A mixed integer linear programming model to reconstruct phylogenies from single nucleotide polymorphism haplotypes under the maximum parsimony criterion

Daniele Catanzaro1*, Ramamoorthi Ravi2 and Russell Schwartz3

Abstract

Background: Phylogeny estimation from aligned haplotype sequences has attracted more and more attention in the recent years due to its importance in analysis of many fine-scale genetic data. Its application fields range from medical research, to drug discovery, to epidemiology, to population dynamics. The literature on molecular phylogenetics proposes a number of criteria for selecting a phylogeny from among plausible alternatives. Usually, such criteria can be expressed by means of objective functions, and the phylogenies that optimize them are referred to as optimal. One of the most important estimation criteria is the parsimony which states that the optimal phylogeny $T^*$ for a set $\mathcal{H}$ of $n$ haplotype sequences over a common set of variable loci is the one that satisfies the following requirements: (i) it has the shortest length and (ii) it is such that, for each pair of distinct haplotypes $h_i, h_j \in \mathcal{H}$, the sum of the edge weights belonging to the path from $h_i$ to $h_j$ in $T^*$ is not smaller than the observed number of changes between $h_i$ and $h_j$. Finding the most parsimonious phylogeny for $\mathcal{H}$ involves solving an optimization problem, called the Most Parsimonious Phylogeny Estimation Problem (MPPEP), which is $\mathcal{NP}$-hard in many of its versions.

Results: In this article we investigate a recent version of the MPPEP that arises when input data consist of single nucleotide polymorphism haplotypes extracted from a population of individuals on a common genomic region. Specifically, we explore the prospects for improving on the implicit enumeration strategy of implicit enumeration strategy used in previous work using a novel problem formulation and a series of strengthening valid inequalities and preliminary symmetry breaking constraints to more precisely bound the solution space and accelerate implicit enumeration of possible optimal phylogenies. We present the basic formulation and then introduce a series of provable valid constraints to reduce the solution space. We then prove that these constraints can often lead to significant reductions in the gap between the optimal solution and its non-integral linear programming bound relative to the prior art as well as often substantially faster processing of moderately hard problem instances.

Conclusion: We provide an indication of the conditions under which such an optimal enumeration approach is likely to be feasible, suggesting that these strategies are usable for relatively large numbers of taxa, although with stricter limits on numbers of variable sites. The work thus provides methodology suitable for provably optimal solution of some harder instances that resist all prior approaches.

Keywords: Combinatorial optimization, Exact algorithms, Mixed integer programming, Phylogeny estimation, Haplotype estimation, Maximum parsimony, Single nucleotide polymorphism, Computational biology

*Correspondence: dacatanz@ulb.ac.be

1 Graphes et Optimisation Mathématique (G.O.M.), Computer Science Department, Université Libre de Bruxelles (U.L.B.), Boulevard du Triomphe, CP 210/01, B-1050, Brussels, Belgium

Full list of author information is available at the end of the article

© 2013 Catanzaro et al; licensee BioMed Central Ltd. This is an Open Access article distributed under the terms of the Creative Commons Attribution License (http://creativecommons.org/licenses/by/2.0), which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.
Background
Molecular phylogenetics studies the hierarchical evolutionary relationships among species, or taxa, by means of molecular data such as DNA, RNA, amino acid or codon sequences. These relationships are usually described through a weighted tree, called a phylogeny, whose leaves represent the observed taxa, internal vertices represent the intermediate ancestors, edges represent the estimated evolutionary relationships, and edge weights represent measures of the similarity between pairs of taxa.

Accurately characterizing evolutionary relationships between organisms and their genomes is the basis of comparative genomic methods for interpreting meaning in sequence data, and for this reason the use of molecular phylogenetics has become widely used (and sometimes indispensable) in a multitude of research fields such as systematics, medical research, drug discovery, epidemiology, and population dynamics [3]. For example, the use of molecular phylogenetics was of considerable assistance in predicting the evolution of human influenza A [4], understanding the relationships between the virulence and the genetic evolution of HIV [5,6], identifying emerging viruses as SARS [7], recreating and investigating ancestral proteins [8], designing neuropeptides causing smooth muscle contraction [9], and relating geographic patterns to macroevolutionary processes [10].

The literature on molecular phylogenetics proposes a number of criteria for selecting the phylogeny of a set of haplotypes extracted from n taxa from among plausible alternatives. The criteria can usually be quantified and expressed in terms of objective functions, giving rise to families of optimization problems whose general paradigm can be stated as follows [11]:

**Problem 1.** – The Phylogeny Estimation Problem (PEP)

\[
\text{optimize } f(T) \\
\text{s.t. } g(\mathcal{H}, T) = 1 \\
T \in \mathcal{T},
\]

where \( T \) a phylogeny of \( \mathcal{H} \), \( \mathcal{T} \) the set of all possible phylogenies of \( \mathcal{H} \), \( f : \mathcal{T} \rightarrow \mathbb{R} \) a function modeling the selected criterion of phylogeny estimation, and \( g : \mathcal{H} \times \mathcal{T} \rightarrow \mathbb{R} \) is a characteristic function equal to one if a phylogeny \( T \) is compatible (according to the selected criterion) for the set \( \mathcal{H} \). A specific optimization problem is completely characterized by defining the functions \( f \) and \( g \), and the phylogeny \( T^* \) that optimizes \( f \) and satisfies \( g \) is referred to as optimal.

One of the classic criteria for phylogeny estimation is the parsimony criterion, which assumes that one taxon evolves from another by means of small changes and that the most plausible phylogeny will be that requiring the smallest number of changes. That evolution proceeds by small rather than smallest changes is due to the fact that the neighborhood of possible alleles that are selected at each instant of the life of a taxon is finite and, perhaps more important, that the selective forces acting on the taxon may not be constant throughout its evolution [12,13]. Over the long term (periods of environmental change, including the intracellular environment), the accumulation of small changes will not generally correspond to the smallest possible set of changes consistent with the observed final sequences. Nevertheless, it is plausible to believe, at least for well-conserved molecular regions where mutations are reasonably rare and unlikely to have occurred repeatedly at any given variant locus, that the process of approximating small changes with smallest changes could provide a good approximation to the true evolutionary process of the observed set of taxa [14]. Such an assumption is likely to be reasonable, for example, in intraspecies phylogenetics, where few generations have elapsed since the observed taxa shared a common ancestor and thus the expected number of mutations per locus is much less than one. When such assumptions hold, a phylogeny of \( \mathcal{H} \) is defined to be optimal under the parsimony criterion if it satisfies the following requirements: (i) it has the shortest length, i.e., the minimum sum of the edge weights, and (ii) it is such that, for each pair of distinct haplotypes \( h_i, h_j \in \mathcal{H} \), the sum of the edge weights belonging to the path from \( h_i \) to \( h_j \) in \( T^* \) is not smaller than the observed number of changes between \( h_i \) and \( h_j \) [11]. The first condition imposes the assumption that the smallest number of mutations consistent with the observed sequences is a good approximation to the true accumulated set of mutations; the second condition correlates the edge weights to the observed data.

The parsimony assumption can be considered accurate in the limit of low mutation rates or short time scales, making it a reasonable model for situations such as analysis of intraspecies variation where little time is presumed to have elapsed since the existence of a common ancestor of all observed taxa. Maximum parsimony also remains valuable as a model for novel methodology development in phylogenetics because of its relative simplicity and amenity to theoretical analysis. Novel computational strategies, such as those developed in this paper, might therefore productively be developed and analyzed in the context of maximum parsimony before being extended to more complicated models of phylogenetics.

Finding the phylogeny that satisfies the parsimony criterion involves solving a specific version of the PEP, called the Most Parsimonious Phylogeny Estimation Problem (MPPEP). Some of the variants of the MPPEP, see e.g., [15,16], can be solved in polynomial time, however, in the most general case, the problem is \( \mathcal{NP} \)-hard [11,17] and this fact has justified the development of a number of exact and approximate solution approaches,
such those described in [11,17,18]. Some recent versions of the MPPEP, such as the Most Parsimonious Phylogeny Estimation Problem from SNP haplotypes (MPPEP-SNP) investigated in this article, play a fundamental role in providing predictions of practical use in several medical bioinformatics applications, such as disease association studies [19] or reconstruction of tumor phylogenies [20,21]. In these contexts, it would be highly desirable to have the most accurate inferences possible for specific applications, but this in turn would imply to have algorithms able to exactly solve instances of such versions. As regards the MPPEP-SNP, the literature describes some (rare) circumstances for which it is possible to solve the problem in polynomial time (see Section Methods). Unfortunately, in the general case the MPPEP-SNP is \( \mathcal{NP} \)-hard and solving provably to optimality therefore generally requires the use of exact approaches based on implicit enumeration algorithms, similar to the mixed integer programming strategies described in [1,2,22].

