

# The Central Spanning Tree Problem

Enrique Fita Sanmartín<sup>1</sup>, Christoph Schnörr<sup>1 2</sup>, Fred A. Hamprecht<sup>1</sup>

<sup>1</sup>IWR, <sup>2</sup>IPA at Heidelberg University, 69120 Heidelberg, Germany

{enrique.fita.sanmartin@iwr,schnoerr@math fred.hamprecht@iwr}.uni-heidelberg.de

## Abstract

Spanning trees are an important primitive in many data analysis tasks, when a data set needs to be summarized in terms of its “skeleton”, or when a tree-shaped graph over all observations is required for downstream processing. Popular definitions of spanning trees include the minimum spanning tree and the optimum distance spanning tree, a.k.a. the minimum routing cost tree. When searching for the shortest spanning tree but admitting additional branching points, even shorter spanning trees can be realized: Steiner trees. Unfortunately, both minimum spanning and Steiner trees are not robust with respect to noise in the observations; that is, small perturbations of the original data set often lead to drastic changes in the associated spanning trees. In response, we make two contributions when the data lies in a Euclidean space: on the theoretical side, we introduce a new optimization problem, the “(branched) central spanning tree”, which subsumes all previously mentioned definitions as special cases. On the practical side, we show empirically that the (branched) central spanning tree is more robust to noise in the data, and as such is better suited to summarize a data set in terms of its skeleton. We also propose a heuristic to address the NP-hard optimization problem, and illustrate its use on single cell RNA expression data from biology and 3D point clouds of plants.

## 1 Introduction

Many data analysis tasks call for the summary of a data set in terms of a spanning tree, or use tree representations for downstream processing. Examples include the inference of trajectories in developmental biology [30, 10], generative modeling in chemistry [1], network design [42] or skeletonization in image analysis [2, 39]. The problem is akin to, but more complicated than, the estimation of principal curves because good recent methods such as [24] cannot account for branched topologies. For a spanning tree representation to be meaningful, it is of paramount importance that the tree structure be robust to minor perturbations of the data, e.g. by measurement noise. In this work, we address the geometric stability of spanning trees over points lying in an Euclidean space.

The minimum spanning tree (mST) is surely the most popular spanning tree, owing to its conceptual simplicity and ease of computation. For a graph  $G = (V, E)$  with edge costs, the mST is a tree that spans  $G$  while minimizing the total sum of edge costs. It prioritizes shorter edges that connect closely located nodes enhancing data faithfulness. Unfortunately, its

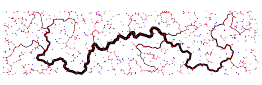
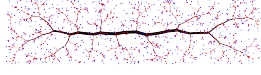
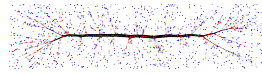
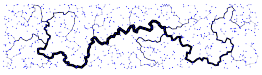
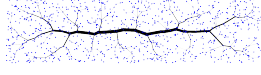
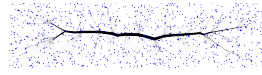
		Edge Centrality Exponent $\alpha$		
		0	(0, 1)	1
Add Steiner points	SP			
	YES	<b>Steiner Tree</b> $\min_{X_{B,T}} \sum_{(i,j) \in E_T}^{[11]} \ x_i - x_j\ $ 	<b>Branched Central Spanning Tree</b> $\min_{X_{B,T}} \sum_{(i,j) \in E_T} (m_{ij}(1 - m_{ij}))^\alpha \ x_i - x_j\ $ 	<b>Branched Minimum Routing Cost Tree</b> $\min_{X_{B,T}} \sum_{i,j \in V \times V} d_T(i, j)$ 
NO	<b>Minimum Spanning Tree</b> $\min_T \sum_{(i,j) \in E_T}^{[21]} \ x_i - x_j\ $ 	<b>Central Spanning Tree</b> $\min_T \sum_{i,j \in E_T} (m_{ij}(1 - m_{ij}))^\alpha \ x_i - x_j\ $ 	<b>Minimum Routing Cost Tree</b> $\min_T \sum_{i,j \in V \times V}^{[26]} d_T(i, j)$ 	

Figure 1: **Euclidean Central Spanning Tree Family of Problems with and without Steiner Points.** The central spanning tree weighs the costs of the edges, given by node distances, with the centrality of the edges,  $m_{ij}(1 - m_{ij})$ . The influence of the centrality is regulated by the parameter  $\alpha \in [0, 1]$ . For lower  $\alpha$  values the centrality becomes insignificant, and the tree tends to contain short edges. For higher  $\alpha$  values, the tree encourages central edges of low cost, at the expense of peripheral edges of higher cost. We study central spanning trees with and without additional Steiner points (shown in red). The widths of all edges are proportional to their centrality. The central spanning tree problems includes well-known and novel spanning tree problems, the latter highlighted in green.

greedy nature makes the mST susceptible to small data perturbations that may lead to drastic changes in its structure, see Figure 4. An alternative, the minimum routing cost tree (MRCT), minimizes the sum of pairwise shortest path distances [26]. Unlike the mST, solving the MRCT is NP-hard [43]. Despite this, the MRCT exhibits a more stable geometric structure compared to the mST, as it tends to be more "star-shaped" (see Figure 4). Nevertheless, this star-shaped tendency inclines towards connecting nodes that are spatially distant, introducing a risk of information loss and compromised data fidelity. This effect becomes particularly pronounced in high-dimensional spaces, potentially rendering the MRCT approach unusable (see Figures 5i-5l)). Achieving a balance between data fidelity and geometric robustness is crucial for an effective spanning tree.

**Central spanning trees (CST)** In this paper, we propose a novel parameterized family of spanning trees that interpolate and generalize all the aforementioned ones. Unless otherwise stated, we will assume a complete Euclidean graph, where the nodes are embedded in the Euclidean space with coordinates  $X_V = \{x_1, \dots, x_{|V|}\} \subset \mathbb{R}^n$  and the edge costs are given by the distances between the vertices,  $c_{ij} = \|x_i - x_j\|$ . We define the CST as the spanning tree of  $G$  that minimizes the objective

$$\text{CST} := \arg \min_T \sum_{e \in E_T} (m_e(1 - m_e))^\alpha c_e = \arg \min_T \sum_{(i,j) \in E_T} (m_{ij}(1 - m_{ij}))^\alpha \|x_i - x_j\|, \quad (1)$$



where  $m_e$  and  $(1 - m_e)$  are the normalized cardinalities of the components resulting from the removal of the edge  $e$  from  $T$ . Thus,  $m_e(1 - m_e)$  is the product of the number of nodes on both sides of the edge divided by  $|V|^2$ . Because this value is proportional to the “edge betweenness centrality”<sup>1</sup> of  $e$  in  $T$  [6], we call the problem the “central spanning tree” (CST) problem. The exponent  $\alpha$  is the interpolating parameter that modulates the effect of the edge centrality. For  $\alpha$  close to 0, the centrality becomes insignificant, so the tree tends to contain lower cost edges overall. For  $\alpha = 0$  we retrieve the mST. On the other hand, as  $\alpha$  increases, the centrality becomes more relevant, leading the tree to favor topologies with low centrality edges, thus promoting a higher branching effect. For  $\alpha = 1$ , the resulting expression is proportional to the MRCT. Here, each edge cost is multiplied by the number of shortest paths it belongs to, leading the total sum to represent the sum of shortest path distances (see Appendix D).

As will be seen in Section 3, the  $\alpha$  parameter has an effect on the geometric stability of the spanning tree, with higher  $\alpha$  resulting in greater robustness. The robustness of spanning trees has been explored in various contexts. For instance, researchers have investigated the robustness of the mST cost under edge weight uncertainty [19, 35] and the robustness against node or edge failure in networks [23]. Additionally, studies have delved into the stability regions of mST, under which any change in vertex location does not alter the mST [27, 28]. The central tree problem [5], related by name to ours, focuses on computing a tree that minimizes the maximal distance to a set of given trees. To our knowledge, we are the first to propose a spanning tree whose geometric structure is stable and resilient to data perturbations such as noise.

Finally, we remark the connection between the the CST problem and the Minimum Concave Cost Network Flow (MCCNF) problem [44, 12]. The MCCNF problem aims to minimize the cost of distributing a certain commodity from source to sink nodes. Such a problem models the cost of an edge as a concave function of the transportation flow. The CST can be reinterpreted as MCCNF where a commodity with mass equal to  $|V| - 1/|V|$ , concentrated into a single source node, must be transported to the rest of nodes. In our case, the term  $m_e$  in (1) can be interpreted as the flow of such problem. Since the function  $(m_e(1 - m_e))^\alpha c_e$  is concave with respect to  $m_e$  for  $\alpha \in [0, 1]$ , we deduce that the CST is an instance of the MCCNF. A more detailed discussion of the interpretation of the CST as a MCCNF problem is offered in Appendix C.

Considering the CST from the perspective of an MCCNF problem, it becomes clear that it falls into the NP-hard category. Indeed, the authors of [13] showed that single-source MCCNF problems with strictly concave functions are NP-hard. Consequently, we conclude that the CST problem is NP-hard for  $\alpha \in ]0, 1]$  due to the strictly concave nature of the edge cost function  $(m_e(1 - m_e))^\alpha c_e$ .<sup>2</sup>

**Branched central spanning trees (BCST)** Inspired by Lippmann et al. [22], we also study the variant of the CST problem which allows for the introduction of additional nodes, known in the literature as branching points or Steiner points (SPs). Formally, we

---

<sup>1</sup>The edge betweenness centrality measures an edge’s frequency in shortest paths between nodes, with more traversed edges being deemed more central. In trees, it’s the product of nodes on opposite sides of the edge.

<sup>2</sup>The same argument applies to the NP-hardness of the branched version of the CST problem, which is explained next.

distinguish between two types of nodes. On the one hand, we have the terminal nodes with fixed coordinates given by  $X_V$ . On the other hand, we allow for an extra set of points,  $B$ , whose coordinates  $X_B$  must be jointly optimized with the topology  $T$ . In this case,  $T$  is a spanning tree defined over the nodes  $V \cup B$ . Accordingly, the objective of the CST problem becomes

$$\min_{T, X_B} \sum_{(i,j) \in E_T} (m_{ij}(1 - m_{ij}))^\alpha \|x_i - x_j\|. \quad (2)$$

In this generalization, which we refer to as the branched CST (BCST), the well-known Steiner tree problem [15, 40] arises when  $\alpha = 0$ . Figure 1 summarizes (B)CST and its limiting cases, only some of which have been studied in the literature so far.

**Contributions.** 1) We present the novel (B)CST problem and provide empirical evidence for its greater stability on toy and real-world data; 2) We propose an iterative heuristic to estimate the optimum of the BCST problem.<sup>3</sup> By exploiting the connection between the branched and unbranched versions of the CST problem, we are able to use the heuristic defined for the BCST to also approximate the Euclidean CST without Steiner points. We benchmark this heuristic and show its competitiveness. 3) On the theoretical side, we prove that for large  $\alpha$  or large  $|V|$  and  $\alpha > 1$ , (B)CST converges to a star-tree (hinting modeling limitations when  $\alpha > 1$ ), and for  $\alpha \rightarrow -\infty$ , it tends towards a path graph. Additionally, we show analytically that if the terminal points lie on a plane, then for  $\alpha \in [0, 0.5] \cup \{1\}$  the Steiner points of the optimal solution can have up to degree 3, and we provide empirical evidence that this holds also for  $\alpha \in ]0.5, 1[$ .

**Outline** In Section 2, we explore the limiting cases of the (B)CST optimum as  $\alpha$  approaches  $\pm\infty$ , along with the scenario where the number of terminals tends to infinity for  $\alpha > 1$ . Section 3 demonstrates empirically the stability of the (B)CST as  $\alpha$  increases. In Section 4, we establish a relationship between feasible CST and BCST topologies. Section 5 analyzes the geometry of optimal BCST topologies, providing analytical expressions for the branching angles at the Steiner points. Moreover, it discusses the feasibility of 4-degree SPs when the BCST is restricted to the Euclidean plane. Section 6 presents a heuristic to approximate the (B)CST optimal solution, while Section 7 benchmarks this heuristic on small toy datasets. The conclusions of this work are given in section 8.

## 2 Limit Cases of the CST/BCST Problems Beyond the Range $\alpha \in [0, 1]$

The CST problem, as defined in (1), as well as its branched version are parameterized by  $\alpha$ . Throughout the manuscript our attention will be on the  $\alpha$ -range of  $[0, 1]$ , nonetheless it is worth studying the problem beyond this range. We will show that when  $\alpha \rightarrow \infty$  or the number of terminals approaches infinity when  $\alpha > 1$ , the (B)CST tends to a star graph centered on the medoid of the graph, i.e. the node that minimizes the distance to the rest of nodes. Consequently, the case with  $\alpha > 1$  becomes inadequate for modeling data structure, as the tree becomes increasingly trivial with a growing number of terminals. Conversely, as  $\alpha \rightarrow -\infty$ , the CST tends towards the path graph that minimizes the CST objective. These

---

<sup>3</sup>Code available at <https://github.com/sciai-lab/CST>

scenarios prove inadequate for modeling data structure, thus we restrict our focus exclusively to the  $\alpha$ -range of  $[0, 1]$ .

## 2.1 Limit Cases Where the Optimum (B)CST Transforms into a Star-Tree

In this section, we delve into scenarios at the limits where the optimal solutions for both the CST and BCST problems converge to a star-tree configuration. Specifically, we demonstrate that this outcome occurs as  $\alpha$  approaches infinity and as the number of terminals, denoted by  $N$ , tends to infinity when  $\alpha > 1$ . The later limit case is of special relevance, as it indicates that when the parameter  $\alpha$  exceeds 1, both CST and BCST exhibit inadequacy in extracting meaningful structural information from the data. In this situation, an increasing number of data points lead to the formation of a star-tree, which, lacks the capacity to convey pertinent information about the underlying dataset.

Before studying the limit cases, we will analyze the optimum star-tree that minimizes the CST and BCST costs. Note that in a star-tree, all edges are adjacent to a leaf node and, therefore all have the same normalized centrality value, which is equal to  $(N - 1)/N^2$ , where  $N$  is the number of terminals. In this scenario, the centrality of the edges in the cost function from Equation (1) can be factored out, simplifying the problem to the identification of the star graph with the minimum cost. Indeed, if  $u$  denotes the center node of a star graph, then its CST objective is equal to

$$\sum_{v \neq u} \frac{N - 1}{N^2} c_{uv} = \frac{N - 1}{N^2} \sum_{v \neq u} c_{uv}.$$

Consequently, the optimal solution for the CST problem manifests as a star graph centered at the node that minimizes the total distance to all other nodes, effectively the medoid. In the context of the BCST problem, where Steiner Points can be introduced, the star graph is centered at the geometric median. This is because the Steiner Points strategically position themselves to minimize the distance to all nodes. The next result formalizes this statement.

**Lemma 2.1.** The tree-star that minimizes the CST cost is the star-shaped tree centered at the medoid of the terminals, that is, centered at the terminal which minimizes the sum of distances to all nodes. For the BCST case, the tree is centered at the geometric median of all terminals.

As a consequence of this result, we infer that the limit cases wherein both CST and BCST converge to a star-tree will yield stable trees. This stability arises from the consistent output of star-trees, with their centers being the medoid and geometric median—both robust points resilient to noise.

In order to study limit cases where the star-tree emerges as the optimal solution, we establish first a sufficient condition for the CST optimal solution to take the form of a star tree. This condition was first identified by Hu [14] in the context of the Minimum Routing Cost Tree (MRCT), corresponding to the CST with  $\alpha = 1$ . Hu showed that if a “stronger” variant of the regular triangle inequality holds, then the optimum solution of the MRCT is a star tree. The following theorem extends and generalizes this result for arbitrary values of  $\alpha$ .

**Theorem 2.1.** Let  $N$  be the number of terminals and  $c_{ij}$  be the edge-costs of any pair of points (Steiner or terminals)  $i, j$ . If there exists

$$t \leq \min_{\ell \in [2, N/2]} \frac{\left(\frac{\ell(N-\ell)}{N-1}\right)^\alpha - 1}{\ell - 1}$$

such that

$$c_{kv} + tc_{uv} \geq c_{ku} \tag{3}$$

for all triplets of nodes  $u, k, v$ , then there exists an optimum (B)CST evaluated at  $\alpha$  which is a star tree.

We defer the complete proof to Appendix E.1, though we sketch briefly the key idea. The proof demonstrates that we can always iteratively increase the degree of certain node, by connecting all neighbors of one of its neighbors to it. If the “stronger” variant of the triangle inequality holds, then this process does not increase the cost. Eventually, a node will reach maximum degree, indicating the formation of a star-tree structure.

**Remark 2.1.** Note that Theorem 2.1 states only a sufficient condition, which means that the optimum can be a star tree even if the strong triangle inequality does not hold. Additionally, it is worth to highlight that Theorem 2.1 also holds true for the CST problem even when the nodes lack embedding in any specific space, allowing for edge costs with arbitrary values.

Let us define

$$h_1(\ell, N, \alpha) := \frac{\left(\left(\frac{\ell(N-\ell)}{N-1}\right)^\alpha - 1\right)}{\ell - 1} = \frac{\left(1 + \frac{(\ell-1)(N-\ell-1)}{N-1}\right)^\alpha - 1}{\ell - 1}. \tag{4}$$

which characterizes the upper limit of the threshold  $t$  in Theorem 2.1. Equation (3) represents a weighted version of the triangle inequality. Specifically, when  $t = 1$ , equation (3) recovers the standard triangle inequality. Moreover, if  $h_1(\ell, N, \alpha) \geq 1$ , there exists a value  $t$  such that  $h_1(\ell, N, \alpha) \geq t \geq 1$ . This implies that the relation is weaker than the triangle inequality. Therefore, if the triangle inequality holds, equation (3) will also hold. As a first consequence, is evident that

$$\lim_{\alpha \rightarrow \infty} h_1(\ell, N, \alpha) = \infty, \quad \forall N \geq 3, \ell \in [2, N/2].$$

This indicates that as  $\alpha$  tends to infinity, the optimal tree tends to become a star-tree. The intuition behind this is clear: as  $\alpha$  increases, the CST/BCST aims to minimize the centrality of the edges, since the edge costs become relatively insignificant in comparison. Among all edges in a tree, those adjacent to a leaf have the least centrality. Thus, any star graph is a tree that minimizes simultaneously the edge centrality of all its edges

One can also show that  $h_1$  is greater than 1 when  $\alpha > 1$  and  $N$  approaches infinity:

$$\lim_{N \rightarrow \infty} h_1(\ell, N, \alpha) \geq 1, \quad \forall \alpha > 1, \ell \in [2, N/2],$$

though we leave the technical proof for the Appendix E.2. Consequently, we obtain the following corollary

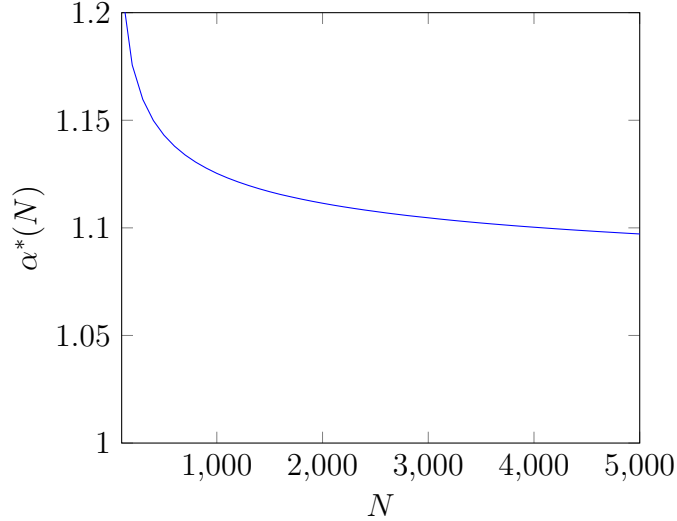


Figure 2: **Threshold Function  $\alpha^*(N)$  Guaranteeing Optimal (B)CST Star-Tree.** The threshold function  $\alpha^*(N)$ , tied to the number of terminals  $N$ , defines the minimum  $\alpha$  value ensuring the optimal solution for CST/BCST is a star-tree. For all  $\alpha > \alpha^*(N)$ , the optimum solution is a star-tree. The plot depicts the transition at which the optimum is ensured to be a star-tree is around  $\alpha \approx 1$ . As  $N$  increases,  $\alpha^*(N)$  approaches 1, implying that in the limit as  $N$  tends to infinity, CST/BCST with  $\alpha > 1$  converges to a star-tree.

**Corollary 2.1.** As the parameter  $\alpha$  approaches infinity, or  $N$  approaches infinity and  $\alpha > 1$  the CST/BCST optimal solution is a star-shaped tree.

Corollary 2.1 states that the optimum tree is a star-tree when  $\alpha > 1$  and  $N \rightarrow \infty$ . What’s intriguing is that this limit is reached at relatively low values of  $\alpha \approx 1$  for moderate values of  $N$ . In Appendix E.3, we show that for  $N$  nodes, the threshold  $\alpha^*(N)$  at which  $h_1(\ell, N, \alpha) \geq 1$  for all  $\ell \in [2, N/2]$  and  $\alpha \geq \alpha^*(N)$  is given by

$$\alpha^*(N) = \max \left( \frac{\log(2)}{\log \left( 1 + \frac{N-3}{N-1} \right)}, \frac{\log(N/2)}{\log \left( 1 + \frac{(N/2-1)^2}{N-1} \right)} \right) \quad (5)$$

The function  $\alpha^*(N)$  serves as a threshold, ensuring that the optimal solution adopts a star-tree configuration. To illustrate this threshold, Figure 2 depicts the function  $\alpha^*(N)$ . Indeed, we see that when  $N = 1000$ ,  $\alpha = 1.15$  is enough to guarantee that the optimum is a star tree. A toy example is presented in Figures 3, with  $N = 1000$ , showcasing an instance where the optimum is indeed a star tree.

**Remark 2.2.** The star-shaped tendency becomes more prominent in higher dimensions. Empirical observations indicate that relatively small values of  $\alpha$  ( $\alpha \lesssim 1$ ) tend to produce an almost star-shaped tree, which compromises the preservation of data structure. This behavior can be attributed to the curse of dimensionality, where the majority of point pairs in a high-dimensional Euclidean space become nearly equidistant. Consequently, the strong triangle inequality derived in Theorem 2.1 will hold for most triplets of points due to the

approximate equality of  $c_{ku} \approx c_{kv}$  in equation (3). Figures 5i-5l) exemplify this effect on a 50-dimensional dataset.

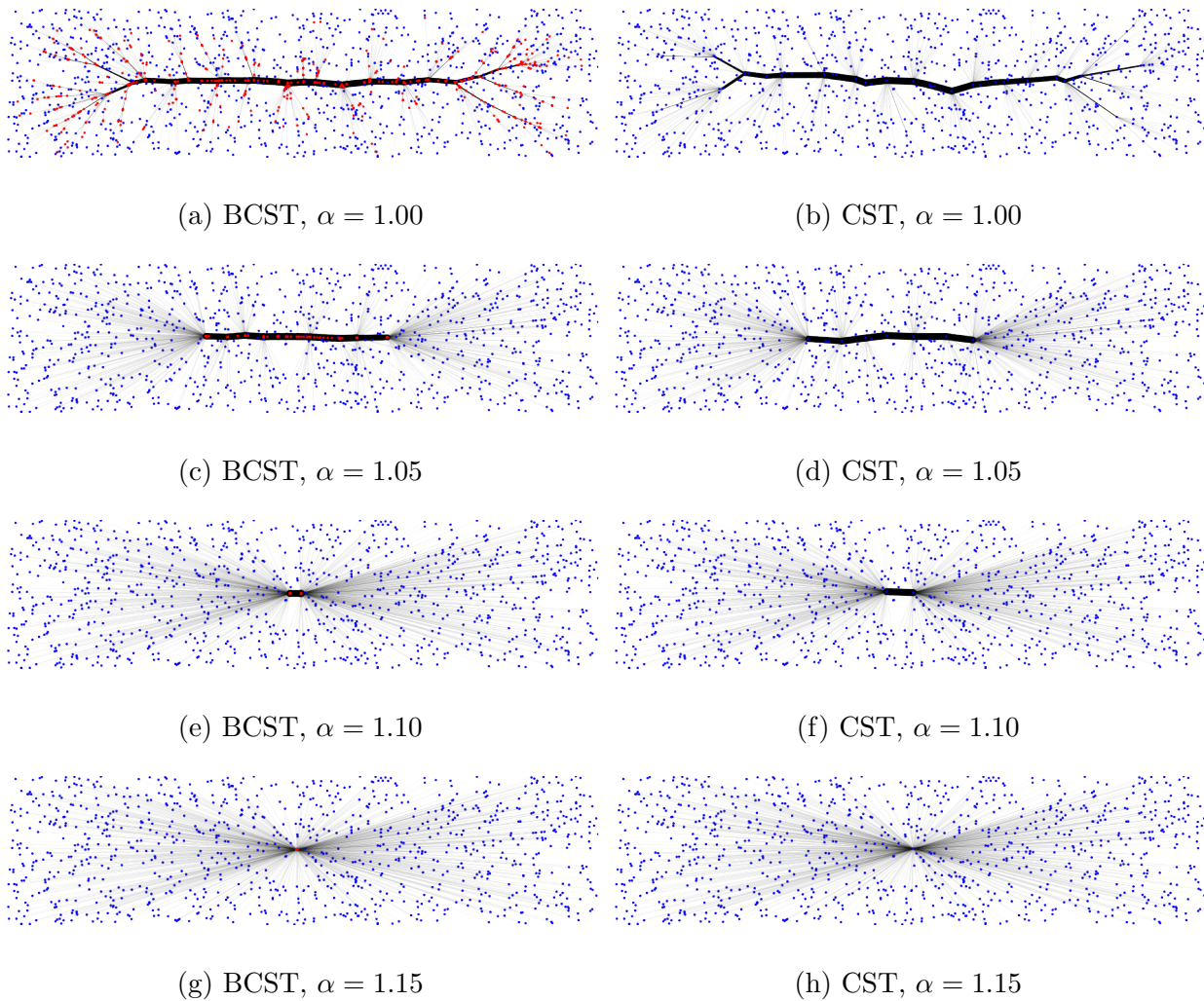


Figure 3: **(B)CST Star-Tree Optimality with Respect to  $\alpha \gtrsim 1$  in Samples Uniformly Drawn from a Rectangle.** As  $\alpha$  increases, both CST and BCST exhibit a transition towards a star graph. This effect may manifest relatively early. As anticipated in Figure 2, for a sample set with 1000 points, the value  $\alpha = 1.15$  transforms both CST and BCST into a star graph.

## 2.2 Limit Cases Where the Optimum (B)CST Transforms into a Path-Tree

Negative values of  $\alpha$  favour high central edges, as  $(m_e(1 - m_e))^\alpha$  will be lower. Consequently, for sufficiently negative values of  $\alpha$ , the CST/BCST problem will prioritize minimizing the number of leaves since the centrality of its adjacent edges attain the minimum centrality. The

tree that minimizes the number of leaves is a path. Therefore, when  $\alpha \rightarrow -\infty$  the optimum tree will be the Hamiltonian path that minimizes the CST/BCST objective function. A Hamiltonian path is a path that visits each node exactly once.

Echoing Theorem 2.1 we show that if a variant of the triangle inequality holds, then the optimum (B)CST will be a tree.

**Theorem 2.2.** Let  $N$  be the number of nodes and  $c_{ij}$  be the edge-costs of any pair of points (Steiner or terminals)  $i, j$ . If there exists

$$t \leq \min_{\substack{1 \leq s \leq N-3 \\ 1 \leq \ell \leq \min(s, (N-s)/2-1)}} \frac{(\ell(N-\ell))^\alpha - ((\ell+s)(N-\ell-s))^\alpha}{(s(N-s))^\alpha}$$

such that

$$c_{kv} + tc_{uv} \geq c_{ku}$$

for all triangles in the graph, then there exists an optimum (B)CST evaluated at  $\alpha$  which is a Hamiltonian path.

In contrast to the proof presented in Theorem 2.1, we demonstrate that we can systematically decrease the degree of nodes by iteratively connecting the neighbors of a specific node to one of its neighbors. This iterative reduction does not inflate the cost, provided the triangle inequality variant holds. Ultimately, all nodes will have at most degree 2, meaning that a path has been formed. We defer the complete proof to Appendix E.4.

Let

$$h_2(\ell, s, N, \alpha) = \frac{(\ell(N-\ell))^\alpha - ((\ell+s)(N-\ell-s))^\alpha}{(s(N-s))^\alpha}.$$

As a consequence of Theorem 2.2, if  $h_2(\ell, s) \geq 1$  for  $1 \leq s \leq N-3$  and  $1 \leq \ell \leq \min(s, (N-s)/2-1)$ , then the satisfaction of the triangle inequality is a sufficient condition to ensure that the optimal (B)CST topology is a path. We can easily check that as  $\alpha$  approaches minus infinity the following limit holds

$$\begin{aligned} \lim_{\alpha \rightarrow -\infty} h_2(\ell, s, N, \alpha) &= \lim_{\alpha \rightarrow -\infty} \left( \underbrace{\left( \frac{\ell(N-\ell)}{s(N-s)} \right)^\alpha}_{\leq 1} - \left( \underbrace{\frac{((\ell+s)(N-\ell-s))}{(s(N-s))}}_{>1} \right)^\alpha \right) \\ &= \begin{cases} 1, & \text{if } \ell = s \\ \infty, & \text{if } \ell > s \end{cases}, \end{aligned} \quad (6)$$

where we have used the inequalities  $1 \leq s \leq N-3$  and  $1 \leq \ell \leq \min(s, (N-s)/2-1)$ . Consequently, as  $\alpha$  approaches  $-\infty$ , the optimum tree will tend to a path tree. Note however, that if  $\ell = s$ , then  $h_2(\ell, \ell, N, \alpha) < 1$ . In this case, according to Theorem 2.2, the optimum tree can only be a path for  $\alpha$  negative enough, if the triangle inequality holds strictly for all triplets of nodes. Nonetheless, in Corollary E.1, we show that when the points lie on a geodesic space, then the requirement of the strict triangle inequality is not necessary for the optimum to become a path as  $\alpha$  approaches  $-\infty$ .

