Universal Approximations for TSP, Steiner Tree, and Set Cover*

Lujun Jia   Guolong Lin   Guevara Noubir   Rajmohan Rajaraman
Ravi Sundaram
College of Computer and Information Science
Northeastern University
Boston MA 02115.
{lujunjia,lingl,noubir,rraj,koods}@ccs.neu.edu.

July 22, 2009

Abstract

We introduce a notion of *universality* in the context of optimization problems with partial information. Universality is a framework for dealing with uncertainty by guaranteeing a certain quality of goodness for all possible completions of the partial information set. Universal variants of optimization problems can be defined that are both natural and well-motivated. We consider universal versions of three classical problems: TSP, Steiner Tree and Set Cover.

We present a polynomial-time algorithm to find a universal tour on a given metric space over n vertices such that for any subset of the vertices, the sub-tour induced by the subset is within $O(\log^4 n/\log\log n)$ of an optimal tour for the subset. Similarly, we show that given a metric space over n vertices and a root vertex, we can find a universal spanning tree such that for any subset of vertices containing the root, the sub-tree induced by the subset is within $O(\log^4 n/\log\log n)$ of an optimal Steiner tree for the subset. Our algorithms rely on a new notion of sparse partitions, that may be of independent interest. For the special case of doubling metrics, which includes both constant-dimensional Euclidean and growth-restricted metrics, our algorithms achieve an $O(\log n)$ upper bound. We complement our results for the universal Steiner tree problem with a lower bound of $\Omega(\log n)$ for metrics and $\Omega(\log n/\log\log n)$ for Euclidean metrics. We also show that a slight generalization of the universal Steiner Tree problem is coNP-hard and present nearly tight upper and lower bounds for a universal version of Set Cover.

1 Introduction

Consider a courier who delivers packages to different houses and businesses in a city every day. One challenge faced by the courier is to determine a suitable route every day, given the packages to be delivered that day. A natural question that the courier may ask is the following: is there a universal tour of all locations, such that for any subset, when the locations in that subset are visited in the order of their appearance in the universal route, then the resulting tour is close to optimal for that subset? Such a tour can be viewed as a *universal* TSP tour.

Moving to a much larger scale, consider Walmart, which has thousands of stores spread throughout the world. Headquarters in Bentonville, Arkansas, may often have a need to teleconference with various subsets of these stores. They may not wish to set up a new multicast network for each possible subset; instead they may wish to come up with one *universal* tree such that for any subset they simply restrict this tree to that subset to create the desired multicast network. And, they may wish to ensure that for every subset it is the

*A preliminary edition appears in STOC'2005, Baltimore, MD

case that the network so generated is not much more expensive than the optimal network (for the subset under consideration).

A universal solution to the Steiner tree problem described above is also useful for sensor networks where nodes have limited memory. Low cost trees are required for data aggregation and information dissemination for subsets of the sensor nodes. It is, however, not realistic to expect sensors to compute and memorize optimal trees for each subset. Universal trees provide a practical solution; a sensor node only needs to know its parent in the universal tree while being oblivious to the other nodes involved in the data movement.

A unifying theme among the above three scenarios is that each seeks the design of a single structure that *simultaneously approximates* an optimal solution for every possible input. We refer to such problems as universal problems and their solutions as universal approximations. Universal problems and approximations have applications in scenarios where the input is uncertain; such uncertainty may arise, for instance, due to limited knowledge about the future or limited access to global information that may be distributed among multiple sources.

1.1 Our results

In this paper, we introduce a notion of *universality* in the context of optimization problems with uncertain inputs, and study universal versions of classical optimization problems.

- We develop a general framework for universal versions of optimization problems. Our framework, which is described in Section 2, allows the definition of a universal version of any optimization problem, given two additional notions: a subinstance relation that is a partial ordering among instances, and a restriction function that takes a solution for a given instance and a subinstance, and returns a solution for the subinstance.

We formulate and study the Universal Traveling Salesman (UTSP), Universal Steiner Tree (UST), and Universal Set Cover (USC) problems. Our main technical results concern the UTSP and UST problems.

- For UTSP problem, we obtain a universal tour C for a given metric space over n vertices such that for any subset S of the vertices, the sub-tour of C induced by S is within $O(\log^4 n/\log\log n)$ of an optimal tour for S. For the special case of doubling metrics, which includes both constant-dimensional Euclidean and growth-restricted metrics, our algorithm yields an $O(\log n)$ bound. These results appear in Section 4.

- We adapt our UTSP algorithm to the UST problem, and show that given a metric space over n vertices and a root vertex, one can find a spanning tree T in polynomial-time such that for every subset S of vertices containing the root, the sub-tree of T induced by S is within $O(\log^4 n/\log\log n)$ of an optimal Steiner tree for S. As for UTSP, our algorithm achieves an $O(\log n)$ upper bound for doubling metrics. We complement these results with a lower bound of $\Omega(\log n/\log\log n)$ for UST that holds even when all the vertices are on the plane. These results appear in Section 5. Our algorithms for UTSP and UST both rely on a new notion of sparse partitions, defined in Section 3, that may be of independent interest.

- For USC, we show that given a weighted set cover instance with n elements, we can compute an assignment from elements to sets such that, for any subset of the elements, the weight of the sets to which the elements in the subset are assigned is within $O(\sqrt{n\log n})$ of the weight of an optimal cover for the subset. We improve the bound to $O(\sqrt{n})$ for unweighted USC and present a matching lower bound for this case. These results are described in Section 6.

Universal problems are naturally captured by Σ_2^P, the second level of the polynomial-hierarchy [34, 24], since they have the following form: there *exists* a solution for a given instance such that *for all* subinstances the solution (suitably modified) is (close to) optimal. We believe that UST and UTSP are, in fact, Σ_2^P-hard [35] and present preliminary evidence towards this conjecture.

- We establish the coNP-hardness of a slight generalization of UST in which the universal tree is required to connect a given subset of the vertices. We also establish that given a spanning tree, finding a subset for which the tree has worst-case performance is NP-hard. We discuss the complexity of UST, UTSP, and USC problems in Section 8.

We hope this work will stimulate the reexamination of classical problems in a universal context. It would be especially interesting to identify problems for which there exist universal algorithms that are almost as good (within a constant factor) as the best algorithms in the standard approximation framework.

1.2 Related work

The existing literature contains numerous approaches for dealing with the problems posed by an uncertain world. These include competitive analysis, stochastic optimization, probabilistic approximations, distributional assumptions on inputs and many others. Here we survey a fraction of this vast body of work.