In this article, we explore the prospects for improving on the implicit enumeration strategy of [1,2] using a novel problem formulation and a series of additional constraints to more precisely bound the solution space and accelerate implicit enumeration of possible optimal phylogenies. We present a formulation for the problem based on an adaptation of [23]’s mixed integer formulation for the Steiner tree problem extended with a number of preprocessing techniques and reduction rules to further decrease its size. We then show that it is possible to exploit the high symmetry inherent in most instances of the problem, through a series of strengthening valid inequalities, to eliminate redundant solutions and reduce the practical search space. We demonstrate through a series of empirical tests on real and artificial data that these novel insights into the symmetry of the problem often leads to significant reductions in the gap between the optimal solution and its non-integral linear programming bound relative to the prior art as well as often substantially faster processing of moderately hard problem instances. More generally, the work provides an indication of the conditions under which such an optimal enumeration approach is likely to be feasible, suggesting that these strategies are usable for relatively large numbers of taxa, although with stricter limits on numbers of variable sites. The work thus provides methodology suitable for provably optimal solution of some harder instances that resist all prior approaches. In future work, it may provide useful guidance for strategies and prospects of similar optimization methods for other variants of phylogeny inference.

Methods

Notation and problem formulation

Before proceeding, we shall introduce some notation and definitions that will prove useful throughout the article.

The human genome is divided in 23 pairs of chromosomes, i.e., organized structures of DNA that contain many genes, regulatory elements and other nucleotide sequences. When a nucleotide site of a specific chromosome region shows a variability within a population of individuals then it is called a Single Nucleotide Polymorphism (SNP). Specifically, a site is considered a SNP if for a minority of the population a certain nucleotide is observed (called the minor allele) while for the rest of the population a different nucleotide is observed (the major allele). The minor allele, or mutant type [24], is generally encoded as ‘1’, as opposed to the major allele, or wild type [24], generally encoded as ‘0’. A haplotype is a set of alleles, or more formally, a string of length \( m \) over an alphabet \( \Sigma = \{0,1\} \) [25].

Given a set \( \mathcal{H} \) of \( n \) haplotypes, denote \( S = \{1,2,\ldots,m\} \) as the set of alleles and \( h_i(s), s \in S \), as the \( s \)-th allele of haplotype \( h_i \in \mathcal{H} \). Given two distinct haplotypes \( h_i, h_j \in \mathcal{H} \), we denote \( S_{hi,hj} \) as the subset of different alleles between \( h_i \) and \( h_j \), \( d_{hi,hj} = \sum_{s \in S_{hi,hj}} |h_i(s) - h_j(s)| \) as the distance between \( h_i \) and \( h_j \), and we say that \( h_i \) and \( h_j \) are adjacent if \( d_{hi,hj} = 1 \). From a biological point of view, the adjacency between a pair of distinct haplotypes means that one of the two haplotypes evolved from the other by mutation of a specific SNP over time.

Consider a graph \( G = (\mathcal{H},E) \) having a vertex for each haplotype in \( \mathcal{H} \) and an edge for each pair of adjacent haplotypes \( h_i, h_j \in \mathcal{H} \). Then, a phylogeny \( T \) of \( \mathcal{H} \) is a spanning tree of \( G \), i.e., an acyclic subgraph of \( G \) in which a pair of vertices \( h_i, h_j \in \mathcal{H} \) is adjacent in \( T \) if \( d_{hi,hj} = 1 \) . It is worth noting that, according to the definition of the edge set \( E \), in general a phylogeny of \( \mathcal{H} \) may not exist as the graph \( G = (\mathcal{H},E) \) might not be connected. To always ensure the existence of a phylogeny for \( \mathcal{H} \), we introduce the set \( \mathcal{H}' \) which consists of the minimum number of haplotypes that should be added to \( \mathcal{H} \) in such a way that, defined \( \overline{\mathcal{H}} = \mathcal{H} \cup \mathcal{H}' \) and \( \overline{E} = \{(h_i,h_j) : h_i, h_j \in \overline{\mathcal{H}} \text{ and } d_{hi,hj} = 1\} \), the graph \( \overline{G} = (\overline{\mathcal{H}},\overline{E}) \) is connected. From a biological point of view, the set \( \mathcal{H}' \) represents the set of haplotypes that are ancestors of the observed ones but either had gone extinct or just were not observed in that sample (also called Steiner nodes).

Denote \( \overline{T} \) as a phylogeny of \( \overline{\mathcal{H}}, \overline{E}(\overline{T}) \) as the edge set of \( \overline{T} \) and \( L(\overline{T}) \) as the length of the phylogeny \( \overline{T} \), i.e., the sum of the distances \( d_{hi,hj} \), for all \((h_i,h_j) \in \overline{E}(\overline{T}) \). Then, the problem of finding a phylogeny of \( \mathcal{H} \) that satisfies the parsimony criterion consists of solving the following optimization problem:

**Problem 2. The Most Parsimonious Phylogeny Estimation Problem from SNP haplotypes (MPPEP-SNP).**

Given a set \( \mathcal{H} \) of \( n \) haplotypes having \( m \) alleles each, find...
the minimum cardinality haplotype set $H'$ to be added to $H$ so that the phylogeny $T^*$ has minimum length.

If the haplotype set $H'$ is empty, i.e., if $G = (H, E)$ is connected, then MPPEP-SNP can be solved in polynomial time as the minimum spanning tree is a (optimal) solution to the MPPEP-SNP. Similarly, if the input haplotype set $H$ satisfies the perfect phylogeny condition, i.e., the requirement that each allele changes only once throughout the cardinality of the set characterizing the programming model for the MPPEP-SNP that uses a minimum spanning tree as the optimal solution, then MPPEP-SNP can be still solved in polynomial time [26-28]. Unfortunately, it is possible to prove that in the general case the MPPEP-SNP is NP-hard (see [1,22]). In fact, the binary nature of the SNP haplotypes allows us to interpret a generic haplotype $h_i \in H$ as a vertex of a $m$-dimensional unit hypercube, its $s$-th allele as the $s$-th coordinate of the vertex $h_i$, and the set $H'$ as the set of Steiner vertices of the unit hypercube. Hence the MPPEP-SNP can be seen as particular case of the Steiner tree problem in a graph, a notorious NP-hard combinatorial optimization problem [29].

Finding the optimal solutions to the MPPEP-SNP is fundamental to validating the parsimony criterion, i.e., to verify whether, for a given instance of the MPPEP-SNP, the results predicted by the criterion match the experimental ones. Unfortunately, the NP-hardness of the MPPEP-SNP limits the size of the instances analyzable to the optimum, which in turn affects the ability to validate the parsimony criterion, hence the practical utility of the predictions themselves. In order to address this concern, in the following section we shall develop an integer programming model able to provide optimal solutions to real instances of the MPPEP-SNP.

A mixed integer programming model for the MPPEP-SNP
Let $V = \{1, 2, \ldots, n, n + 1, n + 2, \ldots, n + |H'|\}$ be the set of potential vertices of a phylogeny $T$ of $H$ and assume the convention to denote the $n$ haplotypes in $H$ as the first $n$ vertices in $V$. The first task in designing an integer programming model for the MPPEP-SNP that uses a polynomial-size number of variables consists of characterizing $V$, i.e., determining an upper and a lower bound on the cardinality of the set $H'$. In fact, observe that $H'$ contains potentially an exponential number of haplotypes, namely all vertices of the unit hypercube that belong to the set $\{0, 1\}^m \setminus H$. However, we can easily bound the cardinality of $H'$ by means of the following approach. Consider the complete graph $\hat{G} = (H, E)$, where $E = \{(h_i, h_j) : h_i, h_j \in H\}$, and construct a minimum spanning tree $T_{\hat{G}}$ of $\hat{G}$. Denote $E(T_{\hat{G}})$ as the set of edges $(h_i, h_j)$ of $T_{\hat{G}}$. Then, an upper bound $UB$ on the overall number of Steiner vertices of the optimal phylogeny $T^*$ can be obtained by computing the sum

$$UB = \sum_{(h_i, h_j) \in E(T_{\hat{G}})} (d_{h_i h_j} - 1).$$

Similarly, denote $L(T_{\hat{G}}) = \sum_{(h_i, h_j) \in E(T_{\hat{G}})} d_{h_i h_j}$, a lower bound $LB$ on the overall number of Steiner vertices of $T^*$ can be obtained as [30,31]:

$$LB = \left\lfloor \frac{L(T_{\hat{G}})}{2} - n + 1 \right\rfloor.$$

Denote $u_i, i \in V$, as a decision variable equal to 1 if the $i$-th vertex of $V$ is considered in the optimal solution to the MPEPP-SNP and 0 otherwise; $x^j_i$ as a decision variable equal to 1 if the vertex $i \in V$ has a change at $s$-th coordinate, and 0 otherwise; and $y_{ij}$ as a decision variable equal to 1 if the pair of distinct vertices $i, j \in V$ is adjacent in the optimal solution to the MPEP-SNP and 0 otherwise. Finally, let $V_H = \{1, 2, \ldots, n\}$, $V_H' = \{n + 1, n + 2, \ldots, n + UB\}$, and $Q = \{1, 2, \ldots, n + LB\}$. Then, a valid formulation for the MPPEP-SNP is the following:

**Formulation 1. Basic Model**

$$\begin{align*}
\min & \sum_{i \in V} \sum_{j \in S} z^i_j \\
\text{s.t.} & \quad x^j_i = h_i(s) \quad \forall s \in S, i \in H \quad (1a) \\
& \quad x^j_i \leq u_i \quad \forall s \in S, i \in V \quad (1b) \\
& \quad z^i_j \geq x^j_i - x^j_i \quad \forall s \in S, i, j \in V : i \neq j \quad (1c) \\
& \quad z^i_j \geq -x^j_i + x^j_i + y_{ij} \quad \forall s \in S, i, j \in V : i \neq j \quad (1d) \\
& \quad \sum_{s \in S} z^i_j = y_{ij} \quad \forall i, j \in V : i \neq j \quad (1e) \\
& \quad y_{ij} \leq u_i \quad \forall i, j \in V : i \neq j \quad (1f) \\
& \quad y_{ij} \leq u_j \quad \forall i, j \in V : i \neq j \quad (1g) \\
& \quad \sum_{j \in V : i \neq j} y_{ij} \geq u_i \quad \forall i \in V \quad (1h) \\
& \quad \sum_{j \in C : i \neq j} y_{ij} \leq \sum_{i \in C} u_i - 1 \quad \forall C \subset V : C \cap V_H \neq \emptyset \quad (1i) \\
& \quad \sum_{i \in V : i \neq j} y_{ij} = \sum_{i \in V} u_i - 1 \quad \forall j \in Q \quad (1j) \\
& \quad \sum_{i \in V : i \neq j} u_i = n + LB \quad \forall j \in Q \quad (1k) \\
& \quad u_i, x^j_i, z^i_j, y_{ij} \in \{0, 1\}. \quad (1\ell)
\end{align*}$$

The objective function (1a) aims at minimizing the length of the optimal phylogeny. Constraints (1b) impose that the coordinates of the first $n$ vertices in $V$ are exactly...
the ones of the input haplotype set $\mathcal{H}$. Constraints (1c) impose that the $s$-th coordinate of vertex $u_i$, $i \in V$, can assume value 1 only if vertex $u_i$ is considered in the optimal solution to the problem. Constraints (1d)-(1e) force variables $z_{ij}^x$ to be one if in the optimal solution to the problem there exist a pair of adjacent vertices $i,j \in V$ having a different value at the $s$-th coordinate. Constraints (1f) impose that in an optimal solution to the problem two distinct vertices $i,j \in V$ can be adjacent only if $d_{hi} = 1$. Constraints (1g)-(1h) impose that in the optimal solution to the problem variable $y_{ij}$ may assume value 1 only if both vertices $i$ and $j$ are considered. Vice versa, constraints (1i) impose that if in the optimal solution to the problem a vertex $u_i$, $i \in V$, is considered then at least one variable $y_{ij}$ must assume value 1. Constraints (1j) and (1k) impose the Generalized Subtour Elimination Constraints (GSEC) [23]. Finally, constraints (1l) impose that the first $n + LB$ vertices in $V$ have to be considered in the optimal solution to the problem.

Note that Formulation 1 can be easily extended to the case in which the haplotypes are represented by multi-character data, i.e., sequences over an alphabet $\Sigma = \{0, 1, 2, \ldots, \gamma\}$. In fact, in such a case it is sufficient to replace each character $c$ in the haplotype by a binary $\gamma$-vector $v$ such that the $s$-th coordinate of $v$ is equal to 1 if the character $c$ is equal to $s$, $s \in \Sigma$, and 0 otherwise. For example, if the generic haplotype were, for example, the string ⟨AACGT⟩, then it could be represented as \((1000 0100 0010 0001)\).

Reducing model size

Formulation 1 is characterized by a polynomial number of variables and an exponential number of constraints. Its implementation can be performed by means of standard branch-and-cut approaches that use GSEC separation oracles such as those described in [32].

It is worth noting that variables $x_{ij}^s$ and $z_{ij}^x$ can be relaxed in Formulation 1 as constraints (1c)-(1e) and the convexity constraint (1f) suffice to guarantee their integrality in any optimal solution to the problem. Moreover, Formulation 1 can be reduced in size by removing those variables that are redundant or whose value is known in the optimal solution to the problem. For example, variables $y_{ij}$ can be removed from Formulation 1 as it is easy to realize that they are redundant. Similarly, all variables $z_{ij}^x$ such that $i,j \in V_H$ and $d_{ij} > 1$ do not need to be defined as their value will be always zero for any $s \in S$ and in any feasible solution to the problem. Variables $u_i$, $i \in Q$, do not need to be declared as their value will be always 1 any feasible solution to the problem. Finally, variables $x_{ij}^s$, $i \in V_H$, can be removed as their value is univocally assigned by the input haplotype set $\mathcal{H}$. The reduction process can be further combined with the preprocessing strategies described in [1] to obtain even smaller formulations. Such strategies allow one to remove alleles from the input haplotype set $\mathcal{H}$ without altering the optimal solution to the problem. For example, suppose that the haplotype set $\mathcal{H}$ is such that there exists an allele $\hat{s} \in S$ such that $h_i(\hat{s}) = 1$, for all $h_i \in H_i$; then it is easy to realize that $\hat{s}$ can be removed from $S$ as in any feasible solution to the problem the $\hat{s}$-th coordinate of any vertex in the phylogeny would be characterized by having $x_{ij}^\hat{s} = 1$. A similar situation occurs when there exists an allele $\hat{s} \in S$ such that $h_i(\hat{s}) = 0$, for all $h_i \in H_i$. Analogously, suppose that the input haplotype set $\mathcal{H}$ is characterized by equal alleles, i.e., by the existence of two alleles, say $\hat{s}_1$ and $\hat{s}_2$, such that $h_i(\hat{s}_1) = h_i(\hat{s}_2)$, for all $i \in S$. Then it is easy to realize that if one removes one of the two alleles from $S$, say $\hat{s}_2$, and multiplies the $\hat{s}_1$-th coordinate by 2 does not alter neither the structure nor the value of the optimal solution to the problem. Describing all the preprocessing techniques for shrinking the input haplotype set $\mathcal{H}$ is beyond the scope of the present article. The interested reader will find a systematic discussion of this topic in [1].