In the previous section, we demonstrated that for a given  $\alpha > 1$  we can find  $N$  large enough, such that for any configuration of terminals the optimal solution of the CST/BCST problem would be a star-tree. However, in this case, we cannot ensure that if  $\alpha < 0$ , there exists  $N$  large enough, where the optimum is a path. Indeed, given a fixed  $\alpha \in \mathbb{R}$  and setting  $\ell = N/4 - 1$  and  $s = N/2$ , we have that the limit of  $h_2(N/4 - 1, N/2, N, \alpha)$  as  $N$  increases is given by

$$\lim_{N \rightarrow \infty} h_2 \left( \frac{N}{4} - 1, \frac{N}{2}, N, \alpha \right) = \lim_{N \rightarrow \infty} \frac{\left( \left( \frac{N}{4} - 1 \right) \left( \frac{3N}{4} + 1 \right) \right)^\alpha - \left( \left( \frac{N}{4} + 1 \right) \left( \frac{3N}{4} - 1 \right) \right)^\alpha}{\left( \frac{N}{2} \right)^{2\alpha}} = 0 < 1.$$

Consequently, Theorem 2.2 will not guarantee that the optimum is a path as  $N$  increases, unless all edge-costs are equal.

### 3 Stability of the CST Problem

In this section we argue in favour of the greater robustness of the CST problem to small perturbations in data. We investigate first the robustness of the tree to noise in toy datasets. Next, we showcase two potential applications of the (B)CST problem: single cell trajectory inference and skeletonization of 3D point clouds of plants.

#### 3.1 Toy Data

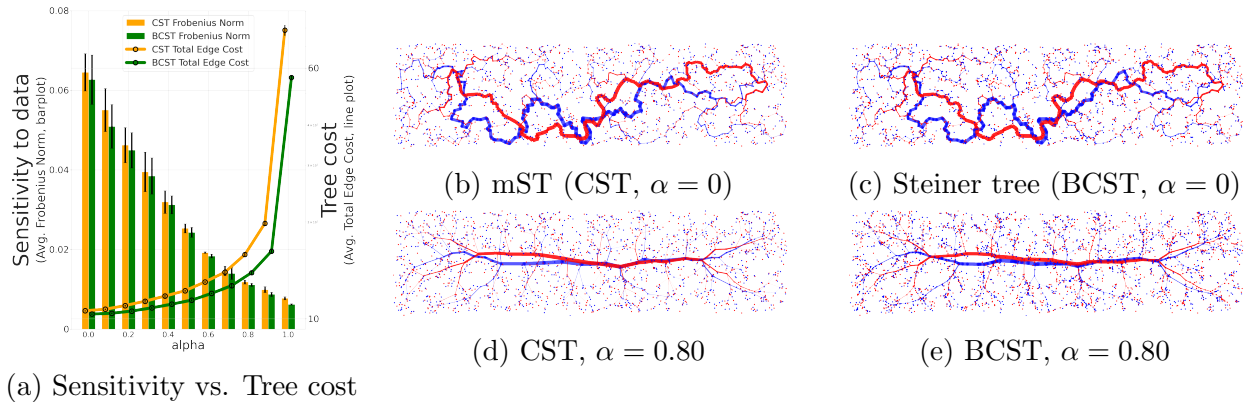


Figure 4: **(B)CST Robustness Analysis.** (B)CST for  $\alpha > 0$  are more robust to noise and adhere to large scale structure in the data better than the mST and Steiner tree. **Left)** When increasing  $\alpha$ , the sensitivity to random density fluctuations in the data decreases (good). At the same time, the total length of the tree increases (bad). This tradeoff can be adjusted with a single hyperparameter  $\alpha$ . More details in Section 3.1. **Right)** CST and BCST of two samples, red and blue, drawn from the same distribution. The tree backbone reflects the global structure more accurately for  $\alpha > 0$  than for  $\alpha = 0$ . Edge widths are proportional to their centrality. All trees except for the mST were computed using the heuristic proposed in Section 6.2. See Appendix A for more examples.



We explore the robustness of the (B)CST problem against data perturbations by comparing the (B)CST topologies as small noise is introduced. In a toy example, three sets of 1000 points, uniformly sampled from a rectangle, are perturbed with Gaussian noise, yielding five perturbed sets per original set. We then compute the CST and BCST for  $\alpha \in \{0, 0.1, \dots, 0.9, 1\}$ , deeming a tree robust if minor data perturbations lead to minor geometrical changes in the tree. Formally, we consider a method  $\delta$ -robust, if, for any sets of points  $P_1$  and  $P_2$  and their respective trees  $T_1$  and  $T_2$ ,

$$d_T(T_1, T_2) \leq \delta d_P(P_1, P_2),$$

where  $d_T$  and  $d_P$  measure tree and set distances, respectively. The set distance  $d_P$  quantifies the perturbations we aim to withstand, where lower  $d_P$  values correspond to sets that are similar based on specific criteria. In our example, we define  $d_P$  as the average distance between points and their perturbed counterparts. Since we apply the same noise to each point, the average distance between points approximates the Gaussian noise’s standard deviation, making it nearly constant. To quantify structural tree changes, we set  $d_T$  equal to the Frobenius norm of the shortest path distance matrices between the original and perturbed (B)CST trees.

Figure 4a shows the average Frobenius norm between the original and corresponding perturbed samples across various  $\alpha$  values. It is evident that as  $\alpha$  increases, there is a noticeable decrease in the Frobenius norm. Since our  $d_P$  is fairly constant, showcasing that the Frobenius norm decreases implies a reduction in  $\delta$  as  $\alpha$  rises, i.e. the trees become more robust. However, we also plot the average cost of the trees (sum of the individual edge costs), which increases with  $\alpha$ . Thus, the improvement in robustness comes at the expense of adding longer edges. This pattern is expected because, as  $\alpha$  increases, the (B)CST tends to a medoid-centered star graph (see Section 2.1). This graph will have long edges but will also exhibit robustness to noise due to the medoid’s inherent stability. According to our definition of  $\delta$ -robustness, the  $\alpha \rightarrow \infty$  (B)CST limiting case, which always outputs a star-graph, will be deemed robust despite its undesirability for describing the data structure.

We associate the data structure with the graph node interconnectivity, wherein shorter edges preserve it better. Thus,  $\alpha$  serves as a parameter trading off stability vs. data fidelity. Indeed, the mST and Steiner tree ( $\alpha = 0$ ) on the right side of Figure 4 are highly sensitive to minor data changes due to their greedy nature, prioritizing shorter edges. Conversely, the (B)CST solutions at  $\alpha = 0.8$  are more stable, faithfully representing the data’s overall layout, albeit with longer edges.

## 3.2 Real-world data

The ability to summarize the structure of data without excessive sensitivity to random jitter is important in many applications. In this section we briefly showcase some potential applications where the (B)CST can be beneficial (see Appendix B for more details):

**Trajectory inference of single cell data:** The high dimensional single cell RNA-sequencing data can be used to model the trajectories defined by the cell differentiation process. We show results on mouse bone marrow data [29], here denoted as Paul dataset. To study robustness we perturb the data by removing half of the samples and then compare how the backbone of different spanning trees is affected by this perturbation. For visualization,

the data is embedded in 2D using PAGA [41]. If a spanning tree aligns well with the 2D embedding, this is an indication that the tree approximates the trajectory well. Figure 5 shows the embedded trees. The mST misses the highlighted bifurcation and it is more sensitive to the noise. The CST and BCST are robust to the perturbation and align well with the PAGA embedding, though the CST may not reconstruct well the finer details. The addition of SPs enables the BCST to follow the trajectory more closely. The case  $\alpha = 1$  results in a star graph, possibly due to the curse of dimensionality (see Remark 2.2).

**3D Plant skeletonization:** The skeletonization of plants is relevant for comprehending plant structure, with applications in subsequent tasks. The skeletons of a 3D point cloud of the surface of a plant [33], obtained using the BCST for various  $\alpha$  values, is shown in Figure 6. Intermediate  $\alpha$  values are able to represent the plant’s structure more accurately than extreme ones, producing qualitatively pleasing skeletons with fewer resources compared to dedicated handcrafted pipelines, which may depend on supplementary attributes like plant segmentation [25].

## 4 Correspondence Between the BCST and CST Topologies

Both the CST and the BCST problems have to be optimized over the set of feasible spanning tree topologies. This optimization is combinatorial in nature and turns out to be NP-hard in both cases. For the CST case, Cayley’s formula tells us that the number of feasible topologies is equal to  $N^{N-2}$  [8], which grows super-exponentially with the number of nodes  $N$ . For the BCST case, w.l.o.g., we can represent any feasible topology as a full tree topology, i.e. as a tree with  $N - 2$  Steiner points, each of degree 3, and with leaves corresponding to the  $N$  terminals. This representation is justified by the fact that any other feasible topology with SPs of degree higher than 3 can be represented by a full tree topology by collapsing two or more SPs, that is, when two or more SPs have the same coordinates. Figure 7 illustrates how a single full tree topology can realize different feasible topologies of the BCST problem. For  $N$  terminals, the number of possible full topologies is equal to  $(2N - 5)!! = (2N - 5) \cdot (2N - 7) \cdot \dots \cdot 3 \cdot 1$  [32], which also scales super-exponentially, but at a lower rate than the number of topologies of the CST. Consequently, an exhaustive search through all trees is not feasible.

The heuristic presented in Section 6 exploits the correspondence between the feasible topologies of the BCST and the CST problems. Given a full tree topology  $T_{\text{BCST}}$ , we say that a topology  $T_{\text{CST}}$  of the CST problem can be derived from  $T_{\text{BCST}}$  if: 1) we can collapse the SPs of  $T_{\text{BCST}}$  with the terminals such that the resulting topology is  $T_{\text{CST}}$ , and 2) for any SP  $s$  that is collapsed with terminal  $t$ , then all SPs along the path connecting  $s$  to  $t$  must also collapse with  $t$ . In other words, a SP cannot overtake any other SP in the collapse process. Figure 8a (from top to bottom) shows the steps to transform a topology  $T_{\text{BCST}}$  into a topology  $T_{\text{CST}}$  by iteratively collapsing the SPs. Analogously, we can derive a topology  $T_{\text{BCST}}$  from  $T_{\text{CST}}$  by spawning SPs from the terminals in  $T_{\text{CST}}$ , i.e., introducing SPs connected to the terminals. Since in a full tree topology SPs have degree 3 and terminals have degree 1, we add one SP per each pair of nodes adjacent to a common terminal node, so that the SP is connected to the triple of nodes.

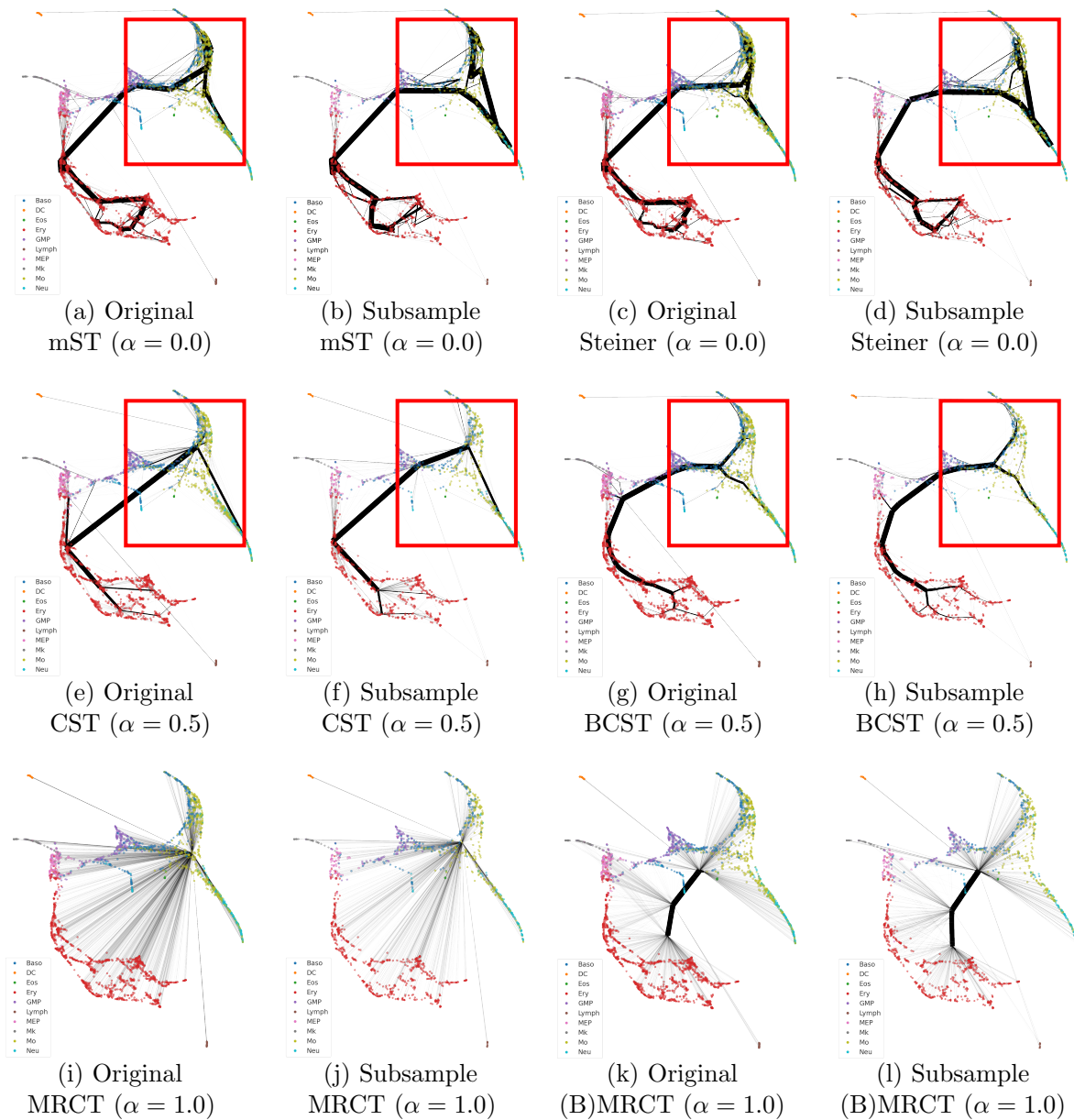


Figure 5: **mST**, **(B)CST** and **BMRCT** of the Paul Dataset. We applied the algorithms to both the original data (top) and a perturbed version with half of the points randomly removed (bottom). PAGA was used for 2D visualization, while the trees were computed in a 50-dimensional PCA projection. Colors represent different cell populations. The width of the edges is proportional to their centralities. 5a-5d) In the original data, the mST and Steiner tree do not faithfully model the trajectory bifurcation highlighted by the rectangle. Moreover, the trajectory changes drastically at this point once a subset of the samples is removed. 5e-5h) The CST and BCST at  $\alpha = 0.5$  are able to detect the bifurcation, and preserve the main backbone of the tree after the data has been perturbed. 5i-5l) The MRCT and its branched version are robust to perturbations due to its star shape structure, but this shape is also responsible for its incapability to model the data properly

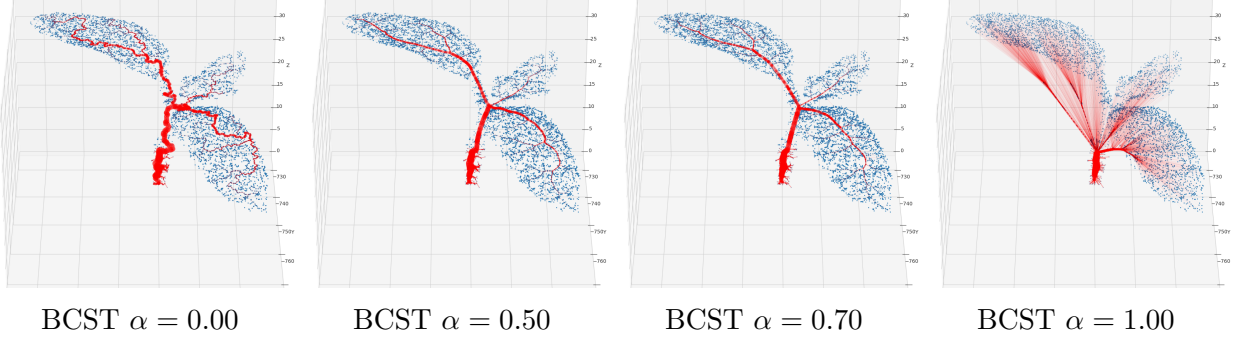


Figure 6: **BCST 3D plant skeletonization.** BCST for different  $\alpha$  values of a 3D point cloud with 5000 samples capturing the surface of a plant. With  $\alpha = 0.00$ , the tree branches exhibit greater irregularity, while at  $\alpha = 1.00$ , finer details are obscured. Intermediate  $\alpha$  values offer a more faithful representation of the plant’s structure.

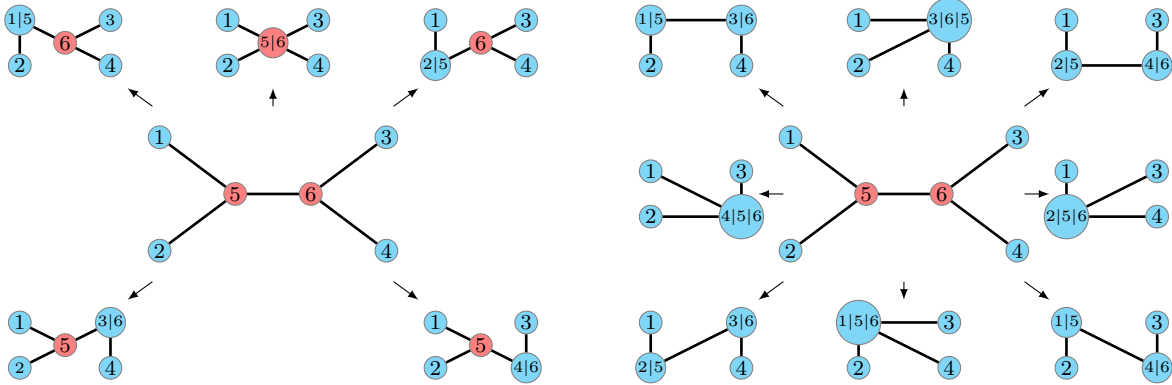
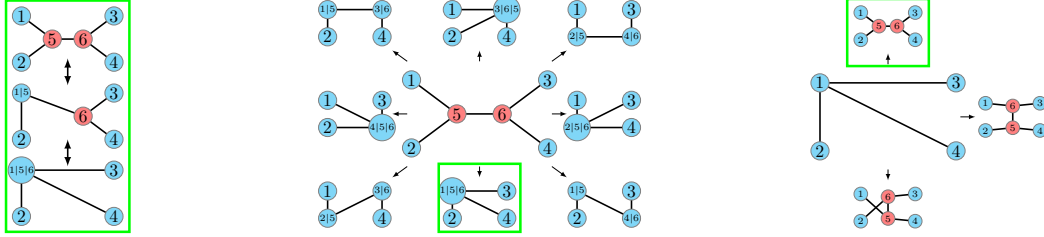


Figure 7: **Realizable BCST Solutions Topologies from a Full Tree Topology with 4 Terminals.** Terminal nodes and SPs are represented in blue and red, respectively. Different topologies emerge from a single full tree topology depending on how the SPs collapse with other nodes.

The correspondence mapping between  $T_{\text{CST}}$  and  $T_{\text{BCST}}$ , as shown in Figures 8b and 8c, is not unique. Multiple  $T_{\text{CST}}$  can be derived from a single  $T_{\text{BCST}}$  and vice versa. At most, any  $T_{\text{BCST}}$  can generate  $O(3^{N-2})$   $T_{\text{CST}}$  topologies because each SP locally has 3 nodes to merge with. In concrete, the number of  $T_{\text{CST}}$  topologies derivable from a singular full tree topology is given by the determinant of a submatrix of the Laplacian matrix, denoted as  $L_{SP_s, SP_s}$ . This submatrix is obtained by selecting the rows and columns indexed by the SPs. Hence, the number of derivable topologies is precisely  $\det L_{SP_s, SP_s}$ . This assertion is proven in Theorem F.2 (refer to Appendix F.2 for details). The proof leverages the bijective relationship between the derivable CST topologies and the terminal separating spanning forests of a full tree topology. By combining this bijectivity with the fact that the minors of the Laplacian matrix provide the count of forests separating the non-indexed rows and columns, the desired result is established.



(a) Steps  $T_{\text{CST}} \leftrightarrow T_{\text{BCST}}$  (b) Topologies derived from  $T_{\text{BCST}}$  (c) Topologies derived from  $T_{\text{CST}}$

Figure 8: **CST and BCST Topology Correspondence.** 8a) Mapping from a BCST topology,  $T_{\text{BCST}}$ , to a CST one,  $T_{\text{CST}}$  and vice versa. Terminal nodes and SPs are represented in blue and red, respectively. From top to bottom it is shown how the SPs couple to different terminals. From bottom to top, the SPs are spawned by different pairs of adjacent neighbors. First by the pair 3 and 4 and later by 2 and 6. 8b)  $T_{\text{CST}}$  feasible topologies derived from a single  $T_{\text{BCST}}$ . 8c)  $T_{\text{BCST}}$  feasible topologies derived from a single  $T_{\text{CST}}$ .

Similarly, the number of  $T_{\text{BCST}}$  topologies derived from  $T_{\text{CST}}$  is given by

$$\prod_{v : d_v \geq 2} (2d_v - 3)!!, \quad (7)$$

where  $d_v$  is the degree of terminal  $v$  in the  $T_{\text{CST}}$  topology. Higher-degree terminals can generate more topologies as they can spawn more pairs of nodes. For more details on the cardinalities of derivable topologies, see Appendix F.

Despite the mapping ambiguity between the topologies, we can reduce the number of trees to explore in the CST/BCST given a BCST/CST topology. Although the optimum of one problem is not guaranteed to be derived from the optimum of the other (see Figure 9), we show empirically that the heuristic proposed in Section 6 can exploit the positions of the SPs together with the correspondence between the sets of topologies of both problems to produce competitive results.

## 5 Geometry of Optimal BCST Topologies

In this section, we will analyze the geometry of the optimal topologies of a BCST problem. Concretely, we will determine an analytical formula of the angles that form the edges at a SP. Using this relation, we will prove that when the terminal points lie in a plane, then SPs with degree higher than 3 are not realized in optimal solutions for  $\alpha \in [0, 0.5] \cup \{1\}$  unless they collapse with a terminal. For  $\alpha \in ]0.5, 1[$  we provide empirical evidence that the statement also holds in that case.

### 5.1 Branching Angles at the Steiner Points

In this section, we formulate the branching angles in terms of the centralities of the edges for a given topology of the BCST problem. The derivation of the angles is based on previous works [4, 22], which apply analogous arguments for the Branched Optimal Transport (BOT)

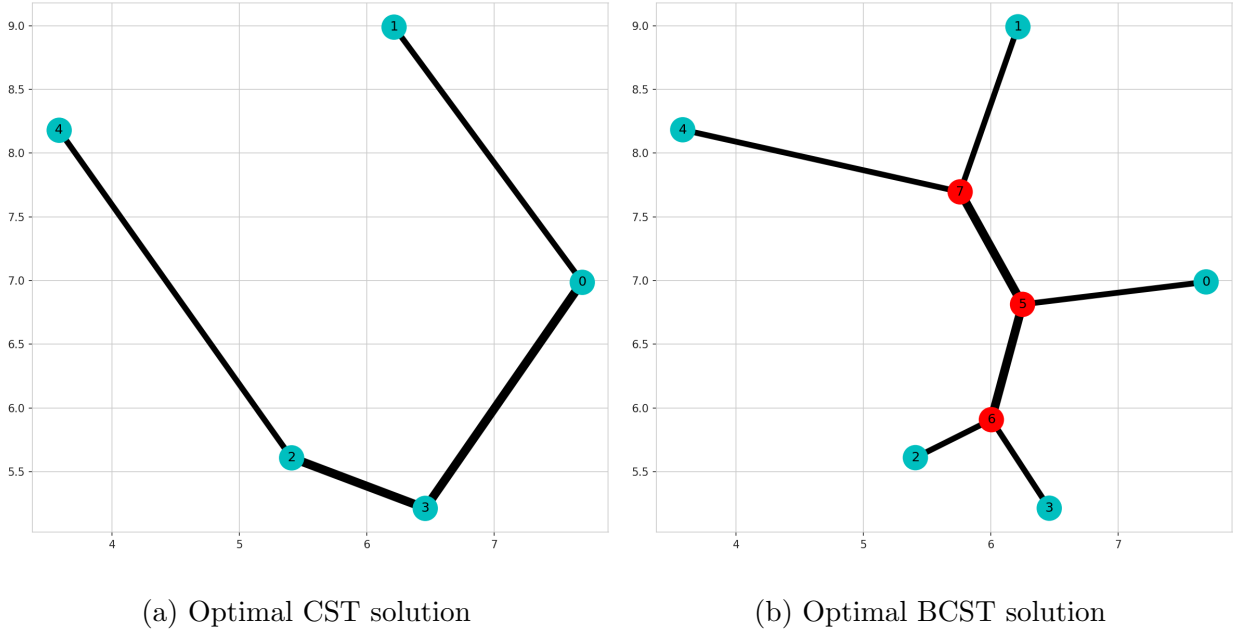


Figure 9: **Optimal CST and BCST Topologies May Not Be Derived from Each Other Given the Same Terminal Configuration.** Left: Optimal CST solution. Right: Optimal BCST solution. The CST topology cannot include nodes 4 and 1 as direct neighbors if derived from BCST, as it would result in nodes adjacent to a common neighbor. Similarly, the optimal BCST topology cannot be derived from the CST topology, as nodes 1 and 4 would not be connected to a common SP.

problem. The main difference lies in the weighting factors that multiply the distances in the objective function (2), which in our case are the edge betweenness centralities, and in BOT are flows matching supply to demand. Appendix C.1 elaborates on the similarities and differences between the BOT and BCST problems.

First and foremost, we emphasize the locality characteristic of the geometric optimization of SPs of the BCST problem. Because of the convexity of the BCST objective (2) given a fixed topology, it can be shown that the geometric optimization of the SPs coordinates can be solved locally, meaning that the optimal position of a SP is determined by its neighbors and weighting factors. Lemma 5.1 formalizes this statement. For a proof, we refer to Lemma 2.1 of [22], where the same statement was shown for the BOT problem. Since the proof is independent of the weighting factors multiplying the distances, the result applies to the BCST problem as well.

**Lemma 5.1.** Given a topology, its SPs are in optimal position w.r.t. the BCST problem if and only if any individual SP interconnects its neighbors at minimal cost. Moreover, the optimal topology of the BCST is optimal if and only if for any subset of connected nodes the corresponding subtopology solves the respective subproblem.

*Proof.* See Lemma 2.1 of [22]. □

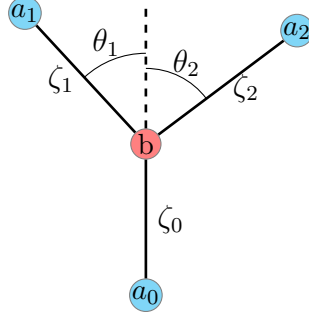


Figure 10: **Branching Angles at Steiner Point.** The symbols  $\zeta_i$  represent the normalized centralities of the edges, that is  $\zeta_i := m_{ba_i}(1 - m_{ba_i})$ .

