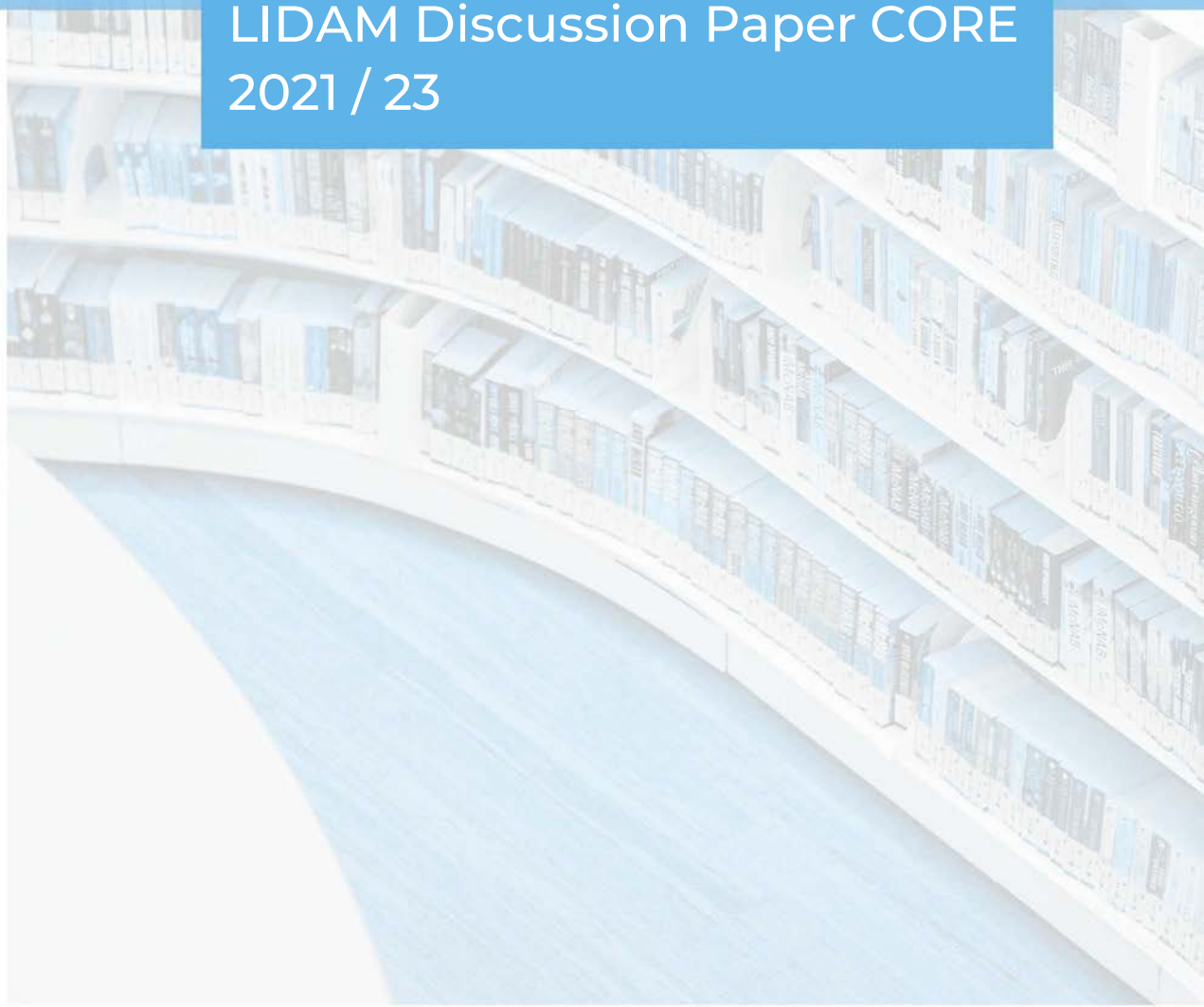
A MASSIVELY PARALLEL EXACT SOLUTION ALGORITHM FOR THE BALANCED MINIMUM EVOLUTION PROBLEM

Daniele Catanzaro, Martin Frohn, Raffaele Pesenti

LIDAM Discussion Paper CORE
2021 / 23



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[a,2] and Raffaele Pesenti[c,3]

[a]*Université Catholique de Louvain, Voie du Roman Pays 34, L1.03.01, B-1348 Louvain-la-Neuve, Belgium.*
[c]*Department of Management, Università Ca' Foscari, San Giobbe, Cannaregio 837, I-30121 Venezia, Italy.*

ARTICLE INFO

Keywords:
Combinatorial optimization; network design; balanced minimum evolution; implicit enumeration algorithms; parallel computing; numerical stability;

ABSTRACT

The *Balanced Minimum Evolution Problem* (BMEP) is an $\mathcal{APX}$-hard nonlinear network design problem that consists of finding a phylogeny that minimizes the cross-entropy of the molecular sequences extracted from a given set of taxa. By combining massive parallelism with recent theoretical advances on the polyhedral combinatorics of the problem and new insights on the relationships between the BMEP and information entropy, we design a new exact solution algorithm that proves to be up to an order of magnitude faster than the current state-of-the-art sequential-version on generic instances and able to solve up to 25% more taxa within the same time limit. We also investigate some issues related to numerical stability and statistical consistency of the BMEP, arising in particular when dealing with large instances. We show, as a negative finding, that no rescaling technique may ensure numerical stability, by guaranteeing at the same time 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 (e.g., DNA, RNA, codon sequences, or whole genomes), hereafter referred to as *taxa* (see, e.g., Figure 1). A phylogeny of Γ is an ordered triplet $(U, \phi, \mathbf{w})$ such that U is an *Unrooted Binary Tree* (UBT) (also known as *cubic tree*) having n leaves, ϕ is a bijection between the leaves of U and the taxa in Γ, and $\mathbf{w}$ is a vector of weights associated to the edges of U [9]. For example, Figure 2 shows a possible phylogeny for the set of taxa shown in Figure 1. Now, consider 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 between the molecular sequences of taxa i and j in Γ (see again Figure 1). Then, the *Balanced Minimum Evolution Problem* (BMEP) 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, w_e is the weight associated to the edge $e \in E(T)$, 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 [11, 48] (see Figure 2).

The BMEP is a nonlinear *Network Design Problem* [41, 54] defined over the class of the UBTs. It was introduced in the literature on molecular phylogenetics by Desper and Gascuel [21], based on a particular edge-weight estimation criterion proposed by Pauplin [53] about 21 years ago (see [9] for a historical review of the problem). The optimal solution T^* to a given instance of the BMEP minimizes the *cross entropy* associated to the considered taxa [10] and provides a *statistically consistent estimate* of their evolutionary relationships. In other words, as long as the dissimilarity measure used to compute the entries of the matrix $\mathbf{D}$ provides consistent estimates of the *true evolutionary distances* from taxa, T^* provably converges to the *true phylogeny* of Γ as long as the sequence lengths of taxa diverges to infinity [9, 27, 33, 35]. This highly desirable property, together with a computational burden lower than other phylogenetic estimation models such as *maximum likelihood* or *Bayesian inference* (see [7, 9]) as well as the ability to

*Corresponding author
ORCID(s): 0000-0001-9427-1562 (D. Catanzaro); 0000-0002-5002-4049 (M. Frohn); 0000-0001-5890-4238 (R. Pesenti)
[1]Email: daniele.catanzaro@uclouvain.be (Daniele Catanzaro)
[2]Email: martin.frohn@uclouvain.be (Martin Frohn)
[3]Email: pesenti@unive.it (Raffaele Pesenti)

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

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

Figure 1: A dummy example of a set Γ of five DNA sequences (*taxa*) and the corresponding distance matrix $\mathbf{D}$. The generic entry d_{ij} of $\mathbf{D}$ encodes the Hamming distance between the sequences associated to taxa i and j, respectively.

tackle virtually any kind of data for which a dissimilarity measure is available, make the BMEP particularly suitable for general-purpose phylogenetic analyses [58].

A distinctive aspect that characterizes the BMEP in the context of network design is the particular form of the length function (i.e., the sum of the edge weights of a phylogeny), which depends just on the input evolutionary distances d_{ij} and on the unknown path-lengths τ_{ij} encoding the *topology* (i.e., the structure) of T. The weights w_e associated to the edges of a phylogeny T can be implicitly computed by means of specific formulae (see, e.g., [9]) once that the topology of T has been identified. We observe that, besides being a network design problem, the BMEP can also be seen as: (i) a restriction of the *Minimum Evolution Problem* discussed in [8]; (ii) a version of the *Steiner tree problem* in which the number of Steiner nodes is known a priori, the tree must be unrooted an binary, and the length function depends upon the topology of the tree [17, 18, 24, 25, 37, 40, 45, 46, 55, 62]; and (iii) a generalization of the *Quadratic Assignment Problem* [16], in which one of the two matrices involved in the Shur product must be determined within the set of matrices $\{\tau_{ij}\}$ that encode UBTs [1, 31]. These alternative perspectives hide the presence of strong connections between the BMEP and the mentioned problems, some of which have been explored just recently in [8, 31].

The BMEP is proven to be in general $\mathcal{NP}$-hard and inapproximable within c^n, for some positive constant $c > 1$, unless $\mathcal{P} = \mathcal{NP}$ [28]. 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 [33]:

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

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 [28].

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 [22, 33, 35], its connections with information theory [10], its combinatorics [11, 14, 20, 29, 30, 38, 59], as well as the development of implicit enumeration algorithms [1, 11, 48] and heuristics [28, 33, 48] to tackle and solve instances of practical size. The reader interested in an introduction to the problem and on a review of the most important recent advances may refers to [9]. This article further adds to the literature, by developing a new massively parallel exact solution algorithm that redefines the performance reference for the BMEP.

The earliest exact solution algorithm described in the literature was proposed by Pardi [48] in 2009. The author derived a number of lower bounds on the value of T^* based on Semple and Steel [59]'s combinatorial interpretation of

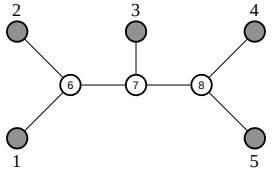


Figure 2: An example of a phylogeny T for the set of five taxa shown in Figure 1. Each internal vertex of T has degree three (i.e., T is unrooted and binary) and each taxon in Γ is assigned precisely to a leaf of T. The topological distance between the pair of taxa 1 and 3, denoted as τ_{13}, is equal to four. Similarly, the topological distance between the pair of taxa 2 and 3, denoted as τ_{23}, is equal to three. The phylogeny T is provably optimal for the distance matrix D shown in Figure 1.

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. The interested reader may however find more information at `https://oeis.org/A000672`, accessible by clicking on the hyperlink A000672 provided in the table.

the length function (1) that, 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 [11].

In 2011, Aringhieri et al. [1] proposed an alternative exact solution algorithm exploiting *tree isomorphism* among phylogenies [43] to explore the solution space of the problem. In particular, the authors observed that groups of phylogenies whose underlying UBTs are isomorphic identify specific classes of equivalence, called *unlabeled UBTs* (see Figure 3). Hence, they proposed to enumerate just unlabeled UBTs and then finding, for each of them, the best possible assignment of taxa to the leaves of the unlabeled UBT so as to minimize (1). Aringhieri et al.' algorithm is characterized by two fundamental strengths: first, the number of unlabeled UBTs grows exponentially with n, but in a way much slower than the corresponding number of phylogenies of Γ (see, e.g., Table 1); second, the minimization of the (quadratic) assignment of taxa to the leaves of each unlabeled UBT is a $\mathcal{NP}$-hard problem that can be processed and solved in parallel [1, 31]. These characteristics allowed Aringhieri et al. [1] to solve instances of the BMEP larger than Pardi [48] within the same amount of time.

Although the enumeration of unlabeled UBTs can be done off line and is quite fast for small value of n, their generation and storage becomes unpractical for $n \geq 27$. Hence, a third implicit enumeration-based exact solution algorithm for the BMEP was proposed proposed by Catanzaro et al. [11] in 2012. The theoretical foundations of this algorithm lay in the deep connections between the BMEP, the *Huffman Coding Problem* (HPC) [52], and information theory [10] (see the next sections). These connections allowed the authors to identify a number of fundamental equalities characterizing UBTs that appropriately included in specific integer programming formulations for the BMEP allow to dramatically increase the lower bound on its optimal solution. 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 and Aringhieri et al. [1]'s algorithms, by solving instances of the BMEP containing up to 26 taxa within 1h computing time. This performance makes Catanzaro et al. [11]'s approach 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 taxa (see, e.g., SARS-CoV-2 genomes available at `https://www.ncbi.nlm.nih.gov/genbank/sars-cov-2-seqs/` and [2, 3, 4, 19, 23, 35, 42, 44, 48, 49, 50, 51, 63]), it is highly desirable to design exact solution algorithms able to tackle and solve instances much

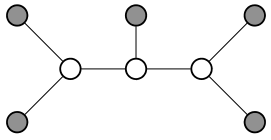


Figure 3: An example of an unlabeled UBT with five leaves. The phylogeny obtained from the one shown in Figure 2 by permuting the assignment of taxa to the leaves of the underlying UBT are all isomorphic to the unlabeled UBT shown in this figure.

larger than Catanzaro et al. [11]'s algorithm. In this article we make a further step in this direction, by building upon the results described in [1, 8, 10, 11, 12, 14, 34, 35, 35, 59] 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 [12] to enumerate by brute-force all of the vertices of the BMEP polyhedron, by enabling so the analysis of its polyhedral combinatorics discussed in [14]. In that work, we encoded phylogenies as Prüfer codes [6] and we showed algorithms and methods to parallelize this enumeration both on CPUs and GPUs. We will show in this article that the combination of massive parallelism with new theoretical results on the polyhedral combinatorics of the BMEP [14] and on its relationships with information theory [10] allow to redefine a new reference in terms of performance: up to an order of magnitude faster than the current state-of-the-art exact sequential algorithm on generic instances of the BMEP and up to 25% more taxa solved within the same time limit. We will also see that there exists a theoretical limit on the performance achievable by an exact algorithm for the BMEP, which materializes whenever the generic addend $d_{ij}2^{-\tau_{ij}}$ in the length function approaches (or get smaller than) the machine epsilon. The study of the numerical stability aspect remained marginal so in literature on the BMEP; however, it has deep implications on the statistical consistency of the optimal solution T^*. In particular, we will see that no rescaling technique may guarantee at the same time the numerical stability of the problem and the statistical consistency of the optimal solution. Before proceeding, we introduce in the next section 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 3, we briefly recall, for the sake of completeness, the current state-of-the-art sequential exact solution algorithm for the BMEP. In Section 6, we discuss lower bounds and upper bounds on the optimal solution to the problem that will prove useful for the development of the new massively parallel exact solution algorithm. In Section 7, we close with a discussion of the practical performance of our parallel method and numerical stability as well as statistical consistency issues connected to it.[4]

2. Notation and properties of the topological distances

In this section, we introduce some notation and nomenclature of phylogenetics that will prove useful throughout the article. We will give particular emphasis to the concept and the properties of a "path-length sequence" of a phylogeny because it has a central role in the combinatorics of the BMEP [14]. Before proceeding, we briefly recall the fact that, from a topological point of view, an UBT T with n leaves is characterized by $2n-2$ 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, classical results from graph theory (see [56]) state that

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

Moreover, because internal vertices have degree three, it also holds 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 internal vertices and the set of edge of T respectively. Moreover, for a given subset of taxa $S \subseteq \Gamma$, we define a *sub-phylogeny* T_S of Γ as a phylogeny that involves just the taxa in S, i.e., 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 $\alpha < \beta$, let $[\alpha, \beta]$ denote the discrete interval constituted by the integers included between α and β. Similarly to Parker and Ram [52], we define a *sequence* as a collection of nonnegative real values such as $\mathbf{s} = [s_1, s_2, \dots, s_m]$, $s_j \in \mathbb{R}_{0^+}$. Repetition of values in the sequence is permitted: the values s_j need not to be distinct.

