M = L(G_r)$, $\mathcal{D}_3 = (\mathcal{D}_l \cup \{a\} | \mathcal{D}_r \cup \{b\})$, $\mathcal{D}_4 = (\mathcal{D}_l \cup \{b\} | \mathcal{D}_r \cup \{a\})$, $\mathcal{D}_5 = (\mathcal{D}_l | \mathcal{D}_r \cup \{a, b\})$. For each $\mathcal{D}_i$, $i = 1 \ldots 5$ do: If $\mathcal{D}_{i_r} \neq \emptyset$ goto Step 4. Otherwise goto Step 5.

Step 2 Compute a candidate pair (a, b) for $\mathcal{D}$, M and $G_1, \ldots, G_k$. The candidate pair induces branching into the following subsplits of $\mathcal{D}$. $\mathcal{D}_1 = (\mathcal{D}_l \cup \{a, b\} | \mathcal{D}_r)$, $\mathcal{D}_2 = (\mathcal{D}_l \cup \{a\} | \mathcal{D}_r \cup \{b\})$, $\mathcal{D}_3 = (\mathcal{D}_l \cup \{b\} | \mathcal{D}_r \cup \{a\})$, $\mathcal{D}_4 = (\mathcal{D}_l | \mathcal{D}_r \cup \{a, b\})$. For each $\mathcal{D}_i$ do: if $M \neq \emptyset$ then apply Step 5 for $\mathcal{D} = \mathcal{D}_1$, else apply Step 4. Apply Step 4 for $\mathcal{D} = \mathcal{D}_2, \mathcal{D}_3, \mathcal{D}_4$.

Step 3 Compute a candidate pair (a, b) for $\mathcal{D}$, M and $G_1, \ldots, G_k$. Pick the tree T of the forest of G_1, with $M \subseteq L(T)$. Let $N_1 = L(T_l)$, $N_2 = L(T_r)$. The candidate pair induces branching into the following subsplits of $\mathcal{D}$. $\mathcal{D}_1 = (\mathcal{D}_l \cup \{a, b\} | \mathcal{D}_r)$ and $M := M \cap N_1$, $\mathcal{D}_2 = (\mathcal{D}_l \cup \{a, b\} | \mathcal{D}_r)$ and $M := M \cap N_2$, $\mathcal{D}_3 = (\mathcal{D}_l \cup \{a\} | \mathcal{D}_r \cup \{b\})$, $\mathcal{D}_4 = (\mathcal{D}_l \cup \{b\} | \mathcal{D}_r \cup \{a\})$, $\mathcal{D}_5 = (\mathcal{D}_l | \mathcal{D}_r \cup \{a, b\})$. If $\mathcal{D}_r = \emptyset$ then apply Step 5 for $\mathcal{D} = \mathcal{D}_1$ and $\mathcal{D} = \mathcal{D}_2$. Apply Step 4 for $\mathcal{D} = \mathcal{D}_2, \mathcal{D}_3, \mathcal{D}_4$.

Step 4 $\mathcal{C} := \mathcal{C} - cost(G_1, \ldots, G_k, \mathcal{D})$. If $\mathcal{C} < 0$ then stop with answer "No". Delete all vertices in $Dups(G_1, \ldots, G_k, \mathcal{D})$ from $G_1, \ldots, G_k$. Compute a completion in case there exist any. If so, then stop. Otherwise goto Step 2.

Step 5 $\mathcal{C} := \mathcal{C} - cost'(G_1, \ldots, G_k, \mathcal{D}, M)$. If $\mathcal{C} < 0$ then stop with answer "No". Delete all vertices in $Dups'(G_1,\ldots,G_k,\mathcal{D},M)$ from $G_1, \ldots, G_k$. Compute a completion if there exist any. If so, then stop. Goto Step 3.

Since in every branch we decrease $\mathcal{C}$ by at least 1, the height of the search tree is bounded by $\mathcal{C}$.

Theorem 2 *The overall running time of the algorithm given above is* $O(4^{\mathcal{C}} \cdot n^3 \cdot k^2)$.

Proof: A straightforward recurrence. ∎

Although the Gene-Duplication Problem is shown to be fixed-parameter tractable parameterized by $\mathcal{C}$ it remains open whether the Duplication-and-Loss Problem is also fixed-parameter tractable. We suspect the problem to be in *FPT* when parameterized by both the number of duplication and loss events and the number of gene trees. We conjecture the Duplication-and-Loss Problem to be $W[1]$-hard when parameterized by the number of duplications and losses only.

References

[BE90] S. Benner and A. Ellington. Evolution and Structural Theory. The frontier between chemistry and biochemistry. *Bioorg. Chem. Frontiers* 1 (1990), 1–70.