Recall that any feasible topology of the BCST problem can be represented as a full tree topology where each SP has degree 3. Thus, as a consequence of Lemma 5.1, it is enough to study the geometric optimization of 3 nodes connected by a single SP. Consider the problem configuration depicted in Figure 10, where node  $b$  represents the branching point whose coordinates need to be optimized, nodes  $\{a_i\}_i$  are the terminals with fixed positions and  $\{\zeta_i := m_{ba_i}(1 - m_{ba_i})\}_i$  are the normalized centralities of the edges  $\{(b, a_i)\}_i$ . The objective to be minimized is

$$C(b) = \zeta_0 \|b - a_0\| + \zeta_1 \|b - a_1\| + \zeta_2 \|b - a_2\| \quad (8)$$

Bernot et al. showed that when the  $b$  does not collapse with any terminal, then the angles  $\theta_1$  and  $\theta_2$  are given by

$$\begin{aligned} \cos(\theta_1) &= \frac{\zeta_0^{2\alpha} + \zeta_1^{2\alpha} - \zeta_2^{2\alpha}}{2\zeta_0^\alpha \cdot \zeta_1^\alpha} \\ \cos(\theta_2) &= \frac{\zeta_0^{2\alpha} + \zeta_2^{2\alpha} - \zeta_1^{2\alpha}}{2\zeta_0^\alpha \cdot \zeta_2^\alpha} \\ \cos(\theta_1 + \theta_2) &= \frac{\zeta_0^{2\alpha} - \zeta_1^{2\alpha} - \zeta_2^{2\alpha}}{2\zeta_1^\alpha \cdot \zeta_2^\alpha} \end{aligned} \quad (9)$$

Alternatively, we can analyze when node  $b$  collapses with one of the terminals. Assuming without loss of generality that  $b$  collapses with  $a_0$ , let  $\gamma := \angle a_1 a_0 a_2$ . According to Lippmann et al.,  $b$  collapses with  $a_0$  if

$$\gamma \geq \arccos\left(\frac{\zeta_0^{2\alpha} - \zeta_1^{2\alpha} - \zeta_2^{2\alpha}}{2\zeta_1^\alpha \cdot \zeta_2^\alpha}\right) = \theta_1 + \theta_2. \quad (10)$$

Hence,  $b$  collapses to  $a_0$  if  $\angle a_1 a_0 a_2$  exceeds the optimal angle specified by (9). This scenario results in the so-called  $V$ -branching.

For a comprehensive derivation of the angles, please refer to Appendix G, where we present the arguments from the works of Bernot et al. [4] and Lippmann et al. [22].

## 5.2 Infeasibility of Degree-4 Steiner Points in the Plane

In this section, we will prove the infeasibility of degree-4 SPs in the optimal solution of the BCST. Specifically, we will focus on the scenario where the terminal nodes lie in the

plane and the value of  $\alpha$  falls within the range  $\alpha \in [0, 0.5] \cup \{1\}$ . Moreover, we will provide compelling evidence to support the validity of the statement for the case where  $\alpha \in ]0.5, 1[$ . We will divide the proof into two parts: one for  $\alpha \in [0, 0.5]$ , presented here, and the other for  $\alpha = 1$ , which is detailed in Appendix H.

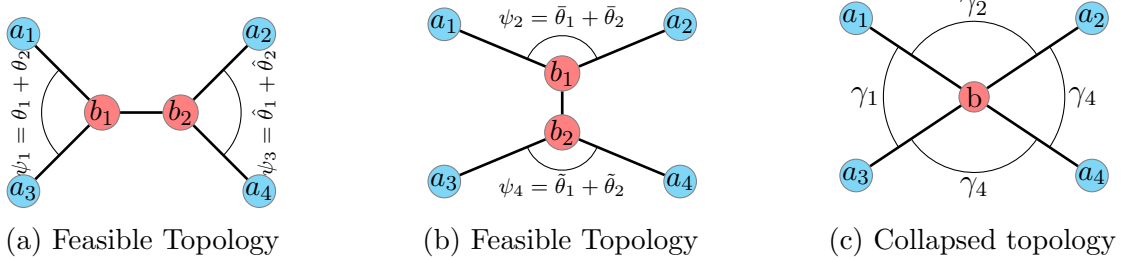


Figure 11: **Optimal Angles in Degree-4 SP Require  $\psi \leq \gamma_i$ .** Figures 11a) and 11b) depict two topologies, where the optimal angles given by equation (10) are represented by  $\psi_i$ . Figure 11c illustrates the collapsed solution with the corresponding angles  $\gamma_i$ . An essential requirement for the optimality of Figure 11c is that  $\psi_i \leq \gamma_i$ .

**Theorem 5.1.** Let  $\alpha \in [0, 0.5]$ . Given a set of terminals which lie in the plane, then the SPs of the optimal solution of the BCST problem will not contain SPs of degree-4 unless these collapse with a terminal.

*Proof.* The optimality of a solution in the BCST problem relies on the locality characteristic, as stated in Lemma 5.1. Specifically, each subtopology within a connected subset must solve its respective problem for the overall solution to be optimal. Consequently, the realization of a degree-4 SP, as depicted in Figure 11, requires the collapse of  $b_2$  with  $b_1$ . Moreover, this collapse must occur in any topology. As we have discussed in Section 5.1, both nodes  $b_1$  and  $b_2$  will collapse if a  $V$ -branching occurs, that is if the angle realized between the collapsed node and the two other nodes connected to it exceeds the optimal angle given by (10). Therefore, from Figure 11 it follows that  $\gamma_i \geq \psi$ . We will demonstrate that the sum of  $\sum_{i=1}^4 \gamma_i$  is greater than  $2\pi$ , rendering a SP of degree-4 infeasible. To do this we will prove that  $\psi_i > \pi/2$  for all  $i$ .

W.l.o.g. let us consider  $i = 1$  and denote  $\psi_1$  as  $\psi$ . If  $\cos(\psi) < 0$ , the angle  $\psi \in [0, \pi]$  will be greater than  $\pi/2$ . Based on (10), we can derive the following:

$$\cos(\psi) = \frac{F(m_{a_3b_1} + m_{a_1b_1})^{2\alpha} - F(m_{a_3b_1})^{2\alpha} - F(m_{a_1b_1})^{2\alpha}}{2F(m_{a_3b_1})^\alpha F(m_{a_1b_1})^\alpha} \quad (11)$$

The function  $F(x)^{2\alpha} = (x(1-x))^{2\alpha}$  is strictly subadditive in  $\mathbb{R}^+$  for  $\alpha \in [0, 0.5]$ ,<sup>4</sup> that is  $F(x+y)^{2\alpha} < F(x)^{2\alpha} + F(y)^{2\alpha}$  if  $x, y > 0$ . Thus, the numerator of (11) is negative and therefore  $\cos(\psi) < 0$ . Consequently,  $\psi > \pi/2$ . This argument applies to all  $\psi_i$ , hence their sum will be greater than  $2\pi$ . Consequently, a SP with degree-4 cannot be part of an optimal solution.  $\square$

<sup>4</sup>In fact, all concave functions,  $f$ , with  $f(0) \geq 0$  are subadditive on the positive domain. It is easy to see that  $(x(1-x))^{2\alpha}$  is concave for  $\alpha \in [0, 0.5]$ .



In [22], Lippmann et al. applied the same argument to establish the infeasibility of degree-4 SPs in the context of the BOT problem for  $\alpha \in [0, 0.5]$ . Our proof shows explicitly that this argument generalizes to other minimization problems of the same nature, where the edge-lengths are multiplied by weighting factors  $F(m_{ij})$ , with  $F(\cdot)$  being a positive function dependent on  $m_{ij}$  and exhibiting subadditivity when squared. For higher  $\alpha > 0.5$  the argument used in the previous theorem does not apply. Indeed, we can find values for the edge centralities for which the lower bounds of  $\gamma_i$ , given by (10), are all lower than  $\pi/2$ .<sup>5</sup> Now, we will take a more general approach that can rule out degree-4 branching for higher  $\alpha$  values. In concrete, we will show the infeasibility for  $\alpha = 1$ .

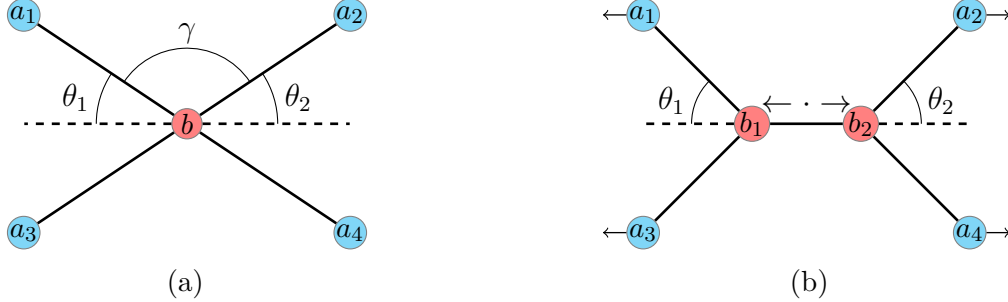


Figure 12: **Splitting Collapsed SP while Preserving Optimal Angles.** 12a) illustrates the collapsed solution of a 4-terminal configuration. (12b) demonstrates that it is possible to move jointly the terminal points  $\{a_1, a_3\}$  in a specific but opposite direction to the one of the terminals  $\{a_2, a_4\}$ , resulting in the splitting of the collapsed SP  $b$  into two distinct SPs,  $b_1$  and  $b_2$ . Remarkably, this split can be executed while preserving the angles  $\theta_1$  and  $\theta_2$ . Importantly, these angles must correspond to the optimal angles given by (9).

The optimal position of the SPs is continuously dependent on the terminal positions and solely relies on the branching angles, as shown in Section 5.1. Consequently, assuming that there exists a configuration such that the SPs collapse, it is possible to find terminal positions that lead to an unstable collapse of the SPs. Here, instability refers to a configuration where an infinitesimal translation of the terminals results in the splitting of the SPs. This scenario is depicted in Figure 12. In such cases, the angles realized by the terminals and the SPs will reach the upper bounds specified by (10). Therefore, the angles depicted in Figure 12a fulfill the condition

$$\gamma = \pi - \theta_1 - \theta_2, \quad (12)$$

where the angles satisfy

$$\cos(\gamma) = \frac{F(m_{a_1b} + m_{a_2b})^{2\alpha} - F(m_{a_1b})^{2\alpha} - F(m_{a_2b})^{2\alpha}}{2F(m_{a_1b})^\alpha F(m_{a_2b})^\alpha}, \quad (13)$$

$$\cos(\theta_1) = \frac{F(m_{a_3b} + m_{a_1b})^{2\alpha} + F(m_{a_1b})^{2\alpha} - F(m_{a_3b})^{2\alpha}}{2F(m_{a_3b} + m_{a_1b})^\alpha F(m_{a_1b})^\alpha}, \quad (14)$$

$$\cos(\theta_2) = \frac{F(m_{a_2b} + m_{a_4b})^{2\alpha} + F(m_{a_2b})^{2\alpha} - F(m_{a_4b})^{2\alpha}}{2F(m_{a_2b} + m_{a_4b})^\alpha F(m_{a_2b})^\alpha}, \quad (15)$$

<sup>5</sup>For instance for  $\alpha = 1$ ,  $m_{a_3b_1} = m_{a_1b_1} = 0.2$  and  $m_{a_2b_2} = m_{a_4b_2} = 0.3$  all  $\psi_i$  angles are acute. Hence their sum is also lower than  $2\pi$ .

with  $F(x) = x(1-x)$ . By processing further equation 12, we arrive at the following expression (see Appendix H for further details)

$$(\cos(\gamma) + \cos(\theta_1) \cos(\theta_2))^2 - (1 - \cos(\theta_1))^2 (1 - \cos(\theta_2))^2 = 0. \quad (16)$$

Solving (16) analitically for all  $\alpha$  is difficult. Nonetheless, in Appendix H we show analitically that for  $\alpha = 1$  (12) can not hold given the constraints on the terms  $m_{a_i,b}$ , namely  $\sum_{i=1}^4 m_{a_i,b} = 1$  and  $0 < m_{a_i,b} < 1$  for all  $i$ .

**Theorem 5.2.** Let  $\alpha = 1$ . Given a set of terminals which lie in the plane, then the SPs of the optimal solution of the BCST problem will not contain SPs of degree-4 unless these collapse with a terminal.

*Proof.* See Appendix H □

Though we have not been able to prove analytically the infeasibility of degree-4 SPs for  $\alpha \in ]0.5, 1[$ , we strongly believe that the statement still holds. Figure 13 shows the surface plots of the numerator of the equality 16 (once expanded) w.r.t.  $m_1$  and  $m_2$  for different fixed values of  $\alpha$  and  $m_3$ . Upon analysis, it appears that the numerator exhibits an increasing trend with respect to  $\alpha$  within the interval  $[0.5, 1]$ . This observation leads us to hypothesize that if the equality holds for  $\alpha = 0.5$  and  $\alpha = 1$ , it is likely to hold for intermediate values as well. However, due to the complexity of the formula, it is challenging to verify this hypothesis analytically.

In section 2.1, we have demonstrated that as  $\alpha > 1$  and  $N \rightarrow \infty$ , the BCST tends to converge to a star graph centered at the geometric median. Consequently, for  $\alpha > 1$ , a degree-4 SP becomes feasible.

**Remark 5.1.** In [22], it was shown for the BOT problem that if degree-4 SPs are not feasible then higher degree SPs are not possible either. The same reasoning applies for the BCST, since the proof does not depend on the weighting factors. Thus for  $\alpha \in [0, 0.5] \cup \{1\}$ , only degree-3 SPs are feasible unless they do collapse with a terminal node. Due to the compelling evidence shown, we also believe this is the also the case for  $\alpha \in ]0.5, 1[$ . Lippmann et al. [22] also showed that some of the results of the BOT problem obtained on the plane can be extended to other 2-dimensional manifolds. Again, this is also the case for the BCST problem. Among these properties, we emphasize the optimal angles formulae exposed in section 5.1 and the infeasibility of degree-4 SPs for appropriate  $\alpha$  values. We refer to Appendix F of [22] for more details.

## 6 CST and BCST Optimization Algorithm

This section details the proposed heuristic for optimizing the BCST and CST problems. We will first focus on the BCST. The heuristic iterates over two steps: First, given a fixed topology, the algorithm finds the geometric positions of the Steiner points (SPs) that exactly minimize the cost conditioned on the topology. Given the optimal coordinates of the SPs, we then update the topology of the tree by computing an mST over the terminals and SPs. This procedure is iterated until convergence or until some stopping criterion is met.



Figure 13: SPs of Degree-4 in the Plane are Likely not Feasible for  $\alpha \in ]0.5, 1[$ . Surface plots are depicted, illustrating the left side of equation (16), as a function of  $m_1$  and  $m_2$ , with different fixed values of  $\alpha$  and  $m_3$ . From left to right:  $m_3$  is fixed and  $\alpha$  ranges over  $\{0.5, 0.6, 0.7, 0.8, 0.9, 1.0\}$ . From top to bottom:  $\alpha$  is fixed and  $m_3$  ranges over  $\{0.1, 0.2, \dots, 0.8, 0.9\}$ . For  $m_3$  fixed,  $m_1$  and  $m_2$  range over the domain defined by  $\{(x, y) : 0 < x + y + m_3 < 1\}$ . In all plots, the function values are negative and tend towards 0 as  $m_1$ ,  $m_2$ , or  $m_3$  approaches 0. We can observe that for fixed  $m_1$ ,  $m_2$  and  $m_3$ , the function seems to be increasing with respect to  $\alpha$  (from right to left). Since we have previously demonstrated that the left side of equation (16) does not equal zero in the desired domain for  $\alpha = 0.5$  and  $\alpha = 1$ , the plots suggest that this is also the case for  $\alpha \in ]0.5, 1[$ .

## 6.1 Geometry Optimization

The BCST problem can be divided into two subproblems: combinatorial optimization of the tree topology and geometric optimization of the coordinates of the SPs,  $X_B$ . When conditioning on a topology  $T$ , the BCST objective (2) is a convex problem w.r.t.  $X_B$ . Despite its convexity, the objective is not everywhere differentiable. We build on the iteratively reweighted least squares (IRLS) approach from Smith [36] and Lippmann et al. [22] to efficiently find the positions of the SPs.

Starting from arbitrary SPs coordinates, denoted as  $X^{(0)} = \{x_i^{(0)}\}_{i=1}^{2N-2}$ , the algorithm iteratively solves the following linear system of equations.

$$x_i^{(k+1)} = \frac{\sum_{j:(i,j) \in E} \zeta_{ij}^\alpha \frac{x_j^{(k+1)}}{\|x_i^{(k)} - x_j^{(k)}\|}}{\sum_{j:(i,j) \in E} \frac{\zeta_{ij}^\alpha}{\|x_i^{(k)} - x_j^{(k)}\|}}, \quad \forall N+1 \leq i \leq 2N-2. \quad (17)$$

where  $\zeta_{ij} = m_{ij}(1 - m_{ij})$ . We assume, without loss of generality, that the coordinates corresponding to the SPs are indexed from  $N+1$  to  $2N-2$ , where  $N$  is the number of terminals. The coordinates for the terminals, which remain fixed throughout all iterations, are represented by the other indices. Thanks to the tree structure of the graph, the linear systems can be efficiently solved in linear time.

In Appendix I, we show that the algorithm is agnostic to the weighting factors that multiply the distances, and can therefore be applied to compute any weighted geometric mean.

## 6.2 Heuristic Optimizer for the (B)CST Problem

We now present a heuristic which alternates between the SP geometric coordinate optimization (convex) and a topology update (combinatorial). The heuristic's main characteristic is how it exploits the location of the Steiner points given an initial topology guess. The heuristic's underlying assumption is that the optimum position of the Steiner points may suggest a more desirable topology.

Unless otherwise stated, the heuristic we propose starts from the mST over all terminal nodes. At this point, the mST does not contain any SPs and is therefore not a full tree topology. Thus, we need to transform the mST into a full tree topology. As mentioned in Section 4, and highlighted in Figure 8, this process is not unambiguous. In particular, for each terminal node  $v$  with degree  $d_v \geq 2$ , we have to add  $d_v - 1$  SPs. Consequently, there are  $(2d_v - 3)!!$  ways to connect these SPs to the neighbors of  $v$ . Among all possible subtopologies connecting the SPs with  $v$  and its neighbors, we choose the one given by the dendrogram defined by the hierarchical single linkage clustering algorithm applied to  $v$  and its neighbors. In practice, this choice tends to work relatively well since nearby terminals are also closer in the subtopology.

Once we have a full tree topology, we can apply the geometry optimization step to obtain the optimal coordinates of the SPs. We assume that the optimal positions of the SPs indicate which connections between nodes might be more desirable, since they may be biased to move

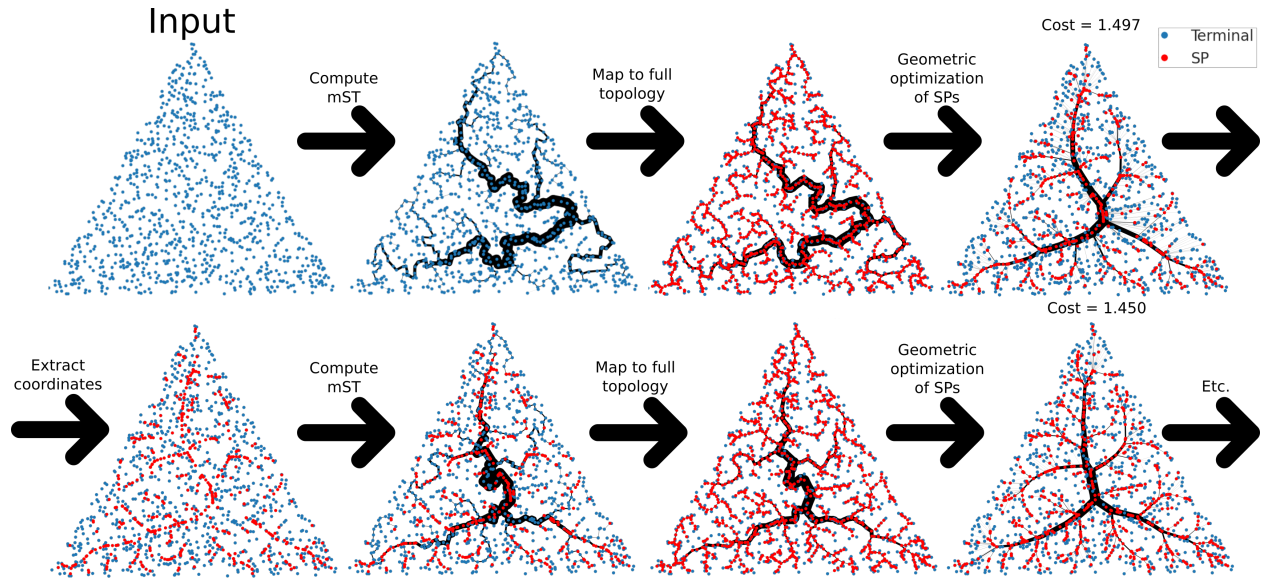


Figure 14: **mSTreg Heuristic**. The mSTreg heuristic iteratively transforms an mST to an approximate BCST. Given a set of points, the heuristic first computes the mST over all points and transforms it into a full tree topology by adding Steiner points (SPs). Next, the optimal positions of the SPs are computed using iteratively reweighted least squares. Given the updated SPs coordinates, the heuristic recomputes the mST over the union of the terminals nodes and the previous SPs. This mST produces a new topology, which is again transformed into a full tree topology by adding SPs whose coordinates are optimized. This process is repeated until some stopping criterion is satisfied.

closer to other nodes than the ones to which they are connected. Therefore, we propose to recompute a mST over the terminals together with the SPs. This new mST defines a new topology that needs to be transformed into a full tree topology for the geometry optimization. Once we have a valid full tree topology, we recompute the optimal positions of the SPs. This process is repeated iteratively until convergence or until some stopping criterion is met. We refer to this algorithm as the mST regularization (mSTreg) heuristic. The algorithm’s steps are illustrated in Figure 14, and its pseudocode is provided in Algorithm 1. The algorithm’s complexity is  $\mathcal{O}(dn \log(n)^2)$ . A detailed complexity analysis is available in Appendix J. We remark that the mSTreg heuristic is independent of the weighting factors that multiply the distances, thus it can also be used to approximate other problems as well, for instance a generalized version of the optimum communication tree with SPs.

Optionally, before the mST step is computed over the terminals and previous SPs, we can add intermediate points along the edges of the output generated by the geometry optimization step. These additional points will allow the mST to more reliably follow the edges of the geometry-optimized tree from previous step. Moreover, in case the initial topology was poor, these extra points may help to detect and correct edge crossings, which are known to be suboptimal. An illustration of the effect of these extra points can be found in Appendix K.

The heuristic designed for the BCST problem can also be applied to the CST problem by transforming BCST topologies at each iteration into CST topologies. While this transformation isn’t unique, we found that iteratively collapsing one SP at a time with the neighbor that

---

**Algorithm 1:** mSTreg Heuristic

---

**Input:**  $X$ , num\_iterations, sampling\_frequency, optimize\_CST**Output:** Tree

```
1  $mST_{init}$  =minimum_spanning_tree( $X$ ) // Define initial topology as mST
2  $T_{BCST}$  =transform2fulltopo( $mST_{init}$ ) // transform topology to full tree
   topology
3  $SP$  =compute_SP( $T_{init}$ ) // compute optimal  $SP$  coordinates
4 bestcost_BCST = $\infty$  if optimize_CST then
5    $\lfloor$  bestcost_CST = $\infty$ 

6 while  $it <$  num_iterations do
7   if sampling_frequency  $>$  2 then
8     /* sample extra points from edges */
9      $Y$  =sample_from_edge( $T_{BCST}, X \cup SP$ , sampling_frequency)
10     $SP = SP \cup Y$ 
11   $mST_{X \cup SP}$  =minimum_spanning_tree( $X \cup SP$ )
12   $T_{BCST}$  =transform2fulltopo( $mST_{X \cup SP}$ ) // transform  $mST_{X \cup SP}$  to full
   tree topology
13   $SP$  =compute_SP( $T_{reg}$ ) // compute optimal  $SP$  coordinates
14  if cost( $T_{BCST}$ )  $<$  bestcost_BCST then
15    bestcost_BCST =cost( $T_{BCST}$ )
16     $T_{BCST_{best}} = T_{BCST}$ 
17  if optimize_CST then
18     $T_{CST}$  =remove_SP( $T_{BCST}$ ) // Derive CST topology from BCST
   topology
19    if cost( $T_{CST}$ )  $<$  bestcost_CST then
20      bestcost_CST =cost( $T_{CST}$ )
21       $T_{CST_{best}} = T_{CST}$ 
```

---

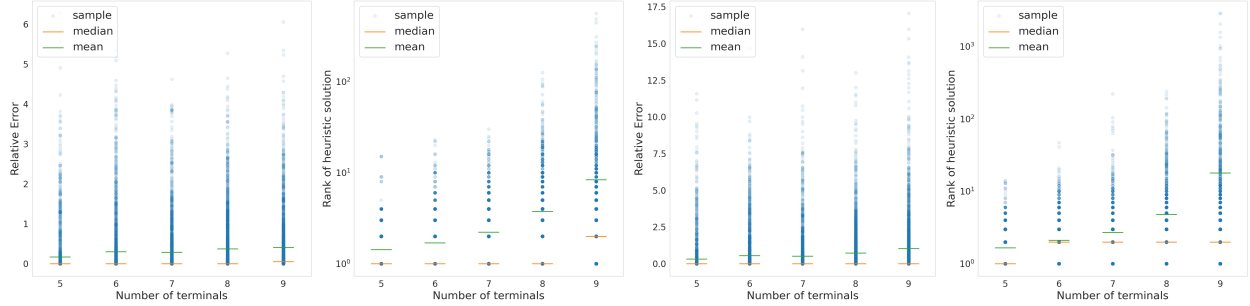
leads the smallest increase in cost produces compelling results. Additionally, when collapsing SPs together, centering the new node at the weighted geometric median of its new neighbors improves results slightly. Further details can be found in Appendix L.

## 7 Benchmark

### 7.1 Brute Force Benchmark

To assess the quality of the mSTreg heuristic, we compare the cost of the trees computed by the mSTreg algorithm with the globally optimal solutions obtained by brute force of configurations with up to nine terminals. We generate 200 instances with  $N \in \{5, 6, 7, 8, 9\}$





(a) BCST relative error (b) BCST rank of heuristic (c) CST relative error (d) CST rank of heuristic

**Figure 15: Bruteforce mSTreg Benchmark.** Relative cost errors between the mSTreg heuristic and optimal solutions; and sorted position of the heuristic tree for different number of terminals,  $N$ . For each  $N$ , we uniformly sampled 200 different terminal configuration and solved them for different  $\alpha$  values. Most runs ended up close to the global optimum, though the heuristic is slightly better for the BCST problem.

terminals sampled from a unit square. Both CST and BCST problems are solved for each  $\alpha \in \{0, 0.1, \dots, 0.9, 1\}$ . In Figure 15, the relative error, calculated as  $100(c_h - c_o)/c_o$  where  $c_h$  and  $c_o$  are heuristic and optimal costs, is shown for different  $N$ . We also show how the heuristic solution ranks, once the costs of all topologies are sorted. The heuristic attains the optimum in the majority of cases, with slightly better performance in the BCST problem than in the CST one. Appendix M provides  $\alpha$ -based results.

## 7.2 Steiner and MRCT Benchmark

In addition, we evaluate mSTreg on bigger datasets from the OR library<sup>6</sup> [3] for the Steiner tree ( $\alpha=0$ ) and the MRCT ( $\alpha=1$ ) problems. This dataset includes exact solutions of Steiner problem instances of up to 100 nodes randomly distributed in a unit square. The used instances are labeled as  $en.k$ , where  $n$  denotes the number of terminals, and  $k$  represents the instance number. Figure 16a compares the cost of our heuristic with the optimal cost. We also provide for reference the costs of the mST and the topology obtained by transforming the mST into a full tree topology with its SP coordinates optimized (referred to as mST\_fulltopo). Though our heuristic does not reach the optimal cost, it produces good topologies. The average relative error is lower than 1%.

For the MRCT, we compare our heuristic with the Campos [7] and GRASP\_PR [31] algorithms. Campos modifies Prim’s algorithm with heuristic rules, while GRASP\_PR conducts local search by exchanging one edge at a time. We test the algorithms on the OR library datasets for problem instances with 50, 100 and 250 terminals. Figure 16b shows the relative errors. In this case, we do not have access to the optimal cost, therefore we use GRASP\_PR costs cited from Sattari and Didehvar [31] as reference. Campos costs are obtained from our own implementation. For reference, we also show the 2-approximation [42] given by the star graph centered at the data centroid. While mSTreg proves competitive (surpassing Campos but falling short of GRASP\_PR by a modest average relative error

<sup>6</sup><http://people.brunel.ac.uk/~mastjjb/jeb/orlib/esteinfo.html>

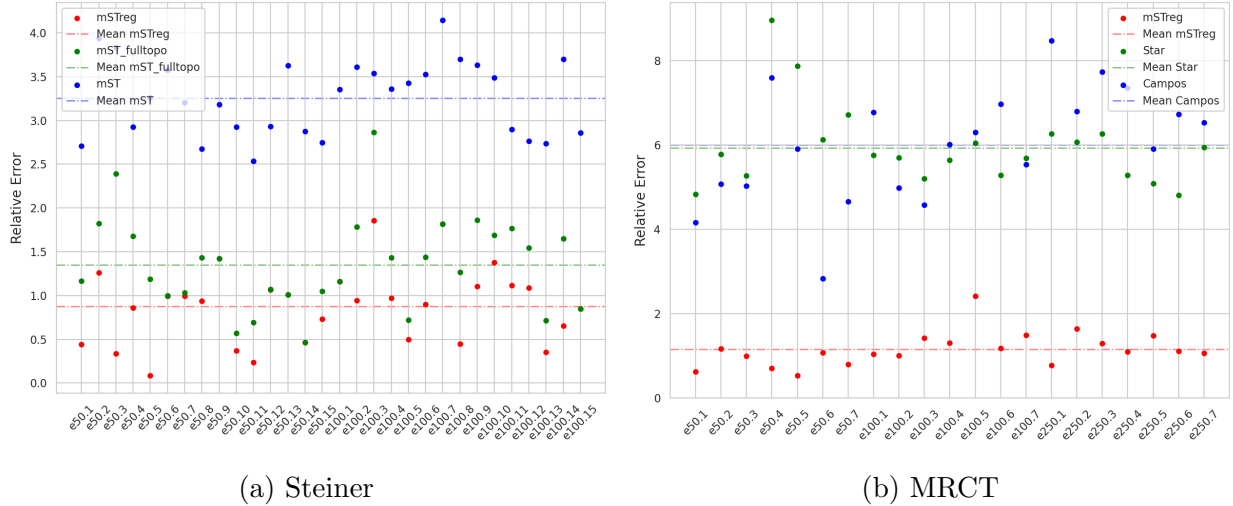


Figure 16: **Steiner and MRCT OR Library Dataset Benchmark.** Relative cost error with respect to a reference cost for the Steiner and MRCT problems for different instances and different methods (lower is better). Left: For the Steiner problem, the reference cost is the optimal cost. mSTreg finds good solutions and improves over mST\_fulltopo. Right: For MRCT the reference cost is given by the GRASP\_PR algorithm. The heuristic beats all other methods, but is still slightly worse than GRASP\_PR algorithm.

of 1.16%), it is worth noting that GRASP\_PR relies on a time-consuming local search. Leveraging the competitive solution provided by mSTreg as an initial step can enhance the performance and convergence of local search based algorithms, such as GRASP\_PR.

