Unconditional Lower Bounds on the Time-Approximation Tradeoffs for the Distributed Minimum Spanning Tree Problem * †

[Extended Abstract]

Michael Elkin
Yale University, Department of Computer Science New Haven, CT, 06512.
elkin@cs.yale.edu

ABSTRACT

The design of distributed *approximation* protocols is a relatively new rapidly developing area of research. However, so far little progress was done in the study of the *hardness of distributed approximation*.

In this paper we initiate the systematic study of this subject, and show strong *unconditional lower bounds on the time-approximation tradeoff* of the distributed *minimum spanning tree* problem, and some of its variants.

Categories and Subject Descriptors

F.2.3 [**Analysis of Algorithms and Problem Complexity**]: Tradeoffs between Complexity Measures

General Terms

Theory

Keywords

Minimum Spanning Tree, Hardness of Approximation

1. INTRODUCTION

1.1 Distributed Computing

Consider a *synchronous* network of processors with *unbounded computational power*, modeled by an n-vertex graph. The initial knowledge of the processors (henceforth, vertices) is very limited. Specifically, each of them has its own *local* perspective of the network (henceforth, graph), which is confined to its immediate neighborhood. The vertices, however, have to compute some *global* function of the graph, such as its *minimum spanning tree* (henceforth, MST).

For this end distributed algorithms (henceforth, *protocols*) are designed. There are several measures of *efficiency* of protocols, but in this paper we restrict our attention to one of them, called the *running time*, which is defined as the number of rounds of distributed communication. On each round of communication at most B bits can be sent through each edge, and B is a parameter of the model. The running time efficiency measure of protocols naturally gives rise to a complexity measure of problems, called *time complexity*.

The design of efficient protocols for this model, as well as proving lower bounds on their efficiency, is a vivid area of study known as *locality-sensitive distributed computing* (henceforth, *distributed computing*) (see [24] and the references therein).

1.2 Distributed Approximation and Hardness of Approximation

While traditionally the research in the area of distributed computing concentrated on designing protocols that solve the problem at hand *exactly*, some of the more recent research focuses on providing *approximate* solutions for various distributed problems. Most notably, several approximation protocols were recently devised for the minimum dominating set problem [18, 9, 21], and for the minimum edge-coloring problem [26, 10, 17, 4]. Also, an approximation protocol for the maximum matching problem was recently devised in [7].

However, the situation with lower bounds on approximability of distributed problems is by far less satisfactory. Specifically, the existing results on hardness of distributed approximation can be divided to two categories.

First, there are inapproximability results that are based on lower bounds on the time required for *exact* solution of certain problems, and on integrality of the objective functions of these problems. For example, there is a classical result due to Linial [22] saying that 3-coloring an n-vertex ring requires $\Omega(\log^* n)$ time. In particular, it implies that any 3/2-approximation protocol for vertex-coloring problem requires $\Omega(\log^* n)$ time.

Second, there are inapproximability results that assume that the vertices are computationally limited, e.g., are allowed to perform at most polynomial in n number of oper-

*Part of this work was done in the Institute for Advanced Study, School of Mathematics, Princeton, NJ, 08520.

†This work was supported by the DoD University Research Initiative (URI) administered by the Office of Naval Research under Grant N00014-01-1-0795.

STOC'04 June 13-15, 2004, Chicago, Illinois, USA.

ations. Obviously, under this assumption any NP-hardness inapproximability result immediately gives rise to an analogous result in the distributed model.

Note, however, that neither of these inapproximability results sheds a new light on our understanding of the limitations of distributed computing. Specifically, the results of this sort are just somewhat different *semantic interpretations* of already known lower bounds. Additionally, we believe that imposing restrictions on the computational power of the vertices is as unnatural as limiting the computational power of the parties in the two-party communication complexity model (see [19]). In both cases the abstraction of computationally unbounded vertices or parties is necessary to make possible the study of the role that *communication* plays in computation (see also [22, 24]).

To summarize, while sophisticated distributed approximation protocols were developed for various problems, so far no real progress was made in the study of the *hardness of distributed approximation*. In this paper we initiate the systematic study of this subject. Specifically, we study the inapproximability of the distributed *MST* problem, and show *strong unconditional lower bounds on the time-approximation tradeoffs* for this problem and some of its variants.

1.3 Distributed *MST* Problem

The (distributed) *MST* problem is one of the most important problems in the area of distributed computing, and was subject of extensive research [13, 8, 16, 1, 14, 20, 25, 23, 11].

The most time-efficient protocol known for this problem is due to Elkin [11], and its running time is $O(\mu(G,\omega)\cdot\log^3 n + \sqrt{\frac{n\log^* n}{B}}\log n)$, where $\mu(G,\omega)$ stands for the *MST-radius* of the weighted graph (G,ω). The definition of the *MST*-radius $\mu(G,\omega)$ is somewhat involved (see [11]), but for the rest of this discussion it is sufficient to keep in mind that for any graph (G,ω), $\mu(G,\omega) \le \Lambda(G) \le n$, where $\Lambda(G)$ stands for the *unweighted diameter* of the graph G.

On the negative side, Peleg and Rubinovich [25] have shown a lower bound of $\Omega(\frac{\sqrt{n}}{B})$ on the time complexity of the *MST* problem restricted to graphs of small diameter (at most $O(n^\delta)$ for arbitrarily small positive $\delta > 0$).

In this paper we show that *approximating*[1] the *MST* problem within a ratio H on graphs of small diameter requires $T = \Omega(\sqrt{\frac{n}{H\cdot B}})$ time. In other words, we derive an *unconditional lower bound on the time-approximation tradeoff* for the *MST* problem, specifically, $T^2\cdot H = \Omega(\frac{n}{B})$. Substituting $H = O(1)$ into this formula shows that approximating the *MST* problem within *any constant factor* requires $\Omega(\sqrt{\frac{n}{B}})$ time, improving the lower bound of [25] by a factor of $\sqrt{B}$ (recall that the lower bound of [25] applies only for the *exact* solution of the *MST* problem).

Moreover, our lower bound implies that for any $0 < \epsilon < 1$, approximating the *MST* problem within a factor of $\left(\frac{n}{B}\right)^{1-\epsilon}$ requires $\Omega\left(\left(\frac{n}{B}\right)^{\epsilon/2}\right)$ time. The latter means, in particular, that the $\left(\frac{n}{B}\right)^{1-\epsilon}$-approximate *MST* problem is *not a local* problem, i.e., cannot be solved in time polylogarithmic in n. This lower bound, like all the other lower bounds that we prove in this paper, applies even to *randomized*[1] protocols.

We remark that lower bounds on time-approximation tradeoffs are rather rare even outside the area of distributed computing. In the standard Turing machine model of computation, a huge body of research on hardness of approximation is being conducted (see, e.g., [3] and the references therein). However, it seems that the only lower bounds on time-approximation tradeoffs in this model are implicit. One example is the maximum clique problem, where strengthening the complexity-theoretic assumptions leads to stronger lower bounds [12]. Implicit lower bounds on time-approximation tradeoffs were also shown by Chakrabarti et al. [6] in the *cell-probe* model of computation for the nearest-neighbor problem. Finally, in the context of *property testing* such lower bounds are explicit, but it should be noticed that in that context a different notion of approximation is employed.

1.4 Additional Results

One direction of recent research on the distributed *MST* problem was to refine the lower bound of Peleg and Rubinovich [25] that applies to the *exact* computation of the *MST* on graphs G with diameter $\Lambda(G) = O(n^\delta)$, $0 < \delta < 1/2$, and to prove similar lower bounds for the *MST* problem restricted to graphs of even smaller diameter. Specifically, Peleg and Rubinovich themselves [25] have shown a lower bound of $\Omega(\frac{\sqrt{n}}{B\cdot\log n})$ for the *MST* problem restricted to graphs G of diameter $\Lambda(G) = O(\log n)$, and Lotker et al. [23] have shown lower bounds of $\Omega(\frac{n^{1/3}}{B})$ (respectively, $\Omega(\frac{n^{1/4}}{B})$) for the *MST* problem restricted to graphs of diameter $\Lambda(G) \le 4$ (resp., $\Lambda(G) \le 3$). Recall that all these lower bounds apply only to the *exact MST* problem.

In addition to the lower bound on the time-approximation tradeoff for the general variant of the *MST* problem, we also show a lower bound on the time-approximation tradeoff for the *MST* problem restricted to graphs of diameter $\Lambda(G) \le \Lambda$, for $\Lambda = 3$ and all even Λ in the range $4 \le \Lambda = O(\log n)$. Specifically, denoting the running time of an approximation protocol by T, and its approximation ratio by H, we show that $T^{2+\frac{2}{\Lambda-2}}\cdot H = \Omega(\frac{n}{\Lambda\cdot B})$.