The word "universal" itself has been used many times before, notably in the context of hash functions [8], and routing [36]. Here, universal has meant the use of randomization to convert a bad deterministic performance guarantee to a good expected solution; further the randomized solution is oblivious to (some aspects of) the input.

The study of online algorithms considers problems in which the input is given one piece at a time, and upon receiving an input, the algorithm must take an action without the knowledge of future inputs [7, 11, 33]. In contrast, a universal algorithm computes a single solution, whose performance is measured against all possible inputs. Several researchers have considered settings where a certain distribution over the space of input is assumed [22, 17, 20, 26]. The *a priori* traveling salesman problem is studied in [21] where the set of vertices to be visited is drawn from a probability distribution. Stochastic optimization, studied in [17, 9], is a variant where the input is allowed to be modified rather than just completed. In these situations the goal is to optimize the expectation over the input distribution. Recently, incremental variants of facility location problems have been introduced and studied in [23, 25, 29]. These are similar in spirit to the universal problems we consider in that they are oblivious to the number of facilities. In fact, these problems fit within the framework of Section 2. Similarly, the recent seminal results on oblivious routing [3, 30, 31], when viewed in terms of flows rather than routes, are analogous to the universal results in this paper; the oblivious routing solution is universal over all demand matrices much as the solutions in this paper are universal over all subinstances of a given problem. We note that oblivious flows are exactly computable in polynomial time [3], whereas the problems we study here are intractable and appear to be much harder.

Of particular relevance to our results on UST is the substantial body of work on tree-embeddings of metric spaces [4, 5, 10]. It follows from these results that one can construct a spanning tree over any metric of n vertices such that for any subset of the vertices, the *expected cost* of the subtree induced by the subset is within $O(\log n)$ of the optimal. From a technical standpoint, our UST results are incomparable; while we obtain a single tree, rather than a distribution over trees, that achieves a *deterministic* poly-logarithmic performance guarantee, our guarantee applies only for subsets containing a distinguished root. It is worth noting here that a version of UST without a fixed root does not admit an $o(n)$ performance guarantee. Also related to our work is [13], which constructs a single aggregation tree for a fixed set of sinks that simultaneously approximates the optimal for all concave aggregation functions.

For the special case of UTSP on the plane, our $O(\log n)$ bound also follows from an early work of Platzman and Bartholdi [28]. (See also the related work of [6].) Their result is, in fact, stronger than ours for this special case; they show that any space-filling curve within a certain class yields a solution with an $O(\log n)$ performance guarantee. Our overall results for UTSP are more general, however, since our $O(\log n)$ bound applies to doubling metrics, and we also obtain a polylogarithmic bound for arbitrary metrics.

As mentioned in Section 1.1, our UST and UTSP algorithms rely on a new notion of sparse partitions. Our definition is closely related to the sparse partitions and covers of Awerbuch and Peleg [27, 2]. Indeed, the sparse covers of [2] form an integral part of our partitioning scheme.

Finally, we note that subsequent to the initial publication of these results, several related results have been obtained. Improved bounds have been obtained for UTSP. In particular, [15] present an $O(\log^2 n)$ approximation for UTSP, while [18] present an $\Omega((\log n/ \log\log n)^{1/6})$ lower bound on the stretch of any UTSP solution on an $n \times n$ grid. New results on a priori and universal TSP on tree metrics have been obtained in [32]. In [14], stochastic variants of the universal set cover, universal facility location problem, and other covering problems have been studied.

2 A framework for universal approximation

In this section, we introduce a framework for universal approximation of optimization problems. Let Π denote any optimization problem. Let $\mathit{Insts}(\Pi)$ denote the set of instances of Π, and for any instance $I \in \mathit{Insts}(\Pi)$, let $\mathit{Sols}(I)$ denote the set of feasible solutions for I. For a feasible solution S of an instance, let $\mathit{Cost}(S)$ denote the cost of the solution.

We develop a universal version of Π in terms of two additional notions: a *subinstance* relation $\preceq$ and a *restriction* function R. The relation $\preceq$ is a partial order on $\mathit{Insts}(\Pi)$; we say that I' is a subinstance of I whenever $I' \preceq I$. A restriction R takes an instance I of Π ($I \in \mathit{Insts}(\Pi)$), a subsinstance I' of I ($I' \preceq I$), and a feasible solution S of I ($S \in \mathit{Sols}(I)$), and returns a feasible solution $R(I, I', S)$ of I' ($R(I, I', S) \in \mathit{Sols}(I')$). A universal version of Π is given by the triple Π, $\preceq$, and R.

We now define universal approximation. Fix a minimization problem Π and an associated subinstance relation $\preceq$ and a restriction R. Let I be an instance of Π and S be any feasible solution of I. We define the *stretch* of S for instance I as

$$\max_{I' \preceq I} \frac{\mathit{Cost}(R(I, I', S))}{\mathit{OptCost}(I')},$$

where $\mathit{OptCost}(I')$ is the cost of an optimal solution for I'. Let $\mathcal{A}$ denote an algorithm for Π; it takes as input an instance I and outputs a solution $S \in \mathit{Sols}(I)$. We say that $\mathcal{A}$ has a universal approximation of f for $\langle \Pi, \preceq, R \rangle$, where f is a function from positive integers to reals, if for every instance I of Π of size n sufficiently large, the stretch is at most $f(n)$. The definition of universal approximation can be extended to maximization problems by appropriately redefining the stretch.

3 Sparse partitions

We introduce a new notion of sparse partition, which is used in our algorithms for UST and UTSP.

Definition 1 ((r, σ, I)-partition) *A (r, σ, I)-partition of a metric space (V, d) is a partition $\{S_i\}$ of V such that (i) the diameter of every set S_i in the partition is at most $r \cdot \sigma$ and (ii) for every node $v \in V$, the ball $B_r(v)$ intersects at most I sets in the partition, where $B_r(v) = \{u \in V | d(u, v) \leq r\}$.*

A (σ, I)-**partition scheme** is a procedure that computes a (r, σ, I)-partition for any $r > 0$.

3.1 General metric spaces

We present a polynomial-time $(O(\log n), O(\log n))$-**partition scheme** for general metric spaces. This is obtained using the sparse cover construction of Awerbuch-Peleg [2]. A cover of some $U \subset V$ is defined to be a collection of subsets of V such that for any $v \in U$, there is a subset containing v in the collection. Given a metric space (V, d), a real r, and a cover $\{B_r(v) | v \in V\}$, Awerbuch and Peleg give a polynomial-time algorithm to compute a (coarsening) cover $\mathcal{C}$ such that: (1) for each $v \in V$, $B_r(v)$ is contained in at least one set in $\mathcal{C}$; (2) every vertex v is contained in at most $4\lceil \log n \rceil$ sets in $\mathcal{C}$; (3) each set in $\mathcal{C}$ has radius at most $2r\lceil \log n \rceil$. (See Theorem 3.1 of [2] for details.)