### 7.3 Comparing GRASP\_PR and mSTreg for the CST Problem

In the previous section, the GRASP\_PR algorithm by [31] outperformed the mSTreg heuristic when solving the MRCT problem. However, it’s important to note that the complexity of each iteration of the GRASP\_PR scales quadratically with the number of nodes for complete graphs. This is due to its local search algorithm, which involves swapping edges. In contrast, the complexity per iteration of the mSTreg algorithm is  $\mathcal{O}(dn \log(n)^2)$  (refer to Appendix J), making it more efficient.

Additionally, GRASP\_PR requires an initial random guess to initiate the local search. The quality of this guess impacts the number of iterations needed for the local search to converge. We will show that initializing the local search with a solution generated by the mSTreg allows for initializations that enhance the performance of GRASP\_PR. To ensure variability in the mSTreg output, we initialize it with random topologies instead of the mST. Despite this modification, as the mST initialization yielded favorable results, we opt to sample trees similar to it. To achieve this, we construct a tree by randomly sampling edges, prioritizing shorter ones. In concrete, we sample edge  $(i, j)$  with probability proportional to  $\exp(-\mu \|x_i - x_j\|)$  for some inverse temperature  $\mu$ . This sampling approach is akin to the one used in Karger’s algorithm for approximating the minimum cut [18, 17] and differs from the GRASP\_PR construction phase in that it doesn’t require that one end of the edges belongs



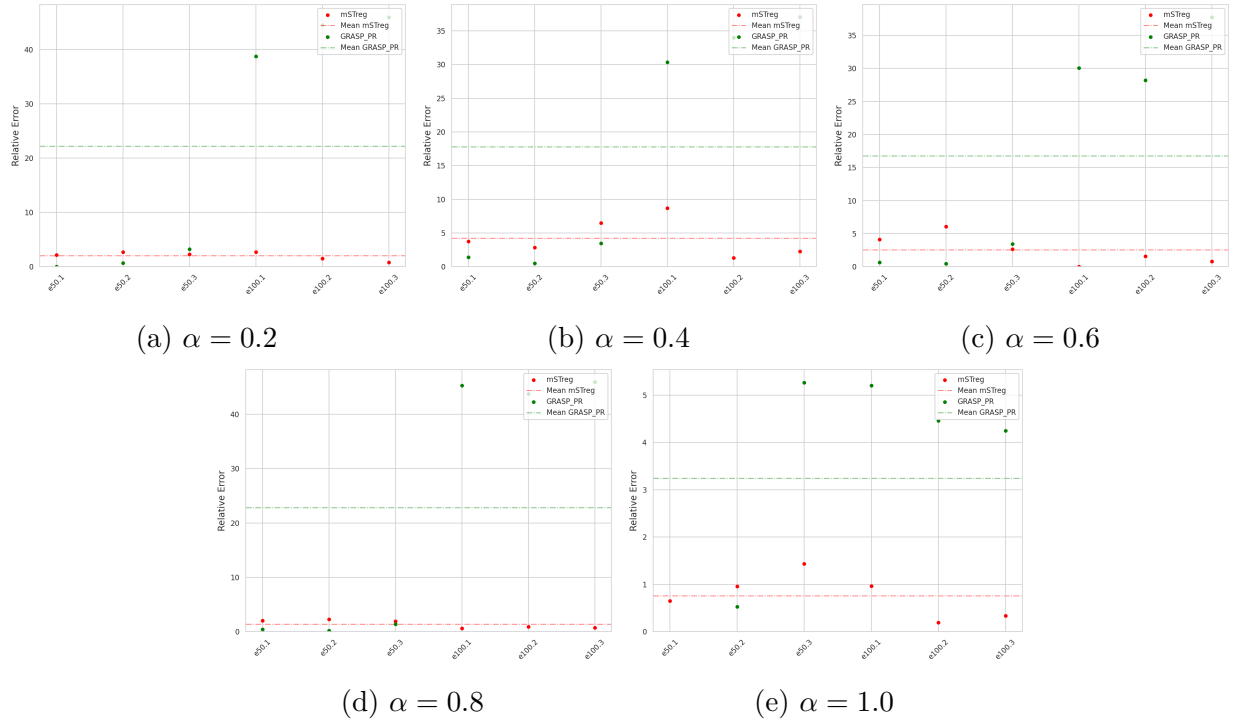


Figure 17: **Comparing mSTreg and GRASP\_PR.** Relative cost error concerning the CST problem at different  $\alpha$  values in the set  $\{0.2, 0.4, 0.6, 0.8, 1\}$  for various instances of the OR library dataset (lower is better). The comparison involves mSTreg, GRASP\_PR, and the combination of GRASP\_PR initialized with the mSTreg solution (referred to as GRASP\_PR\_mSTreg). The relative error is computed using the cost from GRASP\_PR\_mSTreg as a reference. The combination of the mSTreg heuristic with GRASP\_PR consistently achieved the lowest cost, with all relative costs above 0, demonstrating the enhancement of GRASP\_PR performance by mSTreg. To manage time constraints, a threshold of 5 minutes was imposed for methods utilizing GRASP\_PR.

to the current tree.

While the previous section relied on the MRCT costs reported in [31] for the GRASP\_PR algorithm, we now validate our claims using our implementation of GRASP\_PR. Given that the GRASP\_PR is a versatile algorithm, we used it to compute the CST with alternative  $\alpha$  values besides 1. Acknowledging that for lower  $\alpha$  values, the optimum trees will be more similar to the mST, we adapted the construction phase of GRASP\_PR. Specifically, for  $\alpha < 0.7$ , it generates the random tree based on the Karger’s sampling method mentioned above. For  $\alpha \geq 0.7$ , it uses the construction proposed by [31]. Due to the relatively slow performance of our Python implementation of GRASP\_PR, we imposed a 5-minute time threshold. If exceeded, the algorithm returns the best solution at the end of the path relinking (PR) phase of the GRASP\_PR algorithm.

To assess performance, we conducted tests using GRASP\_PR, GRASP\_PR initialized with mSTreg (GRASP\_PR\_mSTreg), and the mSTreg algorithms on OR library datasets for problems with 50 and 100 terminals. Analogously to the plots shown in the previous section, Figure17 displays the relative errors using GRASP\_PR\_mSTreg cost as a reference

for different  $\alpha$  values in the set  $\{0.2, 0.4, 0.6, 0.8, 1\}$ . Notably, the combination of the mSTreg heuristic with GRASP\_PR consistently achieved the lowest cost, as all relative costs are above 0, proving that the mSTreg can enhance GRASP\_PR performance.

Across 30 runs (combining 5  $\alpha$  values, 2 problem sizes, and 3 instances), GRASP\_PR outperformed mSTreg in only 12 instances. Moreover, achieving a lower cost took minutes for GRASP\_PR, while mSTreg run in the order of seconds. Consequently, mSTreg emerges as a favorable alternative, delivering a descent solution quickly.

## 8 Conclusion

We introduced the novel problem of the (branched) central spanning tree, which encompasses the minimum spanning, the Steiner and minimum routing cost trees as particular cases. The CST weighs the edge-costs with the edge-centralities, whose influence are regulated by the parameter  $\alpha$ . We have focused on the Euclidean version of the problem, where the nodes are embedded in an Euclidean space. Moreover, we presented a variant of the problem allowing the addition of extra nodes, referred to as Steiner points (SPs). In addition, we provided empirical evidence for the robustness of the (B)CST tree structure to perturbations in the data, which increases with  $\alpha$ . In this regard,  $\alpha$  serves as a parameter that trades-off between data-fidelity and stability. We also provided examples of potential applications such as 3D plant skeletonization or single cell trajectory inference.

On the theoretical side, we showed that as  $\alpha \rightarrow \infty$  or the number of terminals approaches infinity when  $\alpha > 1$ , (B)CST converges to a star-tree, indicating inadequacy in extracting structural information when  $\alpha > 1$ . Conversely, as  $\alpha \rightarrow -\infty$ , the (B)CST tends towards a path graph. Additionally, thanks to the closed formulae of the branching angles, we proved the infeasibility of degree-4 SPs when the terminals lie on a plane and  $\alpha \in [0, 0.5] \cup \{1\}$ . We also provided evidence that suggests a similar case for  $\alpha \in ]0.5, 1[$ .

Based on an efficient algorithm to compute the optimal locations of the SPs, we have proposed the mSTreg heuristic, which exploits the optimal position of the SPs and the correspondence between the CST and BCST topologies to find approximate solutions for either. We benchmarked this algorithm and showed its competitiveness on small toy data sets. Since the proposed heuristic is agnostic to the weighting factors that multiply the distances, we leave as future work to test whether it is equally competitive for other problems, like the general optimum communication tree. Another open question is whether the algorithm can be adapted to perform well on non-complete or non-Euclidean graphs.

## Acknowledgements

This work is supported by the Deutsche Forschungsgemeinschaft (DFG, German Research Foundation) under Germany’s Excellence Strategy EXC 2181/1 - 390900948 (the Heidelberg STRUCTURES Excellence Cluster).

## References

- [1] S. Ahn, B. Chen, T. Wang, and L. Song. Spanning tree-based graph generation for molecules. *International Conference on Learning Representations*. 2022 .
- [2] X. Bai, L. Ye, Z. Liu, and B. Liu. Promask: Probability mask representation for skeleton detection. *Neural Networks*, 2023. 162: 11–20.
- [3] J. E. Beasley. A heuristic for Euclidean and rectilinear Steiner problems. *European Journal of Operational Research*, 1992. 58: 284–292. Publisher: Elsevier.
- [4] M. Bernot, V. Caselles, and J.-M. Morel. *Optimal transportation networks: models and theory*. Springer, 2008.
- [5] S. Bezrukov, F. Kaderali, and W. Poguntke. On central spanning trees of a graph. *Combinatorics and Computer Science: 8th Franco-Japanese and 4th Franco-Chinese Conference Brest, France, July 3–5, 1995 Selected Papers*. Springer, 1996 53–57.
- [6] U. Brandes. On variants of shortest-path betweenness centrality and their generic computation. *Social Networks*, 2008. 30: 136–145.
- [7] R. Campos and M. Ricardo. A fast algorithm for computing minimum routing cost spanning trees. *Computer Networks*, 2008. 52: 3229–3247.
- [8] A. Cayley. A theorem on trees. *Quart. J. Math.*, 1878. 23: 376–378.
- [9] S. Chaiken. A combinatorial proof of the all minors matrix tree theorem. *SIAM Journal on Algebraic Discrete Methods*, 1982. 3: 319–329.
- [10] L. Chizat, S. Zhang, M. Heitz, and G. Schiebinger. Trajectory inference via mean-field langevin in path space. S. Koyejo, S. Mohamed, A. Agarwal, D. Belgrave, K. Cho, and A. Oh, eds., *Advances in Neural Information Processing Systems*, vol. 35. Curran Associates, Inc., 2022 16731–16742.
- [11] E. N. Gilbert and H. O. Pollak. STEINER MINIMAL TREES. *SIAM Journal on Computing*, 1968. 16.
- [12] G. M. Guisewite and P. M. Pardalos. Minimum concave-cost network flow problems: Applications, complexity, and algorithms. *Annals of Operations Research*, 1990. 25: 75–99.
- [13] G. M. Guisewite and P. M. Pardalos. Algorithms for the single-source uncapacitated minimum concave-cost network flow problem. *Journal of Global Optimization*, 1991. 1: 245–265.
- [14] T. C. Hu. Optimum Communication Spanning Trees. *SIAM Journal on Computing*, 1974. 3.
- [15] F. K. Hwang and D. S. Richards. Steiner tree problems. *Networks*, 1992. 22: 55–89. Publisher: Wiley Online Library.

- [16] W. R. Inc. Mathematica, Version 13.2. 2022. Champaign, IL, 2022.
- [17] E. Jenner, E. F. Sanmartín, and F. A. Hamprecht. Extensions of karger’s algorithm: Why they fail in theory and how they are useful in practice. *Proceedings of the IEEE/CVF International Conference on Computer Vision*. 2021 4602–4611.
- [18] D. R. Karger. Global min-cuts in rnc, and other ramifications of a simple min-cut algorithm. *Soda*, vol. 93. Citeseer, 1993 21–30.
- [19] A. Kasperski and P. Zieliński. On the approximability of robust spanning tree problems. *Theoretical Computer Science*, 2011. 412: 365–374.
- [20] G. Kirchhoff. Ueber die auflösung der gleichungen, auf welche man bei der untersuchung der linearen vertheilung galvanischer ströme geführt wird. *Annalen der Physik*, 1847. 148: 497–508.
- [21] J. B. Kruskal. On the Shortest Spanning Subtree of a Graph and the Traveling Salesman Problem. *Proceedings of the American Mathematical Society*, 1956. 7: 48–50. Publisher: American Mathematical Society.
- [22] P. Lippmann, E. Fita Sanmartín, and F. A. Hamprecht. Theory and Approximate Solvers for Branched Optimal Transport with Multiple Sources. *Advances in Neural Information Processing Systems*, 2022. 35: 267–279.
- [23] X. Liu, X. Chen, L. Yang, Q. Chen, J. Guo, and S. Wu. Dynamic topology control in optical satellite networks based on algebraic connectivity. *Acta Astronautica*, 2019. 165: 287–297.
- [24] H. Lyu, N. Sha, S. Qin, M. Yan, Y. Xie, and R. Wang. Manifold denoising by nonlinear robust principal component analysis. H. Wallach, H. Larochelle, A. Beygelzimer, F. Alché-Buc, E. Fox, and R. Garnett, eds., *Advances in Neural Information Processing Systems*, vol. 32. Curran Associates, Inc., 2019 .
- [25] F. Magistri, N. Chebrolu, and C. Stachniss. Segmentation-based 4d registration of plants point clouds for phenotyping. *2020 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE, 2020 2433–2439.
- [26] A. Masone, M. E. Nenni, A. Sforza, and C. Sterle. The minimum routing cost tree problem: State of the art and a core-node based heuristic algorithm. *Soft Computing*, 2019. 23: 2947–2957. Publisher: Springer.
- [27] C. Monma and S. Suri. Transitions in geometric minimum spanning trees. *Proceedings of the seventh annual symposium on Computational geometry*. 1991 239–249.
- [28] M. Niendorf and A. R. Girard. Robustness of communication links for teams of unmanned aircraft by sensitivity analysis of minimum spanning trees. *2016 American Control Conference (ACC)*. IEEE, 2016 4623–4629.

- [29] F. Paul, Y. Arkin, A. Giladi, D. A. Jaitin, E. Kenigsberg, H. Keren-Shaul, D. Winter, D. Lara-Astiaso, M. Gury, and A. Weiner. Transcriptional heterogeneity and lineage commitment in myeloid progenitors. *Cell*, 2015. 163: 1663–1677. Publisher: Elsevier.
- [30] W. Saelens, R. Cannoodt, H. Todorov, and Y. Saeys. A comparison of single-cell trajectory inference methods. *Nature Biotechnology*, 2019. 37: 547–554.
- [31] S. Sattari and F. Didehvar. A metaheuristic algorithm for the minimum routing cost spanning tree problem. *Iranian Journal of Operations Research*, 2015. 6: 65–78. Publisher: Iranian Journal of Operations Research.
- [32] E. Schröder. Vier combinatorische probleme. *Z. Math. Phys*, 1870. 15: 361–376.
- [33] D. Schunck, F. Magistri, R. Rosu, A. Cornelißen, N. Chebrolu, S. Paulus, J. Léon, S. Behnke, C. Stachniss, H. Kuhlmann, and L. Klingbeil. Pheno4D: A spatio-temporal dataset of maize and tomato plant point clouds for phenotyping and advanced plant analysis . 2021. 16: 1–18.
- [34] M. Setty, V. Kiseliovas, J. Levine, A. Gayoso, L. Mazutis, and D. Pe’Er. Characterization of cell fate probabilities in single-cell data with palantir. *Nature biotechnology*, 2019. 37: 451–460.
- [35] P. Sharma, S. Singh, D. Ghosh, and R. Chandrasekaran. Robust discrete spanning tree problem: local search algorithms. *OPSEARCH*, 2022. 59: 632–644.
- [36] W. D. Smith. How to find Steiner minimal trees in Euclidean d-space. *Algorithmica*, 1992. 7: 137–177. Publisher: Springer.
- [37] K. Street, D. Risso, R. B. Fletcher, D. Das, J. Ngai, N. Yosef, E. Purdom, and S. Dudoit. Slingshot: cell lineage and pseudotime inference for single-cell transcriptomics. *BMC genomics*, 2018. 19: 1–16. Publisher: Springer.
- [38] W. Tutte. *Graph Theory*. Encyclopedia of mathematics and its applications. Addison-Wesley Publishing Company, Advanced Book Program, 1984.
- [39] Y. Wang, Y. Xu, S. Tsogkas, X. Bai, S. Dickinson, and K. Siddiqi. Deepflux for skeletons in the wild. *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 2019 5287–5296.
- [40] D. M. Warme, P. Winter, and M. Zachariasen. *Exact algorithms for plane Steiner tree problems: A computational study*. Springer, 2000.
- [41] F. A. Wolf, F. K. Hamey, M. Plass, J. Solana, J. S. Dahlin, B. Göttgens, N. Rajewsky, L. Simon, and F. J. Theis. PAGA: graph abstraction reconciles clustering with trajectory inference through a topology preserving map of single cells. *Genome Biology*, 2019. 20: 59.
- [42] R. T. Wong. Worst-case analysis of network design problem heuristics. *SIAM Journal on Algebraic Discrete Methods*, 1980. 1: 51–63. Publisher: SIAM.

- [43] B. Y. Wu and K.-M. Chao. *Spanning trees and optimization problems*. CRC Press, 2004.
- [44] W. I. Zangwill. Minimum Concave Cost Flows in Certain Networks. *Management Science*, 1968. 14: 429–450.
- [45] G. X. Zheng, J. M. Terry, P. Belgrader, P. Ryvkin, Z. W. Bent, R. Wilson, S. B. Ziraldo, T. D. Wheeler, G. P. McDermott, and J. Zhu. Massively parallel digital transcriptional profiling of single cells. *Nature communications*, 2017. 8: 14049. Publisher: Nature Publishing Group UK London.

# Overview of Appendix Contents

The appendix is organized into the following sections:

- A Stability Examples:** Illustrates the effect of  $\alpha$  upon computing the CST and BCST on different 2-dimensional toy datasets.
- B Applications:** Explores further the applications mentioned in the main paper, namely trajectory inference of single-cell data and 3D plant skeletonization, and provides further examples.
- C Reinterpreting CST as a Minimum Concave Cost Flow Problem:** Presents how the CST can be interpreted as an instance within the broader category of problems posed by the Minimum Concave Cost Network Flow (MCCNF) problems. In addition, it also discusses the relation to the Branched Optimal Transport problem
- D Equivalence of the CST Problem with  $\alpha = 1$  and the Minimum Routing Cost Tree Problem:** Demonstrates the equivalence between the Minimum Routing Cost Tree (MRCT) and the CST when  $\alpha = 1$ .
- E Limit cases of the CST/BCST problems beyond the range  $\alpha \in [0, 1]$ :** Proves the statements regarding the limiting trees of both the CST and BCST as  $\alpha$  approaches negative and positive infinity, and as the number of terminals,  $N$ , approaches infinity.
- F Exploring the Number of Derivable Topologies from CST and BCST Topologies:** Analytically illustrates the quantity of BCST topologies that can be derived from a single CST topology, and vice versa.
- G Branching Angles at the Steiner Points in the BCST problem:** Analytically examines the angles formed by the edges meeting at Steiner points in the BCST problem.
- H Infeasibility of Degree-4 Steiner Points in the Plane for  $\alpha = 1$ :** Proves the infeasibility of degree-4 Steiner points in the 2-dimensional Euclidean BCST problem for  $\alpha = 1$ .
- I Iteratively Reweighted Least Square for the Geometric Optimization of the Steiner Points:** Extends the iteratively reweighted least square approach to compute the optimal positions of the Steiner points presented in [36, 22] to the BCST case.
- J Complexity mSTreg heuristic:** Analyzes the complexity of the mSTreg heuristic.
- K Effect of Additional Intermediate Points in the mSTreg heuristic:** Examines the impact of introducing additional intermediate points, sampled along the edges of the tree, during the mST step in the mSTreg heuristic.
- L Strategies to Transform a Full Tree Topology into a CST Topology:** Analyzes different strategies to transform a BCST topology into a CST topology in order to optimize the CST problem using the mSTreg heuristic.

**M Further Details on the Brute Force Experiment:** Provides additional details on the brute force experiment discussed in Section 7 and examines the behavior of the mSTreg heuristic concerning  $\alpha$ .

**N Selection of  $\alpha$ :** Offers empirical insights to guide the selection of the optimal value for  $\alpha$ .

**O Implementation Details:** Provides further details regarding the implementation.

## A Stability Examples

In this section, we show how stable the BCST and CST for different  $\alpha$  values are. We sample 1000 points uniformly from uniform distributions over different supports and perturb them by adding zero centered Gaussian noise. We generate two perturbations and show how the tree structure evolves across different values. See Figures 18 19 and 20. As we increase the value of  $\alpha$ , the trees exhibit a more pronounced "star-shaped" pattern and enhanced stability. The parameter  $\alpha$  provides a trade-off mechanism between preserving the structure of the data and ensuring the stability of the resulting tree.



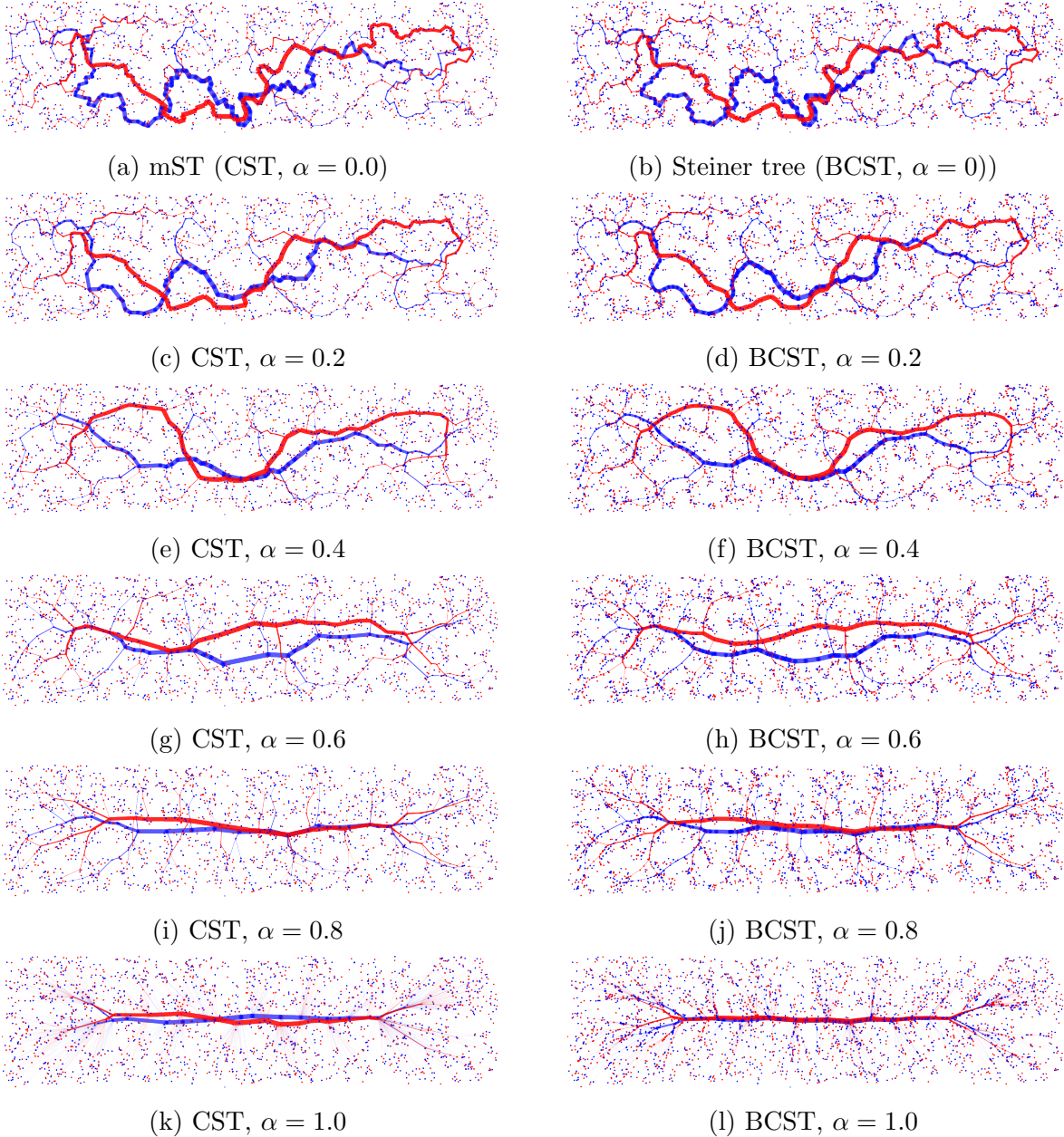


Figure 18: **(B)CST Examples from Rectangle Uniform Distribution.** CST and BCST are computed for two perturbed instances generated by adding zero-centered Gaussian noise to points derived from a common sample uniformly taken within a rectangle. (B)CST for higher  $\alpha$  values are more robust to noise and adhere to large scale structure in the data better. The width of each edge is proportional to its centrality. All trees except for the mST were computed using the heuristic proposed in Section 6.2.

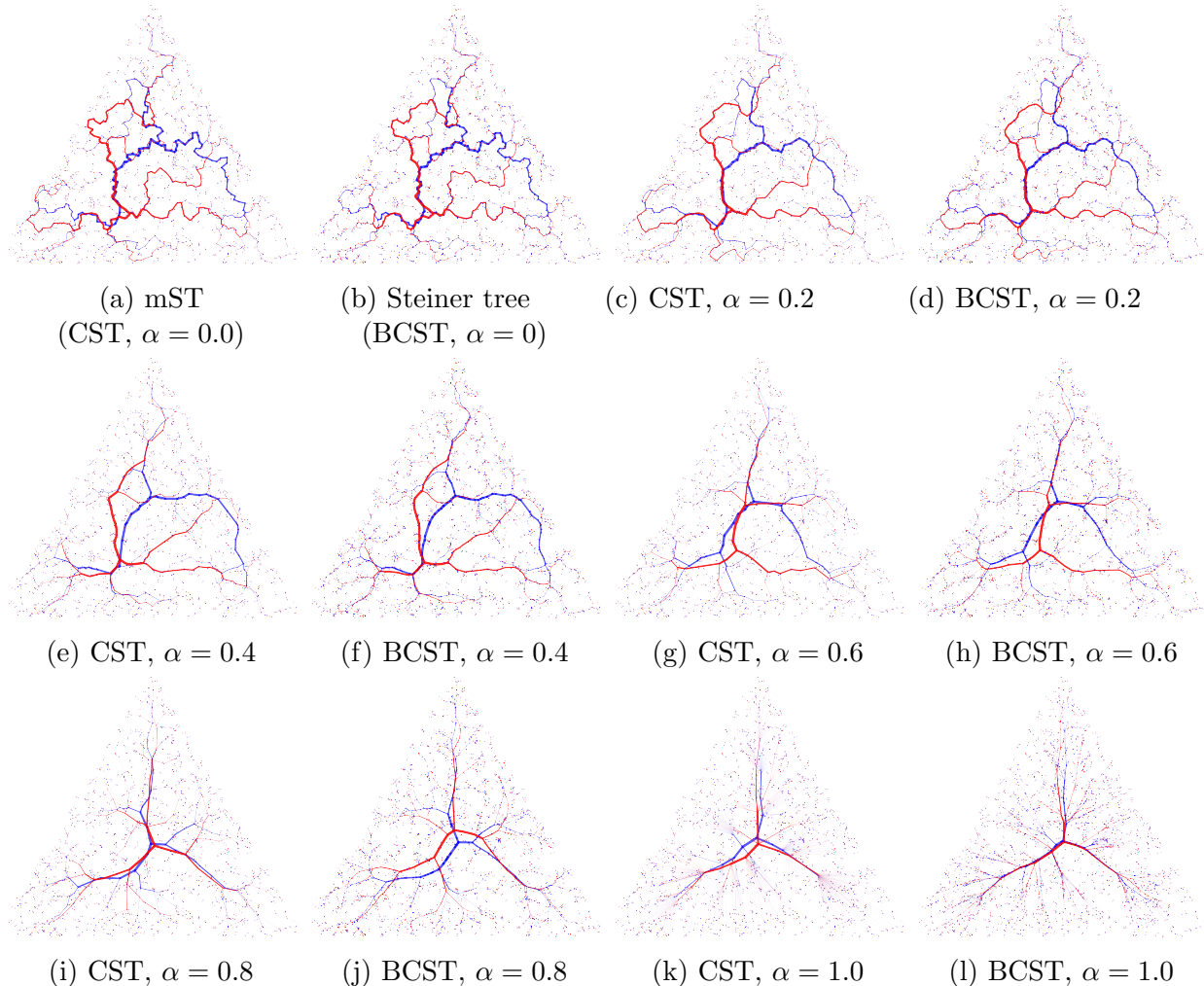


Figure 19: **(B)CST Examples from Triangle Uniform Distribution.** CST and BCST are computed for two perturbed instances generated by adding zero-centered Gaussian noise to points derived from a common sample uniformly taken within a triangle. (B)CST for higher  $\alpha$  values are more robust to noise and adhere to large scale structure in the data better. The width of each edge is proportional to its centrality. All trees except for the mST were computed using the heuristic proposed in Section 6.2.