Note that this result improves all the previous lower bounds for the *exact* computation of the *MST*. Specifically, it improves the result of [25] for $\Lambda = O(\log n)$ by a factor of $\sqrt{B\cdot\log n}$, and the results of [23] for $\Lambda = 4$ (resp., $\Lambda = 3$) by a factor of $B^{2/3}$ (resp., $B^{3/4}$). Moreover, our result gives rise to a lower bound of $\Omega((\frac{n}{B})^{1/2-\epsilon})$ for the exact computation (or even approximation within *any constant factor*) of the *MST* on graphs of constant diameter $O(1/\epsilon)$, significantly improving the previously best-known lower bound of $\Omega(\frac{n^{1/3}}{B})$. Table 1 summarizes the previously known lower bounds on the time complexity of the *MST* problem restricted to graphs of diameter at most Λ, parameterized by Λ, along with our improved lower bounds on this problem.

On the positive side, we devise an H-approximation protocol for the *MST* problem with running time $O(\Lambda(G) + \frac{\omega_{max}}{H-1}\cdot\log^* n)$, where ω_{max} is the ratio between the maximal and the minimal weight of an edge in the input graph (G,ω). It follows that the approximate *MST* problem becomes easy when ω_{max} is small (our lower bounds on the H-approximate *MST* problem apply for $\omega_{max} = \Omega(\sqrt{n}\cdot H^{3/2})$).

Structure of the paper: Our main result (the lower bound

[1]See Section 2 for the formal definitions of the notions of *approximation* and *randomization* in this context.

Λ	Lower bound on the exact computation	Our improved lower bound on the exact computation	Our lower bound on the time-approximation tradeoff
n^{δ}, $0 < \delta < 1/2$	$\Omega(\frac{\sqrt{n}}{B})$ [25]	$\Omega(\sqrt{\frac{n}{B}})$	$T^2 \cdot H = \Omega(\frac{n}{B})$
$\Theta(\log n)$	$\Omega(\frac{\sqrt{n}}{B \cdot \log n})$ [25]	$\Omega(\sqrt{\frac{n}{B \cdot \log n}})$	$T^2 \cdot H = \Omega(\frac{n}{B \cdot \log n})$
Constant (at least 3)	$\Omega(\frac{n^{1/3}}{B})$ [23]	$\Omega((\frac{n}{B})^{\frac{1}{2} - \frac{1}{2\Lambda - 2}})$	$T^{2+\frac{2}{\Lambda-2}} \cdot H = \Omega(\frac{n}{B \cdot \Lambda})$
4	$\Omega(\frac{n^{1/3}}{B})$ [23]	$\Omega((\frac{n}{B})^{1/3})$	$T^3 \cdot H = \Omega(\frac{n}{B})$
3	$\Omega(\frac{n^{1/4}}{B})$ [23]	$\Omega((\frac{n}{B})^{1/4})$	$T^4 \cdot H = \Omega(\frac{n}{B})$

Table 1: The summary of previously known and new lower bounds on the *MST* problem restricted to graphs of diameter at most Λ.

of $T^2 \cdot H = \Omega(\frac{n}{B})$ on the time-approximation tradeoff for the general variant of the *MST* problem) is proved in Section 3. In Section 4 we describe our approximation protocol for the *MST* problem.

2. PRELIMINARIES

For a graph $G = (V, E)$, a *spanning tree* is an acyclic connected subgraph $\tau = (V, E')$, $E' \subseteq E$. For a weighted graph $(G = (V, E), \omega)$ with a non-negative weight function $\omega : E \to R$, a *minimum spanning tree (MST)* is a spanning tree $\tau = (V, E')$ with minimum weight $\omega(\tau) = \sum_{e \in E'} \omega(e)$.

An *H-approximate MST* τ for a graph (G, ω) is a spanning tree of weight that is at most H times greater than the weight of the *MST* of the graph (G, ω). A protocol Π is said to be an *H-approximation* for the *MST* problem if for every input graph (G, ω) it outputs an H-approximate *MST* τ.

Our lower bounds apply to *randomized* protocols with bounded worst-case running time. In other words, these protocols necessarily terminate within specified time bounds, but they are allowed to err with some constant probability $0 < q < 1/2$. Two possible types of error are allowed. First, a protocol may produce a subgraph of the input graph (G, ω) that is not an H-approximate *MST* of (G, ω). This subgraph may contain cycles or multiple connectivity components. Secondly, the protocol may return $\perp$, indicating that it failed to compute the correct answer.

For a pair of vertices $u, w \in V$, we denote by $dist_G(u, w)$ the *unweighted distance* between u and w in the graph $G = (V, E)$.

3. LOWER BOUNDS ON THE TIME-APPROXIMATION TRADEOFFS

3.1 The *CorruptedMail* Problem

In this section we show a lower bound on the tradeoff between the possible approximation ratio for the *MST* problem and the running time of a distributed protocol that may achieve this approximation ratio.

We start with describing the family of graphs that will be used in the proof of our lower bounds.

For a sufficiently large positive integer n, let Γ, m and p be positive integer parameters that satisfy $p \leq \log n$ and $(m+1)\Gamma + \frac{(m+1)^{1+1/p} - 1}{(m+1)^{1/p} - 1} = n$. Let $d = (m+1)^{1/p}$ (assume that d is integer; non-integrality issues affect only lower-order terms of our results, and are, therefore, ignored).

Consider a family $\mathcal{G}$ of graphs that contains one unweighted n-vertex graph $G_n = (V_n, E_n)$ for infinitely many positive integers n. The vertex set V_n is comprised of Γ vertex-disjoint paths $P_1, P_2, \ldots, P_\Gamma$ with $m+1$ vertices each, and a d-regular tree τ of depth p with its own vertex set $V(\tau)$ (that is disjoint from $\bigcup_{i=1}^{\Gamma} V(P_i)$). Observe that $|V(\tau)| = 1 + d + \ldots + d^p = \frac{d^{p+1} - 1}{d - 1} = \frac{(m+1)^{1+1/p} - 1}{(m+1)^{1/p} - 1}$. Let rt be the root of τ, i.e., the only vertex that had degree d in τ; all the other vertices have either d children and a parent, or they have only a parent. The latter vertices are called *leaves* of τ. Let $s = z_0, z_1, \ldots, z_m = r$ denote the leaves of τ. The edge set of the graph G_n consists of the edge set $E(\tau)$ of the tree τ, the edge set $\bigcup_{j=1}^{\Gamma} E(P_j)$ of the Γ paths $P_1, P_2, \ldots, P_\Gamma$, and $m+1$ stars S_i, $i = 0, 1, \ldots, m$. Let $P_j = (v_j^{(0)}, v_j^{(1)}, \ldots, v_j^{(m)})$, $j = 1, 2, \ldots, \Gamma$, denote the vertices of the path P_j, $j = 1, 2, \ldots, \Gamma$. Then the edge set of the star S_i, for $i = 0, 1, \ldots, m$, is the set $\{(z_i, v_j^{(i)}) \mid j = 1, 2, \ldots, \Gamma\}$ of edges. (See Figure 1.)

Family $\mathcal{G}^{\omega}$ of *weighted* graphs contains 2^{Γ} n-vertex graphs for each n such that $G_n \in \mathcal{G}$. Each of these 2^{Γ} graphs has the vertex set V_n and the edge set E_n, but the weights of edges are different for at least one edge in any two distinct n-vertex graphs of $\mathcal{G}^{\omega}$. The edges of the paths $P_1, P_2, \ldots, P_\Gamma$, as well as the edges of the tree τ, are all of weight zero in all 2^{Γ} graphs. The edges of the stars S_i for $i = 1, 2, \ldots, m-1$ are all of weight infinity in all 2^{Γ} graphs. All the edges of the star S_m have unit weight in all 2^{Γ} graphs. Each of the Γ edges of the star S_0 may have weight zero or infinity.

The family $\mathcal{G}$ of unweighted graphs generalizes the family (that we refer to as $\mathcal{G}''$) of unweighted graphs that was used in [23] to prove a lower bound of $\Omega(n^{1/3}/B)$ on the number of rounds that are required for a distributed protocol to compute *exact MST* on graphs of diameter 4. However, the choice of weights above is inherently different from the one used in [23], and this difference will be discussed below.

Consider the *MST* problem restricted to the family $\mathcal{G}^{\omega}$ of graphs. We will show that for any $1 \leq H = o(n)$, *any* distributed protocol that given a graph $G \in \mathcal{G}^{\omega}$ constructs an H-approximate *MST* for G requires $\Omega((\frac{n}{p \cdot B \cdot H})^{\frac{1}{2} - \frac{1}{2(2p+1)}})$ rounds.

