Combinatorial Optimization Solutions for the Maximum Quartet Consistency Problem

António Morgado and Joao Marques-Silva

School of Electronics and Computer Science, University of Southampton, UK
ajrm@soton.ac.uk,jpms@ecs.soton.ac.uk

Abstract

Given a set of taxa S and a complete set of quartet topologies Q over S, the problem of determining a phylogeny that satisfies the maximum number of topologies is called the Maximum Quartet Consistency (MQC) problem. The MQC problem is an NP-problem. MQC has been solved both heuristically and exactly. Exact solutions for MQC include Constraint Programming, Answer Set Programming and Pseudo-Boolean Optimization (PBO). This paper extends the range of solutions for MQC, by developing two new PBO models and also by introducing models based on the optimization of Satisfiability Modulo Theories (SMT). The models were experimentally compared with existing exact solutions. The results show that for instances with small percentage of quartet errors, the models based on SMT can be competitive, whereas for instances with higher number of quartet errors the PB-models are more efficient.

1 Introduction

Evolutionary trees (or phylogenies) of a given set of taxa (a set of biological species) are often used for studying the evolutionary relationship between the given taxa. Current phylogenetic analyses are carried out using biological data (such as DNA or protein sequences). Since the amount of available data is different for different species, biologists have to balance the number of taxa used and the amount of available data common to all the species. This is know as the Data Disparity problem [1]. In order to overcome the Data Disparity problem, quartet based methods have been suggested.

Quartet based methods divide the task of inferring an evolutionary tree in two subtasks. First, a set of trees for subsets of four taxa is obtained. Then from the trees obtained a global tree is constructed. Considering only four taxa in the first step, maximizes the amount of available data used, and the confidence on the tree produced. These evolutionary trees for subsets of four taxa are called quartet topologies. The quartet topologies produced may still be conflicting or even missing. Given a set of quartet topologies, the problem of computing an evolutionary tree that agrees with the maximum number of quartet topologies is referred to as the *Maximum Quartet Consistency* (MQC) problem. In recent years, solutions based on Answer-Set Programming (ASP) have been identified as the most efficient [18]. This paper proposes alternative combinatorial optimization solutions for the MQC problem. The first set of models extend recent work on using Pseudo-Boolean

Proceedings of the 15th International RCRA workshop (RCRA 2008):
Experimental Evaluation of Algorithms for Solving Problems with Combinatorial Explosion
Udine, Italy, 12–13 December 2008

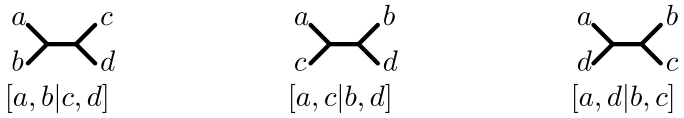

Figure 1: Graphical representation of the quartet topologies $[a, b|c, d]$, $[a, c|b, d]$, $[a, d|b, c]$.

Optimization (PBO) [10]. Moreover, the paper also proposes models based on *Satisfiability Modulo Theories* (SMT).

The paper is organized as follows. The next section introduces the MQC problem and surveys representative solutions. Afterwards, Section 3 summarizes recent work on using PBO for solving the MQC problem, and develops new PBO models. The use of SMT for MQC is described in Section 4. Section 5 analyzes experimental results on real and artificially generated instances. Finally, the paper concludes in Section 6.

2 Maximum Quartet Consistency

This section defines the problem of Maximum Quartet Consistency (MQC), and reviews existing solutions, either heuristic or exact.

Consider a given set of taxa $S = \{s_1, \ldots, s_n\}$. A *phylogeny* on S is a tree such that each leave is mapped to a taxon in S. Each internal node in the phylogeny represents an extinct or hypothesized ancestor of the taxa in the subtrees that the internal node connects. A phylogeny may be *rooted* or *unrooted*. In a rooted phylogeny a link between two nodes represent a parent-child relation, in which the root of the phylogeny is a common ancestor of all the taxa. A phylogeny is called *binary* or *resolved* if every internal node has degree three (in a rooted phylogeny the root node has degree two). In the rest of the paper the term phylogeny refers to an unrooted resolved phylogeny, unless explicitly stated.

A subset of four different taxa is called a *quartet*. A phylogeny for a quartet is called a *quartet topology* (simply refered as *topology*). In this paper, for a quartet $\{a, b, c, d\} \subseteq S$ three different topologies are considered (up to symmetry) denoted by $[a, b|c, d]$, $[a, c|b, d]$ and $[a, d|b, c]$ as represented in Figure 1. Consider a quartet $q \subseteq S$ and a phylogeny T on S. The topology obtained from T by removing all nodes not in the paths connecting the nodes mapped to taxa in q is called the topology of q *derived* from T. On the left side of Figure 2, a phylogeny is shown where the paths that connect taxa a, b, c and f have been marked with dotted lines. In this example the derived topology for quartet $\{a, b, c, f\}$ is $[a, b|c, f]$ (respresented on the right side of Figure 2).

A topology qt is said to be *consistent* with a phylogeny T or T *satisfies* qt if the derived topology for the quartet of qt from T is the same as qt. The topology in Figure 2 is consistent with the phylogeny in the same figure.

