Communication Tree Problems [*,†]

Carme Àlvarez Rafel Cases Josep Díaz Jordi Petit Maria Serna

ALBCOM Research group
Departament de Llenguatges i Sistemes Informàtics
Universitat Politècnica de Catalunya
Edifici Ω, Campus Nord
c/ Jordi Girona Salgado 1-3
08034 Barcelona (Spain)
{alvarez,cases,diaz,jpetit,mjserna}@lsi.upc.edu

Abstract

In this paper, we deal with the problem of constructing optimal communication trees satisfying given communication requirements. We consider two constant degree tree communication models and several cost measures. First, we analyze whether a tree selected at random provides a good randomized approximation algorithm, and we show that such a construction fails for some of the measures. Secondly, we provide approximation algorithms for the case in which the communication requirements are given by a random graph in two different random models, namely the classical $G_{n,p}$ and random geometric graphs. Finally, we conclude with some open problems.

1 Introduction

General communication problems involve a set of locations with communication requirements between pairs of them. The goal is to establish a communication pattern, often a tree, optimizing some communication parameter. Problems in which a communication tree has to be constructed arise in many applications. For instance, in phone communication, it is usual to have several locations with a known expected number of phone calls between each pair of locations. In this case the goal is to design a network to handle these calls in an optimal way. In distributed or mobile computing, there are shared resources as disks, input, output devices, etc., and system requirements that force the establishment of an optimal point-to-point communication. In tree-structured computations, the computational activity often is limited to the leaves of the tree. In such a case, it is important not only to distribute the tasks evenly among the leaves but also to build an adequate computation tree taking into account the communication parameters.

[*]This research was partially supported by the EU within the 6th Framework Programme under contract 15964 (AEOLUS) and the Spanish project TIN2005-25859-E. The work of the first and fifth authors was also supported by the Spanish CICYT project TIN-2005-09198-C02-02 (ASCE). The work of the third author was partially supported by the Distinció per a la recerca of the Generalitat de Catalunya. The work of the third and fourth authors was also supported by the Spanish CICYT project TIN-2004-07925-C03-01 (GRAMMARS)

[†]Part of this work was presented in the Workshop on Approximation and Randomized algorithms in Communication Networks (ARACNE-2000), Proceedings in Informatics, vol 8, pp 99-110, 2000.

Given a collection of terminal sites where some pairs of them want to exchange information, a *communication tree* is a tree that contains the set of sites but that might not contain direct links between communicating pairs [13]. Following the nomenclature in [20], when the terminal sites must appear as the leaves of the communication tree we speak of a *routing tree.* We model the input by a graph, whose nodes correspond to terminal sites and whose edges join pair of nodes that can communicate directly. In this paper we are interested in the problem of finding communication/routing trees of constant degree, minimizing different communication measures like *congestion*, *dilation*, *load* and *total communication* (see Section 2 for formal definitions).

In the case that the maximum degree of the communication tree is 2, the tree becomes a path and the communication tree becomes a *linear layout.* In this case, the corresponding problems are known as *cutwidth*, *bandwidth*, *vertex separation* and *optimal linear arrangement.* There is an extensive literature on linear layout problems see for example [15, 8]. Few hardness results are known for communication trees of maximum degree 3: the *minimum dilation communication tree* problem is NP-hard even for trees [17]; the *minimum congestion communication tree* problem is NP-hard for planar graphs [21], in contraposition with the *minimum congestion routing tree* problem (or minimum *carving-width*) which is solvable in polynomial time for planar graphs, but is NP-hard in general [20]. It is shown in [10] that there is a logarithmic gap between the minimum congestion and the minimum dilation of a given graph, where the minimum is taken over all routing trees with maximum degree 3. In the case that we consider communication trees with unbounded maximum degree the problems become easier: finding a communication tree of minimum total communication cost is in P [12], and an analogous result for the case of a routing tree is given in [1]. To the best of our knowledge no complexity results are known for the other optimization problems that we consider in this paper.

Our first question is whether the average cost over the uniform distribution provides a good approximation to the optimal cost. This is a natural question because this will allow the design of a simple approximation algorithm by simply selecting a tree according to the uniform distribution. We answer this question on the negative for some of the problems: we show that for a given graph, and for some measures, the average measure cost is far away from the optimal. The second question concerns the approximability of the problems when the input graph, representing the communication requirements, is a random graph. We deal with two models, the classical Erdös-Renyi model $\mathcal{G}_{n,p}$ [3] and the *random geometric* model $\mathcal{G}(n;r)$ [19].

We show that for any of the measures considered in this paper, we can produce a communication (routing) tree that with high probability has cost within a constant of the optimum when the graph is drawn at random. For the $\mathcal{G}_{n,p}$ model we show that, with high probability, any *balanced* routing tree will have cost within a constant of the optimum. For the $\mathcal{G}(n;r)$ model, an adequate balanced routing tree provides, with high probability, a constant approximation. In order to get this last result we will also give deterministic constant approximation algorithms when the given graph is an square mesh.

The paper starts by defining all the measures and problems we are interested in. Then we give a full treatment of the case in which we want to construct an optimal routing tree with maximum degree 3. We show how to extend the approximability results to the corresponding communication tree problems, and to the case of trees with maximum degree bounded by a constant, and we finish giving a list of interesting open problems.

2 The problems

We use standard notation to describe asympotics: O, Θ, Ω, o, and ω are used as defined in [5]. Recall that a sequence of events $(\mathcal{E}_n)_{n\geq 1}$ is said to occur *with high probability* if $\lim_{n\to\infty} \mathbf{Pr}[\mathcal{E}_n] = 1$, and that a sequence of events $(\mathcal{E}_n)_{n\geq 1}$ is said to occur with *with overwhelming probability (w.o.p.)* if $\mathbf{Pr}[\mathcal{E}_n] \geq 1 - 2^{-\Omega(n)}$ for all n large enough.

We also use standard graph theory notation: for an undirected graph G, we denote by $V(G)$ ($E(G)$) its vertex (edge) set, by $\Delta(G)$ the maximum degree of the vertices in the graph and by $\mathrm{diam}(G)$ its diameter. We also denote by uv the edge that joins u and v. Given a graph G and an edge uv of G, $G \setminus \{uv\}$ is the graph $(V(G), E(G) \setminus \{uv\})$.

Unless explicitly said all our trees are non-rooted. For a tree T, any node with degree one will be called a *leaf*, any non-leaf node will be called *internal*. Let $L(T)$ denote the set of leaves of a tree T. Given a tree T and two nodes $x, y \in V(T)$, $d_T(x, y)$ denotes the distance between x and y in T, counted as the number of edges of the unique path joining x and y.

Given a graph G, a *communication tree* for G is a tree T such that $V(G) \subseteq V(T)$ (no relationship is required between the set $E(G)$ and $E(T)$). A *routing tree* for G is a communication tree T such that $V(G) \subseteq L(T)$, and a *tree layout* for G is a routing tree such that every non-leaf node has exactly degree 3. A *linear layout* is a communication tree T such that T is a line and $V(G) = V(T)$. See Figure 1 for an illustration of different communication trees.

A communication tree T for a graph G associates in a natural way to each edge $uv \in E(G)$ the unique path $P_T(uv)$ that connects u and v in T.

Given a graph G and a communication tree T for G our basic communication measures are the following:

- The *dilation* $\lambda(uv, T, G)$ of an edge uv in $E(G)$ is the distance, from u to v in T, so

$$\lambda(uv, T, G) = d_T(u, v).$$

- The *congestion* $\theta(xy, T, G)$ of an edge xy in $E(T)$ is the number of edges uv in $E(G)$ such that the path from u to v in T traverses xy, therefore

$$\theta(xy, T, G) = |\{uv \mid uv \in E(G) \text{ and } P_T(uv) \text{ contains } xy\}|.$$

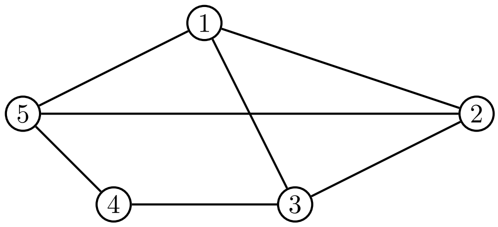
- The *congestion* $\vartheta(x, T, G)$ of a vertex x in $V(T)$ is the number of edges uv in $E(G)$ such that the path from u to v in T goes through x, that is

$$\vartheta(x, T, G) = |\{uv \mid uv \in E(G) \text{ and } P_T(uv) \text{ contains } x\}|.$$

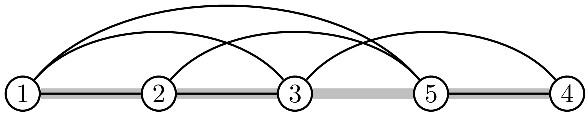
- The *communication load* $\delta(xy, T, G)$ of an edge xy in $E(T)$ is the number of vertices u in $V(G)$ such that some of its neighboring vertices in G lies in a different component of T after the removal of the edge xy, so

$$\delta(xy, T, G) = |\{u \mid \exists uv \in E(G)\ u \text{ and } v \text{ lie in different components of } T \setminus \{xy\}\}|.$$

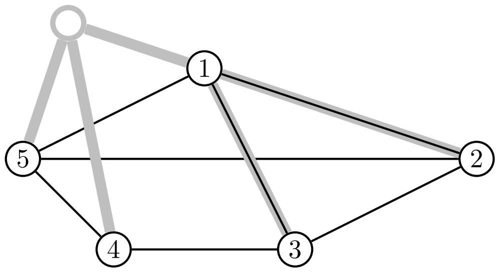
Observe that in a routing tree for a graph G there might be some free leaves that are not associated to any vertex in G. Those leaves can be ignored except in those cases when there is a strict degree requirement for the internal nodes. In the particular case of



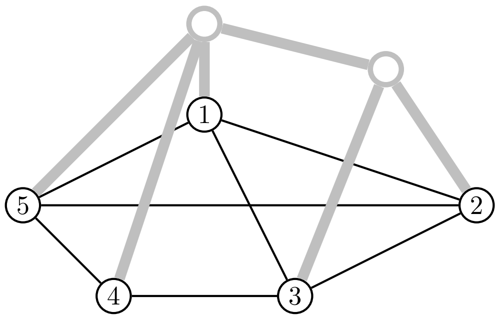
(a) G



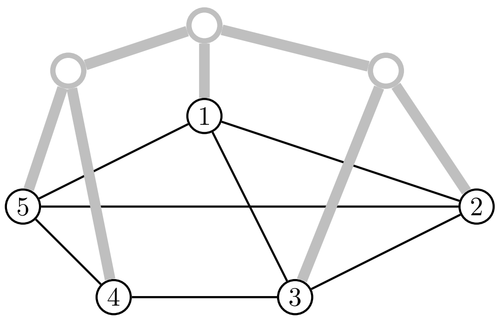
(b) Linear layout



(c) Communication tree



(d) Routing tree



(e) Tree layout

Figure 1: Different types of communication trees for a graph.

a tree layout, a free leaf can be removed when the operation is followed by the elimination of the parent of the leaf and the addition of an edge between the two siblings. Observe that this modification preserves any of the measures defined above.

