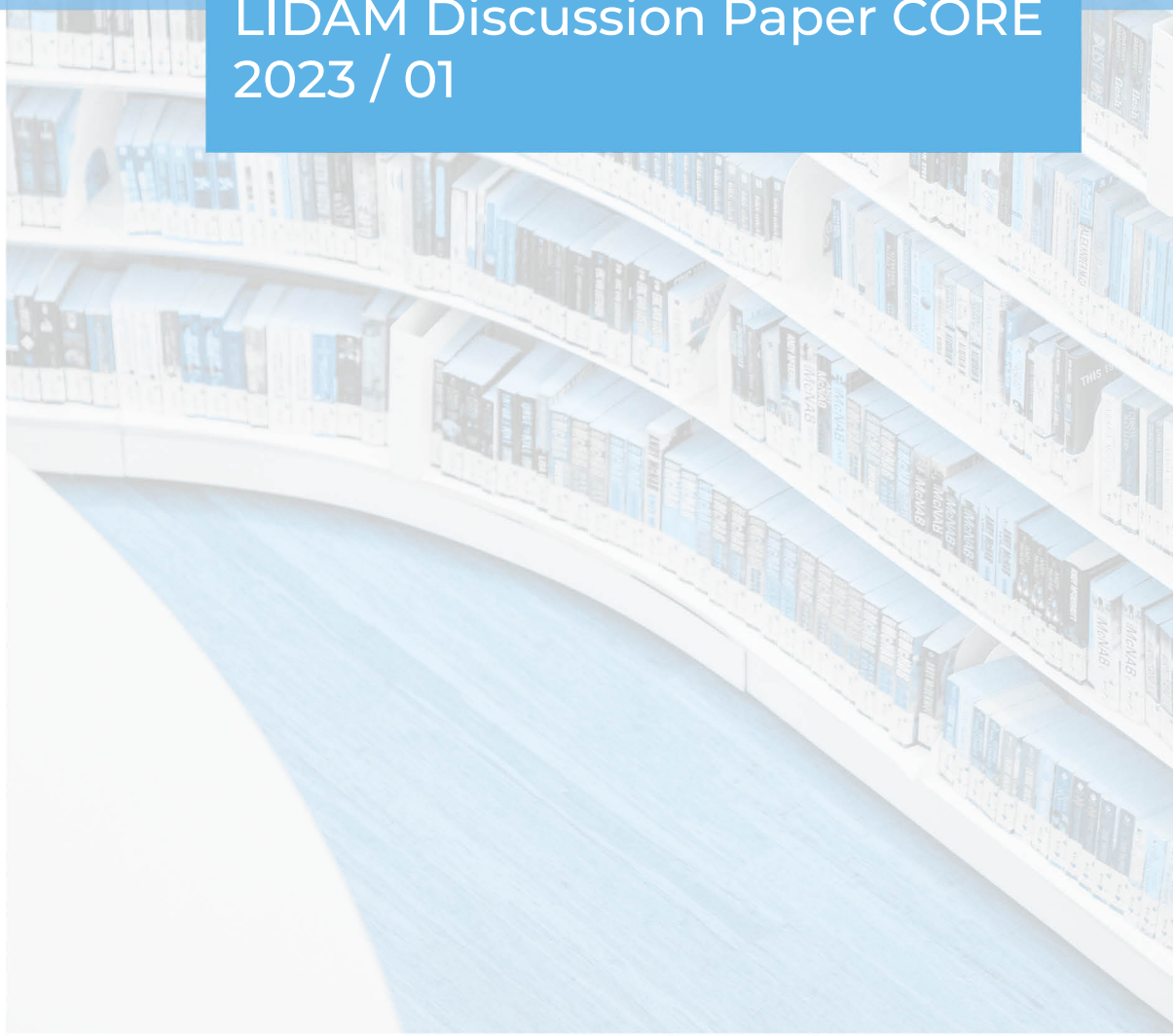
A MASSIVELY PARALLEL EXACT SOLUTION ALGORITHM FOR THE BALANCED MINIMUM EVOLUTION PROBLEM

Daniele Catanzaro, Martin Frohn, Olivier Gascuel, Raffaele Pesenti

LIDAM Discussion Paper CORE
2023 / 01



UCLouvain

CORE | LIDAM
Louvain Institute of Data Analysis and Modeling in economics and statistics

CORE

Voie du Roman Pays 34, L1.03.01

B-1348 Louvain-la-Neuve

Tel (32 10) 47 43 04

Email: immaq-library@uclouvain.be

https://uclouvain.be/en/research-institutes/lidam/core/core-discussion-papers.html

A Massively Parallel Exact Solution Algorithm for the Balanced Minimum Evolution Problem

Daniele Catanzaro[a,1], Martin Frohn[b,*,2], Olivier Gascuel[c,3] and Raffaele Pesenti[d,4]

[a]*Center for Operations Research and Econometrics, Université Catholique de Louvain, Voie du Roman Pays 34, L1.03.01, B-1348 Louvain-la-Neuve, Belgium.*

[b]*Department of Mathematics and Computer Science, Eindhoven University of Technology, De Groene Loper 5, 5612 AZ Eindhoven, Netherlands.*

[c]*Institut de Systématique, Evolution, Biodiversité (ISYEB - UMR 7205 CNRS & Muséum National d'Histoire Naturelle), Paris, France.*

[d]*Department of Management, Università Ca' Foscari, San Giobbe, Cannaregio 837, I-30121 Venezia, Italy.*

ARTICLE INFO

Keywords:
Combinatorial optimization; integer programming; network realization; network design; balanced minimum evolution; phylogenetics; distance methods; parallel branch-and-bound.

ABSTRACT

We build upon recent theoretical advances on the *Balanced Minimum Evolution Problem* (BMEP) to design a new massively parallel exact solution algorithm that proves to be up to one order of magnitude faster than the current state-of-the-art under the same computing settings and environment. We also investigate, for the first time, the theoretical connections between numerical stability and statistical consistency of the BMEP and we show that some rescaling techniques introduced to numerically stabilize the problem may affect negatively the statistical consistency of the optimal solution to the problem.

1. Introduction

Consider a set $\Gamma = \{1, 2, \dots, n\}$ of $n \geq 3$ distinct aligned molecular sequences, hereafter referred to as *taxa*, and a $n \times n$ symmetric distance matrix $\mathbf{D}$, whose generic entry d_{ij} – equal to zero on the main diagonal and strictly positive otherwise – encodes a measure of the similarity (or an estimate of the *evolutionary distance* [31, 52]) between the pair of taxa $i, j \in \Gamma$. A *phylogeny* of Γ is an ordered triplet $(T, \phi, \mathbf{w})$ such that: T is an *Unrooted Binary Tree* (UBT) having n leaves; ϕ is a bijection between the leaves of T and the taxa in Γ; and $\mathbf{w}$ is a vector of non-negative weights associated to the edges of T [12] (see Figure 1).

Taxa	Sequence		
Taxon 1	A	A	A
Taxon 2	A	C	C
Taxon 3	C	G	C
Taxon 4	C	C	G
Taxon 5	G	A	G

$$\mathbf{D} = \begin{array}{c} \\ 1 \\ 2 \\ 3 \\ 4 \\ 5 \end{array} \begin{array}{c} \begin{array}{ccccc} 1 & 2 & 3 & 4 & 5 \end{array} \\ \begin{bmatrix} 0 & 2 & 3 & 3 & 2 \\ 2 & 0 & 2 & 2 & 3 \\ 3 & 2 & 0 & 2 & 3 \\ 3 & 2 & 2 & 0 & 2 \\ 2 & 3 & 3 & 2 & 0 \end{bmatrix} \end{array}$$

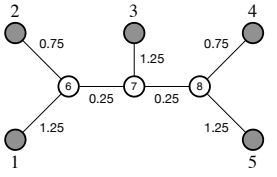


Figure 1: A small example of a set Γ of five DNA sequences (*taxa*) and of a possible distance matrix $\mathbf{D}$ associated to Γ. In this specific case, the entries of $\mathbf{D}$ encode Hamming distances between pairs of molecular sequences. The weighted tree T on the right encodes a phylogeny of Γ. Note that each internal vertex of T has degree three (i.e., T is unrooted and binary) and that each taxon in Γ figures as a leaf of T (i.e., Γ is the leaf-set of T). The vector $\mathbf{w}$ of non-negative weights is obtained from $\mathbf{D}$ by using Paulin's direct calculation formulas (see [58] or [12] for a recent tutorial on the BMEP). The length of the unique path in T connecting taxa 1 and 3, denoted as τ_{13}, is equal to 4. Similarly, τ_{23} is equal to 3.

The *Balanced Minimum Evolution Problem* (BMEP) is a nonlinear *Network Realization Problem* (NRP) [5, 8, 9]

*Corresponding author

ORCID(s): 0000-0001-9427-1562 (D. Catanzaro); 0000-0002-5002-4049 (M. Frohn); 0000-0002-9412-9723 (O. Gascuel); 0000-0001-5890-4238 (R. Pesenti)

[1]Email: daniele.catanzaro@uclouvain.be (Daniele Catanzaro)

[2]Email: m.frohn@tue.nl (Martin Frohn)

[3]Email: olivier.gascuel@mnhn.fr (Olivier Gascuel)

[4]Email: pesenti@unive.it (Raffaele Pesenti)

that consists of finding a phylogeny T of Γ that minimizes the *length function*

$$L(T) = \sum_{e \in E(T)} w_e = \sum_{i \in \Gamma} \sum_{j \in \Gamma \setminus \{i\}} \frac{d_{ij}}{2^{\tau_{ij}}}, \tag{1}$$

where $E(T)$ is the edge-set of T and τ_{ij} represents the *path-length* between taxa i and j in T, i.e., the number of edges belonging to the (unique) path in T connecting taxon i to taxon j [14, 25, 53]. For example, the phylogeny shown in Figure 1 is provably optimal for the distance matrix shown at the center of the same figure and its length is equal to 5.75.

The BMEP has been introduced in the literature on molecular phylogenetics by Desper and Gascuel [24], based on a particular edge-weight estimation criterion proposed by Pauplin [58] in 2000 (see [12] for a recent historical review of the problem). The optimal solution T^* to the BMEP provably minimizes the cross-entropy associated to the molecular sequences of taxa and provides a *statistically consistent estimate* of their evolutionary relationships [12, 13, 31, 38, 40]. This information, in turn, is central in many practical applications arising from medicine and life sciences [12].

Besides being a NRP, the BMEP can also be seen as: (i) a restriction of the *Minimum Evolution Problem* discussed in [11]; (ii) a version of the *Steiner tree problem* in which the number of Steiner nodes is known a priori, the tree must be unrooted and binary, and the length function depends upon the topology of the tree [20, 21, 27, 28, 41, 44, 50, 51, 60, 67]; and (iii) a generalization of the *Quadratic Assignment Problem* (QAP) [18], in which one of the two matrices involved in the Shur product must be determined within a factor of the set of matrices $\tau = \{\tau_{ij}\}$ that encode UBTs with n leaves [2]. The BMEP is $\mathcal{NP}$-hard and inapproximable within c^n, for some constant $c > 1$, unless $\mathcal{P} = \mathcal{NP}$ [32]. This negative result persists even in the case the underlying UBT is fixed and just the assignment of taxa to the leaves of U must be identified so as to minimize (1) [35]. The problem is instead solvable in polynomial-time if the input distance matrix $\mathbf{D}$ is *additive*, i.e., if its entries satisfy the following condition [38]:

$$d_{ij} + d_{kr} \leq \max\{d_{ik} + d_{jr}, d_{ir} + d_{jk}\} \quad \forall\, i, j, k, r \in \Gamma.$$

In the case $\mathbf{D}$ is just *metric*, i.e., if its entries satisfy just the triangle inequality, then the optimal solution to the BMEP can be approximated within a factor of two [32].

The last 20 years registered extensive research efforts focused on the characterization of several theoretical and algorithmic aspects of the BMEP, including its statistical consistency [25, 38, 40], its connections with information theory [13], its combinatorics [14, 17, 23, 33, 34, 42, 63], as well as the development of implicit enumeration algorithms [2, 14, 53] and heuristics [32, 38, 53] to tackle and solve instances of practical size. This article further adds to this literature in that it presents a massively parallel exact solution algorithm for the BMEP that sets a new benchmark in this specific area.

The earliest exact solution algorithm described in the literature on the BMEP has been proposed by Pardi [53] in 2009. The author derived a number of lower bounds on the value of T^* based on Semple and Steel's combinatorial interpretation of the length function (1) [63]. These bounds, combined with an implicit enumeration of all of the possible solutions to the problem, led to a Branch-&-Bound algorithm able to solve instances containing up to 20 taxa within one hour computing time [14].

In 2011, Aringhieri et al. [2] proposed an alternative exact solution algorithm for the problem, based on *tree isomorphism* [47]. The authors observed that groups of phylogenies whose underlying UBTs are isomorphic identify specific classes of equivalence, called *unlabeled UBTs*, whose number grows exponentially with n, but in a way much slower than the corresponding number of phylogenies of Γ (which is known to be equal to $(2n-5)!!$, see Table 1). Hence, the authors proposed to enumerate just unlabeled UBTs and then find, for each of them, the best possible assignment of taxa to the leaves of the unlabeled UBT so as to minimize (1). Although the minimization of the (quadratic) assignment of taxa to the leaves of each unlabeled UBT is an $\mathcal{NP}$-hard problem [35], this task can be solved in parallel [2], by enabling solution times shorter than those of Pardi's algorithm [53]. The enumeration of unlabeled UBTs can be done off-line and is quite fast for small values of n. However, generating and storing unlabeled UBTs becomes intractable for $n \geq 27$, and this fact in turn disables the use of Aringhieri et al.'s approach [2] to solve practical instances of the problem.

A third implicit enumeration-based exact solution algorithm for the BMEP was proposed by Catanzaro et al. [14] in 2012. The theoretical foundations of this algorithm lies in the deep connections between the BMEP and information

Taxa	Phylogenies	Unlabeled UBTs	Taxa	Phylogenies	Unlabeled UBTs
3	1	1	14	$3.2 \cdot 10^{11}$	135
4	3	1	15	$7.9 \cdot 10^{12}$	265
5	15	1	16	$2.1 \cdot 10^{14}$	552
6	105	2	17	$6.2 \cdot 10^{15}$	1132
7	945	2	18	$1.9 \cdot 10^{17}$	2410
8	10,395	4	19	$6.3 \cdot 10^{18}$	5098
9	135,135	6	20	$2.2 \cdot 10^{20}$	11 020
10	2,027,025	11	25	$2.5 \cdot 10^{28}$	565 734
11	34,459,425	18	26	$1.19 \cdot 10^{30}$	1 265 579
12	654,729,075	37	...	...	...
13	$1.4 \cdot 10^{10}$	66	n	$(2n-5)!!$	A000672

Table 1
Number of phylogenies and unlabeled phylogenies for increasing number of taxa. The general formula for the unlabeled UBTs is quite long and is omitted.

theory [13, 57] (see the next sections). These connections allowed the authors to identify a number of fundamental combinatorial equalities and inequalities characterizing UBTs that, appropriately included in specific integer programming formulations for the BMEP, allow to define very tight lower bound on the optimal solution to the problem. Thanks to the use of specific branching strategies, the authors transformed the best of these relaxations into an implicit enumeration algorithm able to outperform Pardi's [53] and Aringhieri et al.'s algorithms [2], by solving instances of the BMEP containing up to 26 taxa within one hour computing time. This performance makes Catanzaro et al.'s approach [14] the current state-of-the-art sequential exact solution algorithm for the BMEP.

Because practical instances of the BMEP may include even hundreds or thousands of taxa (see, e.g., SARS-CoV-2 genomes available at `https://www.ncbi.nlm.nih.gov/genbank/sars-cov-2-seqs/` and [3, 6, 7, 22, 26, 40, 46, 49, 53, 54, 55, 56, 68]), it is highly desirable to design exact solution algorithms able to tackle and solve instances much larger than Catanzaro et al.'s algorithm [14]. In this article we make a further step in this direction, by building upon the results described in [2, 11, 13, 14, 15, 17, 39, 40, 63] and by taking advantage of specific tree-coding schemes, mixed integer programming, and the multicore (shared-memory) features of modern CPUs. The use of massive parallelism in the context of the BMEP is quite recent. We introduced it in Catanzaro and Pesenti [15] to enumerate by brute-force all of the vertices of the BMEP polyhedron, by enabling so the analysis of its polyhedral combinatorics discussed in [17]. In that work, we encoded phylogenies as Prüfer codes [10] and we showed algorithms and methods to parallelize this enumeration both on CPUs and GPUs. We show in this article that the combination of massive parallelism with new theoretical results on the polyhedral combinatorics of the BMEP [17] and on its relationships with information theory [13] allows to define a new reference in terms of performance that can be concisely summarized as follows: up to one order of magnitude faster than the current state-of-the-art exact sequential algorithm in the same computing environment on generic instances of the BMEP and up to 25% more taxa solved within a prefixed time limit.

A second contribution of this article concerns the study of the relationships between numerical stability and statistical consistency of the BMEP. These relationships remained marginal in literature on the BMEP so far, but their characterization may have deep implications in the context of practical phylogenetic analyses. We show here that there exists some theoretical limits on the performance achievable by an exact algorithm for the BMEP, that materialize whenever the generic term $d_{ij}2^{-\tau_{ij}}$ in the BMEP length function approaches (or get smaller than) the machine epsilon. We will see that some natural rescaling techniques introduced to numerically stabilize the problem do not ensure the statistical consistency of its optimal solution, by leading so to misleading phylogenetic predictions.

The paper is organized as follows. In the next section, we introduce some notation and definitions that will prove useful throughout the article as well as some fundamental equations that describe the combinatorial properties of phylogenies. In Section 2 we introduce some notation and discuss fundamental properties of topological distances. In Section 3, we briefly recall, for the sake of completeness, the current state-of-the-art sequential exact solution algorithm for the BMEP. In Sections 4 and 5, we discuss possible strategies to obtain lower bounds and upper bounds on the optimal solution to the BMEP, respectively. In Section 6, we develop a new massively parallel exact solution algorithm for the BMEP and in Section 7 we report on its performance on a set of benchmark instances from the literature and with respect to the current state-of-the-art. Finally, in Section 8, we discuss the numerical stability of the BMEP and the impact it has on the statistical consistency of the optimal solution.