Consider 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 of the topological distances relative to the $n-1$ (unique) paths in T from taxon i to each taxon $j \in \Gamma_i$. For example, consider the phylogeny showed in Figure 2. Then,

[4] Where do you describe the parallel algorithm? Where do you present computational experiments? This part should be updated consequently.

$\tau_1 = [2, 3, 4, 4]$ and $\tau_4 = [4, 4, 3, 2]$. It is worth noting that the path-length sequence τ_i associated to a phylogeny T of Γ describes the UBT 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. Fixing 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 concept of a path-length sequence proves particularly useful to model the BMEP: we define the *path-length matrix* τ associated to a phylogeny T of Γ as a $n \times n$ integer matrix having as generic entry τ_{ij}, for all $i, j \in \Gamma$. For example, the following path-length matrix is associated to the phylogeny shown in Figure 2 under the assumption that the relative taxa are ordered according to their labels:

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

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. In analogy to Θ_i, we denote Θ as the set of the path-length matrices τ associated to all the possible phylogenies of Γ. Moreover, for a fixed $\tau \in \Theta$ we define $2^{-\tau}$ as the matrix obtained from τ by setting its non-diagonal entries to $2^{-\tau_{ij}}$ and we set $2^{-\Theta} = \{2^{-\tau} : \tau \in \Theta\}$. A question that naturally arises and that proves useful to study the polyhedral combinatorics of the BMEP is whether it is possible to characterize Θ and the sets Θ_i, for all $i \in \Gamma$. The following two sections address this issue.

2.1. Characterizing Θ_i

As observed in Catanzaro et al. [11] and [14], 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* [52]. 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 [52]) Consider a set Ψ of n symbols. Then, $\rho = [\rho_j, \rho_2, \dots, \rho_n]$ 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 2, 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 [11]) 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,

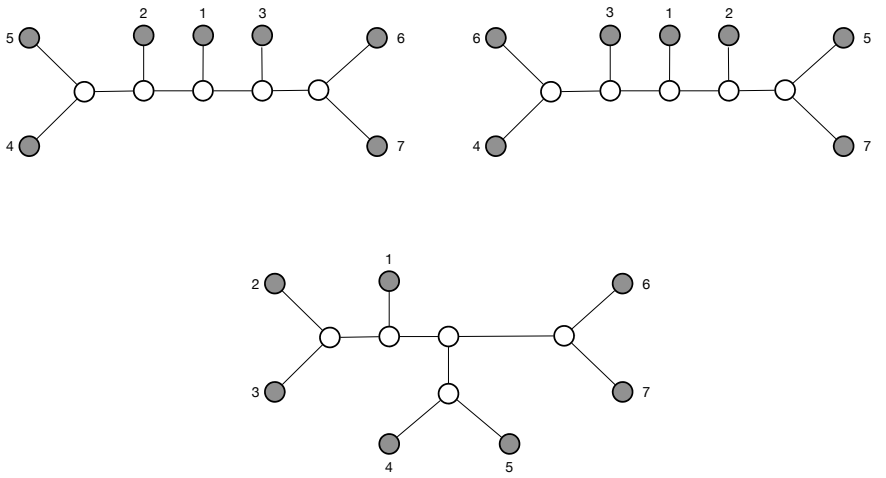


Figure 4: An example of the three distinct phylogenies that can be reconstructed from the path-length sequence $\tau_1 = [3, 3, 4, 4, 4, 4]$.

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 4 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 has a central role in the context of solving exactly the BMEP. Although this task still remains incomplete, we know a number of necessary conditions that any path-length matrix τ must satisfy in order to belong to Θ. For example, because phylogenies are (non-oriented) acyclic graphs (trees), 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 *triangular inequality*. Now, because the tree encoded by any path-length matrix $\tau \in \Theta$ 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 [52]. This insight has been exploited in Catanzaro et al. [10, 11, 14] to identify a further condition that the entries of any path-length matrix 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* $2^{-\tau_{ij}}$ [52], and encodes the geometric locus of the matrices $2^{-\tau}$ having constant cross entropy (see Catanzaro et al. [10] for more information).

It is easy to see that conditions (7)-(11) are independent. As an example, the following minimal-dimension matrix

$$\begin{pmatrix} 0 & 3 & 3 & 3 & 3 \\ 3 & 0 & 3 & 3 & 3 \\ 3 & 3 & 0 & 3 & 3 \\ 3 & 3 & 3 & 0 & 3 \\ 3 & 3 & 3 & 3 & 0 \end{pmatrix}$$

satisfies all the considered conditions but (11). In fact,

$$\sum_{i \in \Gamma} \sum_{j \in \Gamma_i} \tau_{ij} 2^{-\tau_{ij}} = \frac{15}{2} \neq 7 = (2n - 3).$$

Similarly, the following matrix

$$\begin{pmatrix} 0 & 5 & 5 & 5 & 4 & 5 \\ 5 & 0 & 2 & 2 & 3 & 4 \\ 5 & 2 & 0 & 2 & 3 & 4 \\ 5 & 2 & 2 & 0 & 3 & 4 \\ 4 & 3 & 3 & 3 & 0 & 4 \\ 5 & 4 & 4 & 4 & 4 & 0 \end{pmatrix}$$

satisfies all the considered conditions but (10). In fact, we have that

$$\sum_{j \in \Gamma_1} 2^{-\tau_{1j}} = 0.1875 \neq \frac{1}{2}.$$

Furthermore, the following minimal-dimension matrix

$$\begin{pmatrix} 0 & 3 & 5 & 5 & 2 & 4 \\ 3 & 0 & 3 & 4 & 4 & 3 \\ 5 & 3 & 0 & 2 & 5 & 4 \\ 5 & 4 & 2 & 0 & 5 & 3 \\ 2 & 4 & 5 & 5 & 0 & 3 \\ 4 & 3 & 4 & 3 & 3 & 0 \end{pmatrix}$$

satisfies all of the considered conditions but (9). In fact, $\tau_{12} + \tau_{12} - \tau_{13} = 1 < 2$. It is worth noting that because any path-length matrix $\tau \in \Theta$ must encode a tree, the triangular inequalities (9) can be further generalized by means of Buneman's *additive property* [5, 26, 61], 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:

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

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., $p = q$). Now, conditions (6)-(8) and (10)-(12) are necessary to characterize Θ, but proving (or disproving) the following conjecture still remains an open problem:

Conjecture 1. *Are conditions* (6)-(8) *and* (10)-(12) *sufficient to characterize* Θ.

Addressing this conjecture implies proving (or disproving) whether Buneman's additive property restricted just to quartets of taxa (rather than to the whole vertex set of a phylogeny) may imply the acyclicity of the combinatorial structure associated to the collection of paths encoded by τ. If these conditions should prove sufficient, it would be possible to develop compact mixed integer programming formulations for the BMEP. Alternatively, if these conditions should prove insufficient, there could be still room for the identification of fundamental properties describing collections of path-lengths of phylogenies.

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 quite delicate, as a wrong choice of the space may impact negatively the solution times of a formulation. The space of path-lengths τ_{ij} looks a natural choice for the BMEP; 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. [29] and Catanzaro et al. [14], 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.

Denoted $\mathcal{L} = \{2, \dots, n-1\}$ as the set of values taken by the generic path-length τ_{ij}, Catanzaro et al. [11] proposed to *discretize* path-lengths, i.e., 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 connecting each distinct pair of taxa $i, j \in \Gamma$ in any feasible solution to the problem, in this space the objective function of the BMEP 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

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

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

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

The modeling of Buneman's conditions (12) restricted to Γ can be obtained by introducing the following binary decision variables

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

and by imposing the following constraints:

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

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

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 Conjecture 1 holds true or not, i.e., 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. [11] 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 redefined y_{ijqt} variables as

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

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

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

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.

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

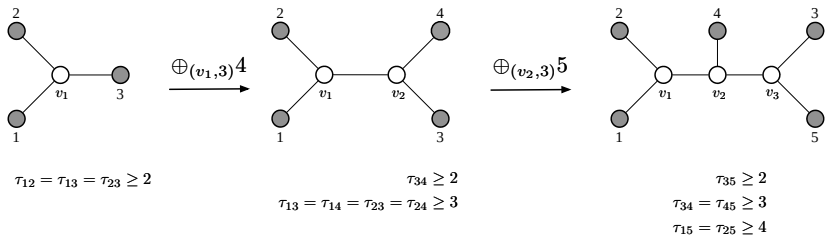


Figure 5: An example of two consecutive insertions on a sub-phylogeny of three taxa.

In particular, 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.

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. [11] 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 ensuring the potential enumeration of all of the possible phylogenies of the given set Γ of taxa. These rules are recursive in nature and based on the *Stepwise Addition Strategy* (SAS), first formalized by Hendy and Penny [39] in 1982. This strategy allows to enumerate all of the possible phylogenies of Γ by means the idea of *insertion* of a taxon on a sub-phylogeny. Specifically, given a subset of taxa $S \subset \Gamma$, a sub-phylogeny T_S of Γ, a taxon $t \in \Gamma \setminus S$, an internal vertex $v \in I \setminus V_l(T_S)$, 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

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

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 5. Specifically, 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 on an insertion of a taxon t on a sub-phylogeny $T_{S\cup\{t\}}$ is referred to as *removal* and is defined by

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

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 allow to enumerate the phylogenies of Γ, e.g., by means of Algorithm 1. A graphical example of its execution is shown in Figure 6.

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 T_S between, it holds that

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

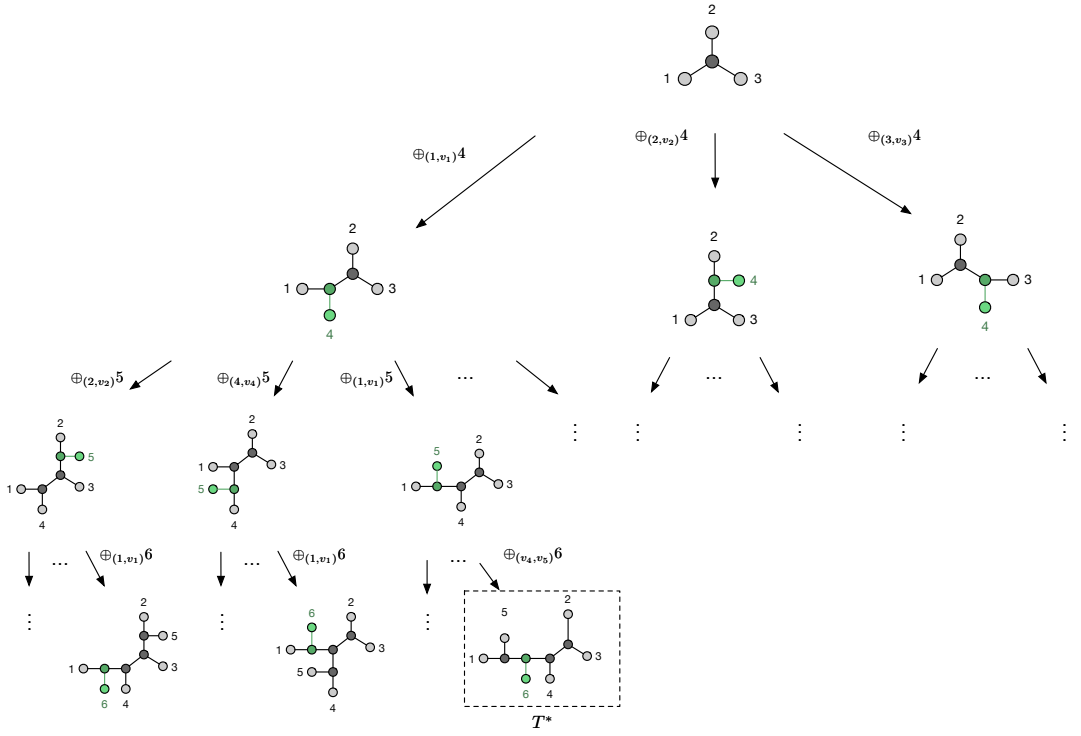


Figure 6: An example of the enumeration tree of a SAS for a set Γ of six taxa.

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 5, we have that $x_{13}^3 = x_{23}^3 = x_{34}^2 = x_{15}^2 = x_{15}^3 = x_{25}^2 = x_{25}^3 = x_{45}^2 = 0$.

4. Lower bounds on the optimal solution to the BMEP

The first lower bounds on the optimal solution to the BMEP were developed by Pardi [48]. The author explored the combinatorial structure of the insertion operation we introduced in the last section. To this end, he investigated Pauplin [53]'s formula to estimate the length of an external edge $e = (v, t)$ that is created by the insertion of taxon t into an edge (a, b) of a sub-phylogeny T_S. Specifically, given the following recursive formula, for taxa subsets $\Gamma_j = \Gamma_p \cup \Gamma_q$,

$$\delta^T_{\Gamma_i|\Gamma_j} = \begin{cases} d_{ij} & \text{if } \Gamma_i = \{i\} \text{ and } \Gamma_j = \{j\}; \\ \frac{1}{2}\delta^T_{\Gamma_i|\Gamma_p} + \frac{1}{2}\delta^T_{\Gamma_i|\Gamma_q} & \text{otherwise} \end{cases}$$

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 $|V_e(T_S)| = |\Gamma|$ **then**
  return;
$t \leftarrow |V_e(T_S)| + 1$;
for *each edge* $e \in E(T_S)$ **do**
  Enumerate($\Gamma, S, T_S \oplus_e t$);

Pauplin [53] showed that the length of edge (v, t) can be estimated, for $\Gamma = \Gamma_1 \cup \Gamma_2$, as

$$\frac{1}{2}\left(\delta^T_{\Gamma_1|\{t\}} + \delta^T_{\Gamma_2|\{t\}} - \delta^T_{\Gamma_1|\Gamma_2}\right). \tag{36}$$

Then, using expression (36), the difference

$$L\left(T_{S\cup\{t\}}\right) - L\left(T_S\right) = L\left(T_S \oplus_{(a,b)} t\right) - L\left(T_S\right) \tag{37}$$

can be restated as a weighted average, for $i \in \Gamma_1$ and $j \in \Gamma_2$, of the terms

$$\lambda^{ij}_t = \frac{1}{2}\left(d_{it} + d_{jt} - d_{ij}\right).$$

In light of these results, Pardi [48] proved the validity of the following lower bounds:

Proposition 3. *Let T_S be a sub-phylogeny with $S = \{1, 2, \dots, t-1\} \subseteq \Gamma$ and let λ_t be the vector $(\lambda^{ij}_t)_{1\le i<j<t}$ in non-decreasing order. Then,*

$$L\left(T_S \oplus_{(a,b)} t\right) - L\left(T_S\right) \geq (\lambda_t)_1 \tag{38}$$

$$\textit{and} \quad L\left(T_S \oplus_{(a,b)} t\right) - L\left(T_S\right) \geq \sum_{i=1}^{t-3} \frac{1}{2^i}(\lambda_t)_i + \frac{1}{2^{t-3}}(\lambda_t)_{(t-2)}. \tag{39}$$

Subsequently, for an optimal solution T^* to the BMEP, we have

$$L\left(T^*\right) - L\left(T_S\right) \geq \sum_{i\notin S}(\lambda_t)_1 \tag{40}$$

$$\textit{and} \quad L\left(T^*\right) - L\left(T_S\right) \geq \sum_{i\notin S}\left(\sum_{i=1}^{t-3} \frac{1}{2^i}(\lambda_t)_i + \frac{1}{2^{t-3}}(\lambda_t)_{(t-2)}\right). \tag{41}$$

Another set of lower bounds on the optimal solution to the BMEP was introduced by Catanzaro et al. [10]. The authors proved that the length function (1) is invariant under a doubly stochastic rescaling $\hat{D}$ of the distance matrix D up to a positive constant K. Furthermore, they showed that (1) can be rewritten as a cross-entropy

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

between probability distributions

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

where the *Kullback-Leibler divergence* D_{KL} is given by

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

Based on this reformulation of $L(T)$, they introduced the following family of lower bounds on the optimal solution to the BMEP and an algorithm to calculate these bounds in polynomial time.