Now we introduce the problems, for clarity in the exposition, at first we define only the problems for the case in which we ask for the minimum a measure over the set of tree layouts:

- Minimum Tree Dilation (MinTD):
  $\text{MINTD}(G) = \min_T \text{TD}(T, G)$ where $\text{TD}(T, G) = \max_{uv \in E(G)} \lambda(uv, T, G)$.

- Minimum Tree Congestion (MinTC):
  $\text{MINTC}(G) = \min_T \text{TC}(T, G)$ where $\text{TC}(T, G) = \max_{xy \in E(T)} \theta(xy, T, G)$.

- Minimum Tree Vertex Congestion (MinTVC):
  $\text{MINTVC}(G) = \min_T \text{TVC}(T, G)$ where $\text{TVC}(T, G) = \max_{x \in V(T)} \vartheta(x, T, G)$.

- Minimum Tree Communication Load (MinTCL):
  $\text{MINTCL}(G) = \min_T \text{TCL}(T, G)$ where $\text{TCL}(T, G) = \max_{xy \in E(T)} \delta(xy, T, G)$.

- Minimum Tree Length (MinTL):
  $\text{MINTL}(G) = \min_T \text{TL}(T, G)$ where $\text{TL}(T, G) = \sum_{uv \in E(G)} \lambda(uv, T, G)$. Notice that we also have $\text{TL}(T, G) = \sum_{xy \in E(T)} \theta(xy, T, G)$.

To distinguish the different problems and measures, we use the following notation: For any measure $\text{F} \in \{\text{TD}, \text{TC}, \text{TVC}, \text{TCL}, \text{TL}\}$ and a graph G,

- d-MINRF(G) is the minimum value of F over all routing trees for G with internal nodes of degree exactly d.

- $d_{\leq}$-MINRF(G) is the minimum value of F over all routing trees T for G with $\Delta(T) \leq d$.

- $d_{\leq}$-MINCF(G) is the minimum value of F over all communication trees T for G with $\Delta(T) \leq d$.

We will omit the d prefix when $d = 3$.

Notice that by replacing an internal node of degree 2 by an edge joining its two neighbors we obtain a routing tree with smaller cost (for any of the considered measures). Therefore the optimal routing tree with maximum degree 3 can be found on trees with internal nodes of degree 3. However, this is not true for higher degrees.

Without loose of generality we will assume from now on that any graph G is a connected graph. Observe that the optimal communication tree cost, for any of the introduced measures is attained by connecting optimal trees for all the connected components of a graph.

The following basic upper bounds on the cost of a tree layout will prove to be useful.

Lemma 1. Let G be any graph with n nodes and m edges. Let T be any tree layout of G. Then, $\text{TC}(T, G) \leq m$, $\text{TCL}(T, G) \leq n$, $\text{TD}(T, G) \leq \text{diam}(T)$, $\text{TL}(T, G) \leq m\,\text{diam}(T)$.

We make use of *balanced trees*. We say that a tree with n leaves is balanced if there exists a node u such that for any leaf v the distance between u and v is at most $\log n$. As there is always a balanced tree with n leaves and this tree has diameter at most $2\log n$, then the previous lemma implies that we have $\textsc{MinTD}(G) \le 2\log n$, and $\textsc{MinTL}(G) \le 2m\log n$. On the other hand, notice that $\textsc{tc}(T,G)$ can be $\Theta(n^2)$ and $\textsc{tcl}(T,G)$ can be $\Theta(n)$, for instance when G is a complete graph on n vertices.

The following basic inequalities follow from the definitions.

Lemma 2. Let G be any graph with n nodes and m edges and let T be any tree layout for G. Then, $\textsc{tcl}(T,G) \le 2\textsc{tc}(T,G)$ and $\textsc{tc}(T,G) \le \textsc{tvc}(T,G) \le 3\textsc{tc}(T,G)$.

Let us finish this section with some results on trees that we will need later. We say that an edge in a tree T is a *s-splitter* if its removal splits T in two rooted trees, each one with at least $\lfloor s \rfloor$ leaves. The following lemma is similar to a result referred in [16].

Lemma 3. Any tree layout T with n leaves contains a $n/3$-splitter edge.

Proof. We prove it by induction on the number of leaves. The base case is $n = 3$. There only exists a tree layout with three leaves, which satisfies the property. Assume that any tree layout with n' leaves always contains a $\frac{1}{3}n'$-splitter edge for some $n' \ge 3$. Let T be any tree with $n = n' + 1$ leaves.

Take any node with two adjacent leaves from T and substitute those three vertices by a marked leaf m, obtaining a new tree T' with $n' = n - 1$ leaves. By the induction hypothesis, T' contains a $\frac{1}{3}n'$-splitter edge uv that splits T' into two rooted trees T'_u and T'_v rooted at u and v respectively. Also, uv splits T into two rooted trees T_u and T_v rooted at u and v respectively. Let a' be the number of leaves in T'_u and let b' be the number of leaves in T'_v. Also, let a be the number of leaves in T_u, and let b be the number of leaves in T_v. Without loss of generality, assume $a' \le b'$.

As uv is a $\frac{1}{3}n'$-splitter edge for T', we have

$$\tfrac{1}{3}n' \le a' \le \tfrac{1}{2}n' \le b' \le \tfrac{2}{3}n' \quad \text{and} \quad a' + b' = n'.$$

We break the proof in several cases:

Case 1: m belongs to T'_u and $\frac{1}{3}n' \le a' < \frac{1}{2}n'$. As $a = a' + 1$, we have $\frac{1}{3}n \le a \le \frac{1}{2}n$ and so uv is a $\frac{1}{3}n$-splitter edge for T.

Case 2: m belongs to T'_u and $a' = \frac{1}{2}n'$. In this case, n must be even, $b' = a'$ and $a = a' + 1$. So, $\frac{1}{2}n \le a = \frac{1}{2}(n+1) \le \frac{2}{3}n$ and $\frac{1}{3}n \le b = \frac{1}{2}(n-1) \le \frac{1}{2}n$. This proves that uv is a $\frac{1}{3}n$-splitter edge for T.

Case 3: m belongs to T'_v and $\frac{1}{2}n' \le b' < \frac{2}{3}n'$. As $b = b' + 1$, we have $\frac{1}{2}n \le b \le \frac{2}{3}n$ and so uv is a $\frac{1}{3}n$-splitter edge for T.

Case 4: m belongs T'_v and $b' = \frac{2}{3}n'$. In this case, n' is a multiple of 3, $a' = \frac{1}{3}n$ and $b = b' + 1$. Let uw and ut be the two edges adjacent to v in T'_v. Let T'_w the subtree of T'_v rooted at w and let T'_t the subtree of T'_v rooted at t. Also, let b'_1 be the number of leaves in T'_w and let b'_2 be the number of leaves in T'_t. Assume, without loss of generality, that $b'_1 \le b'_2$. Notice that $1 \le b'_1 \le \frac{1}{3}n \le b'_2 \le \frac{2}{3}n$. Several new sub-cases must be considered:

Case 4.1: m belongs T'_t. In this case, let T_1 be the tree rooted at v obtained by splitting T with vt. Let c be the number of leaves in T_1. Then, $c = a' + b'_1$ and so $\frac{1}{3}n \le c \le \frac{2}{3}n$. Therefore, vt is a $\frac{1}{3}n$-splitter edge for T.

Case 4.2: m belongs T'_w and $b'_1 < \frac{1}{3}n'$. As in the previous case, let T_1 be the tree rooted at v obtained by splitting T with vt. Let c be the number of leaves in T_1. Then, $c = a' + b'_1 + 1$ and so $\frac{1}{3}n \le c \le \frac{2}{3}n$. Therefore, vt is also a $\frac{1}{3}n$-splitter edge for T.

Case 4.3: m belongs T'_w and $b'_1 = \frac{1}{3}n'$. In this case, let T_2 be the tree rooted at w obtained by splitting T with vw. Let d be the number of leaves in T_2. Then, $d = b'_1 + 1$ and so $\frac{1}{3}n \le d \le \frac{2}{3}n$. Therefore, vw is a $\frac{1}{3}n$-splitter edge for T.

As there are no other possible cases, the induction step is proved and the lemma follows. $\square$

For any node u in a given tree T and any integer i, let $L_{>i}(T,u)$ denote the set of leaves of T at distance greater than i from u. All through the paper $\log n$ means $\log_2 n$ and in many cases $\lfloor \log_2 n \rfloor$.

Lemma 4. Let $\alpha, \beta \in (0,1)$. Let T be a tree with internal nodes of degree 3 and with n leaves. Then, for any node u in T, it holds that $L_{>\alpha \log n}(T,u) \geq \beta n$ for large enough n.

Proof. Starting at a vertex u, consider a breadth first search process in T. At iteration i, all nodes at distance i from u have been marked and there can be at most $3 \cdot 2^{i-1}$ such nodes. Therefore,

$$L_{>\alpha \log n}(u) \geq n - 1 - \sum_{i=1}^{\alpha \log n} 3 \cdot 2^{i-1} \geq n - 3n^{\alpha} + 2 \geq \beta n$$

by the assumption that n is large enough. $\square$

3 Average cost under the uniform distribution

In this section we proceed to fix a graph G and seek for the average costs of the MinTD and MinTL problems over all possible tree layouts with n leaves with uniform distribution.

Let us define the basic nomenclature and notation we are going to use through this section.

The degree $d(x)$ of a node x in a non-rooted tree is defined as the number of adjacent nodes to x, whereas the degree $d(x)$ of a node x in a rooted tree is defined as the number of subtrees of x. Therefore, a leaf in a rooted tree has degree 0. Let $l(T)$ denote the number of leaves in T. A *k-ary tree* is a rooted tree such that each internal node has degree k.