This proof is done by a reduction from a problem of distributed delivery of information throughout the graphs of

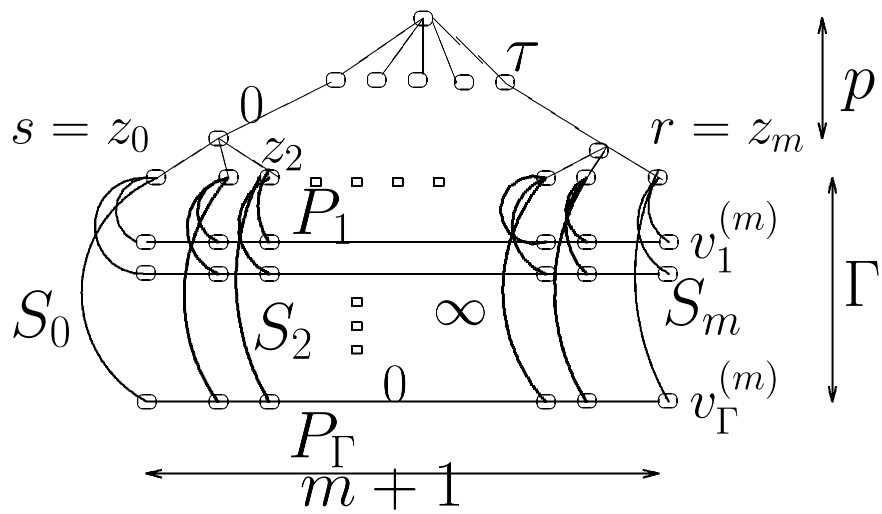

Figure 1: The family $\mathcal{G}$. The edges of the tree τ and of the paths $P_1, \ldots, P_\Gamma$ have weight zero. The edges of the "internal" stars $S_1, \ldots, S_{m-1}$ have infinite weights. The edges of the right-most star S_m have unit weights, and the edges of the left-most star S_0 have weights zero or infinity, depending on the particular graph.

family $\mathcal{G}$. The problem, referred by us as $CorruptedMail$ problem, generalizes the mailing problem of [25, 23] in two senses. First, the family $\mathcal{G}$ of graphs is somewhat more general than the corresponding families $\mathcal{G}'$ and $\mathcal{G}''$ in [25, 23], and this enables us to get lower bounds that are parameterized on the diameter. Secondly, the mailing problem of [25, 23] requires *exact* delivery of all the input bits, whereas our $CorruptedMail$ problem allows the protocol to make a restricted number of *one-sided errors* (i.e., some zero input bits might be delivered as ones, but not vice versa). This modification is geared to capture the situation when the (oracle) protocol for the *MST* problem (its existence is assumed by the reduction) does not compute the exact *MST*, as it is assumed in [25, 23], but rather provides to the reduction some (possibly very loose) approximation of it.

Let α and β, $0 < \alpha < \beta < 1$, be two additional parameters of the construction that will be fixed later. For a bit string $\chi \in \{0,1\}^\Gamma$ and an index $j = 1, 2, \ldots, \Gamma$, let χ_j denote the jth bit of χ. The *Hamming weight* of χ, denoted $hwt(\chi)$, is the number of indices $j = 1, 2, \ldots, \Gamma$ such that $\chi_j = 1$. For two bit strings $\chi, \chi' \in \{0,1\}^\Gamma$, the string χ' is said to *dominate* χ if for each $j = 1, 2, \ldots, \Gamma$, $\chi_j = 1$ implies $\chi'_j = 1$. Consider some mapping $\phi : \{0,1\}^\Gamma \to \{0,1\}^\Gamma$, and suppose $\phi(\chi) = \chi'$. The mapping ϕ is said to make an error of the the *first* (respectively, *second*) *type* in the j'th position, if $\chi_j = 0$ and $\chi'_j = 1$ (resp., $\chi_j = 1$ and $\chi'_j = 0$).

$CorruptedMail(\alpha, \beta)$ problem is defined on unweighted graphs $G_n \in \mathcal{G}$. Recall that for each graph $G \in \mathcal{G}$, there are two designated vertices s and r, $s, r \in V_n$. The input to this problem is a bit string $\chi \in \{0,1\}^\Gamma$ of length Γ with Hamming weight $hwt(\chi) = \alpha\Gamma$. The input is provided to the vertex s only. The output, returned by the vertex r, is a string $\chi' \in \{0,1\}^\Gamma$ of Hamming weight at most $\beta\Gamma$, and it is required that the output string χ' will dominate the input string χ. As the parameters α and β are fixed throughout this section, we will refer to $CorruptedMail(\alpha, \beta)$ problem as $CorruptedMail$ problem.

$CorruptedMail$ problem is a generalization of $Mail$ problem. $Mail$ problem was introduced in [25], and used later on in [23]. It was defined on graphs of families $\mathcal{G}'$ and $\mathcal{G}''$ (in [25] and [23], respectively), and each graph $G \in \mathcal{G}' \cup \mathcal{G}''$ has a pair of designated vertices, s and r (like each graph in the family $\mathcal{G}$ has). The input to this problem is also a bit string $\chi \in \{0,1\}^\Gamma$, and the input is provided to the vertex s only. However, in $Mail$ problem χ may be an arbitrary bit string of length Γ, and it is not required to have a Hamming weight of $\alpha\Gamma$. More importantly, the output of $Mail$ problem, returned by the vertex r (as in $CorruptedMail$ problem), should be the bit string χ (in other words, the output bit string χ' is required to be equal to the input bit string χ). This condition is far more restrictive then the condition that we introduced in $CorruptedMail$ problem, specifically, that χ' should dominate χ and have Hamming weight at most $\beta\Gamma$.

Observe, however, that the restriction that χ' should dominate χ guarantees that the errors that are done throughout the delivery of the bit string χ are *one-sided* (i.e., all the errors are of the first type). This is in contrast to the usual setting of error-correcting codes, where two-sided error is allowed. One of the crucial properties required for a reduction from $CorruptedMail$ problem to approximate *MST* problem to work is that approximating the *MST* with sufficiently small factor causes only one-sided errors in the delivery of the bit string χ. In our reduction, the latter is ensured by an appropriate choice of the weights of the edges in $\mathcal{G}^\omega$. This is the essential difference between our choice of weights of edges for $\mathcal{G}^\omega$, and choices of weights of edges in analogous families of graphs in [25] and [23].

We remark that enabling a symmetric two-sided error error in $CorruptedMail$ problem, and using error-correcting codes (with possible list-decoding) directly would also lead to a proof of hardness of distributed approximation of the *MST* problem within some small constant factor (smaller than 2, to the best of our knowledge). The bottleneck in this case is the fact that corrupting a bit string χ that has Hamming weight $\alpha\Gamma$ in at least $\alpha\Gamma$ arbitrary positions in an arbitrary way may make the string almost indistinguishable from a string that is drawn out of the uniform distribution over all the strings of Hamming weight at most $2\alpha\Gamma$, even from information-theoretic point of view. However, this is not the case when only one-sided error is allowed. Then, as we will show, one can allow $(\beta-\alpha)\Gamma$ corrupted positions for $1 > \beta \gg \alpha > 0$, and still the corrupted string χ' will carry on a significant amount of information (roughly, $\Gamma \cdot \alpha \cdot \log(1/\beta)$ bits for sufficiently small positive $\alpha > 0$; note that the amount of information carried by χ is $\Gamma \cdot \mathit{Entropy}(\alpha) = \Gamma \cdot (\alpha \log(1/\alpha) + (1-\alpha)\log(1/(1-\alpha))))$. This way, we are able to prove a hardness of distributed approximation of the *MST* problem within a factor of, roughly, $1/\alpha$, for arbitrarily small $\alpha > 0$.

As we mentioned, to ensure one-sided error of the reduction, an appropriate setting of the weights of edges of graphs in $\mathcal{G}^\omega$ is required. The setting that we described has, however, the drawback of huge ratio, denoted ω_{max}, between the biggest weight of an edge and the smallest one. Specifically, it is $\frac{\infty}{0}$. It is easy to see, however, that this ratio can be reduced to $O(n^2)$. However, if one wants to get an even smaller ω_{max}, it is no longer possible to guarantee a one-sided error in the delivery of the bit string χ in the reduction. Nevertheless, it can be shown that controlling the ratio between n^2 and ω_{max} controls also the number of errors of the second type. In this case we get to a situation when the delivery may suffer an *asymmetric two-sided error*, with a huge possible number of errors of the first type (specifically, $(\beta-\alpha)\Gamma$ for some $1 > \beta \gg \alpha > 0$), and a reasonably small number of errors of the second type (specifically, some constant fraction of $\alpha\Gamma$). This way one can achieve the same lower bound on time-approximation tradeoff as in the case that $\omega_{max} = \frac{\infty}{0}$ (that is, $T^2 \cdot H = \Omega(n/B)$), but with $\omega_{max} = \sqrt{n} \cdot H^{3/2}$, which is significantly smaller than n^2 for most values of H. The details of this argument are omitted from this extended abstract.