Proposition 4. *Let $\hat{D}$ be a doubly stochastic rescaling of distance matrix D and let $\alpha \in [0, 1]$ be a fixed scalar. Then, there exists a positive constant K such that*

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

A third approach to obtain lower bounds on the optimal solution to the BMEP can be taken by solving relaxations of Formulation 1. Here we want to discuss the three possible relaxations connected to the fundamental equalities (10) and (11). First, we only require that the Kraft equality holds. Hence, we consider the following relaxation of Formulation 1:

Formulation 2.

$$\min \quad z = \sum_{i\in\Gamma}\sum_{j\in\Gamma_i} d_{ij}x_{ij} \tag{42}$$

$$s.t. \quad \sum_{j\in\Gamma_i} x_{ij} = \frac{1}{2} \qquad \forall\, i \in \Gamma \tag{43}$$

$$x_{ij} \in [0,1] \qquad \forall\, i,j \in \Gamma \tag{44}$$

As shown in the next proposition, Formulation 2 can be solved analytically in polynomial time because

$$\sum_{j\in\Gamma_i} x_{ij} = \frac{1}{2} \quad \text{ if and only if } \quad x_{ij} = 2^{-\tau_{ij}}.$$

For this purpose let $\text{sort}(x)$ denote the vector x in non-decreasing order and let d_i denote the i-th row of D.

Proposition 5. *Let $D \in \mathbb{R}^{n\times n}$ be a distance matrix and let τ_i be a path-length sequence in Θ_i. Then*

(i) $\sum_{k=1}^{n} d_{ik}\cdot 2^{-\tau_{ik}} \geq \sum_{k=1}^{n} sort(d_i)_k \cdot 2^{-\tau_{ik}}$

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

Proof.

Case 1. Consider indices $k < l$ for $d_i = (d_{i1},\dots,d_{ik},d_{il},\dots,d_{in})$ with $d_{ik} \geq d_{il}$. Since $2^{-\tau_{ik}} \geq 2^{-\tau_{il}}$, we know that $2^{-\tau_{ik}}\cdot d_{ik} + 2^{-\tau_{il}\cdot d_{il}} \geq 2^{-\tau_{ik}}\cdot d_{il} + 2^{-\tau_{il}}\cdot d_{ik}$. Hence, $\sum_{k=1}^{n} d_{ik}\cdot 2^{-\tau_{ik}}$ gets smaller if we swap d_{ik} and d_{il} in d_i. We can repeat this process until d_i is non-decreasing.

Case 2. Consider a path-length sequence $\tau_i' \in \Theta_i$ such that

$$\sum_{k=1}^{l-1} 2^{-\tau'_{ik}} \geq \sum_{k=1}^{l-1} 2^{-\tau_{ik}} \quad \forall l = 2,\dots,n \;\forall\, \tau_i \in \Theta_i. \tag{45}$$

As shown in [52], τ_i' exists and is uniquely defined by $\tau_i' = (2,3,\dots,(n-1),(n-1))$. Without loss of generality assume that d_i is sorted non-decreasing. Then

$$\begin{aligned}
\sum_{k=1}^{n} d_{ik}\cdot 2^{-\tau_{ik}} &= d_{i1}\cdot\sum_{k=1}^{n} 2^{-\tau_{ik}} + \sum_{l=2}^{n}\left((d_l - d_{l-1})\cdot\sum_{k=l}^{n} 2^{-\tau_{ik}}\right)\\
&\overset{(4)}{=} \frac{3d_{i1}}{2} + \sum_{l=2}^{n}\left((d_l - d_{l-1})\cdot\left(\frac{3}{2} - \sum_{k=1}^{l-1} 2^{-\tau_{ik}}\right)\right)\\
&\overset{(45)}{\geq} \frac{3d_{i1}}{2} + \sum_{l=2}^{n}\left((d_l - d_{l-1})\cdot\left(\frac{3}{2} - \sum_{k=1}^{l-1} 2^{-\tau'_{ik}}\right)\right)\\
&\overset{(4)}{=} d_{i1}\cdot\sum_{k=1}^{n} 2^{-\tau'_{ik}} + \sum_{l=2}^{n}\left((d_l - d_{l-1})\cdot\sum_{k=l}^{n} 2^{-\tau'_{ik}}\right)\\
&= \sum_{k=1}^{n} d_{ik}\cdot 2^{-\tau'_{ik}}.
\end{aligned}$$

Thus our claim follows.

□

Now, instead of the Kraft equality (10) we can employ the phylogenetic manifold (11) and require symmetry to obtain the following relaxation of Formulation 1:

Formulation 3.

$$\min \quad z = \sum_{i\in\Gamma}\sum_{j\in\Gamma_i} d_{ij}\left(\sum_{\ell\in P_{ij}(T_S)} 2^{-\ell}\, x_{ij}^{\ell}\right) \tag{46}$$

$$s.t. \quad \sum_{\ell\in P_{ij}(T_S)} x_{ij}^{\ell} = 1 \qquad \forall\, i,j\in\Gamma,\ i\neq j \tag{47}$$

$$x_{ij}^{\ell} = x_{ji}^{\ell} \qquad \forall\, i,j\in\Gamma, i\neq j, \forall\,\ell\in P_{ij}(T_S) \tag{48}$$

$$\sum_{i\in\Gamma}\sum_{j\in\Gamma_i}\sum_{\ell\in P_{ij}(T_S)} \ell 2^{-\ell}\, x_{ij}^{\ell} = (2n-3) \tag{49}$$

$$x_{ij}^{\ell} \in [0,1] \qquad \forall\, i,j\in\Gamma, i\neq j, \forall\,\ell\in P_{ij}(T_S) \tag{50}$$

[Placeholder Formulation 3]

Finally, combining the constraints of Formulations 2 and 3 leads to the following relaxation of Formulation 1:

Formulation 4.

$$\min \quad z = \sum_{i\in\Gamma}\sum_{j\in\Gamma_i} d_{ij}\left(\sum_{\ell\in P_{ij}(T_S)} 2^{-\ell}\, x_{ij}^{\ell}\right) \tag{51}$$

$$s.t. \quad \sum_{\ell\in P_{ij}(T_S)} x_{ij}^{\ell} = 1 \qquad \forall\, i,j\in\Gamma, i\neq j \tag{52}$$

$$x_{ij}^{\ell} = x_{ji}^{\ell} \qquad \forall\, i,j\in\Gamma, i\neq j, \forall\,\ell\in P_{ij}(T_S) \tag{53}$$

$$\sum_{j\in\Gamma_i}\sum_{\ell\in P_{ij}(T_S)} 2^{-\ell}\, x_{ij}^{\ell} = \frac{1}{2} \qquad \forall\, i\in\Gamma \tag{54}$$

$$\sum_{i\in\Gamma}\sum_{j\in\Gamma_i}\sum_{\ell\in P_{ij}(T_S)} \ell 2^{-\ell}\, x_{ij}^{\ell} = (2n-3) \tag{55}$$

$$x_{ij}^{\ell} \in \{0,1\} \qquad \forall\, i,j\in\Gamma, i\neq j, \forall\,\ell\in P_{ij}(T_S)$$

Since Catanzaro et al. [11] showed that the optimal solution $z^*(T_S)$ of Formulation 4 provides a valid lower bound on the optimal solution to the BMEP, we call the optimal objective function value of its LP relaxation $z^*_{\text{lin}}(T_S)$ the *Linear Programming Lower Bound* (LPLB). Notice that we can reduce the computational cost to obtain $z^*_{\text{lin}}(T_S)$ by removing equalities (53) and restricting constraints (54) and (55) as well as the objective function to $i < j$ for taxa $i, j \in \Gamma$. However, solving the LPLB remains computationally expensive. Therefore, Catanzaro et al. [11] also introduced a Lagrangian relaxation (LagLB) of the LPLB by using Lagrangian multipliers for constraints (54) and (55) which yields the following model:

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

$$\text{s.t.} \quad \sum_{\ell\in P_{ij}(T_S)} x_{ij}^{\ell} = 1 \ \ \forall\, i,j\in\Gamma, i\neq j$$

$$x_{ij}^{\ell} \in [0,1] \ \ \forall\, i,j\in\Gamma, i\neq j, \forall\,\ell\in P_{ij}(T_S)$$

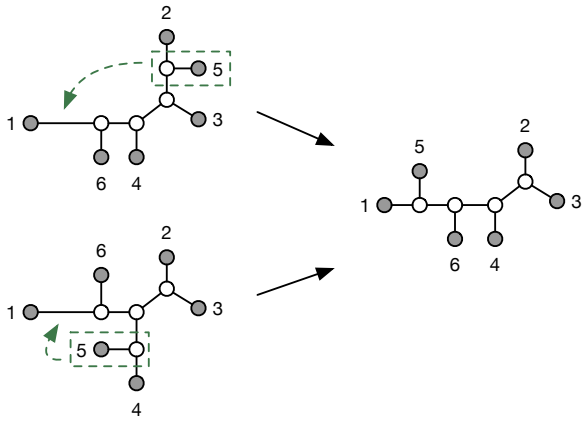


Figure 7: Two examples where the phylogeny on the right is obtained through a *Subtree Pruning and Regrafting* (SPR) operation [27] on either one of the phylogenies on the left.

This approach makes the decision variables x_{ij}^{ℓ} independent of one another. Therefore, for given multipliers μ and λ, LagLB can be solved analytically a lot faster than the LPLB to obtain a valid lower bound on the optimal solution to the BMEP:

$$z^*_{\text{lag}}(T_S, \mu, \lambda). \tag{56}$$

5. Upper bounds on the optimal solution to the BMEP

We have seen in Section 3 that $z^*_{\text{lin}}(T_S)$ is a valid lower bound for a SAS on the BMEP derived from some characteristics of the set Θ introduced in Section 2.2. Moreover, its Lagrangian relaxation was employed to speed-up the pruning of the search-tree of the SAS. Specifically, sub-phylogenies are pruned and discarded if they do not improve upon the shortest length of all previously constructed phylogenies. Instead of discarding all of them, one might observe that a non-optimal phylogeny shares a majority of topological distances with an optimal solution to the BMEP. For example, the SAS shown in Figure 6 constructs, among others, the three phylogenies shown in Figure 7. The phylogeny on the right is an optimal solution to the BMEP and the two phylogenies on the left are not optimal. However, we can make the small modification of removing taxon 5, its adjacent internal vertex and the external edge connecting them and reinserting both vertices and the external edge on the path between taxon 1 and the internal vertex connected to it. Then, we obtain an optimal solution to the BMEP in constant time after building either one of the two phylogenies on the left. Hence, we can improve the lower bounds evaluated in the SAS by local topological interchanges of taxa, subtrees of taxa and/or internal edges on the phylogenies we construct. The classification of these operations in the literature are shown in Figure 8. To keep the computational cost of the local search for local topological interchanges inexpensive, i.e., keep the neighbourhood of all phylogenies that can be obtained through one topological interchange small, we restrict ourselves to NNIs.

Outside of a Branch-&-Bound-framework, both the SAS and the agglomerative approach can be used as a basis for the greedy method to approximate the BMEP, too. For the SAS, this means we evaluate consecutive branchings of the initial sub-phylogeny on three taxa by only continuing our search on branchings T which are minimal with respect to a selection criterion $C(T)$ (see Algorithm 2). For the agglomerative approach, we specify the greedy method by starting with the star graph and inserting new internal edges (u_1, u_2), replacing internal vertices u. Such an internal vertex u is determined by the minimization of a selection criterion on all bipartitions $P = (N_1, N_2)$ of the neighbourhood of u. Then, the neighbourhood of u_1 and u_2 are set to be $N_1 \cup \{u_2\}$ and $N_2 \cup \{u_1\}$, respectively. We continue this process of internal edge insertions until we obtain a phylogeny, i.e., until we have agglomerated $(2n-3)$ edges (see Algorithm 3). We have shown how both approaches operate in Figures 5 and 9 and now added a greedy selection criterion as a specification of each step for both strategies. Thus, Algorithms 2 and 3 are correct greedy algorithms for the BMEP. To specify these algorithms we choose greedy selection criteria with a low computational cost that yield

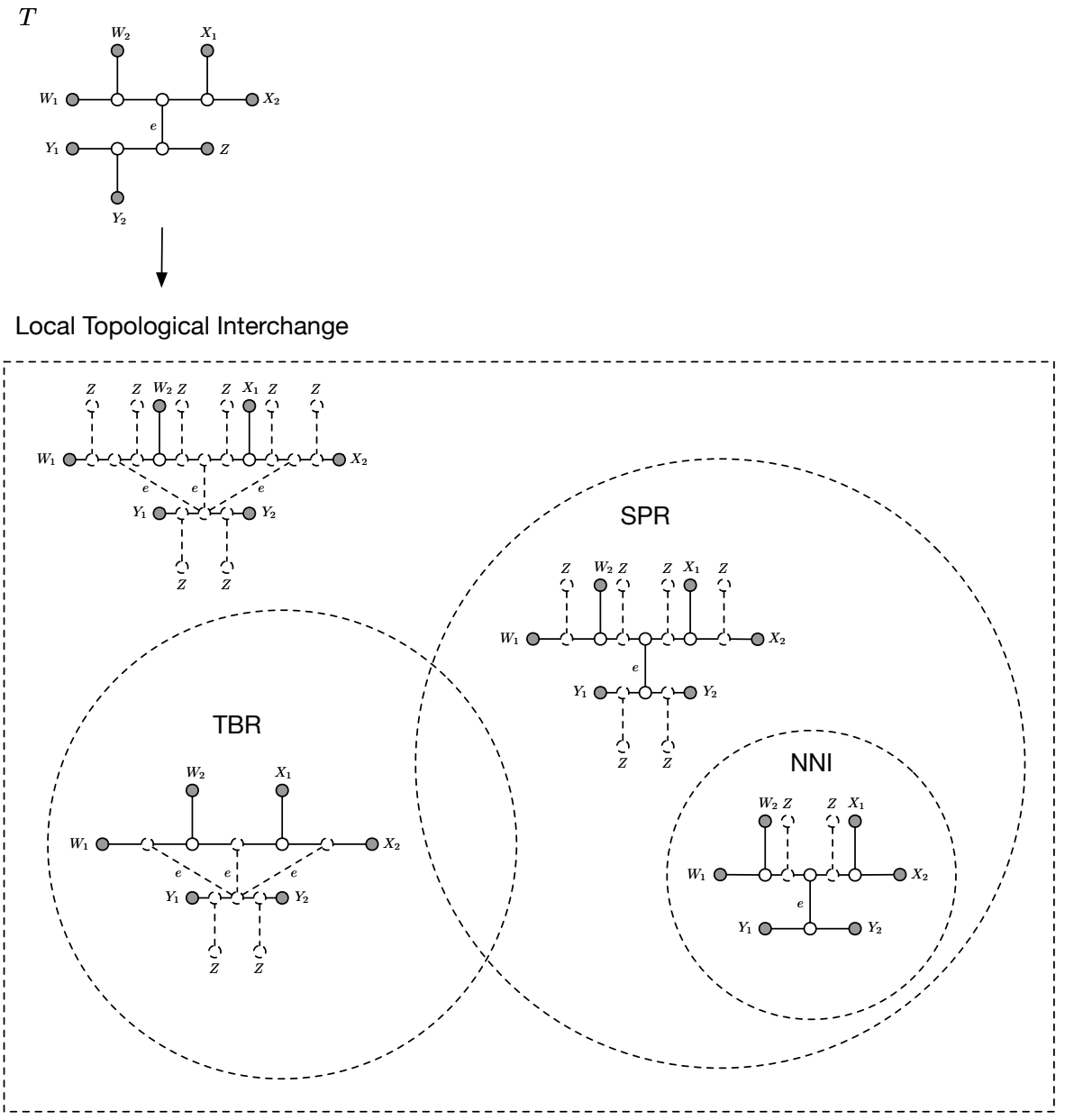


Figure 8: Shows a phylogeny as a graph T in which each external vertex represents an arbitrary but fixed binary tree rooted in the vertices $W_1, W_2, X_1, X_2, Y_1, Y_2$ and Z, respectively. Moreover, all topological interchanges shown below can be obtained through the reconnection of the edge e and/or subtree Z in T. Among all possible local topological interchanges, we can distinguish between the *Nearest Neighbor Interchange* (NNI), the *Subtree Pruning and Regrafting* (SPR), and the *Tree Bisection and Reconnection* (TBR), respectively [27, 33].

good upper bounds on the optimal solution in our following experiments. For Algorithm 2, we consider

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

the lower bound on the optimal solution that we used in the example shown in Figure 6, which was introduced as a selection criterion for the greedy method by Pardi [48]. For Algorithm 3, we pick u according to the bipartition

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;

$P = (N_1, N_2)$ with $|N_1| > |N_2| = 2$ which minimizes the change in Semple and Steel [59]'s extension of length function (1) to unrooted non-binary trees. This method is known in the literature as the *Neighbour-Joining algorithm* (NJ) [32, 35, 57, 60].