2. Notation and properties of the topological distances

We introduce here some notation, definitions, and basic results from the literature that will prove useful throughout the article. We start by recalling that an UBT T with n leaves has $2n-2$ vertices, n of which are leaves and $n-2$ are internal vertices, and by $2n-3$ edges, n of which are external (i.e., connecting leaves) and $n-3$ are internal (i.e., connecting internal vertices). Indeed, denoted $E_e(T)$, $E_i(T)$, $V_e(T)$ and $V_i(T)$ as the sets of external edges, internal edges, external vertices and internal vertices of T, respectively, by classical results from graph theory we have that [61]

$$|E_i(T)| + |E_e(T)| = |V_i(T)| + |V_e(T)| - 1. \tag{2}$$

Moreover, because internal vertices have degree three, we have that

$$2|E_i(T)| + 2|E_e(T)| = 3|V_i(T)| + |V_e(T)|. \tag{3}$$

By combining (2) and (3), it follows that $|V_i(T)| = (n-2)$ and $|E_i(T)| = (n-3)$. We denote $V(T) = V_e(T) \cup V_i(T)$ and $E(T) = E_e(T) \cup E_i(T)$ as the set of vertices and the set of edges of T respectively. Moreover, for a given subset of taxa $S \subseteq \Gamma$, we call a phylogeny T of S a *sub-phylogeny* of Γ and write T_S instead of T to indicate that we involve just the taxa in S, i.e., T_S is defined as a pair constituted by an UBT with $|S|$ leaves and an assignment of the taxa in S to the leaves of the UBT. We denote $V(T_S)$ and $E(T_S)$ as the sets of internal vertices and edges of T_S, respectively. Moreover, whenever the context will be sufficiently clear for the sake of notation we will drop the index S from T. For example, we shall write just T whenever $S = \Gamma$.

Given two integers α and β, with $\alpha < \beta$, we denote $[\alpha, \beta]$ as the discrete interval constituted by the integers included between α and β, and similarly to Parker and Ram [57], we define a *sequence* as a collection of nonnegative real values such as $\mathbf{s} = [s_1, s_2, \dots, s_m]$, $s_j \in \mathbb{R}$. Repetition of values in the sequence is permitted, as the reals s_j do not need to be pairwise distinct. We denote by $\uparrow(\mathbf{s})$ and $\downarrow(\mathbf{s})$ the sequences obtained from $\mathbf{s}$ by sorting its entries in non-decreasing order and non-increasing order, respectively, and we shall extend this notation to any vector in $\mathbb{R}^m$.

Given a phylogeny T of a set Γ of n taxa and a taxon $i \in \Gamma$, we denote by Γ_i the set $\Gamma \setminus \{i\}$ and we define the *path-length sequence* $\tau_i = [\tau_{ij} \in [2, n-1] : j \in \Gamma_i]$ as the sequence whose generic entry encodes the length of the unique path in T connecting taxon i to each taxon $j \in \Gamma_i$. To also make the encoding τ_i unique we sum up all edges with a constant weight of one on paths between i and $j \in \Gamma \setminus \{i\}$. For example, consider the phylogeny showed in Figure 1. Then, $\tau_1 = [2, 3, 4, 4]$ and $\tau_4 = [4, 4, 3, 2]$. Note that the path-length sequence τ_i describes the UBT underlying T from the "perspective" of taxon $i \in \Gamma$, hence, with an abuse of nomenclature, we will say that τ_i describes the phylogeny T *rooted in taxon* i. Fixed a taxon $i \in \Gamma$, we denote Θ_i as the set of the path-length sequences τ_i encoding the phylogenies of Γ rooted in i.

The following extension of the definition of a path-length sequence proves particularly useful to obtain a mathematical programming formulation for the BMEP. Specifically, we define the *Path-Length Matrix* (PLM) τ associated to a phylogeny T of Γ as a $n \times n$ symmetric integer matrix having as generic entry τ_{ij}, if $i \neq j$, and 0 otherwise. For example, by assuming that taxa are ordered according to their labels, the following matrix

$$\tau = \begin{pmatrix} 0 & 2 & 3 & 4 & 4 \\ 2 & 0 & 3 & 4 & 4 \\ 3 & 3 & 0 & 3 & 3 \\ 4 & 4 & 3 & 0 & 2 \\ 4 & 4 & 3 & 2 & 0 \end{pmatrix}$$

is the PLM is associated to the phylogeny shown in Figure 1. Observe that, apart from the diagonal entries, each row (or each column) of τ is a path-length sequence of the considered phylogeny, rooted in a taxon $i \in \Gamma$. In particular, the first row refers to τ_1, the second row to τ_2, and so on. Hence, a PLM of an UBT with n leaves can be seen as a *collection* of path-length sequences of rooted phylogenies, i.e., $\tau = [\tau_i, i \in \Gamma]$. Thus, the BMEP can be seen as a *Network Design Problem* [45, 59] defined over PLMs encoding UBTs with n leaves or, alternatively, over collections of path-length sequences of phylogenies rooted in the taxa of Γ. We denote Θ as the set of PLMs τ encoding UBTs with n leaves and, for a fixed $\tau \in \Theta$, we define $2^{-\tau}$ as the matrix obtained from τ by setting its non-diagonal entries to $2^{-\tau_{ij}}$. Finally, we define $2^{-\Theta} = \{2^{-\tau} : \tau \in \Theta\}$.

Characterizing Θ is a necessary task to obtain a mathematical programming formulation for the BMEP. Because PLMs of UBTs are collections of path-length sequences, this task involves characterizing as well the sets Θ_i, for all $i \in \Gamma$. The next two sections address both issues.

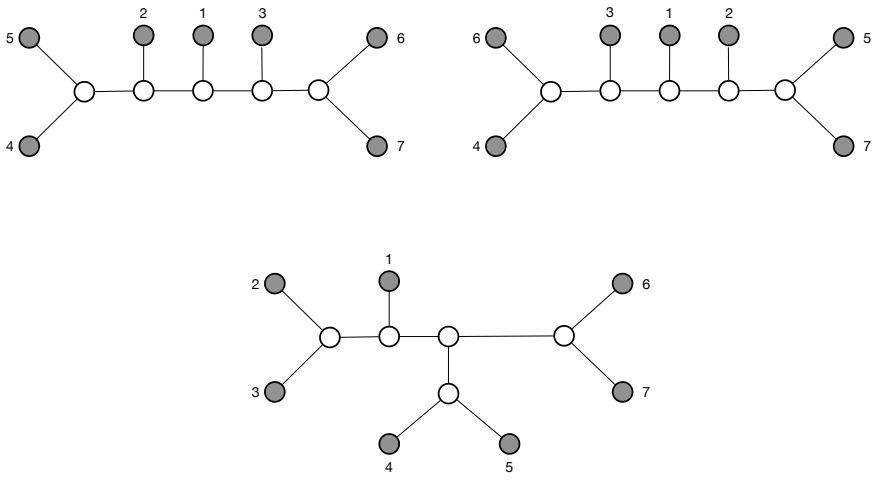


Figure 2: An example of the three distinct phylogenies that can be reconstructed from the path-length sequence $\tau_1 = [3, 3, 4, 4, 4, 4]$.

2.1. Characterizing Θ_i

As observed in Catanzaro et al. [14] and [17], a characterization of the set Θ_i, for a fixed $i \in \Gamma$, can be achieved by exploiting the similarities between phylogenies and *Huffman trees* [57]. Specifically, Huffman trees are rooted binary trees used in coding theory to represent symbols belonging to a given alphabet Ψ. The leaves of a Huffman tree correspond to the symbols in Ψ and the tree itself can be described by means of a path-length sequence $\rho = [\rho_j : j \in \Psi]$ whose generic entry ρ_j represents the topological distance of the shortest path from the root of the tree to the symbol $j \in \Psi$. In this context, the following well-known necessary and sufficient condition relates rooted binary trees and path-length sequences:

Proposition 1. *(Kraft's equality [57]) Consider a set Ψ of n symbols. Then, $\rho = [\rho_j, \rho_2, \ldots, \rho_n]$ is the sequence of topological distances of a rooted binary tree having Ψ as leafset if and only if*

$$\sum_{j \in \Psi} 2^{-\rho_j} = 1. \tag{4}$$

Proposition 1 can be adapted to provide a characterization of the set Θ_i. Specifically, consider a phylogeny T of Γ and a taxon $i \in \Gamma$. Denote $\hat{i}$ as the "father" of taxon i, i.e., as the only internal vertex adjacent to i in T. For example, by referring to the phylogeny shown in Figure 1, if $i = 1$ then $\hat{i} = 6$. We observe that if we disregard the edge $(i, \hat{i})$ then the remaining tree can be seen as a Huffman tree rooted in $\hat{i}$ and coding the symbols in $\Psi = \Gamma_i$. Thus, Proposition 1 can be restated as follows:

Proposition 2. *(Kraft's equality for phylogenies [14]) Let Γ be a set of n taxa, and let $i \in \Gamma$. A sequence of positive integers $\tau_i = [\tau_{ij} \in [2, n-1] : j \in \Gamma_i]$ is a path-length sequence of a phylogeny T of Γ if and only if the entries of τ_i satisfy the following condition:*

$$\sum_{j \in \Gamma_i} 2^{-\tau_{ij}} = \frac{1}{2}. \tag{5}$$

Kraft's equality for phylogenies is a very powerful condition: it allows to capture in a closed form both the connectivity of the rooted phylogeny and the respect of the degree constraint on its internal vertices. A phylogeny corresponding to a given path-length sequence $\tau_i = [\tau_{ij} \in [2, n-1] : j \in \Gamma_i]$, for some $i \in \Gamma$, can be easily reconstructed, e.g., by sorting τ_i in ascending order and by drawing the path-lengths from i to all of the remaining taxa in Γ_i. However, it is worth noting that no bijective relation between the set Θ_i and either the set of phylogenies or the set of the UBTs can be defined, mainly because a path-length sequence may correspond to multiple distinct phylogenies. For example, Figure 2 shows that there exists three possible phylogenies that can be reconstructed from the path-length sequence $\tau_1 = [3, 3, 4, 4, 4, 4]$.

2.2. Characterizing Θ

Characterizing the set Θ is still an open problem. However, we know a number of necessary conditions that any PLM τ must satisfy in order to belong to Θ. For example, because phylogenies are (non-oriented) acyclic graphs, the entries of τ must satisfy the following conditions:

$$\tau_{ij} \in [2, n-1] \qquad \forall\, i,j \in \Gamma : i \neq j \tag{6}$$
$$\tau_{ii} = 0 \qquad \forall\, i \in \Gamma \tag{7}$$
$$\tau_{ij} = \tau_{ji} \qquad \forall\, i,j \in \Gamma : i < j \tag{8}$$
$$\tau_{ij} + \tau_{jk} - \tau_{ik} \geq 2 \qquad \forall\, i,j,k \in \Gamma : i \neq j \neq k. \tag{9}$$

Specifically, the *integrality condition* (6) imposes that the value of each path-length of a phylogeny $T \in \Theta$ ranges between 2 (due to the degree constraint on the internal vertices of an UBT) and $n-1$. The *diagonal condition* (7) trivially imposes that the path-length between a taxon $i \in \Gamma$ and itself is 0. The *symmetry condition* imposes that the length of the unique path in T from a taxon $i \in \Gamma$ to a taxon $j \in \Gamma$ must be equal to the length of the (same) path from taxon j to taxon i. Finally, condition (9) imposes the satisfaction of the *triangle inequality*. Now, because the tree encoded by any PLM $\tau \in \Theta$ must be an UBT, the rows of τ must satisfy Kraft's equalities for phylogenies, i.e.,

$$\sum_{j \in \Gamma_i} 2^{-\tau_{ij}} = \frac{1}{2} \quad \forall\, i \in \Gamma. \tag{10}$$

Note that, up to a factor 2, each entry $2^{-\tau_{ij}}$ in (10) can be interpreted as a probability [57]. This insight has been exploited in Catanzaro et al. [13, 14, 17] to identify a further condition that the entries of any PLM τ must satisfy in order to belong to Θ, namely

$$\sum_{i \in \Gamma} \sum_{j \in \Gamma_i} \tau_{ij} 2^{-\tau_{ij}} = (2n-3). \tag{11}$$

Equation (11), usually referred to as the *phylogenetic manifold*, relates the path-lengths τ_{ij} to their *path-length densities* $2^{-\tau_{ij}}$ [57], and encodes the geometric locus of the matrices $2^{-\tau}$ having constant cross entropy (see Catanzaro et al. [13] for more information).

It is worth noting that because any PLM $\tau \in \Theta$ must encode a tree, the triangle inequalities (9) can be further generalized by means of Buneman's *additive property* [8, 29, 66], which states that a given graph is a tree if and only if any quartet of its vertices, say i, j, p and q, satisfies exactly one of the following conditions:

$$\begin{aligned} \tau_{ij} + \tau_{pq} + 2 \leq \tau_{iq} + \tau_{jp} = \tau_{ip} + \tau_{jq} \\ \tau_{iq} + \tau_{jp} + 2 \leq \tau_{ij} + \tau_{pq} = \tau_{ip} + \tau_{jq} \\ \tau_{ip} + \tau_{jq} + 2 \leq \tau_{ij} + \tau_{pq} = \tau_{iq} + \tau_{jp}. \end{aligned} \tag{12}$$

It is easy to see that (9) is a restriction of (12) when i, j, p and q belong to Γ and any two indices are equal (e.g., for $p = q$ the first line in (12) is equivalent to (9)). Now, conditions (6)-(12) are independent and necessary to characterize Θ [17]. If the assumption of having constant edge weights for the encoding of path-length sequences is dropped, then condition (12) is necessary and sufficient to characterize Θ if and only if one of the three inequalities in condition (12) is strict [4]. In constrast, little is known for the case in which such an assumption holds.

3. On the state-of-the-art exact solution algorithm for the BMEP

Formulating a discrete optimization problem in terms of integer (linear) programming implies identifying a proper space to model the involved variables. This task is notoriously delicate, as a wrong choice of the space may impact negatively on the solution times of a formulation. The space of path-lengths τ_{ij} looks a natural choice for the BMEP as it allows to capture the minimal necessary information to describe the problem; Kraft equalities (10) and the phylogenetic manifold (11), however, have a nonlinear expression in it. A possible attempt to overcome this issue was proposed by Forcey et al. [33], by means of the introduction of variables

$$x_{ij} = 2^{n-1-\tau_{ij}} \quad \forall\, i,j \in \Gamma, i \neq j. \tag{13}$$

In this space, indeed, Kraft equalities (10) become linear:

$$\sum_{j \in \Gamma_i} x_{ij} = 2^{n-2} \quad \forall\, i \in \Gamma.$$

The expression of the phylogenetic manifold, however, does not linearize and it seems hard to find a space in which both Kraft equalities (10) and the phylogenetic manifold (11) may look linear at the same time. Catanzaro et al.'s discretization approach [14] constitutes a possible way to derive a valid formulation of the BMEP. Specifically, denoted $\mathcal{L} = \{2, \dots, n-1\}$ as the set of values taken by the generic path-length τ_{ij}, Catanzaro et al. [14] proposed to model path-lengths by means of the following binary decision variables:

$$x_{ij}^{\ell} = \begin{cases} 1 & \text{if } \tau_{ij} = \ell \\ 0 & \text{otherwise} \end{cases} \quad \forall\, i, j \in \Gamma, i \neq j,\ \ell \in \mathcal{L}.$$

Provided the satisfaction of the following *convexity constraint*

$$\sum_{\ell \in \mathcal{L}} x_{ij}^{\ell} = 1 \quad \forall\, i, j \in \Gamma,\ i \neq j \tag{14}$$

ensuring the existence of exactly one path-length connecting each distinct pair of taxa $i, j \in \Gamma$ in any feasible solution to the problem, in this space the objective function (1) can be written as

$$z = \sum_{i \in \Gamma} \sum_{j \in \Gamma_i} d_{ij} \left(\sum_{\ell \in \mathcal{L}} 2^{-\ell} x_{ij}^{\ell} \right)$$

and conditions (8), (10) and (11) become

$$x_{ij}^{\ell} = x_{ji}^{\ell} \qquad \forall\, i, j \in \Gamma, i \neq j, \forall\, \ell \in \mathcal{L} \tag{15}$$

$$\sum_{j \in \Gamma_i} \sum_{\ell \in \mathcal{L}} 2^{-\ell} x_{ij}^{\ell} = \frac{1}{2} \qquad \forall\, i \in \Gamma \tag{16}$$

$$\sum_{i \in \Gamma} \sum_{j \in \Gamma_i} \sum_{\ell \in \mathcal{L}} \ell 2^{-\ell} x_{ij}^{\ell} = (2n-3). \tag{17}$$

Buneman's conditions (12) restricted to Γ can be modeled by introducing the following binary decision variables

$$y_{ijqt} = \begin{cases} 1 & \text{if } \tau_{it} + \tau_{jq} \geq \tau_{iq} + \tau_{jt} \\ 0 & \text{otherwise} \end{cases} \quad \forall\, i, j, q, t \in \Gamma, i \neq j, q, t,\ j \neq q, t,\ q \neq t$$

and by imposing the following constraints:

$$\sum_{\ell \in \mathcal{L}} \ell (x_{ij}^{\ell} + x_{qt}^{\ell}) - \sum_{\ell \in \mathcal{L}} \ell (x_{iq}^{\ell} + x_{jt}^{\ell}) \leq (2n-2) y_{ijqt} \qquad \forall\, i, j, q, t \in \Gamma,\ i \neq j, q, t,\ j \neq q, t,\ q \neq t \tag{18}$$

$$\sum_{\ell \in \mathcal{L}} \ell (x_{ij}^{\ell} + x_{qt}^{\ell}) - \sum_{\ell \in \mathcal{L}} \ell (x_{it}^{\ell} + x_{jq}^{\ell}) \leq (2n-2)(1 - y_{ijqt}) \qquad \forall\, i, j, q, t \in \Gamma,\ i \neq j, q, t,\ j \neq q, t,\ q \neq t. \tag{19}$$

It is worth noting that the above set of constraints (14)-(19) does not lead per se to a valid formulation for the BMEP, mostly because we do not know yet whether the above set of constraints implies the acyclicity of the combinatorial structure identified by x_{ij} variables. To ensure the acyclicity of any feasible solution to the problem, Catanzaro et al. [14] introduced a further set of variables, called *edge variables*. In particular, the authors first extended the set of values $\mathcal{L}$ by including paths of length one (i.e., edges):

$$\mathcal{L} = \{1, 2, \dots, n-1\}.$$

Then, the authors denoted $I = \{n+1, n+2, \dots, 2n-2\}$ as a set of $n-2$ vertices representing the internal vertices of an UBT and extended y_{ijqt} variables to the set of indices I by

$$y_{ijqt} = \begin{cases} 1 & \text{if } \tau_{it} + \tau_{jq} \geq \tau_{iq} + \tau_{jt} \\ 0 & \text{otherwise} \end{cases} \quad \forall\, i, j, q, t \in \Gamma \cup I,\ i \neq j, q, t,\ j \neq q, t,\ q \neq t.$$

Subsequently, the authors extended the *path variables* x_{ij}^{ℓ} to x_{ij}^{ℓ}, $i, j \in \Gamma \cup I$, $i \neq j$, $\ell \geq 2$ and denoted the edge variables as x_{ij}^{ℓ} with $i, j \in \Gamma \cup I$, $i \neq j$, $\ell = 1$. Finally, the authors linked the *path variables*, i.e., x_{ij}^{ℓ}, $\ell \geq 2$, to the edge variables x_{ij}^{1} by means of the following constraints:

$$x_{ij}^{\ell} + 1 \geq x_{ik}^{(\ell-1)} + x_{kj}^{1} \qquad \forall i, j \in \Gamma, i \neq j, k \in I, \ell \in \mathcal{L} \setminus \{1, n-1\},$$
$$x_{ij}^{\ell} + x_{ij}^{(\ell-2)} + 1 \geq x_{ik}^{(\ell-1)} + x_{kj}^{1} \qquad \forall i, j, k \in \Gamma \cup I, i \neq j, k,\ j \neq k, \ell \in \mathcal{L} \setminus \{1, 2, n-1\}$$

by giving rise to the following valid integer linear formulation for the BMEP:

