A Note on the Fixed Parameter Tractability of the Gene-Duplication Problem

Mukul S. Bansal and Ron Shamir

***Abstract*— The NP-hard gene-duplication problem takes as input a collection of gene trees and seeks a species tree that requires the fewest number of gene duplications to reconcile the input gene trees. An oft-cited, decade-old result by Stege states that the gene-duplication problem is fixed parameter tractable when parameterized by the number of gene duplications necessary for the reconciliation. Here we uncover an error in this fixed parameter algorithm and show that this error cannot be corrected without sacrificing the fixed parameter tractability of the algorithm. Furthermore, we show a link between the gene-duplication problem and the minimum rooted triplets inconsistency problem which implies that the gene-duplication problem is (i) W[2]-hard when parameterized by the number of gene duplications necessary for the reconciliation and (ii) hard to approximate to better than a logarithmic factor.**

I. Introduction

Accurately reconstructing the phylogenetic tree depicting the evolutionary history of a given set of species is a fundamental problem in computational biology. Typically, to build a phylogenetic tree for a set of species, one constructs a phylogenetic tree from genes taken from those species. Such trees are called *gene trees*. The implicit assumption is that the evolution of the chosen genes mimics the evolution of the species themselves. However, due to complex evolutionary processes such as gene duplication and loss, recombination, and horizontal gene transfer, trees constructed on genes do not always accurately represent the evolutionary history of the corresponding species.

The gene duplication model, introduced by Goodman et al. [1], provides a framework for inferring species phylogenies (also called *species trees*) from a collection of gene trees that are confounded by complex histories of gene duplication events. In particular, the *gene-duplication* problem seeks a species tree that can explain the incongruence of (i.e. reconcile) the input gene trees using the fewest number of gene duplication events. The gene-duplication problem is NP-hard [2], and has been extensively studied; see, for example, [3], [4], [5], [6], [7], [8], [9], [10], [11], [12], [13]. An oft-cited, decade-old result on the gene-duplication problem states that this problem is fixed parameter tractable when parameterized by the number of gene duplications necessary for the reconciliation [14]. This result has been significant because it placed the gene-duplication problem in the exclusive class of NP-hard problems that can be solved exactly within reasonable time when the parameter of interest (in this case the number of gene duplication events) is small. Unfortunately, as we reveal in this manuscript, there is a fundamental error in the suggested fixed parameter algorithm. Moreover, we uncover a link between the gene-duplication problem and the minimum rooted triplets inconsistency problem which implies that the gene-duplication problem is (i) W[2]-hard when parameterized by the reconciliation cost and (ii) not approximable to better than a logarithmic factor unless P=NP.

M. S. Bansal and R. Shamir are at the Blavatnik School of Computer Science, Tel-Aviv University, Tel Aviv, Israel. {bansal, rshamir}@tau.ac.il

II. The Gene-Duplication Problem

Given a rooted tree T, we denote its node set, edge set, and leaf set by $V(T)$, $E(T)$, and $Le(T)$ respectively. The root node of T is denoted by $rt(T)$. Given a node $v \in V(T)$, we denote its parent by $pa_T(v)$, its set of children by $Ch_T(v)$, and the subtree of T rooted at v by T_v. Given a non-empty subset $L \subseteq Le(T)$ in tree T, we denote by $lca_T(L)$, the least common ancestor (lca) of all the leaves in L in tree T. Throughout this work, the term tree refers to a rooted binary tree in which all non-leaf nodes have exactly two children.

A *species tree* is a tree that depicts the evolutionary relationships of a set of species. Given a gene family for a set of species, a *gene tree* is a tree that depicts the evolutionary relationships among the sequences encoding only that gene family in the given set of species. Thus, the nodes in a gene tree represent genes. We assume that each leaf of the gene trees is labeled with the species from which that gene was sampled. We say that a gene tree G and a species tree S are *comparable* if S contains all the leaves (species) in G.