6. A massively parallel exact solution algorithm for the BMEP

In this section we show how the exact solution algorithm of Catanzaro et al. [11] presented in the last section can be parallelized to improve the method. Moreover, we discuss lower bounds and upper bounds on the optimal solution to the BMEP. To this end, we first investigate how a parallel version of the stepwise addition strategy can operate and why it is well suited for parallel exact solution algorithms for the BMEP.

6.1. Branching on partial solutions to the BMEP in parallel

A parallel Branch-&-Bound scheme for the BMEP could be based on a parallelized SAS to make updates of the global intermediate optimal solution T^* less frequent and therefore prune more nodes of the search tree. A naive idea to achieve this objective would be to start, for some fixed $m \in \mathbb{N}$, by enumerating all sub-phylogenies with less than m taxa iteratively. Then, assign every available branching on m taxa to a fixed core of a multi core processor and run a SAS initialized with $S = \{1, \dots, m\}$ on each core. The issue with this idea is twofold. On the one hand, for any choice of m there are between $((2m-7)!!+1)$ and $(2m-5)!!$ sub-phylogenies for which a SAS needs to be completed. Hence, the number of SAS's to be completed can be a vast mismatch with the number of cores available, leading to many idle cores. On the other hand, even if all or most of the cores are employed to enumerate candidates for an optimal solution to the BMEP, the workload of each individual core can be very unequal. Looking at the example in Figure 6 we can see that if we would assign all branchings on four taxa to three different cores, then two of the cores would turn idle long before the optimal phylogeny is found. To mitigate both problems, we introduce a queue to (i) manage the distribution of the workload for all cores by storing a sufficient among of sub-phylogenies in the queue for idle cores to pop and process, and (ii) rebalance the workload for each individual core by pushing sets of unprocessed sub-phylogenies from some working cores to the queue.

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 neighbours;
  $V \leftarrow V \setminus \{u\} \cup \{u_1, u_2\}$;
  $E \leftarrow E \setminus \{(u, v) : (u, v) \in E\} \cup \{(u_1, v) : v \in N_1\} \cup \{(u_2, v) : v \in N_2\} \cup \{\{u_1, u_2\}\}$;
return (V, E);

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
$T^* \leftarrow$ NULL;
$L(T^*) \leftarrow \infty$;
$S \leftarrow \{1, 2, 3\}$;
$T_S \leftarrow$ The only sub-phylogeny of Γ for 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 ProcessJob$(D, Q, \text{pop}(Q), T^*)$ on an idle core;
return T^*;

To introduce this procedure formally, we define a *job* as a triple (Γ, T_S, F) where T_S is a sub-phylogeny of Γ and $F \subseteq E(T_S)$ a subset of edges in T_S. Then, we establish a queue managing our jobs as shown in Algorithm 4. First, the star graph on three taxa T_S is constructed and for all three branchings of T_S on taxon 4 we push a job into the queue Q. All calculations until the end of the for-loop can be processed on one core. Then, the while-loop distributes jobs in the queue Q to all available cores of the multi core processor. Furthermore, we process a job $J = (\Gamma, T_S, F)$ from Q on one

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

Input: Distance matrix D for Γ, queue Q of jobs, job $J = (\Gamma, T_S, F)$, phylogeny T^* of Γ
continue $\leftarrow$ FALSE;
if $|\text{taxa}(T_S)| = 3$ **then**
    if $z^*_{lin}(T_S) < L(T^*)$ **then**
        continue $\leftarrow$ TRUE;
else
    if $z^*_{lag}(T_S) < L(T^*)$ **then**
        if $z^*_{lin}(T_S) < L(T^*)$ **then**
            continue $\leftarrow$ TRUE;
if continue **then**
    if $\text{taxa}(T_S) = \Gamma$ **then**
        $T^* \leftarrow T_S$;
    else
        $t \leftarrow |\text{taxa}(T_S)| + 1$;
        for $e \in F$ **do**
            $T \leftarrow T_S \oplus_e t$;
            if There exist idle cores **or** queue Q contains less jobs than there are overall cores **then**
                Partition $E(T) = F_1 \cup F_2$ such that $||F_1| - |F_2|| \leq 1$;
                if queue Q contains less jobs than there are overall cores **then**
                    Push (Γ, T, F_2) into Q;
                    ProcessJob$(D, Q, (\Gamma, T, F_1), T^*)$;
                else
                    ProcessJob$(D, Q, (\Gamma, T, F), T^*)$;
            ProcessJob$(D, Q, (\Gamma, T, F), T^*)$;

core C by following the same stepwise addition search strategy as Algorithm **??** while also rebalancing the number of branchings that core C has to build and evaluate before turning idle as shown in Algorithm 5. After the construction of an unpruned branching T we either partition $E(T)$ into two disjoint subsets F_1 and F_2 whose cardinalities differ at most by one or process the new job (Γ, T, F). This decision depends on the length of queue Q because the creation of new jobs for subsets F_1 and F_2 instead of F can not decrease the total idle time of all cores if every core can already process a job from Q. We verify this state of the queue again once job (Γ, T, F_2) is pushed into Q and the recursion continues on core C. This is necessary due to the fact that either idle cores can decrease the length of Q or working cores different from C can increase the length of Q. Both cases are checked by linking two if-conditions as shown in Algorithm 5.

Proposition 6. *Algorithm 4 terminates and returns an optimal solution for the BMEP.*

Proof. Suppose by contradiction that Algorithm 4 does not terminate. This means, queue Q is never empty. Hence, an infinite number of jobs is created in Algorithm 5 because one job is removed from Q in every iteration of the while-loop. However, for the job (Γ, T, F_2) that can be pushed into queue Q and the given job $J = (\Gamma, T_S, F)$ in Algorithm 5, $|F_2|$ is not larger than $|F|$. Therefore, the number of jobs that Algorithms 4 and 5 create is finite, leading to a contradiction. Thus, Algorithm 4 terminates.

To see that Algorithm 4 returns an optimal solution to the BMEP observe that the enumeration order of Algorithms 4 and 5 coincides with the enumeration order of all phylogenies by Algorithms **??** and **??** if the edges in F_1 and F_2 of Algorithm 5 follow the same order as the edges in F. Notice that no additional pruning of the search-tree is introduced throughout the recursion of Algorithm 5. Thus, T^* is an optimal solution for the BMEP by enumeration. □

Proposition 6 shows that reorganizing the SAS in parallel with a queue of jobs does extent single-core solvers like Catanzaro et al.'s method. Any implementation of Algorithms 4 and 5 then requires the specification of

(A) the order of taxa Γ,

(B) the order of edges F,

(C) the order of jobs in the queue Q,

(D) the partition of edges $E(T)$.

We can see in Algorithm 5 that the partitioning aspect (D) can be generalized to not only include bipartitions, but any partition $F_1 \cup \cdots \cup F_p$ composed of ordered edge sets $F_i \subseteq E(T), 1 \leq i \leq p$. Since we can only process one job at a time on one core we push jobs (Γ, T, F_i) for $i = 2, \ldots, p$ into the queue Q. Moreover, we keep the assumption $||F_i| - |F_j|| \leq 1$ for $i, j = 1, \ldots, p$ because then every job $(\Gamma, T, F_i), i = 2, \ldots, p$ offers a roughly equal number of branchings to be build and evaluated. This extends aspect (D) of Algorithm 5 to include

(D.1) the number $p \leq |E(T)|$ of partition sets F_i, $i = 1, \ldots, p$,

(D.2) for some positive constant integer c, 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, $i = 1, \ldots, p$.

The latter definition arises from the restriction $p \leq |E(T)|$. Since $|E(T)|$ grows linearly in the number of taxa, large values for p may produce an overabundance of jobs for sub-phylogenies defined on few edges. The lower bound (D.2) on the number of edges included in one job ensures that the rebalancing 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 multi core processor. However, as a sufficiently large queue is the bottleneck for any parallel exact algorithm following a SAS, it is unlikely a partition minimizing the total idle time of all cores could be defined a-priori.

To conclude the discussion on branchings, we consider a popular alternative to the SAS to enumerate all phylogenies called the *agglomerative approach* [33]. These are constructive methods which start from an infeasible solution

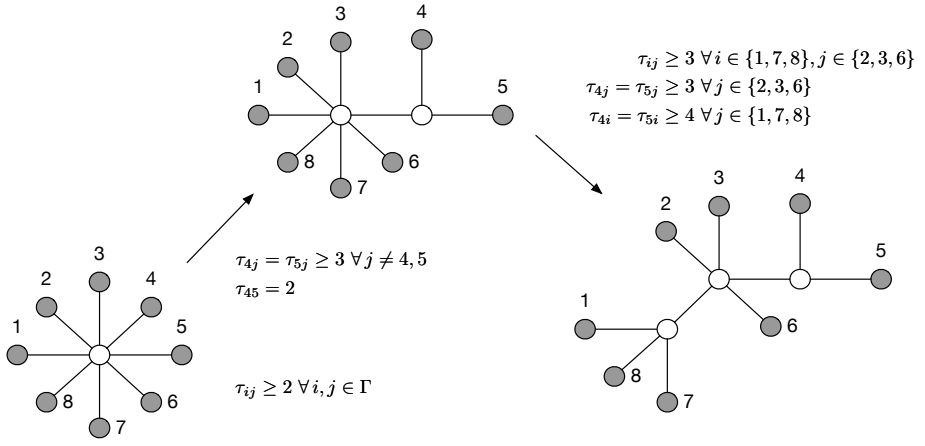


Figure 9: An example of two steps of the agglomerative approach in the case of eight taxa.

to the BMEP and iteratively cluster taxa according to a specific optimality criterion, until when a phylogeny of Γ is obtained. An example of two consecutive steps of the agglomerative approach are shown in Figure 9. First, an internal edge is inserted between the subtrees on taxa $\{1, 2, 3, 6, 7, 8\}$ and $\{4, 5\}$ into the star graph on eight taxa. Then, an internal edge is inserted between the subtrees on taxa $\{1, 7, 8\}$ and $\{2, 3, 4, 5, 6\}$ in the resulting tree. Like in the SAS, every step of the agglomerative approach corresponds to branchings on specific topological distances. However, since the insertion of internal edges is fully determined by the incident internal vertices and not the taxa of the tree, this technique does not break symmetries in the search-tree. We can see this by continuing our example: reversing the order in which we insert the internal edges produces a tree not different from the one on the right in Figure 9. Hence, we can not expect to obtain a Branch-&-Bound scheme through the agglomeration of taxa which is competitive with the SAS in terms of computation time.

As the two branching approaches presented so far capture all topological information that could be retrieved by inserting external or internal edges, we conclude that the SAS is a suitable choice to compute branchings in our parallel exact solution algorithm for the BMEP.

In the next section we will use all the techniques presented so far to improve upon Catanzaro et al. [11]'s exact solution algorithm for the BMEP.

6.2. Experiments

We assemble all algorithms discussed in this article to solve the BMEP in parallel and establish new benchmarks for the solution time of the BMEP on the same instances used by Catanzaro et al. [11] in their experiments. Recall that these instances are built from real aligned DNA data sets [13]. To this end, we call the parallelized SAS consisting of Algorithm 4 and 5 where

(A) the order of taxa Γ is identical with the order of taxa in the distance matrix $\boldsymbol{D}$,

(B) the edges F are ordered by the time at which each edge was created during the execution of current SAS,

(C) the jobs are added and removed from the queue $\boldsymbol{Q}$ in a first-in-first-out order,

(D) parameters p and c are set to 2 and 1, respectively, and subsets F_1 and F_2 contain the first $\lfloor |F|/2 \rfloor$ edges of F and last $\lceil |F|/2 \rceil$ edges of F, respectively,

the *Parallel Stepwise Addition Strategy With Default Configuration* (PSAS-C1111). Furthermore, we preprocess Algorithms 2 and 3 as specified in the last section to calculate two upper bounds on the optimal solution to the BMEP and store the better one before starting the PSAS. Moreover, we apply all possible NNI operations to the phylogeny underlying the stored upper bound and every newly stored phylogeny T^* during the execution of PSAS to potentially improve them. We refer to this extension of the PSAS-C1111 as EPSAS-C1111. Next, we consider alternative configurations by considering variations of specifications (A) to (C). This means,

(A) the order of taxa Γ is identical with the order of taxa

(A.1) in the distance matrix D,

(A.2) in the optimal solution to the Traveling Salesman Problem for distance matrix D,

(A.3) obtained by shortcutting the tree returned by Algorithm 2,

(A.4) obtained by shortcutting the tree returned by Algorithm 3;

(B) the edges F are ordered

(B.1) by the time at which each edge was created during the execution of the current SAS,

(B.2) non-decreasing in the pre-calculated values lowerbound(T) one would obtain in the next recursion step by branching on the corresponding edge;

(C) the jobs are added and removed from the queue Q

(C.1) in a first-in-first-out order,

(C.2) in a first-in-last-out order,

(C.3) in non-increasing order of the number of edges contained in the underlying sub-phylogeny,

(C.4) in non-decreasing order of the number of edges contained in the underlying sub-phylogeny.

When referring to one fixed combination of specifications to modify the implementation of algorithm EPSAS-C1111 we change the suffix C1111 accordingly, e.g., C4231 means we chose (A.4), (B.2) and (C3) instead of (A.1), (B.1) and (C.1), respectively.