We compute a partition $\mathcal{P}$ from $\mathcal{C}$ as follows. For each v we select an arbitrary set $S(v)$ in $\mathcal{C}$ that contains $B_r(v)$. We set $\mathcal{P} = \{\{v : S(v) = S\} : S \in \mathcal{C}\} - \{\emptyset\}$.

Lemma 1 *The collection $\mathcal{P}$ is a $(r, 4\lceil \log n \rceil, 4\lceil \log n \rceil)$-partition.*

Proof: Since each v is assigned to a unique $S(v)$, $\mathcal{P}$ is a partition. Also, since every set in $\mathcal{P}$ is a subset of a set in $\mathcal{C}$, and every set in $\mathcal{C}$ has radius at most $2r\lceil \log n \rceil$ by property 3, every set in $\mathcal{P}$ has diameter $4r\lceil \log n \rceil$. It remains to show that for every node v, $B_r(v)$ intersects $\leq 4\lceil \log n \rceil$ sets in $\mathcal{P}$. Consider two distinct sets X and Y in $\mathcal{P}$ that intersect $B_r(v)$; let x and y be nodes in $X \cap B_r(v)$ and $Y \cap B_r(v)$, respectively. It follows that node v belongs to both $B_r(x)$ and $B_r(y)$, which are contained in $S(x)$ and $S(y)$, respectively. Since X and Y are distinct, so are $S(x)$ and $S(y)$. Thus the number of sets in $\mathcal{P}$ that intersect $B_r(v)$ is at most the number of sets in $\mathcal{C}$ that contain v, which is bounded by $4\lceil \log n \rceil$ by property 2 above. □

3.2 Special metric spaces

We present an improved partition scheme for doubling metric spaces, which include constant dimensional Euclidean spaces and growth-restricted metric spaces. A metric space (V, d) is called *doubling* if every ball in V can be covered by at most λ balls of half the radius [16]. The minimum value of such λ is called the *doubling constant* of the space.

Lemma 2 *For a doubling metric space (V, d) with doubling constant λ, a $(r, 1, \lambda^3)$-partition can be computed efficiently for any $r > 0$.*

Proof: Given the metric space (V, d) and r, we compute the partition as follows. Start from $i = 1$, pick some arbitrary $s_i \in V' = V$, let $S_i = \{v \in V' : d(s_i, v) \leq r/2\}$. $V' \leftarrow V' - S_i, i = i + 1$. Repeat until V' is empty. Let S be the collection of the *centers*, s_i, of the partition subsets. Obviously, $\forall x, y \in S, d(x, y) > r/2$. We verify the two conditions. The diameter of each partition subset is at most r by construction; As for the intersection condition, consider any ball $B_r(v), v \in V$, and assume that it intersects m partition subsets S_i. Now, $B_{2r}(v)$ completely contains these subsets, and it can be covered by at most λ^3 balls of radius $r/4$ due to the doubling property. But covering S, the m centers of these subsets, requires at least m balls of radius $r/4$. Hence $m \leq \lambda^3$. □

The existence of $(O(1), O(1))$-partition schemes for constant dimensional Euclidean and growth-restricted metric space follows from the lemma above. We obtain slightly better parameters by a more direct argument.

Lemma 3 *If the points are in k-dimensional Euclidean space, a $(\rho, \sqrt{k}(2 + \epsilon), 2^k)$-partition can be computed efficiently for any $\rho > 0$.*

Proof: Divide the space into k-cubes with edge size $(2 + \epsilon)\rho$, where ϵ is any positive real. Each k-cube is a potential set of the partition we want, each node v is assigned to some k-cube that contains it. It is easy to see that the resulting nonempty k-cubes form a $(\rho, \sqrt{k}(2 + \epsilon), 2^k)$-partition. □

It is known that growth-restricted metrics form a subclass of doubling metric space [16].

Lemma 4 *[16] A growth-restricted metric space (V, d) with expansion rate c is a doubing metric space with doubling constant $\lambda \leq c^4$.*

Lemmas 2 and 4 imply that for a growth-restricted metric space (V, d) with expansion rate c, a $(\rho, 1, c^{12})$-partition can be computed efficiently for any $\rho > 0$. We show slightly better parameters in the following lemma.

Lemma 5 *For a growth-restricted metric space (V, d) with expansion rate c, a $(\rho, 4, c^3)$-partition can be computed efficiently for any $\rho > 0$.*

Proof: Given the metric space (V, d) and ρ, we compute the partition as follows. Start from $i = 1$, pick some arbitrary $s_i \in V' = V$, let $S_i = \{v \in V' : d(s_i, v) \leq 2\rho\}$. $V' \leftarrow V' - S_i, i = i+1$. Repeat until V' is empty. Let S be the collection of the *centers*, s_i, of the partition subsets. Obviously, $\forall x, y \in S, d(x, y) > 2\rho$. We verify the two conditions. The diameter of each partition subset is at most 4ρ by construction. We now consider the intersection condition. Consider any ball $B_\rho(v), v \in V$, and assume that it intersects m partition subsets. Wlog, let the subsets be S_i centered at s_i, $i = 1, \ldots, m$. Let $B = \cup_{i=1}^m B_{2\rho}(s_i)$. For each s_i, we have $B \subset B_{8\rho}(s_i)$. Hence, by the growth-restriction property, $|B_\rho(s_i)| \geq |B_{8\rho}(s_i)|/c^3 \geq |B|/c^3$. But $B_\rho(s_i) \subset B$ and $B_\rho(s_i)$ is disjoint from $B_\rho(s_j)$ for $i \neq j$. Therefore,

$$|B| \geq \sum_{i=1}^{m} |B_\rho(s_i)| \geq m \cdot |B|/c^3$$

from which we conclude that $m \leq c^3$. This proves the intersection condition and hence the lemma. □

4 Universal TSP

We present a polynomial-time algorithm for UTSP that achieves polylogarithmic stretch for arbitrary metrics and logarithmic stretch for doubling metrics.

Definition 2 (UTSP) *An instance of* UTSP *is a metric space (V, d). For any cycle (tour) C containing all the vertices in V and a subset S of V, let C_S denote the unique cycle over S in which the ordering of vertices in S is consistent with their ordering in V. The stretch of C is defined as $\max_{S \subseteq V} \|C_S\|/\|OptTr_S\|$, where $OptTr_S$ denotes the minimum cost tour on set S. The goal of* UTSP *is to find a tour on V with minimum stretch.*

In the framework of Section 2, we have: (i) for any instance $I = (V, d)$, $Sols(I)$ is the set of Hamiltonian cycles over V; (ii) a subinstance of (V, d) is (V', d'), where $V' \subseteq V$ and d' is the restriction of d to V'; and (iii) for any instance $I = (V, d)$, subinstance $I' = (V', d')$, and solution C for I, $R(I, I', C)$ is $C_{V'}$.