For $0 < \alpha < \beta < 1$, let

$$\begin{aligned} l(\alpha,\beta) &= (\beta-\alpha)\log(\beta-\alpha) \\ &- (1-\alpha)\log(1-\alpha) - \beta\log\beta \ . \end{aligned} \tag{1}$$

LEMMA 3.1. *For any deterministic protocol* Π *for the* $CorruptedMail(\alpha,\beta)$ *problem, its set of all possible outputs* $\{\Pi(\chi) \mid \chi \in \{0,1\}^\Gamma, hwt(\chi) = \alpha\Gamma\}$ *contains at least* $\Omega(2^{l(\alpha,\beta)\Gamma})$ *elements.*

PROOF. Let $\mathcal{A} = \{\chi \in \{0,1\}^\Gamma \mid hwt(\chi) = \alpha \cdot \Gamma\}$ be the set of all possible bit strings that may serve as input for the vertex s. Let $\Upsilon = \{\chi' \in \{0,1\}^\Gamma \mid hwt(\chi') \le \beta \cdot \Gamma\}$ be the set of all possible bit strings that may be returned by the vertex r. Consider the bipartite graph $(\mathcal{A}, \Upsilon, E(\mathcal{A}, \Upsilon))$ with $\mathcal{A}$ serving as the set of left-hand vertices, Υ serving as the set of right-hand vertices, and $E(\mathcal{A},\Upsilon) = \{(\chi,\chi') \mid \chi \in \mathcal{A}, \chi' \in \Upsilon$ s.t. χ' dominates $\chi\}$. Observe that $|\mathcal{A}| = \binom{\Gamma}{\alpha\Gamma} = \frac{\Gamma!}{(\alpha\Gamma)!((1-\alpha)\Gamma)!}$. Consider some bit string $\chi' \in \Upsilon$. The number of the bit strings $\chi \in \mathcal{A}$ that are dominated by χ' is at most $D = \binom{\beta\Gamma}{\alpha\Gamma} = \frac{(\beta\Gamma)!}{((\beta-\alpha)\Gamma)!(\alpha\Gamma)!}$.

A subset $\Upsilon' \subseteq \Upsilon$ is said to *dominate* the set $\mathcal{A}$, if for each bit string $\chi \in \mathcal{A}$ there exists a bit string $\chi' \in \Upsilon'$ that dominates χ. As each bit string $\chi' \in \Upsilon$ may dominate at most D bit strings of $\mathcal{A}$, it follows that any subset $\Upsilon' \subseteq \Upsilon$ that dominates $\mathcal{A}$ has cardinality at least

$$|\Upsilon'| \ge \frac{|\mathcal{A}|}{D} = \frac{\Gamma!((\beta-\alpha)\Gamma)!}{((1-\alpha)\Gamma)!(\beta\Gamma)!} \ .$$

Using Stirling formula to approximate the factorials, we get

$$|\Upsilon'| \ \ge \ \left(\frac{(\beta-\alpha)^{\beta-\alpha}}{(1-\alpha)^{1-\alpha}\beta^\beta}\right)^\Gamma \cdot \sqrt{\frac{\beta-\alpha}{(1-\alpha)\beta}} \cdot (1-o(1)) \ .$$

For β and α such that $\beta - \alpha > 0$ is at least some constant it follows that $|\Upsilon'| = \Omega(\left(\frac{(\beta-\alpha)^{\beta-\alpha}}{(1-\alpha)^{1-\alpha}\beta^\beta}\right)^\Gamma)$.

Let $h(\alpha,\beta) = \frac{(\beta-\alpha)^{\beta-\alpha}}{(1-\alpha)^{1-\alpha}\beta^\beta}$, and note that $l(\alpha,\beta) = \log h(\alpha,\beta)$. It follows that $|\Upsilon'| = \Omega(2^{l(\alpha,\beta)\Gamma})$, for any subset $\Upsilon' \subseteq \Upsilon$ of bit strings that dominate $\mathcal{A}$. ∎

We next show that for any protocol Π with worst-case running time at most t for t in certain range, its set of possible outputs (over all possible inputs) is small, namely, at most exponential in t. (To argue this we use determinism of the protocol, and that the vertex r that returns the output could not get too much information about the input.) It will follow that the running time t of a protocol Π cannot be too small, unless the protocol Π is incorrect. To prove an upper bound on the size of the set of possible outputs of any correct protocol, we show that the set of all possible *configurations* of the vertex r is small as function of t.

For some fixed sufficiently large n, consider again the graph $G = (V,E) = G_n \in \mathcal{G}$ that was described in the beginning of the section. Intuitively, we next argue that information can be delivered through G_n from s to r in quite a slow rate. Similar statement was proven in [25, 23] regarding somewhat different families of graphs $\mathcal{G}'$ and $\mathcal{G}''$.

Our proof has a similar structure to that of [25], but as the structure of the family $\mathcal{G}$ is somewhat more complicated than that of $\mathcal{G}'$ or $\mathcal{G}''$, the proof requires more delicate argument. Basicly, all the three proofs (due to [25], [23], and ours) construct a sequence of low-capacity cuts, and argue that each bit has to cross all these cuts. As the cuts have low capacities, no cut can be crossed by "many" bits simultaneously, implying a lower bound on the number of rounds of distributed computation. However, while the choice of the cuts and the proof that they have low capacity are rather straightforward in [25, 23], it is somewhat more involved in our case.

The lower bounds in [23] on the *exact* computation of the *MST* on graphs with constant diameter (specifically, 3 and 4) are no bigger than $\Omega(n^{1/3}/B)$. Our construction enables to get a lower bound of $\Omega((n/B)^{1/2-\epsilon})$ for any $\epsilon > 0$ for graphs with constant diameter (equal to $O(1/\epsilon)$) on *approximate* computation of the *MST*. For the specific case of diameter 3 and 4 our construction improves the lower bounds of [23] by factors $B^{3/4}$ and $B^{2/3}$, respectively.

We remark that one could use the family of graphs of [25] to prove a somewhat weaker lower bound on the time-approximation tradeoff of the *MST* problem restricted to graphs with diameter at most polynomial in n, and the family $\mathcal{G}$ in our proof is required to prove such a tradeoff for this problem restricted to graphs of constant diameter, and obtaining a stronger lower bound for the general problem. This simpler version of our result (that uses the family $\mathcal{G}'$ of [25] and does not introduce the family $\mathcal{G}$) could actually be proved by defining appropriate weights for the edges of the graphs of the family $\mathcal{G}'$, and proving the lower bound on distributed complexity of the $CorruptedMail$ problem on $\mathcal{G}'$. (However, the different choice of weights is crucial to ensure the one-sided error, as was already discussed.)

For a rooted tree (τ', rt'), let the *ancestor-descendent* (resp., *parent-child*) relation, denoted $AD(\tau', rt')$ (resp., $PC(\tau', rt')$), be the set of pairs of vertices $(u, w) \in V(\tau')^2$ such that the vertex u is an ancestor (resp., parent) of the vertex w in the tree (τ', rt'). For a vertex $u \in V(\tau')$, let $par_{(\tau', rt')}(u)$ denote the *parent* of u in the rooted tree (τ', rt').

Recall that the graph $G = G_n$ contains as a subgraph the d-regular rooted tree (τ, rt) of height p, with $m+1$ leaves $s = z_0, z_1, \ldots, z_m = r$. For $i = 0, 1, 2, \ldots, m$, let $\tau(i)$ denote the connected subtree of τ with minimal number of vertices, such that its set of leaves, $Leaves(\tau(i))$, is equal to $\{z_i, z_{i+1}, \ldots, z_m\}$. Let the *root* of $\tau(i)$, denoted $rt(\tau(i))$, be the closest vertex of $\tau(i)$ to the root rt of the tree τ.

Observe that for any pair of vertices $(x, y) \in AD(\tau, rt)$ such that $x, y \in V(\tau(i))$ for some $i = 0, 1, \ldots, m$, $(x, y) \in AD(\tau(i), rt(\tau(i)))$. Also, $AD(\tau(i), rt(\tau(i))) \subseteq AD(\tau, rt)$, for each $i = 0, 1, \ldots, m$. Note that both statements are true for the parent-child relation PC as well.

We define the *tail* sets, $T_0, T_1, \ldots, T_m$ as follows. The tail set T_0 contains the entire vertex set V of G except the vertex s, i.e., $T_0 = V \setminus \{s\}$. For $i = 1, 2, \ldots, m$, $T_i = \{v_j^{(i')} \mid j = 1, 2, \ldots, \Gamma, i' = i, i+1, \ldots, m\} \cup V(\tau(i))$. Fix also some arbitrary total order ψ on on the vertex set V, and let $\vec{T_i} = \psi(T_i)$ denote the ordered sequence of elements of T_i, for $i = 0, 1, \ldots, m$.

Consider some protocol Π for the $CorruptedMail$ problem, and consider an execution φ_χ of this protocol on some input bit string χ. Let the *state* of the vertex v at the beginning of round t during the execution φ_χ of the protocol Π, denoted $\sigma(v, t, \chi)$, be the $deg(v)$-tuple of sequences of messages that the vertex v received on its incoming links. For the vertex s, the state of s also includes the input string χ. (Other vertices receive no input in the $CorruptedMail$ problem.)

For some subset $U \subseteq V$ of vertices, such that $\vec{U} = (u_1, u_2, \ldots, u_\ell)$, let *configuration* of U in the execution φ_χ at the beginning of round t, denoted $C(U, t, \chi)$, be the sequence of states $(\sigma(u_1, t, \chi), \sigma(u_2, t, \chi), \ldots, \sigma(u_\ell, t, \chi))$. Let $\mathcal{C}(U, t)$ denote the collection of all possible configurations of the subset U at the beginning of round t over all possible executions φ_χ (i.e., for all possible legal input bit strings χ; given a bit string, the execution φ_χ is fixed). Let $\rho(U, t)$ denote the cardinality of $\mathcal{C}(U, t)$.