To test the performance of algorithms EPSAS-C1111 to EPSAS-C4241, we used the same data sets as Catanzaro et al. [11] and set one hour as the maximum time limit. All numerical experiments were performed on the Cesam server provided by the supercomputing facilities of the Université catholique de Louvain which is equipped with a 2x8 Core E5-2667v3 Processor at 3.2 GHz and 256GB RAM. The source code is written in C++ and compiled with the GCC compiler version 4.8.5.

The results of the first series of tests are shown in Figures 10 to 14.

First, we observe that the choice for the order of jobs in the queue (C) has an impact of multiple magnitudes smaller than the choice for taxa order (A) and edge order (B) and that no queue order yields a statistical significant advantage compared to all others.

Next, we can see that edge order (B.1) has a positive impact on the overall runtime of all tested configurations compared to (B.2). Specifically, the worst case and 75% quartile runtimes improve drastically for increasing n. This leads to a twofold conclusion. On the one hand, the additional preprocessing overhead for the calculation of lower bounds required in each iteration to determine edge order (B.2) affects the runtime for small n negatively. On the other hand, the calculation of lower bounds throughout the tested algorithms dominates all other contributions to the total computation time. This means, preprocessing lower bounds for future iterations does not scale well with a higher number of cores and increasing n because at the time a pre-calculated lower bound is employed by one core the corresponding node of the search-tree might already be pruned by a different core.

Concerning the taxa order, the test results are less clear. The best worst-case performance is always reached by taxa order (A.4), but the lowest best-case and 25% quartile runtime is achieved by taxa orders different from (A.4) for most n. However, since taxa order (A.4) performs statistically significant better than all other configurations for $n = 20$ we keep this order as the best choice for further experiments.

Thus, overall we conclude that configurations C411 to C414 perform the best. Now, we consider further configurations extending specification (D) with

$$\text{(D.1) } p = 2, c = 1, \text{ (D.2) } p = 3, c = 1, \dots, \text{ (D.8) } p = 9, c = 1.$$

Then, Figures 15 to 17 show the computational results for configurations C4111 to C4148. Here we can observe that no choice of parameters for specification (D) 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 algorithm EPSAS. This observation matches our previous assessment that smaller 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

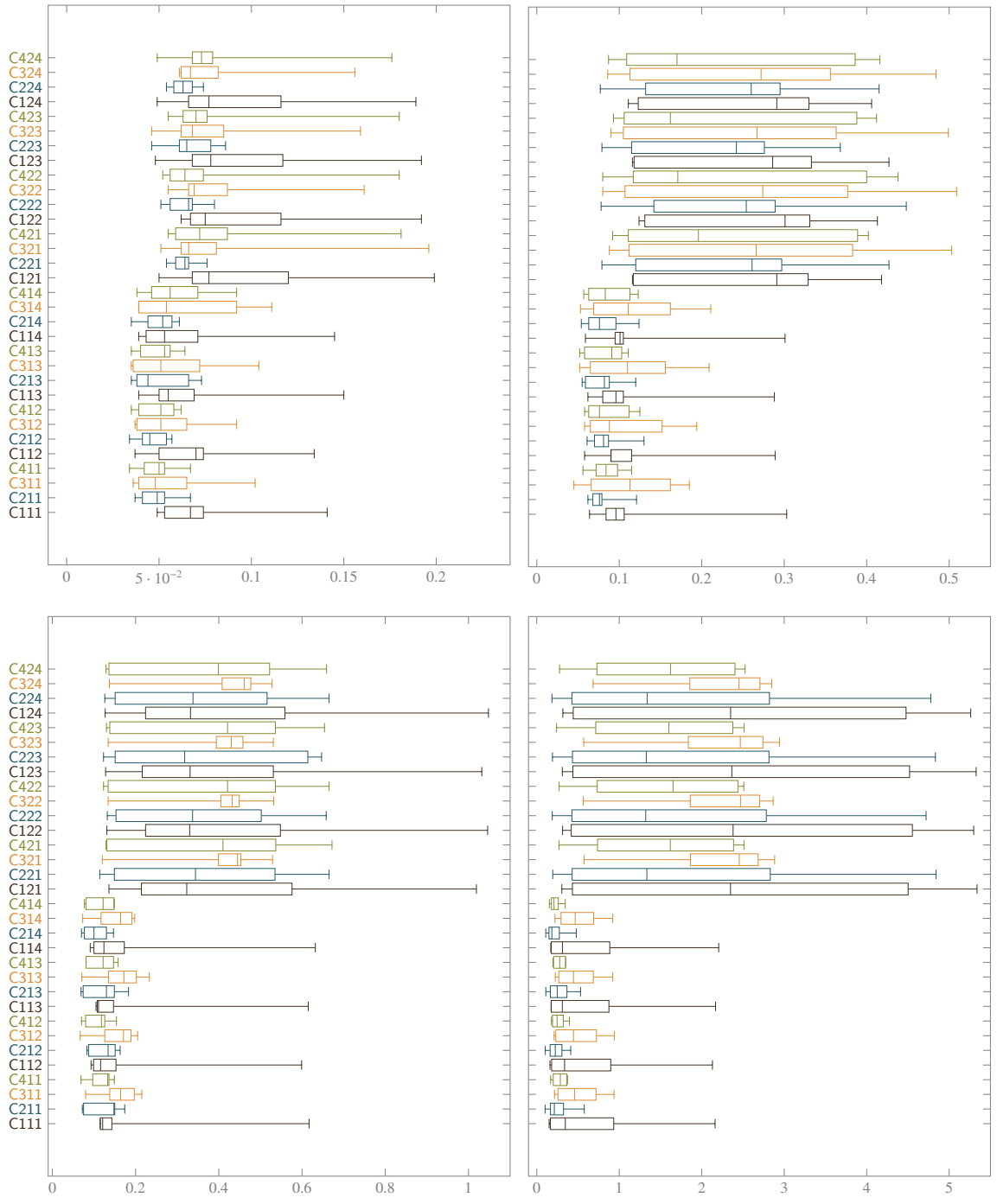


Figure 10: A boxplot of the runtime in seconds for the 32 algorithms EPSAS-C1111 to EPSAS-C4241 for $n = 10$ (upper left), $n = 11$ (upper right), $n = 12$ (lower left) and $n = 13$ (lower right).

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 and extend specification (D) further as follows:

$$\text{(D.9) } p = 2, c = 2, \ \text{(D.10) } p = 2, c = 3, \ \text{(D.11) } p = 2, c = 4,$$
$$\text{(D.12) } p = 3, c = 2, \text{(D.13) } p = 3, c = 3, \ \text{(D.14) } p = 3, c = 4.$$

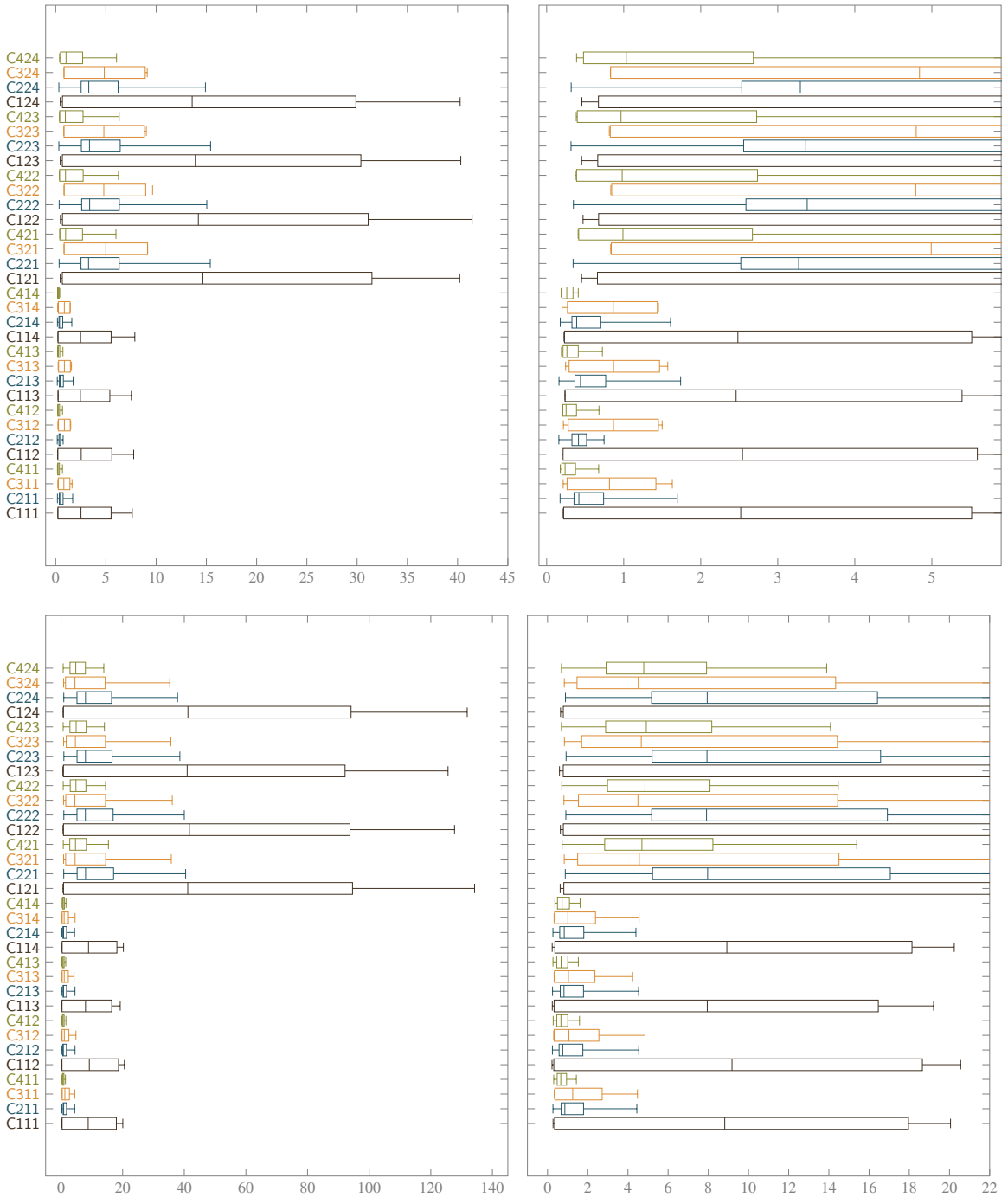


Figure 11: A boxplot of the runtime in seconds for the 32 algorithms EPSAS-C1111 to EPSAS-C4241 for $n = 14$ (top) and $n = 15$ (bottom).

Then, Figures 18 to 20 show the computational results for configurations C4119 to C41414 and C4112, C4113 to C4142, C4143, respectively. Again, we observe that no specification of (D) is significantly better compared to all tested configurations. Moreover, executions of EPSAS with smaller choices for the constant c remain among the best performances for increasing n. This strengthens our suspicion that the best strategy to partition the set of jobs of a sub-phylogeny can not be defined a-priori and requires the employment of more adaptive methods.

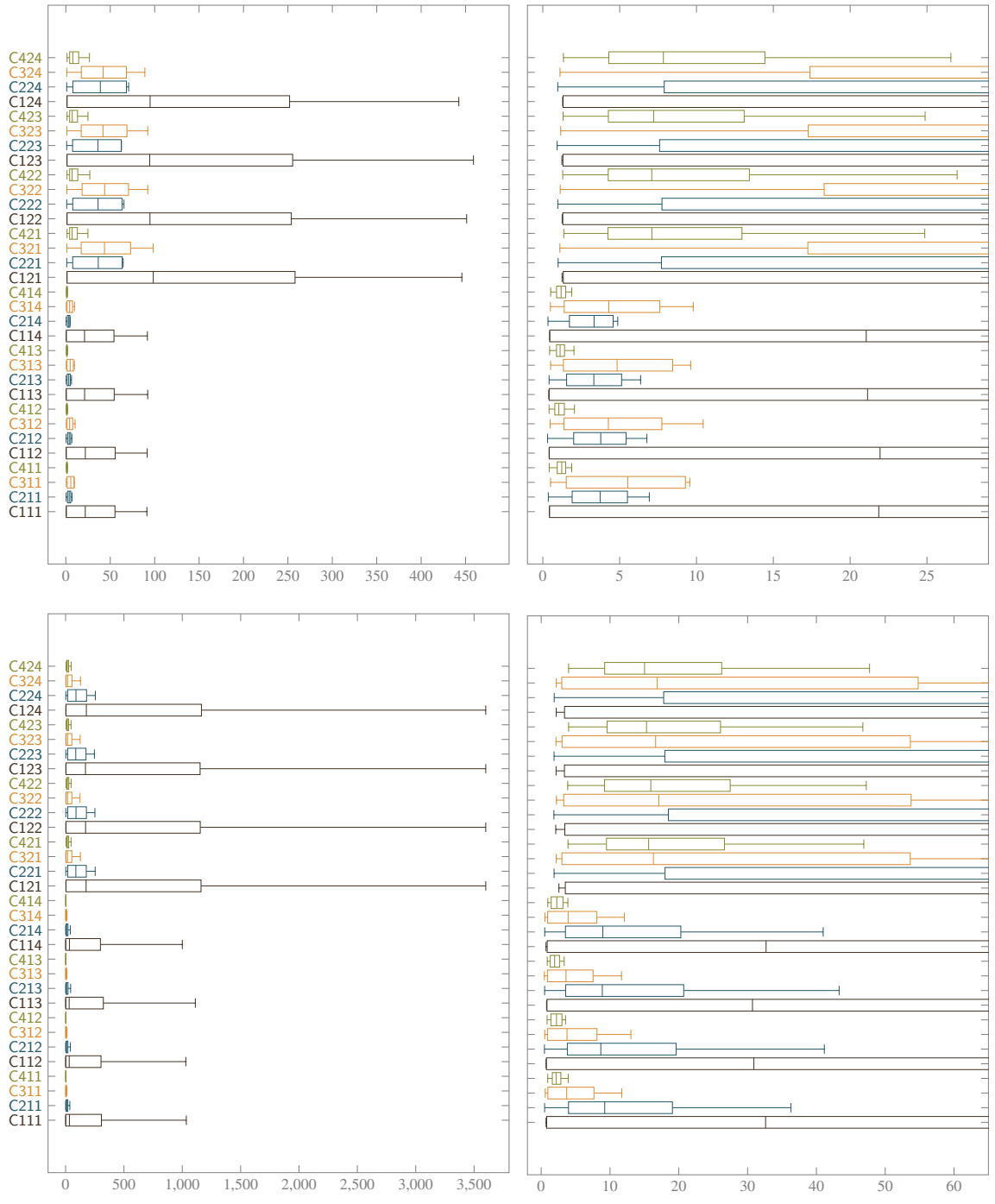


Figure 12: A boxplot of the runtime in seconds for the 32 algorithms EPSAS-C1111 to EPSAS-C4241 for $n = 16$ (top) and $n = 17$ (bottom).