Following [14] we define two families of rooted trees. An *ordered tree* is defined recursively as formed by a root and an *ordered sequence* (possibly empty) of ordered trees, called subtrees of the root. A *non-ordered tree* is defined recursively as formed by a root and a *multiset* (possibly empty) of non-ordered trees, called also subtrees of the root. Notice that a non-ordered tree can have multiple representations using ordered rooted trees. Finally, we use the the term *n-binary trees*, to refer to the ordered and non-labeled, rooted binary trees with n internal nodes.

Given a non-ordered tree T let us consider, for each node $x \in V(T)$, the multiset of subtrees hanging from x. In this multiset, subtrees can have different degrees of multiplicity. Let $\beta_i(x)$ be the number of non-isomorphic subtrees that have degree of multiplicity exactly i. Let $\beta_i(T) = \sum_{x \in V(T)} \beta_i(x)$ (see Figure 2).

Given a non-ordered tree T, let $\mathcal{T}(T)$ be the set of all ordered trees, which are representations of T. Let $\mathcal{E}(T)$ be the set on non-ordered trees obtained by labelling the leaves in T, with $\{1, \ldots, l(T)\}$. Define $c(T) = |\mathcal{T}(T)|$ and $e(T) = |\mathcal{E}(T)|$. In order to construct the set $\mathcal{T}(T)$ from a given T, we have to

- permute in all possible ways all the subtrees of each $x \in V(T)$,
- and remove duplicated trees

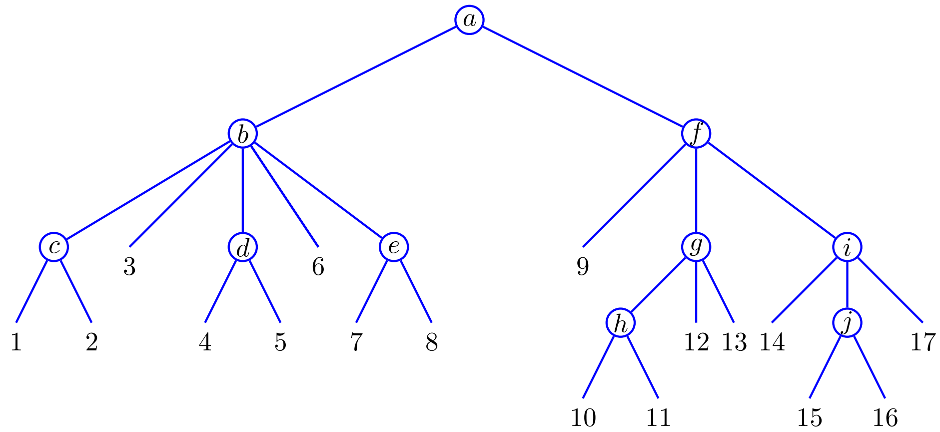


Figure 2: A rooted-tree T in which $\beta_2(T) = 9$ (as $\beta_2(x) = 1$ for $x \in \{b, c, d, e, f, g, h, i, j\}$), $\beta_3(T) = 1$ (as $\beta_3(b) = 1$), and $\beta_i(T) = 0$ for $i \geq 4$.

Notice that the total number of such permutations is given by,

$$\prod_{x \in V(T)} d(x)!, \tag{1}$$

and in the case that T is a k-ary tree with n internal nodes, expression (1) is $(k!)^n$. If T has many nodes with identical subtrees, the described construction will yield each element in $\mathcal{T}(T)$ a plurality of times. Moreover, this number of repetitions will be the same for every element in $\mathcal{T}(T)$.

In the same manner, starting from T we can construct the set $\mathcal{E}(T)$ by considering the $l(T)!$ possible labelling and removing the duplicated trees.

Lemma 5. Let T be a non-ordered tree. Then, the number of repetitions obtained by performing the construction described above to obtain $\mathcal{T}(T)$ is the same that the number of repetitions obtained in the construction of the set $\mathcal{E}(T)$, and this number is,

$$\prod_{i>1} (i!)^{\beta_i(T)}.$$

Proof. In both constructions, the repetitions are associated with the existence of identical trees hanging from the same node. For every constructed tree, each time that i identical subtrees occur in a node, there are $i!$ identical constructions on the same tree. $\square$

From this result, we obtain the expressions for $c(T)$ and $e(T)$,

Lemma 6. The number of ordered trees representing a given non-ordered tree T is

$$c(T) = \frac{\prod_{x \in V(T)} d(x)!}{\prod_{i>1} (i!)^{\beta_i(T)}}.$$

The number of different non-ordered labeled trees obtained by labelling the leaves of a given non-ordered tree T is

$$e(T) = \frac{l(T)!}{\prod_{i>1} (i!)^{\beta_i(T)}}.$$

Noticing that if T is a k-ary tree with n internal nodes the number of leaves is $l(T) = (k-1)n + 1$, we obtain the following interesting corollary.

Corollary 1. For any non-ordered k-ary tree T, $\frac{e(T)}{c(T)}$ is independent of the shape of T and depends only on the size of T. If T has n internal nodes, we get

$$\frac{e(T)}{c(T)} = \frac{((k-1)n+1)!}{(k!)^n}.$$

After this necessary introduction to notation and basic results, let us return to our goal. We need to consider non-ordered trees with labels on their leaves. Let n-RLN denote the set of non-ordered trees with $n+1$ labeled leaves and such that each of its n internal nodes has degree 2 (recall that in a rooted tree the degree is defined as the outdegree). In contraposition, the tree layouts we are using to define our problems are non-rooted trees, with n internal nodes and $n+2$ labeled leaves (associated to the vertices of G) and such that each of its n internal nodes has degree 3. Let us denote such trees as n-NLN trees.

Lemma 7. The set of n-NLN trees is in one-to-one correspondence with the set of n-RLN trees.

Proof. Let us define the following bijection between the n-NLN trees and the n-RLN trees: given a n-NLN tree, suppress the leaf with label $n+2$ and make its neighbor the root of the new n-RLN tree. □

The isomorphism just described, will allow us to interchange the study of both families of trees.

Given a binary tree T and a property function f on T, as for example internal path length, height, etc, we say f is *order invariant* if the value of f is the same for all binary trees that are equivalent to T as non-ordered trees. The concept of order invariance can be extended to RLN trees in the obvious way.

Lemma 8. For any given property function on a tree, which is order invariant, the average value of the function is the same on the n-RLN trees and on the n-binary trees.

Proof. Let $\mathcal{B}_n$ denote the set of all n-binary trees, $\mathcal{E}_n$ the set of all n-RLN trees and $\mathcal{C}_n$ the set of all non-ordered, rooted and non-labeled binary trees with n internal nodes. Also, let $f(T)$ be a property function of a tree T that is order invariant. Do the following decompositions,

$$\mathcal{B}_n = \bigcup_{T \in \mathcal{C}_n} \mathcal{T}(T) \qquad \text{and} \qquad \mathcal{E}_n = \bigcup_{T \in \mathcal{C}_n} \mathcal{E}(T),$$

where $\mathcal{T}(T)$ and $\mathcal{E}(T)$ are restricted to binary trees. Then

$$c(T) = |\mathcal{T}(T)| = 2^{|T| - \beta_2(T)} \text{ and } e(T) = |\mathcal{E}(T)| = \frac{(|T|+1)!}{2^{\beta_2(T)}}.$$

Therefore,

$$\sum_{T' \in \mathcal{B}_n} f(T') = \sum_{T \in \mathcal{C}_n} \sum_{T' \in \mathcal{T}(T)} f(T') = \sum_{T \in \mathcal{C}_n} f(T) c(T)$$
$$\sum_{T'' \in \mathcal{E}_n} f(T'') = \sum_{T \in \mathcal{C}_n} \sum_{T'' \in \mathcal{E}(T)} f(T'') = \sum_{T \in \mathcal{C}_n} f(T) e(T).$$

So, we get

$$\sum_{T\in\mathcal{C}_n} f(T)e(T) = \sum_{T\in\mathcal{C}_n} f(T)c(T)\frac{e(T)}{c(T)} = \frac{(n+1)!}{2^n}\sum_{T\in\mathcal{C}_n} f(T)c(T),$$

where the last equality follows from Corollary 1 taking $k = 2$, and therefore

$$\sum_{T''\in\mathcal{E}_n} f(T'') = \frac{(n+1)!}{2^n}\sum_{T'\in\mathcal{B}_n} f(T').$$

As a consequence,

$$\frac{\sum_{T''\in\mathcal{E}_n} f(T'')}{|\mathcal{E}_n|} = \frac{\sum_{T'\in\mathcal{B}_n} f(T')}{|\mathcal{B}_n|},$$

and we can conclude that the average value of of $f(T'')$ for $T'' \in \mathcal{E}_n$ is the same that the average value of $f(T')$ for $T' \in \mathcal{B}_n$. $\square$

The average distance between two different leaves among all n-binary trees is known to be one unit more than the average depth of a leaf, which is $4^n/\binom{2n}{n}-1 = \sqrt{\pi n}-1+o(\sqrt{n})$ (see Section 2.3.4.5 of [14]). This result, together with Lemma 8, implies the following theorem.

Theorem 1. Given a graph $G = (V, E)$ with $|V| = n$ and $|E| = m \geq 1$ the average tree layout length for G is $\Theta(m\sqrt{n})$, and the average tree layout dilation for G is $\Theta(\sqrt{n})$; the average being taken over all possible n-NLN trees.

The previous theorem says that using a random NLN as routing tree, will provide communication costs far away from the optimal ones, as by Lemma 1, selecting a routing tree of logarithmic diameter will do better than a randomly selected routing tree. Note however that a routing tree with logarithmic diameter not always provides the optimum, in particular when the graph G is a line or a cycle, a worm (caterpillar with hair length 1) gives the optimum (see Figure 3).

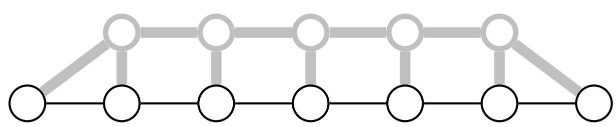


Figure 3: A worm (caterpillar tree) layout for a line.

The previous results can be extended to tree layouts with internal nodes of degree d, for any constant $d \geq 3$. In such a case the average distance has the same order, however with a different constant depending on d.

4 $\mathcal{G}_{n,p}$ graphs

In this section we show that w.o.p. all of our tree layout problems are approximable within a constant for random graphs drawn from the classical $\mathcal{G}_{n,p_n}$ model provided that

$C_0/n \leq p_n \leq 1$ for some properly characterized parameter $C_0 > 1$. This particular probability bound guaranties random graphs with a giant component. In fact, our results establish that the cost of any *balanced* tree layout for such a random graph is within a constant of the optimal cost w.o.p..

