On the Multiple Gene Duplication Problem

Michael Fellows[1], Michael Hallett[2], and Ulrike Stege[2]

[1] Dep. of Computer Science, Univ. of Victoria, Victoria, B.C. Canada V8W 3P6
mfellows@csr.uvic.ca
[2] Comp. Biochemistry Research Group, ETH Zürich, CH-8092 Zürich, Switzerland
{hallett,stege}@inf.ethz.ch

Abstract. A fundamental problem in computational biology is the determination of the correct *species tree* for a set of taxa given a set of (possibly contradictory) *gene trees*. In recent literature, the Duplication/Loss model has received considerable attention. Here one measures the similarity/dissimilarity between a set of gene trees by counting the number of *paralogous gene duplications* and subsequent *gene losses* which need to be postulated in order to explain (in an evolutionarily meaningful way) how the gene trees could have arisen with respect to the species tree. Here we count the number of *multiple gene duplication events* (duplication events in the genome of the organism involving one or more genes) without regard to gene losses. Multiple Gene Duplication asks to find the species tree S which requires the fewest number of multiple gene duplication events to be postulated in order to explain a set of gene trees $G_1, G_2, \ldots, G_k$. We also examine the related problem which assumes the species tree S is known and asks to find the explanation for $G_1, G_2, \ldots, G_k$ requiring the fewest multiple gene duplications. Via a reduction to and from a combinatorial model we call the Ball and Trap Game, we show that the general form of this problem is *NP*-hard and various parameterized versions are hard for the complexity class $W[1]$. These results immediately imply that Multiple Gene Duplication is similarily hard. We prove that several parameterized variants are in *FPT*.

1 Introduction to the Model

A fundamental problem arising in computational biology is the determination of the (correct) evolutionary topology for a set of taxa given a set of (possibly contradictory) *gene trees*. A gene tree is a complete rooted binary tree formed over a family of homologous genes for a set of taxa. For various reasons, two or more gene trees for the same set of taxa may not always agree (see [2,5] amongst others). The question then arises of how to reconstruct the correct species tree for these taxa from the given gene trees. Several models have appeared in the literature including possibly the most famous MAST [7,11,12,13]. One such cost model which has received considerable attention of late is the Gene Duplication and Loss model introduced in [8] and discussed in [9,10,14]. 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

K.-Y. Chwa and O. H. Ibarra (Eds.): ISAAC'98, LNCS 1533, pp. 347–357, 1998.

the gene trees could have arisen with respect to the species tree. We use angle brackets $<,>$ to denote multisets. All trees in this paper are rooted and leaf labeled. Let $T = (V, E, L)$ be such a rooted tree where V is the vertex set, E is the edge set, and $L \subseteq V$ is the leaf label set. For a vertex $u \in V - L$, let T_u be the subtree of T rooted by u. Let $root(T)$ denote the root of T and, for any vertex $v \in V$, let $parent_T(v)$ be the parent of v in T, and for binary trees, let $left_T(v)$ be the left kid of v and $right_T(v)$ be the right kid of v. Where $L = \{1, 2, \ldots, n\}$ we call these leaf labeled trees either a *species* or a *gene* tree. 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. In [10], a function t_{dup} is given which returns the minimum number of *duplication events* necessary to rectify a gene tree with a species tree (here the algorithm from [9] is modified to ignore losses). Our function M below maps a gene tree G into a species tree S by defining functions $loc_{G,S}$ and $event_{G,S}$. It is the case that $t_{dup} = |\{u | u \in V_G - L, event_{G,S}(u) = dup\}|$.

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

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

The following problem lies at the heart of the DUPLICATION model:

OPTIMAL SPECIES TREE (DUPLICATION MODEL):
Input: Set of gene trees $G_1, \ldots, G_k$. *Question:* Does there exist a species tree S with minimum duplication cost $\sum_{i=1}^{k} t_{dup}(G_i, S)$?