To conclude our experiments for the instances evaluated so far, we establish new benchmarks for the solution time of the BMEP by using algorithm EPSAS-C41412, which offers the best worst case performance among all analysed configurations of EPSAS for $n = 20$. Recall that this means we implement algorithm EPSAS such that

(A) the order of taxa Γ is obtained by shortcutting the tree returned by Algorithm 3,

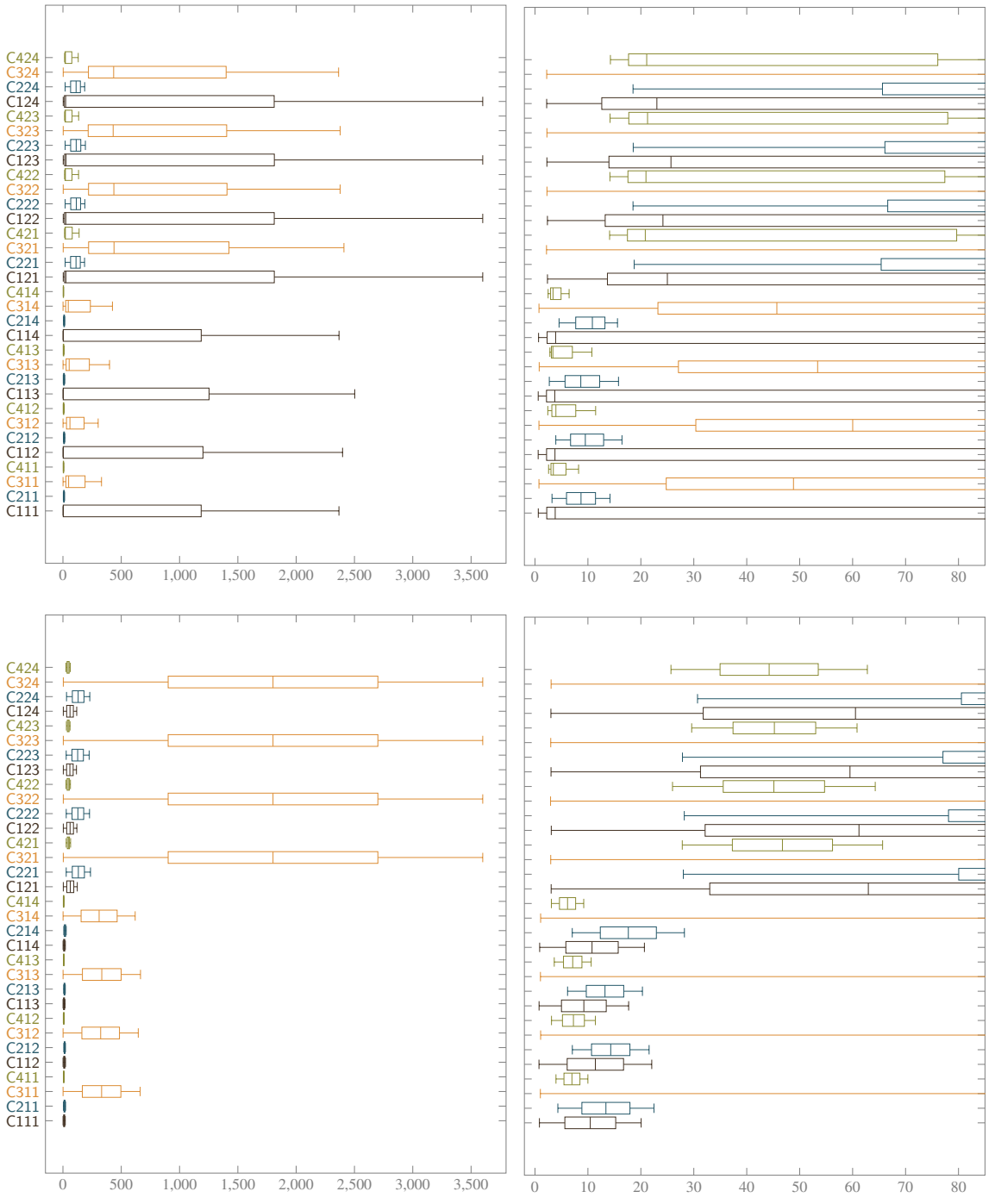


Figure 13: A boxplot of the runtime in seconds for the 32 algorithms EPSAS-C1111 to EPSAS-C4241 for $n = 18$ (top) and $n = 19$ (bottom).

(B) the edges F are ordered by the time at which each edge was created during the execution of current SAS,

(C) the jobs are added and removed from the queue Q in non-decreasing order of the number of edges contained in the underlying sub-phylogeny,

(D) parameters p and c are set to 3 and 2, respectively, and subsets F_1 and F_2 contain the first $\lfloor |F|/2 \rfloor$ edges of F

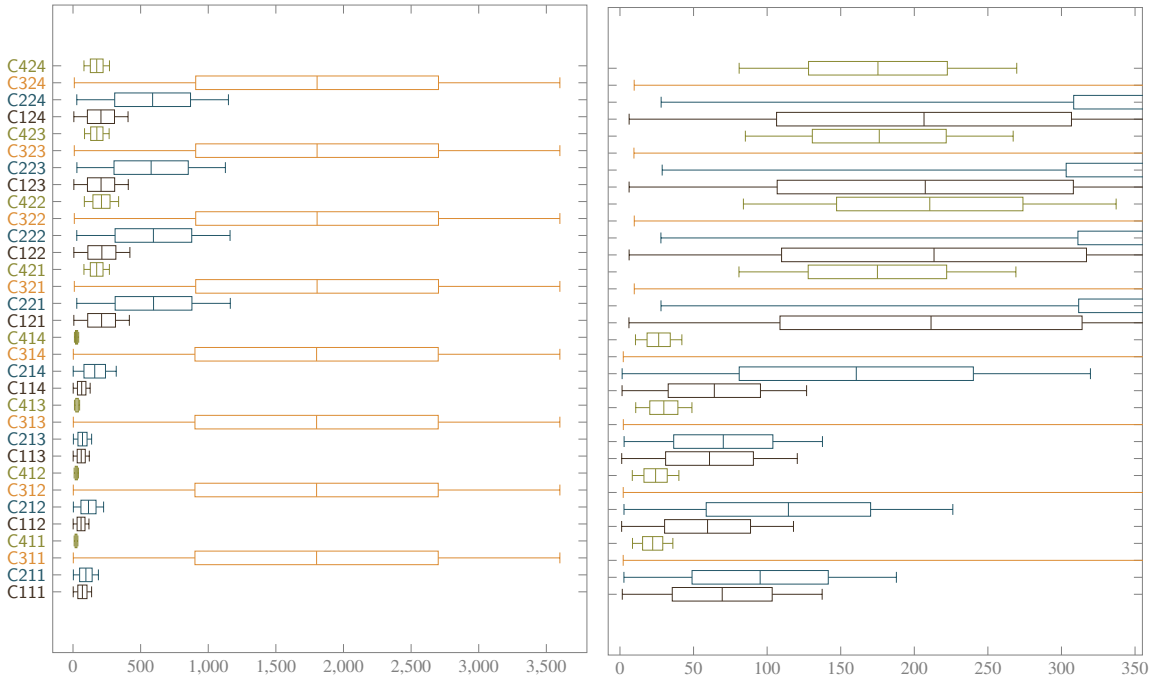


Figure 14: A boxplot of the runtime in seconds for the 32 algorithms EPSAS-C1111 to EPSAS-C4241 for $n = 20$.

and last $\lceil |F|/2 \rceil$ edges of F, respectively,

Tables 4 and 5 summarize the numerical results with respect to the solution time and the number of branchings needed in the EPSAS-C41412 to solve a specific instance. In general we see that the single core EPSAS-C41412 as well as the multi core version solve the BMEP faster than Catanzaro et al. [11]'s state-of-the-art solution algorithm. 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 EPSAS-C41412 on a few taxa is more expensive than the potential for improvements in the solution time through parallelization. Furthermore, we observe that the multi core EPSAS-C41412 is twice as fast as the single core version 99% of the time. The relative improvements when comparing both versions of the EPSAS-C41412 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 in under an hour. Thus, EPSAS-C41412 increases the number of taxa for which the BMEP can be solved in a reasonable amount of time. This raises the question how much farther faster algorithms than the EPSAS-C41412 could go before meeting the numerical limits of the BMEP. We discuss this issue and a possible resolution in the next two sections.

7. On the numerical stability of the BMEP

The objective function of the BMEP is subjected to numerical instabilities whenever the generic addend $d_{ij}2^{-\tau_{ij}}$ approaches (or get smaller than) the machine epsilon ϵ, i.e., if there exist at least a pair of distinct taxa $i, j \in \Gamma$, such that

$$d_{ij} \cdot 2^{-\tau_{ij}} \leq \epsilon. \tag{57}$$

This situation limits the analysis of practical instances involving hundreds or thousands taxa (see, e.g., the phylogenetic analysis of the datasets related to SARS-Cov2 [9]) and may occur because of specific values of either the path-length τ_{ij} or the distance d_{ij} (or both). In order to characterize the impact of path-lengths on the numerical stability of the BMEP, we start by recalling that in any feasible solution to the problem the value of each τ_{ij} may vary within the discrete interval $[2, \ldots, n-1]$ [14]. Then, it is licit to wonder which is the smallest number n of taxa for which

Data set	Taxa	Optimum	(40)	(41)	ManifoldLB	KraftLB	LPLB
Primates12	10	124.968	108.969	117.552	72.422	88.273	121.762
	11	332.834	288.448	308.570	184.726	242.881	324.211
	12	802.589	694.463	740.163	455.455	623.237	782.755
M17	10	105.134	95.768	100.189	72.929	87.143	104.033
	11	261.233	235.047	246.490	170.613	211.179	258.146
	12	541.632	484.067	508.402	349.652	439.381	535.065
	13	1,181.598	1,046.775	1,098.695	780.718	969.277	1,162.043
	14	2,408.066	2,128.879	2,236.155	1,545.662	1,965.73	2,368.216
	15	4,998.295	4,405.708	4,630.741	3,200.613	4,112.441	4,915.784
	16	10,225.560	8,912.794	9,371.332	6,455.794	8,375.861	10,051.994
	17	20,788.112	18,078.766	19,031.724	12,949.645	16,983.045	20,447.707
M18	10	190.331	149.038	164.735	145.096	156.581	186.288
	11	396.942	293.971	330.568	289.819	322.578	386.718
	12	805.937	593.128	667.791	567.978	643.918	784.056
	13	1,758.111	1,210.911	1,382.077	1,231.903	1,422.968	1,700.618
	14	3,599.678	2,410.664	2,763.052	2,431.544	2,845.200	3,469.282
	15	7,746.218	4,977.406	5,793.984	5,305.496	6,274.095	7,477.146
	16	15,973.016	10,035.621	11,754.719	11,093.488	13,003.910	15,411.761
	17	32,345.662	19,137.158	22,911.055	22,455.971	26,184.750	31,202.619
	18	66,029.112	37,905.480	45,790.686	44,556.665	52,988.767	63,683.315
SeedPlant25	10	91.458	65.995	77.449	77.494	74.902	86.549
	11	206.250	148.666	173.268	170.437	168.952	195.988
	12	427.816	304.805	356.940	360.248	352.544	407.022
	13	943.774	655.587	761.057	771.943	771.069	893.893
	14	1,929.219	1,341.091	1,553.194	1,591.185	1,574.575	1,831.570
	15	4,085.763	2,852.389	3,287.307	3,338.153	3,326.320	3,895.098
	16	8,353.457	5,419.463	6,334.814	6,598.428	6,707.819	7,872.086
	17	17,314.429	10,557.810	12,441.439	13,493.130	13,858.091	16,239.100
	18	36,156.527	21,524.378	25,344.199	27,666.698	28,491.876	33,654.096
	19	75,261.323	44,530.420	52,409.879	56,998.660	59,384.548	70,014.666
	20	164,507.262	93,563.110	109,511.370	119,602.162	126,196.890	149,517.567
M43	10	105.020	95.911	99.003	85.251	93.391	103.652
	11	209.282	190.152	196.591	164.251	180.851	204.268
	12	434.326	387.748	401.064	335.465	375.195	426.557
	13	895.424	796.566	823.990	680.809	767.502	879.878
	14	1,808.970	1,592.982	1,650.212	1,328.144	1,524.650	1,777.358
	15	3,965.234	3,435.268	3,569.271	2,852.487	3,314.804	3,904.670
	16	8,219.835	7,077.141	7,356.954	6,004.717	6,968.688	8,095.798
	17	16,798.759	14,392.029	14,985.303	12,336.357	14,162.839	16,493.312
	18	35,383.158	30,094.004	31,350.532	25,772.321	30,070.622	34,874.323
	19	71,769.903	60,992.591	63,521.426	52,611.255	60,908.263	70,743.386
	20	157,472.424	133,024.643	138,290.054	112,460.414	132,924.962	155,142.306

Table 2
Numerical results obtained for the lower bounds from Section 4 at the root node of the stepwise addition strategy.

Data set	Taxa	Optimum	(40)	(41)	ManifoldLB	KraftLB	LPLB
RbcL55	10	152.482	132.582	140.757	111.541	125.572	149.733
	11	328.645	285.263	303.619	243.473	272.983	323.349
	12	685.972	589.787	628.710	511.399	572.729	670.868
	13	1,502.870	1,251.220	1,335.970	1,104.865	1,257.331	1,467.890
	14	3,094.888	2,500.265	2,684.557	2,242.954	2,591.957	3,024.746
	15	6,448.259	5,169.294	5,558.499	4,678.343	5,377.984	6,289.820
	16	13,455.292	10,547.637	11,352.353	9,709.872	11,214.342	13,119.319
	17	27,804.246	20,939.017	22,685.482	19,643.587	23,144.066	26,987.288
	18	56,175.236	40,637.044	44,449.031	38,573.673	46,045.202	54,544.159
	19	114,623.865	80,064.634	88,318.788	76,318.718	93,223.277	111,223.724
	20	236,424.336	154,333.474	171,925.735	150,025.379	185,399.115	225,473.659
M62	10	163.876	155.264	158.801	117.649	133.127	162.117
	11	350.425	330.942	338.429	248.494	285.167	345.178
	12	753.821	710.045	725.293	525.766	611.003	738.269
	13	1,580.764	1,480.439	1,514.172	1,091.071	1,279.976	1,545.823
	14	3,345.564	3,119.516	3,192.456	2,287.762	2,728.098	3,267.091
	15	7,161.078	6,683.296	6,830.675	4,843.656	5,850.231	6,972.194
	16	14,980.693	13,981.411	14,288.715	10,160.761	12,358.874	14,599.862
	17	31,293.082	29,090.231	29,723.927	21,186.236	25,969.206	30,511.422
	18	66,187.974	60,941.887	62,386.458	45,177.687	56,019.756	64,773.333
	19	145,642.424	133,354.121	136,472.180	96,765.463	121,741.789	142,447.095
	20	298,561.239	271,091.866	277,545.708	203,310.439	249,372.527	290,179.592
Rana64	10	41.223	37.963	39.010	33.271	33.213	38.859
	11	87.162	79.163	81.538	74.004	70.755	82.854
	12	183.042	160.251	166.237	158.827	146.886	174.551
	13	382.700	328.796	342.310	331.373	301.172	362.074
	14	730.466	651.903	681.998	657.890	595.441	730.466
	15	1,603.120	1,329.902	1,394.179	1,320.593	1,215.405	1,499.345
	16	3,290.745	2,655.900	2,788.659	2,563.864	2,438.181	3,034.799
	17	6,745.447	5,452.851	5,722.298	5,038.760	4,979.091	6,205.151
	18	15,435.268	12,456.546	13,014.861	11,113.052	11,695.514	14,652.750
	19	36,052.455	29,731.352	30,943.295	24,488.000	27,725.699	34,437.478
	20	81,105.131	67,088.578	69,694.729	53,684.172	63,394.399	77,259.420
M82	10	53.170	41.200	45.653	42.694	43.772	50.834
	11	106.443	75.234	85.710	77.799	82.608	101.633
	12	225.836	160.378	181.823	171.660	177.903	215.831
	13	543.127	394.340	438.867	390.489	419.419	521.854
	14	1,238.322	878.301	972.214	862.604	940.812	1,182.460
	15	2,515.808	1,757.127	1,956.781	1,782.431	1,898.903	2,358.434
	16	5,098.459	3,470.203	3,886.246	3,602.747	3,860.230	4,837.115
	17	10,483.717	6,911.314	7,779.587	7,364.237	7,996.127	9,985.172
	18	21,508.330	13,403.968	15,299.374	15,001.009	16,261.250	20,318.424
	19	43,997.450	26,220.471	30,240.065	30,063.243	32,637.547	41,198.996
	20	89,431.696	50,805.935	59,127.952	57,422.867	63,356.787	82,092.395

Table 3
The continuation of Table 4.