To compute the reconciliation cost of gene tree G and a given comparable species tree S, we first construct a mapping $\mathcal{M}_{G,S}\colon V(G) \rightarrow V(S)$ that maps each node $g \in V(G)$ to the node $lca_S(Le(G_g))$ in S. A node $g \in V(G) \setminus Le(G)$ is a *(gene) duplication* if $\mathcal{M}_{G,S}(g) \in \mathcal{M}_{G,S}(Ch(g))$ and we define $Dup(G,S) = \{g \in V(G) \setminus Le(G)\colon\ g \text{ is a duplication}\}$. The *reconciliation cost* of G and S, denoted by $\Delta(G,S)$, is defined to be $|Dup(G,S)|$. Similarly, given a collection $\mathcal{G}$ of gene trees, the reconciliation cost of $\mathcal{G}$ and S is denoted by $\Delta(\mathcal{G},S)$ and is equal to $\sum_{G\in\mathcal{G}} \Delta(G,S)$.

Given a collection $\mathcal{G}$ of gene trees, the *gene-duplication problem* is to find a comparable species tree S such that $\Delta(\mathcal{G},S)$ is minimized.

Next, we give a brief description of the fixed parameter algorithm of Stege [14] to solve the gene-duplication problem. Note that the parameter here is the reconciliation cost. Throughout this work we use the following terminology: $\mathcal{G}$ is a given set of gene trees, G is a gene tree from $\mathcal{G}$, and S is a species tree such that $Le(S) = \bigcup_{G\in\mathcal{G}} Le(G)$.

III. A Description of the Fixed Parameter Algorithm

This algorithm is based on the standard bounded-depth search tree technique. The input to the algorithm is the set of gene trees $\mathcal{G}$ and a parameter $\mathcal{C}$. The goal of the algorithm is to output a tree S for which $\Delta(\mathcal{G},S) \leq \mathcal{C}$, or to report that such a tree does not exist.

Briefly, the algorithm of [14] proceeds as follows: The algorithm starts at the root node of the species tree S. It splits the leaf set of S into all possible bi-partitions (for the left and right subtrees of the root of S), that cause less than $\mathcal{C}$ duplications in the input trees. Thus, the algorithm considers all feasible ways of building the leaf sets for the left and right subtrees of the root of the species tree. The algorithm then proceeds recursively on the left and right subtrees, for each of these possible splits (bi-partitions).

Since the algorithm proceeds recursively, we only describe how the algorithm constructs all the possible bi-partitions at the root of S. The algorithm builds the bi-partitions by adding the leaves incrementally. Suppose we have an incomplete bi-partition. Then, either it is possible to add all the leaves to this bi-partition without increasing the number of gene duplications, or (according to [14]) it is possible to find a pair of leaves such that each of the four ways of adding these two leaves to the current bi-partition increases the number of duplications by at least one. Thus, in terms of the bounded-depth search tree, the nodes represent partially completed bi-partitions. Each of the four ways of adding these leaves to the current bi-partition become nodes on the next level of the bounded-depth search tree. Since the number of gene duplications increases by at least one in each successive level, the depth of the search tree is bounded by $\mathcal{C}$. We refer the reader to [14] for further details on this algorithm.

IV. A Fundamental Error in the Algorithm

As seen in the previous section, the algorithm depends critically on its ability to either add all the missing leaves to the current incomplete split efficiently without increasing the number of gene duplications, called a *completion* in [14], or to find a pair of leaves such that each of the four ways of adding these two leaves to the current bi-partition increases the number of duplications by at least one. Such a pair of leaves is called a *candidate pair* in [14], and this idea is formalized therein as Theorem 1.[1] We restate this theorem:

Theorem 4.1 ([14]): Given leafset L and gene trees $G_1, \ldots, G_k$, where $Le(G_i) \subseteq L$ $(i = 1, \ldots, k)$. Let $\mathcal{D}$ be an incomplete split of L with leafsets $\mathcal{D}_l$ and $\mathcal{D}_r$, $\mathcal{D}_l, \mathcal{D}_r \neq \emptyset$. Then either there is a completion of $G_1, G_2, \ldots, G_k$ or there is a candidate pair (a, b), $a, b \in L - L(\mathcal{D})$.