Given a set of topologies Q whose set of taxa is S, Q is said to be *complete*

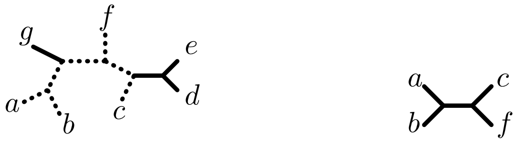

Figure 2: Graphical representation of a phylogeny and of a topology.

if Q contains a topology for each of the C_4^n quartets of S; otherwise Q is said to be *incomplete*. The set Q is said to be *compatible* if there exists a phylogeny T on S that satisfies all the topologies in Q, and T is called the *associated* phylogeny of Q. The MQC problem can now be defined as:

Definition 1 (Maximum Quartet Consistency problem (MQC))**.** *Given a set of taxa $S = \{s_1, \ldots, s_n\}$ and a set of topologies Q on S, the* Maximum Quartet Consistency *problem is the problem of determining a phylogeny T on S that satisfies the maximum number of topologies of Q.*

The MQC problem is a NP-hard problem [2]. If Q is complete, then MQC admits a polynomial-time approximation scheme [8]. If Q is incomplete then MQC is MAX SNP-hard [8]. Existing approaches for the MQC problem can be categorized as either heuristic or exact. Heuristic approaches return an approximation to the optimal phylogeny. Among the many heuristics proposed over the years, a few are overviewed next. The short quartet methods of Erdos et al. [5], the use of semi-definite programming by Ben-Dor et al. [1], the quartet cleaning algorithms of Berry et al. [2]. More recently, in 2008, Wu et al. [15] proposed three algorithms that return a phylogeny with a probability of success.

This paper focuses on exact solutions for the MQC problem, which guarantee that the phylogeny returned satisfies the maximum number of topologies. Existing exact approaches can be categorized in four main classes. Bendor et al. [1] proposed the use of Dynamic Programming to solve the problem of MQC where the topologies given have weights associated to them. Their objective is to compute a phylogeny with a maximal score, that is, a phylogeny whose sum of weights of the satisfied topologies is maximal.

The Fixed-Parameter Algorithm was proposed by Gramm and Niedermeier [7]. The objective of the algorithm is to compute a phylogeny that fails to satisfy no more than a given k topologies. Their algorithm works in time $O(4^k n + n^4)$, where k is the given number of errors and $n = |S|$.

Wu et al. [16] proposed the use of Answer Set Programming (ASP) for MQC. The proposed approach searches for an ultrametric matrix that satisfies the maximum number of topologies. From the ultrametric matrix it is possible to obtain a phylogeny in polynomial time.

More recently, in 2005, Wu et al. [17] presented a lookahead branch-and-bound algorithm for MQC. The algorithm can be seen as an improvement over the Fixed-Parameter Algorithm [7], and has the same running time of $O(4^k n + n^4)$. However, in contrast with the Fixed-Parameter Algorithm, the number of quartet errors is not required to be known.

Finally, in 2008, Morgado and Marques-Silva [10] translated the problem of MQC to Pseudo-Boolean Optimization (PBO) .

3 Pseudo-Boolean Optimization Models for MQC

Pseudo-Boolean Optimization (PBO) models for MQC were first proposed in [10]. This section reviews in some detail the PBO models followed by the description of two new PBO models based on the existing models.

3.1 Existing Pseudo Boolean Optimization Models

Given a complete set of topologies Q over a set of taxa S, the objective of the PBO models for MQC is to encode a phylogeny T of S that satisfies the maximum number of topologies in Q. In order to obtain T, and similarly to the use of ASP, the PBO models encode an ultrametric matrix M. From the ultrametric matrix M it is possible to obtain a rooted phylogeny T.

The *ultrametric matrix* M is a square symmetric matrix $n \times n$ (where $n = |S|$) whose entry values are labels that range from 0 to $n-1$. Each row and column of the ultrametric matrix M is indexed by the given taxa, and the value of an entry $M(i,j)$ is the label of the internal node in T corresponding to the *lowest common ancestor* between leaf nodes s_i and s_j. The labels of the internal nodes of T have the property that for any path from root node to a leaf, the corresponding labels are in decreasing order. To guarantee that M is an ultrametric matrix, M must satisfy the following three properties [16]: (a) all values in the matrix must be between 1 and n, except for the elements on the main diagonal which must be 0, that is $M(i,i) = 0$; (b) M is symmetric, thus $M(i,j) = M(j,i)$; and (c) for each triple (i,j,l) such that $1 \leq i,j,l \leq n$ then

$$\begin{aligned} M(i,j) = M(i,l) &\quad \wedge \quad M(i,l) > M(j,l) \;\; \vee && (1)\\ M(i,j) = M(j,l) &\quad \wedge \quad M(j,l) > M(i,l) \;\; \vee && (2)\\ M(j,l) = M(i,l) &\quad \wedge \quad M(i,l) > M(i,j) && (3) \end{aligned}$$

Wu et al. [18] proved that in order to obtain an optimal phylogeny, the values of the entries of M can be restricted to $M(i,j) \leq \lceil \frac{n}{2} \rceil$.

The PBO models can be divided in three main parts: encoding of the ultrametric matrix; encoding of the consistency of the topologies; and encoding of the cost function.

3.1.1 Basic PBO Model

We start by detailing the Basic PBO model. To encode the ultrametric matrix, the Basic PBO model considers the boolean variables $M_{i,j,k}$ where $1 \leq i < j \leq n$ and $1 \leq k \leq \lceil \frac{n}{2} \rceil$. The semantics of variable $M_{i,j,k}$ is $M_{i,j,k} = 1$ if $M(i,j) = k$, otherwise $M_{i,j,k} = 0$. For a given pair (i,j) one and only one k is allowed to have $M_{i,j,k} = 1$. To ensure this condition, the Basic PBO model introduces the following constraints:

$$\sum_{k=1}^{\lceil \frac{n}{2} \rceil} M_{i,j,k} = 1 \tag{4}$$

The value of an entry $M(i,j)$ is given by $M(i,j) = \sum_{k=1}^{\lceil \frac{n}{2} \rceil} k \times M_{i,j,k}$. To ensure that the resulting matrix M is ultrametric, the Basic PBO model has to guarantee that one of the conditions (1), (2) or (3) is satisfied. The Basic PBO model associates conditions (1), (2) and (3) with new boolean variables $c1_{i,j,l}$, $c2_{i,j,l}$ and $c3_{i,j,l}$. Variable $cx_{i,j,l}$ has value 1 if the corresponding condition x is satisfied, otherwise has value 0. To force one of the variables to have value 1 the Basic PBO model introduces following constraints:

$$c1_{i,j,l} + c2_{i,j,l} + c3_{i,j,l} \geq 1 \tag{5}$$

Consider variable $c1_{i,j,l}$ and the corresponding condition (1). Condition (1) is a logical AND of two conditions (an equality condition and a greater than condition). To encode the value of variable $c1_{i,j,l}$, the Basic PBO model introduces two new boolean variables $c1^{(=)}_{i,j,l}$ and $c1^{(>)}_{i,j,l}$ and associates them with the equality condition and the greater than condition, respectively. $c1_{i,j,l}$ is encoded as an AND gate that is:

$$c1_{i,j,l} \equiv AND(c1^{(=)}_{i,j,l}, c1^{(>)}_{i,j,l}) \tag{6}$$

The values of variables $c1^{(=)}_{i,j,l}$ and $c1^{(>)}_{i,j,l}$ are encoded as comparator circuits on the unary representation of $M(i,j)$, $M(i,l)$ and $M(j,l)$ (that is, using variables $M_{i,j,k}$, $M_{i,l,k}$ and $M_{j,l,k}$). Variables $c2_{i,j,l}$ and $c3_{i,j,l}$ are encoded in an analogous way.

According to [16], an ultrametric matrix M satisfies a topology $[i,j|l,m]$ if and only if one of the following conditions is satisfied:

$$(M(i,l) > M(i,j) \quad \wedge \quad M(j,m) > M(i,j)) \; or \tag{7}$$

$$(M(i,l) > M(l,m) \quad \wedge \quad M(j,m) > M(l,m)) \tag{8}$$

The Basic PBO model considers a new boolean variable q_t that encodes the consistency of the t-th topology. Suppose that the t-th topology in Q is topology $[i,j|l,m]$. The model encodes the value of q_t by introducing two new boolean variables $q1_t$, $q2_t$ and associates them with conditions (7) and (8), respectively. The value of q_t is then encoded as an OR gate:

$$q_t \equiv OR(q1_t, q2_t) \tag{9}$$

The values of variables $q1_t$ and $q2_t$ are encoded in a similar way as variables $c1_{i,j,l}$. Finally, the cost function of the Basic PBO model is to maximize the number of variables q_t with value 1:

$$\max \sum_{t=1}^{|Q|} q_t \tag{10}$$

3.1.2 Reduced PBO model

The Reduced PBO model is based on the Basic PBO model, but where auxiliary variables are used that allow a simpler encoding of greater than conditions ($M(i,j) > M(l,m)$) using fewer variables. In the Reduced model, constraint (4) is split into two conditions:

$$\sum_{k=1}^{\lceil \frac{n}{2} \rceil} M_{i,j,k} \geq 1, \qquad \sum_{k=1}^{\lceil \frac{n}{2} \rceil} M_{i,j,k} \leq 1 \tag{11}$$

The first condition is in PB format and so can be added to the model, whereas the second condition is encoded using sequential counters by Sinz [14]. These counters introduce auxiliary variables $s_k^{i,j}$ with the semantics that if $M(i,j) = a$, then for all $1 \leq k < a$, $s_k^{i,j} = 0$ and for $a \leq k \leq \lceil \frac{n}{2} \rceil$, $s_k^{i,j} = 1$.

Instead of using comparator circuits for encoding constraints of the form $M(i,j) > M(l,m)$ as in the Basic PBO model, the Reduced PBO model uses variables $s_k^{i,j}$ and $s_k^{l,m}$. The constraint $M(i,j) > M(l,m)$ is satisfied if and only if there is a k' such that $s_{k'}^{i,j} = 0$ and $s_{k'}^{l,m} = 1$. The constraint is encoded using for each k an AND gate of $NOT(s_k^{i,j})$ and $s_k^{l,m}$. The final output is encoded as an OR gate of the AND gates. Moreover, all the constraints of the Basic PBO model are maintained in the Reduced PBO model (except for constraint (4)), but using the new variables $s_k^{i,j}$.

3.1.3 Binary PBO model

The last PBO model proposed in [10] was the Binary PBO model. The rationale of the Binary PBO model is to replace the encoding of $M(i,j)$ by changing the semantics of variables $M_{i,j,k}$. In the Binary PBO model, variable $M_{i,j,k}$ represents the k-th bit of the binary representation of $M(i,j)$. As a consequence the limits of index k are reduced to $0 \leq k \leq \lfloor \log_2(\lceil \frac{n}{2} \rceil) \rfloor$, therefore reducing the number of variables.

Variables $M(i,j)$ are now encoded as $M(i,j) = \sum_{k=0}^{\lfloor \log_2(\lceil \frac{n}{2} \rceil) \rfloor} 2^k \times M_{i,j,k}$. Moreover, all the constraints of the Basic PBO model are maintained (using the new limits of index k) except for constraints (4) which are replaced by constraints ($\sum_{k=0}^{\lfloor \log_2(\lceil \frac{n}{2} \rceil) \rfloor} M_{i,j,k} \geq 1$) and a comparator circuit to represent that $\sum_{k=0}^{\lfloor \log_2(\lceil \frac{n}{2} \rceil) \rfloor} M_{i,j,k} \leq \lceil \frac{n}{2} \rceil$.