In [10], it is shown that the problem of finding the species tree S which minimizes the number of gene duplications is *NP*-complete (here the gene trees may contain leaf labels which appear more than once). When each gene tree may contain a leaf label at most once (a gene tree is formed over exactly one gene per taxa), the problem remains *NP*-complete and the parameterized (by k) version is hard for $W[1]$ (see [6]). A similar question to OPTIMAL SPECIES TREE arises if we ask for the species tree which implies the minimum number of *multiple gene duplications* for a given set of gene trees. A duplication event in the genome of an organism involves a stretch of DNA where one or more genes may reside. Previous models for rectifying gene trees with species trees considered a duplication event to effect one gene at a time. However, there is evidence that genomes of, for example, eukaryotic organisms, have been entirely duplicated one or more times or individual chromosomes have been duplicated multiple times. In either case, sets of genes were duplicated in one event creating a set of paralogous genes. Such paralogous duplications make finding the correct species topology especially difficult [5]. Consider a vertex u in a species tree S. Each gene tree G_i has some number of vertices (possibly zero) with $loc_{G,S}$ equal to u and $event_{G,S}$ equal to *dup*. Let $Dup = \{d_1, d_2, \ldots, d_c\}$ denote this set. We can partition the Dup into sets with the property that each set has at most one element from each G_i and so that these sets are maximal. One such set is termed a *multiple gene duplication* and it counts exactly one to the overall number of multiple gene duplications required to rectify the gene trees with respect to the species tree. The multiple gene duplication score for the vertex u is the total number of such partitions. By "moving" gene duplication events in G_i towards the root of S according to a set of rules, we can decrease the total number of multiple gene duplications required. Fig. 1 gives a concrete example.

Observation. *The duplication mapping function M given above (modified from [9,10]) provides an upper bound for the number of multiple gene duplications for a set of gene trees $G_1, G_2, \ldots, G_k$ and a species tree S.*

Let $G = (V_G, E_G, L)$ be a gene tree and $S = (V_S, E_S, L')$ a species tree, $L \subseteq L'$. Consider a vertex $u \in V_G$ such that $event_{G,S}(u) = dup$ and $u \neq root(G)$. Let $v = parent_G(u)$. The rules for moving duplication events towards the root of a species tree are as follows *Move 1: $event_{G,S}(v) = dup$.* We may move the duplication associated with u from $loc_{G,S}(u)$ to $loc_{G,S}(v)$ without creating any new duplications. Now $loc_{G,S}(u) = loc_{G,S}(v)$. *Move 2: $event_{G,S}(v) = spec$.* When moving the duplication associated with u from $loc_{G,S}(u)$ to $loc_{G,S}(v)$, we must change $event_{G,S}(v)$ to be dup. Now $loc_{G,S}(u) = loc_{G,S}(v)$.

Definition 1. *Given a gene tree G, a species tree S, and the functions $loc_{G,S}$ and $event_{G,S}$ mapping G into S, we say that S* receives *G, if the configuration given by $loc_{G,S}$ and $event_{G,S}$ can be reached by a series of moves starting from the initial configuration obtained by applying $M(G,S)$.*

The Multiple Gene Duplication problem can now be stated as follows:
Multiple Gene Duplication I
Input: Set of gene trees $G_1, \ldots, G_k$, integer c. *Question:* Do there exist a species tree S and functions $loc_{G_i,S}$, $event_{G_i,S}$, for $1 \leq i \leq k$, s.t. S receives $G_1, \ldots, G_k$ with at most c multiple gene duplications?

We state an easier version of Multiple Gene Duplication:
Multiple Gene Duplication Problem II (GDII)
Input: Set of gene trees $G_1, \ldots, G_k$, a species tree S, integer c. *Question:* Do there exist functions $loc_{G_i,S}$, $event_{G_i,S}$, for $1 \leq i \leq k$, s.t. S receives $G_1, \ldots G_k$ with at most c multiple gene duplications?