Here $L(\mathcal{D})$ denotes the leaves in $\mathcal{D}$. As we illustrate with a simple example, this theorem is incorrect. But first, we need some notation. A *rooted triplet* is a (binary) tree with exactly three leaves. We denote by $ab|c$ the unique rooted triplet on leaf set a, b, c for which the lca of a and b is a proper descendant of the lca of a and c. In our example, all the input gene trees are rooted triplets. In particular, let $\{ab|c, bc|d, cd|e, de|a\}$ be the set of gene trees, and let the incomplete split $\mathcal{D}$ be such that $\mathcal{D}_l = a$ and $\mathcal{D}_r = e$. Observe that this incomplete split does not cause any necessary gene duplications in any of the gene trees, simply because all the individual triplets are consistent with the split. Also observe that it is not possible to complete this split without incurring a cost of at least one gene duplication. Yet, a simple case analysis shows that there does not exist any candidate pair.

Consider a generalization of our simple example so that the set of input gene trees is $\{a_1a_2|a_3, a_2a_3|a_4, \ldots, a_{k-1}a_k|a_1\}$, where $k \geq 5$, and the incomplete split $\mathcal{D}$ is such that $\mathcal{D}_l = a_1$ and $\mathcal{D}_r = a_k$. This example illustrates that it is, in general, not possible to extend the notion of a candidate pair to any constant sized *candidate subset*; that is, a subset of leaves for which each of the different ways of adding all these leaves to the current bi-partition increases the number of duplications by at least one. Thus, the approach of either finding a completion or a candidate pair (or any constant sized candidate subset) seems inherently flawed.

A second fundamental error. There appears to be another mistake, independent of the one pointed out above, in the fixed parameter algorithm. Consider Step 1 of this algorithm, as given in [14]. This step calls for the computation of a candidate pair when the initial incomplete split $\mathcal{D}$ is such that $\mathcal{D}_l = A$, for some leaf A, and $\mathcal{D}_r = \emptyset$. Since $\mathcal{D}_r = \emptyset$, there cannot be any candidate pair simply because any given pair of leaves can be added to $\mathcal{D}_l$ without causing any gene-duplications. A possible fix for this mistake would be to start the search tree from all the $|Le(S)| - 1$ possible incomplete splits for which $\mathcal{D}_l = A$ and $\mathcal{D}_r \in Le(S) \setminus \{A\}$. However, due to the recursive nature of the algorithm, this fix renders the worst-case running time of the algorithm exponential in $|Le(S)|$.

V. A link with Minimum Rooted Triplets Inconsistency, W[2]-hardness, and inapproximability

Consider a rooted triplet $X = ab|c$ and a tree T such that $a, b, c \in Le(T)$. We say that X is *consistent* with T if the lca of a and b is a proper descendant of the lca of a and c in T. Otherwise, X is *inconsistent* with T.

Given a collection C of rooted triplets with leaf set L, the *minimum rooted triplets inconsistency (MTI)* problem is to find a tree T, where $Le(T) = L$, that minimizes the number of rooted triplets from C that are inconsistent with T. We denote the number of triplets from C that are inconsistent with any given T, where $Le(T) = L$, by $\Gamma(C, T)$. The MTI problem is NP-hard [16], and along with its dual, the *maximum rooted triplets consistency* problem, is well studied [17], [18], [19]. The following lemma uncovers a strong link between the MTI problem and the gene-duplication Problem.