## B Applications

### B.1 Single cell transcriptomic data

Single-cell transcriptomics analyzes the gene expression levels of individual cells in a particular population by counting the RNA transcripts of genes at a given time. The high dimensional single cell RNA-sequencing data can be used to model the gene expression dynamics of a cell population as well as the cell differentiation process. The reconstruction of these trajectories can help discover which genes are critical to understand the underlying biological process. It is often assumed that these trajectories can be represented as trees [30, 37], and therefore

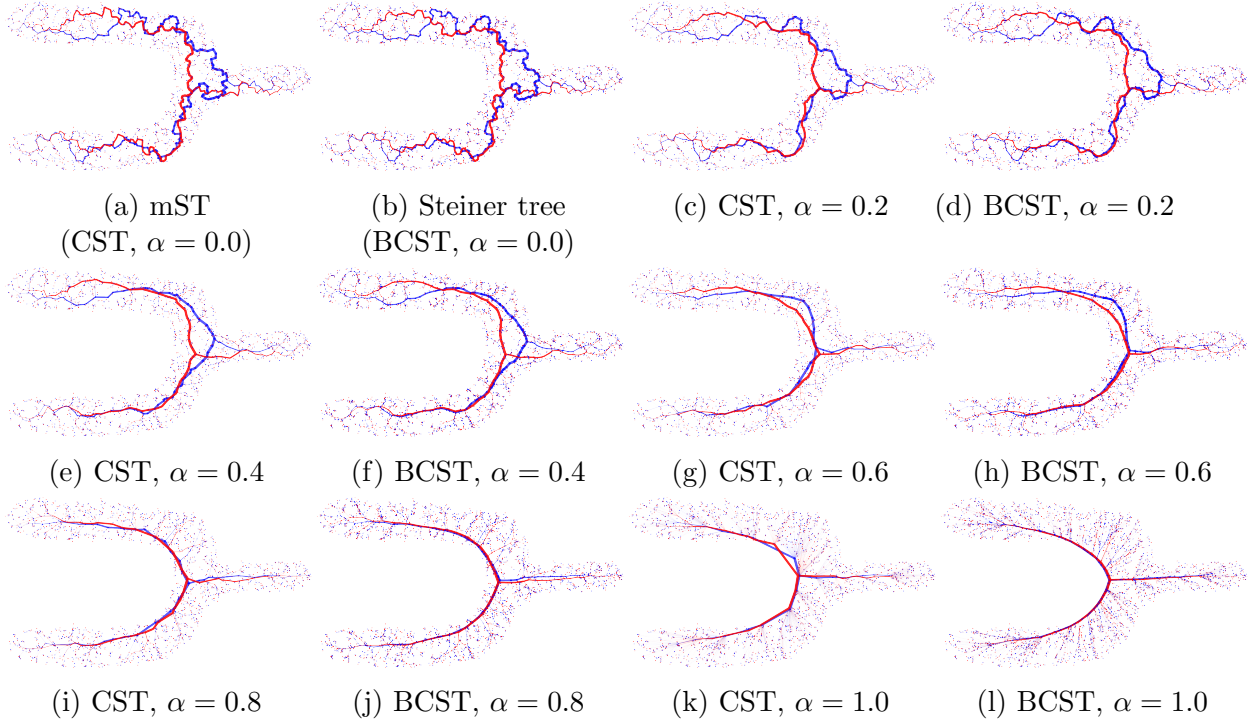


Figure 20: **(B)CST Examples from Non-Convex Uniform Distribution.** CST and BCST are computed for two perturbed instances generated by adding zero-centered Gaussian noise to points derived from a common sample uniformly taken within a non convex shape. (B)CST for higher  $\alpha$  values are more robust to noise and adhere to large scale structure in the data better. The width of each edge is proportional to its centrality. All trees except for the mST were computed using the heuristic proposed in Section 6.2.

the (B)CST can be applied to model such trajectories.

In this section we will give additional details on the example shown in the main paper regarding the Paul dataset. In addition, we show another example of the performance of the (B)CST with a different dataset, namely the Setty dataset [34]

**Paul dataset** The Paul dataset consists of gene expressions measurements of cells of mouse bone marrow Paul et al. [29]. The original dataset is formed by 2730 cells each with 3451 gene measurements. The data is preprocessed using the recipe described in Zheng et al. [45], which reduces the dimensionality to 1000 by selecting the most relevant genes. We further reduce the dimensionality of the data, by applying PCA with 50 principal components. Finally, we apply the corresponding spanning tree algorithm. For visualization purposes, we used the PAGA algorithm [41], one of the best algorithms for single cell trajectory inference [30]. PAGA was designed to faithfully represent the trajectories. Thus, if a spanning tree aligns well with the embedding, this is an indication that the tree approximates the trajectory well. Figure 5 in the main part shows the mST, Steiner tree ( $\alpha = 0$ ) and CST, BCST (at  $\alpha = 0.5$ ) and (B)MRCT ( $\alpha = 1$ ) of the original sample and a perturbed sample with 50% of the cells randomly sampled. The mST and Steiner tree miss the highlighted bifurcation and they are more sensitive to the noise. The CST and BCST are robust to the perturbation

and align well with the PAGA embedding, though the CST may not reconstruct well the finer details. The addition of SPs enables the BCST to follow the trajectory more closely. The MRCT results in a star tree, and its branched version nearly resembles a star tree as well. In high-dimensional data, such as the Paul dataset with 50 dimensions, the star-shaped tendency becomes more prominent. For  $\alpha = 1$ , most of the intricate structure of the data is lost. Therefore, for higher dimensions lower  $\alpha$  values become more relevant.

**Setty dataset:** Extending our analysis, we applied the same experiment from Section 3.2 to a different real dataset: the human bone marrow dataset from Setty et al. [34] referred here as the Setti dataset. After applying the same preprocessing as the one used for the Paul dataset, this dataset consists of 5780 samples within a 50-dimensional space, as opposed to the unprocessed data, which existed in a 14651-dimensional space. We computed BCST, CST, Steiner tree and mST on both the original dataset and a downsampled version (25% of the original), comparable in size to the Paul dataset’s downsampled version. For visualization purposes we used the TSNE projection of the data provided by the scvelo library.<sup>7</sup> Figure 21 presents the results, showing a consistent pattern akin to the Paul dataset. BCST and CST exhibited superior robustness compared to mST and Steiner tree, with BCST providing enhanced trajectory modeling.<sup>8</sup> This additional example reinforces the efficacy and robustness of our proposed methods.

## B.2 3D Plant skeletonization

Plant skeletonization is a foundational technique for elucidating growth patterns, branching hierarchies, and responses to environmental factors in plants. It simplifies complex plant structures into skeletal representations, often described by spanning trees. In this section, we demonstrate how the BCST can model a plant’s skeleton using a point cloud of its surface.<sup>9</sup> We provide additional examples beyond those presented in the main paper.

For the skeletonization process, we utilized the 4D Plant Registration Dataset.<sup>10</sup> It consists of 3D point cloud data that captures the surface of different plants at different growth stages. Specifically, our analysis focused on the point clouds of the tomato plant on days 5, 8, and 13. From each point cloud, we subsample uniformly at random 5000 points and then we compute the BCST.

Figure 22 presents the BCST results computed for different  $\alpha$  values. When  $\alpha = 0.00$ , the tree branches exhibit greater irregularity, while at  $\alpha = 1.00$ , the finer details are obscured. Intermediate  $\alpha$  values offer a more faithful representation of the plant’s structure. However, some modeled branches deviate from the data, creating shortcut connections between points. This issue can be readily addressed by incorporating prior information about the root’s

<sup>7</sup><https://scvelo.readthedocs.io/en/stable/scvelo.datasets.bonemarrow/>

<sup>8</sup>In this particular case, we have excluded the  $\alpha = 1.00$  scenario. This is because, similar to the Paul dataset, both the CST and BCST algorithms yield star-shaped graphs under this condition.

<sup>9</sup>Note that we do not test the CST in this context, as the ideal skeleton should closely follow the object’s centerline. While the BCST has Steiner points, which naturally align with the center of the surrounding points to minimize the distance, the CST lacks this flexibility. Consequently, the backbone of the CST can not align with the centerline, given that the terminals lie on the surface.

<sup>10</sup>Data accessible at <https://www.ipb.uni-bonn.de/data/4d-plant-registration/>

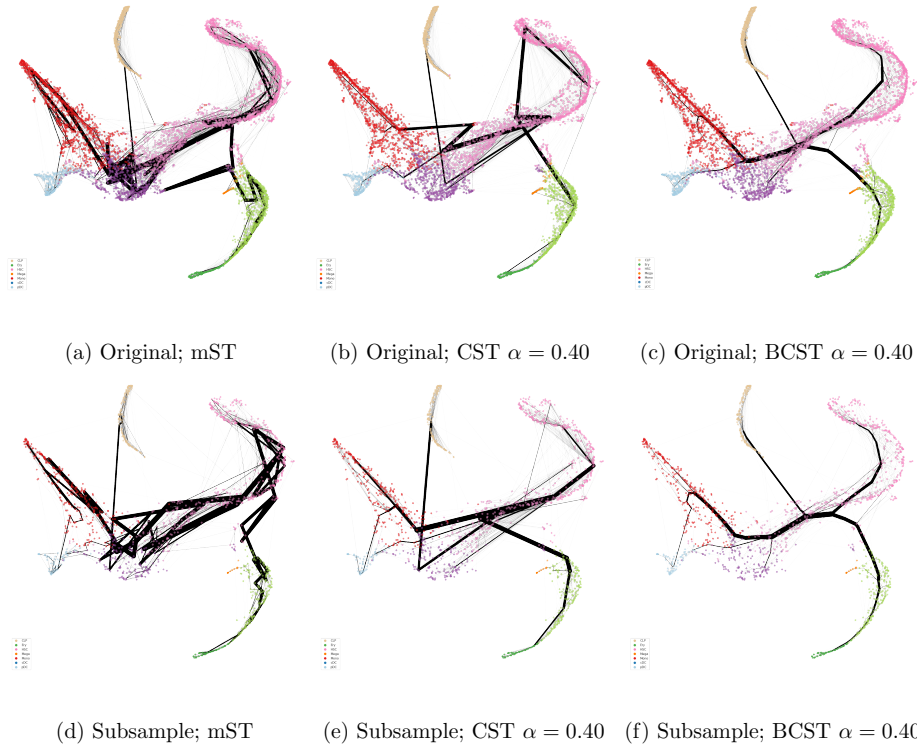


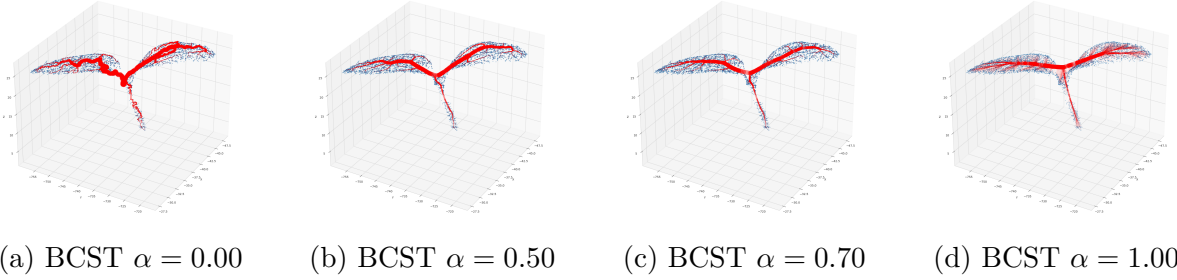
Figure 21: **mST, CST and BCST of the the Setty dataset [34]**. We applied the algorithms on the original data (top) and perturbed version where 25% of the points have been randomly removed (bottom). We use the TSNE algorithm to visualize the data in 2D, though the trees were computed on a 50 dimensional PCA projection of the preprocessed data. Colors correspond to different ground truth cell populations. The width of the edges is proportional to their centralities. The mST fails to accurately represent the trajectory and proves to be highly susceptible to noise. The CST and BCST are more faithful and robust, though the BCST performance is superior.

position within the plant. By augmenting the point density near the root through virtual point replication (creating five times as many virtual points as there are original points at the root location), we encourage the branches to closely follow the natural branch density, resulting in a more faithful representation of the data. That is, if the original point cloud contains 5000 points, we introduce 25000 virtual points at the root location.

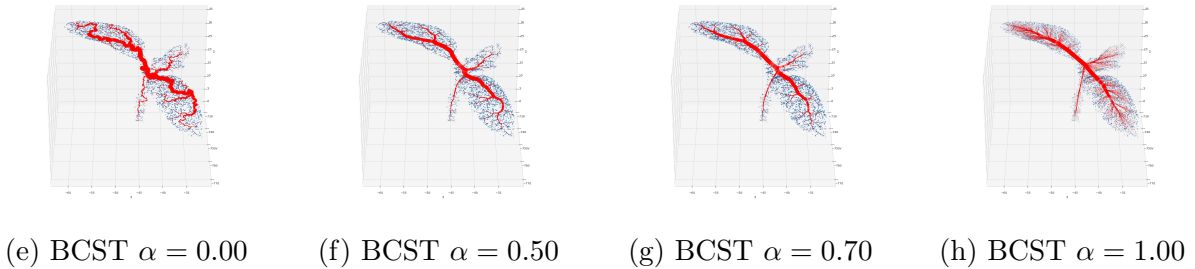
Figure 23 displays the results after incorporating this prior information for the point cloud of the plant at different growth stages.<sup>11</sup> Notably, the model exhibits greater fidelity for intermediate  $\alpha$  values when the prior is applied.

<sup>11</sup>Day 8 of Figure 23 corresponds to Figure 6 in the main paper

Tomato plant 2, day 5.



Tomato plant 2, day 8.



Tomato plant 2, day 13.

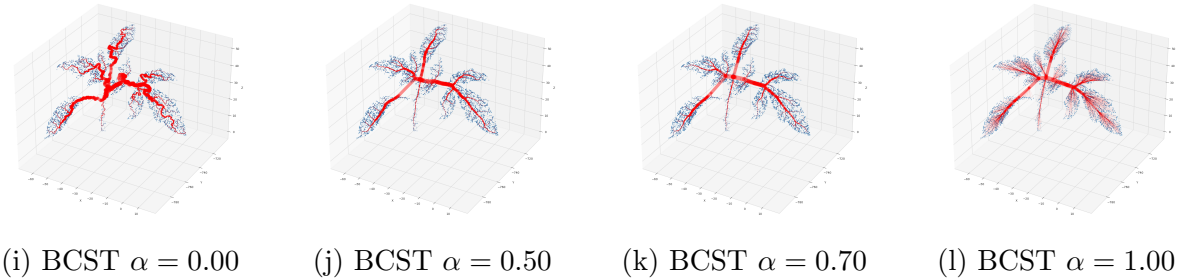


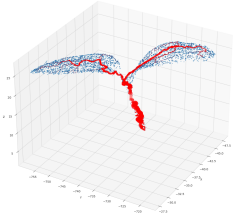
Figure 22: Skeletons at different  $\alpha$  values of 3D point clouds of a tomato plant at different growth stages. The skeletons are modeled using the BCST with varying  $\alpha$  values. With  $\alpha = 0.00$ , the tree branches exhibit greater irregularity, while at  $\alpha = 1.00$  the finer details are missed. Intermediate  $\alpha$  values offer a more faithful representation of the plant’s structure. Nonetheless these may present some slight deviation, where some modeled branches do not align through the center of the actual branches (see day 13). This effect is alleviated once prior information concerning the root’s location is added (see Figure 23).

## C Reinterpreting CST as a Minimum Concave Cost Flow

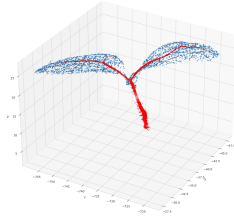
In this section, we will pose the central spanning tree (CST) problem as a minimum concave cost network flow (MCCNF) problem. The MCCNF problem minimizes the transportation



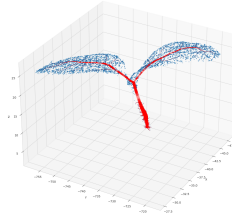
Tomato plant 2, day 5.



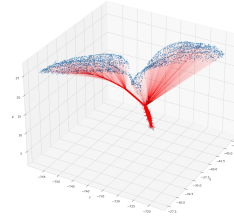
(a) BCST  $\alpha = 0.25$



(b) BCST  $\alpha = 0.50$

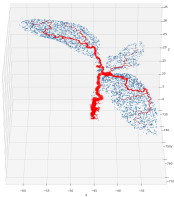


(c) BCST  $\alpha = 0.70$

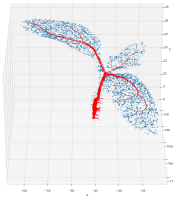


(d) BCST  $\alpha = 1.00$

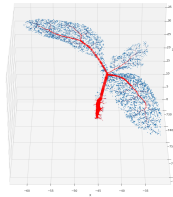
Tomato plant 2, day 8.



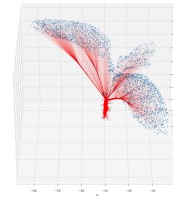
(e) BCST  $\alpha = 0.00$



(f) BCST  $\alpha = 0.50$

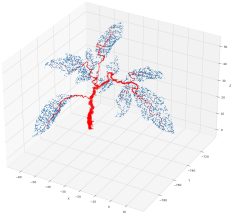


(g) BCST  $\alpha = 0.70$

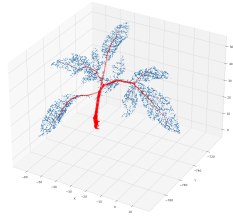


(h) BCST  $\alpha = 1.00$

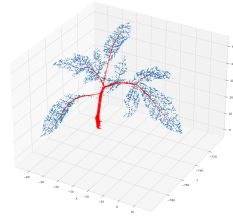
Tomato plant 2, day 13.



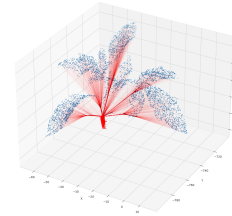
(i) BCST  $\alpha = 0.00$



(j) BCST  $\alpha = 0.50$



(k) BCST  $\alpha = 0.70$



(l) BCST  $\alpha = 1.00$

Figure 23: Skeletons at different  $\alpha$  values of 3D point clouds of a tomato plant at different growth stages. The skeletons are modeled using the BCST with varying  $\alpha$  values, incorporating prior information about the root's location. In contrast to Figure 22, we have enhanced the density of points at the root location, by virtually augmenting the number of points at the root's coordinates. This enhancement results in a more accurate and faithful representation of the plant's skeleton.

cost of a commodity from sources to sinks. Here, the edge costs are modeled by concave functions that depend on the edge flow. Formally, given a demand vector  $\mu \in \mathbb{R}^N$  with

$\sum_{i=1}^N \mu_i = 0$  and a network  $G = (V, E)$  with  $N$  nodes, we define the MCCNF problem as

$$\begin{aligned} \min_f \quad & \sum_{ij \in E} C_{ij}(f_{ij}), \quad \text{subject to} \\ & \sum_{(i,j) \in E} f_{ij} - \sum_{(j,i) \in E} f_{ji} = \mu_i, \quad \forall i \in V. \\ & f_{ij} \geq 0 \end{aligned} \tag{18}$$

In the equation,  $f_{ij}$  represents the flow associated with edge  $(i, j)$  and  $C_{ij}$  is a concave function dependent on  $f_{ij}$  which determines the cost of the edge  $(i, j)$ . Note that the network defined by the flow, i.e. by the edges with  $f_{ij} > 0$ , does not necessarily have to be a tree. We will refer to nodes with negative demands as sources and nodes with positive demands as sinks.

To be able to represent the CST problem as an MCCNF, we need to identify the terms  $m_e$  as flows. Since the function  $(m_e(1 - m_e))^\alpha c_e$  is concave for  $\alpha \in [0, 1]$ , it will follow that the CST is an instance of the MCCNF problem.

Next, we will show how the  $m_e$  can be interpreted as the flow along an edge of a particular single source flow problem. Consider a graph with  $N$  nodes, where there is one source node  $s$  with a mass  $(N - 1)/N$  that needs to be transported to the rest of the nodes. Each sink node has a demand of  $1/N$  mass. Thus, (18) becomes

$$\begin{aligned} \min_x \quad & \sum_{ij \in E} c_{ij}(f_{ij}(1 - f_{ij}))^\alpha, \quad \text{subject to} \\ & \sum_{(j,s) \in E} f_{js} - \sum_{(s,j) \in E} f_{sj} = \frac{N - 1}{N}, \\ & \sum_{(i,j) \in E} f_{ij} - \sum_{(j,i) \in E} f_{ji} = \frac{1}{N}, \quad \forall i \neq s \\ & f_{ij} \geq 0 \end{aligned} \tag{19}$$

For single source flow problems, it is well-known that the optimal solutions can only form trees [44]. We will show that for this particular problem,  $f_{ij}$  is equal to  $m_{ij}$  for any tree. Recall that for a given tree  $T$ , the value  $m_e$  associated with an edge  $e$  was defined as the number of nodes that lie on one of the sides of  $e$  divided by the total number of nodes. Although for our purposes the chosen side of the edge is arbitrary, since the objective function is symmetrized thanks to being multiplied by  $(1 - m_e)$ , that is not the case when we want to interpret it as a flow. However, which side to choose will be canonically determined by the flow direction.

Let  $T$  be a feasible solution of (19), i.e. a tree. The flow  $f_{ij}$  at edge  $(i, j)$  of  $T$  indicates the outgoing mass that is transported through the edge. This mass is equal to the sum of the demands of the nodes that lie in the side of the edge  $(i, j)$  indicated by the flow. Given that each node has a demand of  $1/N$ , then the flow  $f_{ij}$  is equal to  $1/N$  multiplied by the number of nodes in the side in question, that is  $f_{ij} = m_{ij}$ .



## C.1 Relation to the Branched Optimal Transport Problem

The branched or irrigation optimal transport (BOT) problem is also a particular MCCNF instance. In the BOT problem, the nodes are embedded in a Euclidean space and also allows for the inclusion of additional Steiner points. It is an extension of the optimal transport problem, distinguished by its diminishing costs which lead to a branching effect by promoting the joint transportation of mass.

Formally, the BOT problem minimizes

$$\begin{aligned} \min_{x_B, m_E} \quad & \sum_{(i,j) \in E} m_{ij}^\alpha \|x_i - x_j\|_2, \text{ subject to} \\ & \sum_{(i,j) \in E} m_{ij} - \sum_{(j,i) \in E} m_{ji} = \mu_i, \quad \forall i \in V \cup B, \\ & m_{ij} \geq 0 \end{aligned} \tag{20}$$

where  $m_{ij}$  is the flow transported along an edge  $(i, j)$  and  $B$  and  $x_B$  are the set of SPs and their coordinates, respectively. As before, the vector  $\mu$  represents the demands of the nodes. The demands of the SPs are set to zero.

In the scenario where there is a single source and all nodes share the same demand, the BCST and BOT problems differ only in the factors that multiply the distances. In the BOT problem, these factors correspond to  $m_{ij}^\alpha$ , representing the mass transported along an edge raised to the power of  $\alpha$ . In the BCST problem, the factors are given by  $(m_{ij}(1 - m_{ij}))^\alpha$ , representing the centralities of the edges raised to the power of  $\alpha$ . It is worth mentioning, that both problems converge to the Steiner tree problem when  $\alpha = 0$ .

The primary distinction between the two problems lies in the selection of a source node. In the BOT problem, the selection of the source node determines the optimal topology of the network. However, that is not the case for the BCST problem. Indeed, for the BCST problem, the specific source node chosen is irrelevant due to the symmetrization effect caused by the term  $m_e(1 - m_e)$ . In other words, the location of the source determines the edge orientation, which then defines the value of  $m_e$ . Nonetheless, this effect is nullified when multiplied by  $(1 - m_e)$ . This independence of the source node choice makes the BCST a more natural extension of the Steiner tree problem, since it does not require the choice of sources and sinks, unlike the BOT problem.

## D Equivalence of the CST Problem with $\alpha = 1$ and the Minimum Routing Cost Tree Problem

As mentioned in the main text, the term  $m_{ij}(1 - m_{ij})$  is proportional to the betweenness centrality of the edge  $(i, j)$ . This centrality quantifies the number of shortest paths that traverse the given edge. Thus, the multiplication of the length of each edge by its frequency in a shortest path is a rearrangement of the sum over all shortest path costs, i.e. the MRCT

cost. Formally,

$$\begin{aligned}
\sum_{i,j \in V \times V} d_T(i,j) &= \sum_{i,j \in V \times V} \sum_{(u,v) \in \wp_{ij}} \|x_u - x_v\| \\
&= \sum_{(u,v) \in T} \underbrace{|\{\wp_{ij} : (u,v) \in \wp_{ij}, i,j \in V \times V\}|}_{(u,v) \text{ betweenness centrality}} \|x_u - x_v\| \\
&\propto \sum_{(u,v) \in T} m_{uv}(1 - m_{uv}) \|x_u - x_v\|,
\end{aligned} \tag{21}$$

where  $d_T(i,j)$  is the shortest path distance in tree  $T$  between  $i$  and  $j$  realized by the path  $\wp_{ij}$ .

## E Limit Cases of the CST/BCST Problems Beyond the Range $\alpha \in [0, 1]$

In this section, we investigate the topologies of the limit cases of the CST as  $\alpha$  approaches  $\pm\infty$ . We will use the following notation.

- $N$  will represent the number of terminals.
- For a given tree  $T$  containing edge  $(x,y)$ ,  $m_{xy}^T$  indicates the proportion of nodes (normalized by  $N$ ) that are reachable from  $x$ , once edge  $(x,y)$  is removed from  $T$ . That is, the normalized number of nodes that lie in the side of  $x$ .
- For a given tree  $T$ , the term  $\mathcal{N}_x$  denotes the set of neighbors of  $x$  in  $T$

### E.1 Proof Theorem 2.1

Theorem 2.1 states that when a “stronger” variant of the triangle inequality holds, then the optimum solution of the BCST problem is a star tree. We divide the proof in two lemmas, Lemma *E.1* that proves it for the CST problem; and Lemma *E.2* for the BCST case.

**Lemma E.1.** Given a complete graph  $G$  with  $N$  nodes, let  $c_{ij}$  be the edge-costs of any pair of nodes  $i, j$  in the graph. If there exists

$$t \leq \min_{\ell \in [2, N/2]} \frac{\left(\frac{\ell(N-\ell)}{N-1}\right)^\alpha - 1}{\ell - 1}$$

such that

$$c_{kv} + tc_{uv} \geq c_{ku} \tag{22}$$

for all triangles in the graph, then there exists an optimum CST evaluated at  $\alpha$  which is a star tree.

*Proof.* We will show that given a tree  $T$ , we can always increase the degree of a particular node without increasing the CST-cost. Without loss of generality, we can assume  $n \geq 4$ , otherwise, any possible tree is a star-tree and the result holds trivially.

Let us assume that  $T$  is not a star-tree. Thus, there exist at least two nodes,  $u$  and  $v$ , with degrees higher than 1, such that they are adjacent to each other. Moreover, we can assume that one of them, say  $v$ , is an extreme inner node, meaning that all its neighbors (except  $u$ ) have degree 1. Without loss of generality, we can assume that  $\ell := |\mathcal{N}_v| \leq N/2$ , where  $\mathcal{N}_v$  is the set of neighbors of  $v$  in  $T$ . Otherwise, we could have chosen a different extreme inner node. Note that the centrality of the edge  $(u, v)$  is  $m_{uv}(1 - m_{uv}) = \frac{\ell(N-\ell)}{N^2}$ .

We will show that the topology  $T'$  which connects all  $k \in \mathcal{N}_v \setminus \{u\}$  to  $u$  instead of  $v$  has a lower CST cost. The only edge centralities affected by this change are those associated with the edges  $(u, v)$ ,  $(u, k)$ , and  $(k, v)$  for all  $k \in \mathcal{N}_v \setminus \{u\}$ . To compare the costs of the topologies, it suffices to focus on these specific edges.