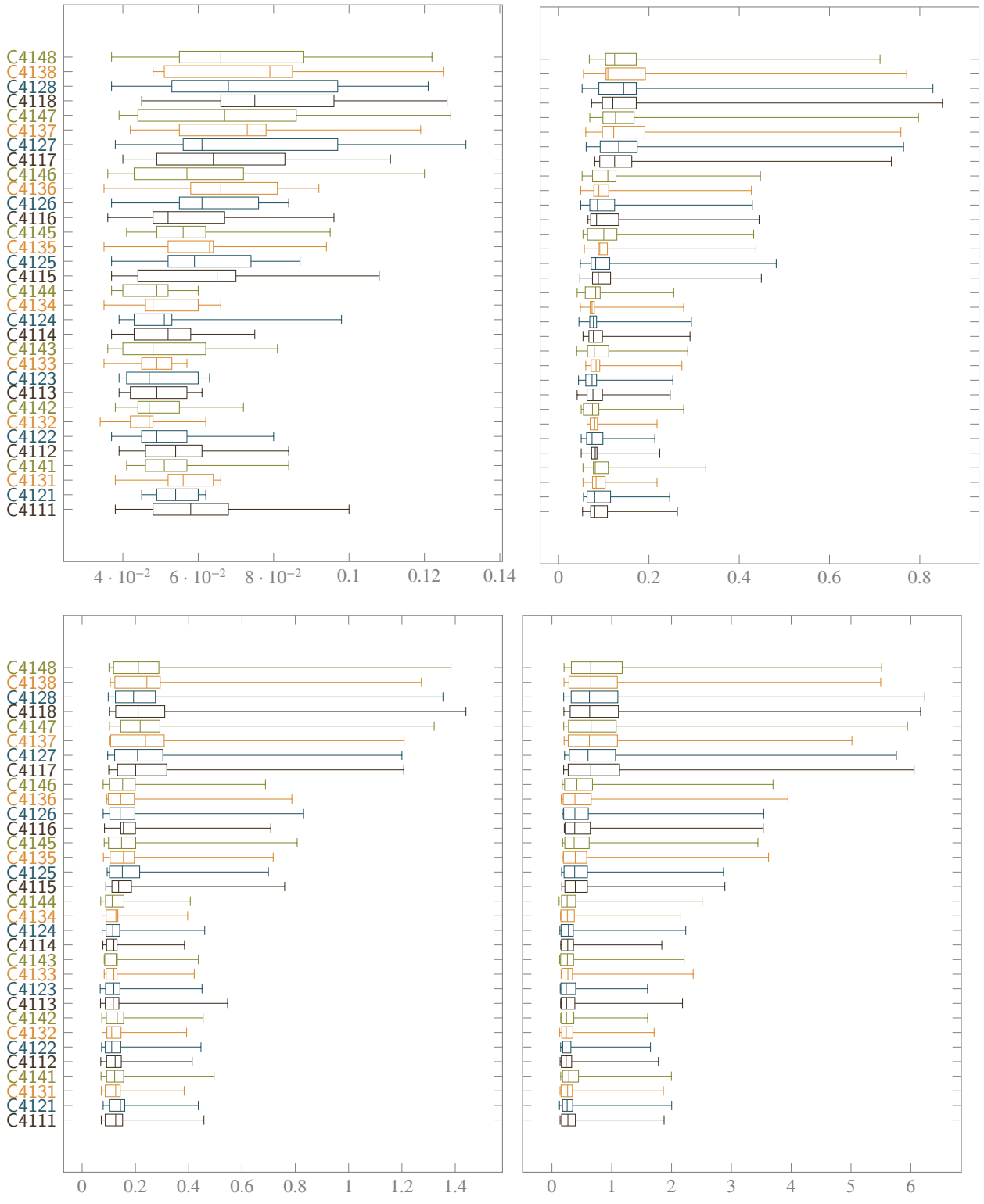


Figure 15: A boxplot of the runtime in seconds for the 32 algorithms EPSAS-C4111 to EPSAS-C4148 for $n = 10$ (upper left), $n = 11$ (upper right), $n = 12$ (lower left) and $n = 13$ (lower right).

Figure 16: A boxplot of the runtime in seconds for the 32 algorithms EPSAS-C4111 to EPSAS-C4148 for $n = 14$ (upper left), $n = 15$ (upper right), $n = 16$ (lower left) and $n = 17$ (lower right).

Figure 17: A boxplot of the runtime in seconds for the 32 algorithms EPSAS-C4111 to EPSAS-C4148 for $n = 18$ (upper left), $n = 19$ (upper right) and $n = 20$ (bottom).

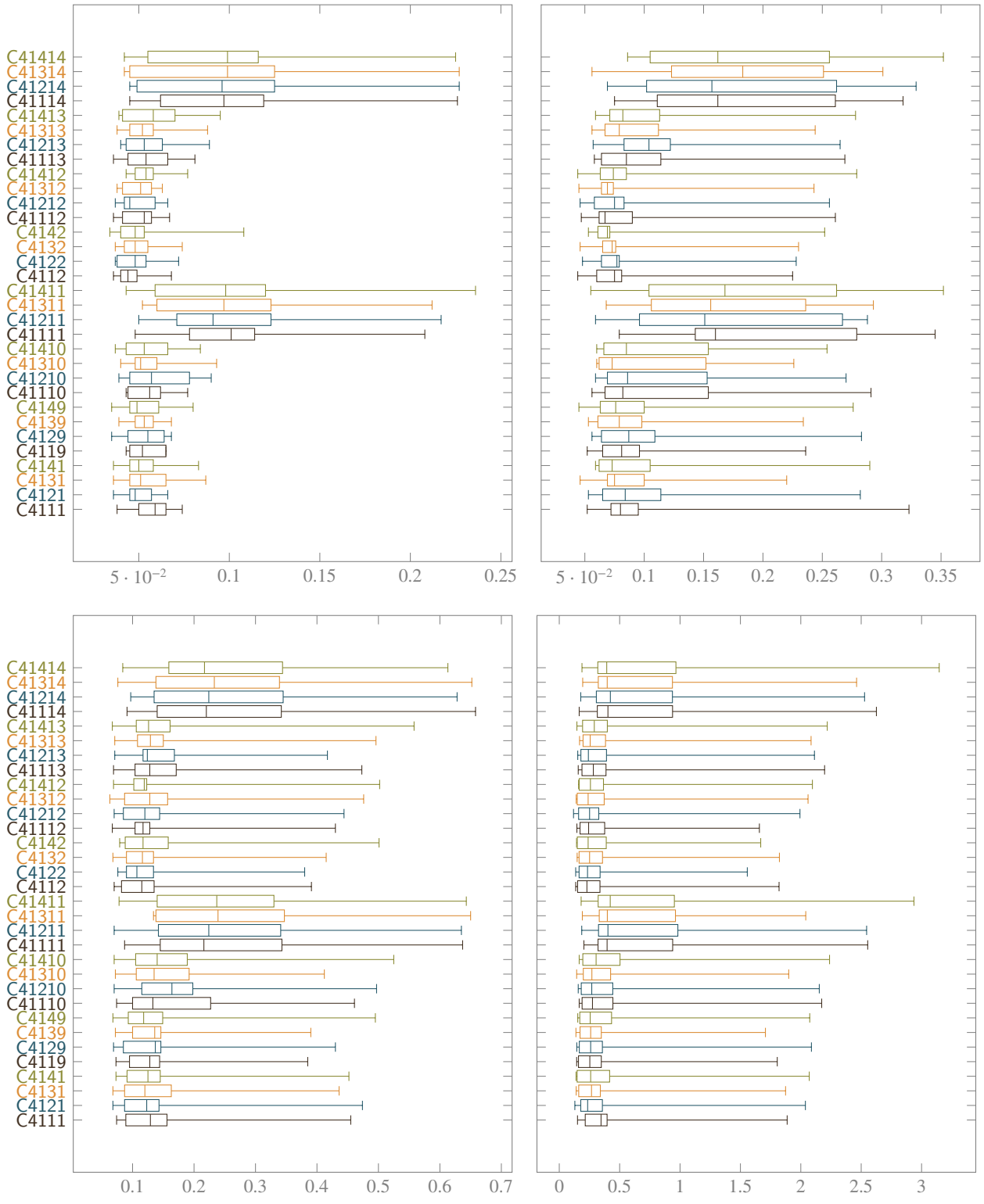


Figure 18: A boxplot of the runtime in seconds for the 32 algorithms EPSAS-C4119 to EPSAS-C41416 for $n = 10$ (upper left), $n = 11$ (upper right), $n = 12$ (lower left) and $n = 13$ (lower right).

Figure 19: A boxplot of the runtime in seconds for the 32 algorithms EPSAS-C4119 to EPSAS-C41416 for $n = 14$ (upper left), $n = 15$ (upper right), $n = 16$ (lower left) and $n = 17$ (lower right).

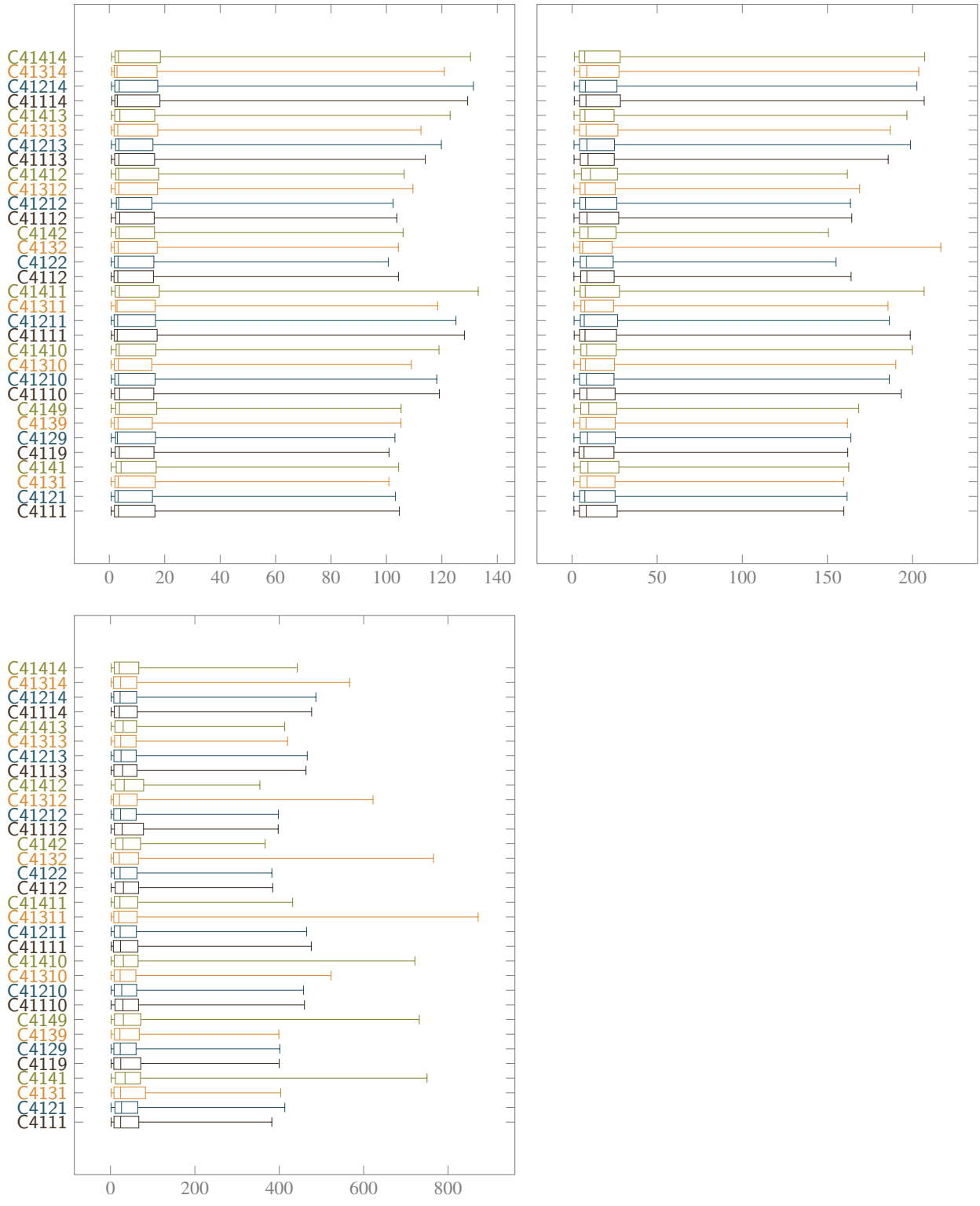


Figure 20: A boxplot of the runtime in seconds for the 32 algorithms EPSAS-C4119 to EPSAS-C41416 for $n = 18$ (upper left), $n = 19$ (upper right) and $n = 20$ (bottom).

				Single Core		Multi Core		
Data set	Taxa	Optimum	Gap (%)	Time (s)	Branchings	Time (s)	Branchings	Improvement (%)
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 4
Numerical results obtained for the algorithm EPSAS-C41412 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 the EPSAS-C41412 on one core. Bold indicates the approach that performed the best.

				Single Core		Multi Core		
Data set	Taxa	Optimum	Gap (%)	Time (s)	Branchings	Time (s)	Branchings	Improvement (%)
RbcL55	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
	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
M62	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
	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
Rana64	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
	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
M82	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
	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 5
The continuation of Table 4.

Data set	Taxa	Optimum	Gap (%)	Time (s)	Branchings
SeedPlant25	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
	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.
M43	20	155,142.306	1.48	9.23	454,778
	21	320,726.566	1.45	11.49	531,549
	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.
RbcL55	20	225,473.659	2.96	65.77	3,846,795
	21	480,122.632	2.79	130.04	7,032,882
	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.
M62	20	290,179.592	2.18	11.43	589,755
	21	576,663.719	4.87	13.07	563,435
	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.
Rana64	20	77,259.420	4.74	1.40	42,519
	21	179,623.137	3.48	3.47	127,983
	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.
M82	20	82,092.395	5.77	359.83	20,752,303
	21	169,136.637	5.43	852.38	44,877,124
	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 6
continuation

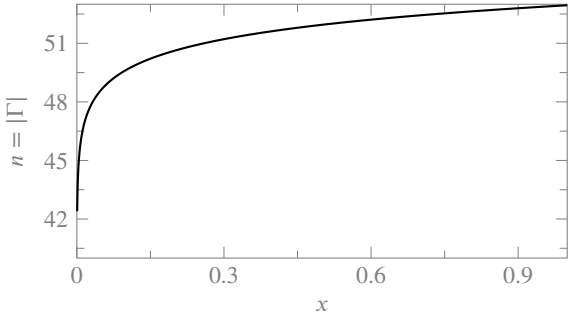


Figure 21: Behavior of the function $\log_2\left(\frac{x}{\epsilon}\right) + 1$ for $0 < x \leq 1$.