Let us recall the definition of the class of random graphs: Let n be a positive integer and p a probability. The class $\mathcal{G}_{n,p_n}$ is a probability space over the set of undirected graphs $G = (V, E)$ on the vertex set $V = \{1, \ldots, n\}$ determined by $\mathbf{Pr}[uv \in E] = p$ with these events mutually independent.

We recall from [9] the definition of a class of graphs that captures the properties we need to bound our tree layout costs on graphs randomly selected from the $\mathcal{G}_{n,p_n}$ distribution.

Definition 1 (Mixing graphs). Let $\epsilon \in (0, \frac{1}{9})$, $\gamma \in (0,1)$ and define $C_{\epsilon,\gamma} = 3(1 + \ln 3)(\epsilon\gamma)^{-2}$. Consider a sequence $(c_n)_{n\geq 1}$ such that $C_{\epsilon,\gamma} \leq c_n \leq n$ for all $n \geq n_0$ for some natural n_0. A graph $G = (V, E)$ with $|V| = n$ and $|E| = m$ is said to be (ϵ, γ, c_n)-*mixing* if $m \leq (1+\gamma)\frac{1}{2}nc_n$ and for any two disjoint subsets $A, B \subset V$ such that $|A| \geq \epsilon n$ and $|B| \geq \epsilon n$, it holds that

$$1 - \gamma \leq \left. \frac{\mathsf{cut}(A,B)}{|A||B|} \middle/ \frac{c_n}{n} \right. \leq 1 + \gamma,$$

where $\mathsf{cut}(A, B)$ denotes the number of edges in E having one endpoint in A and another in B.

Our interest in mixing graphs is motivated by the fact that, with overwhelming probability, $\mathcal{G}_{n,p}$ graphs are mixing:

Lemma 9 ([9]). Let $\epsilon \in (0, \frac{1}{9})$, $\gamma \in (0,1)$ and define $C_{\epsilon,\gamma} = 3(1 + \ln 3)(\epsilon\gamma)^{-2}$. Consider a sequence $(c_n)_{n\geq 1}$ such that $C_{\epsilon,\gamma} \leq c_n \leq n$ for all $n \geq n_0$ for some natural n_0. Then, for all $n \geq n_0$, random graphs drawn from $\mathcal{G}_{n,p_n}$ with $p_n = c_n/n$ are (ϵ, γ, c_n)-mixing with probability at least $1 - 2^{-\Omega(n)}$.

Let us say that a graph G with n nodes satisfies the *dispersion* property if, for any two disjoint subsets A and B of $V(G)$ with $|A| \geq \epsilon n$ and $|B| \geq \epsilon n$, it is the case that there is at least one edge between A and B. From Definition 1 we get $\mathsf{cut}(A, B) \geq (1-\gamma)\epsilon^2 nc_n$, which implies $\mathsf{cut}(A, B) \geq 1$ for n large enough. Therefore mixing graphs satisfy the dispersion property.

Using a balanced tree and the dispersion property, it is possible to obtain a constant approximation for the MinTC, MinTD, MinTL and MinTTCL problems on mixing graphs:

Lemma 10. Let $\epsilon \in (0, \frac{1}{9})$, $\gamma \in (0,1)$. Consider a sequence $(c_n)_{n\in\mathbb{N}}$ such that $C_{\epsilon,\gamma} \leq c_n \leq n$ for all $n \geq n_0$ for some natural n_0. Let G be any (ϵ, γ, c_n)-mixing graph with n nodes where n is large enough. Let T_b be a balanced tree layout of G. Then,

$$\frac{\textsc{tc}(T_b, G)}{\textsc{mintc}(G)} \leq \frac{1+\gamma}{2(1-\gamma)\epsilon^2}, \quad \frac{\textsc{td}(T_b, G)}{\textsc{mintd}(G)} \leq \frac{1+\gamma}{(1-\gamma)^2},$$
$$\frac{\textsc{tl}(T_b, G)}{\textsc{mintl}(G)} \leq \frac{(1+\gamma)^2}{(1-\gamma)^3\epsilon^2}, \quad \frac{\textsc{tcl}(T_b, G)}{\textsc{mintcl}(G)} \leq \frac{3}{2(1-7\epsilon)}.$$

Proof. To prove this result, we present lower and upper bounds to each of the considered problems. The lower bounds hold for any tree layout, while the upper bounds are obtained through a balanced tree layout.

Lower bound for $\text{MINTC}(G)$: Consider any tree layout T of G. Let uv be a $n\sqrt{\epsilon}$-splitter edge of T that separates T into two binary trees T_u and T_v rooted at u and v respectively. Such an edge must exist by Lemma 3. Let $\alpha, \beta \in (0, 1)$ be two parameters to be determined latter. By Lemma 4, the set of leaves $L_u = L_{\alpha \log(n\sqrt{\epsilon}}(T_u, u)$ verifies $|L_u| \geq \beta n\sqrt{\epsilon}$. Also, the set of leaves $L_v = L_{\alpha \log(n\sqrt{\epsilon}}(T_v, v)$ verifies $|L_v| \geq \beta n\sqrt{\epsilon}$. Setting $\beta = \sqrt{\epsilon}$, we have $|L_u| \geq \epsilon n$ and $|L_v| \geq \epsilon n$. As G is (ϵ, γ, c_n)-mixing, we have $\mathsf{cut}(L_u, L_v) \geq (1-\gamma)|L_u||L_v|c_n/n \geq (1-\gamma)\epsilon^2 nc_n$. Thus, $\theta(uv, T, G) \geq (1-\gamma)\epsilon^2 nc_n$. So, $\text{TC}(T, G) \geq (1-\gamma)\epsilon^2 nc_n$ and as T is arbitrary we get $\text{MINTC}(G) \geq (1-\gamma)\epsilon^2 nc_n$.

Lower bounds for $\text{MINTD}(G)$ and $\text{MINTL}(G)$: Observe that for all $x \in L_u$ and all $y \in L_v$, $d_T(x, y) \geq 2\alpha \log(n\sqrt{\epsilon}) + 1$. Setting $\alpha = 1 - \gamma$, we have

$$\text{TD}(T, G) \geq 2\alpha \log(n\sqrt{\epsilon}) + 1 \geq (1-\gamma)^2 2 \log n$$

and

$$\text{TL}(T, G) \geq (1-\gamma)\epsilon^2 nc_n(2\alpha \log(n\sqrt{\epsilon}) + 1) \geq (1-\gamma)^3 2\epsilon^2 c_n n \log n.$$

As T is arbitrary, $\text{MINTD}(G) \geq (1-\gamma)^2 2 \log n$ and $\text{MINTL}(G) \geq 2(1-\gamma)^3 \epsilon^2 c_n n \log n$.

Lower bound for $\text{MINTCL}(G)$:

Let xy be a $\lfloor \frac{1}{3}n \rfloor$-splitter edge of T separating T into two binary trees T_x and T_y rooted at x and y respectively. Let L_x and L_y denote the leaves of T_x and T_y respectively. For n large enough, $|L_x| \geq (1-\epsilon)\frac{1}{3}n$ and $|L_y| \geq (1-\epsilon)\frac{1}{3}n$. Let L_x^1 be a subset of size $\lceil \epsilon n \rceil$ of L_x and let L_y^1 be a subset of the same size of L_y. Because of dispersion, there must be at least one edge in $E(G)$ connecting a node from L_x^1 to a node in L_y^1. Let $u_x^1 u_y^1$ be such edge and let v_x^1 be a node in $L_x \setminus L_x^1$ and let v_y^1 be a node in $L_y \setminus L_y^1$

Now we will construct recursively two sequences of sets L_x^i and L_y^i for $1 < i \leq (1-\epsilon)\frac{1}{3}n - (1+\epsilon)\epsilon n$: Let $L_x^i = (L_x^{i-1} \setminus \{u_x^{i-1}\}) \cup \{v_x^{i-1}\}$, and let $L_y^i = (L_y^{i-1} \setminus \{u_y^{i-1}\}) \cup \{v_y^{i-1}\}$.

As, the two sets have size $\lceil \epsilon n \rceil$, by dispersion, there must be at least one edge in $E(G)$ connecting a node from L_x^i to a node in L_y^i. Call $u_x^i u_y^i$ the endpoints of such an edge.

Let v_x^i be a node in $L_x \setminus (L_x^i \cup \{u_x^j \mid 1 \leq j \leq i\})$ and similarly let v_y^i be a node in $L_y \setminus (L_y^i \cup \{u_y^j \mid 1 \leq j \leq i\})$, notice that such nodes must exist.

By construction, all nodes in $\{u_x^i \mid 1 \leq i \leq (1-\epsilon)\frac{1}{3}n - (1+\epsilon)\epsilon n\}$ are connected in G to some node in L_y and, likewise, all nodes in $\{u_y^i \mid 1 \leq i \leq (1-\epsilon)\frac{1}{3}n - (1+\epsilon)\epsilon n\}$ are connected in G to some node in L_x. Therefore,

$$\text{TCL}(T, G) \geq 2 \cdot ((1-\epsilon)\tfrac{1}{3}n - (1+\epsilon)\epsilon n) \geq (1-7\epsilon)\tfrac{2}{3}n.$$

As T is arbitrary, we have $\text{MINTCL}(G) \geq (1-7\epsilon)\frac{2}{3}n$.

Upper bounds: Let m denote the number of edges of the graph G. Using Lemma 1, we have $\text{TCL}(T_b, G) \leq n$. Moreover, as G is mixing, we also obtain $\text{TC}(T_b, G) \leq m \leq (1+\gamma)\frac{1}{2}nc_n$. As T_b is a balanced tree of G, its diameter is at most $2\lceil \log n \rceil \leq 2(1+\gamma)\log n$. Therefore, we have $\text{TD}(T_b, G) \leq 2(1+\gamma)\log n$ and $\text{TL}(T_b, G) \leq 2m(1+\gamma)\log n \leq (1+\gamma)^2 nc_n \log n$. □

Notice in the proof of the previous lemma, the lower bound works for any tree layout, while the upper bound is obtained through a balanced tree layout.

As an immediate consequence of Lemmas 2, 9, and 10, we get the following result.