By applying the previously cited reduction strategies to Formulation 1 and denoting $\hat{S}$ as the set of alleles resulting from the application of the preprocessing strategies described in [1], $w^s$ as the number of alleles in $S$ equal to the $s$-th, $s \in \hat{S}$, $Z$ as the set for which variables $x_{ij}^s$ are defined, $R = \{n + LB + 1, n + LB + 2, \ldots, n + UB\}$, and $C_H = \{i \in C : i \in V_H\}$, for any $C \subset V$, the following reduced formulation for the MPPEP-SNP can be obtained:

**Formulation 2. Reduced Model**

\[
\min_{i,j \in V, s \in \hat{S}} \sum_{i,j \in Z} w^s z_{ij}^x \tag{2a}
\]
\[
s.t. \quad x_{ij}^s \leq u_i \quad \forall s \in \hat{S}, i \in R \tag{2b}
\]
\[
\sum_{s \in \hat{S}, i \neq s} x_{ij}^s + h_i(s) - x_{ij}^s \leq 1 \quad \forall s \in \hat{S}, i \in V_H, j \in V_H \tag{2c}
\]
\[
\sum_{s \in \hat{S}, i \neq s} x_{ij}^s - h_i(s) + x_{ij}^s \leq 1 \quad \forall s \in \hat{S}, i \in V_H, j \in V_H \tag{2d}
\]
\[
\sum_{s \in \hat{S}, i \neq s} x_{ij}^s + x_{ij}^s - x_{ij}^s \leq 1 \quad \forall s \in \hat{S}, i,j \in V_H : i,j \in Z \tag{2e}
\]
\[
\sum_{s \in \hat{S}, i \neq s} x_{ij}^s + x_{ij}^s \leq 1 \quad \forall s \in \hat{S}, i,j \in V_H : i,j \in Z \tag{2f}
\]
\[
\sum_{s \in \hat{S}} z_{ij}^s \leq 1 \quad \forall i,j \in V \setminus R : i,j \in Z \tag{2g}
\]
\[ \sum_{j \in S} x_{ij} \leq u_i \quad \forall i \in R, j \in V : i, j \in Z \quad (2h) \]
\[ \sum_{i \in S} z_{ij} \leq u_j \quad \forall j \in R, i \in V : i, j \in Z \quad (2i) \]
\[ \sum_{i \in Q} \sum_{j \in Z} z_{ij} \geq 1 \quad \forall i \in Q \quad (2j) \]
\[ \sum_{i \in R} \sum_{j \in Z} z_{ij} \geq u_i \quad \forall i \in R \quad (2k) \]
\[ \sum_{i \in C \subseteq V : C \cap V_{H'} \neq \emptyset} \sum_{j \in Z} z_{ij} \leq \sum_{i \in R} u_i + |C_H| - 1 \quad (2l) \]
\[ \sum_{j \in S} z_{ij} = n + LB + \sum_{i \in R} u_i - 1 \quad (2m) \]
\[ u_i, x^1_i, z_{ij}, y_j \in \{0, 1\}. \quad (2n) \]

Note that in Formulation 2 variables \( x^2_i \) and \( z_{ij} \) cannot be relaxed anymore.

**Strengthening valid inequalities**

By exploiting the integrality of variables \( u_i, x^1_i, \) and \( z_{ij}, \) a number of valid inequalities can be developed to strengthen Formulation 2.

**Proposition 1.** Constraints

\[ u_{i+1} \leq u_i \quad \forall i \in V \setminus (Q \cup (n + LB)) \quad (3) \]

are valid for Formulation 2.

**Proof.** In a feasible solution to the problem variable \( u_i, \ i \in V \setminus (Q \cup (n + LB)), \) can assume only value 0 or 1. If \( u_i = 0, \) constraint (3) reduces to \( u_{i+1} \leq 0 \) which is trivially valid for Formulation 2. If \( u_i = 1, \) constraint (3) reduces to \( u_{i+1} \leq 1 \) which is again valid.

Constraints (3) impose an ordering on the variables \( u_i, \ i \in R, \) so that vertex \( u_{i+1} \) can be considered in the optimal solution to the problem only if vertex \( u_i \) has been already considered.

**Proposition 2.** Constraints

\[ \sum_{j \in S} \sum_{i \in R} z_{ij} \geq 2u_i \quad \forall i \in R \quad (4) \]

are valid for Formulation 2.

**Proof.** In a feasible solution to the problem a vertex \( u_i, \ i \in V_{H'}, \) cannot be a terminal vertex. In fact, if such a condition held, a cheaper solution could be obtained by dropping \( u_i \) from \( T', \) contradicting the optimality of \( T' \) itself. Hence, the degree of any vertex in \( V_{H'} \) must be at least 2. Now, in a feasible solution to the problem variables \( u_i \in \{0, 1\}. \) If \( u_i = 0, \) constraint (4) reduces to

\[ \sum_{j \in S} \sum_{i \in R} z_{ij} \geq 0 \]

which is trivially valid. Vice versa, if \( u_i = 1, \) constraint (4) reduces to

\[ \sum_{j \in S} \sum_{i \in R} z_{ij} \geq 2 \]

which is again valid for the above arguments. \( \square \)

**Proposition 3.** Constraints

\[ +x^2_i - x^2_j \leq 2(1 - z_{ij}) - \sum_{s \in S \setminus s \neq s_1} z_{ij} \]

\[ \forall s_1, s_2 \in S : s_1 \neq s_2, i, j \in V_{H'} : i, j \in Z \quad (5) \]

\[ -x^2_i + x^2_j \leq 2(1 - z_{ij}) - \sum_{s \in S \setminus s \neq s_1} z_{ij} \]

\[ \forall s_1, s_2 \in S : s_1 \neq s_2, i, j \in V_{H'} : i, j \in Z \quad (6) \]

are valid for Formulation 2.

**Proof.** As observed in the previous proposition, in a feasible solution to the problem \( \sum_{i \in R} z_{ij}, i \in V_{H'}, i, j \in Z, \) can assume only value 0 or 1. If \( \sum_{s \in S \setminus s \neq s_1} z_{ij} = 0, \) then constraint (5) (respectively constraint (6)) reduces to \( +x^2_i - x^2_j \leq 2 \) (respectively \( -x^2_i + x^2_j \leq 2), \) which is trivially valid due to the integrality of variables \( x^1_i. \) If \( \sum_{s \in S \setminus s \neq s_1} z_{ij} = 1, \) then either \( \sum_{s \in S \setminus s \neq s_1} z_{ij} = 1 \) or \( z_{ij} = 1. \) If \( \sum_{s \in S \setminus s \neq s_1} z_{ij} = 1 \) then constraint (5) (respectively constraint (6)) reduces to \( +x^2_i - x^2_j \leq 1 \) (respectively \( -x^2_i + x^2_j \leq 1)), \) which is trivially valid. If \( z_{ij} = 1 \) then constraint (5) (respectively constraint (6)) reduces to \( +x^2_i - x^2_j \leq 0 \) (respectively \( -x^2_i + x^2_j \leq 0), \) which is again valid. \( \square \)

Similar arguments can be used to prove the following proposition:
**Proposition 4. Constraints**

\[ +h_i(s_2) - x_i^2 \leq 2(1 - x_{i2}^1) - \sum_{s \in S_i, s \neq s_1} z_{ij} \]

\[ \forall s_1, s_2 \in \hat{S} : s_1 \neq s_2, i \in V_H, j \in V_H' \]  

\[ -h_i(s_2) + x_i^2 \leq 2(1 - x_{i2}^1) - \sum_{s \in S_i, s \neq s_1} z_{ij} \]

\[ \forall s_1, s_2 \in \hat{S} : s_1 \neq s_2, i \in V_H, j \in V_H' \]  

are valid for Formulation 2.

Given an input haplotype set \( H \) and a pair of non-adjacent haplotypes \( h_1 \) and \( h_j \), there exit multiple equivalent paths that may connect \( h_1 \) and \( h_j \) in the unary hypercube. This characteristic constitutes the principal class of symmetries for the MPPEP-SNP and may lead to poor relaxations for the problem. For example, suppose that the input haplotype set \( H \) is constituted by haplotypes \( h_1 = (00) \) and \( h_2 = (11) \). Then a possible solution to the instance may consist either of a star centered in haplotype \( h_3 = (10) \) or a star centered in haplotype \( h_3 = (01) \) (see Figure 1). Note that both solutions are feasible and optimal for the specific instance. A possible strategy to break this class of symmetries consists of imposing the following constraints:

**Proposition 5. Constraints**

\[ \sum_{p=1}^{s} 2^{i-p} x_{i2}^p \leq \sum_{p=1}^{s} 2^{i-p} x_{i2}^{p+1} \]

\[ \forall s \in \hat{S}, i \in V_H' \setminus R \]  

\[ \sum_{p=1}^{s} 2^{i-p} x_{i2}^p \leq \sum_{p=1}^{s} 2^{i-p} x_{i2}^{p+1} + \sum_{p=1}^{s} 2^{i-p}(1 - u_{i+1}) \]

\[ \forall s \in \hat{S}, i \in R \setminus \{n + UB\} \]  

are valid for Formulation 2.

**Proof.** The statement trivially follows from the integrality of variables \( x_i^1 \) and from constraints (2b).

Constraints (9)-(10) impose an ordering on the coordinates of the vertices in \( V_{H'} \) by means of the smallest big-M possible, i.e., a power of 2. Note that the distinction between constraints (9) and (10) is necessary, as in principle vertices in \( R \) may not be needed in the optimal solution to the problem.