Formulation 1.

$$\min \quad z = \sum_{i \in \Gamma} \sum_{j \in \Gamma_i} d_{ij} \left(\sum_{\ell \in \mathcal{L} \setminus \{1\}} 2^{-\ell} x_{ij}^{\ell} \right) \tag{20}$$

s.t.

$$\sum_{\ell \in \mathcal{L} \setminus \{1\}} x_{ij}^{\ell} = 1 \qquad \forall i, j \in \Gamma \cup I,\ i \neq j \tag{21}$$

$$x_{ij}^{\ell} = x_{ji}^{\ell} \qquad \forall i, j \in \Gamma, i \neq j, \forall \ell \in \mathcal{L} \setminus \{1\} \tag{22}$$

$$\sum_{j \in \Gamma_i} \sum_{\ell \in \mathcal{L} \setminus \{1\}} 2^{-\ell} x_{ij}^{\ell} = \frac{1}{2} \qquad \forall i \in \Gamma \tag{23}$$

$$\sum_{i \in \Gamma} \sum_{j \in \Gamma_i} \sum_{\ell \in \mathcal{L} \setminus \{1\}} \ell 2^{-\ell} x_{ij}^{\ell} = (2n-3) \tag{24}$$

$$\sum_{\ell \in \mathcal{L} \setminus \{1\}} \ell (x_{ij}^{\ell} + x_{qt}^{\ell}) - \sum_{\ell \in \mathcal{L} \setminus \{1\}} \ell (x_{iq}^{\ell} + x_{jt}^{\ell}) \leq (2n-2) y_{ijqt} \qquad \forall i, j, q, t \in \Gamma \cup I, i \neq j, q, t,\ j \neq q, t,\ q \neq t \tag{25}$$

$$\sum_{\ell \in \mathcal{L} \setminus \{1\}} \ell (x_{ij}^{\ell} + x_{qt}^{\ell}) - \sum_{\ell \in \mathcal{L} \setminus \{1\}} \ell (x_{it}^{\ell} + x_{jq}^{\ell}) \leq (2n-2)(1 - y_{ijqt}) \qquad \forall i, j, q, t \in \Gamma \cup I, i \neq j, q, t,\ j \neq q, t,\ q \neq t \tag{26}$$

$$x_{ij}^{1} = 0 \qquad \forall i, j \in \Gamma, i \neq j \tag{27}$$

$$\sum_{\substack{i,j \in \Gamma \cup I \\ i \neq j}} x_{ij}^{1} = (2n-3) \tag{28}$$

$$\sum_{j \in I} x_{ij}^{1} = 1 \qquad \forall i \in \Gamma \tag{29}$$

$$\sum_{\substack{j \in \Gamma \cup I \\ i \neq j}} x_{ij}^{1} = 3 \qquad \forall i \in I \tag{30}$$

$$x_{ij}^{1} + x_{ik}^{1} + x_{kj}^{1} \leq 2 \qquad \forall i, j, k \in I, i \neq j, k,\ j \neq k \tag{31}$$

$$x_{ij}^{\ell} + 1 \geq x_{ik}^{(\ell-1)} + x_{kj}^{1} \qquad \forall i, j \in \Gamma, i \neq j, k \in I, \ell \in \mathcal{L} \setminus \{1, n-1\} \tag{32}$$

$$x_{ij}^{\ell} + x_{ij}^{(\ell-2)} + 1 \geq x_{ik}^{(\ell-1)} + x_{kj}^{1} \qquad \forall i, j, k \in \Gamma \cup I, i \neq j, k,\ j \neq k, \ell \in \mathcal{L} \setminus \{1, 2, n-1\} \tag{33}$$

$$x_{ij}^{\ell} \in \{0, 1\} \qquad \forall i, j \in \Gamma \cup I, \ell \in \mathcal{L} \tag{34}$$

$$y_{ijqt} \in \{0, 1\} \qquad \forall i, j, q, t \in \Gamma \cup I, i \neq j, q, t,\ j \neq q, t,\ q \neq t. \tag{35}$$

Constraints (27)-(33) describe the structure of a phylogeny. Specifically, constraint (27) imposes that no edge exists between taxa in Γ. Constraint (28) imposes that exactly $(2n-3)$ edges be present in a phylogeny. Constraints (29) and (30) impose the degree constraint on the leaves and internal vertices of a phylogeny. Constraints (31) prevent triangles. Finally, constraints (32)-(33) relate edge variables to path variables.

Formulation 1 is valid for the BMEP; however, the inclusion of the edge variables and, above all, of the y variables, slows down dramatically its resolution. Catanzaro et al. [14] observed that this situation persists even when replacing y variables with the more conventional subtour elimination constraints. To speed up the resolution of Formulation 1, the authors proposed to remove y variables and to use specific branching rules on the remaining x variables so as to ensure the potential enumeration of all of the possible phylogenies of Γ. These rules are recursive in nature and based on the *Stepwise Addition Strategy* (SAS), first introduced by Hendy and Penny [43] in 1982. This strategy allows to enumerate all of the possible phylogenies of Γ by means of the idea of *insertion* of a taxon on a sub-phylogeny. Specifically, given

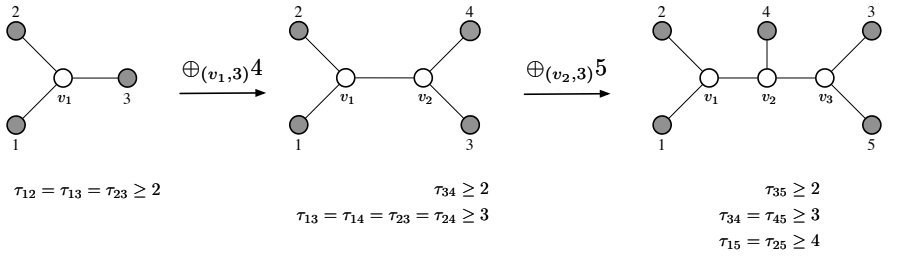


Figure 3: An example of two consecutive insertions on a sub-phylogeny of three taxa. The inequalities indicate restrictions on path-lengths between pairs of taxa for further insertions.

a subset of taxa $S \subset \Gamma$, a sub-phylogeny T_S of Γ, a taxon $t \in \Gamma \setminus S$, an internal vertex $v \in I \setminus V_i(T_S)$, and an edge $(a, b) \in E(T_S)$, an *insertion* of taxon t into the edge (a, b) of T_S is the sub-phylogeny $T_{S\cup\{t\}}$ defined by

$$E\left(T_{S\cup\{t\}}\right) = E\left(T_S\right) \setminus \{(a, b)\} \cup \{(a, v), (v, b), (v, t)\}\,.$$

For the sake of notation, we denote the above insertion operation as

$$T_{S\cup\{t\}} = T_S \oplus_{(a,b)} t.$$

An example of two consecutive insertions on a sub-phylogeny of three taxa is shown in Figure 3. Taxon 4 is initially inserted into edge $(v_1, 3)$ of the given sub-phylogeny (i) by introducing a new internal vertex v_2; (ii) by removing the edge $(v_1, 3)$; and (iii) by adding edges (v_1, v_2), $(v_2, 3)$ and $(v_2, 4)$. Subsequently, taxon 5 is inserted into the new sub-phylogeny of four taxa by adding internal vertex v_3; by removing edge $(v_2, 3)$; and by adding edges (v_2, v_3), $(v_3, 3)$, $(v_3, 5)$.

The opposite of an insertion of a taxon t on a sub-phylogeny $T_{S\cup\{t\}}$ is referred to as a *removal* and, for an edge $(a, b) \in E(T_{S\cup\{t\}})$ and an internal vertex $v \in I \setminus V_i(T_{S\cup\{t\}})$, it is defined by

$$E\left(T_S\right) = E\left(T_{S\cup\{t\}}\right) \setminus \{(a, v), (v, b), (v, t)\} \cup \{(a, b)\}\,.$$

Roughly speaking, a removal of taxon t can be obtained by deleting the last three edges from $E(T_{S\cup\{t\}})$ and by restoring the edge (a, b). The recursive application of the above definitions of an insertion and a removal of a taxon on a sub-phylogeny of Γ allows to enumerate the phylogenies of Γ, e.g., by means of Algorithm 1. A graphical example of its execution is shown in Figure 4. Note that, fixed a pair of distinct taxa $i, j \in \Gamma$, a subset $S \subset \Gamma$, and a sub-phylogeny T_S of Γ, the net result of an insertion of taxon $t \notin S$ on some edge of T_S is the exclusion of specific values for the path-lengths τ_{ij}. Specifically, denoted $P_{ij}(T_S)$ as the set of the possible length values that can be taken by the unique path p_{ij} in a sub-phylogeny derived from T_S by a series of insertions, it holds that

$$P_{ij}(T_S) = \begin{cases} \left\{\ell \in \mathcal{L} \;:\; \tau_{ij} \leq \ell \leq \tau_{ij} + |\Gamma \setminus S|\right\} & \text{if } i, j \in S; \\ \left\{\ell \in \mathcal{L} \;:\; 2 \leq \ell \leq \max\limits_{s\in S} \tau_{is} + |\Gamma \setminus S|\right\} & \text{if } i \in S, j \in \Gamma \setminus S; \\ \left\{\ell \in \mathcal{L} \;:\; 2 \leq \ell \leq \max\limits_{s,t\in S} \tau_{st} + |\Gamma \setminus S|\right\} & \text{if } i, j \in \Gamma \setminus S. \end{cases} \tag{36}$$

Algorithm 1: Enumerate - Stepwise Addition Strategy

Input: An ordered set of taxa $\Gamma = \{1, \ldots, n\}$; a subset $S \subset \Gamma$; a sub-phylogeny T_S of Γ

```
if |S| = |Γ| then
    return;
t ← |S| + 1;
for each edge e ∈ E(T_S) do
    Enumerate(Γ, S, T_S ⊕_e t);
```

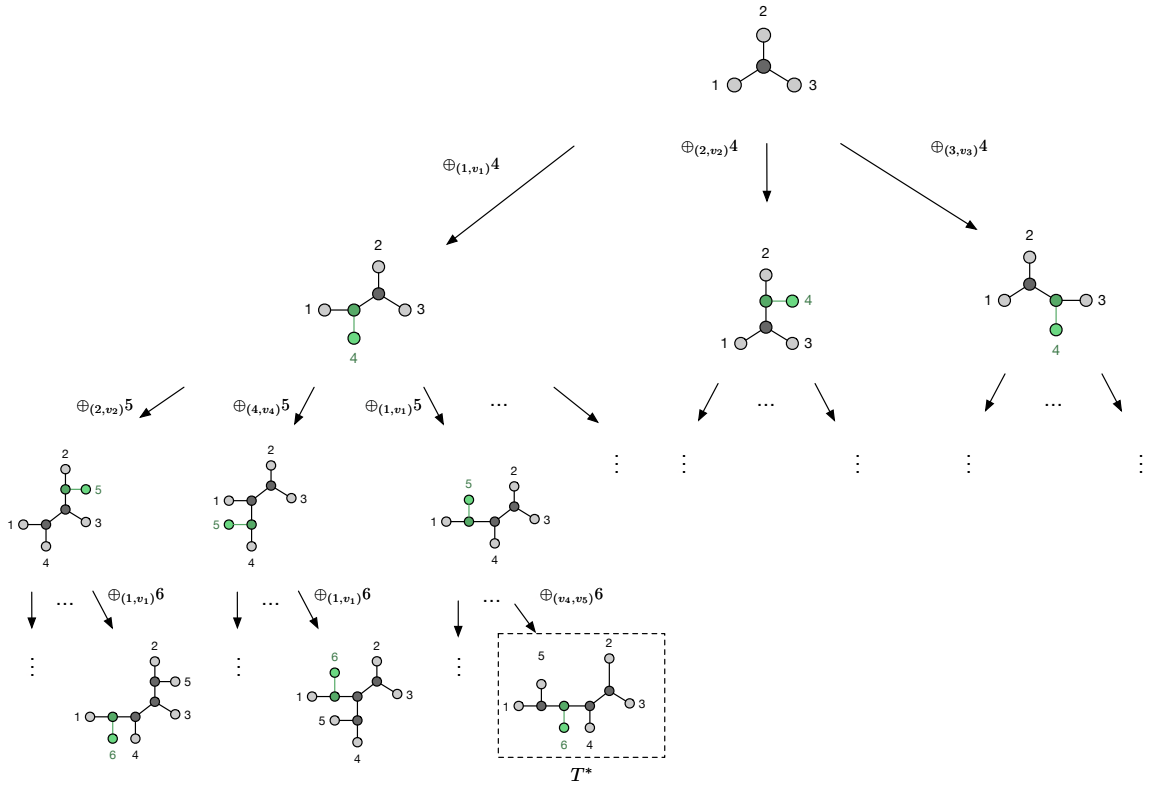


Figure 4: An example of the enumeration tree of a SAS for a set Γ of six taxa.

This fact implies that, following an insertion of taxon t on T_S and fixed a pair of distinct taxa i and j in Γ, all of the x_{ij}^{ℓ} variables whose ℓ indices fall out of the index sets $P_{ij}(T_S)$ can be set to zero, by giving rise to the above mentioned SAS-based branching rules. For example, by referring to the insertion of taxon 5 in Figure 3, we have that $x_{13}^3 = x_{23}^3 = x_{34}^2 = x_{15}^2 = x_{15}^3 = x_{25}^2 = x_{25}^3 = x_{45}^2 = 0$ and $x_{12}^{n-2} = x_{14}^{n-1} = x_{35}^{n-1} = x_{35}^{n-2} = x_{45}^{n-1} = 0$.

4. Lower bounds on the optimal solution to the BMEP

Identifying tight bounds on the optimal solution to a given discrete optimization problem is central to develop effective exact solution algorithms of practical use. In this section, we report on and compare the lower bounds on the optimal solution to the BMEP that are currently known in the literature and we present new ones. The earliest lower bounds on the optimal solution to the BMEP have been presented by Pardi [53] in 2009. Given a subset of taxa $S = \{1, 2, \dots, t-1\} \subset \Gamma$, a sub-phylogeny T_S of Γ, and a taxon $t \in \Gamma \setminus S$, the author first defined the quantities

$$\lambda_t^{ij} = \frac{1}{2}\left(d_{it} + d_{jt} - d_{ij}\right) \qquad \forall\, i, j \in S, i < j$$

and the vector $\lambda(t)$ having the quantities λ_t^{ij}, $i, j \in S$, $i < j$, as generic entries, sorted in non-decreasing order. Then, denoted $\lambda(t)_k$ as the k-th entry of $\lambda(t)$, Pardi showed that

Proposition 3.

$$L\left(T_S \oplus_{(a,b)} t\right) - L\left(T_S\right) \geq \sum_{k=1}^{t-3} \frac{1}{2^k}\lambda(t)_k + \frac{1}{2^{t-3}}\lambda(t)_{(t-2)} \tag{37}$$

and

$$L\left(T^*\right) - L\left(T_S\right) \geq \sum_{t \notin S} \left(\sum_{k=1}^{t-3} \frac{1}{2^k} \lambda(t)_k + \frac{1}{2^{t-3}} \lambda(t)_{(t-2)} \right). \tag{38}$$

Pardi's bound (38) [53] is very fast to compute and, at present, it constitutes the best non-mathematical programming based lower bound known for the problem (see, Tables 2 and 3 later in this section for more information).

A second set of lower bounds on the optimal solution to the BMEP was introduced by Catanzaro et al. [13]. The authors proved that the length function (1) is invariant under a specific rescaling $\hat{\mathbf{D}}$ of the distance matrix $\mathbf{D}$ up to a non-negative constant K. By introducing the probability distributions

$$p_i = \left\{ 2^{1-\tau_{ij}} \ : \ j \in \Gamma_i \right\} \quad \text{and} \quad q_i = \left\{ 2^{-\hat{d}_{ij}} \ : \ j \in \Gamma_i \right\}$$

and their *Kullback-Leibler divergence*

$$D_{KL}(p_i || q_i) = \sum_{j \in \Gamma_i} \left(-p_{ij} \log_2 \left(q_{ij} \right) + p_{ij} \log_2 \left(p_{ij} \right) \right),$$

the authors showed that the length function (1) can be rewritten as

$$L(T) = \frac{1}{2} \sum_{i \in \Gamma} \left(D_{KL} \left(p_i || q_i \right) \right) + \frac{3n-6}{2} + K$$

thereby transforming the BMEP into a cross-entropy minimization problem. Based on this reformulation of the BMEP length function, the authors introduced a family of lower bounds on the optimal solution to the BMEP, states as follows:

Proposition 4. *Let $\hat{\mathbf{D}}$ denote a rescaled form of an input distance matrix $\mathbf{D} \in \mathbb{R}^{n \times n}$ such that the matrix $(2^{-\hat{d}_{ij}})_{i,j \in \Gamma}$ is doubly stochastic, and let α denote a fixed scalar in $[0, 1]$. Then, there exists a non-negative constant K (with $K = 0$ being the optimal choice for $\hat{\mathbf{D}} = \mathbf{D}$) such that*

$$L\left(T^*\right) \geq \frac{1}{2} \sum_{i \in \Gamma} \min_{\tau_i \in \Theta_i} \sum_{j \in \Gamma_i} \frac{\hat{d}_{ij} - \alpha(\tau_{ij} - 1)}{2^{\tau_{ij}-1}} + \alpha \frac{3n-6}{2} + K. \tag{39}$$

This lower bound can be calculated in $\mathcal{O}(n^2 \log n)$ but, as shown in Tables 2 and 3, it proves to be looser than Pardi's one [53].

An alternative approach to obtain a lower bound on the optimal solution to the BMEP consists of using mathematical programming. Specifically, consider the problem of minimizing the BMEP length function subject to just Kraft equalities (10), i.e.,

Formulation 2.

$$\begin{aligned}
\min \quad & z = \sum_{i \in \Gamma} \sum_{j \in \Gamma_i} d_{ij} 2^{-\tau_{ij}} & \\
s.t. \quad & \sum_{j \in \Gamma_i} 2^{-\tau_{ij}} = \frac{1}{2} & \forall\, i \in \Gamma \\
& \tau_{ij} \in [2, n-1] & \forall\, i, j \in \Gamma, i \neq j.
\end{aligned}$$

Formulation 2 can be reformulated in linear fashion by eliminating from Formulation 1 all constraints but Kraft's, the convexity, and the integrality ones. The optimal solution to Formulation 2 coincides with the right-hand side of (39) when both $\alpha = 0$ and $\hat{\mathbf{D}} = \mathbf{D}$, and from a combinatorial perspective, can be interpreted as a forest of caterpillar UBTs each rooted in taxon $i \in \Gamma$. In particular, the following proposition holds:

Proposition 5. *Let $\mathbf{d}_i$ denote the i-th row of $\mathbf{D}$ and let τ_i be a path-length sequence in Θ_i. Then,*

(i) if τ_i is sorted in non-decreasing order then $\sum_{k=1}^{n} d_{ik} \cdot 2^{-\tau_{ik}} \geq \sum_{k=1}^{n} \uparrow (d_i)_k \cdot 2^{-\tau_{ik}}$;

(ii) $\tau_i' = [2, 3, \dots, (n-1), (n-1)] = \arg\min \left\{ \sum_{k=1}^{n} \uparrow (d_i)_k \cdot 2^{-\tau_{ik}} \ : \ \tau_i \in \Theta_i \right\}$.