In what follows let us assume that the rounds are indexed starting from 0 and not from 1. Obviously, this may affect the lower bounds by at most an additive term of 1. At the beginning of round $t = 0$, i.e., at the beginning of the execution of the protocol, all the vertices except s are in some fixed initial state, that is independent of the input bit string χ. Hence, $\rho(T_0, 0) = \rho(V \setminus \{s\}, 0) = 1$. We next prove an upper bound on the number of configurations of the tail set T_t, after relatively small number of rounds $t \le m - 1$.

LEMMA 3.2. *For $t = 0, 1, \ldots, m-1$, $\rho(T_t, t) \le (2^{B+1} - 1)^{t \cdot p \cdot d}$.*

PROOF. The proof is by induction on t. The induction base was argued above. We next prove the induction step. Observe that $T_0 \supseteq T_1 \supseteq \ldots \supseteq T_m$. Suppose we are given a configuration $C \in \mathcal{C}(T_i, i)$. Note that C uniquely determines the messages that are sent at round i by vertices of T_i. Therefore, the only messages that may cause multiple configurations of T_{i+1} are those that are sent by the vertices of $V \setminus T_i$ to the vertices of T_{i+1}. Denote this set of edges by $E_{i,i+1}$. Observe that the edges of the paths $P_1, P_2, \ldots, P_\Gamma$ do not belong to $\bigcup_{i=0}^{m-1} E_{i,i+1}$. It follows that $E_{i,i+1} \subseteq E(\tau)$, for $i = 0, 1, \ldots, m-1$. For $i = 0$, clearly, $E_{0,1} = \{(z_0, par_{(\tau,rt)}(z_0))\}$. More generally, $E_{i,i+1}$ is a subset of edges of $E(\tau)$ that are incident both to $V(\tau(i+1))$ and to $V \setminus V(\tau(i))$, $i = 0, 1, \ldots, m-1$. See Figure 2 for illustration.

Let Cut_i denote the subset of edges of $E(\tau)$ that are in the cut between $V(\tau(i))$ and the rest of $V(\tau)$, for $i = 1, 2, \ldots, m$. It follows that $E_{i,i+1} \subseteq Cut_{i+1}$, for $i = 0, 1, \ldots, m-1$. We next argue that the size of this cut is never greater than $p \cdot d$.

LEMMA 3.3. *For $i = 1, 2, \ldots, m$, $|Cut_i| \le p \cdot d$.*

PROOF. Observe that for any edge $(u, w) \in Cut_i$, either u is a parent of w in the tree τ or vice versa. For a fixed $i = 1, 2, \ldots, m$, let $L_j = \{(u, w) \in Cut_i \cap AD(\tau, rt) \mid dist_\tau(rt, u) = j\}$, for $j = 0, 1, \ldots, p-1$, be the *jth layer* of the edge set Cut_i. Note that $Cut_i = \bigcup_{j=0}^{p-1} L_j$. It remains to argue that $|L_j| \le d$, for $j = 0, 1, \ldots, p-1$.

To this end, we will show that for any pair of edges $e_1 = (u_1, w_1), e_2 = (u_2, w_2) \in L_j$ (assume, without loss of generality, that $u_1 = par_{(\tau,rt)}(w_1)$ and $u_2 = par_{(\tau,rt)}(w_2)$), u_1 is equal to u_2. It will follow that all the edges of L_j share a common endpoint, and, furthermore, this endpoint is a parent of all the other endpoints of these edges. As every vertex has at most d children in the tree τ, this would imply $|L_j| \le d$.

Consider a pair of edges e_1, e_2 as above. First, suppose $u_1 \notin V(\tau(i))$, $w_1 \in V(\tau(i))$. As the parent-child relation in $\tau(i)$ is a subset of the parent-child relation in τ, it follows that $w_1 = rt(\tau(i))$.

As $dist_\tau(rt, w_1) = j+1$, it follows that w_1 may be incident to at most one edge of L_j, that is, e_1. For any other vertex $v \in V(\tau(i)) \setminus \{w_1\}$, $dist_\tau(rt, v) > dist_\tau(rt, w_1) \ge j+1$. I.e., $dist_\tau(rt, v) \ge j+2$. But for any edge of L_j, both its endpoints are at distance at most $j+1$ from the root rt of τ. Hence, no edge of L_j is incident to a vertex $v \in V(\tau(i)) \setminus \{w_1\}$, proving that in this case $|L_j| = 1$ (and $e_1 = e_2$).

It remains to consider the case when $u_1, u_2 \in V(\tau(i))$ and $w_1, w_2 \in V(\tau) \setminus V(\tau(i))$. Suppose also that $e_1 \ne e_2$, as otherwise they share a common endpoint u_1, and we are done. By definition of $\tau(i)$, as $u_1, u_2 \in V(\tau(i))$, there exist indices $a, b \in \{i, i+1, \ldots, m\}$, such that $(u_1, z_a), (u_2, z_b) \in AD(\tau(i), rt(\tau(i)))$. Note that as $\tau(i)$ is acyclic, $a \ne b$. Assume, without loss of generality, that $i \le a < b$. Consider a descendent z_c of w_2. Note that for any two vertices u_1 and u_2 such that neither (u_1, u_2) nor (u_2, u_1) belong to the ancestor-descendent relation of the rooted tree (τ, rt), if some leaf-descendent of u_1 has smaller index than some leaf-descendent of u_2, then *any* leaf-descendent of u_1 has smaller index than *any* leaf-descendent of u_2. As $dist_\tau(rt, u_1) = dist_\tau(rt, u_2) = j$, it follows that $(u_1, u_2) \notin AD(\tau, rt)$, $(u_2, u_1) \notin AD(\tau, rt)$. Hence, $a < b$ and $(u_1, z_a), (u_2, z_b), (u_2, z_c) \in AD(\tau, rt)$ imply that $c > a \ge i$. By definition of $\tau(i)$, it follows that $z_c \in V(\tau(i))$. Recall that $u_2 \in V(\tau(i))$ as well. As $\tau(i)$ is a connected subtree of τ, the entire path that connects u_2 to z_c in τ belongs to $\tau(i)$.

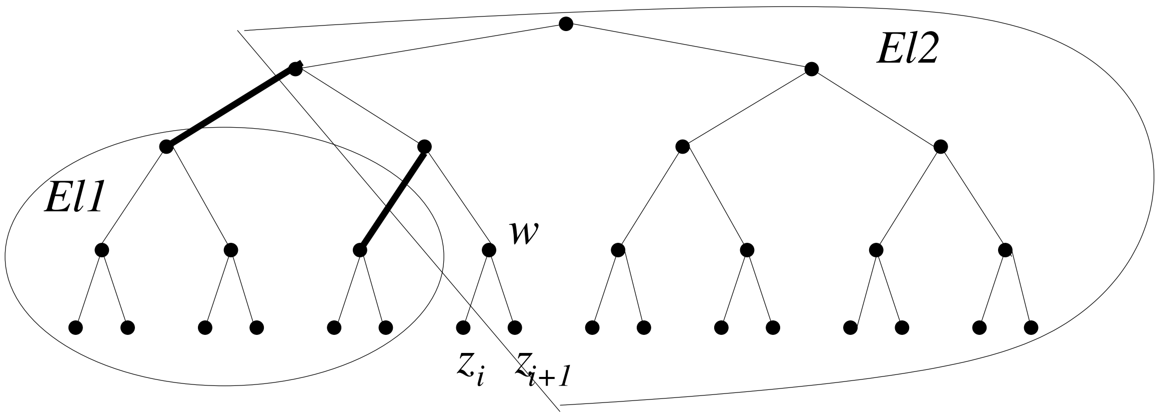

Figure 2: The sets $V(\tau) \setminus V(\tau(i))$ and $V(\tau(i+1))$ are depicted by ellipses $El1$ and $El2$, respectively. Edges of $E_{i,i+1}$ are depicted by thick solid lines. Note that the edge (z_i, w) does not belong to $E_{i,i+1}$, because $z_i \in V(\tau(i))$.

I.e., in particular, $w_2 \in V(\tau(i))$, contradiction. This proves that $u_1 = u_2$. Hence, $|L_j| \leq d$, implying $|Cut_i| \leq p \cdot d$. ∎

We conclude that $|E_{i,i+1}| \leq p \cdot d$ for $i = 0, 1, \ldots, m-1$. In other words, for $i = 0, 1, \ldots, m-1$, there are at most $p \cdot d$ edges that are incident to the tail set T_{i+1}, such that given a configuration C in $\mathcal{C}(T_i, i)$ of the vertices of T_i at the beginning of round i, the messages that are sent over these edges at round i are not determined by this configuration.

Recall that at most B bits can be delivered through an edge in each round. Therefore, the number of possible messages that may be sent through an edge in a given direction in one round is at most $\sum_{\ell=0}^{B} 2^\ell = 2^{B+1} - 1$. Observe that there is only one relevant direction of sending messages in our case, that is, towards the vertices of the tail set T_{i+1}. Hence, the number of possible messages that may be sent through at most $p \cdot d$ edges in one round is at most $(2^{B+1}-1)^{p \cdot d}$. It follows that for $i = 0, 1, \ldots, m-1$, $\rho(T_{i+1}, i+1) \leq (2^{B+1}-1)^{p \cdot d} \cdot \rho(T_i, i)$. This proves Lemma 3.2. ∎

Lemma 3.4. *The deterministic complexity of CorruptedMail problem is* $\Omega(\min\{m, \frac{\Gamma}{p \cdot B \cdot m^{1/p}} \cdot \alpha \cdot \log(1/\beta)\})$.