Via a reduction to and from a combinatorial problem called the Ball and Trap Game, we show $W[1]$-hardness and *NP*-completeness for GDII. We also show the problem is fixed-parameter tractable when various restrictions are placed on the number of gene trees and the number of the gene duplications. For an introduction to parameterized complexity we refer readers to [3,4].

2 The Ball and Trap Game

The Ball and Trap Game is played on a rooted labeled tree $T = (V, E, L)$ decorated with a set of traps D and a set of balls B. Every ball and trap has a color associated with it; this is given by the functions $c_B : B \to [1:k]$ and $c_D : D \to [1:k]$ respectively. The balls and traps are initially associated with internal vertices of T via the attaching functions $l_B : B \to V - L$ and $l_D : D \to V - L$. Each ball $b \in B$ of color $c_B(b)$ is labeled with a (possible empty) subset $R_b \subseteq D$ of

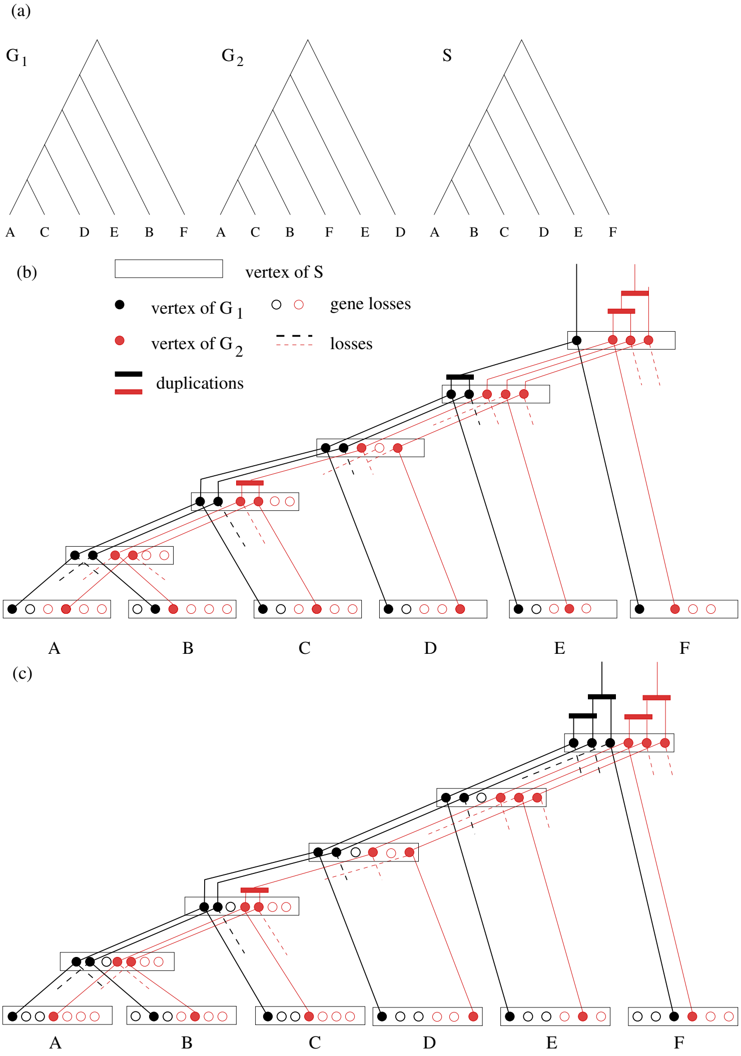

Fig. 1. (a) Two gene trees (G_1, G_2) and a proposed species tree S. (b) The species tree S has G_1 and G_2 embedded inside of it according to the standard Duplication and Loss mapping function M. Note that G_1 causes one duplication (vertex $ABCDE$) whilst G_2 causes 3 duplications (vertices ABC, $ABCDEF$, and again $ABCDEF$). The score according to the Multiple Gene Duplication model is 4. (c) After moving the gene duplication of G_1 located at vertex $ABCDE$ to the root of the species tree S, two additional gene duplications for G_1 need to be postulated. Nevertheless, the score according to the Multiple Gene Duplication model is now 3. Note that it is not beneficial to move the gene duplication from G_2 located at ABC towards the root. The two gene duplications located initially at the root from G_2 cannot move upwards.

traps. For each ball b every trap $d \in R_b$ is of the color $c_D(d) = c_B(b)$. A ball with a given set of traps may occur many times in the tree (i.e. for $b, b' \in B$ $R_b = R'_b$ and $c_B(b) = c_B(b')$ but $l_B(b) \neq l_B(b')$ is possible). Also, a vertex in the tree can be decorated with many different balls and traps.