Proof.

Part i. Assume that $\mathbf{d}_i$ is not sorted in non-decreasing order, otherwise the statement trivially follows. Then, there exist two indices, say $p, q \in [1, n]$, such that $p < q$ and $d_{ip} > d_{iq}$. As $2^{-\tau_{ip}} \geq 2^{-\tau_{iq}}$, then $d_{ip}2^{-\tau_{ip}} + d_{iq}2^{-\tau_{iq}} \geq d_{iq}2^{-\tau_{ip}} + d_{ip}2^{-\tau_{iq}}$. Hence, $\sum_{k=1}^{n} d_{ik}2^{-\tau_{ik}}$ gets smaller if d_{ip} and d_{iq} are swapped in d_i. This argument can be iterated until the entries of d_i are sorted in non-decreasing way.

Part ii. Let $\tau_i' = [2, 3, \dots, (n-1), (n-1)]$ denote a path-length sequence encoding an UBT with n leaves and rooted in i. Observe that

$$\sum_{k=1}^{l-1} 2^{-\tau'_{ik}} \geq \sum_{k=1}^{l-1} 2^{-\tau_{ik}} \quad \forall\, l = 2, \dots, n \ \forall\, \tau_i \in \Theta_i. \tag{40}$$

Without loss of generality assume that d_i is already sorted in non-decreasing way. Then

$$\begin{aligned}
\sum_{k=1}^{n} d_{ik} \cdot 2^{-\tau_{ik}} &= d_{i1} \cdot \sum_{k=1}^{n} 2^{-\tau_{ik}} + \sum_{l=2}^{n} \left(\left(d_l - d_{l-1}\right) \cdot \sum_{k=l}^{n} 2^{-\tau_{ik}} \right) \\
&\overset{(5)}{=} \frac{d_{i1}}{2} + \sum_{l=2}^{n} \left(\left(d_l - d_{l-1}\right) \cdot \left(\frac{1}{2} - \sum_{k=1}^{l-1} 2^{-\tau_{ik}} \right) \right) \\
&\overset{(40)}{\geq} \frac{d_{i1}}{2} + \sum_{l=2}^{n} \left(\left(d_l - d_{l-1}\right) \cdot \left(\frac{1}{2} - \sum_{k=1}^{l-1} 2^{-\tau'_{ik}} \right) \right) \\
&\overset{(5)}{=} d_{i1} \cdot \sum_{k=1}^{n} 2^{-\tau'_{ik}} + \sum_{l=2}^{n} \left(\left(d_l - d_{l-1}\right) \cdot \sum_{k=l}^{n} 2^{-\tau'_{ik}} \right) \\
&= \sum_{k=1}^{n} d_{ik} \cdot 2^{-\tau'_{ik}}.
\end{aligned}$$

Thus the statement follows.

□

An alternative lower bound for the optimal solution to the BMEP can be obtained by minimizing the length function on the phylogenetic manifold, i.e., by solving the problem

Formulation 3.

$$\begin{aligned}
\min \quad & z = \sum_{i \in \Gamma} \sum_{j \in \Gamma_i} d_{ij} 2^{-\tau_{ij}} \\
s.t. \quad & \sum_{i \in \Gamma} \sum_{j \in \Gamma_i} \tau_{ij} 2^{-\tau_{ij}} = 2n - 3 \\
& 2 \leq \tau_{ij} \leq n - 1 \qquad \forall\, i, j \in \Gamma, i \neq j.
\end{aligned}$$

Formulation 3 can be reformulated in linear fashion by relaxing the integrality constraints on τ_{ij} variables in Formulation 1 and by eliminating from Formulation 1 all constraints but the convexity and the phylogenetic manifold ones. Note that the optimal solution to Formulation (3) has a particular geometric interpretation. Specifically, let $\mathbf{O}$ denote a $n \times n$ matrix having $2^{-(n-1)}$ as a generic non-diagonal entry and 0 otherwise. Let $\mathbf{o}$ denote a $n(n-1)$-vector obtained from $\mathbf{O}$ by excluding the diagonal entries and by appending its rows one after the other. It is easy to realize that this vector identifies a point that is internal to the phylogenetic manifold. Let $\mathbf{d}$ denote the $n(n-1)$-vector obtained from $\mathbf{D}$

by excluding its diagonal entries and by appending its rows one after the other. Note that in general the point mapped by $\mathbf{d}$ can be internal, external or on the phylogenetic manifold itself. Finally, let $2^{-\tau}$ denote a $n(n-1)$-vector whose generic entry is $2^{-\tau_{ij}}$. Then, it is easy to see that solving Formulation (3) is equivalent to minimize the scalar product $\mathbf{d} \cdot 2^{-\tau}$ subject to having $2^{-\tau}$ mapping a point on the manifold. It is easy to see that this situation occurs when the angle between $\mathbf{d}$ and $2^{-\tau}$ is zero and that the point on the manifold for which this situation holds corresponds to the intersection between (the possible prolongation of) the segment $\overline{\mathbf{od}}$ and the phylogenetic manifold. In general, this intersection point is not integral; the BMEP, therefore, can be interpreted as the problem of finding a point $2^{-\tau}$ that maps an UBT and that is as close as possible to this intersection point. It is also worth noting that it is not possible to assess a priori the tightness of the lower bound provided by Formulation 2 with respect to the one provided by Formulation (3). In fact, we consider the following input distance matrices

$$\mathbf{D} = \begin{pmatrix} 0 & 0.49 & 0.95 & 1.39 & 1.1 & 1.14 & 0.81 \\ 0.49 & 0 & 0.81 & 1.18 & 1.64 & 1.85 & 0.39 \\ 0.95 & 0.81 & 0 & 1.55 & 3.26 & 3.07 & 1.48 \\ 1.39 & 1.18 & 1.55 & 0 & 3.39 & 4.06 & 1.43 \\ 1.1 & 1.64 & 3.26 & 3.39 & 0 & 0.26 & 1.41 \\ 1.14 & 1.85 & 3.07 & 4.06 & 0.26 & 0 & 2.03 \\ 0.81 & 0.39 & 1.48 & 1.43 & 1.41 & 2.03 & 0 \end{pmatrix} \text{ and } \mathbf{D}' = \begin{pmatrix} 0 & 3.45 & 3.45 & 3.21 & 3.45 & 3.1 & 3 \\ 3.45 & 0 & 2.19 & 1.88 & 1.76 & 2.42 & 2.12 \\ 3.45 & 2.19 & 0 & 0.29 & 1.88 & 1.76 & 2.42 \\ 3.21 & 1.88 & 0.29 & 0 & 1.56 & 1.47 & 2.19 \\ 3.45 & 1.76 & 1.88 & 1.56 & 0 & 0.93 & 0.71 \\ 3.1 & 2.42 & 1.76 & 1.47 & 0.93 & 0 & 0.73 \\ 3 & 2.12 & 2.42 & 2.19 & 0.71 & 0.73 & 0 \end{pmatrix}$$

and observe that the values of the optimal solutions to Formulations 2 and 3 are 3.272 and 3.357, respectively, when considering the distance matrix $\mathbf{D}$. In contrast, the values of the optimal solutions to Formulations 2 and 3 are 4.4 and 3.726, respectively, when considering $\mathbf{D}'$. A lower bound that is provably tighter than both Formulations 2 and 3 can be obtained, however, when merging both formulations and when including also the symmetry constraint (22), i.e.,

Formulation 4.

$$\min \quad z = \sum_{i \in \Gamma} \sum_{j \in \Gamma_i} d_{ij} \left(\sum_{\ell \in \mathcal{L} \setminus \{1\}} 2^{-\ell} x_{ij}^{\ell} \right) \tag{41}$$

$$s.t. \quad \sum_{\ell \in \mathcal{L} \setminus \{1\}} x_{ij}^{\ell} = 1 \qquad \forall i, j \in \Gamma, i \neq j \tag{42}$$

$$x_{ij}^{\ell} = x_{ji}^{\ell} \qquad \forall i, j \in \Gamma, i \neq j, \forall \ell \in \mathcal{L} \setminus \{1\} \tag{43}$$

$$\sum_{j \in \Gamma_i} \sum_{\ell \in \mathcal{L} \setminus \{1\}} 2^{-\ell} x_{ij}^{\ell} = \frac{1}{2} \qquad \forall i \in \Gamma \tag{44}$$

$$\sum_{i \in \Gamma} \sum_{j \in \Gamma_i} \sum_{\ell \in \mathcal{L} \setminus \{1\}} \ell 2^{-\ell} x_{ij}^{\ell} = (2n-3) \tag{45}$$

$$x_{ij}^{\ell} \geq 0 \qquad \forall i, j \in \Gamma, i \neq j, \forall \ell \in \mathcal{L} \setminus \{1\}. \tag{46}$$

We refer to the value of the optimal solution to Formulation 4 as the *Linear Programming Lower Bound* (LPLB). Tables 2 and 3 report on the values of the lower bounds obtained when considering Catanzaro et al.'s benchmark instances of the BMEP [16]. The tables show that Formulation 4 always provides the tightest lower bound on the optimal solution to the BMEP, immediately followed by Pardi's bound [53], the entropic bound (39), and finally the manifold-based bound provided by Formulation (3). The tightness of the LPLB may justify its use in an implicit enumeration solution algorithm for the BMEP. It is worth noting here that the burden required to compute each of these lower bounds on a specific node of the search tree is per se negligible. However, this burden becomes considerable in the context of an implicit enumeration search and this phenomenon is particularly true when considering Formulation 4: as shown in [14], the iterated calls to the simplex algorithm throughout the search tree makes the computation of the LPLB so time consuming that an implicit enumeration algorithm based on Pardi's bound [53] may prove to be even faster despite its poorer quality. One way to decrease the computational burden necessary to compute the LPLB consists of (i) reducing the size of Formulation 4 and (ii) reducing the number of times the LPLB is computed during the search tree. A dummy approach to implement point (i) consists of removing the symmetry constraints and declaring the variables x just for $i < j$. Finding an efficient way to implement point (ii) is instead a less trivial task. One option proposed in [14], consists of considering the Lagrangian relaxation of the reduced Formulation 4, i.e.,

Data set	Taxa	Optimum	Ineq. (38)	Ineq. (39)	Form. 3	Form. 4
Primates12	10	124.968	117.552	113.486	72.422	121.762
	11	332.834	308.570	283.810	184.726	324.211
	12	802.589	740.163	692.126	455.455	782.755
M17	10	105.134	100.189	90.639	72.929	104.033
	11	261.233	246.490	219.211	170.613	258.146
	12	541.632	508.402	456.433	349.652	535.065
	13	1,181.598	1,098.695	1,009.782	780.718	1,162.043
	14	2,408.066	2,236.155	2,050.144	1,545.662	2,368.216
	15	4,998.295	4,630.741	4,288.528	3,200.613	4,915.784
	16	10,225.560	9,371.332	8,752.214	6,455.794	10,051.994
	17	20,788.112	19,031.724	17,800.311	12,949.645	20,447.707
M18	10	190.331	164.735	166.916	145.096	186.288
	11	396.942	330.568	342.986	289.819	386.718
	12	805.937	667.791	686.222	567.978	784.056
	13	1,758.111	1,382.077	1,501.844	1,231.903	1,700.618
	14	3,599.678	2,763.052	3,008.283	2,431.544	3,469.282
	15	7,746.218	5,793.984	6,607.869	5,305.496	7,477.146
	16	15,973.016	11,754.719	13,721.437	11,093.488	15,411.761
	17	32,345.662	22,911.055	27,721.161	22,455.971	31,202.619
	18	66,029.112	45,790.686	56,264.712	44,556.665	63,683.315
SeedPlant25	10	91.458	77.449	84.284	77.494	86.549
	11	206.250	173.268	185.399	170.437	195.988
	12	427.816	356.940	391.828	360.248	407.022
	13	943.774	761.057	848.138	771.943	893.893
	14	1,929.219	1,553.194	1,749.409	1,591.185	1,831.570
	15	4,085.763	3,287.307	3,699.822	3,338.153	3,895.098
	16	8,353.457	6,334.814	7,446.740	6,598.428	7,872.086
	17	17,314.429	12,441.439	15,387.072	13,493.130	16,239.100
	18	36,156.527	25,344.199	31,811.347	27,666.698	33,654.096
	19	75,261.323	52,409.879	65,976.509	56,998.660	70,014.666
	20	164,507.262	109,511.370	139,432.212	119,602.162	149,517.567
M43	10	105.020	99.003	97.534	85.251	103.652
	11	209.282	196.591	189.736	164.251	204.268
	12	434.326	401.064	395.582	335.465	426.557
	13	895.424	823.990	812.050	680.809	879.878
	14	1,808.970	1,650.212	1,619.119	1,328.144	1,777.358
	15	3,965.234	3,569.271	3,517.331	2,852.487	3,904.670
	16	8,219.835	7,356.954	7,403.837	6,004.717	8,095.798
	17	16,798.759	14,985.303	15,134.213	12,336.357	16,493.312
	18	35,383.158	31,350.532	32,020.622	25,772.321	34,874.323
	19	71,769.903	63,521.426	65,160.787	52,611.255	70,743.386
	20	157,472.424	138,290.054	141,187.979	112,460.414	155,142.306
RbcL55	10	152.482	140.757	131.171	111.541	149.733
	11	328.645	303.619	286.910	243.473	323.349
	12	685.972	628.710	605.745	511.399	670.868
	13	1,502.870	1,335.970	1,328.099	1,104.865	1,467.890
	14	3,094.888	2,684.557	2,729.168	2,242.954	3,024.746
	15	6,448.259	5,558.499	5,697.917	4,678.343	6,289.820
	16	13,455.292	11,352.353	11,908.270	9,709.872	13,119.319
	17	27,804.246	22,685.482	24,499.350	19,643.587	26,987.288
	18	56,175.236	44,449.031	48,648.020	38,573.673	54,544.159
	19	114,623.865	88,318.788	98,119.885	76,318.718	111,223.724
	20	236,424.336	171,925.735	195,874.159	150,025.379	225,473.659

Table 2
Values of the lower bounds at the root node of the search tree obtained when considering Catanzaro et al.'s benchmark instances of the BMEP [16]. In order to compute inequality (39) we set $K = 0$ and we choose an optimal scalar α by complete enumeration and by assuming a machine epsilon equal to $\epsilon = 10^{-6}$.

Formulation 5. – ***Lagrangian Lower Bound (LagLB)***

$$\min \quad z_{lag}(T_S, \mu, \lambda) = \sum_{i,j\in\Gamma, i<j} \left(\sum_{\ell\in\mathcal{L}\setminus\{1\}} \left(2d_{ij} - \mu_i - \mu_j - 2\ell\lambda\right) 2^{-\ell} x_{ij}^{\ell} \right) + \sum_{i\in\Gamma} \frac{\mu_i}{2} + (2n-3)\lambda$$

Data set	Taxa	Optimum	Ineq. (38)	Ineq. (39)	Form. 3	Form. 4
	10	163.876	158.801	141.791	117.649	162.117
	11	350.425	338.429	302.397	248.494	345.178
	12	753.821	725.293	647.649	525.766	738.269
	13	1,580.764	1,514.172	1,358.747	1,091.071	1,545.823
	14	3,345.564	3,192.456	2,883.060	2,287.762	3,267.091
M62	15	7,161.078	6,830.675	6,164.045	4,843.656	6,972.194
	16	14,980.693	14,288.715	13,011.075	10,160.761	14,599.862
	17	31,293.082	29,723.927	27,300.251	21,186.236	30,511.422
	18	66,187.974	62,386.458	58,765.435	45,177.687	64,773.333
	19	145,642.424	136,472.180	127,537.285	96,765.463	142,447.095
	20	298,561.239	277,545.708	261,991.533	203,310.439	290,179.592
	10	41.223	39.010	36.463	33.271	38.859
	11	87.162	81.538	79.042	74.004	82.854
	12	183.042	166.237	167.510	158.827	174.551
	13	382.700	342.310	347.048	331.373	362.074
	14	730.466	681.998	691.220	657.890	730.466
Rana64	15	1,603.120	1,394.179	1,410.437	1,320.593	1,499.345
	16	3,290.745	2,788.659	2,822.970	2,563.864	3,034.799
	17	6,745.447	5,722.298	5,710.143	5,038.760	6,205.151
	18	15,435.268	13,014.861	13,116.981	11,113.052	14,652.750
	19	36,052.455	30,943.295	30,728.166	24,488.000	34,437.478
	20	81,105.131	69,694.729	70,151.715	53,684.172	77,259.420
	10	53.170	45.653	47.333	42.694	50.834
	11	106.443	85.710	90.742	77.799	101.633
	12	225.836	181.823	197.266	171.660	215.831
	13	543.127	438.867	463.930	390.489	521.854
	14	1,238.322	972.214	1,035.872	862.604	1,182.460
M82	15	2,515.808	1,956.781	2,107.198	1,782.431	2,358.434
	16	5,098.459	3,886.246	4,292.998	3,602.747	4,837.115
	17	10,483.717	7,779.587	8,840.097	7,364.237	9,985.172
	18	21,508.330	15,299.374	17,982.026	15,001.009	20,318.424
	19	43,997.450	30,240.065	36,198.347	30,063.243	41,198.996
	20	89,431.696	59,127.952	71,064.345	57,422.867	82,092.395

Table 3
Continuation of Table 2.

$$s.t. \quad \sum_{\ell \in \mathcal{L}\setminus\{1\}} x_{ij}^{\ell} = 1 \;\; \forall i,j \in \Gamma, i < j$$
$$x_{ij}^{\ell} \geq 0 \;\; \forall i,j \in \Gamma, i < j, \forall \ell \in \mathcal{L} \setminus \{1\}.$$

Formulation 5 decomposes into a family of $n(n-1)/2$ independent subproblems whose analytical solution is trivial and fast to compute, provided suitable choices for the Lagrangian multipliers μ and λ. If the LPLB has been computed at a given node v of the search tree, then the shadow-prices of constraints (44) and (45), respectively, may constitute licit choices for μ and λ, respectively, and the lower bounds for the children nodes of v can be then computed by solving Formulation (5), hereafter denoted as LagLB, at those nodes. We have observed in computational experiments that the quality of the LagLB degrades quickly in function of the depth of a descendant node with respect to v. In particular, the LagLB is already poorer than any of the previous bounds for the children of the children of a generic node v of the search tree. However, if the computation of the LPLB and of the LagLB is alternated during the search tree then it is possible to reach a good trade off in terms of tightness of the lower bound and solution times.