Theorem 2. Let $\epsilon \in (0, \frac{1}{9})$, $\gamma \in (0,1)$ and define $C_{\epsilon,\gamma} = 3(1+\ln 3)(\epsilon\gamma)^{-2}$. Consider a sequence $(c_n)_{n\geq 1}$ such that $C_{\epsilon,\gamma} \leq c_n \leq n$ for all $n \geq n_0$ for some natural n_0 and let $p_n = c_n/n$. Then, with overwhelming probability, the problems MinTD, MinTC, MinTVC, MinTCL and MinTL can be approximated within a constant on random graphs $\mathcal{G}_{n,p_n}$ using a balanced tree layout. Moreover, in the case of the MinTD, the approximation factor can be made as small as desired.

5 Square meshes

In this section we study our tree layout problems on square meshes. This is intended as an intermediate step to treat random geometric graphs on the next section. In the following we will denote an $n \times n$ mesh by L_n: $V(L_n) = \{1, \ldots, n\}^2$ and $E(L_n) = \{uv \;:\; u \in V(L_n) \wedge v \in V(L_n) \wedge \|u - v\|_2 = 1\}$.

The following result presents a lower bound of the cost of a mesh.

Lemma 11 (Lower bounds). Let n be a sufficiently large natural. Then,

$$\text{MINTC}(L_n) \geq \tfrac{1}{2}n, \qquad \text{MINTD}(L_n) \geq \log n,$$
$$\text{MINTL}(L_n) \geq 6n^2 - 8n + 1, \qquad \text{MINTCL}(L_n) \geq \tfrac{\sqrt{6}}{3}n.$$

Proof. Let (A, B) be a partition of $V(L_n)$. We claim that $\mathsf{cut}(A,B) \geq \min\left\{\sqrt{|A|}, \sqrt{|B|}\right\}$: if A includes an entire row of nodes, and B includes an entire row of nodes, then each column includes an edge with one endpoint in A and the other in B, which contributes 1 to $\mathsf{cut}(A,B)$, so that $\mathsf{cut}(A,B) \geq n$. If B contains no entire row or column, and at least as many rows as columns have non-empty intersection with B, then there are at least $\sqrt{|B|}$ such rows, and each contains a cutting edge which contributes 1 to $\mathsf{cut}(A,B)$, so that $\mathsf{cut}(A,B) \geq \sqrt{B}$. Applying similar arguments to the other possible cases, we have

$$\mathsf{cut}(A,B) \geq \min\left\{\sqrt{|A|},\ \sqrt{|B|},\ n\right\}$$

but this minimum is always achieved at $\sqrt{|A|}$ or at $\sqrt{|B|}$, proving the claim.

Let T be any tree layout of L_n. Let uv be a $\lfloor n^2/3 \rfloor$-splitter edge of T. As uv determines a partition (A,B) of L_n with $|A|, |B| \geq \lfloor n^2/3 \rfloor$, the congestion of edge uv is at least equal to $\min\left\{\sqrt{|A|},\ \sqrt{|B|}\right\} \geq \sqrt{\lfloor n^2/3 \rfloor}$. Therefore, $\text{TC}(T, L_n) \geq \sqrt{\lfloor n^2/3 \rfloor} \geq \frac{1}{2}n$. Now the MINTC result follows because T is arbitrary.

A linear layout φ of a $n \times n$ mesh is a one-to-one function that maps the nodes of the mesh to $\{1, \ldots, n^2\}$. For any $i \in \{1, \ldots, n^2\}$, let $\partial(i, \varphi, L_n)$ denote the number of vertices $u \in V(L_n)$ with $\varphi(u) \leq i$ which are connected in $E(L_n)$ to some other node $v \in V(L_n)$ such that $\varphi(v) > i$. Let φ_D denote the "diagonal layout" of the mesh (see Figure 4). As a special case of [4, Corollary 9], we have that for any linear layout φ on L_n and any $k \in \{1, \ldots, n^2\}$, $\partial(k, \varphi, L_n) \geq \partial(k, \varphi_D, L_n)$. This means that $\partial(\frac{1}{3}n^2, \varphi, L_n) \geq q_n$, where q_n is the smallest positive integer such that $\sum_{i=1}^{q_n} i \geq \frac{1}{3}n^2$. A simple computation shows that $q_n = \lfloor \frac{1}{6}\sqrt{9 + 24n^2} - \frac{1}{2} \rfloor$.

To prove the MINTD and the MINTCL lower bounds, let T be again any tree layout of L_n and, let uv be a $\lfloor n^2/3 \rfloor$-splitter edge of T. Therefore, there are at least q_n leaves from one subtree (let us say rooted at u) connected in L_n to at least one other leaf in the other subtree (rooted at v). Then, at least one of the q_n leaves is at distance at least $\log q_n$ from

					n^2-1	n^2
						n^2-2
$\vdots$						
10						
6	9	$\ddots$				
3	5	8				
1	2	4	7	$\cdots$		

n	$n-1$	$n-2$	$\cdots$	3	2	0
$n-1$	n	n	$n-1$	$\ddots$	4	3
$\vdots$	$n-1$	n	n	$n-1$		4
4		$n-1$	n	n	$n-1$	$\vdots$
3	4	$\ddots$	$n-1$	n	n	$n-1$
2	3	4		$n-1$	n	n
1	2	3	4	$\cdots$	$n-1$	n

Figure 4: At left, diagonal linear layout of the $n \times n$ mesh: at each node u, $\varphi(u)$ is shown. At right, vertex congestion induced by the diagonal ordering: at each node u, $\partial(\varphi_D(u), \varphi_D, L_n)$ is shown.

u. So, we have $\text{TD}(T, L_n) \geq \log q_n + 2$ and $\text{TCL}(T, G) \geq \partial(\frac{1}{3}n^2, \varphi, L_n)$. As T is arbitrary, for n sufficiently large, $\text{MINTD}(L_n) \geq \log q_n + 2 \geq \log n$ and $\text{MINTCL}(L_n) \geq \frac{\sqrt{6}}{3}n$.

To prove the MINTL result, let G be any graph with t nodes. Observe that in any tree layout of G no edge can have length 0 or 1. Also, observe that, at most, only $t/2$ edges can have length 2 (a balanced tree) and that, at most, only $t - 1$ edges can have length 3 (a worm). Finally, observe that all not yet counted edges must have, at least, length 4. In the case of L_n with $t = n^2$ nodes and $m = 2n^2 - 2n$ edges, we get

$$\text{MINTL}(L_n) \geq 2(n^2/2) + 3(n^2 - 1) + 4(m - (n^2/2) - (n^2 - 1)) = 6n^2 - 8n + 1.$$

□

In order to get upper bounds, we shall analyze first a recursive algorithm to produce a tree layout of a $n \times n$ mesh, for the case that n is a power of two. In all the illustrations of tree layouts we use black nodes to represent internal nodes an white nodes to represent leaves.

Definition 2 (The recursive algorithm). Let L_n be a $n \times n$ mesh with $n = 2^k$ for some integer $k \geq 1$. The *recursive algorithm* generates a tree layout of L_n according to the following two rules:

- If $k = 1$: form a tree layout by joining the four nodes of the mesh as shown in Figure 5(a).
- If $k > 1$: divide the mesh in four $L_{n/2}$ sub-meshes (top/left, bottom/left, top/right and bottom/left); recursively create a tree layout for each one of the sub-meshes; join the four tree layouts in one tree layout as shown in Figure 5(b).

Figure 6 illustrates the tree layouts, together with the different measures cost on them, produced by the recursive algorithm on L_2, L_4 and L_8 meshes. Observe that the recursive algorithm generates balanced tree layouts and produces a (2^{2k-1})-splitting edge, which we call the *top* edge, we will also assume that the so obtained tree layout is rooted at the top edge. The following lemma states the costs computed by the recursive algorithm.

Lemma 12. Let L_{2^k} be a $2^k \times 2^k$ mesh with $k \geq 1$. Let T_{2^k} be the tree layout of L_{2^k} computed by the recursive algorithm. Then,

$$\textsc{tc}(T_{2^k}, L_{2^k}) = 2^k, \qquad \textsc{td}(T_{2^k}, L_{2^k}) = 4k - 1,$$
$$\textsc{tcl}(T_{2^k}, L_{2^k}) = 2^{k+1}, \qquad \textsc{tl}(T_{2^k}, L_{2^k}) = 14 \cdot 4^k - 8 \cdot 2^k k - 15 \cdot 2^k.$$

Proof. The proof for TC and TCL is straightforward because the maximal edge and vertex congestion are reached at the top edge.

In order to compute $\textsc{td}(T_{2^k}, L_{2^k})$, observe that T_{2^k} is made of four tree layouts $T_{2^{k-1}}$ for which the distance from any leaf to their respective top edge is $2k-3$. But to construct T_{2^k} one node is inserted in these top edges and to connect a left sub-mesh with a right sub-mesh three edges are added. So, $\textsc{td}(T_{2^k}, L_{2^k}) = 2((2k-3)+1)+3 = 4k-1$.

In the following, let $f(k) = \textsc{tl}(T, L_{2^k})$, note that $f(1) = 10$. For $k \geq 2$, in order to compute $f(k)$ observe that T_{2^k} is made of four tree layouts $T_{2^{k-1}}$, each one containing $2^{k-1} \cdot 2^{k-1}$ nodes and whose height from the top edge is $2k-3$. We obtain the following recurrence:

$$\begin{cases} f(1) = 10, \\ f(k) = 4f(k-1) + 4 \cdot 2^{k-1} + 2 \cdot 2^{k-1}(2(2k-3)+4) + 2 \cdot 2^{k-1}(2(2k-3)+5). \end{cases}$$

The first term comes from the cost of the four recursive tree layouts; the second term comes from the lengthening of the four recursive tree layouts due to the addition of a new node on its top edge; the third term comes from the cost of the length of the vertical edges between the two top trees and the two bottom trees; the fourth term comes from the cost of the length of the horizontal edges between the two left trees and the two right trees.

The resolution of the recurrence yields the result. □

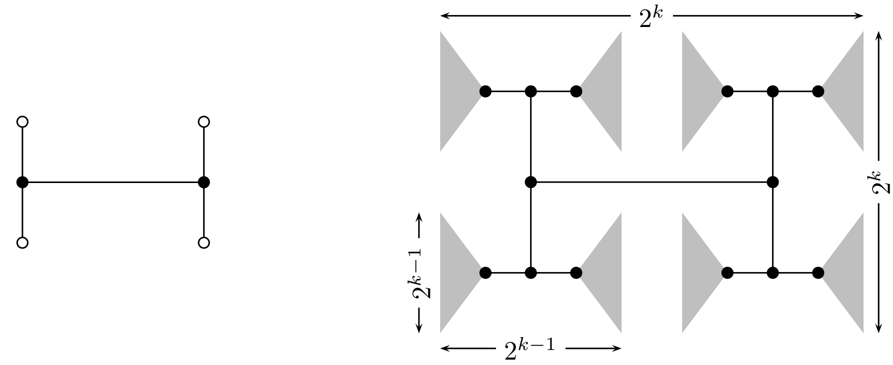
We now generalize the algorithm to handle $n \times n$ meshes, when n is not a power of two.