3.2 New Pseudo Boolean Optimization Models

This section proposes two new PBO models, namely the Unary PBO model and the Binary PBO model with a dedicated comparator.

3.2.1 Unary PBO Model

The rationale of the Unary model is to change the semantics of the selection variables $M_{i,j,k}$ and merge them with the semantics of the auxiliary variables

k	1	2	3	4	5	6
$M_{i,j,k}$	0	0	0	1	0	0
$s_k^{i,j}$	0	0	0	1	1	1

(a)

k	1	2	3	4	5	6
$s_k^{i,j}$	1	1	1	1	0	0

(b)

Figure 3: Example of a valuation for selection and auxiliary variables when $M(i,j) = 4$ and $n = 12$. (a) Represents a valuation as in the Reduced model. (b) Represents a valuation as in the Unary model.

$s_k^{i,j}$. The objective is for the new variables to be able to represent the values in the ultrametric matrix and still allow the simplified encoding of $M(i,j) < M(l,m)$.

Consider for example that $n = 12$, then $k \leq \lceil \frac{n}{2} \rceil = 6$. For some position (i,j) in the matrix, suppose that $M(i,j) = 4$. A valid valuation of the selection variables and of the auxiliary variables is given in Figure 3 (a).

The Unary model uses only the auxiliary variables $s_k^{i,j}$, but with a new semantics, which is if $M(i,j) = a$ for same a, $1 \leq a \leq \lceil \frac{n}{2} \rceil$, for all $1 \leq k \leq a$, then $s_k^{i,j}$ has value 1, and for all $1 < k \leq \lceil \frac{n}{2} \rceil$, then $s_k^{i,j}$ has value 0.

$M(i,j)$ can then be obtained by:

$$M(i,j) = \begin{cases} \sum_{k=1}^{\lceil \frac{n}{2} \rceil - 1} [(s_k^{i,j} - s_{k+1}^{i,j}) \times k] & if(\sum_{k=1}^{\lceil \frac{n}{2} \rceil - 1} [(s_k^{i,j} - s_{k+1}^{i,j}) \times k]) \neq 0 \\ \lceil \frac{n}{2} \rceil & otherwise \end{cases}$$

For the previous example, the new valuation is represented in Figure 3 (b). Given the new semantics, condition $M(i,j) > M(l,m)$ is true if there is a k' such that $s_{k'}^{i,j} = 1$ and $s_{k'}^{l,m} = 0$. The Reduced model considers new boolean variables $e_k^{(i,j)(l,m)}$ The semantics of variables $e_k^{(i,j)(l,m)}$ is $e_k^{(i,j)(l,m)}$ has value 1 if $s_k^{i,j} = 1$ and $s_k^{l,m} = 0$, otherwise $e_k^{(i,j)(l,m)}$ has value 0. Variables $e_k^{(i,j)(l,m)}$ are encoded as AND gates, $e_k^{(i,j)(l,m)} \equiv AND(s_k^{i,j}, NOT(s_k^{l,m}))$.

The greater than condition $M(i,j) > M(l,m)$ is then encoded by a new variable $GT_{(i,j)(l,m)}$ whose value is given as an OR gate of variables $e_k^{(i,j)(l,m)}$, that is:

$$GT_{(i,j)(l,m)} \equiv OR(e_k^{(i,j)(l,m)} : 1 \leq k \leq \lceil \frac{k}{2} \rceil)$$

As with the Reduced model, the cost function and all the constraints of ultrametricity of matrix M and of quartet consistency of the PB model are maintained, but making use of variables $GT_{(i,j)(l,m)}$ as appropriate.

3.2.2 Binary PBO Model with Specialized Comparator

Sinz [14] presented a specialized comparator circuit to represent that a binary word $(b_m b_{m-1} \ldots b_0)$ is smaller or equal to a number n. An optimization of the Binary PBO model corresponds to use the comparator circuit proposed by [14] for encoding that $M(i,j) \leq \lceil\frac{n}{2}\rceil$. Suppose $\lceil\frac{n}{2}\rceil$ is given by the binary word $(b_m b_{m-1} \dots b_0)$. The recursive equations of [14] (that obtain a set of clauses) can be modified to obtain a set of sums:

$$Cp(b_0) = \begin{cases} \{1 - M_{i,j,0}\}, & b_0 = 0 \\ \emptyset & , b_0 = 1 \end{cases}, \quad Cp(b_r) = \begin{cases} \{1 - M_{i,j,r}\} \cup Cp(b_{r-1}), & b_r = 0 \\ \{1 - M_{i,j,r}\} \otimes Cp(b_{r-1}), & b_r = 1 \end{cases}$$

where $A \otimes B = \{x + y | x \in A, y \in B\}$. Then the set of conditions to add to the model is the set $\{s \geq 1 | s \in Cp(b_m)\}$.

All the constraints of the Binary PBO model are maintained except for the comparator circuit that encodes the restriction $M(i,j) \leq \lceil\frac{n}{2}\rceil$, which is now encoded according to the above equations.

4 Satisfiability Modulo Theories for MQC

This section outlines the use of *Satisfiability Modulo Theories* (SMT) for solving the MQC problem, by describing how to optimize a linear cost function subject to a SMT formula. This solution is referred to as the *maximum Satisfiability Modulo Quartet Consistency (mSMQC)* model. The mSMQC model follows the same principles of the PBO models of Section 3, but using the modelling capabilities of SMT. Thus the mSMQC model encodes an ultrametric matrix M and checks for the consistency of the topologies given, against the values in M.