**Proposition 6. Constraints**

\[ \sum_{j \in Z} \sum_{s \in S_i} z_{ij}^s \geq \sum_{j \in Z} \sum_{s \in S_i} z_{ij}^{s+1} \quad \forall i \in V_H' \setminus \{n + UB\} \]

(11)

are valid for Formulation 2.

**Proof.** In a feasible solution to the problem, the sum \( \sum_{j \in Z} \sum_{s \in S_i} z_{ij}^s, i, j \in V_H', i, j \in Z \), can assume only value 0 or 1. If

\[ \sum_{j \in Z} \sum_{s \in S_i} z_{ij}^{s+1} = 0 \]

constraint (11) reduces to

\[ \sum_{j \in Z} \sum_{s \in S_i} z_{ij}^s \geq 0 \]

which is trivially valid. Vice versa, If

\[ \sum_{j \in Z} \sum_{s \in S_i} z_{ij}^{s+1} = 1 \]

constraint (11) reduces to

\[ \sum_{j \in Z} \sum_{s \in S_i} z_{ij}^s \geq 1 \]

which is again valid due to Propositions 1 and 2.

Proposition 6 forces vertices in \( V_{H'} \) to be sorted according to a decreasing degree order. In this way, it is possible to avoid the occurrence of symmetric solutions to the problem differing just for the degree of the Steiner vertices (see e.g., Figure 2).

Let \( Q_3 = \{i, j \in V_H : d_{ij} \geq 3\} \) and \( k \in V, k \notin Q_3 \). Then the following proposition holds:

![Figure 1 An example of two symmetric paths linking haplotypes (00) and (11).](image-url)
Proposition 7. Constraints
\[ \sum_{s \in S} z^i_{sk} + \sum_{s \in S} z^j_{sk} \leq 1 \quad \forall i, j \in Q_3 \] (12)
are valid for Formulation 2.

Proof. In a feasible solution to the problem the path between two distinct haplotypes \( h_i, h_j \in \mathcal{H} \) cannot be shorter than the distance \( d_{h_i h_j} \). Hence, if the distance between \( h_i \) and \( h_j \) is greater or equal to three, vertices \( i \) and \( j \) cannot be adjacent to a same vertex \( k \), i.e., only one of the two sums \( \sum_{s \in S} z^i_{sk} \) or \( \sum_{s \in S} z^j_{sk} \) can be equal to 1. \( \square \)

Note that if \( k \in R \) then (12) can be strengthened by replacing the right-hand-side by \( u_k \). Moreover, Proposition 7 can be generalized as follows. Consider the sets \( Q_d = \{ i, j \in V_\mathcal{H} : d_{ij} \geq d \}, d \in \{ 3, 4, \ldots, m \}, C \subseteq V \) such that \( 2 \leq |C| \leq d - 1 \) and \( C \cap Q_d = \emptyset \), and a path \( p \) that involves only vertices in \( C \). Denote \( p_k \) the \( k \)-th vertex in \( p \). Then the following proposition holds:

Proposition 8. The family of constraints
\[ \sum_{s \in S} z^i_{sp_1} + \sum_{k=1}^{|C|-1} \sum_{s \in S} z^i_{sp_{k+1}} + \sum_{s \in S} z^i_{sp_C} \leq |C| \quad \forall i, j \in K_d, \] (13)
called forbidden path constraints, are valid for Formulation 2.

Proof. Similarly to Proposition 7, in a feasible solution to the problem the path \( p \) between two distinct haplotypes \( h_i, h_j \in \mathcal{H} \) cannot be shorter than the distance \( d_{h_i h_j} \). Hence, if the distance between \( h_i \) and \( h_j \) is greater or equal to \( d \), at most \( |C| \) vertices can belong to \( p \). \( \square \)

Experiments
In this section we analyze the performance of our model to solve the MPPEP-SNP. Our experiments were motivated by a twofold reason, namely: to evaluate, with respect to Formulation 1, the benefits obtained by the removal of the redundant variables and by the inclusion of the valid inequalities previously described; and to allow the analysis of larger datasets with respect to the ones analyzable by means of [1]’s algorithm, currently the best known exact approach to solution of the MPPEP-SNP.

Similar to [1], we emphasize that the experiments aim simply to evaluate the runtime performance of our model for solving MPPEP-SNP. We neither attempt to study the efficiency of MPPEP-SNP for phylogeny estimation nor compare the accuracy of our algorithm to phylogeny estimation solvers that do not use the parsimony criterion. The reader interested in a systematic discussion about such issues is referred to [19,33].

Implementation
We implemented Formulations 1 and 2 by means of Mosel 64 bit 3.2.0 of Xpress-MP, Optimizer version 22, running on a Pentium 4, 3.2 GHz, equipped with 2 GByte RAM and operating system Gentoo release 7 (kernellinux 2.6.17). In both formulations, we computed the overall solution time to solve a generic instance of the problem as the sum of the preprocessing time due to the implementation of [22]’s reduction rules plus the solution time taken by the Optimizer to find the optimal solution to the instance. In preliminary experiments, we observed that Formulation 2 has two main advantages with respect to Formulation 1, namely: it requires much less memory to load the model (at least 27% RAM less in the analyzed instances) and it is characterized by faster linear relaxations at each node of the search tree. As result, Formulation 2 allows potentially the analysis of larger instances than Formulation 1 and may be characterized by faster solution times. Hence, we preferred to use Formulation 2 in our experiments.

We considered two different implementations of Formulation 2, namely: the GESC-based Reduced Model (GESC-RM) and the Flow-based Reduced Model (Flow-RM). The GESC-RM consists of Formulation 2 strengthened by the valid inequalities previously described. The Flow-RM consists of Formulation 2 strengthened by the valid inequalities and such that the GSEC are replaced by a multi-commodity flows. Specifically, by denoting \( f^q_{ij} \) as a decision variable equal to one if there exists a flow from vertex 1 to vertex \( q \in V_\mathcal{H} \) passing through edge \((i, j) \in E\) and 0 otherwise, the Flow-RM can be obtained.
by replacing constraints (2) with:

\[
\begin{align*}
    f^q_{ij} + f^q_{ji} & \leq \sum_{s \in S} z^q_{ij} & \forall q \in V_H : q \neq 1, i, j \in V : i, j \in Z \\
    \sum_{i \in V : i \neq 1} f^q_{ii} & = 1 & \forall q \in V_H : q \neq 1 \\
    \sum_{j \in V : j \neq i} f^q_{ij} - \sum_{j \in V : j \neq i} f^q_{ji} & = 0 & \forall q \in V_H : q \neq 1, i \in V : i \notin [1, q] \\
    \sum_{i \in V : i \neq q} f^q_{iq} - \sum_{i \in V : i \neq q} f^q_{qi} & = 1 & \forall q \in V_H : q \neq 1 \\
    f^q_{ij} & \geq 0 & \forall q \in V_H : q \neq 1, i, j \in V : i, j \in Z.
\end{align*}
\]

(14)  (15)  (16)  (17)  (18)

In preliminary experiments we observed that the Flow-RM outperforms the GESC-RM in terms of solution time. This fact is mainly due to the computational overhead introduced by the GESC separation oracle which seems to be not compensated by the size of the analyzed instances. Hence, we did not consider the GESC-RM any further in our experiments.

During the runtime, we enabled the Xpress-MP automatic cuts and the Xpress-MP pre-solving strategy. Moreover, we also tested a number of generic primal heuristics for the Steiner tree problem to generate a first primal solution at a given node of the search tree and, for example, the valid inequalities (13) are dynamically generated in preliminary experiments we observed that the Flow-RM optimizer, by delaying the solution time of the analyzed instances. Hence, we disabled the use of the generic primal heuristics and enabled the use of the Xpress-MP primal heuristic instead. The source code of the algorithm can be downloaded at http://homepages.ulb.ac.be/~dacatanz/Site/Software_files/iMPPEP.zip.