5. Upper bounds on the optimal solution to the BMEP

A possible approach to obtain an upper bound on the optimal solution to the BMEP consists of using Hendy and Penny [43]' SAS in a greedy fashion. The basic idea consists of evaluating first all of the possible insertions of a taxon into a given sub-phylogeny and subsequently selecting the insertion that results minimal with respect to a given criterion $C(\boldsymbol{T})$ (see Algorithm 2). An alternative to the greedy SAS is the *agglomerative approach* [38], i.e., an algorithm that starts from an infeasible solution to the BMEP (e.g., a star tree with n terminals (taxa)) and iteratively cluster taxa according to a specific optimality criterion, until a phylogeny of Γ is obtained. Algorithm 3 formalizes the idea at the core of an agglomerative approach. The algorithm starts with the star tree and inserts a new internal edge (u_1, u_2), by replacing the internal vertex u. Such an edge is determined by minimizing a selection criterion on all bipartitions

$P = (N_1, N_2)$ of the neighborhood of u. Then, the neighborhood of u_1 and u_2 are set to be $N_1 \cup \{u_2\}$ and $N_2 \cup \{u_1\}$, respectively. The algorithms iterates this process until obtaining a phylogeny of Γ, i.e., until agglomerating $(2n-3)$ edges. Figure 5 shows two consecutive steps of the algorithm. In particular, starting from a star tree with eight taxa, a new tree can be obtained by inserting an edge between the star tree on taxa $\{1,2,3,6,7,8\}$ and the pair $\{4,5\}$. The subsequent step may consist, e.g., of inserting an internal edge between the subtrees on taxa $\{1,7,8\}$ and $\{2,3,4,5,6\}$. Observe that both Algorithms 2 and 3 are correct greedy algorithms for the BMEP. Given a sub-phylogeny of $S \subseteq \Gamma$, a possible greedy selection criterion for Algorithm 2 consists of minimizing

$$C(T) = L(T_S) + \sum_{t \in \Gamma \setminus S} \min_{\substack{i,j \in S \\ i<j<t}} \frac{1}{2}\left(d_{it} + d_{jt} - d_{ij}\right)$$

already introduced by Pardi [53]. A possible greedy selection criterion for Algorithm 3 consists of picking a vertex u according to the bipartition $P = (N_1, N_2)$, $|N_1| > |N_2| = 2$, which minimizes the change in Semple and Steel's extension [63] of length function (1) for unrooted non-binary trees. This method is known in the literature as the *Neighbor-Joining algorithm* (NJ) [36, 40, 62, 65]. After having minimized the greedy selection criterion, we carry on a local search based on the *Nearest Neighbor Interchange* (NNI) described in [1] to improve the quality of the upper bound provided by both algorithms. For the sake of space, we refer the reader interested in a comprehensive discussion on NNI to [1]. Here it suffices to say that this local search swaps subtrees adjacent to a given internal edge of a partial phylogeny until no further improvement is possible.

6. A massively parallel implicit enumeration algorithm

The exploration strategy at the core of Hendy and Penny's SAS [43] is naturally prone to parallelization. Indeed, the solution subspaces generated by the SAS by means of new insertions on a given sub-phylogeny constitute a partition

Algorithm 2: Stepwise addition strategy - Greedy search

Input: Ordered set of taxa $\Gamma = \{1, \dots, n\}$, distance matrix D for Γ
Output: A phylogeny of Γ

$t \leftarrow 3$;
$T \leftarrow$ The only sub-phylogeny of Γ for $\{1,2,3\}$;
while $t < n$ **do**
  $t \leftarrow t + 1$;
  $\hat{e} \leftarrow \arg\min\{C(T \oplus_e t) : e \in E(T)\}$;
  $T \leftarrow T \oplus_{\hat{e}} t$;
return T;

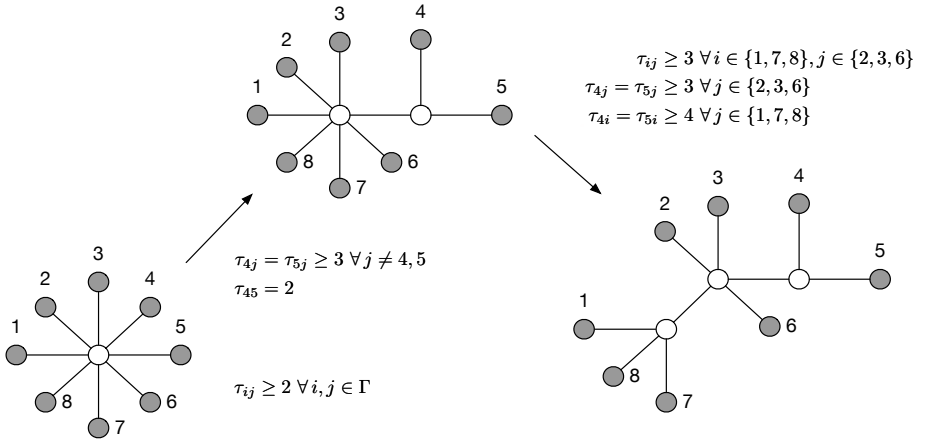


Figure 5: An example of two steps of the agglomerative approach in the case of eight taxa.

Algorithm 3: Agglomerative approach - Greedy search

Input: Set of taxa $\Gamma = \{t_1, \dots, t_q\}$, distance matrix D for Γ,
Output: A phylogeny of Γ

$V \leftarrow \Gamma \cup \{v_1\}$;
$E \leftarrow \{\{t_i, v_1\} \ : \ t_i \in \Gamma\}$;
while $|E| < (2n-3)$ **do**
  $u \leftarrow$ A vertex of degree at least four in (V, E) that minimizes a selection criterion $C(P)$ for a bipartition $P = (N_1, N_2)$ of its neighbors;
  $V \leftarrow V \setminus \{u\} \cup \{u_1, u_2\}$;
  $E \leftarrow E \setminus \{(u, v) \ : \ (u, v) \in E\} \cup \{(u_1, v) \ : \ v \in N_1\} \cup \{(u_2, v) \ : \ v \in N_2\} \cup \{\{u_1, u_2\}\}$;
return (V, E);

of the original solution subspace of the BMEP and can therefore be explored independently from one another. This is the case, e.g., for the subspaces constituted by the phylogenies of Γ derived from the respective three sub-phylogenies obtained by inserting the fourth taxon on the initial star tree in Figure 4. Strangely enough, however, this particular feature of Hendy and Penny's SAS [43] has never been exploited in the literature on the BMEP to speed up the exact resolution of its instances. In this section, we close this gap, by studying possible parallelization paradigms that may prove particularly effective to this end.

We start by observing that the number of computing cores available in a shared memory environment is usually even. In contrast, the number of sub-phylogenies for a fixed subset of taxa $S \subseteq \Gamma$ is odd, precisely equal to $(2|S|-5)!!$, and exponentially growing in function of the cardinality of S. This mismatch makes inefficient any parallelization scheme of the implicit enumeration search that consists of just delegating the exploration of a sub-phylogeny (hence of a subspace) to the available computing cores. Moreover, because entire subspaces could be pruned during the implicit enumeration search, in such parallelization schemes some computing cores could go in a idle status by causing so a useless waste of computing resources.

An efficient parallelization scheme, therefore, must include both a scalable exploration strategy of the solution space and a dynamic balance of the workload so as to avoid as much as possible the absence of cores in idle status. In order to design such a scheme, we first define a *job* as a triplet (Γ, T_S, F) such that T_S is a sub-phylogeny of Γ and $F \subseteq E(T_S)$ a subset of edges in T_S. Moreover, we also introduce a *queue of jobs* Q whose length may be dynamically change in function of the available computing cores that are in a idle status. Then, a possible parallel exploration strategy of the solution space of the BMEP can be outlined as in Algorithm 4.

Specifically, Algorithm 4 first performs an initialization phase. It allocates shared data structures for all the available cores in the system. Shared data contain information about e.g., the branches processed by a core or whether a core

Algorithm 4: Stepwise addition strategy - Parallel initialization and termination

Input: Distance matrix D for Γ, queue Q of jobs, ordered set of taxa $\Gamma = \{1, \dots, n\}$
Output: A phylogeny T^* of Γ

Allocate and initialize shared memory;
Order set Γ;
$T^* \leftarrow$ the best upper bound;
$T_S \leftarrow$ the only sub-phylogeny of Γ for $S = \{1, 2, 3\}$;
Calculate and store the best lower bound for T_S;
for $e \in E(T_S)$ **do**
  $T \leftarrow T_S \oplus_e 4$;
  Push $(\Gamma, T, E(T))$ into Q;
while Q not empty **do**
  Execute in parallel $(\Gamma, T, F) = \text{pop}(Q)$ and $\text{SAS}(D, \Gamma, T, F, T^*, Q)$ on an idle core;
return T^*;

Algorithm 5: Stepwise addition strategy - Implicit enumeration of phylogenies on one core (SAS)

Input: Distance matrix D for Γ, the set of taxa Γ, a sub-phylogeny T_S of Γ, edge set F with $F \subseteq E(T_S)$, phylogeny T^* of Γ, queue Q

```
1  t ← |S| + 1;
2  for e ∈ F do
3      T ← T_S ⊕_e t;
4      Calculate L(T);
5      Update time limit information in the shared memory;
6      if S = Γ and L(T) < L(T*) then
7          T* ← T_S;
8      else
9          continue ← FALSE;
10         if LagLB(T_S) < L(T*) then
11             if LPLB(T_S) < L(T*) then
12                 continue ← TRUE;
13         if continue then
14             if There exist idle cores or queue Q contains less jobs than there are overall cores then
15                 Partition E(T) = F_1 ∪ F_2 such that ||F_1| − |F_2|| ≤ 1;
16                 inserted ← FALSE;
17                 while Q not full and time limit not exceeded and not inserted do
18                     inserted ← Push job (Γ, T, F_2) into Q;
19                 SAS(D, Γ, T, F_1, T*, Q);
20             else
21                 SAS(D, Γ, T, F, T*, Q);
```

is idle or not. Each core can read and modify the records relative to its own identifier. All of the other cores instead can just read such information, independently from one another and in an asynchronous way, by simplifying so the implementation of the stopping criterion of Algorithm 4. The initialization phase terminates with the following steps. The set Γ is ordered and the best upper bound on the optimal solution to the BMEP is calculated and stored as described in Section 5. Then, the algorithm constructs an initial sub-phylogeny T_S for the first 3 taxa in Γ. Furthermore, it calculates and stores the best lower bound for T_S (see Section 4) and it creates a queue of jobs Q, by considering all of the possible insertions of the fourth taxon on T_S. Finally, the algorithm runs the parallel exploration of the subspaces of BMEP solutions derived by all of the possible insertions of the remaining taxa on the sub-phylogenies in Q and stops once that Q is empty. Figure 6 further refines the high-level view of the exploration strategy provided by Algorithm 4.

The global search strategy starts at lines 9 and 10. Note that at the beginning and at the end of the while loop, all cores need to be synchronized to ensure both a proper data initialization and a report of the results. This is indicated by the dashed arrows in Figure 6. Observe also that the stopping criterion at line 9 of Algorithm 4 can be slightly changed so as to account for a time-based stopping criterion (e.g., the computing time exceeded a maximum running time allowed). A proper parallel implementation of Algorithm 4 involves determining how jobs are retrieved from Q and how they are processed and added to Q. These steps are written in bold in Figure 6. Retrieving and deleting a job from the queue Q can be trivially implemented by semaphores, i.e., by making sure that the access to the queue is given to exactly one core at a time. The processing and adding of jobs to Q requires instead a bit more attention. Both operations are resumed in Figure 6 by means of the statement *Run a SAS on the local data* and stated by SAS(D, Γ, T, F, T^*) in Algorithm 4. This line calls for the executions of the steps outlined in Algorithm 5.

The input of Algorithm 5 includes the distance matrix $\mathbf{D}$ and the job (Γ, T_S, F). For every newly enumerated sub-phylogeny T, line 4 of Algorithm 5 computes the objective function value of the BMEP. Subsequently, line 5 of the algorithm checks if the time limit has been exceeded. This is necessary due to the fact that Algorithm 5 is employed by a parallel algorithm but the for-loop in line 2 runs independently from any action taken by other cores. Then, line 6 checks if a new best-so-far solution to the BMEP has been found and in the positive case stores it in T^*.

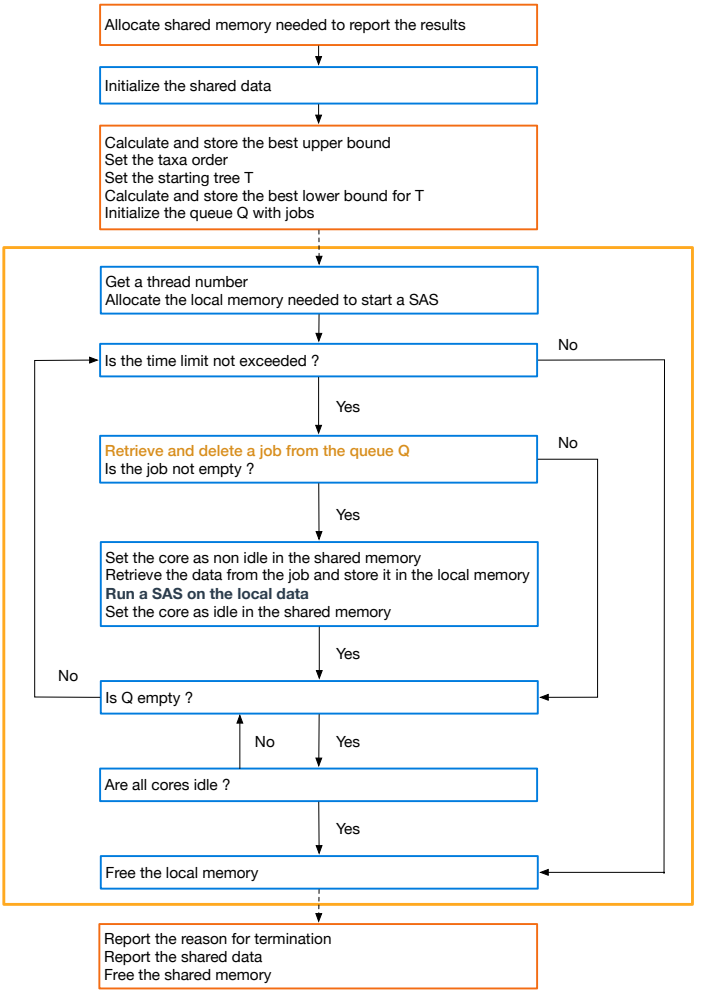


Figure 6: A description of Algorithm 4 in a shared memory system. We are given a set of cores which can access a shared memory and which can run in parallel independently. Statements inside red and blue boxes are processed by a fixed master core and every core, respectively. The yellow box contains all statements that are part of the global search strategy indicated in lines 8 and 9 of Algorithm 4.

This updating step must be carried out by means of semaphores to ensure consistency and avoid overwriting. Next, lines 9 to 12 introduce a bounding procedure for all sub-phylogenies T_S with $S \subset \Gamma$. First, line 10 solves LagLB as explained in Section 4. Subsequently, line 11 solves LPLB to attempt pruning the search for an optimal solution at sub-phylogeny T_S. If LPLB(T_S) can be improved without exceeding the intermediate optimal objective function value $L(T^*)$, then Algorithm 5 continues with our SAS because a series of branchings of T_S might produce an optimal solution to the BMEP. Line 21 calls SAS(D, Γ, T, F, T^*, Q) after the bounding procedure in lines 9 to 12 to complete our extended SAS for the sub-phylogeny T_S and to ensure that our parallel algorithm in Figure 6 terminates after reporting an optimal solution to the BMEP. However, to decrease the overall running time, lines 14 to 19 seek to rebalance the workload for every core by continuously monitoring the length of the queue Q. In particular, if Q is already very long and all other cores are running Algorithm 5, adding new jobs to the queue Q may prove pointless. Instead, if the length of Q is short or there are idle cores, then the adding of at least one new job in Q may prove beneficial. In such case, the algorithm partitions $E(T)$ into two sets F_1 and F_2 of similar size, by giving rise to two new jobs, say (Γ, T, F_1) and (Γ, T, F_2), one of which is processed by the current core and the other is added to the queue Q.

Algorithm	Order of Γ	Order of F	Order of Q	Algorithm	Order of Γ	Order of F	Order of Q
PS01	None	Increasing	FIFO	PS17	Algorithm 2	Increasing	FIFO
PS02	None	Increasing	FILO	PS18	Algorithm 2	Increasing	FILO
PS03	None	Increasing	NINE	PS19	Algorithm 2	Increasing	NINE
PS04	None	Increasing	NDNE	PS20	Algorithm 2	Increasing	NDNE
PS05	None	NDLB	FIFO	PS21	Algorithm 2	NDLB	FIFO
PS06	None	NDLB	FILO	PS22	Algorithm 2	NDLB	FILO
PS07	None	NDLB	NINE	PS23	Algorithm 2	NDLB	NINE
PS08	None	NDLB	NDNE	PS24	Algorithm 2	NDLB	NDNE
PS09	TSP	Increasing	FIFO	PS25	Algorithm 3	Increasing	FIFO
PS10	TSP	Increasing	FILO	PS26	Algorithm 3	Increasing	FILO
PS11	TSP	Increasing	NINE	PS27	Algorithm 3	Increasing	NINE
PS12	TSP	Increasing	NDNE	PS28	Algorithm 3	Increasing	NDNE
PS13	TSP	NDLB	FIFO	PS29	Algorithm 3	NDLB	FIFO
PS14	TSP	NDLB	FILO	PS30	Algorithm 3	NDLB	FILO
PS15	TSP	NDLB	NINE	PS31	Algorithm 3	NDLB	NINE
PS16	TSP	NDLB	NDNE	PS32	Algorithm 3	NDLB	NDNE

Table 4
Settings for the algorithm PSx: the taxa in Γ may be ordered according to (i) the input order of the rows of the distance matrix D (denoted as "None"); (ii) the order provided by shortest Hamiltonian tour obtained when solving the Traveling Salesman Problem (TSP) on the instance encoded by D; and (iii) the order obtained by shortcutting the tree returned by Algorithm 2 and 3, respectively. The set F of edges of a sub-phylogeny can be either (i) ordered in increasing way with respect to the initial star-tree or (ii) non-decreasing ordered by pre-calculating the lower bound one would obtain in the next recursion step by branching on an edge (NDLB). The order of Q can be *Non-Increasing in the Number of Edges* (NINE) or *Non-Decreasing in the Number of Edges* (NDNE). FIFO and FILO stand for the usual *First-In First Out* and *First-In Last-Out* strategies, respectively.

7. Computational experiments

In the previous section, we saw that the parallel implicit enumeration algorithm requires the specification of: the order in which the taxa in Γ are processed, the order in which edges in F are processed in line 2 of Algorithm 5, and the order in which the jobs in the queue Q are extracted and processed. In this section we consider and compare alternative implementation settings for these orders on benchmark sets of biological and artificial instances of the BMEP provided by [16, 24, 38].