A game consists of some number of moves, after which the score is calculated. The rules of the game are as follows: 1. Balls and traps are initially placed at internal vertices of T according to l_B and l_D. 2. Balls may not move down the tree. They may either stay in the same place or move upwards following the topology of T. In each turn, a ball b on a vertex v can be moved to the $parent(v)$. 3. We say that a trap $d \in D$ is *dangerous* for a ball $b \in B$ if $d \in R_b$. A ball b *sets off* a trap if the ball is placed at the vertex of a trap dangerous for b. 4. When a trap d is *set off* by a ball b, it is removed from the game and replaced by two new balls $b_{new}, b_{new'}$ s.t. (a) $c_B(b_{new}) = c_B(b_{new'}) = c_B(b)$, (b) $l_B(b_{new}) = l_B(b_{new'}) = l_D(d)$, (c) $R_{b_{new}} = R_{b_{new'}} = R_B - d$ and $R_b = R_b - d$, and (d) $R_{b'} = R_{b'} - d$, for all $b' \in B$. The goal of the game is to minimize the score of tree T which is defined by $s_{max}(v) = \sum_{v \in V(T)} \max\{s(1, v), ..., s(k, v)\}$ where $s(c, v)$ denotes the number of balls of color c at vertex v in T.

Ball and Trap Game (Optimization)

Input: A rooted labeled tree T, a set of *balls* B, a set of *traps* D, two coloring functions $c_B : B \to [1 : k]$ and $c_D : D \to [1 : k]$, two initial location functions $l_B : B \to V_T - L$, $l_D : D \to V_T - L$, and for each ball $b \in B$ a set $R_b \subseteq D$ where for each $d \in R_b$ $c_D(d) = c_B(b)$.

A Round: Each round of the game consists of the player moving any number of same colored balls up the tree or deciding not to move any balls (halting move).

Output: The location function $l'(B)$ generated according to the above rules which minimizes $\Sigma_{\forall v \in V(T)} s_{max}(v)$.

The input is measured as follows: n denotes the size of T, k denotes the number of colors, r denotes the number of traps, and there are at most m balls on any vertex of T in the initial configuration. The above defined game leads to the following decision variant of the problem:

Ball and Trap (BT) - Decision

Input: A rooted tree T decorated with traps D and balls B in the manner described above, and a positive integer t. *Question:* Can the Ball and Trap Game be played on T to achieve a score of at most t?

Theorem 1. Multiple Gene Duplication II *reduces to* BT (Decision).

Proof(sketch). We construct an instance $I' \in$BT from an instance of $I \in$GDII. Let T equal the species tree S, $t = c$, and the number of colors k' of I' be the number of gene trees k from I. Each color corresponds to one of the input gene trees. Apply $M(G_i, S)$ and consider the functions $loc_{G_i,S}$ and $event_{G_i,S}$, for $1 \leq i \leq k$. We create a ball b with $c_B(b) = i$ for every vertex $u \in V_G$ s.t. $event_{G_i,S}(u) = dup$. Let $l_B(u) = loc_{G_i,S}(u)$. If $loc_{G_i,S}(u) \neq root(S)$, then let $Dup = \{d | d \in V_{G_i}, d$ is an ancestor of u, $event_{G_i,S}(d) = spec\}$. For each $d \in Dup$ we create a trap d' and let $l_D(d') = loc_{G_i,S}(d)$ in T and $c_D(d') = i$. Place d in R_b. The proof that $I' \in$BT$_{``yes''}$ if and only if $I \in$GDII$_{``yes''}$ is straightforward. One need only verify that the legal moves for a ball in the Ball and