**Separation oracle for the forbidden path constraints**

When using the Flow-RM, the valid inequalities (3)-(12) are loaded together with the reduced model. On the contrary, the valid inequalities (13) are dynamically generated by means of a separation oracle working as follows. Before loading the reduced model, we precompute the sets \(Q_d\) for all \(d \in \{3, 4, \ldots, m\}\). Let \((\Pi, \Xi, \Xi)\) be the current fractional solution at a given node of the search tree and, for all \(d \in \{3, 4, \ldots, m\}\), consider a pair of vertices \(i, j \in Q_d\). Then, the forbidden path constraints (13) are violated if there exists a path \(p\) having internal vertices in \(C \subseteq V\), \(2 \leq |C| \leq d - 1\), \(C \cap Q_d = \emptyset\), and such that

\[
\sum_{s \in S} z^p_{s1} + \sum_{k=1}^{[C]-1} \sum_{s \in S} z^p_{sp_{k+1}} + \sum_{s \in S} z^p_{sp_C} > |C|.
\]

(19)

Note that searching for the most violated constraint (19) is in general \(NP\)-hard as it involves solving a longest path problem on the weighted graph \(G^{\cap V \setminus Q_d}_w\), i.e., the graph \(G\) whose edges are weighted by variables \(z\) and whose vertex set is restricted to \((V \setminus Q_d) \cup \{i, j\}\). In order to have a fast separation oracle for the forbidden path constraints we do not solve exactly (19) but we use a heuristic approach instead. Specifically, we first sort edges of \(E\) in decreasing order according to their weights and we select two distinct vertices \(v_1, v_2 \in V \setminus Q_d\) such that edge \((v_1, v_2)\) has the largest weight. Subsequently, we set \(C = \{v_1, v_2\}\), mark \(v_1\) and \(v_2\) as visited, and build a simple path from vertex \(i\) to vertex \(j\) passing by \(v_1\) and \(v_2\). If \(p\) is such that (19) is satisfied then we add the constraint

\[
\sum_{s \in S} z^p_{s1} + \sum_{k=1}^{[C]-1} \sum_{s \in S} z^p_{sp_{k+1}} + \sum_{s \in S} z^p_{sp_C} \leq |C|.
\]

(20)

to the formulation; otherwise, we select a different pair of vertices in \(V \setminus Q_d\) and iterate this procedure until either 10 distinct paths have been generated or all possible pairs of vertices in \(V \setminus Q_d\) have been selected. If all vertices have been selected but less than 10 distinct paths have been generated, then we select a larger subset of \(V \setminus Q_d\) (e.g., a triplet of vertices) and we iterate again the previous steps. It is easy to realize that this procedure may potentially generate all the possible paths violating (13). However, we stop the procedure after generating 10 paths or after considering subset \(C\) containing more than 5 vertices as we observed in preliminary experiments that this strategy provides the best trade-off between speed and tightness of the cut.

**Branching strategies**

In preliminary experiments we observed that the standard branching strategy implemented in the Xpress-MP Optimizer is not appropriate for the problem as it is not able to exploit the dissimilarity of the weights \(w^s\) in the objective function. This inconveniently leads to formulations characterized by slow solution times. To improve this aspect we implemented a different strategy consisting of branching on the following constraints:

\[
\sum_{i \in V : i \neq q} z^s_{ij} \leq \alpha & \quad \forall s \in S \\
\sum_{i \in V : i \neq q} z^s_{ij} > \alpha & \quad \forall s \in S
\]

(21)  (22)

where \(\alpha \in \{1, 2, \ldots, q\}\) and \(q = \min\{\sum_{s \in V_H} h(s), n/2\}\). Constraints (21)-(22) limit the number of changes along a phylogeny with respect to a given coordinate \(s \in S\) and tend to be more effective when the weights \(w^s\) are very
dissimilar among them. This branching strategy can be implemented by introducing a decision variable
\[
\beta_{s}^{\alpha} = \begin{cases} 
1 & \text{if the overall number of changes at coordinate } s \in S \text{ of } T^* \text{ is equal to } \alpha \\
0 & \text{otherwise},
\end{cases}
\]
for all } s \in S \text{ and } \alpha \in \{1, 2, \ldots, q\}, \text{ and by adding the following constraints}
\[
\sum_{i,j \in V: i \not\in Z} z^s_{ij} = \sum_{\alpha=1}^{q} \beta_{s}^{\alpha} \quad \forall s \in S
\]
\[
\sum_{\alpha=1}^{q} \beta_{s}^{\alpha} = 1 \quad \forall s \in S.
\]

We observed that even better runtime performance can be obtained by sorting the coordinates of the input haplotypes in decreasing way according to the weights \(w^o\) and by branching first on variables \(\beta_{s}^{\alpha}\), then on variables \(u_i\), and subsequently on variables \(x^s_{i} \) and finally on variables \(z^s_{ij}\).

**Performance analysis**

In order to obtain a measure of the performance of the Flow-RM, we compared [1]'s polynomial-size formulation versus the Flow-RM on [1]'s real instances of the MPPEP-SNP, namely: Human chromosome Y constituted by 150 haplotypes having 49 SNPs each; bacterial DNA constituted by 17 haplotypes having 1510 SNPs each; Chimpanzee mitochondrial DNA constituted by 24 haplotypes having 1041 SNPs each; Chimpanzee chromosome Y constituted by 24 haplotypes having 1041 SNPs each; and a set of four Human mitochondrial DNA from HapMap [35] constituted by 40 haplotypes having 52 SNPs each, 395 haplotypes having 830 SNPs each, 13 haplotypes having 390 SNPs each, and 44 haplotypes having 405 SNPs each, respectively. Such instances consist only of non-recombining data (Y chromosome, mitochondrial, and bacterial DNA) and can be downloaded at http://homepages.ulb.ac.be/~dacatanz/Site/Software_files/iMPPEP.zip.

Table 1 shows the results obtained by such comparison. Specifically, the fourth and fifth columns refer to the gaps (expressed in percentage) of the respective formulations, i.e., to the difference between the optimal value to a specific instance and the value of linear relaxation at the root node of the search tree, divided by the optimal value. The table shows that, excluding the cases in which the solution to a specific instance was trivially a minimum spanning tree (see e.g., Human chromosome Y, Chimpanzee mtDNA, and Chimpanzee chromosome Y), the Flow-RM is always characterized by (sometimes dramatically) smaller gaps. This fact derives on the one hand from the tightness of the Flow-RM with respect to [1]'s polynomial-size formulation and on the other hand from the efficiency of the strengthening valid inequalities previously described. The poor relaxations of their formulation led [1] to propose an alternative and faster exact approach to solution of the MPPEP-SNP based on the brute-force enumeration of all possible Steiner vertices necessary to solve a specific instance of the problem. To speed up the computation, the brute-force enumeration algorithm makes use of a set of reduction rules based on Buneman graph enumeration to decrease the number of Steiner vertices to be considered. Interestingly, despite the differences in terms of implementation language between the two programs (namely, Mosel for the Flow-RM and C++ for [1]'s brute-force enumeration algorithm), the Flow-RM proved to be competitive with [1]'s enumeration algorithm, being able to solve almost all the considered instances within 1 second computing time. Only in two cases, namely Human mtDNA 40 × 52 and Human mtDNA 395 × 830, the Flow-RM needed more than 5 minutes to find the corresponding optimal solutions. The deterioration of the runtime performance in those instances is mainly due to the overhead necessary to load the formulation (that in both cases is considerably bigger than in the other instances) and to an intensive use

| Dataset                  | Haplotypes | SNPs | GAP (%) [1] | GAP (%) Flow-RM | Optimum | MST Solution |
|--------------------------|------------|------|-------------|-----------------|---------|--------------|
| Human chromosome Y      | 150        | 49   | 0.00        | 0.00            | 16      | yes          |
| Bacterial mtDNA         | 17         | 1510 | 26.04       | 20.83           | 96      | no           |
| Chimpanzee mtDNA        | 24         | 1041 | 20.63       | 20.63           | 63      | yes          |
| Chimpanzee chromosome Y | 15         | 98   | 0.00        | 0.00            | 99      | yes          |
| Human mtDNA             | 40         | 52   | 24.66       | 1.37            | 73      | no           |
| Human mtDNA             | 395        | 830  | 22.64       | 7.55            | 53      | no           |
| Human mtDNA             | 13         | 390  | 12.50       | 6.25            | 48      | no           |
| Human mtDNA             | 44         | 405  | 6.98        | 4.65            | 43      | no           |
of the separation oracle for the forbidden path constraints. Possibly, a more thorough implementation of the separation oracle and the use of more performing languages (e.g., C++) could help in speeding up computations in those instances at least.