Our polynomial-time algorithm, UTSP-ALG defined below, obtains a spanning tree by applying a subroutine CONSTRUCTTREE to the underlying set of vertices and then returns a tour obtained by traversing the vertices inorder, according to the tree. The construction of the spanning tree relies on a hierarchical decomposition of the vertices by iteratively applying the partitioning scheme introduced in section 3. Also, the output from CONSTRUCTTREE(U) can be viewed as a decomposition tree of U, where the vertices of U are the leaves located at the bottom level, and all the internal vertices (leaders) are *copies* of some vertices of U. Physical trees $T^{(i)}$ can be trivially obtained from the tree returned from CONSTRUCTTREE(U) by collapsing copies of each vertex. This view of a leveled decomposition tree is helpful for the analysis of UTSP and the presentation of our UST algorithm.

The following theorem is the main result of this section.

Theorem 1 *Given a metric space (V, d) with n vertices and a (σ, I)-partitioning scheme, $\sigma \geq 1$,* UTSP-ALG *returns a tour with stretch $O(\sigma^2 I \log_\sigma n)$ in polynomial time.*

Algorithm 1 UTSP-ALG

Input: Metric space (V, d).

1. $T = \textsc{ConstructTree}(V)$.
2. *Output:* Recall that T is a leveled decomposition tree where the vertices of V are located at the bottom level, and all the other vertices are (virtual) copies of some vertices of V. Return a tour C on V obtained by traversing T in a depth-first manner, starting from the root of the tree.

$\textsc{ConstructTree}(U)$

1. *Initialization.* Set D to diameter of U, $S_0 = U$, $j = 0$, and $T = \emptyset$.
2. *Levels of hierarchy*. While $|S_j| > 1$ do
   a. Using a (σ, I)-partitioning scheme, compute a (r_j, σ, I)-partition $\mathcal{P}$ of S_j, with $r_j = \min\{D, \rho^j\}$, where $\rho = 4\sigma$.
   b. For every set X in $\mathcal{P}$, select an arbitrary vertex in X as leader(X); add to T an edge between each vertex v in S_j and the leader of the set in $\mathcal{P}$ that contains v. If $v =$ leader(X), the edge between them is virtual and of cost zero.
   c. Set $S_{j+1} = \{\text{leader}(X) : X \in \mathcal{P}\}$, and $j = j + 1$.
3. Return T.

It is clear that the above algorithm is polynomial-time. We assume, without loss of generality, that the minimum distance between any pair of vertices is 1.

We first analyze the procedure $\textsc{ConstructTree}$, called on an input set U of vertices. For any vertex v in U and j, we define $\ell(v, j)$ as the unique vertex in S_j that is an ancestor of v in T. Note that $\ell(v, 0)$ is v. We place an upper bound on $d(v, \ell(v, j))$ as follows. Since the cost of an edge between a vertex in S_k and its parent vertex in S_{k+1} is at most $\rho^k\sigma$ (by property (i) of the partition), we obtain

$$d(v, \ell(v, j)) \leq \sum_{k=0}^{j-1} d(\ell(v, k), \ell(v, k + 1)) \leq \sum_{k=0}^{j-1} \rho^k\sigma \leq \frac{3}{2}\rho^{j-1}\sigma \qquad (1)$$

(In the last inequality, we use $\rho = 4\sigma \geq 3$.) Let T denote the tree returned by $\textsc{ConstructTree}(U)$. Let S be any subset of U. Our analysis of the stretch achieved by UTSP-ALG relies on an upper bound on the cost of T_S, which is obtained by bounding the cost of the edges at each level of T_S separately as follows. For any $j \geq 1$, let PS_j be a maximal subset of vertices of S that are pairwise separated by distance at least $\rho^{j-1}\sigma$.

Lemma 6 *For $j \geq 2$, the cost of edges in T_S at level j is at most $|PS_{j-1}|\rho^j\sigma I$.*

Proof: Consider the set $X = \{\ell(v, j - 1) : v \in PS_{j-1}\}$ and $Y = \{\ell(v, j - 1) : v \in S\}$. Note that X is a subset of Y, which is a subset of S_{j-1}. Since PS_{j-1} is a maximal subset with pairwise separation at least $\rho^{j-2}\sigma$, each vertex in S is within $\rho^{j-2}\sigma$ of some vertex in PS_{j-1}. By Equation 1, each vertex v in S (resp., PS_{j-1}) is within $1.5\rho^{j-2}\sigma$ of $\ell(v, j - 1)$, which lies in Y (resp., X). Therefore, each vertex in Y is within $4\rho^{j-2}\sigma$ of a vertex in X (for n, sufficiently large). Thus the balls of radius $\rho^{j-1} = 4\rho^{j-2}\sigma$ around the vertices in X cover Y. Consider the partitioning of S_{j-1}. The balls of radius ρ^{j-1} around vertices in X, taken together, intersect all the sets in the partition that contain the vertices in Y. By property (ii) of the partitioning scheme, the number of such sets is at most $|X|I$. The total cost of edges at level j is at most $|X|I\rho^j\sigma \leq |PS_{j-1}|\rho^j\sigma I$. □

The following technical lemma is useful in our analysis.

Lemma 7 *If $m_i \geq 1, i = 1, \ldots, k$, $c \geq 2$, then*

$$\sum_{i=1}^{k} m_i \cdot c^i \leq \left(\log_c(\max_i m_i) + 3\right) \cdot \max_i(m_i \cdot c^i)$$

Proof: Let $M = \max_i m_i$, $N = \max_i(m_i \cdot c^i)$, and $i_0 = \lceil \log_c \frac{N}{M} \rceil$. Since $c^k \leq m_k \cdot c^k \leq N$, we have $k \leq \log_c N$. Now

$$\begin{aligned}
\sum_{i=1}^{k} m_i \cdot c^i &= \sum_{i=1}^{i_0-1} m_i \cdot c^i + \sum_{i=i_0}^{k} m_i \cdot c^i \\
&\leq 2M \cdot c^{i_0-1} + (k - i_0 + 1)N \\
&\leq 2N + (\log_c N - \log_c \frac{N}{M} + 1)N \\
&= (\log_c M + 3)N
\end{aligned}$$