Definition 3 (The generalized recursive algorithm). Let L_n be a $n \times n$ mesh. Let k be the integer such that $n \leq 2^k < 2n$ and let T_{2^k} be the tree computed by the recursive algorithm on L_{2^k} rooted at the top edge. The *generalized recursive algorithm* generates a tree layout T_n of L_n applying iteratively the following transformation for all nodes $u \in V(L_{2^k}) \setminus V(L_n)$: let p_1 be the parent of u, let v be the sibling of u and let p_2 be the parent of p_1; remove the nodes u and p_1 from T together with its three incident edges; add the edge $p_2 v$ to T (see Figure 7).

The following theorem states that the generalized recursive algorithm is a constant approximation algorithm for our tree layout problems on meshes:

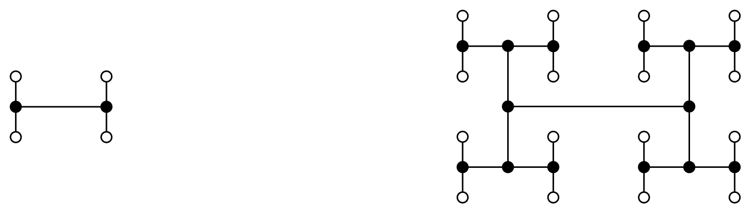
Theorem 3. For all n large enough, let L_n be a $n \times n$ mesh and let T_n be its tree layout computed by the generalized recursive algorithm. Then,

$$\frac{\textsc{td}(T_n, L_n)}{\textsc{mintd}(L_n)} < 5, \qquad \frac{\textsc{tc}(T_n, L_n)}{\textsc{mintc}(L_n)} < 4, \qquad \frac{\textsc{tvc}(T_n, L_n)}{\textsc{mintvc}(L_n)} < 12,$$
$$\frac{\textsc{tcl}(T_n, L_n)}{\textsc{mintcl}(L_n)} < 2\sqrt{6}, \qquad \frac{\textsc{tl}(T_n, L_n)}{\textsc{mintl}(L_n)} < 10.$$



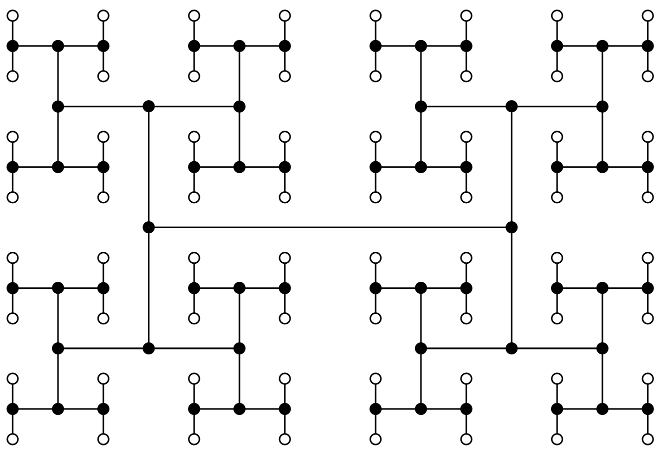
(a) Base case: $k = 1$ (b) Recursive case

Figure 5: Recursive algorithm to build a tree layout for a L_{2^k} mesh.



(a) L_2: TC = 2, TD = 3, TL = 10, TCL = 4.

(b) L_4: TC = 4, TD = 7, TL = 100, TCL = 8.



(c) L_8: TC = 8, TD = 11, TL = 584, TCL = 16.

Figure 6: Illustration of tree layouts computed by the recursive algorithm.

Proof. Let k be the integer such that $n \le 2^k < 2n$ and let T_{2^k} be the tree computed by the recursive algorithm on L_{2^k}. Observe that the iterative deletion of a leaf by the generalized recursive algorithm cannot increase the vertex or edge congestion at an edge of the tree layout. Also, the iterative deletion of a leaf by the generalized recursive algorithm cannot increase the length of a graph edge in the tree layout. Therefore, using Lemmas 12 and 2, we get

$$\begin{aligned}
\textsc{td}(T_n, L_n) &\le \textsc{td}(T_{2^k}, L_{2^k}) \le 4k - 1 < 4\log n + 3,\\
\textsc{tc}(T_n, L_n) &\le \textsc{tc}(T_{2^k}, L_{2^k}) \le 2^k < 2n,\\
\textsc{tvc}(T_n, L_n) &\le 3\textsc{tc}(T_{2^k}, L_{2^k}) \le 3.2^k < 6n,\\
\textsc{tcl}(T_n, L_n) &\le \textsc{tcl}(T_{2^k}, L_{2^k}) \le 22^k < 4n,\\
\textsc{tl}(T_n, L_n) &\le \textsc{tl}(T_{2^k}, L_{2^k}) \le 14 \cdot 4^k - 8 \cdot 2^k \cdot k - 15 \cdot 2^k \le 14 \cdot 4^k \le 56n^2.
\end{aligned}$$

The statement of the theorem follows from the previous upper bounds together with the lower bounds of Lemma 11. □

The generalized recursive algorithm can be extended, in a natural way, to multidimensional meshes. In such a case the approximation rate depends exponentially on the dimension of the mesh.

6 Random geometric graphs

Let r be a positive number and let V be any set of n points in the unit square ($[0,1]^2$). A *geometric graph* $G(V,r)$ with vertex set V and radius r is the graph $G = (V,E)$ where $E = \{uv \mid u, v \in V \wedge 0 < \|u - v\| \le r\}$. In the following, $\|\cdot\|$ denotes the l_∞ norm, but similar results can be obtained with any other l_p norm, $p > 0$.

Let $(r_i)_{i\ge1}$ be a sequence of positive numbers and let $X = (X_i)_{i\ge1}$ be a sequence of independently and uniformly distributed random points in $[0,1]^2$. For any natural n, we write $\mathcal{X}_n = \{X_1, \ldots, X_n\}$ and call $G(\mathcal{X}_n, r_n)$ a *random geometric graph* of n nodes on X. We denote by $\mathcal{G}(n; r_n)$ the distribution of random geometric graphs with n nodes and radius r_n.

In the remainder of this section we restrict our attention to the particular case where the radius is of the form

$$r_n = \sqrt{\frac{a_n}{n}} \quad \text{where} \quad r_n \to 0 \quad \text{and} \quad a_n/\log n \to \infty.$$

It is important to emphasize that through this choice, the construction of sparse yet connected graphs is guaranteed: By defining the connectivity distance ρ_n as the smallest radius r_n such that a random geometric graph is connected, it is known that $\rho_n\sqrt{n/\log n}$ converges to $\frac{1}{2}$ almost surely [2].

For a given graph G, let $\mathsf{cut}(i, G)$ be the minimum number of edges across a cut (A, B) of $V(G)$ in which $|A| = i$ and let $\mathsf{vertex\text{-}cut}(i, G)$ be the minimum number of nodes in a set $A \subseteq V(G)$, with $|A| = i$, which are connected to some vertex in $V(G) - A$. From [6, Lemma 5.2] and [6, Lemma 5.4] wet the following result,

Lemma 13. Let G_n denote a random geometric graph with n nodes drawn from the $\mathcal{G}(n;r_n)$ model. Then, with high probability,

$$\begin{aligned} \mathsf{cut}(i, G_n) &= \Omega(n^2 r_n^3) \\ \mathsf{vertex\text{-}cut}(i, G_n) &= \Omega(n r_n), \end{aligned}$$

for any i, $n/3 \le i \le 2n/3$.

The previous lemma together with Lemma 3 and the fact that a tree with $O(n)$ leaves has height $\Omega(\log n)$, we obtain the following lower bounds:

Lemma 14 (Lower bounds). Let G_n denote a random geometric graph with n nodes drawn from the $\mathcal{G}(n;r_n)$ model. Then, with high probability,

$$\begin{aligned} \textsc{mintd}(G_n) &= \Omega(\log n), & \textsc{mintc}(G_n) &= \Omega(n^2 r_n^3), \\ \textsc{mintcl}(G_n) &= \Omega(n r_n), & \textsc{mintl}(G_n) &= \Omega(n^2 r_n^2 \log n). \end{aligned}$$

We introduce now a subclass of geometric graphs that captures the properties we need to bound our tree layout costs on random geometric graphs.

Definition 4 (Well behaved graphs). Consider any set $\mathcal{X}_n$ of n points in $[0,1]^2$, which together with a radius r_n, induce a geometric graph $G = G(\mathcal{X}_n, r_n)$. Dissect the unit square into $4\lfloor 1/r_n \rfloor^2$ boxes of size $1/(2\lfloor 1/r_n \rfloor) \times 1/(2\lfloor 1/r_n \rfloor)$ placed packed in $[0,1]^2$ starting at $(0,0)$. By construction, all the boxes exactly fit in the unit square, and any two points of $\mathcal{X}_n$ connected by an edge in G will be in the same or neighboring boxes (including diagonals) because $1/(2\lfloor 1/r_n \rfloor) \ge r_n/2$. Given $\epsilon \in (0,1)$, let us say that G is *ϵ-well behaved* if every box of this dissection contains at least $(1-\epsilon)\frac{1}{4}a_n$ points and at most $(1+\epsilon)\frac{1}{4}a_n$ points (see Fig. 8).

Our interest in well behaved graphs is motivated by the fact that, with high probability, random geometric graphs are well behaved,

Lemma 15. Let $\epsilon \in (0, \frac{1}{5})$. Then, with high probability, a random geometric graphs drawn from $\mathcal{G}(n;r_n)$ is ϵ-well behaved.