First, let's determine the values of the centralities for the edges in both trees:

- Normalized centrality of edge  $(u, v)$  in  $T$ :

$$m_{uv}^T(1 - m_{uv}^T) = \frac{\ell(N - \ell)}{N^2}$$

- Normalized centrality of edge  $(u, v)$  in  $T'$ :

$$m_{uv}^{T'}(1 - m_{uv}^{T'}) = \frac{N - 1}{N^2}$$

Note that  $v$  has become a leaf of  $T'$ .

- Normalized centrality of edge  $(k, v)$  in  $T$  and centrality of edge  $(k, u)$  in  $T'$  for all  $k \in \mathcal{N}_v \setminus \{u\}$ :

$$m_{kv}^T(1 - m_{kv}^T) = \frac{N - 1}{N^2} = m_{ku}^{T'}(1 - m_{ku}^{T'})$$

These nodes are leaves in both trees.

The difference between the costs of the topologies is

$$\begin{aligned} \text{CST}(T) - \text{CST}(T') &= c_{uv} \left( \frac{\ell(N - \ell)}{N^2} \right)^\alpha + \sum_{\substack{k \in \mathcal{N}_v \\ k \neq u}} c_{kv} \left( \frac{N - 1}{N^2} \right)^\alpha \\ &\quad - c_{uv} \left( \frac{N - 1}{N^2} \right)^\alpha - \sum_{\substack{k \in \mathcal{N}_v \\ k \neq u}} c_{ku} \left( \frac{N - 1}{N^2} \right)^\alpha \\ &= c_{uv} \left( \left( \frac{\ell(N - \ell)}{N^2} \right)^\alpha - \left( \frac{N - 1}{N^2} \right)^\alpha \right) + \sum_{\substack{k \in \mathcal{N}_v \\ k \neq u}} (c_{kv} - c_{ku}) \left( \frac{N - 1}{N^2} \right)^\alpha \\ &= \sum_{\substack{k \in \mathcal{N}_v \\ k \neq u}} \left[ \frac{c_{uv} \left( \left( \frac{\ell(N - \ell)}{N^2} \right)^\alpha - \left( \frac{N - 1}{N^2} \right)^\alpha \right)}{\ell - 1} + (c_{kv} - c_{ku}) \left( \frac{N - 1}{N^2} \right)^\alpha \right] \\ &= \left( \frac{N - 1}{N^2} \right)^\alpha \sum_{\substack{k \in \mathcal{N}_v \\ k \neq u}} \left[ \frac{c_{uv} \left( \left( \frac{\ell(N - \ell)}{N - 1} \right)^\alpha - 1 \right)}{\ell - 1} + (c_{kv} - c_{ku}) \right] \end{aligned} \tag{23}$$

Thus the decrease in cost will be positive if each term in the summand of the last equality of (23) is positive, namely

$$c_{kv} + \frac{\left(\left(\frac{\ell(N-\ell)}{N-1}\right)^\alpha - 1\right)}{\ell - 1} c_{uv} \geq c_{ku}, \quad (24)$$

which holds by assumption. Therefore,  $T'$  will have a lower cost. Repeating this process, we can always decrease the cost till we form a star tree.  $\square$

The next result extends Lemma E.1 to be applicable to the BCST problem. In contrast to the CST case, the BCST involves Steiner points, which must be treated differently. Lemma E.2 shows that if the “strong” triangle inequality holds, we can collapse sequentially all Steiner points while decreasing the BCST cost of a tree  $T$ .

**Lemma E.2.** Consider a solution  $T$  of the BCST problem with  $N$  terminals. Let  $c_{ij}$  be the edge-costs of any pair of nodes  $i, j$  in the graph (Steiner or terminals). If there exists

$$t \leq \min_{\ell \in [2, N/2]} \frac{\left(\frac{\ell(N-\ell)}{N-1}\right)^\alpha - 1}{\ell - 1}$$

such that

$$c_{kv} + t c_{uv} \geq c_{ku} \quad (25)$$

for all triangles, then there exists a star tree with lower cost.

*Proof.* Analogously to Lemma E.1 we will show that given a tree  $T$ , we can always increase the degree of a particular node without increasing the BCST-cost.

Let us assume that  $T$  is not a star-tree. Thus, there exist at least two nodes,  $u$  and  $v$ , with degree higher than 1 which are adjacent to each other. Moreover, we can assume that one of them, say  $v$ , is an extreme inner node, meaning that all its neighbors (except  $u$ ) have degree 1. Without loss of generality, we can assume that  $m_{uv} := \frac{\ell}{N^2} \leq \frac{1}{2N}$ . Otherwise, we could have chosen a different extreme inner node.

If  $v$  is a terminal node, we can apply the same reasoning as in Lemma E.1 to increase the degree of  $u$ . Let us assume then that  $v$  is a Steiner point. In this case, we will construct a new topology  $T'$  by collapsing  $v$  with  $u$ . This implies that the edge  $(u, v)$  will disappear and that all  $k \in \mathcal{N}_v \setminus \{u\}$  will be connected to  $u$ . In this case the normalized centralities of the edges are not changed.

- Normalized centrality of edge  $(u, v)$  in  $T$ :

$$m_{uv}^T (1 - m_{uv}^T) = \frac{\ell(N - \ell)}{N^2}$$

- Edge  $(u, v)$  is not anymore in  $T'$ :
- Normalized centrality of edge  $(k, v)$  in  $T$  and centrality of edge  $(k, u)$  in  $T'$  for all  $k \in \mathcal{N}_v \setminus \{u\}$ :

$$m_{kv}^T (1 - m_{kv}^T) = \frac{N - 1}{N^2} = m_{ku}^{T'} (1 - m_{ku}^{T'})$$

These nodes are leaves in both trees.

The difference between the costs of the topologies is

$$\begin{aligned}
\text{CST}(T) - \text{CST}(T') &= c_{uv} \left( \frac{\ell(N-\ell)}{N^2} \right)^\alpha + \sum_{\substack{k \in \mathcal{N}_v \\ k \neq u}} c_{kv} \left( \frac{N-1}{N^2} \right)^\alpha - \sum_{\substack{k \in \mathcal{N}_v \\ k \neq u}} c_{ku} \left( \frac{N-1}{N^2} \right)^\alpha \\
&= c_{uv} \left( \frac{\ell(N-\ell)}{N^2} \right)^\alpha + \sum_{\substack{k \in \mathcal{N}_v \\ k \neq u}} (c_{kv} - c_{ku}) \left( \frac{N-1}{N^2} \right)^\alpha \\
&= \sum_{\substack{k \in \mathcal{N}_v \\ k \neq u}} \left[ \frac{c_{uv} \left( \left( \frac{\ell(N-\ell)}{N^2} \right)^\alpha \right)}{\ell-1} + (c_{kv} - c_{ku}) \left( \frac{N-1}{N^2} \right)^\alpha \right] \\
&= \left( \frac{N-1}{N^2} \right)^\alpha \sum_{\substack{k \in \mathcal{N}_v \\ k \neq u}} \left[ \frac{c_{uv} \left( \left( \frac{\ell(N-\ell)}{N-1} \right)^\alpha \right)}{\ell-1} + (c_{kv} - c_{ku}) \right]
\end{aligned} \tag{26}$$

Thus the decrease in cost will be positive if each term in the summand of the last equality of (26) is positive, namely

$$c_{kv} + \frac{\left( \frac{\ell(N-\ell)}{N-1} \right)^\alpha}{\ell-1} c_{uv} \geq c_{ku}, \tag{27}$$

which holds by assumption, since

$$\frac{\left( \frac{\ell(N-\ell)}{N-1} \right)^\alpha}{\ell-1} \geq \min_{\ell \in [2, N/2]} \frac{\left( \frac{\ell(N-\ell)}{N-1} \right)^\alpha}{\ell-1} > \min_{\ell \in [2, N/2]} \frac{\left( \frac{\ell(N-\ell)}{N-1} \right)^\alpha}{\ell-1} - 1.$$

Therefore,  $T'$  will have a lower cost. Repeating this process, we can always decrease the cost till we form a star tree.  $\square$

**Remark E.1.** Note that Lemma E.1 and Lemma E.2 state only a sufficient condition, which means that the optimum can be a star tree even if the strong triangle inequality does not hold. Additionally, it is worth to highlight that Lemma E.1 also holds true for the CST problem even when the nodes lack embedding in any specific space, allowing for edge costs with arbitrary values.

## E.2 Proof $h_1(\ell, N, \alpha) > 1$ as $N$ Approaches Infinity, for $\alpha > 1$

Recall that  $h_1$  is defined as

$$h_1(\ell, N, \alpha) := \frac{\left( \left( \frac{\ell(N-\ell)}{N-1} \right)^\alpha - 1 \right)}{\ell-1} = \frac{\left( 1 + \frac{(\ell-1)(N-\ell-1)}{N-1} \right)^\alpha - 1}{\ell-1}.$$

We will show that for high enough  $N$ ,  $\min_{\ell \in [2, N/2]} h_1(\ell, N, \alpha) > 1$ . By leveraging the Mathematica software [16], we can establish that the function  $h_1(\ell, N, \alpha)$  exhibits concavity concerning  $\ell$  within the interval  $[2, N/2]$  under the conditions  $\alpha > 1$  and  $N > 3$ . Consequently, for fixed values of  $\alpha > 1$  and  $N$ , the minimum of  $h$  is achieved either at  $\ell = 2$  or at  $\ell = N/2$ . Next we show that as  $N$  tends to infinity, both evaluations tend towards a value greater than 1.

- When  $\ell = 2$  we have

$$h_1(2, N, \alpha) = \left(1 + \frac{N-3}{N-1}\right)^\alpha - 1 \xrightarrow{N \rightarrow \infty} 2^\alpha - 1 > 1 \quad (28)$$

- When  $\ell = N/2$  we have

$$h_1(N/2, N, \alpha) = \frac{\left(1 + \frac{(N/2-1)^2}{N-1}\right)^\alpha - 1}{N/2 - 1} \xrightarrow{N \rightarrow \infty} \infty \quad (29)$$

Therefore,  $h_1(\ell, N, \alpha) > 1$  as  $N$  approaches infinity and  $\alpha > 1$ .

Combining this inequality with Theorem 2.1, we conclude that the optimum (B)CST will be a tree as  $N$  approaches infinity and  $\alpha > 1$ .

### E.3 Computation $\alpha^*(N)$

Given  $N$ , recall that  $\alpha^*(N)$  is the minimum  $\alpha$  at which  $h_1(\ell, N, \alpha) > 1$  for all  $\ell \in [2, N/2]$ . Since  $h_1$  is concave with respect to  $\ell$  when  $\alpha > 1$ , our focus narrows down to investigating the cases  $\ell = 2$  and  $\ell = N/2$  –the values where the minima can be attained.

- When  $\ell = 2$  we have

$$\begin{aligned} h_1(2, N, \alpha) = \left(1 + \frac{N-3}{N-1}\right)^\alpha - 1 > 1 &\iff \left(1 + \frac{N-3}{N-1}\right)^\alpha > 2 \\ &\iff \alpha \log\left(1 + \frac{N-3}{N-1}\right) > \log(2) \\ &\iff \alpha > \frac{\log(2)}{\log\left(1 + \frac{N-3}{N-1}\right)} \end{aligned} \quad (30)$$

- When  $\ell = N/2$  we have

$$\begin{aligned} h_1(N/2, N, \alpha) = \frac{\left(1 + \frac{(N/2-1)^2}{N-1}\right)^\alpha - 1}{N/2 - 1} > 1 &\iff \left(1 + \frac{(N/2-1)^2}{N-1}\right)^\alpha > N/2 \\ &\iff \alpha \log\left(1 + \frac{(N/2-1)^2}{N-1}\right) > \log(N/2) \\ &\iff \alpha > \frac{\log(N/2)}{\log\left(1 + \frac{(N/2-1)^2}{N-1}\right)} \end{aligned} \quad (31)$$

Therefore,

$$\alpha^*(N) := \max\left(\frac{\log(2)}{\log\left(1 + \frac{N-3}{N-1}\right)}, \frac{\log(N/2)}{\log\left(1 + \frac{(N/2-1)^2}{N-1}\right)}\right) \quad (32)$$

## E.4 Proof Theorem 2.2

In this section we prove Theorem 2.2 which states that if a variant of the triangle inequality holds, then the optimum (B)CST tree will be a path as  $\alpha$  approaches infinity. First, Lemma E.3 will show that if the triangle inequality holds strictly, then for  $\alpha$  negative enough the optimum CST will be a path. Corollary E.1 demonstrates that when the nodes are embedded in a geodesic space, the triangle inequality does not need to hold strictly for Lemma E.3 to be true. Theorem 2.2 will also be derived as corollary (Corollary E.2).

**Lemma E.3.** Let  $G$  be a complete graph with edge-costs satisfying the condition of the strict triangle inequality for every triplet of nodes  $(u, v, k)$ , defined as

$$c_{uv} + c_{kv} < c_{ku}$$

As the parameter  $\alpha$  approaches negative infinity ( $\alpha \rightarrow -\infty$ ), there exists a Hamiltonian path  $T_\star$  in  $G$  with a lower CST cost than any other tree  $T$  that is not a path.

*Proof.* We will show that for any node  $v$  with degree higher than 2, we can always decrease its degree such that the CST cost of  $T$  decreases. By iteratively applying this process, we ensure that the degrees of all nodes will eventually be reduced to at most 2, culminating in the formation of a path.

Let  $v$  be a node with degree higher than 3. Let  $(k, v)$  and  $(u, v)$  be the two edges adjacent to  $v$  and assume w.l.o.g that  $m_{uv}^T = \min_{i \in \mathcal{N}_v} m_{iv}$ . Since  $m_{kv}^T \geq m_{uv}^T$ , we have

$$(m_{kv}^T(1 - m_{kv}^T)) \geq (m_{uv}^T(1 - m_{uv}^T)).$$

We will now demonstrate that the modified topology  $T'$ , where node  $k$  is connected to node  $u$  instead of node  $v$ , results in a lower CST cost. The only edge centralities affected by this change are those associated with edges  $(u, v)$ ,  $(u, k)$ , and  $(k, v)$ . To compare the costs of the topologies, it suffices to focus on these specific edges.

First, we determine the values of the centralities for these edges in both trees.

- Normalized centrality of edge  $(u, v)$  in tree  $T$ :

$$m_{uv}^T(1 - m_{uv}^T)$$

- Normalized centrality of edge  $(u, v)$  in tree  $T'$ :

$$m_{uv}^{T'}(1 - m_{uv}^{T'}) = (m_{uv}^T + m_{kv}^T)(1 - m_{uv}^T - m_{kv}^T)$$

The equality is due to the fact that once  $k$  is a neighbor of  $u$ , all nodes that were in the same side as  $k$  will be now in the same side as  $u$ .

- Normalized centrality edge  $k, v$  in  $T$  and normalized centrality edge  $k, u$  in  $T'$ :

$$m_{kv}^T(1 - m_{kv}^T) = m_{ku}^{T'}(1 - m_{ku}^{T'})$$

Both  $u$  and  $v$  lie in the same side of the edges, hence the equality of their normalized centralities.

Note that  $m_{uv}^{T'}(1 - m_{uv}^{T'}) > m_{uv}^T(1 - m_{uv}^T)$ . Otherwise, it would imply that

$$\begin{aligned} m_{uv}^{T'}(1 - m_{uv}^{T'}) &< m_{uv}^T(1 - m_{uv}^T) \iff \\ (m_{uv}^T + m_{kv}^T)(1 - m_{uv}^T - m_{kv}^T) &< m_{uv}^T(1 - m_{uv}^T) \iff \\ \min(m_{uv}^T + m_{kv}^T, 1 - m_{uv}^T - m_{kv}^T) &< \min(m_{uv}^T, 1 - m_{uv}^T) = m_{uv}^T \end{aligned}$$

Trivially,  $m_{uv}^T + m_{kv}^T < m_{uv}^T$  leads to a contradiction since  $m_{kv}^T > 0$ . Thus, the only possibility is

$$\begin{aligned} 1 - m_{uv}^T - m_{kv}^T < m_{uv}^T &\iff 1 - m_{kv}^T < 2m_{uv}^T \\ &\iff 1 - m_{kv}^T = m_{uv}^T + \sum_{\substack{i \in \mathcal{N}_v \\ i \neq k, i \neq u}} m_{iv}^T < 2m_{uv}^T \\ &\iff \sum_{\substack{i \in \mathcal{N}_v \\ i \neq k, i \neq u}} m_{iv}^T < m_{uv}^T \end{aligned}$$

which is also a contradiction since by assumption  $m_{uv}^T = \min_{i \in \mathcal{N}_v} m_{iv}^T$ . Now we are able to show that the cost of  $T'$  is lower than the one of  $T$

$$\begin{aligned} \text{CST}(T) - \text{CST}(T') &= c_{uv} (m_{uv}^T(1 - m_{uv}^T))^\alpha + c_{kv} (m_{kv}^T(1 - m_{kv}^T))^\alpha \\ &\quad - c_{uv} ((m_{uv}^T + m_{kv}^T)(1 - m_{uv}^T - m_{kv}^T))^\alpha - c_{ku} (m_{kv}^T(1 - m_{kv}^T))^\alpha \\ &= c_{uv} \left( (m_{uv}^T(1 - m_{uv}^T))^\alpha - ((m_{uv}^T + m_{kv}^T)(1 - m_{uv}^T - m_{kv}^T))^\alpha \right) + (c_{kv} - c_{ku}) (m_{kv}^T(1 - m_{kv}^T))^\alpha \\ &= (m_{uv}^T(1 - m_{uv}^T))^\alpha \left( c_{uv} - c_{uv} \underbrace{\left( \frac{(m_{uv}^T + m_{kv}^T)(1 - m_{uv}^T - m_{kv}^T)}{m_{uv}^T(1 - m_{uv}^T)} \right)^\alpha}_{>1} + (c_{kv} - c_{ku}) \underbrace{\left( \frac{m_{kv}^T(1 - m_{kv}^T)}{m_{uv}^T(1 - m_{uv}^T)} \right)^\alpha}_{\geq 1} \right). \end{aligned}$$

We can differentiate two cases. If  $m_{kv}^T(1 - m_{kv}^T) = m_{uv}^T(1 - m_{uv}^T)$  then

$$\frac{\text{CST}(T) - \text{CST}(T')}{(m_{uv}^T(1 - m_{uv}^T))^\alpha} \xrightarrow{\alpha \rightarrow -\infty} (c_{uv} + c_{kv} - c_{ku}) > 0, \quad (33)$$

where we have used the strict triangle inequality. Otherwise, the limit tends to

$$\frac{\text{CST}(T) - \text{CST}(T')}{(m_{uv}^T(1 - m_{uv}^T))^\alpha} \xrightarrow{\alpha \rightarrow -\infty} c_{uv} > 0.$$

Hence, for sufficiently negative values of  $\alpha$ , the difference  $\text{CST}(T) - \text{CST}(T')$  will be positive. By repeating this process, we can continue reducing the degree of nodes with degree higher than 2 until all nodes have degree at most 2. This process will eventually lead to the formation of a Hamiltonian path with a lower cost than the original tree  $T$ .  $\square$

**Remark E.2.** Due to equation (33), Lemma E.3 required the triangle inequality to hold strictly. However, the strict triangle inequality is not an indispensable for the validity of Lemma E.3. Corollary E.1 demonstrates that, when the nodes of the graph are embedded



in a geodesic metric space (e.g. Euclidean space), the strict triangle inequality becomes unnecessary. This result extends also to the BCST problem.

Nevertheless, the scenario portrayed in Figure 24 serves as an example where the strict triangle inequality is not satisfied, leading to a non-optimal Hamiltonian path. This illustrates that while the strict triangle inequality may be dispensable under certain conditions, there are instances of arbitrary graphs, as demonstrated in the figure, where it cannot be abandoned.

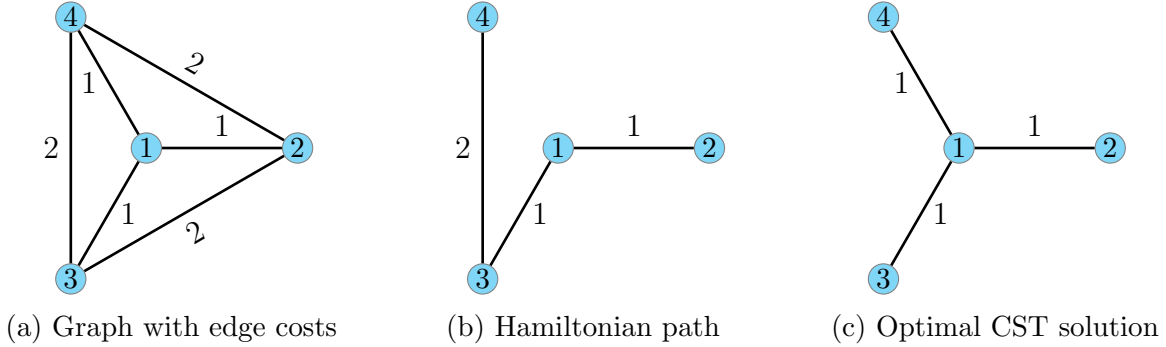


Figure 24: **Necessity of the Triangle Inequality for Path Graph to be (B)CST Optimum as  $\alpha \rightarrow -\infty$ .** 24a) Graph with edge costs depicted, where the triangle inequality is not strictly satisfied. 24b) Optimal hamiltonian path with CST cost equal to  $\frac{3^\alpha \cdot 3 + 4^\alpha}{16^\alpha}$ . 24c) Optimal CST with cost equal to  $\frac{3 \cdot 3^\alpha}{16^\alpha}$ . Thus, if the triangle inequality does not strictly hold, the Hamiltonian path will not necessarily be optimal even for sufficiently negative  $\alpha$  values

**Corollary E.1.** Consider the BCST and CST problem where the nodes are embedded in a geodesic metric space. As  $\alpha$  tends to negative infinity ( $\alpha \rightarrow -\infty$ ) there exists a Hamiltonian path  $T_*$  with a lower CST/BCST cost than any other tree  $T$  that is not a path.

*Proof.* The reasoning aligns with the exposition in Lemma E.3, proving that for any node  $v$  with degree exceeding 2, we can always decrease its degree such that the CST/BCST cost of  $T$  decreases. Through the iterative application of this process, the degrees of all nodes are systematically decreased, ultimately converging to a state where each node has at most a degree of 2, thereby resulting in the formation of a path.

Consider a node  $v$  with a degree higher than 2. To apply the logic presented in Theorem E.3, it is essential to ensure the ability to select two neighbors such that the triangle inequality holds strictly. Let  $u$ ,  $k$ , and  $\ell$  represent three distinct neighbors of  $v$ , and assume that the triangle inequality is an equality between each pair of neighbors and the node  $v$ . In other words, we have

$$c_{uv} + c_{vk} = c_{uk}, \tag{34}$$

$$c_{uv} + c_{v\ell} = c_{u\ell}, \tag{35}$$

$$c_{\ell v} + c_{kv} = c_{\ell k}. \tag{36}$$

Starting with (34), we conclude that  $v$  lies on the geodesic path between  $u$  and  $k$ . Similarly, from (35) and (36), we deduce that  $v$  is positioned between  $u$  and  $\ell$  and also between  $\ell$  and

$k$ . Consequently,  $\ell$  is established to be between  $u$  and  $k$ , as it is situated between  $u$  and  $v$  and also between  $k$  and  $v$ .

As a result, we can infer that  $c_{\ell k} + c_{\ell u} = c_{uk}$ . Summing (34) and (35) and utilizing the derived equality  $c_{\ell k} + c_{\ell u} = c_{uk}$  (owing to the position of  $\ell$  between  $u$  and  $k$ ), we obtain:

$$c_{uv} + c_{vk} + c_{uv} + c_{v\ell} = c_{ul} + c_{uk} \rightarrow 2c_{uv} + c_{\ell k} = c_{\ell k} \rightarrow c_{uv} = 0$$

The deduction that  $c_{uv} = 0$  implies that nodes  $u$  and  $v$  occupy the same position and can effectively be considered as a single node. Consequently, we can systematically remove neighbors of  $v$  by repeating this process until  $v$  attains a degree of 2.

Alternatively, if the triangle inequality must hold strictly for a pair of nodes  $u, k$ , and node  $v$ , we can, w.l.o.g., assume that node  $u$  is chosen such that  $m_{uv} = \min_{i \in \mathcal{N}_v} m_{iv}$ . In this scenario, applying the same reasoning as the one presented in Lemma E.3, we demonstrate that the modified topology  $T'$  –where node  $k$  is connected to node  $u$  instead of node  $v$ – results in a lower BCST/CST cost. □

Lemma E.3 resembles Lemma E.1 in the sense that both lemmas require the satisfaction of a weighted triangle inequality. The following corollary reformulates Lemma E.3, mirroring the structure found in Lemma E.1, and accentuates the correlation between the weighted triangle inequality and the number of terminals denoted by  $N$ .

**Corollary E.2** (Theorem 2.2). Given a complete graph  $G$  with  $N$  nodes, let  $c_{ij}$  be the edge-costs of any pair of nodes  $(i, j)$ . If there exists

$$t \leq \min_{\substack{1 \leq s \leq N-3 \\ 1 \leq \ell \leq \min(s, (N-s)/2-1)}} \frac{(\ell(N-\ell))^\alpha - ((\ell+s)(N-\ell-s))^\alpha}{(s(N-s))^\alpha}$$

such that

$$c_{kv} + tc_{uv} \geq c_{ku}$$

for all triangles in the graph, then there exists an optimum CST evaluated at  $\alpha$  which is a Hamiltonian path.

*Proof.* Lemma E.3 holds if

$$\begin{aligned} & \left( c_{uv} - c_{uv} \left( \frac{(m_{uv}^T + m_{kv}^T)(1 - m_{uv}^T - m_{kv}^T)}{m_{uv}^T(1 - m_{uv}^T)} \right)^\alpha + (c_{kv} - c_{ku}) \left( \frac{m_{kv}^T(1 - m_{kv}^T)}{m_{uv}^T(1 - m_{uv}^T)} \right)^\alpha \right) > 0 \\ \iff & c_{uv} \left( \frac{(m_{uv}^T(1 - m_{uv}^T))^\alpha - ((m_{uv}^T + m_{kv}^T)(1 - m_{uv}^T - m_{kv}^T))^\alpha}{(m_{kv}^T(1 - m_{kv}^T))^\alpha} \right) + c_{kv} > c_{ku} \end{aligned} \quad (37)$$

where it is assumed that  $m_{uv} = \min_{i \in \mathcal{N}_v} m_{iv}$ . Thus, let  $m_{uv} = \frac{\ell}{N}$  and  $m_{kv} = \frac{s}{N}$ . Substituting these values into (37), we derive the following inequality

$$c_{kv} + \frac{(\ell(N-\ell))^\alpha - ((\ell+s)(N-\ell-s))^\alpha}{(s(N-s))^\alpha} c_{uv} \geq c_{ku}.$$

Note that  $\ell \leq \min(s, (N - s)/2 - 1)$  since  $m_{uv} \min_{i \in \mathcal{N}_v} m_{iv}$ . Thus,  $m_{uv} = \frac{\ell}{N} \leq m_{kv} = \frac{s}{N}$ . On the other hand, since in the proof of Lemma E.3,  $v$  has at least three neighbors ( $u$ ,  $k$  and say  $p$ ), the  $N - s$  nodes not lying in the side corresponding to  $k$  of edge  $(k, v)$  must be distributed in the side of  $v$ . At exception of node  $v$ , one part of the remaining  $N - s - 1$  nodes will be in the side corresponding to  $u$  of edge  $(u, v)$  and the other part in the side corresponding to  $p$  of edge  $(p, v)$ . Since  $m_{uv}$  must be minimum, the side corresponding to  $u$  of edge  $(u, v)$  can at most be equal to  $\frac{N-s-1}{2N}$ .

Notice also that  $1 \leq s \leq N - 3$ , since  $v$  has degree at least three and therefore there are at least three nodes ( $v$ ,  $u$ ,  $p$ ) lying in the same side of  $v$  with respect to edge  $(k, v)$ . Thus,  $m_{vk}^T = (1 - m_{kv}^T) = 1 - \frac{s}{N} \geq \frac{3}{N}$ , or equivalently  $s \leq N - 3$ .  $\square$

## F Exploring the Number of Derivable Topologies from CST and BCST Topologies

### F.1 Number of BCST Topologies Derivable from a CST Topology

In this section, we explicitly determine the number of topologies of the BCST problem that can be derived from a single CST topology.

To derive a full tree topology  $T_{\text{BCST}}$  from a CST topology  $T_{\text{CST}}$  with  $N$  terminals, we need to add  $N - 2$  SP. In particular, for each terminal node,  $v$ , with degree  $d_v \geq 2$ , we need to spawn  $d_v - 1$  SP. Since for  $k$  terminal nodes, there exist a total  $(2k - 5)!!$  of full tree topologies [32], there are  $(2(d_v + 1) - 5)!! = (2d_v - 3)!!$  ways to connect the added SP to the neighbors of  $v$  and  $v$  itself. Thus the total number of full tree topologies is equal to the number of possible combinations of subtopologies engendered per terminal neighborhood for terminals with degree higher than 2. Formally, this number is equal

$$\prod_{v : d_v \geq 2} (2d_v - 3)!! \tag{38}$$

Note that, on the one hand, if all nodes have degree lower or equal than 2, i.e. the tree is a path, then a single full tree topology can be derived. On the other hand, if the original  $T_{\text{CST}}$  is a star graph, then there is a single graph with degree higher than 2, which is equal to  $N - 1$ . Thus, the total number of topologies derived from it is equal to  $(2N - 5)!!$ . This is the total number of possible full tree topologies, hence a star graph can generate any full tree topology. In general, the higher the degree of the nodes in  $T_{\text{CST}}$ , the higher the number of derivable full tree topologies.