Proof. Consider a protocol Π for the $CorruptedMail$ problem, and let t denote its worst-case running time on inputs of fixed size Γ. By Lemma 3.2, it follows that if $t \leq m-1$ then $\rho(T_t, t) \leq (2^{B+1}-1)^{t \cdot p \cdot d}$. On the other hand, observe that the number of possible configurations of the vertex r upon the termination of the protocol Π is at least the cardinality of some subset $\Upsilon' \subseteq \Upsilon$ of a subset of bit strings that dominates $\mathcal{A}$. By Lemma 3.1, $|\Upsilon'| = \Omega(2^{l(\alpha,\beta)\Gamma})$. It follows that the (worst-case) running time t of the protocol Π for the $CorruptedMail$ problem is either at least $t \geq m$ rounds, or it satisfies the the following inequality

$$\Omega(2^{l(\alpha,\beta)\Gamma}) = |\Upsilon'| \leq \rho(\{r\}, t) \leq \rho(T_t, t) \leq (2^{B+1}-1)^{t \cdot p \cdot d} . \tag{2}$$

(The inequality $\rho(\{r\}, t) \leq \rho(T_t, t)$ follows from the fact that $r \in T_t$ for $t = 0, 1, \ldots, m$.) Hence, $\Omega(l(\alpha,\beta)\Gamma) - O(1) \leq t \cdot p \cdot d \cdot B$. I.e., $t = \Omega(l(\alpha,\beta) \cdot \Gamma/(p \cdot d \cdot B))$. In other words, in both cases $t = \Omega(\min\{m, (l(\alpha,\beta) \cdot \Gamma)/(p \cdot d \cdot B)\})$. Recall that $n = O(\Gamma m)$, and $d = m^{1/p}$. Hence,

$$t = \Omega(\min\{m, l(\alpha,\beta)\frac{n}{p \cdot m^{1+1/p} \cdot B}\}) . \tag{3}$$

Recall that $l(\alpha, \beta)$ is given by (1). For small $\alpha > 0$,

$$l(\alpha,\beta) = \alpha \cdot \log(1/\beta) + o(\alpha) . \tag{4}$$

(When α does not tend to 0 when n grows to infinity, the inequality (4) may no longer hold, but the inequality $l(\alpha,\beta) = \Omega(\alpha \log(1/\beta))$ obviously holds, as both sides are independent of n.) Set β to be a constant between 0 and 1. For α either constant or tending to 0 when n grows, we get $t = \Omega(\min\{m, \frac{n \cdot \alpha}{p \cdot m^{1+1/p} \cdot B}\})$. ∎

Set $m = \left(\frac{n \cdot \alpha}{p \cdot B}\right)^{1/2 - \frac{1}{2(2p+1)}}$. It follows that any deterministic protocol Π that solves the $CorruptedMail$ problem on *every* input bit string χ requires $\Omega\left(\left(\frac{n \cdot \alpha}{p \cdot B}\right)^{1/2 - \frac{1}{2(2p+1)}}\right)$ rounds in the worst-case.

Henceforth, the term "randomized protocol" will be used as a shortcut for "randomized protocol that succeeds with at least some constant probability on every input".

Lemma 3.5. *Any randomized protocol* Π *for the CorruptedMail problem requires* $\Omega\left(\left(\frac{n \cdot \alpha}{p \cdot B}\right)^{1/2 - \frac{1}{2(2p+1)}}\right)$ *rounds.*

Proof. Consider a deterministic protocol Π' that accepts as input the uniform distribution over the input bit strings χ of Hamming weight $\alpha\Gamma$. For such a protocol to succeed with at least a constant probability q in $t \leq m-1$ rounds, the number of configurations of the vertex r at the end of the protocol has to be at least the size of a subset $\Upsilon'' \subseteq \Upsilon$ that dominates at least a fraction q of the set $\mathcal{A}$. However, by the same considerations as above, such a subset Υ'' must have cardinality at least $q|\mathcal{A}|/D$. In other words, for protocol Π', $\rho(T_t, t) \geq \rho(\{r\}, t) \geq q|\mathcal{A}|/D = \Omega(2^{l(\alpha,\beta)\Gamma})$. In other words, up to lower-order terms, the same lower bound applies also to a protocol Π' as above. By Yao's Minimax theorem, any randomized protocol that succeeds with probability at least q on *every* input of the $CorruptedMail$ problem requires at least as many rounds as requires a deterministic protocol

Π' that succeeds with the same probability on the uniform distribution of inputs. Hence, the lower bound applies to randomized protocols as well. ∎

3.2 Reduction to the Approximate MST Problem

In this section we describe the reduction from the $CorruptedMail(\alpha, \beta)$ problem on the family $\mathcal{G}$ of unweighted graphs to the $\frac{\beta}{\alpha}$-approximate MST problem on the family $\mathcal{G}^\omega$ of weighted graphs (see the beginning of Section 3.1 for the definition of this family).

The protocol Π_{Corr} for the $CorruptedMail(\alpha, \beta)$ problem proceeds in the following way. Given an instance (G, χ), $G \in \mathcal{G}$, $\chi \in \mathcal{A}$, of the $CorruptedMail$ problem, the vertex s computes the weights of edges $(s, v_j^{(0)})$, $j = 1, 2, \ldots, \Gamma$, in the following way. If $\chi_j = 0$ then the weight of the edge $(s, v_j^{(0)})$ is set to zero, otherwise it is to infinity. All the other weights of edges are set by their endpoints to the values that are determined by the definition of the family $\mathcal{G}^\omega$. This setting of weights is performed locally by every vertex, and requires no distributed computation.

Next, the vertices invoke $\frac{\beta}{\alpha}$-approximation protocol Π for the obtained instance $G(\chi)$ of the MST problem. Upon the termination of the protocol, each vertex v knows which edges among the edges that are incident to v belong to the approximate MST tree τ_0 for $G(\chi)$, that was constructed by the protocol. The vertex r calculates the output bit string $\chi' \in \{0,1\}^\Gamma$ in the following way. For each index $j = 1, 2, \ldots, \Gamma$, if the edge $(r, v_j^{(m)})$ belongs to the tree τ_0, the vertex r sets $\chi'_j = 1$. Otherwise, it sets $\chi'_j = 0$. Finally, the vertex r returns the bit string χ'.

Observe that whenever the construction of the approximate MST tree τ_0 is completed, the computation of the bit string χ' is performed locally by the vertex r, and requires no distributed computation. It follows that the running time of the obtained protocol Π_{Corr} for the $CorruptedMail(\alpha, \beta)$ problem is precisely equal to the running time of the $\frac{\beta}{\alpha}$-approximation protocol Π for the MST problem.

LEMMA 3.6. *For each $\chi \in \mathcal{A}$, if τ_0 is a $\frac{\beta}{\alpha}$-approximate MST for $G(\chi)$, then the bit string χ', that is returned by the protocol Π_{Corr}, has Hamming weight at most $\beta \cdot \Gamma$, and, in addition, the bit string χ' dominates the input bit string χ.*

PROOF. Consider a bit string $\chi \in \{0,1\}^\Gamma$ of Hamming weight $hwt(\chi) = \alpha\Gamma$. By construction, precisely $\alpha\Gamma$ edges of the star S_0 have weight ∞ in $G(\chi)$. Therefore, the exact MST contains all the edges of the paths $P_1, P_2, \ldots, P_\Gamma$ and of the tree τ, as all of them have weight zero, and it contains $(1-\alpha)\Gamma$ edges of weight zero that belong to the star S_0. In addition, for each index $j = 1, 2, \ldots, \Gamma$ such that $\omega((s, v_j^{(0)}) = \infty$, it contains the edge $(r, v_j^{(m)})$. Recall that the latter edge has unit weight. Also, note that $\omega((s, v_j^{(0)}) = \infty$ implies that $\chi_j = 1$. As $hwt(\chi) = \alpha\Gamma$, it follows that exactly $\alpha\Gamma$ edges of the star S_0 have weight ∞. Hence, exactly $\alpha\Gamma$ edges of the star S_m belong to the MST, implying that its weight is $\alpha\Gamma$.