The mSMQC model can be divided in four parts:(1) *Domain Restriction*, defines the variables to be used and limits the domains of the variables; (2) *Ultrametric Constraints*, enforces the matrix to be ultrametric; (3) *Quartet Consistency*, encodes the consistency of the topologies given; (4) *Bounding*, maximizes the number of satisfied topologies. The first three define the problem of determining the consistency of a set of topologies as a SMT-instance. The last seeks to maximize the number of satisfied topologies.

4.1 Quartet Consistency as an Instance of SMT

Domain Restriction. Consider a given set of topologies Q over the set of taxa $S = \{s_1, \dots, s_n\}$. In the PB-models the solution to the MQC problem was given by encoding the resulting phylogeny with an ultrametric matrix M. In the mSMQC model, each position of the matrix is associated with one integer variable containing the value in that position. For each pair of taxa $(s_i, s_j) \in S$, such that $1 \leq i < j \leq n$, the mSMQC model considers the integer variable $M_{i,j}$. Variable $M_{i,j}$ has value k with $1 \leq k \leq \lceil\frac{n}{2}\rceil$ if and only if position (i,j) of M has value k.

To constrain the domain of variables $M_{i,j}$ $(1 \leq i < j \leq n)$, the mSMQC model introduces the following constraints:

$$((M_{i,j} \geq 1) \wedge (M_{i,j} \leq \lceil\frac{n}{2}\rceil)) \tag{12}$$

Ultrametric Constraints. Conditions (1), (2) and (3) in page 4 guarantee the ultrametricity of a matrix. These conditions are guaranteed by the mSMQC model by, for each triple of taxa s_i, s_j, $s_l \in S$, such that $1 \leq i < j < l \leq n$, adding the constraints:

$$\begin{gathered}[(M_{i,j} = M_{i,l}) \wedge (M_{i,l} > M_{j,l})] \ \vee \ [(M_{i,j} = M_{j,l}) \wedge (M_{j,l} > M_{i,l})] \ \vee \\ [(M_{i,l} = M_{j,l}) \wedge (M_{j,l} > M_{i,j})]\end{gathered} \tag{13}$$

Quartet Consistency. The consistency of each topology given is associated with a new integer variable in the model. Suppose the t-th topology in Q is of the form $[s_i, s_j | s_l, s_m]$. A new variable q_t is created and q_t will have value 1 if the topology is satisfied by the current solution, q_t will value 0 otherwise. To constraint the value of q_t, the mSMQC uses the *ite* operator, and adds the following constraints:

$$\begin{aligned}&ite[(((M_{i,l} > M_{i,j}) \wedge (M_{j,m} > M_{i,j})) \vee ((M_{i,l} > M_{l,m}) \wedge (M_{j,m} > M_{l,m}))),\\ &\quad (q_t = 1),\ (q_t = 0)]\end{aligned} \tag{14}$$

All the constraints added to the mSMQC model are constraints of Linear Integer Arithmetic (LIA).

4.2 Maximization Problems in SMT

For the PBO models of Section 3, the maximization was achieved through a cost function. SMT is a decision problem and as such, there is no cost function to be optimized. Nevertheless, recent work addressed optimization problems in SMT, including weighted MAX-SMT [11]. In the following we consider how to maximize the number of topologies satisfied by either using bounds or MAX-SMT.

4.2.1 Maximization by Bounding Bottom-Up

Let *nsqt* be the number of satisfied topologies. Thus $1 \leq nsqt \leq |Q|$, that is at least one topology is satisfied and at most all topologies are satisfied.

The motivation for Bounding by Bottom-Up is to search for the maximum number of satisfied topologies iteratively, by creating instances of SMT and giving them to a SMT solver.

In the Bounding by Bottom-Up approach a lower bound *lb_nsqt* for *nsqt* is maintained. This lower bound has the property that is always possible to satisfy at least *lb_nsqt* topologies. The initial value of *lb_nsqt* is 1. In each iteration, a SMT instance is created as described in Section 4.1 to which is added the constraint:

$$(\sum_{r=1}^{|Q|} q_t \ > \ lb\_nsqt) \tag{15}$$

This instance is given to the SMT solver and if it returns SATISFIABLE, then the lower bound is updated and a new iteration starts with the new lower bound. If the SMT solver returns UNSATISFIABLE then mSMQC returns and the maximum number of satisfied topologies is lb_nsqt.

The way the lower bound is updated depends on the SMT solver used. If the SMT solver is capable of reporting an assignment to the variables every time time it returns SATISFIABLE, then the new lower bound is given by the number of topologies satisfied by the assignment reported. Otherwise, the lower bound can only be increased by one.

4.2.2 Maximization by Bounding Top-Down

The idea of the Bounding by Top-Down is the opposite of the previous Bounding by Bottom-Up. Instead of iterating through satisfiable SMT instances, the Top-Down approach will iterate through a set of unsatisfiable SMT instances till a satisfiable instance is found.

The Bounding by Top-Down maintains an upper bound up_nsqt for $nsqt$. This upper bound up_nsqt has the property that is impossible to satisfy $(up\_nsqt + 1)$ topologies. The initial value of up_nsqt is the total number of topologies $|Q|$. In each iteration the constraint is added to the model:

$$(\sum_{r=1}^{|Q|} q_t \ \geq \ up\_nsqt) \tag{16}$$

The instance is given to a SMT solver, and if the SMT solver returns UNSATISFIABLE then up_nsqt is updated to $(up\_nsqt - 1)$, and a new iteration is started. If the SMT solver returns SATISFIABLE then mSMQC returns with a maximum number of satisfied topologies of up_nsqt.

4.2.3 Maximization by a Mixed Bounding

The idea of the Mixed Bounding approach is to search for the maximum number of satisfied topologies using binary search and augment it with the Bounding by Bottom-Up approach.