We considered three possible settings to select a processing order of the taxa in Γ, namely: (i) the order corresponding to the rows of the input distance matrix D (hereafter, this order is denoted as None); (ii) the order provided by shortest Hamiltonian tour obtained when solving the Traveling Salesman Problem (TSP) on the instance encoded by D; and (iii) the order obtained by shortcutting the tree returned by Algorithm 2 and 3, respectively. Concerning the set F of edges of a sub-phylogeny, we considered the case in which F is (i) ordered in increasing way with respect to the initial star-tree or (ii) ordered in non-decreasing way by pre-calculating the lower bound one would obtain in the next recursion step by branching on an edge (hereafter, this order is denoted as NDLB). Finally, concerning the order in which the jobs in Q are extracted and processed, we considered the case in which Q is ordered *Non-Increasing in the Number of Edges* (NINE); the case in which Q is ordered *Non-Decreasing in the Number of Edges* (NDNE); and the classic cases in which the jobs are extracted in *First-In First Out* (FIFO) and *First-In Last-Out* (FILO), respectively. By combining these implementation settings we derived 32 possible implementations of the parallel implicit enumeration algorithm described in the previous section, each of which has been denoted as PSx, $x \in [1, 32]$, and specified in Table 4.

7.1. Implementation details

All the implementations described in this section have been coded in ANSI C++, by relying on FICO Xpress Optimizer libraries v33.01.05 for linear programming and on OpenMP [19] version 3.0 for parallelism. The codes have been compiled by means of GGC compiler version 4.8.5, libc version 2.17, and run on a 2x8 Core E5-2667v3 Processor at 3.2 GHz and 256GB RAM, operating system CentOS Linux 7 release 7.9.2009 (kernel linux 3.10.0-1160.71.1.e17.x86_64). We assumed one hour as maximum runtime per instance and rescaled the objective function by a factor 2^n in order to reduce possible numerical stability problems. Codes and data used in the experiments can be downloaded at `https://github.com/mfrohn/BMEPparallel`.

Algorithm	Order of Q	Value of p	Value of c	Algorithm	Order of Q	Value of p	Value of c
PS33	FIFO	3	-	PS47	NINE	6	-
PS34	FILO	3	-	PS48	NDNE	6	-
PS35	NINE	3	-	PS49	FIFO	7	-
PS36	NDNE	3	-	PS50	FILO	7	-
PS37	FIFO	4	-	PS51	NINE	7	-
PS38	FILO	4	-	PS52	NDNE	7	-
PS39	NINE	4	-	PS53	FIFO	8	-
PS40	NDNE	4	-	PS54	FILO	8	-
PS41	FIFO	5	-	PS55	NINE	8	-
PS42	FILO	5	-	PS56	NDNE	8	-
PS43	NINE	5	-	PS57	FIFO	9	-
PS44	NDNE	5	-	PS58	FILO	9	-
PS45	FIFO	6	-	PS59	NINE	9	-
PS46	FILO	6	-	PS60	NDNE	9	-
PS61	FIFO	2	2	PS73	FIFO	3	2
PS62	FILO	2	2	PS74	FILO	3	2
PS63	NINE	2	2	PS75	NINE	3	2
PS64	NDNE	2	2	PS76	NDNE	3	2
PS65	FIFO	2	3	PS77	FIFO	3	3
PS66	FILO	2	3	PS78	FILO	3	3
PS67	NINE	2	3	PS79	NINE	3	3
PS68	NDNE	2	3	PS80	NDNE	3	3
PS69	FIFO	2	4	PS81	FIFO	3	4
PS70	FILO	2	4	PS82	FILO	3	4
PS71	NINE	2	4	PS83	NINE	3	4
PS72	NDNE	2	4	PS84	NDNE	3	4

Table 5
Settings for the algorithm PSx when assuming that the order of Γ derives from Algorithm 3 and the order of F is increasing with respect to the initial star-tree. FIFO and FILO stand for the usual *First-In First Out* and *First-In Last-Out* strategies, respectively.

7.2. Implementation settings

The systematic analysis of the computational results relative to the 32 different implementation settings considered in this section exceeds the page limits of the journal and has been therefore omitted. The interested reader, however, can find it online at `https://github.com/mfrohn/BMEPparallel` (in particular, in Figures 1 to 5 of the file "Supplementary Material"). In this section we just limit to resume the most important findings arising from these massive computational experiments. We first observed that the choice of the processing order of jobs in Q has an impact of multiple magnitudes smaller than both the choice for the processing order of the taxa in Γ and the processing order of F. Moreover, no queue order yields a statistical significant advantage compared to all others. We also observed that ordering F in increasing fashion leads overall to better runtimes compared to the NDLB order. Specifically, we observed that the worst case and 75% quartile runtimes improve dramatically for increasing n. This leads to a twofold conclusion: on the one hand, the additional computational overhead for the calculation of lower bounds required at each iteration to determine the order of F affects negatively the runtime for small n. On the other hand, the calculation of lower bounds throughout the tested algorithms dominates all other contributions to the total computation time. Preprocessing the lower bounds for future iterations does not scale well with a higher number of cores and increasing n because when a precalculated lower bound is employed by a core the corresponding node of the search-tree might be already pruned by a different core. The computational results relative to the processing order of the taxa in Γ lead instead to less net conclusions. The best worst-case performance is always reached by the order derived from Algorithm 3, but the lowest best-case and 25% quartile runtime is achieved by other orders of Γ for most n. However, since the order derived from Algorithm 3 performs statistically significant better than all other configurations for $n = 20$ we decided to keep this order as the best choice for further experiments. Overall, the analysis indicates that the configurations PS25-PS28 of Table 4 perform the best.

We also considered further modifications of lines 14 to 19 in Algorithm 5 to strengthen the performance of configurations PS25 to PS28. First, we investigate the number p of sets of the partition $E(T) = \bigcup_{i=1}^{p} F_i$ with default value $p = 2$ in Algorithm 5. To this end, we consider configurations PS25 to PS28 for $p \in \{3, \dots, 9\}$. We keep the assumption of similar sized subsets in line 15, i.e., $\Big||F_i| - |F_j|\Big| \leq 1$ for all $i, j = 1, \dots, p$. Then, we consider the modifications of configurations PS25 to PS28 listed in the upper half of Table 5. The computational results for configurations PS33 to PS60 of Table 5 are shown in the Supplementary Material. Here we can observe that no choice of the parameter

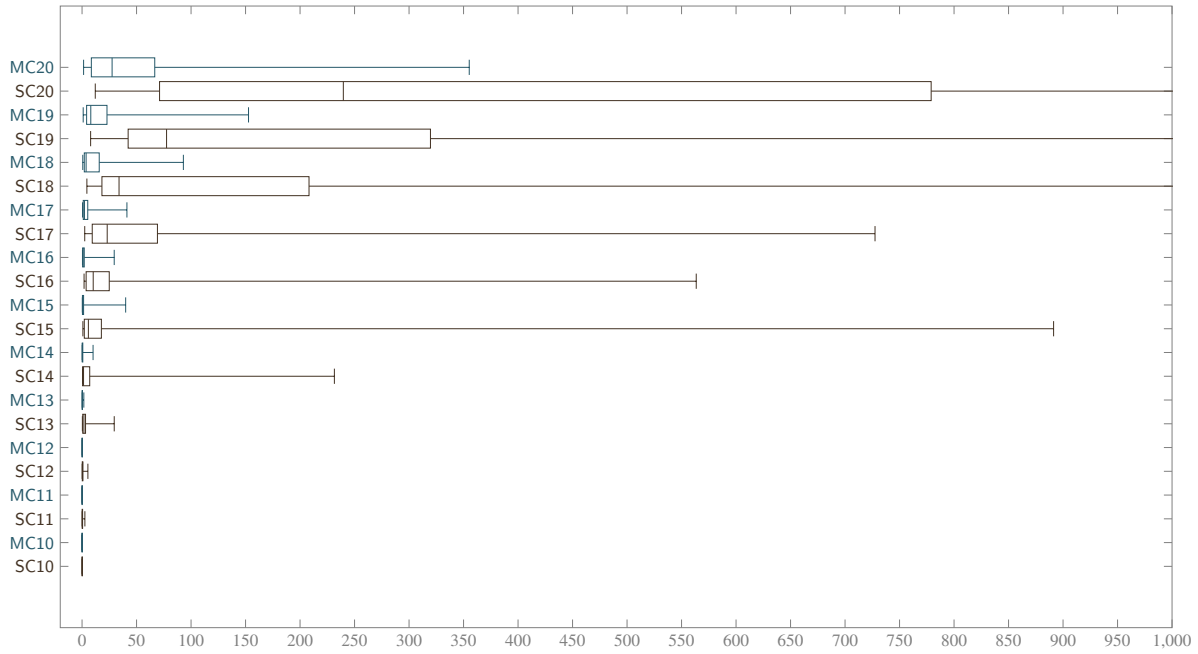


Figure 7: A boxplot of the runtime in seconds for the algorithm PS76 on a single core and on 32 cores, e.g., SC10 is the runtime of the single core version of PS76 on 10 taxa and MC17 is the runtime of the multi core version of PS76 on 17 taxa. The plots are only shown for at most 1,000 seconds of runtime to improve the readability.

p yields a statistical significant advantage compared to all other evaluated choices. However, for sufficiently large n, increasing the number of partitions sets p seems to slow down the execution of the tested algorithms. This observation is consistent with the idea that sufficiently small partition sets lead to an overabundance of jobs in the queue. This increase in computation time for managing the distribution of jobs can not be compensated by the decrease in idle time for each individual core. Hence, we only consider $p = 2$ and $p = 3$ as part of the best performing configurations in the worst case.

As a second extension of the partitioning process of the edge set F in Algorithm 5 we introduce a positive constant integer c to define the minimum cardinality

$$c^*(p) = \min\left\{\max\left\{c, \left\lfloor\frac{|E(T)|}{p}\right\rfloor\right\}, |E(T)|\right\}$$

of partition sets F_i, $i = 1, \ldots, p$. Employing $c^*(p)$ as a lower bound on the number of edges included in one job ensures that the balancing of workloads can be dynamically tailored to the size of sub-phylogenies. Overall, any partition of edges $E(T)$ aims at making at least as many jobs available in the queue as there are cores in the shared memory system. The computational results for configurations PS25 to PS28, PS33 to PS36 and PS61 to PS84 of Table 5 are shown in the Supplementary Material. Again, we observe that no specification is significantly the best among all tested configurations. Smaller choices for the constant c seem to be preferable for increasing n.

7.3. Comparing PS76 versus the state-of-the-art

The implementation setting PS76 offers the best worst case performance among all analyzed configurations PS01 to PS84 for $n = 20$. Tables 6 and 7 summarize the numerical results with respect to the solution time and the number of branches taken by configuration PS76 to solve Catanzaro et al.'s benchmark biological instances of the BMEP [14]. Table 8 completes Tables 6 and 7 by listing all the instances that can be solved by configuration PS76 but not the single core version described in Catanzaro et al. [14]. In general, we can see that the single core version of configuration PS76 as well as the multi core version solve the BMEP instances faster than Catanzaro et al.'s state-of-the-art solution algorithm [14]. Moreover, both tables show that using a single core is only beneficial when dealing with very small sets of taxa. This is explained by the fact that the computational overhead of managing multiple cores for configuration

				Single Core		Multi Core		
Data set	Taxa	Optimum	Gap (%)	Time (s)	Branches	Time (s)	Branches	Imp. (%)
Primates12	10	124.968	2.57	**0.030**	216	0.080	216	-0.77
	11	332.834	2.59	0.086	344	**0.075**	344	0.01
	12	802.589	2.47	0.160	579	**0.074**	579	0.08
M17	10	105.134	1.05	0.202	1454	**0.071**	1,709	0.13
	11	261.233	1.18	0.493	2,796	**0.085**	2,779	0.41
	12	541.632	1.21	0.640	4,749	**0.137**	5,222	0.50
	13	1,181.598	1.65	3.121	18,553	**0.317**	20,685	2.80
	14	2,408.066	1.65	0.504	2,917	**0.150**	4,813	0.35
	15	4,998.295	1.65	7.246	32,492	**0.845**	47,878	6.40
	16	10,225.560	1.70	11.177	43,126	**1.233**	65,529	9.94
	17	20,788.112	1.64	12.930	39,161	**1.611**	65,403	11.32
M18	10	190.331	2.11	0.147	744	**0.049**	934	0.10
	11	396.942	2.57	0.232	1,598	**0.073**	1,681	0.16
	12	805.937	2.71	0.596	3,875	**0.126**	3,845	0.47
	13	1,758.111	3.27	2.486	17,495	**0.315**	18,224	2.17
	14	3,599.678	3.62	7.397	44,057	**0.663**	48,547	6.73
	15	7,746.218	3.47	18.412	94,430	**1.462**	99,285	16.95
	16	15,973.016	3.51	24.440	122,068	**1.984**	138,598	22.45
	17	32,345.662	3.53	43.348	215,505	**3.704**	250,581	39.64
	18	66,029.112	3.55	128.097	584,269	**10.343**	691,333	117.75
SeedPlant25	10	91.458	5.51	0.065	581	**0.046**	926	0.02
	11	206.250	5.10	0.097	717	**0.065**	1,482	0.03
	12	427.816	4.87	0.109	595	**0.088**	595	0.02
	13	943.774	5.30	0.289	1,788	**0.146**	2,074	0.14
	14	1,929.219	5.06	0.483	2,378	**0.207**	3,506	0.28
	15	4,085.763	4.66	0.771	3,173	**0.352**	13,154	0.42
	16	8,353.457	5.76	1.846	8,963	**0.387**	12,092	1.46
	17	17,314.429	6.21	6.211	39,328	**0.816**	55,241	5.39
	18	36,156.527	6.92	33.931	187,267	**3.722**	295,186	30.21
	19	75,261.323	6.97	107.280	516,428	**9.758**	705,960	97.53
	20	164,507.262	9.11	391.165	1,738,599	**43.789**	3,094,726	347.38
M43	10	105.020	1.29	0.105	490	**0.048**	477	0.06
	11	209.282	2.39	0.515	4,940	**0.087**	4,943	0.43
	12	434.326	1.79	0.693	4,584	**0.128**	5,053	0.56
	13	895.424	1.74	1.021	8,062	**0.164**	8,168	0.86
	14	1,808.970	1.75	1.900	12,414	**0.268**	12,477	1.63
	15	3,965.234	1.53	4.354	20,495	**0.507**	24,909	3.85
	16	8,219.835	1.51	9.291	43,883	**0.900**	48,793	8.39
	17	16,798.759	1.82	33.213	129,312	**2.802**	154,811	30.41
	18	35,383.158	1.44	23.212	82,653	**1.940**	86,798	21.27
	19	71,769.903	1.43	40.449	141,845	**3.495**	162,376	36.95
	20	157,472.424	1.48	88.241	312,258	**7.586**	355,902	80.65

Table 6
Numerical results obtained for the algorithm PS76 on a single core and on 32 cores. The last column shows the improvement in the solution time of the multi core approach relative to the time required to terminate PS76 on one core. The improvement measures the difference in solution time between the single core and multi core version. Bold indicates the approach that performed the best.

				Single Core		Multi Core		
Data set	Taxa	Optimum	Gap (%)	Time (s)	Branches	Time (s)	Branches	Imp. (%)
	10	152.482	1.79	0.182	1,201	**0.070**	1,412	0.11
	11	328.645	1.61	0.286	1,960	**0.077**	1,393	0.21
	12	685.972	2.20	1.021	6,249	**0.177**	7,206	0.84
	13	1,502.870	2.33	3.408	16,105	**0.363**	20,661	3.05
	14	3,094.888	2.27	6.913	35,580	**0.730**	40,253	6.18
RbcL55	15	6,448.259	2.46	17.498	66,903	**1.386**	68,164	16.11
	16	13,455.292	2.50	26.492	96,772	**2.101**	105,599	24.39
	17	27,804.246	2.94	146.577	622,052	**9.796**	649,156	136.7 8
	18	56,175.236	2.90	288.358	1,208,182	**21.176**	1,339,498	267.18
	19	114,623.865	2.97	390.419	1,482,378	**27.242**	1,557,301	363.18
	20	236,424.336	4.63	908.280	3,515,861	**74.330**	4,482,361	833.95
	10	163.876	1.07	0.074	235	**0.041**	297	0.03
	11	350.425	1.50	0.137	514	**0.076**	605	0.06
	12	753.821	2.06	0.197	813	**0.083**	1,131	0.11
	13	1,580.764	2.21	0.575	3,195	**0.153**	3,131	0.42
	14	3,345.564	2.35	0.423	1,975	**0.164**	3,931	0.26
M62	15	7,161.078	2.64	2.357	8,785	**0.352**	11,519	2.00
	16	14,980.693	2.54	4.308	16,508	**0.576**	24,084	3.73
	17	31,293.082	2.50	10.388	37,089	**1.185**	53,167	9.20
	18	66,187.974	2.14	13.251	48,506	**2.170**	114,739	11.08
	19	145,642.424	2.19	47.787	159,650	**6.259**	309,798	41.53
	20	298,561.239	2.81	65.398	220,435	**11.339**	583,732	54.06
	10	41.223	5.74	0.301	2,134	**0.084**	2,134	0.22
	11	87.162	4.94	0.463	1,841	**0.092**	1,841	0.37
	12	183.042	4.64	0.457	1,905	**0.100**	1,905	0.36
	13	382.700	5.39	0.633	2,698	**0.153**	2,698	0.48
	14	730.466	5.60	0.824	3,098	**0.209**	3,126	0.62
Rana64	15	1,603.120	6.47	1.163	3,968	**0.268**	4,134	0.89
	16	3,290.745	7.78	2.055	8,094	**0.376**	8,535	1.68
	17	6,745.447	8.01	2.457	8,871	**0.435**	9,480	2.02
	18	15,435.268	5.07	4.428	15,584	**0.667**	19,080	3.76
	19	36,052.455	4.48	7.865	27,222	**1.047**	33,430	6.82
	20	81,105.131	4.74	12.124	36,885	**1.412**	43,256	10.71
	10	53.170	4.39	0.373	3,939	**0.101**	4,567	0.27
	11	106.443	4.52	2.599	37,818	**0.272**	44,071	2.33
	12	225.836	4.43	5.356	54,323	**0.385**	46,455	4.97
	13	543.127	3.92	29.519	264,235	**1.627**	211,810	27.89
	14	1,238.322	4.51	231.497	2,076,753	**10.104**	1,384,383	221.39
M82	15	2,515.808	6.26	891.329	7,101,372	**40.006**	4,888,322	851.32
	16	5,098.459	5.13	563.475	3,276,068	**29.507**	2,710,938	533.97
	17	10,483.717	4.76	727.497	3,716,014	**41.165**	3,331,369	686.33
	18	21,508.330	5.53	1,662.013	7,903,728	**92.955**	6,915,563	1,569.06
	19	43,997.450	6.36	2,648.185	10,924,884	**152.720**	9,662,546	2,495.46
	20	89,431.696	8.21	>3,600	n.a.	**355.224**	20,547,942	3,244.78

Table 7
Continuation of Table 6.