Interestingly, sometimes in real applications the number of haplotypes can be much bigger than the number of SNPs. Hence, it is important to test the ability of an exact algorithm to tackle instances of the MPPEP-SNP containing e.g., hundreds haplotypes. [1] observed that their brute force enumeration algorithm is able to tackle instances of the problem containing up to 270 haplotypes having up to 9 SNPs each within 12 hours computing time. Unfortunately, the authors also observed that their algorithm is unable to solve larger instances of the MPPEP-SNP, no matter the maximum runtime considered. In this context, the Flow-RM makes the difference, being able to tackle instances of the MPPEP-SNP having up to 300 haplotypes and 10 SNPs within 3 hours computing time. To show this result, we considered a number of random instances of the problem containing 100, 150, 200, 250, and 300 haplotypes, respectively. Fixing the number of haplotypes \( n \in \{100, 150, 200, 250, 300\} \), we created an instance of the problem by generating at random \( n \) strings of length 10 over the alphabet \( \Sigma = \{0, 1\} \). During the generation process, we randomly selected the number of SNPs equal to 1 in a given haplotype, and subsequently we randomly chose the sites of the haplotype to be set to 1. We iterated the instance generation process 10 times for a fixed value of \( n \), obtaining eventually an overall number of 50 random instances of the MPPEP-SNP downloadable at http://homepages.ulb.ac.be/~dacatanz/Site/Software_files/iMPPEP.zip.

The results obtained in our experiments are shown in Table 2. Specifically, the column “Time” refers to the solution time (expressed in seconds) necessary to solve exactly a specific instance of the MPPEP-SNP. Analogously, the column “Nodes” refers to the number of explored nodes in the search tree needed to solve exactly the instance. Table 3 does not report on the performance of [1]’s enumeration algorithm, as their algorithm never found the optimal solution to the analyzed instances within the limit runtime of 3 hours. As a general trend, the table shows that the considered instances can be exactly solved within 1 hour computing time. The only exceptions are constituted by the 7th instance of the group 150 \( \times 10 \), the 9th instance of the group 200 \( \times 10 \), the 2th instance of the group 250 \( \times 10 \), and 3th instance of the group 300 \( \times 10 \) which needed 8719.65, 4600.69, 7757.98, and 5371.05 seconds, respectively, to be solved. These instances are much more sparse than the others, are characterized by smaller reduction ratios, and tend to have more degenerate relaxations than the other instances. The presence of these factors might explain the loss of performance of the Flow-RM.

The results showed that the integrality gaps are usually very low, ranging from 0% to 4.63% and assuming in average a value about 1%, confirming once again the tightness of the Flow-RM and the efficiency of the strengthening valid inequalities.

Finally, we also tested the performance of the Flow-RM on a set of 5 HapMap Human mitochondrial DNA instances of the MPPEP-SNP that were not solvable by using [1]’s brute-force enumeration algorithm, namely: f1 constituted by 63 haplotypes having 16569 SNPs each, i2 constituted by 40 haplotypes having 977 SNPs each, k3 constituted by 100 haplotypes having 757 SNPs each, m4 constituted by 26 haplotypes having 48 SNPs each, and p5 constituted by 21 haplotypes having 16548 SNPs each. Such instances can be downloaded at the same address and consist only of non-recombining data (Y chromosome, mitochondrial, and bacterial DNA).

A part from m4, all the remaining instances gave rise to too large formulations (several hundreds Mbytes RAM) to be handled by the Xpress Optimizer. Hence, instead of analyzing entirely each instance we decomposed it into contiguous SNP blocks and analyzed the most difficult block separately. In more detail, we define \( \mathcal{H}_r \) to be the haplotype matrix obtained by the application of [1]’s reduction rules, we sorted the columns of \( \mathcal{H}_r \) according to an increasing ordering of the weights \( w^s \), \( s \in \mathcal{S} \); subsequently, we considered the submatrices obtained by taking \( k \) contiguous SNPs (or \( k \)-block) in \( \mathcal{S} \), \( k \in \{10, 13, 15\} \). We did not consider greater values for \( k \) as we observed that \( k = 15 \) was already a threshold after which the haplotype submatrix gave rise to too large formulations. For each \( k \)-block \( \mathcal{B} \) in \( \mathcal{H}_r \) we considered the Hamming distance \( d_{h_rh_r} = \sum_{s \in \mathcal{B}} \left| h_r(s) - h_r(s) \right| \) between each pair of distinct haplotypes in \( \mathcal{H}_r \), and chose the \( k \)-block maximizing the sum \( \sum_{h_rh_r \in \mathcal{H}_r} d_{h_rh_r} \). Finally, we assumed three hours as maximum runtime per instance.

Table 3 shows the results obtained in our experiments. As for Table 2, the columns “Time” and “Nodes” refer to the solution time (expressed in seconds) and to the number of nodes in the search tree necessary to solve exactly a specific instance of the MPPEP-SNP, respectively. In such a case, the values in the columns “Gap” refers to the gap between the best primal bound found within the limit time and the root relaxation and “Nodes” refers to the number of nodes explored in the tree search within the limit time.

Table 3 shows that, apart from the instances f1 and m4, the Flow-RM was unable to exactly solve, within the limit time, the considered instances for values of \( k \in \{13, 15\} \). Specifically, The Flow-RM exactly solved in less than a minute the instance f1 when considering values of \( k \in \{10, 13\} \); in 20 minutes the instance i2 when considering \( k = 10 \); in less than 3 minutes the instance k3 when considering \( k = 10 \); and the instance m4 in 5
Table 2 Performances of the Flow-RM on a set of random instances of the MPPEP-SNP

| | Instance | | | | | | |
|---|---|---|---|---|---|---|---|
| | | | | | | | |
| | | | | | | | |
| | | | | | | | |
| | | | | | | | |
| | | | | | | | |
| | | | | | | | |
| | | | | | | | |
| | | | | | | | |
| | | | | | | | |
| | | | | | | | |
| | | | | | | | |
| | | | | | | | |
| | | | | | | | |
| | | | | | | | |
| | | | | | | | |
| | | | | | | | |
| | | | | | | | |
| | | | | | | | |
| | | | | | | | |
| | | | | | | | |
| | | | | | | | |
| | | | | | | | |
| | | | | | | | |
| | | | | | | | |
| | | | | | | | |
| | | | | | | | |
| | | | | | | | |
| | | | | | | | |
| | | | | | | | |
| | | | | | | | |
| | | | | | | | |
| | | | | | | | |
| | | | | | | | |
| | | | | | | | |
| | | | | | | | |
| | | | | | | | |
| | | | | | | | |
| | | | | | | | |
| | | | | | | | |
| | | | | | | | |
| | | | | | | | |
| | | | | | | | |
| | | | | | | | |
| | | | | | | | |
| | | | | | | | |
| | | | | | | | |
| | | | | | | | |
| | | | | | | | |
| | | | | | | | |
| | | | | | | | |
| | | | | | | | |
| | | | | | | | |
| | | | | | | | |
| | | | | | | | |
| | | | | | | | |
| | | | | | | | |
| | | | | | | | |
| | | | | | | | |
| | | | | | | | |
| | | | | | | | |
| | | | | | | | |
| | | | | | | | |
| | | | | | | | |
| | | | | | | | |
| | | | | | | | |
| | | | | | | | |
| | | | | | | | |
| | | | | | | | |
| | | | | | | | |
| | | | | | | | |
| | | | | | | | |
| | | | | | | | |
| | | | | | | | |
| | | | | | | | |
| | | | | | | | |
| | | | | | | | |
| | | | | | | | |
| | | | | | | | |
| | | | | | | | |
| | | | | | | | |
| | | | | | | | |
| | | | | | | | |
| | | | | | | | |
| | | | | | | | |
| | | | | | | | |
| | | | | | | | |
| | | | | | | | |
| | | | | | | | |
| | | | | | | | |
| | | | | | | | |
| | | | | | | | |
| | | | | | | | |
| | | | | | | | |
| | | | | | | | |
| | | | | | | | |
| | | | | | | | |
| | | | | | | | |
| | | | | | | | |
| | | | | | | | |
| | | | | | | | |
| | | | | | | | |
| | | | | | | | |
| | | | | | | | |
| | | | | | | | |
| | | | | | | | |
| | | | | | | | |
| | | | | | | | |
| | | | | | | | |
| | | | | | | | |
| | | | | | | | |
| | | | | | | | |
| | | | | | | | |
| | | | | | | | |
| | | | | | | | |
| | | | | | | | |
| | | | | | | | |
| | | | | | | | |
| | | | | | | | |
| | | | | | | | |
| | | | | | | | |
| | | | | | | | |
| | | | | | | | |
| | | | | | | | |
| | | | | | | | |
| | | | | | | | |
| | | | | | | | |
| | | | | | | | |
| | | | | | | | |
| | | | | | | | |
| | | | | | | | |
| | | | | | | | |
| | | | | | | | |
| | | | | | | | |
| | | | | | | | |
| | | | | | | | |
| | | | | | | | |
| | | | | | | | |
| | | | | | | | |
| | | | | | | | |
| | | | | | | | |
| | | | | | | | |
| | | | | | | | |
| | | | | | | | |
| | | | | | | | |
| | | | | | | | |
| | | | | | | | |
| | | | | | | | |
| | | | | | | | |
| | | | | | | | |
| | | | | | | | |
| | | | | | | | |
| | | | | | | | |
| | | | | | | | |
| | | | | | | | |
| | | | | | | | |
| | | | | | | | |
| | | | | | | | |
| | | | | | | | |
| | | | | | | | |
| | | | | | | | |
| | | | | | | | |
| | | | | | | | |
| | | | | | | | |
| | | | | | | | |
| | | | | | | | |
| | | | | | | | |
| | | | | | | | |
| | | | | | | | |
| | | | | | | | |
| | | | | | | | |
| | | | | | | | |
| | | | | | | | |
| | | | | | | | |
| | | | | | | | |
| | | | | | | | |
| | | | | | | | |
| | | | | | | | |
| | | | | | | | |
| | | | | | | | |
| | | | | | | | |
| | | | | | | | |
| | | | | | | | |
| | | | | | | | |
| | | | | | | | |
Table 3 Performances of the Flow-RM on a set of real instances of the MPPEP-SNP