Trap Game correspond to the legal moves for a duplication event for Multiple Gene Duplication. The only slightly tricky situation arises when there is a series $u_1, u_2, \ldots, u_q \in V_{G_i}$ s.t. $event_{G_i,S}(u_p) = dup$, $loc_{G_i,S} = x$ and u_p is a director descendant of u_{p+1} in G_i, for all $1 \leq p < q$. In Multiple Gene Duplication, one must first move duplication event p upwards before moving duplication events $1, \ldots, p-1$. When the ball corresponding to duplication event u_p is moved upwards to the same level as the ball corresponding to duplication event $u_{p'}$, $p < p'$, the traps dangerous for these two balls are equivalent.

3 Easy and Hard Parameterizations of the Ball and Trap Game

We consider the following parameterizations of Ball and Trap.
Ball and Trap:
Input: A rooted tree T decorated with traps D and balls B in the manner described above and integer t.
Parameters: $k = 2$, for each $b \in B$ let $|R_b| \leq 2$, number of traps r. (Version I)
Parameters: k, r, m, t. (Version II)
Parameters: k, r. (Version III)
Question: Can the Ball and Trap Game be played on T to achieve a score of at most t?

3.1 Easy Parameterizations

We can use finite-state dynamic programming on trees (equivalently, finite-state recognition of trees vertex labeled from a finite set of labels) in order to prove the following fixed-parameter tractability result.

Theorem 2. *For every fixed set of parameter values (k, r, m, t), the problem* Ball and Trap II *can be solved in time linear in the size of the tree.*

Proof. Using the methods of [1], we can represent an input tree T as a labeled binary tree (even though T may not be binary), where the labels (which we will refer to as colors) indicate both the structure of T and the adornments of the vertices of T with balls and traps. The colored binary tree that represents an input tree T is called a *parse tree* for T. In this representation of T we can assume that the total number of balls of any given color on T is bounded by t, since otherwise T would be a “No” instance. We argue that there is a finite state tree automaton that recognizes precisely those labeled binary trees that represent “Yes” instances of the problem. Our argument is based on the “method of test sets” of [1,4]. Since there are at most $k2^r$ types of balls, and since each vertex may have m balls, $2^r(k2^r)^m$ colors suffice. The input trees are rooted, and we may assume that the parse tree for a given input tree T is rooted compatibly (i.e., at the same vertex). We use the following parsing operators: (1) the unary operator $\otimes_c$, for each color c, that has the effect of adding a single edge from the root to a new root colored c, and (2) the binary $\oplus$ operator (defined only for trees having the same color root) that takes as arguments two trees T_1 and T_2

and returns the tree $T = T_1 \oplus T_2$ obtained by identifying the root vertices of T_1 and T_2. We describe a set $\mathcal{T}$ of *tests*, each of which is a predicate $p(T)$ about a decorated tree T. We say that T *passes* the test if $p(T) = \texttt{true}$. Decorated trees T and T' are *equivalent*, denoted $T \sim T'$, if they: (1) have roots of the same color (i.e., that are decorated with traps and balls in the same way), and (2) pass the same set of tests in $\mathcal{T}$, i.e., if $\{p \in \mathcal{T} : p(T) = \texttt{true}\} = \{p \in \mathcal{T} : p(T') = \texttt{true}\}$ A test in $\mathcal{T}$ is specified by the following items: (1) A positive integer $t' \leq t$. (2) A length k vector $S = (s_1, ..., s_k)$ of non-negative integers, where each s_i is at most t. (3) The statement: "It is possible to play the Ball and Trap Game on T in such a way that at the end of play: • The total score on the internal vertices of T is at most t'. • The set of balls on the root of T consists of s_1 balls of color 1, s_2 balls of color $2, \ldots,$ and s_k balls of color k? To conclude the theorem it is enough to establish the following three claims. *Claim 1.* The equivalence relation $\sim$ has a finite number of equivalence classes. *Claim 2.* If $T_1 \sim T_2$ and T_1 is a *Yes*-instance for the problem, then T_2 is a *Yes*-instance also. *Claim 3.* The equivalence relation $\sim$ is a congruence with respect to the two parsing operators $\otimes_c$ and $\oplus$ for trees.