Data set	Taxa	Optimum	Gap (%)	Time (s)	Branches
	20	149,517.567	7.08	37.86	2,638,772
	21	308,829.021	7.30	110.97	6,038,213
	22	619,759.057	7.66	184.25	9,742,797
SeedPlant25	23	1,259,681.860	8.24	625.89	33,099,791
	24	2,565,765.151	21.46	>3,600	n.a.
	25	5,403,865.964	25.09	>3,600	n.a.
	20	155,142.306	1.48	9.23	454,778
	21	320,726.566	1.45	11.49	531,549
M43	22	627,364.757	4.86	>3,600	n.a.
	23	1,259,716.112	4.87	>3,600	n.a.
	24	2,661,312.124	8.77	>3,600	n.a.
	25	5,432,750.885	8.63	>3,600	n.a.
	20	225,473.659	2.96	65.77	3,846,795
	21	480,122.632	2.79	130.04	7,032,882
RbcL55	22	995,596.435	2.73	234.98	11,627,808
	23	2,049,037.184	10.33	>3,600.	n.a.
	24	4,145,906.240	10.69	>3,600	n.a.
	25	8,595,977.714	9.74	>3,600	n.a.
	20	290,179.592	2.18	11.43	589,755
	21	576,663.719	4.87	13.07	563,435
M62	22	1,162,187.379	5.37	>3,600	n.a.
	23	2,366,103.293	8.83	>3,600	n.a.
	24	4,903,286.768	8.80	>3,600	n.a.
	25	9,927,436.569	8.86	>3,600	n.a.
	20	77,259.420	4.74	1.40	42,519
	21	179,623.137	3.48	3.47	127,983
Rana64	22	372,354.375	3.53	4.24	146,933
	23	763,885.643	5.22	11.27	285,718
	24	1,558,525.951	5.18	21.58	516,498
	25	3,413,097.109	8.75	>3,600	n.a.
	20	82,092.395	5.77	359.83	20,752,303
	21	169,136.637	5.43	852.38	44,877,124
M82	22	368,956.166	5.90	3,043.69	135,930,327
	23	752,462.574	6.38	>3,600	n.a.
	24	1,549,887.516	6.91	>3,600	n.a.
	25	3,206,058.298	11.92	>3,600	n.a.

Table 8
The continuation of Tables 6 and 7 for the multi core version on 20 to 25 taxa.

Data set	FastME	PS*	Gap (%)	Time (s)	Branches
B-HA-573-585-BMGE	530,072.565	530,072.565	2.57	11.45	411,028
B-NA-684-472-BMGE	554,220.083	554,220.083	3.45	10.96	604,262
B-NS1-284-344-BMGE	600,436.354	600,436.354	4.69	3.21	116,224
proteic_M2577_40x12260_2005	4,793,364.626	4,793,364.626	6.36	43.91	2,779,707
proteic_M2624_139x348_2006-BMGE	3,569,327.875	3,569,327.875	1.43	26.94	1,477,087
proteic_M2883_91x7386_2007-BMGE	2,208,105.905	n.a.	6.87	>3,600	n.a.
proteic_M3068_50x1000_2006	24,638,785.283	n.a.	2.48	>3,600	n.a.
proteic_M3755_77x9918_2008-BMGE	5,063,227.006	5,063,227.006	1.00	228.71	12,371,673
proteic_M3756_77x11234_2008	6,639,947.471	n.a.	2.49	>3,600	n.a.
B-HA-986-1908-BMGE	432,217.763	432,217.763	8.42	90.06	3,446,320
B-NA-633-1751-BMGE	288,685.631	288,685.631	6.08	15.89	599,867
B-NS-629-1111-BMGE	284,088.517	284,088.517	5.43	24.67	1,239,801
eudicots-BMGE	1,297,053.321	n.a.	2.56	>3,600	n.a.
euros1	964,061.058	n.a.	3.75	>3,600	n.a.
euros2	716,214.753	716,214.753	1.08	132.94	8,434,937
nucleic_M2573_346x897_2006	1,324,915.878	1,324,915.878	2.25	43.40	2,520,330
nucleic_M2839_470x829_2006	3,599,570.957	3,599,570.957	3.89	1,718.55	104,776,984
nucleic_M3862_362x1207_2008	3,102,271.026	n.a.	2.60	>3,600	n.a.
rosids	1,038,195.249	n.a.	3.91	>3,600	n.a.

Table 9
Numerical results obtained for FastME and configuration PS* for $n = 20$ reported in the objective function $2^n \cdot L(T)$. For FastME, small deviations in $2^n \cdot L(T)$ compared to the results reported in the supplementary material in [49] arise from a different use of floating-point arithmetic, i.e., rounding errors. For configuration PS*, the gap at the root node of the search, the running time and the number of branchings processed are shown.

PS76 on a few taxa is more expensive than the potential for improvements in the solution time through parallelization. Furthermore, we observe that configuration PS76 is twice as fast on multiple cores as on a single core 99% of the time. The relative improvements when comparing configuration PS76 on multiple cores and a single core also show that the parallel version is at least one order of magnitude faster than the single core version 28% of the time. This is particularly interesting for instances like *M82* on 20 taxa for which the BMEP was not solvable before within one hour. Moreover, the clear difference in runtime between the two solvers is illustrated in Figure 7. Thus, configuration PS76 increases the number of taxa for which the BMEP can be solved in a reasonable amount of time. This raises the natural question about how much farther faster algorithms than configuration PS76 could go before meeting numerical stability issues. We discuss this question in Section 8.

7.3.1. Comparing PS76 versus FastME

A computational experiment that may attract the attention of the biological community concerns how far the solutions provided by popular heuristics for the BMEP, such as FastME [49], can be with respect to the optimum. To address this issue we considered the sets of molecular sequences contained in FastME's supplementary material [49] (see `http://www.atgc-montpellier.fr/fastme/paper.php`). We configured FastME so as to start from an initial phylogeny constructed by the BIONJ algorithm (see [37]) and then to perform local searches on it based on NNI and SPR interchanges (see [1]). We then carried out the following steps to use the solution provided by FastME as the initial upper bound on the optimal solution to the BMEP in configuration PS76.

1. For each set of molecular sequences, we calculated the default distance matrix, i.e., the model introduced in [30] for DNA data and in [48] for protein data, respectively.

2. For each distance matrix, we constructed a $n \times n$ submatrix including the *n most dissimilar taxa*, i.e., the first n taxa in the order of non-increasing phylogenetic diversity [64].

3. For each distance matrix on the n most dissimilar taxa

   (i) we run FastME to calculate a phylogeny;

   (ii) we run configuration PS76 with two modifications: select the phylogeny produced by FastME as the initial upper bound on the optimal solution to the BMEP and insert taxa in non-increasing phylogenetic diversity order [64]. We call this configuration PS*.

The results of this procedure for $n = 20$ are shown in Table 9. The experiments show that FastME solved to optimality 12 instances of the BMEP; nothing can be predicated about the optimality of the solution found for the reamining 7

Data set	n	FastME	PS*	Gap (%)	Time (s)	Branches	Up.
B-HA-573-585-BMGE	25	17,880,701	17,880,701	2.44	546.34	21,598,004	0
	26	35,655,750	n.a.	2.55	>3,600	n.a.	0
B-NA-684-472-BMGE	24	9,467,537	9,467,537	4.25	2,758.68	89,888,929	0
	25	18,911,659	n.a.	4.23	>3,600	n.a.	0
B-NS1-284-344-BMGE	29	341,239,418	341,239,418	6.69	2,124.72	44,871,572	0
	30	689,965,539	n.a.	7.34	>3,600	n.a.	0
proteic_M2577_40x12260_2005	23	41,973,145	41,973,145	7.30	814.48	38,575,777	0
	24	86,565,511	n.a.	7.74	>3,600	n.a.	0
proteic_M2624_139x348_2006-BMGE	25	125,549,247	125,549,247	2.89	703.86	19,630,921	0
	26	257,595,763	n.a.	2.96	>3,600	n.a..	0
proteic_M3755_77x9918_2008-BMGE	22	21,115,546	21,113,170	1.06	>3,600	117,055,834	2
	23	43,191,028	n.a.	1.08	>3,600	n.a.	0
B-HA-986-1908-BMGE	21	869,072	n.a.	7.35	>3,600	n.a.	0
B-NA-633-1751-BMGE	22	1,270,085	1,270,085	7.22	1,449.53	78,057,725	0
	23	2,660,926	n.a.	8.23	>3,600	n.a.	0
B-NS-629-1111-BMGE	26	19,225,291	19,225,291	8.19	1,320.43	49,798,676	0
	27	39,704,665	n.a.	8.44	>3,600	n.a.	0
euros2	22	2,949,200	2,949,200	1.10	84.00	3,802,128	0
	23	5,959,679	n.a.	1.76	>3,600	n.a.	0
nucleic_M2573_346x897_2006	27	171,662,192	171,652,350	2.92	1,587.63	43,125,380	1
	28	345,752,913	345,731,096	2.88	>3,600	95,013,870	2
nucleic_M2839_470x829_2006	21	7,507,514	7,507,514	4.14	3,141.93	170,364,597	0
	22	15,439,953	n.a.	4.27	>3,600	n.a.	0

Table 10
Continuation of Table 9 where the number of taxa and the number of primal solution updates are shown. In the last column the number of updates of the intermediate optimal solution (Up.) during the execution of PS* is shown.

instances as PS* was unable to terminate within the given time limit (one hour computing time). Because practical data sets often include much more than 20 taxa, it is natural to wonder how far configuration PS* can go within a given time limit. Table 10 addresses this issue. In particular, the table reports on the biggest and smallest value of n for which configuration PS* does terminate and does not terminate within one hour computing time, respectively. Interestingly, this analysis shows that FastME was unable to optimally solve instances from the data sets proteic_M3755_77x9918_2008-BMGE and nucleic_M2573_346x897_2006.

In order to get a better view of the average performance of configuration PS* in comparison to FastME we also looked at simulations based on random trees with parameters values chosen so as to cover some features of real data sets. Specifically, we considered a 24-taxa tree set of size 2,000 that was generated at moderate evolutionary rates (see [24] for detailed specifications). Table 11 shows the results obtained when considering these instances. We observe that configuration PS* can not solve 19.90% of the instances within one hour and FastME can not solve 13.25% of the instances up to optimality, respectively. Again, a certificate of optimality for FastME is derived from configuration PS* only if configuration PS* terminates (within one hour). The FastME solutions are obtained in at most one second. Moreover, configuration PS* proves that at least 67.80% of the solutions produced by FastME are optimal. Thus, we can find a significant set of inputs for which FastME can be improved. However, identifying this set can not be done quickly so far due to the average high running time of configuration PS*. The number of taxa of the input instances considered in our experiments for which we can solve the BMEP up to optimality is limited by the performance of the parallel exact solution algorithms we have discussed. However, as we show in the next section, the range of values for the estimated evolutionary distances d_{ij}, $i, j \in \Gamma$, also plays a crucial role in being able to prove optimality or not.

8. On the connection between numerical stability and statistical consistency of the BMEP

From a numerical analysis perspective, the objective function of the BMEP is subjected to instabilities and underflow errors whenever the generic term $d_{ij}2^{-\tau_{ij}}$ approaches (or get smaller than) the machine epsilon ϵ. Numerical instabilities may severely limit the analysis of practical instances involving hundreds or thousands of taxa (see, e.g., the phylogenetic analysis of the datasets related to SARS-Cov2 [12]) and may occur because of specific values of the path-length τ_{ij} or of the distance d_{ij} or both. Because in any feasible solution to the BMEP the value of τ_{ij} can vary within the discrete interval $[2, n-1]$, the following inequality trivially holds:

$$d_{ij} \cdot 2^{-\tau_{ij}} \geq d_{ij} \cdot 2^{-n+1}. \tag{47}$$

Average gap at the root node of the search	5.66%
Configuration PS* terminated (within one hour)	80.10%
Average number of branchings	26,206,315.71
Average number of branchings when configuration PS* terminated	9,963,219.75
Average solution time	1,021.86 s
Average solution time when configuration PS* terminated	381.35 s
FastME does not solve the BMEP up to optimality	13.25%
Average number of primal updates when FastME does not solve the BMEP up to optimality	2.23
Relative gap when FastME does not solve the BMEP up to optimality	0.05%
FastME does solve the BMEP up to optimality	$\geq$ 67.80%

Table 11
Numerical results obtained for FastME and configuration PS* for the random instances on 24 taxa given by the 24-taxa trees set discussed in [24]. The relative gap is the difference between the optimal solution of FastME and configuration PS* relative to PS*.

Then, the smallest value of n for which $d_{ij} \cdot 2^{-n+1} \leq \epsilon$ is

$$n \geq \left\lfloor \log_2 \left(\frac{d_{ij}}{\epsilon} \right) + 1 \right\rfloor. \tag{48}$$

In 128-bit floating point arithmetic, $\epsilon \approx 5 \cdot 10^{-34}$. Hence, by assuming, without loss of generality, that $\mathbf{D}$ is rescaled in double-stochastic a bound on the smallest value of n for which instabilities and underflow errors may arise is

$$n \geq \left\lfloor \log_2 \left(\frac{1}{5 \cdot 10^{-34}} \right) + 1 \right\rfloor = 111. \tag{49}$$

The smaller the non-diagonal entry of $\mathbf{D}$ the smaller the value of n for which numerical instabilities and underflow errors occur (see Figure 8). Note that the best commercial linear programming solvers on the market are highly optimized for 64-bit floating point arithmetic. This fact dramatically lowers the bound on n to 50 taxa or less.

Finding a way to overcome the occurrence of numerical issues is highly desirable for practical phylogenetic analyses based on the BME criterion [12]. One way to achieve this goal consists of finding a scalar $\alpha > 1$ that may increase as much as possible the right-hand side of the following inequality

$$n \geq \left\lfloor \alpha \cdot \log_2 \left(\frac{d_{ij}}{\epsilon} \right) + 1 \right\rfloor. \tag{50}$$

Note, however, that, independently of the contingent value of α, imposing the presence of such a scalar would imply

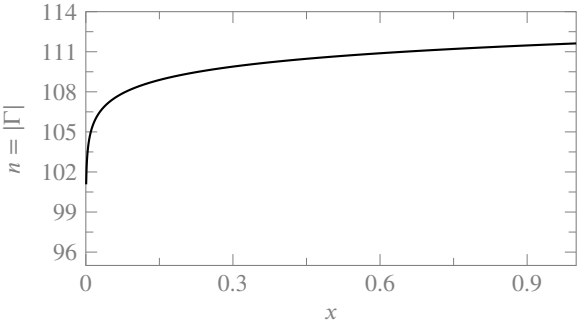


Figure 8: Behavior of the function $\log_2 \left(\frac{x}{\epsilon} \right) + 1$ for $0 < x \leq 1$.

considering the following new length function for the BMEP

$$L_\alpha(T) = \sum_{i\in\Gamma}\sum_{j\in\Gamma\setminus\{i\}} \frac{d_{ij}}{2^{\tau_{ij}/\alpha}}. \tag{51}$$

We refer to the problem of minimizing (51) as the *Rescaled Minimum Evolution Problem* (RMEP).

At a first glance, the BMEP and the RMEP may look quite similar. However, it is possible to show that the RMEP does not preserve the same statistical properties of the BMEP. To see this point, we first observe the following fact:

Proposition 6. *The optimal solutions to the BMEP and the RBMEP are in general different.*

Proof. First note that

$$\lim_{\alpha\to\infty} L_\alpha(T) = \sum_{i\in\Gamma}\sum_{j\in\Gamma\setminus\{i\}} d_{ij}$$

hence, for increasing values of α, the objective function of the RBMEP behaves as the function

$$\tilde{L}(T) = \sum_{i\in\Gamma}\sum_{j\in\Gamma\setminus\{i\}} d_{ij}\tau_{ij} \tag{52}$$

which is linear in τ instead of concave as for the BMEP. Minimizing the RBMEP in general provides optima that are different from those of the BMEP. To see this point, assume $n = 6$ and consider the following input distance matrix

$$\mathbf{D} = \begin{pmatrix} 0 & 2 & 1 & 1 & 1 & 1 \\ 2 & 0 & 1 & 1 & 1 & 1 \\ 1 & 1 & 0 & 2 & 1 & 1 \\ 1 & 1 & 2 & 0 & 1 & 1 \\ 1 & 1 & 1 & 1 & 0 & 2 \\ 1 & 1 & 1 & 1 & 2 & 0 \end{pmatrix}. \tag{53}$$

Then, it is easy to see that the optimum for (52) is attained for the PLM

$$\tau = \begin{pmatrix} 0 & 2 & 4 & 4 & 4 & 4 \\ 2 & 0 & 4 & 4 & 4 & 4 \\ 4 & 4 & 0 & 2 & 4 & 4 \\ 4 & 4 & 2 & 0 & 4 & 4 \\ 4 & 4 & 4 & 4 & 0 & 2 \\ 4 & 4 & 4 & 4 & 2 & 0 \end{pmatrix} \tag{54}$$

which encodes a balanced UBT and has value 120. The optimum for the BMEP is instead attained for the PLM

$$\tau' = \begin{pmatrix} 0 & 2 & 3 & 4 & 5 & 5 \\ 2 & 0 & 3 & 4 & 5 & 5 \\ 3 & 3 & 0 & 3 & 4 & 4 \\ 4 & 4 & 3 & 0 & 3 & 3 \\ 5 & 5 & 4 & 3 & 0 & 2 \\ 5 & 5 & 4 & 3 & 2 & 0 \end{pmatrix}. \tag{55}$$

which encodes a caterpillar UBT whose value is 4.25. □

Proposition 6 suffices to predicate the general statistical inconsistency of the RBMEP. A licit question, however, is whether the optimal solution to the RBMEP may preserve the statistical consistency at least for some values of $\alpha > 1$. Unfortunately, the next two propositions cast a cloud also over this possibility:

Proposition 7. *For any scalar $\alpha > 1$, $L_\alpha(T)$ overestimates the length of the true phylogeny of the set Γ of taxa encoded by the input distance matrix* **D**.

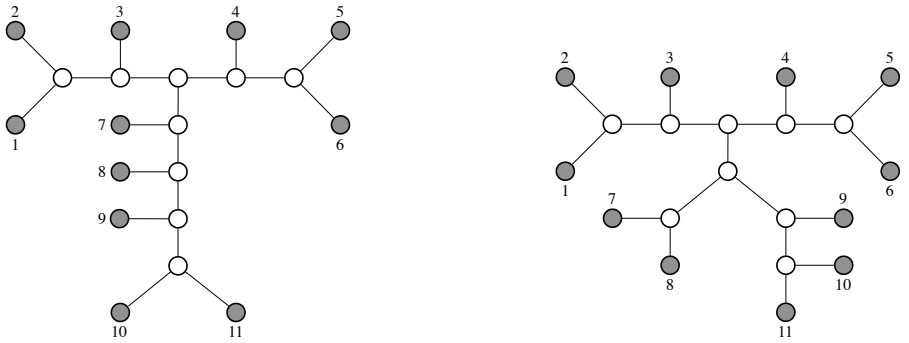


Figure 9: Two different phylogenies T (left) and T' (right) of 11 taxa.