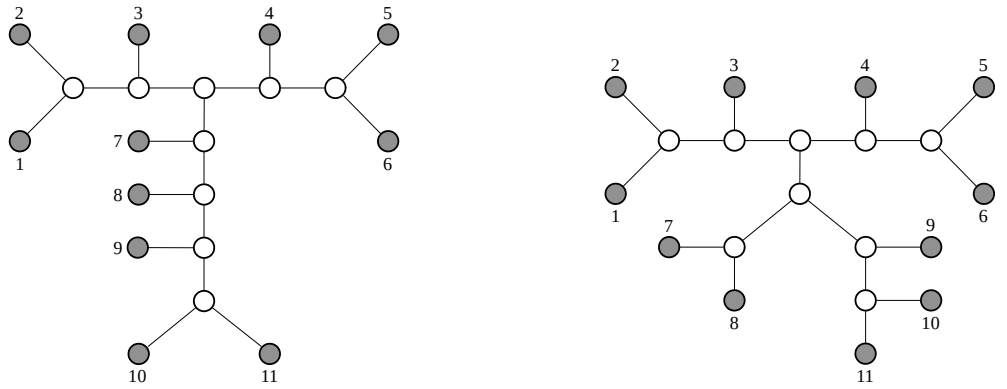


Figure 22: Two different phylogenies T (left) and T' (right) of 11 taxa.

numerical instabilities occur when solving an instance of the BMEP. A trivial algebraic manipulation of (57) provides the following answer: numerical instabilities occur whenever

$$n \geq \left\lfloor \log_2 \left(\frac{d_{ij}}{\epsilon} \right) + 1 \right\rfloor. \tag{58}$$

Now, in 64-bit floating point arithmetic, $\epsilon \approx 2.3 \cdot 10^{-16}$; then, if the smallest non-diagonal entry of $\mathbf{D}$ is at most 1, the BMEP becomes numerically instable for

$$n \geq \left\lfloor \log_2 \left(\frac{1}{2.3 \cdot 10^{-16}} \right) + 1 \right\rfloor = 53. \tag{59}$$

As shown in Figure 21, this bound can become even smaller in function of the smallest non-diagonal entry of $\mathbf{D}$.

To overcome numerical instabilities it would be desirable to pull away as far as possible the bound on n for which numerical stability issues arise. One way to achieve this goal consists of finding a scalar $\alpha > 1$ such that

$$n \geq \left\lfloor \alpha \cdot \log_2 \left(\frac{d_{ij}}{\epsilon} \right) + 1 \right\rfloor \tag{60}$$

may be as large as possible. Imposing (60) means considering the following new length function for the BMEP:

$$L_\alpha(T) = \sum_{i \in \Gamma} \sum_{j \in \Gamma \setminus \{i\}} \frac{d_{ij}}{2^{\tau_{ij}/\alpha}}. \tag{61}$$

We observe that any rescaling of the length function (1) can be written in the form of (61) and we refer to the problem of minimizing (61) as the *Rescaled Balanced Minimum Evolution Problem* (RBMEP). An interesting property of (61) is that in some circumstances it allows to characterize the optimal solution to the rescaled problem, as shown by the following result:

Proposition 7. *For a sufficiently large scalar $\alpha > 1$, there exists an input distance matrix $\mathbf{D}$ for which the caterpillar phylogeny is an optimal solution to the RBMEP.*

Proof. Assume $n = 11$ and first observe that

$$\lim_{\alpha \to \infty} L_\alpha(T) = \frac{n(n-1)}{2} - \sum_{i \in \Gamma} \sum_{j \in \Gamma \setminus \{i\}} \tau_{ij} d_{ij}.$$

Then, for sufficiently large α, the phylogeny T that maximizes

$$\sum_{i \in \Gamma} \sum_{j \in \Gamma \setminus \{i\}} \tau_{ij} d_{ij}$$

is optimal for the RBMEP. Now, observe that, by the Cauchy-Schwarz inequality ([15, 47]), the following inequality holds

$$\sum_{i\in\Gamma}\sum_{j\in\Gamma\setminus\{i\}}\tau_{ij}d_{ij} \leq ||\tau||_F \cdot ||D||_F, \tag{62}$$

where τ is the matrix of path-lengths and $||\cdot||_F$ is the Frobenius norm of a matrix ([36]). Hence, the phylogeny T with maximal $||\tau||_F$ is optimal to the RBMEP. Since we assumed $n = 11$, the maximum $||\tau||_F^2 = 4,064$ is achieved by the caterpillar phylogeny. □

Proposition 7 shows us that there exist scalars $\alpha > 1$ for which the optimal solution to the RBMEP can be identified independently of a given input distance matrix $\mathbf{D}$. This is not the case for the BMEP. Hence, a licit question is whether the new length function (61) allows to preserve the statistical consistency of the optimal solution to the RBMEP for sufficiently small $\alpha > 1$. Unfortunately, the next propositions provide a negative answer.

Proposition 8. *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 $\mathbf{D}$.*

Proof. As shown in Desper and Gascuel [22], $L(T)$ is a consistent estimation of the length of the true phylogeny associated to $\mathbf{D}$. Then, the statement follows by observing that $2^{-\tau_{ij}/\alpha} > 2^{-\tau_{ij}}$ holds true for any scalar $\alpha > 1$, hence $L_\alpha(T) > L(T)$, for any phylogeny T of Γ. □

Proposition 9. *There exists an input distance matrix $\mathbf{D}$ for which the RBMEP is not statistically consistent, for any $\alpha \geq 1.2447$.*

Proof. Assume $n = 11$ and consider the phylogenies T and T', shown in Figure 22, and the associated path-length matrices τ and τ', respectively, shown in Figure 23. 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 RBMEP. 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 9 still leaves some hope to find a scalar $1 < \alpha < 1.2447$ for which the RBMEP 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 up to

$$n \geq \left\lfloor 1.2446 \cdot \log_2\left(\frac{1}{2.3 \cdot 10^{-16}}\right) + 1 \right\rfloor = 65, \tag{63}$$

a value that is not suited for large scale phylogenetic analyses involving hundreds or thousands 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 RBMEP for $\alpha = 2$ are shown as T and T' in Figure 24, 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 23: Two path-length matrices τ and τ' encoding the two distinct phylogenies T and T' of Figure 22, respectively.

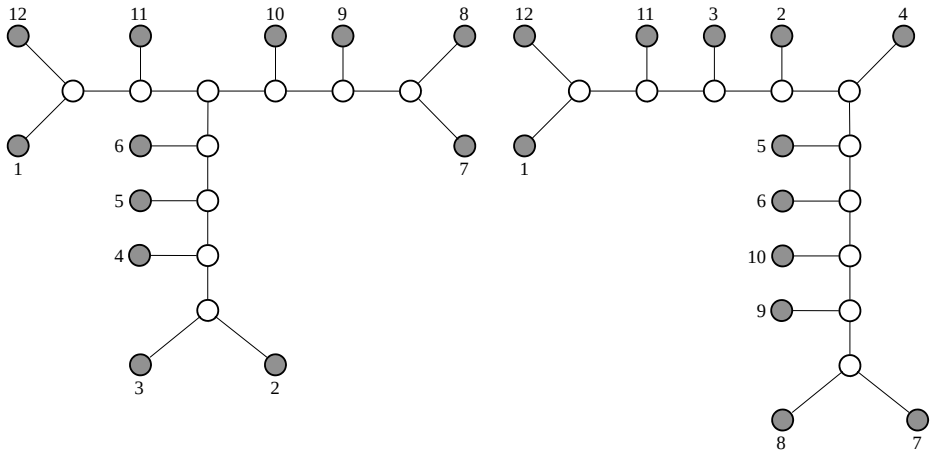


Figure 24: The optimal solutions T (left) and T' (right) to the BMEP and the RBMEP for $\alpha = 2$, respectively, for the instance "Primates12" of 12 taxa.

Acknowledgments

The first author acknowledges support the Université Catholique de Louvain via the "Fonds Spéciaux de Recherche" (FSR) 2017-2021, and the Fondation Louvain via the grant "COALESCENS". 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] Aringhieri, R., Catanzaro, D., Di Summa, M., 2011. Optimal solutions for the balanced minimum evolution problem. Computers and Operations Research 38, 1845–1854.
[2] 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.
[3] 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.
[4] 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.
[5] 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.
[6] Caminiti, S., Finocchi, I., Petreschi, R., 2007. On coding labeled trees. Theoretical Computer Science 382, 97–108.
[7] Catanzaro, D., 2011. Estimating phylogenies from molecular data, in: Bruni, R. (Ed.), Mathematical approaches to polymer sequence analysis and related problems. Springer, NY, pp. 149–176.
[8] 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.
[9] Catanzaro, D., Frohn, M., Gascuel, O., Pesenti, R., submitted 2020. A tutorial on the balanced minimum evolution. European Journal of Operational Research .
[10] Catanzaro, D., Frohn, M., Pesenti, R., 2020a. An information theory perspective on the balanced minimum evolution problem. Operations Research Letters 48, 362–367.
[11] Catanzaro, D., Labbé, M., Pesenti, R., Salazar-Gonzáles, J.J., 2012. The balanced minimum evolution problem. INFORMS Journal on Computing 24, 276–294.
[12] Catanzaro, D., Pesenti, R., 2019. Enumerating vertices of the balanced minimum evolution polytope. Computers and Operations Research 109, 209–217.
[13] 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.
[14] Catanzaro, D., Pesenti, R., Wolsey, L.A., 2020b. On the Balanced Minimum Evolution polytope. Discrete Optimization 36, 1–33.
[15] Cauchy, A.L., 1821. Cours d'Analyse de l'École Royale Polytechnique. L'Imprimerie Royale, Debure frères, Libraires du Roi et de la Bibliothèque du Roi.
[16] Çela, E., 1997. The Quadratic Assignment Problem. Kluwer Academic Publishers, Boston, MA.
[17] Cheng, X., Du, D.Z., 2001. Steiner Trees in Industry. Kluwer Academic Publishers, Boston, MA.
[18] Cieslik, D., 1998. Steiner minimal trees. Springer, Boston, MA, USA.

[19] Criscuolo, A., Michel, C.J., 2009. Phylogenetic inference with weighted codon evolutionary distances. Journal of Molecular Evolution 68, 377–392.
[20] Cueto, M.A., Matsen, F.A., 2011. Polyhedral geometry of phylogenetic rogue taxa. Bulletin of Mathematical Biology 73, 1202–1226.
[21] Desper, R., Gascuel, O., 2002. Fast and accurate phylogeny reconstruction algorithms based on the minimum evolution principle. Journal of Computational Biology 9, 687–705.
[22] 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.
[23] Desper, R., Gascuel, O., 2008. Encyclopedia of Algorithms. Springer. chapter Distance-based Phylogeny Reconstruction (Optimal Radius). pp. 1–99.
[24] Du, D.Z., Hu, X., 2008. Steiner tree problems in computer communication networks. World Scientific Publishing Company, Singapore.
[25] Du, D.Z., Smith, J.M., Rubinstein, J.H., 2000. Advances in Steiner trees. Kluwer Academic Publishers, Boston, MA, USA.
[26] 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.
[27] Felsenstein, J., 2004. Inferring phylogenies. Sinauer Associates, Sunderland, MA.
[28] Fiorini, S., Joret, G., 2012. Approximating the balanced minimum evolution problem. Operations Research Letters 40, 31–35.
[29] Forcey, S., Keefe, L., Sands, W., 2015. Facets of the balanced minimal evolution polytope. Journal of Mathematical Biology 73, 447–468.
[30] 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.
[31] Frohn, M., 2020. On the approximability of the fixed-tree balanced minimum evolution problem. To appear in Optimization Letters .
[32] 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.
[33] Gascuel, O., 2005. Mathematics of evolution and phylogeny. Oxford University Press, New York, NY.
[34] Gascuel, O., Steel, M., 2016. A 'stochastic safety radius' for distance-based tree reconstruction. Algorithmica 74, 1386–1403.
[35] Gascuel, O., Steel, M.A., 2006. Neighbor-joining revealed. Molecular Biology and Evolution 23, 1997–2000.
[36] Golub, G., van Loan, C., 1996. Matrix computations. The Johns Hopkins University Press, London.
[37] Gusfield, D., 1984. The Steiner Tree Problem in Phylogeny. Technical Report 334. Yale University, New Haven, CT.
[38] 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.
[39] Hendy, M.D., Penny, D., 1982. Branch and bound algorithms to determine minimal evolutionary trees. Mathematical Biosciences .
[40] Hwang, F.K., Richards, D.S., Winter, P., 1992. The Steiner tree problem. North-Holland, Amsterdam, The Netherlands.
[41] Johnson, D.S., Lenstra, J.K., Kan, A.H.G.R., 1978. The complexity of the network design problem. Networks 8, 279–285.
[42] Jung, M., 2012. Évolution du VIH: méthodes, modèles et algorithmes. Ph.D. thesis. Université Montpellier II - Sciences et Techniques du Languedoc, France.
[43] Kreher, D.L., Stinson, D.R., 1999. Combinatorial algorithms: Generation, enumeration, and search. CRC Press, Boca Raton, FL.
[44] 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.
[45] Li, X., Mao, Y., 2016. Generalized connectivity of graphs. Springer, Boston, MA, USA.
[46] Lu, C.L., Tang, C.Y., Lee, R.C.T., 2003. The full Steiner tree problem. Theoretical Computer Science 306, 55–67.
[47] Mitrinović, D.S., Pečarić, J.E., Fink, A.M., 1993. Classical and new inequalities in analysis. Kluwer Academic Publishers Group, Dordrecht.
[48] Pardi, F., 2009. Algorithms on Phylogenetic Trees. Ph.D. thesis. University of Cambridge, UK.
[49] Pardi, F., Gascuel, O., 2012. Combinatorics of distance-based tree inference. PNAS 109, 16443–16448.
[50] Pardi, F., Gascuel, O., 2016. Encyclopedia of Evolutionary Biology. Elsevier. chapter Distance-based methods in phylogenetics. pp. 458–465.
[51] Pardi, F., Guillemot, S., Gascuel, O., 2010. Robustness of phylogenetic inference based on minimum evolution. Bulletin of Mathematical Biology 72, 1820–1839.
[52] 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.
[53] Pauplin, Y., 2000. Direct calculation of a tree length using a distance matrix. Journal of Molecular Evolution 51, 41–47.
[54] Pop, P.C., 2012. Generalized network design problems: Modeling and optimization. De Gruyter, Berlin, Germany.
[55] Prömel, H.J., Steger, A., 2002. The Steiner tree problem: A tour through graphs, algorithms, and complexity. Vieweg+Teubner Verlag, Berlin.
[56] Reinhard, D., 2005. Graph Theory. Springer-Verlag, New York, NY.
[57] Saitou, N., Nei, M., 1987. The neighbor-joining method: A new method for reconstructing phylogenetic trees. Molecular Biology and Evolution 4, 406–425.
[58] Schwartz, R., 2019. Computational models for cancer phylogenetics, in: Warnow, T. (Ed.), Bioinformatics and Phylogenetics. Computational Biology. Springer, Cham. volume 29, pp. 243–275.
[59] Semple, C., Steel, M.A., 2004. Cyclic permutations and evolutionary trees. Advances in Applied Mathematics 32, 669–680.
[60] 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.
[61] Waterman, M.S., Smith, T.F., Singh, M., Beyer, W.A., 1977. Additive evolutionary trees. Journal of Theoretical Biology 64, 199–213.
[62] Wu, B.Y., Chao, K.M., 2004. Spanning trees and optimization problems. Chapman and Hall/CRC, Boca Raton, FL.
[63] Yang, Z., 2014. Molecular Evolution: A Statistical Approach. Oxford University Press, Oxford, UK.