Theorem 3. Ball and Trap III *can be solved in time* n^c *where* $c = O((k2^r)^m)$.

Proof(sketch). We use leaf-to-root dynamic programming. At each vertex u of the tree T we calculate a table of pairs (c, S) where S is a set of balls to be passed upwards from u, and c is the minimum total score that can be achieved in the subtree T_u rooted at u assuming that the balls of S are passed upwards. It is easy to calculate this information for a vertex u from the information for the children of u. The best score that can be achieved for T is the value c at the root for $S = \emptyset$.

3.2 *W*-hardness of the Ball and Trap Game

We prove the $W[1]$-hardness of Ball and Trap I for parameter r by means of a polynomial-time parameterized reduction from Clique (Clique is proven complete for $W[1]$ in [3]). As an intermediate step we prove that the following parameterized problem is hard for $W[1]$.

k, r-Small Union

Input: A family $\mathcal{F}$ of subsets of $\{1, ..., n\}$, and positive integers r and k. *Parameter:* (r, k) *Question:* Is there a subfamily $\mathcal{F}' \subseteq \mathcal{F}$ with $|\mathcal{F}'| \geq r$ s.t. the union of the sets in $\mathcal{F}'$ has cardinality at most k?

Lemma 1. Small Union *is* NP-*complete and hard for* $W[1]$.

Theorem 4. Ball and Trap *is* NP-*complete, and* Version I *is hard for* $W[1]$.

Proof. Ball and Trap is well-defined for non-binary trees, and so we describe how Small Union can be reduced to Ball and Trap I. Let $(\mathcal{F}, r, k)$ be an instance of Small Union. We can assume, by the reduction described above, that $\mathcal{F}$ consists of 2-element sets. In order to describe the reduction we must describe a tree T with decorations, and the target value t for the game on T. The tree T is just a star of degree $n = |\mathcal{F}|$ (with the root being the central vertex). Each leaf of T is associated with an element of $\mathcal{F}$. The two colors are *red* and *blue*. There are n red traps $\tau_1, ..., \tau_n$. Each leaf is decorated with a single red ball labeled with the set of traps $\{\tau_u, \tau_v\}$ for the associated ("edge") set $\{u, v\}$. The root is decorated with $k + r$ blue balls, each labeled with the empty set of traps. The root is also decorated with all the n red traps. We set $t = (n - r) + (k + r) = n + k$. The basic idea for the correctness argument can be described as follows. Initially, the score is $n + k + r$. The only possible move is to move a red ball from a leaf to the root. If r balls can be moved up to the root from the leaves, with the r balls chosen so that the union of their trap label sets has cardinality k, then the result is a total of $k + r$ red balls at the root (where there are $k+r$ blue balls, so the cost of the root in the final score remains $k + r$). Thus the score at the end is t. Conversely, if a score of t is achieved by a game g, then necessarily at least r red balls must be moved up from the leaves. Let g' denote the truncated game consisting of the first r moves. There are two possibilities: 1. g' also achieves a score of at most t, and 2. the score for the game g' is greater than t. In case 1, exactly r red balls are moved to the root and consequently the score for the root vertex is at most $k + r$, which implies that the union of the trap label sets for the balls moved up has cardinality at most k. This implies that $(\mathcal{F}, r, k)$ is a "Yes" instance for the Small Union problem. In case 2, there are more than $k + r$ red balls at the root after the moves of g'. Since the number of red balls now exceeds the number of blue balls at the root, each further move of g is of no advantage in decreasing the total score, contradicting that g is a game that achieves a score of at most t.