### F.2 Number of CST Topologies Derivable from a BCST Topology

In this case, we need to collapse each SP to a terminal. The collapse process can be carried out sequentially, where each SP is collapsed to one of its neighboring nodes, until no SP remain. Naively, we might assume that there are  $3^{N-2}$  possible topologies, given that each SP has 3 neighbors available for collapse and there are  $N - 2$  SPs. However, this is not the case because some combinations may result in non-valid topologies. For instance, if all SPs

collapse with neighbors that are also SP, none of the SP will be collapsed with a terminal node.

Before providing the formula for the number of CST topologies that can be derived from a full tree topology, let's introduce the All Minors Matrix Tree Theorem [9], which is necessary to derive the formula. The Matrix Tree Theorem [38, 20] states that the number of spanning trees of a given graph  $G = (V, E)$  can be calculated from the determinant of a submatrix of its Laplacian. Recall that the Laplacian matrix of a graph  $L$  is given by

$$L = D - A, \quad \text{where } A_{ij} = \begin{cases} 1 & \text{if } (i, j) \in E \\ 0 & \text{otherwise} \end{cases} \quad \& \quad D_{ij} = \begin{cases} \sum_k A_{ik} & \text{if } i = j \\ 0 & \text{otherwise} \end{cases}.$$

In other words,  $A$  is the adjacency matrix of the graph, and  $D$  is the diagonal degree matrix. We state a simplified version of the theorem without providing a proof.

**Theorem F.1.** Given a graph  $G = (V, E)$  and a subset  $U$  of nodes in  $G$ , let  $W = V \setminus U$ . We define  $L_{W,W}$  as the submatrix of the Laplacian matrix of  $G$ , which includes the rows and columns indexed by the nodes in  $W$ . In this context, the determinant of  $L_{W,W}$ , denoted as  $\det(L_{W,W})$ , provides a count of the number of spanning forests of  $G$  that consist of  $|U|$  disjoint trees, with the nodes in  $U$  being disconnected across these trees.

*Proof.* See Chaiken [9]. □

Now we are ready to present the main result of this subsection. Consider a full tree topology  $T_{\text{BCST}}$ . The number of topologies for the CST problem that can be derived from  $T_{\text{BCST}}$  is given by

$$\det(L_{SP_s, SP_s}), \tag{39}$$

where  $L_{SP_s, SP_s}$  represents the submatrix of the Laplacian matrix  $L$  of  $T_{\text{BCST}}$ . This submatrix is formed by selecting the rows and columns associated with the SPs. By virtue of Theorem F.1, equation (39) counts the number of spanning forests of the  $T_{\text{BCST}}$  which disconnect the terminal nodes. To demonstrate that this count of forests coincides with the number of topologies that can be derived from the full tree topology  $T_{\text{BCST}}$ , we will establish a bijection.

Indeed, if we have a forest that disconnects all the terminals, each SP within the forest must belong to a component with a single terminal. In this scenario, we can unambiguously collapse each SP to its corresponding terminal. Once we have collapsed the SP, we still need to reconnect the terminals between them to form a valid CST topology. Now, notice that in the original full tree topology  $T_{\text{BCST}}$ , each terminal is uniquely adjacent to a SP. We can connect the terminals between them based on the collapse process of the SP. Specifically, a terminal  $v_t$  is connected to another terminal  $u_t$  if the neighboring SP of  $v_t$  in the original  $T_{\text{BCST}}$  has been collapsed to  $u_t$ . Similarly, we can reverse these steps to map a CST topology back to a unique forest that disconnects the terminals. Figure 25 illustrates the individual steps of this bijection using two examples. We have proven the following theorem

**Theorem F.2.** Let  $T_{\text{BCST}}$  be a full tree topology with  $N$  terminals. Consider the Laplacian matrix  $L$  of  $T_{\text{BCST}}$ . The number of CST topologies that can be derived from  $T_{\text{BCST}}$  is equal to the determinant of  $L_{SP_s, SP_s}$ , which is the submatrix of the Laplacian obtained by selecting the rows and columns indexed by the SPs. Hence, the number of CST topologies derived from  $T_{\text{BCST}}$  can be calculated as  $\det L_{SP_s, SP_s}$ .

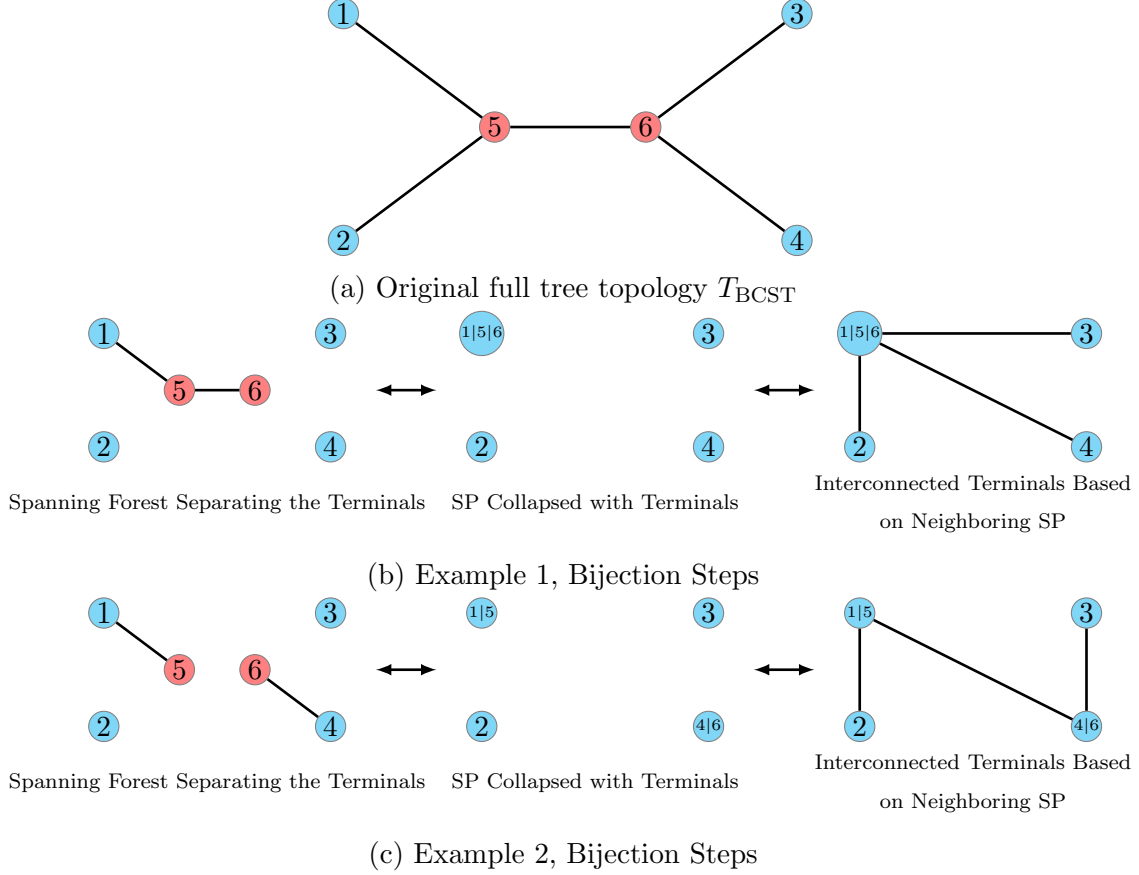


Figure 25: **Bijection Between Derivable CST Topologies from a Full Tree Topology and Their Terminal-Separating Forests.** Figures 25b) and 25c) illustrate two examples of the relationship between a spanning forest and a derived CST topology of a full tree topology depicted in Figure 25a.

## G Branching Angles at the Steiner Points in the BCST Problem

In this section, we formulate the branching angles in terms of the centralities of the edges for a given topology of the BCST problem. As stated in Section 5.1, it is sufficient to study the geometric optimization of 3 nodes connected by a single SP, with minimization objective given by

$$C(b) = \zeta_0 \|b - a_0\| + \zeta_1 \|b - a_1\| + \zeta_2 \|b - a_2\|. \quad (40)$$

Recall that node  $b$  represents the Steiner point whose coordinates need to be optimized, nodes  $\{a_i\}$  are the terminals with fixed positions and  $\zeta_i := m_{ba_i}(1 - m_{ba_i})$  are the centralities of the edges  $(b, a_i)$  (see Figure 26).

We will reproduce the arguments exposed for the BOT problem in Bernot et al. [4] and Lippmann et al. [22] to determine the angles  $\theta_1$  and  $\theta_2$ . We will differentiate two cases: when the SP does not coincide with any other terminal node; and when  $b$  collapses with one of the terminals.

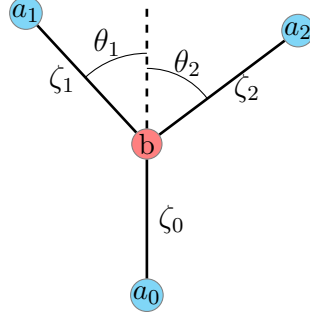


Figure 26: **Branching Angles at Steiner Point.** The symbols  $\zeta_i$  represent the normalized centralities of the edges, that is  $\zeta_i := m_{ba_i}(1 - m_{ba_i})$ .

## G.1 Steiner Point $b$ Does Not Collapse with a Terminal

In this case, the function is differentiable with respect to  $b$ , and therefore we just need to see where the gradient of equation (40) is equal to zero. The formula for the gradient is as follows

$$\nabla_b C(b) = \zeta_0^\alpha n_0 + \zeta_1^\alpha n_1 + \zeta_2^\alpha n_2, \quad (41)$$

where  $n_i = \frac{b-a_i}{\|b-a_i\|}$ . By applying the dot product to  $\nabla_b C(b)$  with each  $n_i$  and setting it equal to zero, we derive the following equalities:

$$\begin{aligned} \langle \nabla_b C(b), n_0 \rangle = 0 &\rightarrow \zeta_0^\alpha + \zeta_1^\alpha \underbrace{\langle n_1, n_0 \rangle}_{-\cos(\theta_1)} + \zeta_2^\alpha \underbrace{\langle n_2, n_0 \rangle}_{-\cos(\theta_2)} = 0 \\ \langle \nabla_b C(b), n_1 \rangle = 0 &\rightarrow \zeta_0^\alpha \underbrace{\langle n_0, n_1 \rangle}_{-\cos(\theta_1)} + \zeta_1^\alpha + \zeta_2^\alpha \langle n_2, n_1 \rangle = 0 \\ \langle \nabla_b C(b), n_2 \rangle = 0 &\rightarrow \zeta_0^\alpha \underbrace{\langle n_0, n_2 \rangle}_{-\cos(\theta_2)} + \zeta_1^\alpha \langle n_1, n_2 \rangle + \zeta_2^\alpha = 0 \end{aligned}$$

Solving the linear system we obtain that the angles satisfy

$$\begin{aligned} \cos(\theta_1) &= \frac{\zeta_0^{2\alpha} + \zeta_1^{2\alpha} - \zeta_2^{2\alpha}}{2\zeta_0^\alpha \cdot \zeta_1^\alpha}, \\ \cos(\theta_2) &= \frac{\zeta_0^{2\alpha} + \zeta_2^{2\alpha} - \zeta_1^{2\alpha}}{2\zeta_0^\alpha \cdot \zeta_2^\alpha}, \\ \cos(\theta_1 + \theta_2) &= \frac{\zeta_0^{2\alpha} - \zeta_1^{2\alpha} - \zeta_2^{2\alpha}}{2\zeta_1^\alpha \cdot \zeta_2^\alpha}. \end{aligned} \quad (42)$$

## G.2 Steiner Point $b$ Collapses with a Terminal

In this case, in order to determine the optimality angles, we will use the subdifferential argument applied in Lippmann et al. [22]. W.l.o.g. we will assume that  $b$  collapses with terminal  $a_0$ .

The subdifferential of a convex function  $h : \mathbb{R}^n \rightarrow \mathbb{R}$  at  $x$  is defined as the following set of vectors

$$\partial g(x) := \{v : h(z) \geq h(x) + \langle v, z - x \rangle, \forall z \in \mathbb{R}^n\}.$$

In other words,  $\partial g(x)$  comprises all vectors  $v$  such that the line passing through  $h(x)$  in the direction of  $v$  lies below the function  $h$  at all points. Each of these vectors is called a subgradient of  $h$  at  $x$ . When a function is differentiable at  $x$ , the subdifferential only contains the gradient of the function at  $x$ .

Fermat rule states that a convex function attains its minimum at  $x$  if and only if  $0 \in \partial g(x)$ . Furthermore, the subdifferential of two convex functions is equal to the union of the pairwise sums of their subgradients. In other words, for  $g(x) = g_1(x) + g_2(x)$  then

$$\partial g(x) = \{v_1 + v_2 : v_1 \in \partial g_1(x), v_2 \in \partial g_2(x)\}. \quad (43)$$

We can apply Fermat's rule to determine when the minimum is attained at  $b = a_0$ . For the function  $g(x) = w \cdot \|x - a\|$ , the subdifferential is given by

$$\partial g(x) = \begin{cases} \{v : \|v\| \leq w\}, & \text{if } x = a \\ \left\{ w \frac{x-a}{\|x-a\|} \right\}, & \text{otherwise} \end{cases}.$$

Thus, applying equation (43), the subdifferential of  $C(b)$  at  $b = a_0$  is given by

$$\partial C(a_0) = \left\{ v + \zeta_1^\alpha \frac{b - a_1}{\|b - a_1\|} + \zeta_2^\alpha \frac{b - a_2}{\|b - a_2\|} : \|v\| \leq \zeta_0^\alpha \right\}.$$

In order for  $b$  to be optimal at  $a_0$ , zero has to belong to  $\partial C(a_0)$ , which is true if and only if

$$\begin{aligned} & \left\| \zeta_1^\alpha \frac{b - a_1}{\|b - a_1\|} + \zeta_2^\alpha \frac{b - a_2}{\|b - a_2\|} \right\| \leq \zeta_0^\alpha \\ \iff & \left\| \frac{b - a_1}{\|b - a_1\|} + \frac{b - a_2}{\|b - a_2\|} \right\|^2 = \zeta_1^{2\alpha} + \zeta_2^{2\alpha} + 2\zeta_1^\alpha \zeta_2^\alpha \cos(\gamma) \leq \zeta_0^{2\alpha} \end{aligned} \quad (44)$$

where  $\gamma$  is the angle of the terminal triangle at  $a_0$ , that is  $\gamma := \angle a_1 a_0 a_2$ . Isolating  $\gamma$ , we get

$$\gamma \geq \arccos \left( \frac{\zeta_0^{2\alpha} - \zeta_1^{2\alpha} - \zeta_2^{2\alpha}}{2\zeta_1^\alpha \cdot \zeta_2^\alpha} \right) = \theta_1 + \theta_2. \quad (45)$$

Thus  $b$  will collapse to  $a_0$  if the angle  $\angle a_1 a_0 a_2$  is greater than the optimal angle given by (42). In such cases, the resulting branching is referred to as a  $V$ -branching.

**Remark G.1.** It is worth noting that the reasoning presented in this section remains independent of the weighting factors, which, in our case, were set equal to the normalized edge centralities powered to  $\alpha$ . As a result, this finding holds true for any weights and can be used to determine an arbitrary weighted geometric median of three points. Furthermore, we emphasize that the position of the SP,  $b$ , depends exclusively on the angles and weighting factors and not on the distances between the terminal nodes.

## H Infeasibility of Degree-4 Steiner Points in the Plane for $\alpha = 1$

As stated in Section 5.2, the optimal position of the SPs is continuously dependent on the terminal positions and solely relies on the branching angles, as shown in Section 5.1.

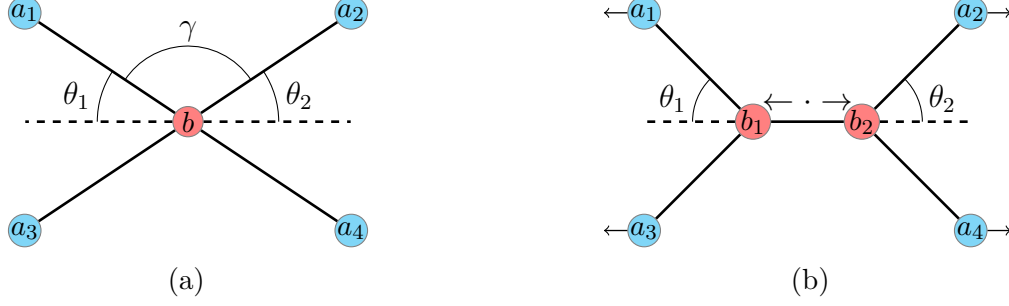


Figure 27: **Splitting Collapsed SP While Preserving Optimal Angles.** 27a) illustrates the collapsed solution of a 4-terminal configuration. 27b) demonstrates that it is possible to move jointly the terminal points  $\{a_1, a_3\}$  in a specific but opposite direction to the one of the terminals  $\{a_2, a_4\}$ , resulting in the splitting of the collapsed SP  $b$  into two distinct SPs,  $b_1$  and  $b_2$ . Remarkably, this split can be executed while preserving the angles  $\theta_1$  and  $\theta_2$ . Importantly, these angles must correspond to the optimal angles given by (42).

Consequently, assuming that there exists a configuration such that the SPs collapse, it is possible to find terminal positions that lead to an unstable collapse of the SPs. Here, instability refers to a configuration where an infinitesimal translation of the terminals results in the splitting of the SPs. This scenario is depicted in Figure 27. In such cases, the angles realized by the terminals and the SPs will reach the upper bounds specified by (45). Therefore, the angles depicted in Figure 27a fulfill the condition

$$\gamma = \pi - \theta_1 - \theta_2, \quad (46)$$

where the angles satisfy

$$\cos(\gamma) = \frac{F(m_{a_1b} + m_{a_2b})^{2\alpha} - F(m_{a_1b})^{2\alpha} - F(m_{a_2b})^{2\alpha}}{2F(m_{a_1b})^\alpha F(m_{a_2b})^\alpha} \quad (47)$$

$$\cos(\theta_1) = \frac{F(m_{a_3b} + m_{a_1b})^{2\alpha} + F(m_{a_1b})^{2\alpha} - F(m_{a_3b})^{2\alpha}}{2F(m_{a_3b} + m_{a_1b})^\alpha F(m_{a_1b})^\alpha} \quad (48)$$

$$\cos(\theta_2) = \frac{F(m_{a_2b} + m_{a_4b})^{2\alpha} + F(m_{a_2b})^{2\alpha} - F(m_{a_4b})^{2\alpha}}{2F(m_{a_2b} + m_{a_4b})^\alpha F(m_{a_2b})^\alpha} \quad (49)$$

We can manipulate (46) in the following way

$$\begin{aligned} \gamma = \pi - \theta_1 - \theta_2 &\iff \cos(\gamma - \pi) = \cos(-\theta_1 - \theta_2) \\ &\iff -\cos(\gamma) = \cos(\theta_1 + \theta_2) \\ &\iff \underbrace{-\cos(\gamma)}_* = \cos(\theta_1) \cos(\theta_2) - \sqrt{(1 - \cos(\theta_1)^2)(1 - \cos(\theta_2)^2)} \end{aligned} \quad (50)$$

where in (\*) we have used the fact that

$$\cos(x + y) = \cos(x) \cos(y) - \sin(x) \sin(y) = \cos(x) \cos(y) - \sqrt{(1 - \cos(x)^2)(1 - \cos(y)^2)}.$$

If we square both sides of 50 and equate to 0 we obtain.

$$(\cos(\gamma) + \cos(\theta_1) \cos(\theta_2))^2 - (1 - \cos(\theta_1)^2)(1 - \cos(\theta_2)^2) = 0. \quad (51)$$



Equation (51) depends on the variables  $m_{a_1b}$ ,  $m_{a_2b}$ ,  $m_{a_3b}$ , and  $\alpha^{12}$ . Equation (51) is generally too complex to be solved analytically. However, with the help of the Mathematica software [16], we have determined that for  $\alpha = 1$ , the equality does not hold within the constraints of the problem, namely

$$\sum_{i=1}^4 m_{a_i,b} = 1 \quad \text{and} \quad 0 < m_{a_i,b} < 1, \quad \forall i.$$

To simplify the notation, let's denote  $m_{a_i b}$  as  $m_i$ . For  $\alpha = 1$ , when we expand equation (51), we find that the numerator of the formula becomes a fourth-degree polynomial with respect to  $m_1$ . The four roots,  $\{m_1^{(j)}\}$  of the polynomial are

$$\begin{aligned} m_1^{(1)} &= \frac{1}{2} \left( 1 - m_2 - m_3 - \sqrt{(-1 + m_2 + m_3)^2 - \frac{4}{3} \left( -2m_2 - 2m_3 + 3m_2m_3 - \sqrt{m_2^2 - m_2m_3 + m_3^2} \right)} \right) \\ m_1^{(2)} &= \frac{1}{2} \left( 1 - m_2 - m_3 + \sqrt{(-1 + m_2 + m_3)^2 - \frac{4}{3} \left( -2m_2 - 2m_3 + 3m_2m_3 - 2\sqrt{m_2^2 - m_2m_3 + m_3^2} \right)} \right) \\ m_1^{(3)} &= \frac{1}{2} \left( 1 - m_2 - m_3 - \sqrt{(-1 + m_2 + m_3)^2 - \frac{4}{3} \left( -2m_2 - 2m_3 + 3m_2m_3 + 2\sqrt{m_2^2 - m_2m_3 + m_3^2} \right)} \right) \\ m_1^{(4)} &= \frac{1}{2} \left( 1 - m_2 - m_3 + \sqrt{(-1 + m_2 + m_3)^2 - \frac{4}{3} \left( -2m_2 - 2m_3 + 3m_2m_3 + 2\sqrt{m_2^2 - m_2m_3 + m_3^2} \right)} \right) \end{aligned} \quad (52)$$

where we have highlighted the difference between the roots. We will show that

$$1 < m_1^{(4)} + m_2 + m_3 < m_1^{(2)} + m_2 + m_3 \quad \text{and} \quad m_1^{(1)} < m_1^{(3)} < 0,$$

which implies that the problem constraints are not satisfied, and therefore SPs of degree 4 are not possible.

**Claim 1:**  $1 < m_1^{(4)} + m_2 + m_3 \leq m_1^{(2)} + m_2 + m_3$ . From (52) it is clear that  $m_1^{(4)} \leq m_1^{(2)}$ . Thus, it is enough to prove the inequality for  $m_1^{(4)}$ :

$$\begin{aligned} m_1^{(4)} + m_2 + m_3 &= \frac{1}{2}(1 + m_2 + m_3) + \frac{1}{2} \sqrt{(-1 + m_2 + m_3)^2 - \frac{4}{3} \left( -2m_2 - 2m_3 + 3m_2m_3 - \sqrt{m_2^2 - m_2m_3 + m_3^2} \right)} \\ &= \frac{1}{2}(1 + m_2 + m_3) + \frac{1}{2} \sqrt{1 + \frac{2}{3}(m_2 + m_3) + (m_2 - m_3)^2 - \frac{8}{3} \sqrt{\underbrace{m_2^2 - m_2m_3 + m_3^2}_{< (m_2 + m_3)^2}}} \\ &> \frac{1}{2}(1 + m_2 + m_3) + \frac{1}{2} \sqrt{1 + \frac{2}{3}(m_2 + m_3) + (m_2 - m_3)^2 - \frac{8}{3}(m_2 + m_3)} \\ &= \frac{1}{2}(1 + m_2 + m_3) + \frac{1}{2} \sqrt{1 + (m_2 - m_3)^2 - \frac{1}{2}(m_2 + m_3)} \\ &> \frac{1}{2} \left( \underbrace{1 + m_2 + m_3}_{> 2} + \sqrt{1 - \frac{1}{2}(m_2 + m_3)} \right) > 1 \end{aligned}$$

For the last inequality, we have used the fact  $0 < m_2 + m_3 < 1$ , that the function  $g(x) = 1 + x + \sqrt{1 - x/2}$  is increasing in  $[0, 1]$  and that  $g(0) = 2$ .

<sup>12</sup>Since  $\sum_{i=1}^4 m_{a_i b} = 1$ ,  $m_{a_4 b}$  can be expressed as  $1 - m_{a_1 b} - m_{a_2 b} - m_{a_3 b}$ .

**Claim 2:**  $m_1^{(1)} \leq m_1^{(3)} < 0$ . From (52) it is clear that  $m_1^{(1)} \leq m_1^{(3)}$ . Thus, it is enough to prove the inequality for  $m_1^{(3)}$ .

$$\begin{aligned}
m_1^{(3)} < 0 \\
&\iff \frac{1}{2} \left( 1 - m_2 - m_3 - \sqrt{(-1 + m_2 + m_3)^2 - \frac{4}{3} \left( -2m_2 - 2m_3 + 3m_2m_3 + 2\sqrt{m_2^2 - m_2m_3 + m_3^2} \right)} \right) < 0 \\
&\iff \left( -2m_2 - 2m_3 + 3m_2m_3 + 2\sqrt{m_2^2 - m_2m_3 + m_3^2} \right) < 0
\end{aligned} \tag{53}$$

Thus, we need to focus on inequality 53. We will differentiate various cases:

- **If  $2/3 > m_3 \geq m_2$ :**

$$\begin{aligned}
&\left( -2m_2 - 2m_3 + 3m_2m_3 + 2\sqrt{m_2^2 - m_2m_3 + m_3^2} \right) \\
&= m_2 \underbrace{(2 - 3m_3)}_{>0 \iff 2/3 > m_3} + 2 \left( \underbrace{m_3 - \sqrt{m_2^2 - m_2m_3 + m_3^2}}_{\geq 0 \iff m_3 \geq m_2} \right) > 0
\end{aligned}$$

For the second term, we have used the fact that

$$\begin{aligned}
m_3 \geq \sqrt{m_2^2 - m_2m_3 + m_3^2} &\iff m_3^2 \geq m_2^2 - m_2m_3 + m_3^2 \iff m_2m_3 \geq m_2^2 \\
&\iff m_3 \geq m_2
\end{aligned}$$

- **If  $2/3 \geq m_3 \geq m_2$ :** Analogous to previous case due to the symmetry of  $m_2$  and  $m_3$  in (53)
- **If  $\max(m_2, m_3) \geq 2/3$ :** W.l.o.g we can assume  $m_2 \geq 2/3$  and  $m_3 < 1/3$  due to the symmetry between  $m_2$  and  $m_3$ . For this case, we will find the roots with respect to  $m_2$  of inequality (53) and see that the constraints on  $m_2$  do not hold. Indeed,

$$\begin{aligned}
-2m_2 - 2m_3 + 3m_2m_3 + 2\sqrt{m_2^2 - m_2m_3 + m_3^2} = 0 &\Rightarrow \\
(-2m_2 - 2m_3 + 3m_2m_3)^2 = 4(m_2^2 - m_2m_3 + m_3^2)
\end{aligned}$$

The roots  $\{m_2^{(j)}\}$  of the polynomial are

$$\begin{aligned}
m_2^{(1)} &= \frac{2m_3}{-5 + 6m_3 + \sqrt{3}\sqrt{7 - 12m_3 + 6m_3^2}} && \& \quad m_3 \neq \frac{2 - \sqrt{2}}{3}, \\
m_2^{(2)} &= \frac{2m_3}{-5 + 6m_3 - \sqrt{3}\sqrt{7 - 12m_3 + 6m_3^2}} && \& \quad m_3 \neq \frac{2 - \sqrt{2}}{3}, \\
m_2^{(3)} &= \frac{2(-3 + 2\sqrt{2})}{3(-2 + 3\sqrt{2})} && \& \quad m_3 = \frac{2 - \sqrt{2}}{3}.
\end{aligned} \tag{54}$$

The denominator of the root  $m_2^{(1)}$  has negative sign for  $0 < m_3 < 1$ , which leads to  $m_1^{(2)} < 0$ , contradicting the initial constraints. Trivially,  $m_2^{(3)}$  is also negative.

The denominator of the root  $m_2^{(2)}$  is negative for  $0 < m_3 < \frac{2-\sqrt{2}}{3}$ , resulting in  $m_2^{(2)} < 0$ . When  $\frac{2-\sqrt{2}}{3} < m_3 < 1/3$ , the denominator becomes positive but remains lower than  $2m_3$ , thus  $m_2^{(2)} > 1$ , which is also a contradiction.

We have ruled out all possible cases, thus we have proven the next theorem.

**Theorem H.1.** Let  $\alpha = 1$ . Given a set of terminals which lie in the plane, then the SPs of the optimal solution of the BCST problem will not contain SPs of degree 4 unless these collapse with a terminal.