Proof. The statement trivially follows by observing that the inequality $2^{-\tau_{ij}/\alpha} > 2^{-\tau_{ij}}$ holds true for any scalar $\alpha > 1$ and any pair of distinct $i, j \in \Gamma$. Hence, $L_\alpha(T) > L(T)$, for any phylogeny T of Γ. □

Proposition 8. *There exists an input distance matrix* **D** *for which the RMEP is not statistically consistent, for any* $\alpha \geq 1.2447$.

Proof. Assume $n = 11$ and consider the phylogenies T and T', shown in Figure 9, and the associated path-length matrices τ and τ', respectively, shown in Figure 10. Set $D = \tau'$. Then, it is possible to see numerically that $L_\alpha(T) < L_\alpha(T')$ for any $\alpha \geq 1.2447$. Now, D corresponds precisely to T', hence T' is the true phylogeny for this specific instance of the RMEP. Thus, for any $\alpha \geq 1.2447$, the problem of minimizing $L_\alpha(T)$ over the set of the phylogenies for Γ does not yield the desired true phylogeny. □

Proposition 8 still leaves some hope to find a scalar $1 < \alpha < 1.2447$ for which the RMEP could be statistically consistent. However, checking whether this hope is well-grounded is of little interest in practice mainly because, in the best case, this α (if it existed) would pull away the arising numerical instability (49) to an lower bound on n of

$$n \geq \left\lfloor 1.2446 \cdot \log_2 \left(\frac{1}{5 \cdot 10^{-34}} \right) + 1 \right\rfloor = 138, \tag{56}$$

a value that is not suited for large scale phylogenetic analyses involving many hundreds or thousands of taxa.

As a striking example of the implications of the above propositions, consider again the input distance matrix "Primates12". The optimal solutions to the BMEP and the RMEP for $\alpha = 2$ are shown as T and T' in Figure 11, respectively, and are unfortunately different.

$$\tau = \begin{bmatrix} 0 & 2 & 3 & 5 & 6 & 6 & 5 & 6 & 7 & 8 & 8 \\ 2 & 0 & 3 & 5 & 6 & 6 & 5 & 6 & 7 & 8 & 8 \\ 3 & 3 & 0 & 4 & 5 & 5 & 4 & 5 & 6 & 7 & 7 \\ 5 & 5 & 4 & 0 & 3 & 3 & 4 & 5 & 6 & 7 & 7 \\ 6 & 6 & 5 & 3 & 0 & 2 & 5 & 6 & 7 & 8 & 8 \\ 6 & 6 & 5 & 4 & 2 & 0 & 5 & 6 & 7 & 8 & 8 \\ 5 & 5 & 4 & 4 & 5 & 5 & 0 & 3 & 4 & 5 & 5 \\ 6 & 6 & 5 & 5 & 6 & 6 & 3 & 0 & 3 & 4 & 4 \\ 7 & 7 & 6 & 6 & 7 & 7 & 4 & 3 & 0 & 3 & 3 \\ 8 & 8 & 7 & 7 & 8 & 8 & 5 & 4 & 3 & 0 & 2 \\ 8 & 8 & 7 & 7 & 8 & 8 & 5 & 4 & 3 & 2 & 0 \end{bmatrix} \qquad \tau' = \begin{bmatrix} 0 & 2 & 3 & 5 & 6 & 6 & 6 & 6 & 6 & 7 & 7 \\ 2 & 0 & 3 & 5 & 6 & 6 & 6 & 6 & 6 & 7 & 7 \\ 3 & 3 & 0 & 4 & 5 & 5 & 5 & 5 & 5 & 6 & 6 \\ 5 & 5 & 4 & 0 & 3 & 3 & 5 & 5 & 5 & 6 & 6 \\ 6 & 6 & 5 & 3 & 0 & 2 & 6 & 6 & 6 & 7 & 7 \\ 6 & 6 & 5 & 3 & 2 & 0 & 6 & 6 & 6 & 7 & 7 \\ 6 & 6 & 5 & 5 & 6 & 6 & 0 & 2 & 4 & 5 & 5 \\ 6 & 6 & 5 & 5 & 6 & 6 & 2 & 0 & 4 & 5 & 5 \\ 6 & 6 & 5 & 5 & 6 & 6 & 4 & 4 & 0 & 3 & 3 \\ 7 & 7 & 6 & 6 & 7 & 7 & 5 & 5 & 3 & 0 & 2 \\ 7 & 7 & 6 & 6 & 7 & 7 & 5 & 5 & 3 & 2 & 0 \end{bmatrix}$$

Figure 10: Two path-length matrices τ and τ' encoding the two distinct phylogenies T and T' of Figure 9, respectively.

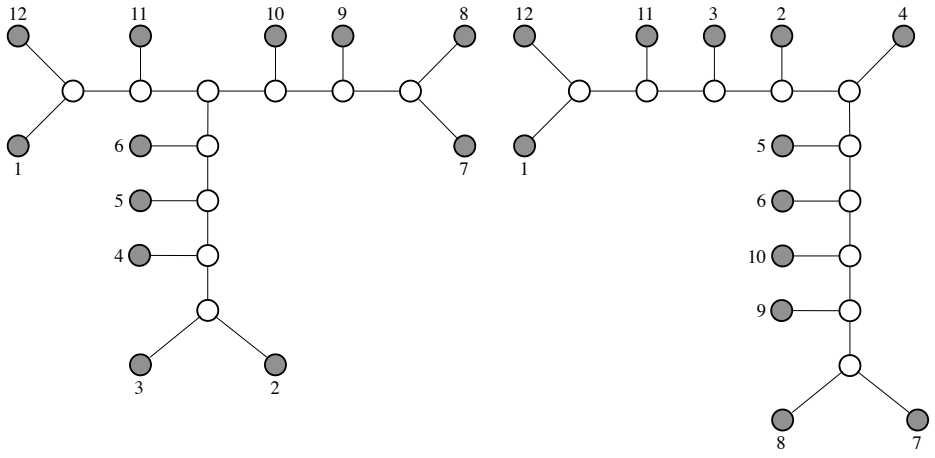


Figure 11: The optimal solutions T (left) and T' (right) to the BMEP and the RMEP for $\alpha = 2$, respectively, for the instance "Primates12" of 12 taxa.

9. Concluding remarks

The recent COVID19 pandemic highlighted the need to equip epidemiologists and, more in general, life sciences researchers and practitioners with estimation models and algorithms able to track evolution of pathogens and, more in general, of taxa in the most reliable and accurate way possible [12]. The optimal solution to the BMEP may constitute an answer to these needs, thanks to its particular mathematical and statistical properties (see [12, 38, 40]). Developing models and algorithms able to exactly solve instances of the BMEP is, therefore, highly desirable in practical applications. In this article, we built upon recent theoretical advances on the combinatorics, information theory, and optimization aspects of the BMEP to design a new massively parallel exact solution algorithm that proves to be up to one order of magnitude faster than the current state-of-the-art and able to solve up to 25% bigger BMEP instances within a prefixed time limit. We have also investigated here the connections between numerical stability and statistical consistency of the BMEP, by showing that some rescaling techniques introduced to numerically stabilize the problem may unfortunately affect the statistical consistency of its optimal solution. The scientific community is therefore called to investigate not only ways to further increase the size of tractable BMEP instances but also ways to overcome numerical issues without altering the statistical consistency of its optimal solution.

Acknowledgments

The first author acknowledges support from the Belgian National Fund for Scientific Research (FNRS) via the grant FNRS PDR 40007831, and the Fondation Louvain via the grant "COALESCENS". The second author acknowledges support from the European Union's Horizon 2020 research and innovation programme under the Marie Skłodowska-Curie grant agreement no. 101034253, and by the NWO Gravitation project NETWORKS under grant no. 024.002.003. Computational resources have been provided by the supercomputing facilities of the Université catholique de Louvain (CISM/UCL) and the Consortium des Équipements de Calcul Intensif en Fédération Wallonie Bruxelles (CÉCI) funded by the Fond de la Recherche Scientifique de Belgique (F.R.S.-FNRS) under convention 2.5020.11 and by the Walloon Region.

References

[1] Allen, B.L., Steel, M., 2001. Subtree transfer operations and their induced metrics on evolutionary trees. Annal of Combinatorics 5, 1–15.

[2] Aringhieri, R., Catanzaro, D., Di Summa, M., 2011. Optimal solutions for the balanced minimum evolution problem. Computers and Operations Research 38, 1845–1854.

[3] Auch, A.F., R., H.S., Holland, B.R., M., G., 2006. Genome BLAST distance phylogenies inferred from whole plastid and whole mitochondrion genome sequences. BMC Bioinformatics 7, 1–5.

[4] Baldisserri, A., 2014. Buneman's theorem for trees with exactly n vertices. CoRR .

[5] Batagelj, V., Pisanski, T., Simoes-Pereira, J.M., 1990. An algorithm for tree-realizability of distance matrices. International Journal of Computer Mathematics 34.

[6] Bordewich, M., Gascuel, O., Huber, K., Moulton, V., 2009a. Consistency of topological moves based on the balanced minimum evolution principle of phylogenetic inference. IEEE/ACM Transactions on Computational Biology and Bioinformatics 6, 110–117.

[7] Bordewich, M., Gascuel, O., Huber, K.T., Moulton, V., 2009b. Consistency of topological moves based on the balanced minimum evolution principle of phylogenetic inference. IEEE/ACM Transactions on Computational Biology and Bioinformatics 6, 110–117.
[8] Buneman, P., 1971. The recovery of trees from measure of dissimilarities, in: Hodson, F.R., Kendall, D.G., Tautu, P. (Eds.), Archaeological and Historical Science. Edinburgh University Press, Edinburgh, UK, pp. 387–395.
[9] Buneman, P., 1974. A note on the metric properties of trees. Journal of combinatorial theory 17, 48–50.
[10] Caminiti, S., Finocchi, I., Petreschi, R., 2007. On coding labeled trees. Theoretical Computer Science 382, 97–108.
[11] Catanzaro, D., Aringhieri, R., di Summa, M., Pesenti, R., 2015. A branch-price-and-cut algorithm for the minimum evolution problem. European Journal of Operational Research 244, 753–765.
[12] Catanzaro, D., Frohn, M., Gascuel, O., Pesenti, R., 2022. A tutorial on the balanced minimum evolution. European Journal of Operational Research 300, 1–19.
[13] Catanzaro, D., Frohn, M., Pesenti, R., 2020a. An information theory perspective on the balanced minimum evolution problem. Operations Research Letters 48, 362–367.
[14] Catanzaro, D., Labbé, M., Pesenti, R., Salazar-Gonzáles, J.J., 2012. The balanced minimum evolution problem. INFORMS Journal on Computing 24, 276–294.
[15] Catanzaro, D., Pesenti, R., 2019. Enumerating vertices of the balanced minimum evolution polytope. Computers and Operations Research 109, 209–217.
[16] Catanzaro, D., Pesenti, R., Milinkovitch, M., 2006. A non-linear optimization procedure to estimate distances and instantaneous substitution rate matrices under the GTR model. Bioinformatics 22, 708–715.
[17] Catanzaro, D., Pesenti, R., Wolsey, L.A., 2020b. On the Balanced Minimum Evolution polytope. Discrete Optimization 36, 1–33.
[18] Çela, E., 1997. The Quadratic Assignment Problem. Kluwer Academic Publishers, Boston, MA.
[19] Chandra, R., Dagum, L., Kohr, D., Menon, R., Maydan, D., McDonald, J., 2001. Parallel programming in OpenMP. Morgan Kaufmann.
[20] Cheng, X., Du, D.Z., 2001. Steiner Trees in Industry. Kluwer Academic Publishers, Boston, MA.
[21] Cieslik, D., 1998. Steiner minimal trees. Springer, Boston, MA, USA.
[22] Criscuolo, A., Michel, C.J., 2009. Phylogenetic inference with weighted codon evolutionary distances. Journal of Molecular Evolution 68, 377–392.
[23] Cueto, M.A., Matsen, F.A., 2011. Polyhedral geometry of phylogenetic rogue taxa. Bulletin of Mathematical Biology 73, 1202–1226.
[24] Desper, R., Gascuel, O., 2002. Fast and accurate phylogeny reconstruction algorithms based on the minimum evolution principle. Journal of Computational Biology 9, 687–705.
[25] Desper, R., Gascuel, O., 2004. Theoretical foundations of the balanced minimum evolution method of phylogenetic inference and its relationship to the weighted least-squares tree fitting. Molecular Biology and Evolution 21, 587–598.
[26] Desper, R., Gascuel, O., 2008. Encyclopedia of Algorithms. Springer. chapter Distance-based Phylogeny Reconstruction (Optimal Radius). pp. 1–99.
[27] Du, D.Z., Hu, X., 2008. Steiner tree problems in computer communication networks. World Scientific Publishing Company, Singapore.
[28] Du, D.Z., Smith, J.M., Rubinstein, J.H., 2000. Advances in Steiner trees. Kluwer Academic Publishers, Boston, MA, USA.
[29] Erdös, P.L., Steel, M.A., Székély, L.A., Warnow, T., 1999. A few logs suffice to build (almost) all trees: Part I. Random Structures and Algorithms 14, 153–184.
[30] Felsenstein, J., 1984. Distance methods for inferring phylogenies: A justification. Evolution 38, 16–24.
[31] Felsenstein, J., 2004. Inferring phylogenies. Sinauer Associates, Sunderland, MA.
[32] Fiorini, S., Joret, G., 2012. Approximating the balanced minimum evolution problem. Operations Research Letters 40, 31–35.
[33] Forcey, S., Keefe, L., Sands, W., 2015. Facets of the balanced minimal evolution polytope. Journal of Mathematical Biology 73, 447–468.
[34] Forcey, S., Keefe, L., Sands, W., 2017. Split-facets for balanced minimal evolution polytopes and the permutoassociahedron. Bulletin of Mathematical Biology, in press 79, 975–994.
[35] Frohn, M., 2020. On the approximability of the fixed-tree balanced minimum evolution problem. To appear in Optimization Letters .
[36] Gascuel, O., 1994. A note on Sattath and Tversky's, Saitou and Nei's and Studier and Keppler's algorithms for inferring phylogenies from evolutionary distances. Molecular Biology and Evolution 11, 961–961.
[37] Gascuel, O., 1997. BIONJ: An improved version of the NJ algorithm based on a simple model of sequence data. Molecular Biology and Evolution 14, 685–695.
[38] Gascuel, O., 2005. Mathematics of evolution and phylogeny. Oxford University Press, New York, NY.
[39] Gascuel, O., Steel, M., 2016. A 'stochastic safety radius' for distance-based tree reconstruction. Algorithmica 74, 1386–1403.
[40] Gascuel, O., Steel, M.A., 2006. Neighbor-joining revealed. Molecular Biology and Evolution 23, 1997–2000.
[41] Gusfield, D., 1984. The Steiner Tree Problem in Phylogeny. Technical Report 334. Yale University, New Haven, CT.
[42] Haws, D.C., Hodge, T.L., Yoshida, R., 2011. Optimality of the neighbor joining algorithm and faces of the balanced minimum evolution polytope. Bulletin of Mathematical Biology 73, 2627–2648.
[43] Hendy, M.D., Penny, D., 1982. Branch and bound algorithms to determine minimal evolutionary trees. Mathematical Biosciences .
[44] Hwang, F.K., Richards, D.S., Winter, P., 1992. The Steiner tree problem. North-Holland, Amsterdam, The Netherlands.
[45] Johnson, D.S., Lenstra, J.K., Kan, A.H.G.R., 1978. The complexity of the network design problem. Networks 8, 279–285.
[46] Jung, M., 2012. Évolution du VIH: méthodes, modèles et algorithmes. Ph.D. thesis. Université Montpellier II - Sciences et Techniques du Languedoc, France.
[47] Kreher, D.L., Stinson, D.R., 1999. Combinatorial algorithms: Generation, enumeration, and search. CRC Press, Boca Raton, FL.
[48] Le, S.Q., Gascuel, O., 2008. An improved general amino acid replacement matrix. Molecular biology and evolution 25, 1307–1320.
[49] Lefort, V., Desper, R., Gascuel, O., 2015. FastME 2.0: A comprehensive, accurate, and fast distance-based phylogeny inference program. Molecular Biology and Evolution 32, 2798–2800.
[50] Li, X., Mao, Y., 2016. Generalized connectivity of graphs. Springer, Boston, MA, USA.

[51] Lu, C.L., Tang, C.Y., Lee, R.C.T., 2003. The full Steiner tree problem. Theoretical Computer Science 306, 55–67.
[52] Page, R.D.M., Holmes, E.C., 1998. Molecular Evolution: A Phylogenetic Approach. Blackwell Science, Oxford, UK.
[53] Pardi, F., 2009. Algorithms on Phylogenetic Trees. Ph.D. thesis. University of Cambridge, UK.
[54] Pardi, F., Gascuel, O., 2012. Combinatorics of distance-based tree inference. PNAS 109, 16443–16448.
[55] Pardi, F., Gascuel, O., 2016. Encyclopedia of Evolutionary Biology. Elsevier. chapter Distance-based methods in phylogenetics. pp. 458–465.
[56] Pardi, F., Guillemot, S., Gascuel, O., 2010. Robustness of phylogenetic inference based on minimum evolution. Bulletin of Mathematical Biology 72, 1820–1839.
[57] Parker, D.S., Ram, P., 1996. The construction of Huffman codes is a submodular ("convex") optimization problem over a lattice of binary trees. SIAM Journal on Computing 28, 1875–1905.
[58] Pauplin, Y., 2000. Direct calculation of a tree length using a distance matrix. Journal of Molecular Evolution 51, 41–47.
[59] Pop, P.C., 2012. Generalized network design problems: Modeling and optimization. De Gruyter, Berlin, Germany.
[60] Prömel, H.J., Steger, A., 2002. The Steiner tree problem: A tour through graphs, algorithms, and complexity. Vieweg+Teubner Verlag, Berlin.
[61] Reinhard, D., 2005. Graph Theory. Springer-Verlag, New York, NY.
[62] Saitou, N., Nei, M., 1987. The neighbor-joining method: A new method for reconstructing phylogenetic trees. Molecular Biology and Evolution 4, 406–425.
[63] Semple, C., Steel, M.A., 2004. Cyclic permutations and evolutionary trees. Advances in Applied Mathematics 32, 669–680.
[64] Steel, M., 2005. Phylogenetic diversity and the greedy algorithm. Systematic Biology 54, 527–529.
[65] Studier, J.A., Keppler, K.J., 1988. A note on the neighbor-joining algorithm of Saitou and Nei. Molecular Biology and Evolution 5, 729–731.
[66] Waterman, M.S., Smith, T.F., Singh, M., Beyer, W.A., 1977. Additive evolutionary trees. Journal of Theoretical Biology 64, 199–213.
[67] Wu, B.Y., Chao, K.M., 2004. Spanning trees and optimization problems. Chapman and Hall/CRC, Boca Raton, FL.
[68] Yang, Z., 2014. Molecular Evolution: A Statistical Approach. Oxford University Press, Oxford, UK.