Theorem 5. Ball and Trap I *remains* $W[1]$*-hard restricted to binary trees, the maximum number of traps per vertex is one per color, and balls are placed on neither leaves nor parents of leaves.*

The proof follows from straightforward modifications to the construction in Theorem 4. We replace the star T by a binary tree and provide a construction where no internal vertices (with the exception of the root) receive balls or traps.

We introduce the following version of Ball and Trap as it is used to establish the hardness of the Multiple Gene Duplication problem.

Ball and Trap IV (BTIV):
Input: A rooted binary tree T decorated with traps D and balls B in the typical manner, and a positive integer t. *Parameters:* $k = 2$ and the number of traps r. *Conditions:* (1) $|R_b| \leq 2$, f.a. $b \in B$. (2) F.a. $v \in V_T$ and each color c, T_v has at most $|T_v| - 2$ c-colored balls. (3) $R_b = R_b'$ if $l_B(b) = l_B(b')$ and $c_B(b) = c_B(b')$. (4) $R_b \subseteq R_{b'}$ if $l_B(b)$ is an ancestor of $l_B(b')$ in T. (5) No *useless* traps are allowed (a trap d is useless if no ball b in the subtree where the trap is located has $d \in R_b$). (6) If $b, b' \in B$ where the vertex $l_B(b)$ is a descendant of the vertex $l_B(b')$, then all traps $d \in R_b - R_{b'}$ are placed at vertices on the path from b to b' (inclusive). *Question:* Can the Ball and Trap Game be played on T to achieve a score of at most t?

Because none of the conditions in specified in Ball and Trap IV are violated in the reduction, we receive the following corollary.

Corollary 1. Ball and Trap IV *is* $W[1]$*-hard.*

4 Hardness of the Multiple Gene Duplication Problem

Corollary 1 and the following theorem are sufficient to prove $W[1]$-hardness and *NP*-completeness of the Multiple Gene Duplication II problem.

Theorem 6. Ball and TrapIV *reduces to* Multiple Gene Duplication II.

Proof(sketch). We construct an instance $I' \in GDII$ from an instance $I \in BTIV$. Our reduction builds a species tree $S = (V_S, E_S, L)$ and gene trees G_1 and G_2. I is restricted to 2 colors; we associate color 1 with G_1 and color 2 with G_2. W.l.o.g. we restrict our attention to balls and traps of one color c. Let $S = T$. For each vertex $v \in V$, if $v \in L$, then let $free(v) = \{v\}$. When $v \in V - L$ but v is not decorated with a ball or trap, then let $free(v) = free(left(v)) \cup free(right(v))$. If $v \in V - L$ and is decorated by a ball but no trap, then we remove a leaf from $free(left(v))$ and a leaf from $free(right(v))$ and set these to be the children of a new vertex w. For each ball b located at v, we remove an element e from $free(left(v))$ or $free(right(v))$, where e is choosen to be a tree if a tree exists in $free(left(v)) \cup free(right(v))$ or otherwise a leaf. Let w' by the parent of w and e. Let w equal w'. Let $free(w) = free(left(v)) \cup free(right(v)) \cup \{w\}$.

If $v \in V - L$ and is decorated by a trap d, then let e_1 be an element removed from the set $free(left(v))$ and e_2 be an element removed from the set $free(right(v))$. We choose e_1 as follows: If there is a tree in $free(left(v))$ (resp. $free(right(v))$) such that $T_{left(v)}$ ($T_{right(v)}$) has a ball b and $d \in R_b$, then choose one such element. Otherwise, choose any element. It is easy to show that one of e_1 or e_2 must be a tree. Create a new vertex w and place e_1 and e_2 as the children of w. For each ball located at v, we perform the same operations done in the case when $v \in V - L$ and not decorated by a trap. When these operations are completed, we remove one tree from $free(root(T))$; call it τ. It is easy to verify that $free(root(T)) = L - L_\tau$. We complete G_c from τ by embedding the remaining leaves $free(root(T))$ in accordance with the topology of T. We build the maximal subtrees $T_v = (V_{T_v}, E_{T_v}, L_{T_v})$ of T over the elements of $free(\cdot)$. For each such tree T_v we compute the sibling w of v in S and specify p, the lca_τ of the leaves of L_{T_v} in τ. Then we subdivide edge $(p, parent_\tau(p))$ in (p, p') and $(p', parent_\tau(p))$ and add T_v as the sibling of p in τ.