Proof. Choose a box in the dissection and let Y be the random variable counting the number of points of $\mathcal{X}_n$ in this box. As the points in $\mathcal{X}_n$ are independently uniformly distributed,

$$\mathbf{E}[Y] = n/\left(4\lfloor 1/r_n \rfloor^2\right) \sim \tfrac{1}{4} n r_n^2 = \tfrac{1}{4} a_n.$$

Let $b_n = a_n/\log n$; by hypothesis, we have $b_n \to \infty$. Using Chernoff's bounds [7, 18], we get

$$\begin{aligned} \mathbf{Pr}\left[Y \ge (1+\epsilon)\tfrac{1}{4}a_n\right] &\le \mathbf{Pr}\left[Y \ge (1+\tfrac{1}{2}\epsilon)\mathbf{E}[Y]\right] \le \exp\left(-\left(\tfrac{1}{2}\epsilon\right)^2 \mathbf{E}[Y]/3\right) \\ &\le \exp\left(-\tfrac{1}{13}\epsilon^2 \tfrac{1}{4} a_n\right) = n^{-\epsilon^2 b_n/52} \end{aligned}$$

and

$$\begin{aligned} \mathbf{Pr}\left[Y \le (1-\epsilon)\tfrac{1}{4}a_n\right] &\le \mathbf{Pr}\left[Y \le (1-\tfrac{1}{2}\epsilon)\mathbf{E}[Y]\right] \le \exp\left(-\left(\tfrac{1}{2}\epsilon\right)^2 \mathbf{E}[Y]/2\right) \\ &\le \exp\left(-\tfrac{1}{9}\epsilon^2 \tfrac{1}{4} a_n\right) = n^{-\epsilon^2 b_n/36} \le n^{-\epsilon^2 b_n/52}. \end{aligned}$$

The number of boxes is certainly smaller than n, so by Boole's inequality, the probability that for some box the number of points in the box is less than $(1-\epsilon)\frac{1}{4}a_n$ or bigger than $(1+\epsilon)\frac{1}{4}a_n$, is bounded by $2n^{1-b_n\epsilon^2/52}$, which tends to 0 as $n \to \infty$. $\square$

We present now a modification to the recursive algorithm to handle geometric graphs. See Figure 8 for a representation of the algorithm.

Definition 5 (The boxed recursive algorithm). Let G be a geometric graph with n nodes and radius r_n. Dissect the unit square into $4\lfloor 1/r_n\rfloor^2$ boxes of size $1/(2\lfloor 1/r_n\rfloor) \times 1/(2\lfloor 1/r_n\rfloor)$ placed packed in $[0,1]^2$ starting at $(0,0)$. The *boxed recursive algorithm* generates a tree layout T of G in the following way:

- All points in the same box are the leaves of a balanced tree layout.
- The generalized recursive tree layout is used to form a tree layout for all the graph, taking as its leaves a node that is inserted at the top edge of each of the balanced trees for each box.

The following lemma presents upper bounds on the cost of tree layout problems on well behaved graphs that match the lower bounds. The proof uses the boxed recursive algorithm.

Lemma 16 (Upper bounds). Let $\epsilon \in (0,1)$ and n large enough. Let G_n denote any ϵ-well behaved geometric graph with n nodes and radius r_n and let T_n be the tree layout computed by the boxed recursive algorithm for G_n. Then,

$$\textsc{td}(T_n, G_n) = O(\log n), \qquad \textsc{tc}(T_n, G_n) = O(n^2 r_n^3),$$
$$\textsc{tcl}(T_n, G_n) = O(n r_n), \qquad \textsc{tl}(T_n, G_n) = O(n^2 r_n^2 \log n).$$

Proof. As in the case of the mesh, it is easy to see that that the maximal edge and vertex congestion is located at the top of T_n. In this place we have an edge which hosts the edges of two rows of $\sqrt{n/a_n}$ boxes, each with at most $(1+\epsilon)a_n$ points and connected to at most 3 neighbors. So, we have

$$\textsc{tc}(T_n, G_n) \leq 3 \cdot (1+\epsilon)a_n^2 \cdot 2\sqrt{n/a_n} = O\left(a_n\sqrt{a_n n}\right) = O(n^2 r_n^3)$$

and

$$\textsc{tcl}(T_n, G_n) \leq 2 \cdot \sqrt{n \cdot a_n} = O(n r_n).$$

The diameter of the tree layout T obtained by the boxed recursive algorithm is bounded from above by $\lceil \log((1+\epsilon)a_n\rceil + 1 + \lceil \log(4\lfloor 1/r_n^2\rfloor)\rceil = O(\log n)$. So, applying Lemma 1, we get that $\textsc{td}(T_n, G_n) = O(\log n)$.

According to the boxed recursive algorithm, we can analyze the congestion of the edges that appear at each level of the mesh-like construction. The total number of levels is $l = \log\sqrt{4\lfloor 1/r\rfloor^2}$. Let us define h_i as the height of the subtree at level i. We have $h_0 = \log((1+\epsilon)\frac{1}{4}a_n)$ and $h_{i+1} = h_i + 2$. Let t_i be the contribution of the edges taken into account in level i. We have $t_0 = ((1+\epsilon)\frac{1}{4}a_n)^2\, h_0\, 4\lfloor 1/r_n\rfloor^2$ and $t_{i+1} = 4 \cdot 8 \cdot 2^i\,((1+\epsilon)\frac{1}{4}a_n)^2\, h_{i+1}\, 4\lfloor 1/r_n\rfloor^2\, 4^{1-i}$. Summing up $\sum_{i=1}^{l} t_i$, we get that $\textsc{tl}(T_n, G_n) = O(n^2 r_n^2 \log n)$ as claimed. □

The combination of Lemmas 14, 15, 16, and 2 leads to our main result on tree layouts for random geometric graphs:

Theorem 4. With high probability, the problems MinTD, MinTC, MinTVC, MinTCL and MinTL can be approximated within a constant on random geometric graphs $\mathcal{G}(n; r_n)$ with $r_n = \sqrt{a_n/n}$, $r_n = o(1)$ and $a_n = \omega(\log n)$ using the tree layout computed by the boxed recursive algorithm.

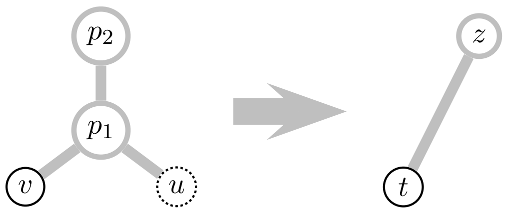


Figure 7: Deleting a leaf after the application of the generalized recursive algorithm.

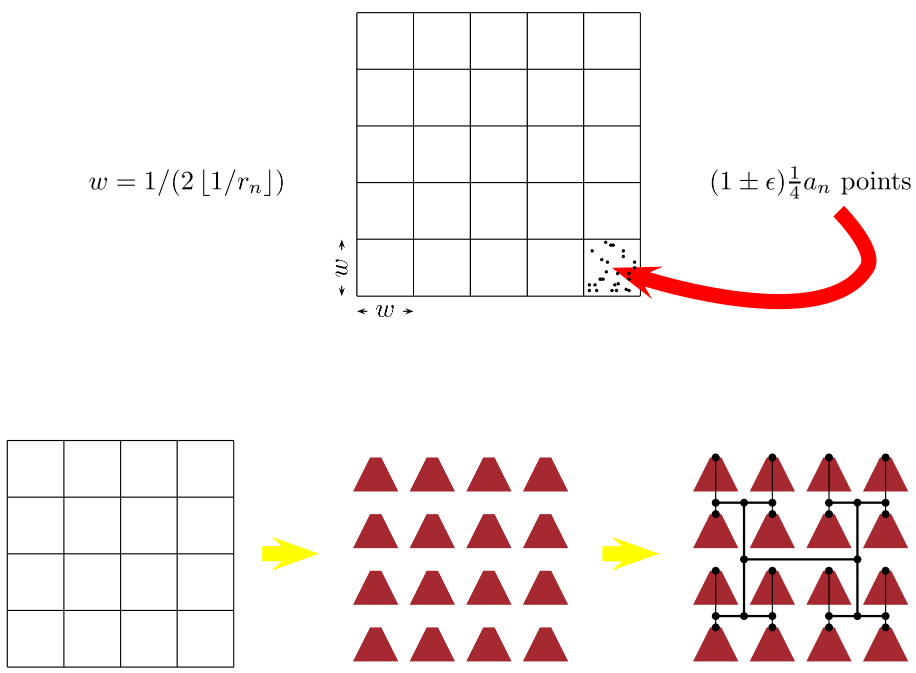


Figure 8: Illustration of tree layouts computed by the boxed algorithm.

7 Approximating problems on communication trees with higher degree

So far we have studied the communication parameters when the proposed communication network is a tree layout with internal nodes of degree 3, but we may also consider the construction of routing or communication trees when the tree has maximum degree $d \geq 3$.

As any tree layout is a routing tree and every routing tree is a communication tree, we trivially have:

Lemma 17. For any measure $\textsc{f} \in \{\textsc{td}, \textsc{tc}, \textsc{tvc}, \textsc{tcl}, \textsc{tl}\}$, any $d \geq 3$ and any graph G,

$$d_{\leq}\text{-}\textsc{mincf}(\mathrm{G}) \leq d_{\leq}\text{-}\textsc{minrf}(\mathrm{G}) \leq d\text{-}\textsc{minrf}(\mathrm{G}).$$

Taking into account that any communication/routing tree T is acceptable for any $d \geq \Delta(T)$ and that the degree of any internal vertex in a communication/routing tree can be increased by connecting it to a new leaf, without changing the cost we have:

Lemma 18. For any measure $\textsc{f} \in \{\textsc{td}, \textsc{tc}, \textsc{tvc}, \textsc{tcl}, \textsc{tl}\}$, any $d \geq 3$ and any graph G we have

$$\begin{aligned}
(d+1)_{\leq}\text{-}\textsc{mincf}(\mathrm{G}) &\leq d_{\leq}\text{-}\textsc{mincf}(\mathrm{G}), \\
(d+1)_{\leq}\text{-}\textsc{minrf}(\mathrm{G}) &\leq d_{\leq}\text{-}\textsc{minrf}(\mathrm{G}), \quad \text{and} \\
(d+1)\text{-}\textsc{minrf}(\mathrm{G}) &\leq d\text{-}\textsc{minrf}(\mathrm{G}).
\end{aligned}$$

The next theorem relates the problems on tree layouts and communication trees.