| Dataset | Haplotypes | SNPs | Block Size | Time (sec.) | GAP (%) | Nodes | MST Solution |
|---------|------------|------|------------|-------------|---------|-------|--------------|
| f1      | 63         | 16569| 10         | 1           |         | 1     | yes          |
|         |            |      | 13         | 56          | 3.11    | 1     | no           |
|         |            |      | 15         | 10286.1     | 26.92   | 773521| no           |
| i2      | 40         | 977  | 10         | 781.85      | 20.00   | 37511 | no           |
| k3      | 100        | 757  | 10         | 150         | 7.65    | 353   | no           |
|         |            |      | 13         | 588.38      | 14.29   | 11265 | no           |
| m4      | 26         | 48   | 10         | 5           | 5.88    | 109   | no           |
| p5      | 21         | 16548| 10         | 22283.4     | 50.79   | 6125448| no          |

of O.rufipogon DNA (red rice) having 1043 sites each; 80 human mtDNA sequences having 245 sites each; 50 HIV-1 reverse transcriptase amino acid sequences having 176 sites each; a set of 500 sequences of mtDNA from the NCBI BLASTN best aligned taxa having 3000 sites each; a set of 500 sequences of mtDNA from the NCBI BLASTN best aligned taxa having 5000 sites each; and a set of 500 sequences of mtDNA from the NCBI BLASTN best aligned taxa having 10000 sites each. When running the same experiments described in [2] we registered a very poor performance for the Flow-RM, mainly due to the large dimension of the considered instances and the presence of symmetries despite the use of constraints (13)-(15). We observed that the combination of these two factors increased the runtime performance of the Flow-RM of 2-3 orders of magnitude with respect to [2] approach. However, we also observed that when performing [2]'s "window analysis" (i.e., when decomposing into blocks of 10 SNPs the input matrix) the Flow-RM performed better than [2]'s, being characterized by an average solution time of 8.27 seconds. This fact suggests that, when considering instances constituted by less than a dozen sites, an exact approach entirely based on integer programming may perform better than the implicit enumeration of the vertices of the generalized Buneman graph. Vice-versa, for larger instances the implicit enumeration of the vertices of the generalized Buneman graph appears more suitable.

Conclusion
In this article we investigated the Most Parsimonious Phylogeny Estimation Problem from Single Nucleotide Polymorphism (SNP) haplotypes (MPPEP-SNP), a recent version of the phylogeny estimation problem that arises when input data consist of SNPs extracted from a given population of individuals. The MPPEP-SNP is \( \mathcal{NP} \)-hard and this fact has justified the development of exact and approximate solution approaches as those described in [1,19,22,26-28]. We explored the prospects for improving on the strategy of [1,2] using a novel problem formulation and a series of additional constraints to more precisely bound the solution space and accelerate implicit enumeration of possible optimal phylogenies. We present a formulation for the problem based on an adaptation of [23]'s mixed integer formulation for the Steiner tree problem extended with a number of preprocessing techniques and reduction rules to further decrease its size. We then show that it is possible to exploit the high symmetry inherent in most instances of the problem, through a series of strengthening valid inequalities, to eliminate redundant solutions and reduce the practical search space. We demonstrate through a series of empirical tests on real and artificial data that these novel insights into the symmetry of the problem often leads to significant reductions in the gap between the optimal solution and its non-integral linear programming bound relative to the prior art as well as often substantially faster processing of moderately hard problem instances. More generally, the work provides an indication of the conditions under which such an optimal enumeration approach is likely to be feasible, suggesting that these strategies are usable for relatively large numbers of taxa, although with stricter limits on numbers of variable sites. The work thus provides methodology suitable for provably optimal solution of some harder instances that resist all prior approaches. Our results may provide also useful guidance for strategies and prospects of similar optimization methods for other variants of phylogenetic inference. In fact, if appropriately adapted, some of the results we presented here (e.g., symmetry breaking strategies) can be generalized with respect to other phylogenetic estimation criteria (e.g., the likelihood criterion) and provide important computational benefits.

Competing interests
The authors declare that they have no competing interests.

Authors' contributions
The authors equally contributed to conceive the work and write the article. DC implemented the algorithms and performed computations. All authors read and approved the final manuscript.

Acknowledgements
The first author acknowledges support from the Belgian National Fund for Scientific Research (FNRS) of which he is Chargé de Recherches, the Belgian American Educational Foundation (BAEF) of which he is honorary fellow, and
the Carnegie Mellon University (CMU). Part of this work has been developed while Dr. Catanzaro was visiting the Tepper School of Business and the Department of Biological Sciences of the CMU during the academic years 2010-2011 and 2011-2012. This work was supported in part by U.S. National Institutes of Health awards 1R01CA140214 and 1R01AI076318.

Author details
1. Graphes et Optimisation Mathématique (G.O.M.), Computer Science Department, Université Libre de Bruxelles (U.L.B.), Boulevard du Triomphe, CP 210/01, B-1050, Brussels, Belgium. 2. Tepper School of Business, Carnegie Mellon University, 5000 Forbes Avenue, Pittsburgh, PA 15213-3890.
3. Department of Biological Sciences and Lane Center for Computational Biology, Carnegie Mellon University, 4400 Fifth Avenue, Pittsburgh, PA, 15213.

Received: 13 June 2012 Accepted: 2 January 2013 Published: 23 January 2013

References
1. Sridhar S, Lam F, Bliehol GE, Ravi R, Schwartz R. Mixed integer linear programming for maximum parsimony phylogeny inference. IEEE/ACM Trans Comput Bioinform 2008, 5(3):323–331.
2. Misra N, Bliehol GE, Ravi R, Schwartz R. Generalized Buneman pruning for inferring the most parsimonious multi-state phylogeny. J Comput Biol 2011, 18(3):445–457.
3. Fitch W, Sturmfels B. The mathematics of phylogenomics. SIAM Rev 2007, 49(3):3–31.
4. Bush RM, Bender CA, Subbarao K, Cox NJ, Fitch WM. Predicting the evolution of human influenza A. Science 1990, 286(5446):1921–1925.
5. Ross HA, Rodrigo AG. Immune-mediated positive selection drives human immunodeficiency virus type 1 molecular variation and predicts disease duration. J Virol 2002, 76(23):11715–11720.
6. Ou CY, Cesarelli CA, Myers G, Bandea CJ, Luo CC, Korber B, Mullins J, Schochetman G, Berkelman RL, Economou AN, Witte JF, Furman LJ, Satten GA, MacInnes KA, Curran JW, Jaffe HW. Molecular epidemiology of HIV transmission in a dental practice. Science 1992, 256(5060):1165–1171.
7. Maria MA, Jones SJ, Aslouch SJ, Holt RA, Brooks-Wilson A, Butterfield YS, Khattra J, Asano JK, Barber SA, Chan SY, Cloutier A, Coughlin SM, Freeman D, Gill N, Griffith OD, Leach SR, Mayo M, McDonald H, Montgomery SB, Pachter L, Sturmfels B. The mathematics of phylogenomics. SIAM Rev 2007, 49(3):3–31.