## I Iteratively Reweighted Least Square for the Geometric Optimization of the Steiner Points

In this section we review briefly the iteratively reweighted least square (IRLS) algorithm proposed in [36]. This algorithm was initially developed for the geometric optimization of Steiner points (SPs) and later adapted in [22] for the branched optimal transport (BOT) problem. We will show that the same algorithm can be adapted for the BCST problem, since the algorithm is agnostic to the weighting factors multiplying the distances involved in the BOT and BCST objectives, as defined in equations (20) and (2), respectively.

Consider the following minimization problem for a fixed tree topology

$$\min_{X_B} C(X) = \min_{X_B} \sum_{(i,j) \in E} w_{ij} \|x_i - x_j\| \quad (55)$$

where  $w_{ij}$  are arbitrary weights,  $E$  is the set of edges of the tree,  $X_B = \{x_{N+1}, \dots, x_{2N-2}\}$  are the coordinates of the SPs, which need to be optimized, and  $X = \{x_1, \dots, x_{2N-2}\}$  is the set of all coordinates (terminals and SPs). Starting from arbitrary SPs coordinates, denoted as  $X^{(0)}$ , the algorithm iteratively solves the following linear system of equations.

$$x_i^{(k+1)} = \frac{\sum_{j:(i,j) \in E} w_{ij} \frac{x_j^{(k+1)}}{\|x_i^{(k)} - x_j^{(k)}\|}}{\sum_{j:(i,j) \in E} \frac{w_{ij}}{\|x_i^{(k)} - x_j^{(k)}\|}}, \quad \forall N+1 \leq i \leq 2N-2. \quad (56)$$

Note that only the coordinates corresponding to the SPs are updated. The coordinates of the terminals are kept fixed and set equal to their original coordinates.

We will show that in each iteration the cost of the objective function decreases, i.e.  $C(X^{(k+1)}) < C(X^{(k)})$ . As shown in [36], this implies that the  $\lim_{k \rightarrow \infty} X^{(k)} = \arg \min C(X)$ .

The algorithm can be considered an IRLS approach because it reinterprets the cost function as a quadratic function. Indeed,  $C(X)$  can be rewritten as

$$C(X) = \sum_{(i,j) \in E} w_{ij} \|x_i - x_j\| = \sum_{(i,j) \in E} \underbrace{\frac{w_{ij}}{\|x_i - x_j\|}}_{W_{ij}(X)} \|x_i - x_j\|^2 = \sum_{(i,j) \in E} W_{ij}(X) \|x_i - x_j\|^2$$

In concrete, the solution of the linear system (56) minimizes the following quadratic function

$$Q^{(k)}(X) = \sum_{(i,j) \in E} W_{ij}(X^{(k)}) \|x_i - x_j\|^2.$$

That is  $Q^{(k)}(X) \geq Q^{(k)}(X^{(k+1)}) \forall X$ . Moreover, note that  $C(X^{(k)}) = Q^{(k)}(X^{(k)})$ . Now we can show that the cost  $C$  decreases at each iteration:

$$\begin{aligned} C(X^{(k)}) &= Q^{(k)}(X^{(k)}) \geq Q^{(k)}(X^{(k+1)}) \\ &= \sum_{(i,j) \in E} w_{ij} \frac{\left( \left| x_i^{(k)} - x_j^{(k)} \right| + \left| x_i^{(k+1)} - x_j^{(k+1)} \right| - \left| x_i^{(k)} - x_j^{(k)} \right| \right)^2}{\left| x_i^{(k)} - x_j^{(k)} \right|} \\ &= C(X^{(k)}) + 2 \left( C(X^{(k+1)}) - C(X^{(k)}) \right) \\ &\quad + \sum_{(i,j) \in E} w_{ij} \frac{\left( \left| x_i^{(k+1)} - x_j^{(k+1)} \right| - \left| x_i^{(k)} - x_j^{(k)} \right| \right)^2}{\left| x_i^{(k)} - x_j^{(k)} \right|} \tag{57} \\ &\iff \\ C(X^{(k+1)}) &\leq C(X^{(k)}) - \underbrace{\sum_{(i,j) \in E} \frac{w_{ij}}{2} \frac{\left( \left| x_i^{(k+1)} - x_j^{(k+1)} \right| - \left| x_i^{(k)} - x_j^{(k)} \right| \right)^2}{\left| x_i^{(k)} - x_j^{(k)} \right|}}_{\geq 0} \\ &\leq C(X^{(k)}) \end{aligned}$$

## J Complexity mSTreg Heuristic

We will delve into the complexity of the two steps of our heuristic.

- **Complexity geometric optimization:** We approximate the optimal SPs coordinates using the IRLS approach presented in Section I. Each iteration of the IRLS requires  $\mathcal{O}(nd)$  operations, where  $n$  is the number of terminals and  $d$  is the data dimensionality. Within each iteration,  $d$  linear systems are solved. These can be solved in linear time and in parallel. The number of iterations needed for the IRLS to converge is not known a priori, however, Lippmann et al. [22] suggest that this number could scale on average like  $\mathcal{O}(\log(n))$ . Consequently, each geometric optimization step takes  $\mathcal{O}(\log(n)nd)$ .
- **Topology optimization step:** In the topology optimization step, we compute the minimum spanning tree (mST) over the terminals and SPs. Given a graph  $G = (V, E)$ , Kruskal's algorithm takes  $\mathcal{O}(|E| \log |V|)$  operations to compute the mST. In a complete graph, this becomes  $\mathcal{O}(n^2 \log(n))$ . However, in some situations, we may expedite the mST computation by computing the mST over a  $k$ -nearest neighbor (kNN) graph. Approximating a kNN graph with  $k$ -d trees can have a complexity of  $\mathcal{O}(dn \log(n)^2)$ . In this case, the number of edges in the graph would be  $|E| \approx kn$ . Hence, the overall mST complexity would be  $\mathcal{O}(dn \log(n)^2 + kn \log(n)) \approx \mathcal{O}(dn \log(n)^2)$ .

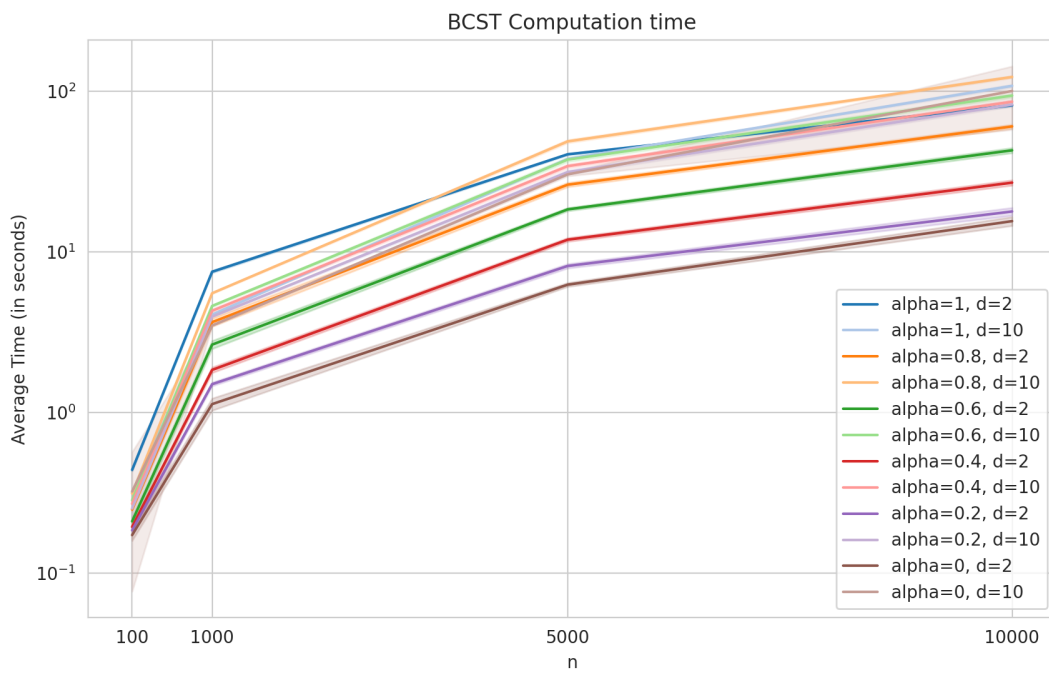


Figure 28: **BCST time complexity.** Computation time of 20 iterations of the mSTreg heuristic averaged over 5 distinct instances with different numbers of terminals  $n$ , data dimensionality  $d$ , and  $\alpha$  values.

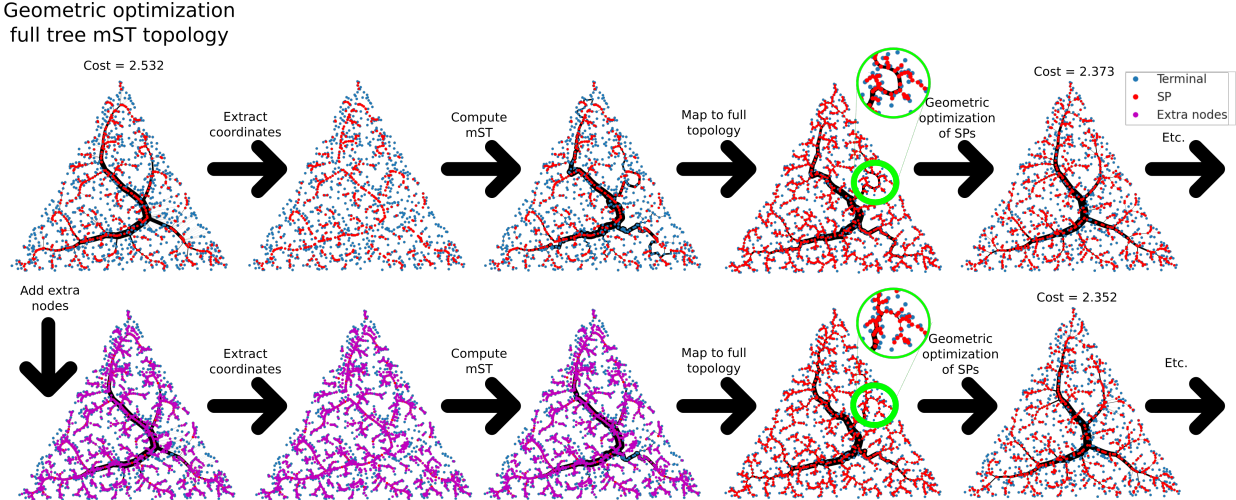


Figure 29: **mSTreg Heuristic with Additional Sampled Points.** Effect of adding extra points per edge (visualized in violet) in the mST computation step of the mSTreg heuristic. Top left: BCST solution obtained once the mST has been mapped to a full tree topology and its Steiner point coordinates have been optimized. Top row: Next steps of the mSTreg heuristic without adding any extra point. Bottom row: Next steps of the mSTreg heuristic once an extra point has been added at the middle of each edge (shown in violet). The addition of extra points may allow the mST to more reliably follow the edges of the geometry-optimized tree from previous step. We zoom in to highlight an improvement in the topology resulting from the addition of these extra points. In this particular case, the cost obtained with the inclusion of the extra nodes is lower than the cost without them.

Therefore, the heuristic’s per-iteration complexity is approximately  $\mathcal{O}(dn \log(n) + n^2 \log(n))$  or  $\mathcal{O}(dn \log(n)^2)$  if the mST is computed over a kNN graph. Throughout our experiments, a limit of 20 iterations was set, though practical convergence often demands fewer. In addition, we gauged the computational time of the heuristic by averaging its performance over 20 iterations across 5 distinct instances, varying  $n$ ,  $d$ , and  $\alpha$ . Data was generated by sampling  $n$  points from a  $d$ -dimensional unit cube. The performance times are presented in Figure 28. The heuristic was executed on an Intel Xeon Gold 6254 CPU @ 3.10GHz.

## K Effect of Additional Intermediate Points in the mSTreg Heuristic

In Section 6.2 we have described the mSTreg heuristic as a solution approach for the BCST problem. The algorithm can be summarized in two steps: 1) Optimization of the SPs coordinates given a fixed topology; 2) topology update by computing the mST over the terminals and SPs. The motivation for the topology update is that the optimal positions of the SPs may suggest a more desirable topology, since they may be biased to move closer to other nodes than the ones to which they are connected. Thus, we hope that the new topology, defined by the mST over the SPs and the terminals, interconnects such nodes.

However, there are instances where the SPs may not be sufficiently close to each other, causing the mST to fail in recovering the desired connections. The addition of intermediate nodes along the edges may address this problem, allowing the mST to more reliably follow the edges of the geometry-optimized tree from the previous step. An illustrative example highlighting the benefits of this approach can be seen in Figure 29. In general, we have seen that adding between 1 and 3 nodes per edge can often yield improvements. However, the impact on the main backbone is minimal. In Algorithm 1, the number of intermediate points that are added along an edge is regulated by the `sampling\_frequency` variable.

## L Strategies to Transform a Full Tree Topology into a CST Topology

When using the mSTreg heuristic described in Section 6.2 for solving the CST problem without branching points, we need to map from a full tree topology to a CST topology. As shown in Section F.2, this process is ambiguous and there may be an exponential number of derivable topologies with respect to the number of terminals. Hence to brute force the one which minimizes the CST cost is out of reach.

In this section, we describe some heuristic rules to transform a full tree topology into a CST topology. In order to transform a full tree topology into a CST topology, we collapse iteratively one SP at a time with one of its neighbors until there are no more SPs to collapse. The first factor to take into account is in which order the SPs are collapsed. We consider two strategies: 1) collapse the SP that is closest to a terminal (“Ordclosestterminal”) or 2) collapse the SP with the closest neighbor, i.e. the one that minimizes the distance to one of each neighbors independently of if it is a terminal or a SP (“Ordclosest”). In practice we did not see any big difference, though “Ordclosest” tends to be slightly better.

The second factor to take into account is to which neighbor should an SP collapse. We again compare two different heuristics: 1) collapsing the SP to the neighbor that minimally increases the CST cost (“greedy”); 2) collapsing the SP to the closest neighbor in terms of distance. We found empirically that the greedy approach yields significantly superior results.

Lastly, we conducted tests on updating the position of the collapsed SP. When a SP, denoted as  $b_1$ , is collapsed with a neighbor  $b_2$ , then the other neighbors of  $b_1$  become neighbors of  $b_2$ . We observed that updating the position of  $b_2$  to the weighted geometric median of its neighbors (including those inherited from  $b_1$ ) yielded improved results compared to not updating the coordinates of  $b_2$ . The coordinates of  $b_2$  were only updated when  $b_2$  was an SP. If  $b_2$  happened to be a terminal, its position was kept fixed. To denote if a strategy updated the position or not we will use the expression “update” and “no\_update” respectively.

To evaluate the effectiveness of the strategies, we conducted a series of experiments by sampling 200 problem instances for each  $N$  in the set  $\{5, 6, 7, 8, 9\}$ , where  $N$  represents the number of terminals. For each instance, we applied the mSTreg heuristic with different  $\alpha$  values and utilized the aforementioned strategies to transform a full tree topology into a CST topology. Figure 30 shows the mean ranking positions obtained by the different strategies, once all feasible solutions have been sorted. The results confirm the observations that we already pointed out. For all of our experiments we used the combination that produced the

best results i.e. “update”+“greedy”+“Ordclosest”.

## M Further Details on the Brute Force Experiment

In this section, we analyze the behavior of the mSTreg heuristic with respect to  $\alpha$ . To investigate this, we utilize the experiment described in Section 7, which compared the cost of the output tree generated by our heuristic with the optimal solution for different numbers of terminal nodes, denoted as  $N$ , while specifically examining the influence of  $\alpha$ .

For each  $N \in \{5, 6, 7, 8, 9\}$ , we sample 200 problem instances. We computed the optimal CST and BCST topologies of all problems via brute-force and with the mSTreg heuristic<sup>13</sup> for all  $\alpha \in \{0, 0.1, \dots, 0.9, 1\}$ . Figures 31 and 32 show the relative error and how the heuristic solution ranks, when the costs of all topologies are sorted. When solving the BCST, though there is not a clear trend, we can observe that for higher  $N$  the heuristic tends to perform worse for higher  $\alpha$  values, since on average the heuristic’s solution ranking is higher. When solving the CST problem this pattern can be more clearly seen.

## N Selection of $\alpha$

In this section, we present our practical insights into determining the optimal value for  $\alpha$ . It is crucial to emphasize that the choice of  $\alpha$  is task-dependent and influenced by the desired level of structure preservation. Nevertheless, we share the observations derived from our empirical experiences.

For simpler examples, as the ones illustrated in Section A, we have consistently found that  $\alpha$  values within the range of  $[0.7, 1]$  yield high stability while preserving the primary data structure. In general, an increase in  $\alpha$  correlates with a heightened inclination toward a star-shaped tree, in line with the limit case discussed in Appendix 2.1. As pointed in Remark 2.2, this pattern intensifies in higher dimensions, being noticeable with even modest  $\alpha$  values ( $\alpha \lesssim 1$ ), which results in a nearly star-shaped tree that compromises the preservation of data structure.

We refrained from further exploring the case of  $\alpha > 1$  due to both practical and theoretical observations pointing towards an excessively star-shaped tree. Indeed, as demonstrated in 2.1, when the number of terminals,  $N$ , is sufficiently large and  $\alpha > 1$ , the optimal solution results in a star-graph. The transition to a star-tree occurs quite early in this scenario. For instance, with moderate values of  $N$  and  $\alpha \gtrsim 1$  (e.g.,  $N = 1000$ ,  $\alpha = 1.13$ ), an optimal star-tree emerges. Refer to Figure 2 to observe how the value of  $\alpha$  at which the optimal solution becomes a star-tree approaches 1 as  $N$  increases.

In summary, our empirical experience suggests that intermediate  $\alpha$  values (around 0.5) effectively preserve the data structure while maintaining relative stability. This choice holds true for the applications highlighted in the thesis. We hope that by sharing our experiences, practitioners can better select an appropriate  $\alpha$  for their respective applications.

---

<sup>13</sup>As described in L, we can use different strategies to transform a full tree topology into a CST topology, when solving the CST problem with the mSTreg heuristic. We used the one that updates the position of SPs and collapses to the closest neighbor.



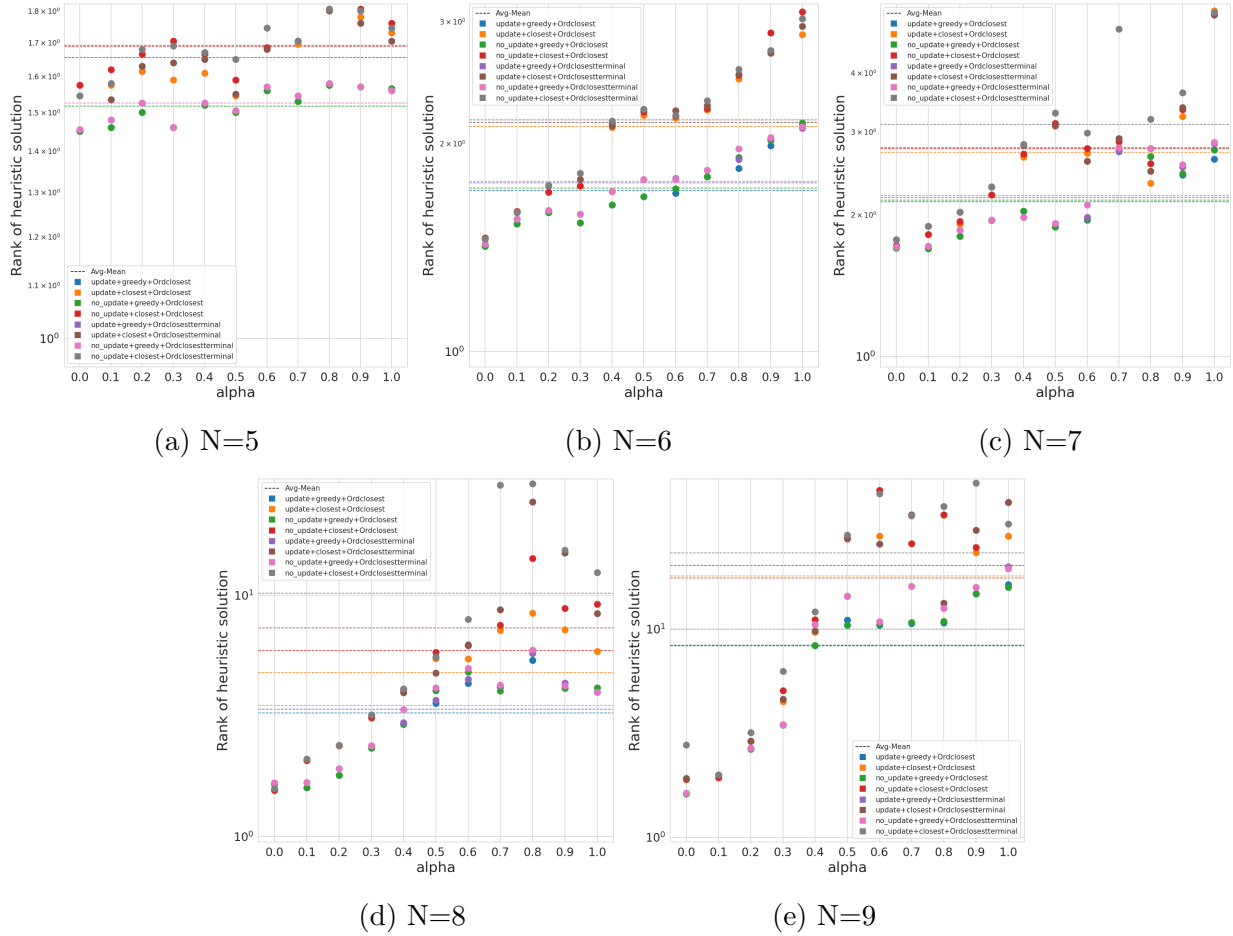
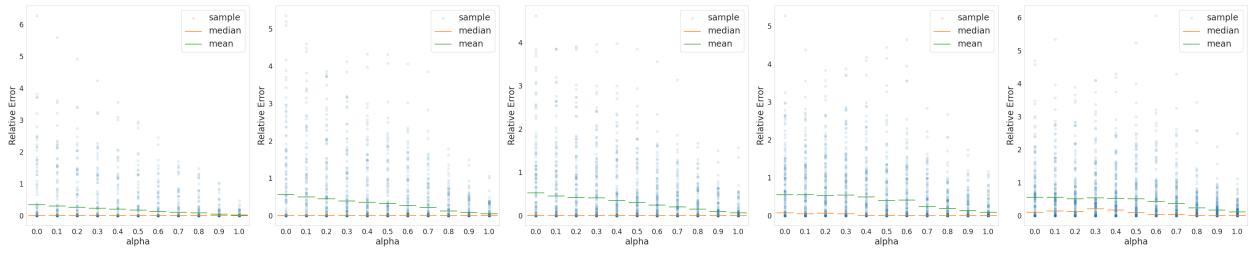
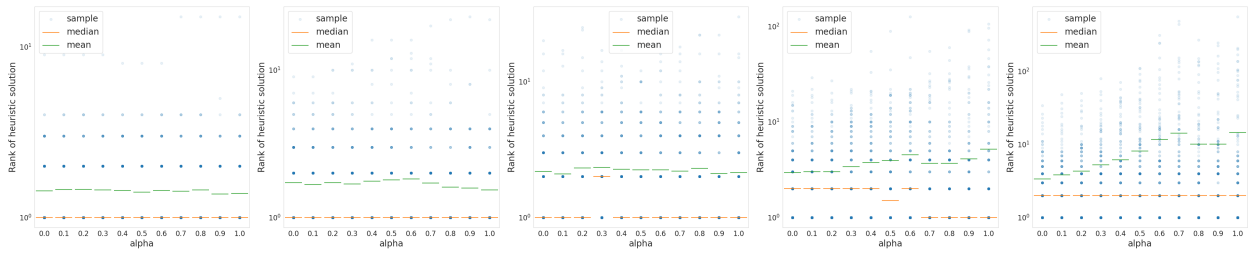


Figure 30: **Performance SPs Collapse Strategies.** Comparison of different collapse strategies to transform a full tree topology into a CST topology. See Section L for a short description of the strategies. We plot the average sorted position of the heuristic for different number of terminals,  $N$  and for different  $\alpha$  values. We observe that the strategy combinations including the “greedy” collapse approach have significantly better results. The combinations which update the position of the collapsed SP (“update”) perform slightly better than the ones that do not (“no\_update”). Analogously, the strategies that order the SPs based on the closeness to the neighbors (“Ordclosest”) is slightly better than “Ordclosestterminal”.



(a)  $N=5$       (b)  $N=6$       (c)  $N=7$       (d)  $N=8$       (e)  $N=9$

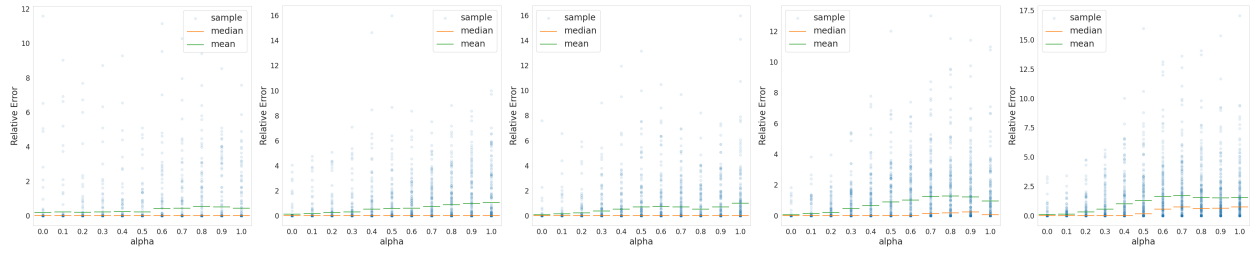
BCST relative error for different number of terminals  $N$



(f)  $N=5$       (g)  $N=6$       (h)  $N=7$       (i)  $N=8$       (j)  $N=9$

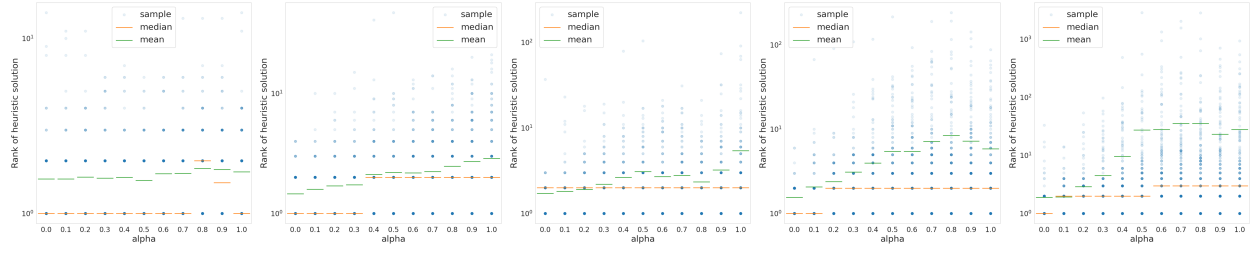
BCST rank of heuristic for different number of terminals  $N$

Figure 31: **BCST Bruteforce Benchmark with Respect to  $\alpha$** . Relative cost errors between the mSTreg heuristic and BCST optimal solutions; and sorted position of the heuristic tree for different number of terminals,  $N$ . For each  $N$  we uniformly sampled 200 different terminal configurations and we solved them for all  $\alpha \in \{0.0, 0.1, \dots, 1.0\}$ . Most runs ended up close to the global optimum. There is no clear pattern with respect to the performance of the heuristic with respect to the value of  $\alpha$ , though for higher number of terminals, it seems that the rank of our solution gets to be worse on average.



(a) N=5      (b) N=6      (c) N=7      (d) N=8      (e) N=9

CST relative error for different number of terminals N



(f) N=5      (g) N=6      (h) N=7      (i) N=8      (j) N=9

CST rank of heuristic for different number of terminals N

Figure 32: **CST Bruteforce Benchmark with Respect to  $\alpha$** . Relative cost errors between the mSTreg heuristic and optimal CST solutions; and sorted position of the heuristic tree for different number of terminals,  $N$ . For each  $N$  we uniformly sampled 200 different terminal configurations and we solved them for all  $\alpha \in \{0.0, 0.1, \dots, 1.0\}$ . Most runs ended up close to the global optimum. There is no clear pattern with respect to the performance of the heuristic with respect to the value of  $\alpha$ , though for higher number of terminals, it seems that the rank of our solution gets to be worse on average.

## O Implementation Details

In this section, we explain some implementation details of the mSTreg heuristic and also the parameters used for the different experiments.

In each iteration of the mSTreg algorithm, it is necessary to compute the mST. Since we are working with a complete graph, the computational complexity of the mST computation is  $O(N^2)$ . To reduce this cost, we compute the mST over a k-nearest neighbor (kNN) graph, where we set the value of  $k$  to  $\log(N)$ . While the resulting mST over the kNN graph may not always match the optimal mST, in practice, they often yield similar results. It is worth noting that the introduction of additional nodes, as described in Section K, may provide more significant benefits when using the mST computed over a kNN graph.

In Section L we have described different approaches to transform a full tree topology into a CST tree topology. The strategy used to collapse the SP nodes upon transforming a full tree topology into a CST tree was the one that updates the collapsed SP to the weighted geometric median, collapses greedily the SPs and determines the SP to be collapsed as the one with minimum distance to one of its neighbors (“update+greedy+Ordclosest”).

In all experiments, we set the `sampling_frequency` variable of Algorithm 1 equal to 3, and we set the maximum number of iterations of the mSTreg heuristic equal to 20.