The proof that $I' \in GDII_{``yes''}$ if and only if $I \in BTIV_{``yes''}$ is straightforward and omitted.

5 Conclusions

Via a combinatorial abstraction called the Ball and Trap Game, we have examined the Multiple Gene Duplication problem from both the classical and parameterized complexity frameworks and provided several FPT algorithms for parameterized versions of the problem. It would be interesting to find useful approximation algorithms (or even meaningful heuristics) for parameterized and restricted versions of this problem. Particularly, we would like a nice, meaningful way to guess a (possibly quite large) set of candidate topologies for the species tree S in Multiple Gene Duplication I. One idea is to use the parameterized complexity framework since FPT algorithms are closely related to the design of useful heuristics [4]. Thus far, our models have used unweighted gene and species trees, which means that we have ignored potentially useful *distance data* between genes from the taxa. Future models should include this information. Additionally, the model so far ignores information about the position of genes along the chromosome (it makes no sense to postulate a multiple gene duplication for a set of genes when they lie on different chromosomes in the genome of the organism unless there is evidence that all genes located on this chromosome where duplicated). It may be possible to encode this type (and other types) of restrictions into the Ball and Trap Game. Lastly, other interesting and biologically meaningful parameterizations may lead to good FPT algorithms.

References

1. K. R. Abrahamson and M. R. Fellows. Finite automata, bounded treewidth and well-quasiordering. In *Graph Structure Theory, Contemporary Mathematics vol. 147*, pp. 539–564. AMS, 1993. 352
2. S. Benner and A. Ellington. Evolution and Structural Theory. The frontier between chemistry and biochemistry. *Bioorg. Chem. Frontiers* 1 (1990), 1–70. 347
3. R. G. Downey and M. R. Fellows. *Parameterized Complexity*, Springer, 1998. 349, 353
4. 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. (1998), to appear. 349, 352, 356
5. J. Felsenstein. Phylogenies from Molecular Sequences: Inference and Reliability. *Annu. Rev. Genet.*(1988), 22, 521–65. 347, 348
6. M. Fellows, M. Hallett, C. Korostensky, and U. Stege. "Analogs & Duals of the MAST Problem for Sequences & Trees," ESA 98, to appear. 348
7. W. Fitch, E. Margoliash. "Construction of Phylogenetic Tree," *Sci.* 155 (1967). 347
8. 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. 347
9. R. Guigó, I. Muchnik, and T. F. Smith. "Reconstruction of Ancient Molecular Phylogeny," *Molec. Phylogenet. and Evol.* (1996),6:2, 189–213. 347, 348, 349
10. B. Ma, M. Li, and L. Zhang. "On Reconstructing Species Trees from Gene Trees in Term of Duplications and Losses," *Recomb 98*. 347, 348, 349
11. J. Neigel and J. Avise. "Phylogenetic Relationship of mitochondrial DNA under various demographic models of speciation," *Evol. Proc. and Th.*(1986), 515–534. 347

12. R. D. M. Page. "Maps between trees and cladistic analysis of historical associations among genes, organisms, and areas," *Syst. Biol. 43* (1994), 58–77. 347
13. R. Page, M. Charleston. "From Gene to organismal phylogeny: reconciled trees and the gne tree/species tree problem," *Molec. Phyl. and Evol. 7* (1997), 231–240. 347
14. L. Zhang. "On a Mirkin-Muchnik-Smith Conjecture for Comparing Molecular Phylogenies," *J. of Comp. Biol.* (1997) 4:2, 177–187. 347