The Mixed Bounding approach maintains both the lower bound lb_nsqt and the upper bound up_nsqt (as in the previous bounding approaches). Consider max_nsqt to be the exact maximum number of topologies satisfied. At each iteration, both the following conditions are true: (1) $max\_nsqt \geq lb\_nsqt$: and (2) $max\_nsqt < up\_nsqt$. When $lb\_nsqt + 1 = up\_nsqt$ then max_nsqt has been found to be lb_nsqt.

In each iteration the mSMQC with the Mixed Bounding approach considers the value in the middle of the bounds. Let $mdl\_nsqt = \lfloor \frac{up\_nsqt + lb\_nsqt}{2} \rfloor$. If $mdl\_nsqt = lb\_nsqt$ then max_nsqt is lb_nsqt and mSMQC returns. Otherwise mSMQC with Mixed Bounding adds the constraints:

$$(\sum_{r=1}^{|Q|} q_t \ \geq \ mdl\_nsqt), \qquad (\sum_{r=1}^{|Q|} q_t \ < \ up\_nsqt) \tag{17}$$

If the SMT solver returns UNSATISFIABLE, then the upper bound is updated to the value in the middle ($up\_nsqt = mdl\_nsqt$) and a new iteration starts. If the SMT solver returns SATISFIABLE, then the lower bound is updated and a new iteration starts with the new lower bound. As in the Bounding by Bottom-Up approach, the way the lower bound is updated depends on the SMT solver used. If the SMT solver is capable of returning satisfying assignments on satisfiable instances, then the lower bound is updated to the number of satisfied topologies by the assignment reported. If the SMT solver is unable to report satisfying assignments, then the lower bound is updated to the value in the middle, that is, $lb\_nsqt = mdl\_nsqt$.

4.2.4 Maximization as a MAX-SMT Instance

In MAX-SMT the overall formula is an unsatisfied conjunction of SMT sub-formulas. Each sub-formula is associated with a cost of unsatisfying the sub-formula. The objective is to determine an assignment that minimizes the sum of the costs of the unsatisfied sub-formulas. For example, consider the formula $\varphi \equiv (x < 10) \wedge (y > 20) \wedge (x - y > 40) \wedge (y < 30)$, where subformulas $(x - y > 40)$ and $(y < 30)$ have costs 3 and 10, respectively. Note that not all sub-formulas need to have a cost. The sub-formulas without a cost have to be satisfied, and are called *hard* sub-formulas. The sub-formulas with a cost (which may or may not be satisfied) are called *soft* sub-formulas. One MAX-SMT solution for φ is $x = 5$, $y = 25$ with cost 3.

In the mSMQC model, all the Domain Restriction constraints, Ultrametric constraints and Quartet Consistency constraints have to be satisfied (hard formulas). What may or may not be satisfied is that the variable that encodes the consistency of the topology has value 1. Thus, the soft sub-formula for the t-th topology is:

$$(q_t = 1) \tag{18}$$

The cost of unsatisfying the sub-formula is set to one. The MAX-SMT solver will try to minimize the cost of unsatisfying the topologies.

5 Experimental Results

This section analyzes the results obtained from running the existing ASP solution [18], the PBO models described in Section 3, and the SMT-based solutions proposed in Section 4. First, the experimental setup is described, followed by the analysis of the experimental results for two sets of instances.

The first set of instances were obtained from [18] and correspond to a set of artificially generated instances. Each instance has 10 taxa and is associated with a percentage of modified topologies (to account for errors). The percentages considered were 1%, 5%, 10%, 15%, 20%, 25%, and 30%.

The second set of instances was generated from a real example with 52 taxa[1]. This example corresponds to a complete set of topologies inferred

[1] This example was kindly provided by Dr. Guohui Lin.

	1%	5%	10%	15%	20%	25%	30%
phy smodels	0.37	**1.35**	**3.98**	**8.79**	19.20	33.71	65.48
phy clasp	3.43	5.32	12.61	23.42	57.25	129.63	275.65
PBO-basic bsolo	5.20	23.24	134.14	339.01	863.75	2147.83	3371.67
PBO-basic minisat+	14.04	14.77	29.03	53.15	69.60	119.50	244.12
PBO-basic pueblo	6.89	13.46	31.56	81.06	259.58	497.87	1083.73
PBO-reduce bsolo	2.00	17.94	67.68	255.80	908.31	2346.07	2605.87
PBO-reduce minisat+	0.78	2.24	5.82	12.74	32.73	61.71	123.20
PBO-reduce pueblo	61.77	90.24	141.93	242.97	343.44	742.47	1200.77
PBO-unary bsolo	4.35	13.17	69.94	303.90	826.14	1906.28	3323.00
PBO-unary minisat+	1.04	2.74	5.91	12.65	25.96	56.15	133.89
PBO-unary pueblo	1.00	6.71	18.84	63.10	166.39	443.57	1144.69
PBO-bin bsolo	2.66	15.17	37.67	140.28	397.74	746.46	2070.61
PBO-bin minisat+	2.30	2.89	5.94	11.91	17.96	**28.27**	**53.11**
PBO-bin pueblo	4.98	11.52	28.50	56.68	127.52	191.49	374.43
PBO-bin-sc bsolo	4.91	16.17	102.70	163.43	363.35	759.69	1847.96
PBO-bin-sc minisat+	1.96	3.71	5.45	9.57	**16.27**	29.36	58.88
PBO-bin-sc pueblo	2.14	14.63	34.31	63.20	86.40	201.02	383.69
SMT bottom-up	34.48	37.63	80.18	101.16	139.93	187.72	339.90
SMT top-down	1.51	6.99	24.43	72.31	215.40	565.97	1382.56
SMT mixed	10.77	15.63	23.43	46.09	101.06	171.83	470.86
SMT max-smt	**0.06**	1.57	4.59	10.66	29.02	52.65	88.85

Table 1: Average running times in seconds for the first set of instances (artificial - random generated) by percentage of modified topologies.

from a set of CCV distances. In order to compare the different models, the instance was divided in 10 sets of 10 taxa each. For a considered set of taxa, all the topologies on that set of taxa were gathered to form a new instance. To introduce a higher number of errors, the same idea of modifying a number of random topologies was used. The same percentages were used.