□

Proof of Theorem 1: Let $S \subset V$, T be the tree constructed by CONSTRUCTTREE(V), and T_S be the induced subtree of T by S. It is easy to see that C_S can be obtained directly from T_S by an inorder walk from root of T_S. Hence the cost of C_S is at most twice that of $\|T_S\|$. In fact we can say something more general. Let S_j denote the set of vertices at level j of T_S and let $m_j = |S_j|$. For any $k \geq 1$, if the pairwise distance between any two vertices in S_k is at most B, then

$$\|C_S\| \leq m_k \cdot B + 2\sum_{j=0}^{k-1} m_j r_j \sigma \qquad (2)$$

For $j = 0$ or 1, $m_j r_j \sigma$ is $O(|S|\sigma^2)$, which is $O(\sigma^2 \|\text{OptTr}_S\|)$. Recall that for any $j \geq 1$, PS_j is a maximal subset of vertices of S that are pairwise separated by distance at least $\rho^{j-1}\sigma$. Let j^* denote the smallest value of j at which $|PS_{j-1}| = 1$; if no such j exists, then let j^* be the value of j at the root of T_S. By Lemma 6, we obtain that for $j < j^*$, $m_j r_j \sigma$ is at most $|PS_{j-1}|\rho^j \sigma I$. On the other hand, the cost of an optimal tour over S has cost at least $|PS_{j-1}|\rho^{j-2}\sigma$. Thus, for $j < j^*$, $m_j r_j \sigma$ is $O(\|\text{OptTr}_S\|\sigma^2 I)$. Plugging this bound into Equation 2 with $k = j^*$ and invoking Lemma 7, we obtain

$$\|C_S\| \leq m_{j^*} B + O(\log_\rho |S| \cdot \sigma^2 I \times \|\text{OptTr}_S\|),$$

where B is the maximum pairwise distance between vertices in X_{j^*}.

By the definition of j^*, either $|PS_{j^*-1}| = 1$ or $|S_{j^*}| = 1$. In the former case, B is at most $(1.5 + 2 + 1.5)\rho^{j-1}\sigma = 5\rho^{j-1}\sigma$, and $m_{j^*}B \leq 5m_{j^*-1}\rho^{j-1}\sigma = 5m_{j^*-1}r_{j-1}\sigma$. In the latter case, $m_{j^*} = 1$ and $B = 0$. Thus, we have $m_{j^*}B$ is $O(\sigma^2 I \|\text{OptTr}_S\|)$. Therefore, $\|C_S\|$ is $O(\log_\rho |S| \cdot \sigma^2 I \times \|\text{OptTr}_S\|)$, completing the proof of the theorem. □

Applying the parameters from Lemmas 1, 2, 3 and 5 to Theorem 1, we derive the stated bounds on the stretch achieved by UTSP-ALG in general and special metric spaces.

Corollary 1 *For any metric space over n vertices, the algorithm* UTSP-ALG *returns a tour with stretch* $O(\log^4 n / \log\log n)$.

Corollary 2 *For any doubling, Euclidean, or growth-restricted metric space over n vertices,* UTSP-ALG *returns a tree with stretch* $O(\log n)$.

We remark here that if the underlying vertices (V) are in Euclidean space, we can give a succinct description of our universal ordering, which is, in fact, independent of V. This property is a key aspect of the work of [28] on the use of space-filling curves for TSP on the plane.

5 Universal Steiner trees

We present a polynomial-time algorithm for UST that achieves polylogarithmic stretch for arbitrary metrics and logarithmic stretch for doubling metrics (Section 5.1), and also derive a nearly logarithmic lower bound for the optimal stretch achievable in the Euclidean plane (Section 5.2).

We begin by introducing some notation and definitions. Given a metric space (V, d), where V is the underlying set of vertices and d is the metric distance function over V, let $\Delta = \max_{u,v \in V}\{d(u, v)\}$ denote the diameter of V and $\nu = \lfloor \log(\Delta) \rfloor$. We assume, without loss of generality, that the minimum distance between any pair of vertices is 1. For any graph $G = (V, E)$ over the vertices in the metric space (V, d), we define the cost of G, $\|G\|$, to be the sum of the distances of the edges of G according to the metric d; that is, $\|G\| = \sum_{(u,v) \in E} d(u, v)$. For any tree T spanning V and subset S of V, let T_S denote the minimal subtree of T that connects S. For notational convenience, we use $S + x$ to denote $S \cup \{x\}$, for any set S. We denote by OptSt_S a minimum Steiner tree spanning S.

Definition 3 (UST) *An instance of the Universal Steiner Tree (*UST*) problem is a triple $\langle V, d, r\rangle$ where (V, d) forms a metric space, and r is a distinguished vertex in V that we refer to as the* root. *For any spanning tree T of V, define the* stretch *of T as $\max_{S \subseteq V} \|T_{S+r}\|/\|OptSt_{S+r}\|$. The goal of the* UST *problem is to determine a spanning tree with minimum stretch.*

In the framework of Section 2: (i) the set of solutions of any instance I, $Sols(I)$, is the set of spanning trees; (ii) a subinstance of a UST instance $\langle V, d, r\rangle$ is a triple $\langle V', d', r\rangle$, where V' is a subset of V that contains r, and d' equals d restricted to the subset V'; and (iii) for any instance $I = \langle V, d, r\rangle$, a spanning tree T of V, and any subinstance $I' = \langle V', d', r\rangle$, $R(I, I', T)$ is given by $T_{V'}$.

5.1 A UST with polylogarithmic stretch

Our algorithm, UST-ALG defined below, begins by organizing the vertices of the metric space in "bands", according to the distance from the root, and then computing, for each band, a tree that spans the vertices within the band and the root using the subroutine CONSTRUCTTREE introduced in UTSP-ALG. We formalize the notion of bands in the following.

For a nonnegative integer i, we define a *band* of level i, denoted by Band_i, to be a set of vertices with distance from r of at least 2^i and less than 2^{i+1}; thus, $\text{Band}_i = \{v \in V | 2^i \le d(r, v) < 2^{i+1}\}$. A set S is said to be *banded* if all the vertices of S lie in Band_i, for some $i, 0 \le i \le \nu$. A tree is said to be *rooted* if it contains r. A rooted tree is said to be *banded* if all the vertices in $T - \{r\}$ lie in Band_i for some $i, 0 \le i \le \nu$. A rooted tree is said to be *bandwise* if it is the edge disjoint union of banded trees with at most one banded tree for each $i, 0 \le i \le \nu$.

Theorem 2 *Given a metric space (V, d) with n vertices, a root $r \in V$, and a (σ, I)-partitioning scheme for (V, d), a spanning tree with stretch $O(\sigma^2 I \log_\sigma n)$ can be constructed in polynomial time.*

Algorithm 2 UST-ALG

Input: Metric space (V, d) and a root r.