Consider a $\frac{\beta}{\alpha}$-approximate MST τ_0. By definition of approximate MST, its weight is at most $\beta\Gamma$. Hence, in particular, it contains at most $\beta\Gamma$ edges of the star S_m, implying that $hwt(\chi') \leq \beta\Gamma$. Consider an index $j = 1, 2, \ldots, \Gamma$ such that $\chi_j = 1$. It follows that the weight of the edge $(s, v_j^{(0)})$ in $G(\chi)$ is ∞. As no edge with infinite weight may belong to τ_0 (as its weight is at most $\beta\Gamma$), it follows that neither $(s, v_j^{(0)})$ nor $(z_i, v_j^{(i)})$ for some $i = 1, 2, \ldots, m-1$ may belong to the tree τ_0. It follows that the edge $(r, v_j^{(m)})$ belongs to τ_0, as otherwise the vertex s would not be connected in τ_0 to the vertices of the path P_j. The latter would imply that τ_0 is not a spanning tree of the graph $G(\chi)$, contradicting the assumption that it is an approximate MST for $G(\chi)$. Hence, the edge $(r, v_j^{(m)})$ belongs to the tree τ_0. Hence, the bit χ'_j is set to 1 by the reduction. It follows that the output bit string χ' dominates the input bit string χ, and that $hwt(\chi') \leq \beta\Gamma$. ∎

Therefore, if Π is a $\frac{\beta}{\alpha}$-approximation protocol for MST on the family $\mathcal{G}^\omega$ of weighted graphs, then Π_{Corr} is a protocol for the $CorruptedMail(\alpha, \beta)$ problem on the family $\mathcal{G}$ of unweighted graphs, with the same running time. Recall that any (deterministic or randomized) protocol for the $CorruptedMail(\alpha, \beta)$ problem on the family $\mathcal{G}$ of graphs requires $t = \Omega\left(\left(\frac{n \cdot \alpha}{p \cdot B}\right)^{1/2 - \frac{1}{2(2p+1)}}\right)$ rounds. Observe that all the graphs in the family $\mathcal{G}^\omega$ have the same unweighted diameter $2p+2$. The next theorem follows.

THEOREM 3.7. *Any randomized H-approximation protocol for the MST problem on graphs of diameter at most Λ for $\Lambda = 4, 6, 8, \ldots$ requires $T = \Omega\left(\left(\frac{n}{H \cdot \Lambda \cdot B}\right)^{1/2 - \frac{1}{2(\Lambda-1)}}\right)$ rounds of distributed computation. I.e., $T^{2+\frac{2}{\Lambda-2}} \cdot H = \Omega\left(\frac{n}{\Lambda \cdot B}\right)$.*

In particular, for any $\epsilon > 0$, approximation of MST on graphs with constant diameter $\Lambda \geq 4$ within a factor of $\left(\frac{n}{B}\right)^{1-\epsilon}$ requires at least $\Omega(\left(\frac{n}{B}\right)^{\epsilon(1/2 - \frac{1}{2\Lambda-2})})$ rounds of distributed computation.

4. AN UPPER BOUND

In this section we devise a distributed protocol for the approximate MST problem. Our protocol runs in $O(\Lambda(G) + n^\epsilon \cdot \log^* n)$ rounds and constructs $O(\frac{\omega_{max}}{n^\epsilon})$-approximate MST, where ω_{max} is the ratio between the weights of the heaviest and the lightest edges in the graph G. This result applies for any $0 < \epsilon < 1$.

Throughout this section we assume that $B = \log n$. Hence the protocol can be used whenever $B = \Omega(\log n)$, and its running time will be the same. Also, obviously it can be adapted to the case $B = o(\log n)$ incurring an overhead of $O(\frac{\log n}{B}) = O(\log n)$. Also, we assume that the weights of edges are scaled between 1 and ω_{max}.

Consider an MST τ_0 of the graph G. A connected subtree of τ_0 is called a *fragment* of τ_0. A *k-MST forest* $\mathcal{F}$ of a graph $G = (V, E)$ is a collection of vertex-disjoint trees that satisfy

1. $\bigcup_{T \in \mathcal{F}} V(T) = V$, $\bigcup_{T \in \mathcal{F}} E(T) \subseteq E$.
2. $|V(T)| = \Omega(k)$, $\Lambda(T) = O(k)$.
3. There exists an MST τ_0 for the graph G, such that each tree $T \in \mathcal{F}$ is its fragment.

The notion of k-MST forest is related to the notion of (σ, ρ) spanning forest of [20]. Trees $T \in \mathcal{F}$ of a (σ, ρ) spanning forest $\mathcal{F}$ have to satisfy properties (1) and (2) with $|V(T)| \geq \sigma$ and $\Lambda(T) \leq \rho$, but may not satisfy property (3). It was demonstrated in [20] that a k-MST forest of an

n-vertex graph G can be constructed in $O(k \cdot \log^* n)$ rounds of distributed computation.

The first step of our n^ϵ-approximation protocol for the MST problem is to construct an n^ϵ-MST forest $\mathcal{F}$. This requires $O(n^\epsilon \cdot \log^* n)$ rounds. The second step is to construct a BFS tree τ of the entire graph, rooted at some arbitrary vertex $rt = rt(\tau)$. This requires $O(\Lambda(G))$ rounds. After the construction of the tree τ, each vertex v in the graph knows its unweighted distance to the root rt, $dist_\tau(rt, v)$. At the third step of the protocol, convergecasts are conducted in parallel over the spanning trees of the fragments $T \in \mathcal{F}$ of the n^ϵ-MST forest $\mathcal{F}$. Throughout the convergecast over a fragment T, the root of the fragment T learns the identity of the vertex $v \in V(T)$ that is closest in τ to the root rt of τ. The draws are broken arbitrarily. The fourth step involves broadcasts over the spanning trees of the fragments of the identities of these chosen vertices. Both convergecasts and broadcasts are done in parallel (recall that the fragments are vertex-disjoint), and require $O(\max\{\Lambda(T) \mid T \in \mathcal{F}\})$ rounds. As $\mathcal{F}$ is an n^ϵ-MST forest, it follows that $\Lambda(T) = O(n^\epsilon)$ for every tree $T \in \mathcal{F}$. Hence, these steps require $O(n^\epsilon)$ rounds. After the broadcasts are over, the fifth step occurs. On the fifth step in each fragment $T \in \mathcal{F}$, the chosen vertex $v \in V(T)$ inserts the edge $e_v = (par_{(\tau, rt(\tau))}(v), v)$ into the tree τ_0, that the protocol constructs. It also informs his parent in τ, $par_{(\tau, rt(\tau))}(v)$, that it was chosen, and the parent inserts the edge e_v into τ_0 as well. In addition, each vertex w in the graph inserts into the tree τ_0 the edges of the k-MST forest $\mathcal{F}$ that are incident to w. Observe that the fifth step requires only one round of distributed computation (for sending the messages by the chosen vertices to their parents in τ).

This completes the description of the protocol. It follows from our discussion that its running time is $O(\Lambda(G) + n^\epsilon \cdot \log^* n)$. We next argue that it is indeed an $O(\frac{\omega_{max}}{n^\epsilon})$-approximation protocol for the MST problem.

LEMMA 4.1. *The subgraph τ_0, that is constructed by the protocol, is acyclic.*

PROOF. Suppose for contradiction that there is a cycle $((u_0, w_0), P_0, (u_1, w_1), P_1, \ldots, (u_{t-1}, w_{t-1}), P_{t-1}))$, where P_i is a path in some fragment T_i between w_i and $u_{((i+1) \pmod t)}$, and $(u_i, w_i) \in E(\tau)$, $i = 0, 1, \ldots, t-1$ (where τ is the BFS spanning tree of the graph). Fragments may appear more than once. (Observe that the cycle cannot be contained entirely in one fragment, because for each fragment $T \in \mathcal{F}$, the edges of τ_0 with both endpoints in T all belong to the spanning tree of T.) It follows that $|dist_\tau(rt, u_i) - dist_\tau(rt, w_i)| = 1$ for $i = 0, 1, \ldots, t-1$.

Assume, without loss of generality, that $dist_\tau(rt, w_0) - dist_\tau(rt, u_0) = 1$. I.e., $u_0 = par_{(\tau, rt)}(w_0)$. Then $u_1 = par_{(\tau, rt)}(w_1)$, because otherwise both edges $(u_0, w_0), (u_1, w_1)$ connect vertices of the same fragment (the vertices w_0 and u_1) to their parents in the BFS tree τ. However, at most one such an edge is inserted into the tree τ_0 for each fragment in the approximation protocol. It follows that $u_i = par_{(\tau, rt)}(w_i)$ for $i = 0, 1, \ldots, t-1$. Hence,

$$\begin{aligned} dist_\tau(rt, u_0) \quad &> \quad dist_\tau(rt, w_0) \geq dist_\tau(rt, u_1) > \\ \ldots \geq \ldots \quad &> dist_\tau(rt, w_{t-1}) \geq dist_\tau(rt, u_0) \ . \end{aligned}$$

This is a contradiction, implying that the subgraph τ_0 is acyclic. ∎

LEMMA 4.2. *The subgraph τ_0 is connected, and it is spanning all the vertices of the graph G.*

PROOF. The second assertion follows directly from the observation that the edge set of the tree τ_0 contains the edge set of the n^ϵ-MST forest $\mathcal{F}$, and from the definition of a k-MST forest.