The encoders/solvers used in the experiments can be divided in 3 categories, phy+ASP-solver, PBO-encoder+PBO-solver and SMT-based solver.

phy+ASP-solver. *phy* is an encoder of the MQC problem into ASP, which was obtained from [18]. The approach of using phy with an ASP-solver is currently one of the best approaches in the literature. The ASP encoder used in the experiments is derived from the encoder phy [18]. phy receives as arguments, the number of taxa, the maximum number of allowed quartet errors and the instance with the topologies. In the experiments the maximum number of allowed errors was set to the total number of topologies.

Two ASP-solvers were considered for the experiments, SMODELS [13] and CLASP [6]. Other ASP-solvers were considered, but these two were the only that allowed maximizing with phy. The maximization was done by setting the ASP-solvers to enumerate all the stable models.

PBO-encoder+PBO-solver. The *PBO-encoder + PBO-solver* is characterized by first using one of the encoders described in Section 3 to obtain a pbo file. The pbo file is then given to a pbo-solver. Three different PBO-solvers were used in the experiments, the *bsolo* solver [9], the *minisat+* solver [4] and the *pueblo* solver [12].

SMT-based solver. The models based in SMT described in Section 4 require interaction with the SMT-solver. The SMT-solver used in the exper-

	0%	1%	5%	10%	15%	20%	25%	30%
phy smodels	1.24	1.44	2.50	7.00	13.40	23.59	44.03	69.50
phy clasp	1.99	2.26	4.11	9.17	17.25	45.58	105.82	242.34
PBO-basic bsolo	7.32	8.37	36.24	179.70	460.27	972.40	2473.14	3031.09
PBO-basic minisat+	9.12	4.89	17.48	23.95	56.37	61.42	110.26	177.85
PBO-basic pueblo	3.88	3.22	12.37	44.55	101.41	236.38	521.18	1252.61
PBO-reduce bsolo	1.70	2.90	18.34	84.74	262.27	1031.08	2718.71	3489.77
PBO-reduce minisat+	1.48	1.65	1.98	**4.43**	9.89	18.60	39.04	73.86
PBO-reduce pueblo	93.21	82.51	88.10	167.56	245.20	379.72	638.59	1396.20
PBO-unary bsolo	2.81	3.88	23.47	95.88	372.25	788.38	2146.68	3162.33
PBO-unary minisat+	1.20	1.58	3.96	6.00	9.94	20.73	39.57	89.02
PBO-unary pueblo	2.05	1.49	4.44	20.65	55.54	168.44	544.00	1129.22
PBO-bin bsolo	3.09	6.01	16.61	85.99	171.04	352.50	892.75	1594.01
PBO-bin minisat+	2.17	2.19	4.13	5.84	10.09	16.98	29.50	46.17
PBO-bin pueblo	3.12	7.70	9.43	22.05	55.76	93.14	183.08	418.72
PBO-bin-sc bsolo	4.78	3.45	14.65	68.95	175.88	345.95	905.19	1823.66
PBO-bin-sc minisat+	2.89	2.14	5.04	7.55	**7.84**	**15.60**	**22.54**	**42.97**
PBO-bin-sc pueblo	2.57	3.69	17.04	26.36	74.35	120.68	209.84	421.12
SMT bottom-up	34.20	21.64	36.28	58.15	102.37	146.85	242.39	320.84
SMT top-down	0.46	1.38	6.84	24.94	77.48	231.31	580.47	1395.21
SMT mixed	0.46	9.58	13.32	18.93	47.50	99.73	201.33	436.74
SMT max-smt	**0.04**	**0.05**	**1.48**	4.44	10.91	27.91	51.90	86.17

Table 2: Average running times in seconds for the second set of instances (real instances).

iments was the Yices solver [3]. Yices offers a C API that allows the interaction with the solver, where it is possible to ask the solver for a satisfying assignment to the variables when the solver declares that the smt-instance is satisfiable. Yices is also able to solve max-smt instances.

The results were obtained in a set of nine Intel Xeon 5160, 3GHz dual core servers with 4GB of RAM. The timeout was set to 3600 seconds. Table 1 presents the results for the first set of instances. The percentages of modified topologies are shown in the top of the columns, and the encoders and solvers used are indicated on the left column. In terms of timeouts, the only timeouts were for instances with 20% or 30%. For 20%, the pbo-binary model has timed-out with bsolo and pueblo in one instance, and the pbo-bin-sc model with pueblo has timed-out in one instance. For 30%, the pbo-basic has timed-out seven times with bsolo and once with pueblo. The pbo-reduce with bsolo timeout nine times and pbo-unary has timed out five times with bsolo and once with pueblo.

FromTable 1 and omparing the results of phy with both the ASP-solvers, we can conclude that smodels is on average four times faster than clasp for these instances. Comparing the results among the PBO-models, minisat+ stands out as being the fastest in these instances. The difference among the PB-encoders is not as clear. For instances with percentage of 5% or smaller, the pbo-reduced is the fastest model. On the other hand on instances with percentage of 10% or higher, then the binary models are faster. For the SMT-based approaches, the max-smt approach is the most efficient.