1. *Bandwise decomposition*. Partition V into bands, Band_i for $0 \le i \le \nu$.
2. *Bandwise tree*. For $0 \le i \le \nu$, $T^{(i)} = \text{CONSTRUCTTREE}(\text{Band}_i)$.
3. *Output.* Connect the root of $T^{(i)}$ to r, for $0 \le i \le \nu$, and return the union of the resulting trees.

It is clear that our algorithm is polynomial-time. The approximation guarantee of $O(\sigma^2 I \log_\sigma n)$ is obtained in two steps. We first show that for any subset S there exists an equivalent bandwise rooted tree whose value is within a constant factor of the optimal rooted Steiner tree on S. This allows us to restrict our attention only to banded sets. We then show that for any banded set U, $\textsc{ConstructTree}(U)$ returns a tree with cost within $O(\sigma^2 I \log_\sigma n)$ of the optimum.

Lemma 8 *For any subset S of V containing r, there exists a bandwise rooted tree spanning S with cost within a constant factor of $\|OptSt_S\|$.*

Proof: Let $S_i = (S \cap \text{Band}_i) + r$ for $0 \leq i \leq \nu$. Let $S_o = \cup_{i \text{ odd}} S_i$ and $S_e = \cup_{i \text{ even}} S_i$. Do an Euler walk on the tree OptSt_S that visits all vertices in S and split the walk into two trees, using shortcuts, one spanning S_o and the other spanning S_e. Let these trees be T_e and T_o. Since the Euler walk traverses each edge at most twice and shortcuts do not increase cost, we obtain that $\|T_e\| \leq 2\|\text{OptSt}_S\|$ and $\|T_o\| \leq 2\|\text{OptSt}_S\|$, i.e. $\|T_e\| + \|T_o\| \leq 4\|\text{OptSt}_S\|$. Observe that T_e and T_o are disjoint. Let $T = T_e \cup T_o$.

Define an *inter-band* edge to be any edge such that neither of the endpoints is the root and the two endpoints are not in the same band. Let $e = (u, v) \in T$ be such an inter-band edge. Let u be in Band_i and v in Band_j where $i \leq j - 2$. Note that $\|e\| \geq 2^{j-1}$. Consider the two edges (r, v) and (r, u). Note that $\|(r, u)\| \leq 2^{i+1} \leq 2^{j-1} \leq \|e\|$. Thus, $\|(r, u)\| + \|(r, v)\| \leq 2\|(r, u)\| + \|(u, v)\| \leq 3\|e\|$. Hence, if we remove e and replace it with the two edges (r, v) and (r, u), then we increase the cost by at most $3\|e\|$. Observe that while the resulting graph may not be a tree, it continues to be connected. We perform this operation of replacing every inter-band edge by two edges from the root to its endpoints to yield a graph that spans all the vertices in S and has cost at most $3\|T\|$. We select an arbitrary spanning tree T^* of the resulting graph. Tree T^* has no inter-band edges and hence is a bandwise rooted tree. Thus we have a bandwise rooted tree spanning S with cost at most $12\|\text{OptSt}_S\|$. □

Lemma 9 *Let T denote the rooted tree obtained after connecting the root of* $\textsc{ConstructTree}(Band_i)$ *to r, for some i, $0 \leq i \leq \nu$. For any subset $S_r \subseteq Band_i + \{r\}$ that contains r, $\|T_{S_r}\|$ is at most $O(\sigma^2 I \frac{\log |S_r|}{\log \sigma} \|OptSt_{S_r}\|)$.*

Proof: Wlog, we assume that $|S| \geq 2$. Let S be $S_r - \{r\}$. The cost of the edge connecting r to the root of $\textsc{ConstructTree}(\text{Band}_i)$ is clearly at most $2\|\text{OptSt}_{S_r}\|$. So we focus our attention on the remaining subtree T_S of T_{S_r}. Our algorithm starts with a radius of 1 and increases the radius at each level by a factor of ρ until the radius exceeds D, which is at most 2^{i+2}, after which $|S_j|$ will be 1. Consider the tree T_S, let m_j be the number of vertices at level j (e.g., $m_0 = |S|$). Using Lemma 7, $\|T_S\|$, the cost of tree T_S can be bounded as follows:

$$\sum_j m_j r_j \sigma \leq \sum_j m_j \rho^j \sigma \leq (\log_\rho m_0 + 3) \max_j m_j \rho^j \sigma \tag{3}$$

The strategy of our proof is to show that at each level j, the bound on the cost of the edges selected for T_S, $m_j \rho^j \sigma$, does not exceed $\|\text{OptSt}_{S_r}\|$ by more than a factor of $O(\sigma^2 I)$. Hence $\max_j m_j \rho^j \sigma \leq O(\sigma^2 I)\|\text{OptSt}_{S_r}\|$ and the total cost of T_S is $O(\sigma^2 I \frac{\log |S|}{\log \rho})$ times that of $\|\text{OptSt}_{S_r}\|$.

The arguments for $j = 0$ and $j = 1$ differ from those at level $j \geq 2$. Consider the level $j = 0$. Observe that $\|\text{OptSt}_{S_r}\| \geq |S|$ since there are at least $|S|$ edges in OptSt_{S_r} and each edge has cost at least 1. The cost of the edges of T_S at level 0 is at most $|S| r_0 \sigma = |S|\sigma$, while at level 1 is at most $|S| r_1 \sigma = 4|S|\sigma^2$. Therefore, the total cost of edges at both levels 0 and 1 is $O(\sigma^2 I \|\text{OptSt}_{S_r}\|)$.