For the first assertion, consider some pair of vertices u and w in V. Let T_u and T_w be the pair of fragments of the n^ϵ-MST forest $\mathcal{F}$ such that $u \in T_u$, $w \in T_w$. If $T_u = T_w$ then u and w are connected in τ_0, because the subgraph τ_0 contains a spanning tree of each fragment T of the n^ϵ-MST forest $\mathcal{F}$, and, in particular, of $T_u = T_w$. Otherwise, let $u_0 \in T_u$ (resp., $w_0 \in T_w$) be the closest vertex in T_u (resp., T_w) to $rt = rt(\tau)$ (in terms of the unweighted distance). To prove that there is a path between u and w in τ_0, it suffices to prove that there is a path between u_0 and w_0. We prove this by induction on $dist_\tau(rt, u_0) + dist_\tau(rt, w_0)$.

The induction base is the case when the sum is 0, i.e., $u_0 = w_0 = rt$, and then the assertion is obvious.

For the induction step, consider the parent $v = par_{(\tau, rt)}(u_0)$ of the vertex u_0 in the BFS tree τ. Observe that $dist_\tau(rt, v) = dist_\tau(rt, u_0) - 1 < dist_\tau(rt, u_0)$, and that the edge $e = (u_0, v)$ belongs to τ_0. Thus, it suffices to prove that there is a path between v and w_0 in τ_0. Let T_v be the fragment of $\mathcal{F}$ that contains the vertex v, and let v_0 be the vertex that was chosen by the convergecast on this fragment. It follows that $dist_\tau(rt, v_0) \leq dist_\tau(rt, v) < dist_\tau(rt, u_0)$, and, thus, the induction hypothesis is applicable to the pair of vertices, v_0 and w_0. As the fragment T_v is connected, the lemma follows. ∎

It follows from Lemmas 4.1 and 4.2 that τ_0 is a spanning tree of the graph G.

LEMMA 4.3. *The tree τ_0 is a $(1 + O(\frac{\omega_{max}}{n^\epsilon}))$-approximate MST of the graph G.*

PROOF. Let $\omega : E \to R^+$ denote the weight function that is associated with the graph G, and let $\omega(MST)$ denote the weight of the MST. Observe that the weight of the n^ϵ-MST forest $\mathcal{F}$ is at most $\omega(MST)$, and that the convergecast and broadcast procedures insert into τ_0 at most $O(n^{1-\epsilon})$ additional edges. Hence, $\omega(\tau_0) = O(n^{1-\epsilon} \cdot \omega_{max}) + \omega(MST)$. Thus,

$$\frac{\omega(\tau_0)}{\omega(MST)} = 1 + O\left(\frac{n^{1-\epsilon} \cdot \omega_{max}}{\omega(MST)}\right) \ .$$

Recall that by our assumption, all the weights are scaled between 1 and ω_{max}. Hence, $\omega(MST) \geq n - 1$. It follows that $\omega(\tau_0)/\omega(MST) = O(\omega_{max}/n^\epsilon)$. ∎

To conclude,

THEOREM 4.4. *For any $0 < \epsilon < 1$, there exists a protocol that constructs an H-approximate MST for an n-vertex weighted graph (G, ω) in $O(\Lambda(G) + n^\epsilon \cdot \log^* n)$ rounds with $H = 1 + O(\frac{\omega_{max}}{n^\epsilon})$. Let $T = n^\epsilon \cdot \log^* n$. Then $T \cdot H = O(\omega_{max} \cdot \log^* n)$.*

Note that for graphs with small ω_{max} (e.g., constant, or polylogarithmic in n) this protocol provides an approximation ratio that is arbitrarily close to 1, and the running time of the protocol is arbitrarily close to $O(\Lambda(G))$.

Acknowledgements: The author is grateful to Michael Langberg, Zvika Lotker, Alessandro Panconesi, David Peleg, Alexander Razborov, Oded Regev, Vitaly Rubinovich and Avi Wigderson for helpful discussions.

5. REFERENCES

[1] B. Awerbuch, Optimal distributed algorithms for minimum weight spanning tree, counting, leader election and related problems. In *Proc. 19th ACM Symp. on Theory of Computing*, pp. 230-240, May 1987.

[2] B. Awerbuch, A. Goldberg, M. Luby and S. Plotkin, Network decomposition and locality in distributed computation. in *Proc. 30th IEEE Symp. on Foundations of Computer Science*, pp. 364-369, 1989.

[3] S. Arora, C. Lund. Hardness of Approximation, in *Approximation Algorithms for NP-hard Problems*, D. Hochbaum Ed., pp. 399-440, (PWS Publishing Company), 1997.

[4] A. Czygrinow, M. Hanckowiak, and M. Karonski, Distributed O(Delta logn)-edge-coloring algorithm, in *Proc. 9th Annual European Symp. on Algorithms*, LNCS 2161, pp. 345-355, 2001.

[5] D. Bertsekas and R.G. Gallager, *Data Networks*. 2nd Edition. Prentice-Hall International, London, 1992.

[6] A. Chakrabarti, B. Chazelle, B. Gum and A. Lvov, A lower bound on the complexity of approximate nearest-neighbor searching on the Hamming cube, *Proc. 31st Ann. ACM Symp. Theory Comput.*, pp. 305-311, 1999.

[7] A. Czygrinow, M. Hanckowiak, E. Szymanska, Distributed Algorithm for Approximating the Maximum Matching, manuscript.

[8] F. Chin and H.F. Ting, An almost linear time and $O(n \log n + e)$ messages distributed algorithm for minimum-weight spanning trees, in *Proc. 26th IEEE Symp. on Foundations of Computer Science*, Los Alamitos, CA, 1985, pp.257-266.

[9] D. Dubhashi, A. Mei, A. Panconesi, J. Radhakrishnan, and A. Srinivisan, Fast Distributed Algorithm for (Weakly) Connected Dominating Sets and Linear-Size Skeletons. In *Proc. of the ACM-SIAM Symp. on Discrete Algorithms*, pp. 717-724, 2003.

[10] G. De Marco and A. Pelc, Near-optimal distributed edge coloring, *Proc. 3rd Annual European Symp. on Algorithms*, pp. 448-459, 1995.

[11] M. L. Elkin, A Faster Distributed Protocol for Constructing Minimum Spanning Tree, In *Proc. of the ACM-SIAM Symp. on Discrete Algorithms*, pp. 352-361, New Orleans, LA, USA, 2004.

[12] L. Engebrethsen and J. Holmerin, Towards optimal lower bounds for clique and chromatic number, in *Proc. to 27th Inter. Colloq. on Automata, Languages and Programming*, pp. 2-12, 2000.

[13] R.G. Gallager, P.A. Humblet and P.M. Spira, A distributed algorithm for minimum-weight spanning trees, *ACM Trans. on Programming Lang. and Syst.*, 5:66-77, 1983.

[14] J.A. Garay, S. Kutten and D. Peleg, A sublinear time distributed algorithm for minimum-weight spanning trees, *SIAM J. Comput.*, 27:302-316, 1998.

[15] O. Goldreich, S. Goldwasser, and D. Ron, Property testing and its connection to learning and approximation, In *Proc. of 37th IEEE Symp. on Foundations of Computer Science*, 1996, pp. 339-348.

[16] E. Gafni, Improvements in the time complexity of two message-optimal election algorithms, in *Proc. 4th Symp. on Principles of Distributed Computing*, ACM, New York, 1985, pp. 175-185.

[17] D. Grable and A. Panconesi, Nearly optimal distributed edge colouring in O(log log n) rounds, Random Structures and Algorithms, Vol. 10, No. 3, pp. 385-405, May 1997.

[18] L. Jia, R. Rajaraman, and R. Suel, An Efficient Distributed Algorithm for Constructing Small Dominating Sets, In *Proc. of the 20th ACM Symp. on Principles of Distributed Computing*, pp. 33-42, 2001.

[19] E. Kushilevitz and N. Nisan, *Communication Complexity*, Cambridge Press, 1996.

[20] S. Kutten and D. Peleg, Fast distributed construction of k-dominating sets and applications, *J. Algorithms*, 28:40-66, 1998.

[21] F. Kuhn and R. Wattnhofer, Constant-Time Distributed Dominating Set Approximation, In *Proc. of the 22nd ACM Symp. on Principles of Distributed Computing*, Boston, Massachusetts, USA, July 2003.

[22] N. Linial, Locality in distributed graph algorithms. *SIAM J. Comput.*, 21:193-201,1992.

[23] Z. Lotker, B. Patt-Shamir and D. Peleg, Distributed MST for Constant Diameter Graphs, In *Proc. 20th ACM Symp. on Principles of Distributed Computing*, pp. 63-72, Newport, Rhode Island, August, 2001.

[24] D. Peleg. *Distributed Computing: A Locality-Sensitive Approach*, SIAM, Philadelphia, PA, 2000.

[25] D. Peleg and V. Rubinovich, A near-tight lower bound on the time complexity of distributed MST construction, In *Proc. 40th IEEE Symp. on Foundations of Computer Science*, pp.253-261, 1999.

[26] A. Panconesi and A. Srinivasan, Randomized Distributed Edge Coloring via an Extension of the Chernoff-Hoeffding Bounds, SIAM J. on Computing, Vol. 26, No. 2, pp. 350-368, 1997.

[27] V. Rubinovich, Distributed minimum spanning tree construction. M. Sc. Thesis, Bar-Ilan University, Ramat-Gan, Israel, 1999.