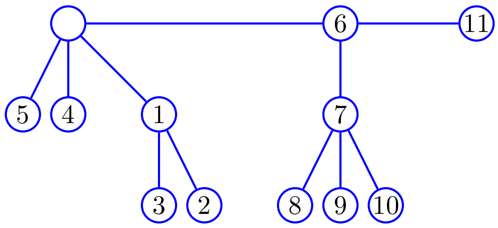
Lemma 19. For any $d \geq 3$ and any non-empty graph G we have

$$\begin{aligned}
\textsc{mintd}(G) &\leq c_1 \cdot d\text{-}\textsc{minctd}(\mathrm{G}), \\
\textsc{mintc}(G) &\leq c_2 \cdot d\text{-}\textsc{minctc}(\mathrm{G}), \\
\textsc{mintvc}(G) &\leq c_3 \cdot d\text{-}\textsc{minctvc}(\mathrm{G}), \\
\textsc{mintcl}(G) &\leq c_4 \cdot d\text{-}\textsc{minctcl}(\mathrm{G}), \\
\textsc{mintl}(G) &\leq c_5 \cdot d\text{-}\textsc{minctl}(\mathrm{G}).
\end{aligned}$$

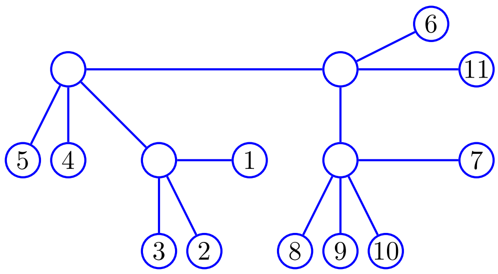
where $c_1, \ldots, c_5$, are constants that only depend on d.

Proof. The proof considers a way of constructing a tree layout starting from a communication tree. In such a construction we add new vertices an leaves to the original tree in such a way that at the end of the process the tree has enough labeled leaves to keep all the vertices in in the graph. In Figure 9 we give a sketch of the construction. Assume that T is a communication tree for G with $\Delta(T) \leq d$, construct the routing tree T' in the following way: for every vertex $x \in V(G)$, let $t(x)$ be the corresponding node in T. If $t(x)$ is not a leaf add a new leaf connected to $t(x)$ with label x. If $t(x)$ is a leaf assign label x to $t(x)$. The resulting tree is a routing tree T' for G with $\Delta(T') \leq d+1$. Then we have,

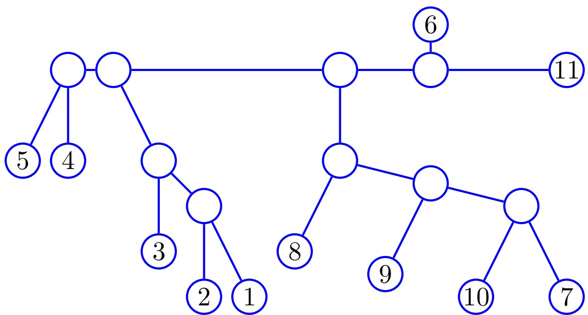
$$\begin{aligned}
\textsc{td}(T', G) &\leq \textsc{td}(T, G) + 2 \leq 3 \cdot \textsc{td}(T, G), \\
\textsc{tc}(T', G) &\leq d \cdot \textsc{tc}(T, G), \\
\textsc{tvc}(T', G) &\leq \textsc{tvc}(T, G), \\
\textsc{tcl}(T', G) &\leq d \cdot \textsc{tcl}(T, G), \\
\textsc{tl}(T', G) &\leq \textsc{tl}(T, G) + 2|E(G)| \leq 3 \cdot \textsc{tl}(T, G).
\end{aligned}$$



(a) 4-communication tree



(b) Intermediate 5-routing tree



(c) Final tree layout

Figure 9: Transforming a 4-communication tree into a tree layout.

Observe that the edge congestion bound follows from the fact that after adding a leaf labeled with a vertex in G associated to an internal vertex, the congestion of the edge connecting this leaf cannot be more than the congestion of d edges in the initial tree. The same argument is used for the bound on the communication load.

To reduce the degree in the routing tree T' we construct a new routing tree T'' for G, as follows: replace every node t in $V(T')$ with degree $\alpha > 3$ by a worm layout with α leaves. Notice that as $\alpha \leq d$ the number of nodes in such a caterpillar is a constant which is different for each value of d. Then we have

$$\begin{aligned}
\text{TD}(T'', G) &\leq d \cdot \text{TD}(T', G), \\
\text{TC}(T'', G) &\leq (d-2) \cdot \text{TC}(T', G), \\
\text{TVC}(T'', G) &\leq \text{TVC}(T', G), \\
\text{TCL}(T'', G) &\leq (d-2) \cdot \text{TCL}(T', G), \\
\text{TL}(T'', G') &\leq d \cdot \text{TL}(T', G).
\end{aligned}$$

As T'' is a tree layout for G the result follows, considering as starting point an optimal tree according to each of the measures. □

The previous lemmas say that if we can construct a tree layout with a guaranteed cost, we can construct a routing/communication tree with the required degree bounds, adding a set of new leaves connected to the internal vertices, with the same cost. And this is independent of the measure. Therefore, combining Theorems 2, 3 and 4 we get constant approximability algorithms for the communication/routing tree problems with higher degrees.

Theorem 5. Let $\epsilon \in (0, \frac{1}{9})$, $\gamma \in (0,1)$ and define $C_{\epsilon,\gamma} = 3(1+\ln 3)(\epsilon\gamma)^{-2}$. Consider a sequence $(c_n)_{n\geq 1}$ such that $C_{\epsilon,\gamma} \leq c_n \leq n$ for all $n \geq n_0$ for some natural n_0 and let $p_n = c_n/n$. Then, for any $d \geq 3$, and for any measure $\text{F} \in \{\text{TD}, \text{TC}, \text{TVC}, \text{TCL}, \text{TL}\}$, the problems $d_{\leq}$-MINCF, $d_{\leq}$-MINRF and d-MINRF can be approximated within a constant on random graphs $\mathcal{G}_{n,p_n}$ with overwhelming probability.

Theorem 6. For any $d \geq 3$ and for any measure $\text{F} \in \{\text{TD}, \text{TC}, \text{TVC}, \text{TCL}, \text{TL}\}$, and for all n large enough, the problems $d_{\leq}$-MINCF, $d_{\leq}$-MINRF and d-MINRF can be approximated asymptotically on square meshes.

Theorem 7. For any $d \geq 3$ and for any measure $\text{F} \in \{\text{TD}, \text{TC}, \text{TVC}, \text{TCL}, \text{TL}\}$, and for all n large enough, the problems $d_{\leq}$-MINCF, $d_{\leq}$-MINRF and d-MINRF can be approximated, with high probability, within a constant on random geometric graphs $\mathcal{G}(n; r_n)$ with $r_n = \sqrt{a_n/n}$, $r_n = o(1)$ and $a_n = \omega(\log n)$.

8 Open problems

Table 1 summarizes the complexity status of the communication tree problems. So far we have not been able to prove the NP-completeness of any of the open problems. For the case of the vertex congestion tree layout problem, the first idea is to mimic the reduction presented in [20], the main difficulty relies in the fact that for a complete graph, the vertex congestion of balanced is not optimal. However we conjecture that all the problems in table 1, with unknown classification are NP-complete.

	Communication tree	Tree layout
Dilation	NPC for trees when $d = 3$ [17]	Open
Congestion	NPC when G is planar P if G is outerplanar for any $d \geq 3$ [21]	NPC P if G is planar for $d \geq 3$ [20]
V. Congestion	Open	Open
Comm. Load	Open	Open
Length	Open	Open

Table 1: Complexity status for communication tree problems

It is also of interest to find graph classes for which polynomial time or approximate algorithm exist and the study of the problem on other graph classes, in particular for the class of unit disk graphs that is closely related to random geometric graphs.

Acknowledgements

We thank two anonymous referees for giving useful information to improve the paper.

References

[1] S. Agarwal, A. K. Mittal, and P. Sharma. Constrained optimum communications trees and sensitivity analysis. *SIAM Journal on Computing*, 13:315–328, 1984.

[2] M. J. Appel and R. P. Russo. The connectivity of a graph on uniform points in $[0, 1]^d$. Technical Report (preprint), University of Iowa, 1996.

[3] B. Bollobas. *Random Graphs.* Academic Press, London, 1985.

[4] B. Bollobas and I. Leader. Compressions and isoperimetric inequalities. *Journal of Combinatorial Theory Series A*, 56:47–62, 1991.

[5] T. Cormen, Leisserson, Rivest, C. Stein . *Introduction to algorithms* 2nd edition. MIT Press, 2002.

[6] J. Díaz, M. D. Penrose, J. Petit, and M. Serna. Approximating Layout Problems on Random Geometric Graphs *Journal of Algorithms*, 39:78–116, 2001. Science, June 1999.

[7] J. Díaz, J. Petit, and M. Serna. A guide to concentration bounds. In Handbook of Randomized Computing. Vol II. S. Rajasekaran, P. Pardalos, J. H. Reif and J. Rolim, Eds. Kluwer, 2001.

[8] J. Díaz, J. Petit, and M. Serna. A Survey of Graph Layout Problems. *ACM Computing Surveys*, 34(3):313-356, 2002

[9] J. Díaz, J. Petit, M. Serna, and L. Trevisan. Approximating layout problems on random graphs. *Discrete Mathematics. Elsevier North–Holland*, 235:245–253, 2001.

[10] G. Ding and B. Oporowski. Some results on tree decomposition of graphs. *Journal of Graph Theory*, 20(4):481–499, 1995.

[11] M. R. Garey and D. S. Johnson. *Computers and Intractability. A guide to the theory of NP-completeness.* Freeman and Company, 1979.

[12] T. C. Hu. Optimal communication spanning tree. *SIAM Journal on Computing*, (3):189–195, 1974.

[13] D. S. Johnson, J. K. Lenstra, and A. H. G. R. Kan. The complexity of the network design problem. *Networks*, (8):279–285, 1978.

[14] D. E. Knuth. *The Art of Computer Programming: Fundamental Algorithms*, volume 1. Addison-Wesley, Reading, Mass., 1968.

[15] Y. L. Lai and K. Williams. A Survey of Solved Problems and Applications on Bandwidth, Edgesum and Profile of Graphs. *Journal of Graph Theory*, pages 75–94, 1999.

[16] R. J. Lipton and R. E. Tarjan. A separator theorem for planar graphs. *SIAM Journal on Applied Mathematics*, 36:177–189,1979.

[17] B. Monien. The complexity of embedding graphs into binary trees. In *Fundamentals of computation theory (Cottbus, 1985)*, pages 300–309. Springer, Berlin, 1985.

[18] R. Motwani and P. Raghavan. *Randomized Algorithms.* Cambridge University Press, 1995.

[19] M. Penrose. *Random Geometric Graphs.* Oxford University Press, 2004.

[20] P. D. Seymour and R. Thomas. Call routing and the ratcatcher. *Combinatorica*, 14(2):217–241, 1994.

[21] S. Simonson. A variation on the min cut linear arrangement problem. *Math. Systems Theory*, 20(4):235–252, 1987.