For level $j \geq 2$, we have an upper bound on the cost of the edges of T_S from Lemma 6. We now place a lower bound on the cost of the optimal Steiner tree on S_r. If $|PS_{j-1}| > 1$, we derive a lower bound as:

$$\|\text{OptSt}_{S_r}\| \geq \|\text{OptSt}_S\| \geq (|PS_{j-1}| \rho^{j-2} \sigma)/2.$$

If $|PS_{j-1}| = 1$, we derive a lower bound as:

$$\|\text{OptSt}_{S_r}\| \geq 2^i \geq \rho^{j-1}/4 \geq |PS_{j-1}|\rho^{j-2}\sigma/2.$$

Using the bound from Lemma 6, we obtain that for $j \geq 2$, $m_j r_j \sigma \leq 2\rho^2 I \|\text{OptSt}_{S_r}\| = 32\sigma^2 I \|\text{OptSt}_{S_r}\|$. And using Equation 3 mentioned above, we obtain that $\|T_S\|$ is within a factor of $O(\sigma^2 I \frac{\log|S|}{\log \sigma})$ of $\|\text{OptSt}_{S_r}\|$. Since $|S_r| = |S| + 1$, the lemma is proved. □

Proof of Theorem 2: Putting Lemma 8 and Lemma 9 together, we can now prove the main theorem. Construct the bandwise rooted spanning tree T^* on the set V as specified in the algorithm above. Consider any subset S of the vertices containing r. First observe that by Lemma 8 there exists a bandwise rooted tree Z on the set S such that $\|Z\|$ is within a constant factor of $\|\text{OptSt}_S\|$. Let $S_i = (S \cap \text{Band}_i) + r$ for $0 \leq i \leq \nu$. Let Z_i denote the rooted banded subtree of Z spanning S_i for $0 \leq i \leq \nu$. By Lemma 9, $\|T^*_{S_i}\|$ is within $O(\sigma^2 I \log_\sigma |S_i|)$ of $\|\text{OptSt}_{S_i}\|$ and hence within $O(\sigma^2 I \log_\sigma |S_i|)$ of $\|Z_i\|$. But by definition $\|Z\| = \Sigma_{0 \leq i \leq \nu}\|Z_i\|$ and $\|T^*_S\| = \Sigma_{0 \leq i \leq \nu}\|T^*_{S_i}\|$. Hence by summing over all bands we get that $\|T^*_S\|$ is within $O(\sigma^2 I \log_\sigma |S|)$ of $\|Z\|$ and hence within $O(\sigma^2 I \log_\sigma |S|)$ of $\|\text{OptSt}_S\|$. □

We can instantiate Theorem 2 with parameters from Lemma 1, $\sigma = 4\lceil \log n \rceil$ and $I = 4\lceil \log n \rceil$, to derive the following corollary.

Corollary 3 *For any metric space over n vertices,* UST-ALG *returns a tree with stretch $O(\log^4 n / \log\log n)$.*

For special metrics, we can apply the parameters from Lemmas 2, 3 and 5 to derive the following bound.

Corollary 4 *For any doubling, Euclidean, or growth-restricted metric space over n vertices,* UST-ALG *returns a tree with stretch $O(\log n)$.*

As in the case of UTSP, for the special case of vertices in Euclidean space, we can give a succinct description of our universal tree, which is, in fact, independent of even the global set V of vertices.

5.2 A lower bound for UST

We exploit a straightforward relation between universal Steiner tree problem and online Steiner tree problem to prove a lower bound for UST.

Theorem 3 *There exists a metric over n vertices for which every spanning tree has an $\Omega(\log n)$ stretch. There exists a set of n vertices in two-dimensional Euclidean space, for which every spanning tree has an $\Omega(\frac{\log n}{\log\log n})$ stretch.*

We derive the above theorem from two results on the online Steiner tree problem, one on metrics [19] and the other on the plane [1].

Theorem 4 ([19], Theorem 1) *There exists a graph over n nodes for which every online Steiner tree algorithm has a competitive ratio of $\Omega(\log n)$.*

Theorem 5 ([1], Theorem 1.1) *No on-line algorithm can achieve a competitive ratio which is better than $\Omega(\log n / \log\log n)$ for the Steiner tree problem of n vertices in the plane, or even for n vertices in the n by n grid.*

We make use of the above two lower bounds. Given any algorithm A for contructing a UST with stretch s, we obtain an online algorithm as follows. Let v_1 be the first vertex given. Build a UST T spanning all the vertices with root v_1. For each vertex v_i given, connect it to the previous ones by following edges of T. Since T has a stretch of s, the competitive ratio thus achieved for the online Steiner tree problem is at most s. The lower bounds on the competitive ratio given in Theorems 4 and 5 thus yield the desired bounds on stretch in Theorem 3.

6 Universal set/vertex cover

In this section, we define the universal set cover problem and present nearly tight upper and lower bounds for the problem.

Definition 4 (Universal Set Cover (USC)) *An instance of* USC *is a triple* $\langle U, \mathcal{C}, c\rangle$, *where* $U = \{e_1, e_2, \ldots, e_n\}$ *is a ground set of elements,* $\mathcal{C} = \{S_1, S_2, \ldots, S_m\}$ *is a collection of sets, and c is a cost function mapping* $\mathcal{C}$ *to* Q^+. *We define an* assignment f *as a function from* U *to* $\mathcal{C}$ *that satisfies* $e \in f(e)$ *for all e in* U. *We extend the definition of* f *to apply to any* $S \subseteq U$ *as follows:* $f(S)$ *is the set* $\{f(e) : e \in S\}$. *We next define the cost of* $f(S)$, $\|f(S)\|$, *as* $\sum_{X \in f(S)} c(X)$ *and the* stretch *of an assignment* f *as* $\max_{S \subseteq U} \frac{c(f(S))}{Opt(S)}$, *where* $Opt(S)$ *is the cost of the optimal (minimum) set cover solution to* S. *And the goal is to compute an assignment* f *with minimum stretch.*

In the framework of Section 2, we have: (i) the set of solutions for any instance I, $Sols(I)$, is the set of assignments for I; (ii) a subinstance of a USC instance $\langle U, \mathcal{C}, c\rangle$ is a triple $\langle U', \mathcal{C}, c\rangle$ satisfying $U' \subseteq U$; and (iii) for any instance $I = \langle U, \mathcal{C}, c\rangle$, a assignment f for I, and any subinstance $I' = \langle U', \mathcal{C}, c\rangle$, $R(I, I', f)$ is given by f restricted to the domain U'.

Algorithm 3 USC-ALG

1. $D \leftarrow \emptyset$.
2. While $D \neq U$, do
   Find the set S that minimizes $\frac{c(S)}{\sqrt{|S-D|}}$; we refer to this ratio as the cost-effectiveness of S. For every $e \in S - D$, we set $f(e) = S$.
3. Output f.

Theorem 6 *For any* USC *instance with* n *elements,* USC-ALG *has a stretch of* $O(\sqrt{n \ln n})$.