By analyzing all the models and solvers, we can conclude that for instances with percentage of 1% smt max-smt is faster than any other solver.

For instances with percentage of 5%, 10% or 15% the phy + smodels approach has better performance. For instances with 20% or higher the pbo-binary and pbo-binary-sc with minisat+ present better results.

Table 2 shows for the second set of instances the average running time of the instances per encoder/solver and percentage of modified topologies. The timeouts for these instances occurred for 1%, 25% and 30%. The pbo-binary model with bsolo timeout once for 1%. The pbo-basic with bsolo timed-out once for 25% and nine times for 30%. The pbo-unary with bsolo timeout once for 25% and seven times for 30%. The pbo-reduced with bsolo timeout nine times for 30%. For instances that have 0%, 1% or 5%, smt max-smt is the solver with best performance results, whereas if the instance has 10% of modified topologies, the pbo-reduced encoder with minisat+ is the fastest. For instances with 15% or more, the pbo-binary-sc encoder with minisat+ presents the best performance results.

6 Conclusions

This paper proposes new approaches for solving the Maximum Quartet Consistency (MQC) problem. Building on recent work on using Pseudo-Boolean Optimization (PBO) models for MQC [10], a number of new alternative PBO models is proposed. In addition, the paper shows how maximization variants of Satisfiability Modulo Theories (SMT) can be used for solving the MQC problem. To our best knowledge, this is the first concrete example of using SMT for solving computational problems in bioinformatics.

The experimental results suggest that the SMT models are the most adequate for instances with a small percentage of modified quartet topologies, whereas for instances with a larger percentage of modified quartet topologies the most effective solution is currently based on PBO using the Minisat+ solver. The results in the paper provide evidence that there are a number of effective alternative approaches for solving the MQC problem, including ASP, PBO and SMT.

Future research directions will include better integration of SMT solvers in addressing the MQC problem.

Acknowledgments

This work was supported in part by Microsoft Research through its PhD Scholarship Programme.

References

[1] A. Ben-Dor, B. Chor, D. Graur, O. R., and D. Pelleg. From four-taxon trees to phylogenies (preliminary report): the case of mammalian evolution. In *Int. Conf. on Computational Molecular Biology (Recomb98)*, pages 9–19, 1998.

[2] V. Berry, T. Jiang, P. Kearney, M. Li, and T. Wareham. Quartet cleaning: Improved algorithms and simulations. *European Symposium on Algorithms*, 1643:313–324, 1999.

[3] B. Dutertre and L. de Moura. The yices smt solver. Technical report, Computer Science Laboratory, SRI International, available at http://yices.csl.sri.com.

[4] N. Eén and N. Sorensson. Translating pseudo-boolean constraints into sat. *Journal of Satisfiability, Boolean Modeling and Computation*, 2:1–26, 2006. Available at http://minisat.se/MiniSat+.html.

[5] P. Erdos, M. Steel, L. Szekely, and T. Warnow. Constructing big trees from short sequences. *Int. Colloquium on Automata, Languages, and Programming*, 1997.

[6] M. Gebser, B. Kaufmann, A. Neumann, and T. Schaub. *clasp* : A conflict-driven answer set solver. In *Int. Conf. on Logic Programming and Nonmonotonic Reasoning (LPNMR2007)*, volume 4483, pages 260–265, 2007. Available at http://www.cs.uni-postdam.de/clasp.

[7] J. Gramm and R. Niedermeier. A fixed-parameter algorithm for minimum quartet inconsistency. *Computer and System Sciences*, 67(4):723–741, December 2003.

[8] T. Jiang, P. Kearney, and M. Li. Orchestrating quartets: approximation and data correction. *Symp. on Foundations of Computer Science*, pages 416–425, 1998.

[9] V. Manquinho and J. P. Marques-Silva. Satisfiability-based algorithms for boolean optimization. *Annals of Mathematics and Artificial Intelligence*, (40):3–4, 2004. Available at http://sat.inesc-id.pt/bsolo/.

[10] A. Morgado and J. Marques-Silva. A pseudo-boolean solution to the maximum quartet consistency problem. *WCB08 - Workshop on Constraint Based Methods for Bioinformatics*, May 2008.

[11] R. Nieuwenhuis and A. Oliveras. On SAT Modulo Theories and Optimization Problems. In *Theory and Applications of Satisfiability Testing*, pages 156–169, 2006.

[12] H. Sheini and K. Sakallah. Pueblo: A modern pseudo-boolean sat solver. In *Design, Automation and Test in Europe (DATE'05)*, volume 2, pages 684 – 685, 2005.

[13] P. Simons. Computing the stable model semantics. Available at http://www.tcs.hut.fi/software/smodels/.

[14] C. Sinz. Towards an optimal cnf encoding of boolean cardinality constraints. In *Principles and Practice of Constraint Programming*, pages 827–831, 2005.

[15] G. Wu, M. Kao, G. Lin, and J. You. Reconstructing phylogenies from noisy quartets in polynomial time with a high success probability. *Algorithms for Molecular Biology*, 3(1), 2008.

[16] G. Wu, G. Lin, and J. You. Quartet based phylogeny reconstruction with answer set programming. *Int. Conf. on Tools with Artificial Intelligence*, 00:612–619, 2004.

[17] G. Wu, J. You, and G. Lin. A Lookahead Branch-and-Bound Algorithm for the Maximum Quartet Consistency Problem. *Workshop on Algorithms in Bioinformatics*, pages 65–76, October 2005.

[18] G. Wu, J. You, and G. Lin. Quartet-based phylogeny reconstruction with answer set programming. *IEEE/ACM Trans. Comput. Biol. Bioinformatics*, 4(1):139–152, 2007.