[CCDF97] L. Cai, J. Chen, R. Downey, M. Fellows. "The Parameterized Complexity of Short Computation and Factorization," *Arch. for Math. Logic*, 36(1997), 321–337.

[DF95a] R. G. Downey and M. R. Fellows. "Fixed Parameter Tractability and Completeness I: Basic Theory," *SIAM Journal of Computing* 24 (1995), 873-921.

[DF95b] R. G. Downey and M. R. Fellows. "Fixed Parameter Tractability and Completeness II: Completeness for $W[1]$," *Theoretical Computer Science A* 141 (1995), 109-131.

[DF95c] R. G. Downey and M. R. Fellows. "Parametrized Computational Feasibility," in: *Feasible Mathematics II, P. Clote and J. Remmel (eds.)* Birkhauser, Boston (1995) 219-244.

[DF98] R. G. Downey and M. R. Fellows. *Parameterized Complexity*, Springer, 1998.

[DFS99a] R. Downey, M. Fellows, and U. Stege. "Parameterized Complexity: A Framework for Systematically Confronting Parameterized Complexity," in *The Future of Discr. Mathem.: Proc. of the 1st DIMATIA Symp., Czech Republic, June 1997*, Roberts, Kratochvil, Nešetřil (eds.), AMS-DIMACS Proc. Ser. (1999), to appear.

[DFS99b] R. G. Downey, M. R. Fellows, and U. Stege. *Computational Tractability: The View From Mars*, to appear in the Bulletin of the EATCS.

[F88] J. Felsenstein. Phylogenies from Molecular Sequences: Inference and Reliability. *Annu. Rev. Genet.*(1988), 22, 521–65.

[FHKS98] M. Fellows, M. Hallett, C. Korostensky, and U. Stege. "Analogs & Duals of the MAST Problem for Sequences & Trees," ESA 98, LNCS 1461, pp. 103–114.

[FHKS99] M. Fellows, M. Hallett, C. Korostensky, and U. Stege. "Analogs & Duals of the MAST Problem for Sequences & Trees," Journal version, manuscript (1999).

[FHS98] M. Fellows, M. Hallett, and U. Stege. "On the Multiple Gene Duplication Problem", *Algorithms and Computation, 9th International Sympsium, ISAAC'98*, LNCS 1533 (December 1998), pp. 347–356.

[FM67] W. Fitch, E. Margoliash. "Construction of Phylogenetic Tree," *Sci.* 155 (1967).

[GCMRM79] M. Goodman, J. Czelusniak, G. Moore, A. Romero-Herrera, G. Matsuda. "Fitting the Gene Lineage into its Species Lineage: A parsimony strategy illustrated by cladograms constructed from globin sequences," *Syst.Zool.*(1979), 28.

[GMS96] R. Guigó, I. Muchnik, and T. F. Smith. "Reconstruction of Ancient Molecular Phylogeny," *Molec. Phylogenet. and Evol.* (1996),6:2, 189–213.

[HJWZ95] J. Hein, T. Jiang, L. Wang, and K. Zhang. "On the Complexity of Comparing Evolutionary Trees", *Proceedings of the 6th Annual Symposium on Combinatorial Pattern Matching*, LNCS 937 (1995), pp. 177–190.

[HJWZ96] J. Hein, T. Jiang, L. Wang, and K. Zhang. "On the Complexity of Comparing Evolutionary Trees", *DAMATH: Discrete Applied Mathematics and Combinatorial Operations Research and Computer Science 71* (1996).

[MLZ98] B. Ma, M. Li, and L. Zhang. "On Reconstructing Species Trees from Gene Trees in Term of Duplications and Losses," *Recomb 98.*

[NA86] J. E. Neigel and J. C. Avise. "Phylogenetic Relationship of mitochondrial DNA under various demographic models of speciation," *Evol. Proc. and Th.*(1986), Ac. Press NY, 515–534.

[NR98] R. Niedermeier, P. Rossmanith. "Upper Bounds for Vertex Cover Further Improved," STACS 99, to appear.

[P94] R. D. M. Page. "Maps between trees and cladistic analysis of historical associations among genes, organisms, and areas," *Syst. Biol. 43* (1994), 58–77.

[PC97] R. Page, M. Charleston. "From Gene to organismal phylogeny: reconciled trees and the gene tree/species tree problem," *Molec. Phyl. and Evol. 7* (1997), 231–240.

[S91] D. L. Swofford. "When are phylogeny estimates from molecula and morphological data incongruent?" in *Phylogenetic analysis of DNA sequences*, Oxford University Press (1991), pp. 295–333

[Z97] L. Zhang. "On a Mirkin-Muchnik-Smith Conjecture for Comparing Molecular Phylogenies," *Journal of Comp. Biol.* (1997) 4:2, 177–187.