Lemma 5.1: Given a collection C of rooted triplets with leaf set L and a tree T such that $Le(T) = L$, we must have $\Gamma(C, T) = \Delta(C, T)$.

Proof: Consider any triplet $X \in C$. X is consistent with T if and only if $|Dup(X, T)| = 0$. Moreover, since $|Dup(X, T)| \in \{0, 1\}$, any inconsistent triplet contributes exactly one to $\Delta(C, T)$. Thus, we must have $\Gamma(C, T) = \Delta(C, T)$. ■

It was recently shown [19] that the MTI problem (even when restricted to binary trees) is (i) W[2]-hard when parameterized by the number of inconsistent triplets, and (ii) inapproximable to within a factor of $c \cdot \ln n$ for some constant $c > 0$ (unless P = NP), where n is the size of the leaf set L. Thus, in light of Lemma 5.1, we have the following theorem.

Theorem 5.1: The gene-duplication problem is W[2]-hard when parameterized by the reconciliation cost. The gene-duplication problem cannot be approximated to within a factor of $c \cdot \ln n$ for some constant $c > 0$, where n is the size of the resulting species tree, unless P=NP.

Proof: W[2]-hardness: We give a parameterized reduction from the MTI problem to the gene-duplication problem. Let (C, k) be an instance of the parameterized version of the MTI problem, which asks if there exists a tree T such that $\Gamma(C, T) \leq k$.

[1] We note that this theorem appears without proof in [14]. The full version of [14] appears in Stege's PhD Thesis [15], but the proof of this theorem is also missing therein (see Theorem 7.8 in [15]).

Construct an instance of the gene-duplication problem by setting the set of input gene trees $\mathcal{G}$ to be C, and the parameter to be k. Thus, we ask if there exists a tree S comparable with $\mathcal{G}$ such that $\Delta(\mathcal{G}, S) \leq k$. Note that both T and S are trees on the same leaf set. Now, by Lemma 5.1, if $\Gamma(C, T) \leq k$ then $\Delta(\mathcal{G}, T) \leq k$, and if $\Delta(\mathcal{G}, S) \leq k$ then $\Gamma(C, S) \leq k$. Thus, the instance (C, k) of the MTI problem has a *yes* answer if and only if the instance $(\mathcal{G}, k)$ of the gene-duplication problem has a *yes* answer. Since our reduction is computable in polynomial time and preserves the parameter, the proof is complete.

Inapproximability: By a completely analogous argument it follows that if the gene-duplication problem can be approximated to better than a logarithmic factor then we can obtain a better-than-logarithmic factor solution to the MTI problem within polynomial time as well. ■

This relationship between the MTI and gene-duplication problems also provides a simple alternative proof of the NP-hardness of the gene-duplication problem. We point out that by an analogous argument, the results of Theorem 5.1 apply also to the *duplication-loss problem*, which is a variant of the gene-duplication problem; see, e.g., [2], [8], [20] for a definition of this problem. This analogous argument for the duplication-loss problem relies on the simple observation that, in the context of Lemma 5.1, if any triplet from C is consistent with T then it has a duplication-loss cost of 0, while if it is inconsistent with T then it has a duplication-loss cost of exactly 4 (i.e., 1 duplication + 3 losses).

ACKNOWLEDGEMENTS

We thank Cedric Chauve for bringing to our attention ref. [19]. MSB was supported in part by a postdoctoral fellowship from the Edmond J. Safra Bioinformatics program at Tel-Aviv university. RS was supported in part by the Israel Science Foundation (Grant 802/08).

REFERENCES

[1] M. Goodman, J. Czelusniak, G. W. Moore, A. E. Romero-Herrera, and G. Matsuda, "Fitting the gene lineage into its species lineage. a parsimony strategy illustrated by cladograms constructed from globin sequences," *Systematic Zoology*, vol. 28, pp. 132–163, 1979.

[2] B. Ma, M. Li, and L. Zhang, "From gene trees to species trees," *SIAM J. Comput.*, vol. 30, no. 3, pp. 729–752, 2000.

[3] R. D. M. Page, "Maps between trees and cladistic analysis of historical associations among genes, organisms, and areas," *Systematic Biology*, vol. 43, no. 1, pp. 58–77, 1994.

[4] R. Guigó, I. Muchnik, and T. F. Smith, "Reconstruction of ancient molecular phylogeny," *Molecular Phylogenetics and Evolution*, vol. 6, no. 2, pp. 189–213, 1996.

[5] B. Mirkin, I. Muchnik, and T. F. Smith, "A biologically consistent model for comparing molecular phylogenies," *Journal of Computational Biology*, vol. 2, no. 4, pp. 493–507, 1995.

[6] O. Eulenstein and M. Vingron, "On the equivalence of two tree mapping measures," *Discrete Applied Mathematics*, vol. 88, pp. 101–126, 1998.

[7] L. Zhang, "On a Mirkin-Muchnik-Smith conjecture for comparing molecular phylogenies," *Journal of Computational Biology*, vol. 4, no. 2, pp. 177–187, 1997.

[8] M. T. Hallett and J. Lagergren, "New algorithms for the duplication-loss model," in *RECOMB*, 2000, pp. 138–146. [Online]. Available: citeseer.nj.nec.com/article/hallett99new.html

[9] K. Chen, D. Durand, and M. Farach-Colton, "Notung: a program for dating gene duplications and optimizing gene family trees," *Journal of Computational Biology*, vol. 7, pp. 429–447, 2000.

[10] P. Bonizzoni, G. D. Vedova, and R. Dondi, "Reconciling a gene tree to a species tree under the duplication cost model," *Theor. Comput. Sci.*, vol. 347, no. 1-2, pp. 36–53, 2005.

[11] P. Górecki and J. Tiuryn, "Dls-trees: A model of evolutionary scenarios," *Theor. Comput. Sci.*, vol. 359, no. 1-3, pp. 378–399, 2006.

[12] C. Chauve, J.-P. Doyon, and N. El-Mabrouk, "Gene family evolution by duplication, speciation, and loss," *Journal of Computational Biology.*, vol. 15, no. 8, pp. 1043–1062, 2008.

[13] M. S. Bansal, O. Eulenstein, and A. Wehe, "The gene-duplication problem: Near-linear time algorithms for NNI-based local searches," *IEEE/ACM Transactions on Computational Biology and Bioinformatics*, vol. 6, no. 2, pp. 221–231, 2009.

[14] U. Stege, "Gene trees and species trees: The gene-duplication problem is fixed-parameter tractable," in *WADS*, 1999, pp. 288–293.

[15] ——, "Resolving conflicts in problems from computational biology," Ph.D. dissertation, Swiss Federal Institute Of Technology (ETH), Zürich, 1999.

[16] D. Bryant, "Hunting for trees, building trees andcomparing trees: theory and method in phylogeneticanalysis," Ph.D. dissertation, Dept. of Mathematics, University of Canterbury, 1997.

[17] B. Y. Wu, "Constructing the maximum consensus tree from rooted triples," *J. Comb. Optim.*, vol. 8, no. 1, pp. 29–39, 2004.

[18] S. Snir and S. Rao, "Using max cut to enhance rooted trees consistency," *IEEE/ACM Trans. Comput. Biology Bioinform.*, vol. 3, no. 4, pp. 323–333, 2006.

[19] J. Byrka, S. Guillemot, and J. Jansson, "New results on optimizing rooted triplets consistency," *Discrete Applied Mathematics*, (accepted).

[20] M. S. Bansal, J. G. Burleigh, and O. Eulenstein, "Efficient genome-scale phylogenetic analysis under the duplication-loss and deep coalescence cost models," *BMC Bioinformatics*, vol. 11, no. Suppl 1, p. S42, 2010.