Proof: Let S be an arbitrary subset of U and let $s = |S|$. We consider two cases. The first case is when S is in $\mathcal{C}$. Let k be the number of iterations performed by the algorithm in step 2, and let $S_1, S_2, \cdots, S_k$ be the sets selected in that order. For a given set S_i, let N_i (resp., n_i) be the number of elements in U (resp., S) that are assigned to S_i by the algorithm. That is, $N_i = |\{e \in U : f(e) = S_i\}|$ and $n_i = |\{e \in S : f(e) = S_i\}|$. Since S is always a candidate set, our selection of S_i according to the cost-effectiveness criteria implies that

$$\frac{c(S_i)}{\sqrt{N_i}} \leq \frac{c(S)}{\sqrt{s - n_1 - \cdots - n_{i-1}}}$$

By reordering, summing up, and invoking Schwarz inequality, we get

$$\begin{aligned}
& \frac{c(S_1) + c(S_2) + \cdots + c(S_k)}{c(S)} \\
\leq\ & \frac{\sqrt{N_1}}{\sqrt{s}} + \frac{\sqrt{N_2}}{\sqrt{s - n_1}} + \cdots + \frac{\sqrt{N_k}}{\sqrt{s - n_1 - \cdots - n_{k-1}}} \\
\leq\ & \sqrt{\sum_{i=1}^{k} N_i} \cdot \sqrt{\frac{1}{s} + \frac{1}{s - n_1} + \cdots + \frac{1}{s - n_1 - \cdots - n_{k-1}}} \\
\leq\ & \sqrt{n} \times O(\sqrt{\ln s}) \\
=\ & O(\sqrt{n \ln n})
\end{aligned}$$

We now consider the second case when $S \notin \mathcal{C}$. Let $S_1, S_2, \cdots, S_k$ be an optimal collection of sets that together cover S. From the first case, we know that $c(f(S_i)) \leq O(n \ln n) c(S_i)$. Hence $c(f(S)) \leq \sum_i c(f(S_i)) \leq O(n \ln n) \sum_i c(S_i)$. □

For the special case of USC in which every set in the collection $\mathcal{C}$ has the same cost, a slightly more careful analysis of USC-ALG achieves an upper bound of $\sqrt{2n}$.

The proof is similar to that for USC and we just provide a brief sketch here. We only consider the first case. The second case carries over similarly from the first as in USC. Using the same notations as in the proof of Theorem 6, we obtain for the first case:

$$\begin{aligned} \frac{1}{\sqrt{N_i}} &\leq \frac{1}{\sqrt{n_i + n_{i+1} + \cdots + n_k}} \\ N_i &\geq n_i + n_{i+1} + \cdots + n_k \geq k - i + 1 \end{aligned}$$

Summing them up, we have

$$n \geq N_1 + \cdots + N_k \geq k(k+1)/2 \geq k^2/2$$

Hence, $\frac{c(f(S))}{Opt(S)} = k \leq \sqrt{2n}$.

Theorem 7 *There exists an n-element instance of* USC *with uniform costs for which the best stretch achievable is $\Omega(\sqrt{n})$.*

We provide two different proofs. The second proof shows that the natural definition of the *universal vertex cover* problem also has a lower bound of $\Omega(\sqrt{n})$.

Proof: Let q be some prime number between $\sqrt{n}/2$ and $\sqrt{n}$, whose existence is justified by the well known Bertrand postulate. We now describe the n elements of the ground set U. We include q^2 elements, each represented by (x, y), for all x and y belonging to the finite field Z_q. We also include an additional $n - q^2$ elements, denoted by $e_1, \ldots, e_{n-q^2}$, respectively, to complete the definition of U.

We now describe the covering set collection $\mathcal{C}$. Consider the collections of subsets defined as follows:

$$S_{a,b,c} = \{(x, y) \in U : x \in Z_q, y = P_{a,b,c}(x)\}$$

where $a, b, c \in Z_q$ and $P_{a,b,c}(x) = ax^2 + bx + c$ is a polynomial of degree at most 2 over Z_q which uniquely identifies $S_{a,b,c}$. With a, b, c ranging over Z_q, we obtain q^3 distinct subsets of $\mathcal{C}$. We also add one more subset $S_0 = \{e_i : 1 \leq i \leq n - q^2\}$ to complete the definition of $\mathcal{C}$. And let the cost of each subset be 1.

Let f be any assignment for the above USC instance. We focus our attention on the q^2 elements. Since each element of U is assigned to a single subset of $\mathcal{C}$ and $q^3 > q^2$, we know that at least one of $S_{a,b,c}$ is not assigned to by f. The optimal cost for $S_{a,b,c}$ is 1. Since no two distinct polynomials of degree at most 2 can intersect at more than 2 points, $S_{a,b,c}$ does not intersect with any other $S_{a',b',c'}$ on more than 2 elements. Therefore, the actual cost incurred by the assignment f is $c(f(S)) \geq \frac{q}{2}$. This proves that the achievable stretch is lower bounded by $q/2 = \Omega(\sqrt{n})$. □

Another Proof. Recall that the vertex cover problem is a special case of the set cover problem, where each vertex (edge) is respectively a set (element. Our current proof considers an example from the vertex cover problem.

Consider a $\sqrt{n} \times \sqrt{n}$ complete (undirected) bipartite graph. For any assignment f of the edges to the vertices, we orient the edges toward the vertices assigned to. For example, if $e = \{u, v\}$ is assigned to u, we make the edge e a directed one by pointing to u.

We count the total edge number in two ways. The first way is the sum, over the vertices, of the out-degree of each vertex. The second way is simple: $\sqrt{n} \times \sqrt{n} = n$. Now there are $2\sqrt{n}$ vertices. According to the Pigeon-hole principle, there exists a vertex r with out-degree at least $\sqrt{n}/2$. From our definition of the orientation of the edges, these $\sqrt{n}/2$ edges are assigned to different vertices with cost $\sqrt{n}/2$, while the optimal cost is 1 since they are coincident on r. This incurs a stretch of $\sqrt{n}/2$, which completes the proof of the theorem. □

Thus our lower bound is within a constant factor of the upper bound for the unweighted USC and within an $O(\sqrt{\log n})$ factor in general.

7 Universal facility location

In this section, we define one version of the universal facility location problem and prove a polynomial lower bound for this problem.

Definition 5 (Universal Facility Location(UFL)) *An instance of the* UFL *problem has the same input as the regular facility location problem. We have a metric space, with demands on the client points C, and facility costs on the facility points F. For a given mapping $p : C \to F$, define the* stretch *to be the maximum, over all subsets S of the clients, of the ratio*

$$\frac{\text{facility-cost}(p(S)) + \text{service-cost}(p, S)}{\mathit{OptSt}_S},$$

where facility-cost$(p(S))$ *is the facility cost of opening the facility set $p(S)$ and* service-cost(p, S) *is the serving (connection) cost of S using p. And OptSt_S is the optimum solution cost of serving S, under the definition of the regular facility location problem. The goal is to find a mapping with minimum stretch.*

Theorem 8 *There exists an instance of* UFL *for which of best stretch is $\Omega(n^{1/6})$.*

Proof: (Wlog, assume the quantities involved are all whole numbers.) Let $k = \sqrt{n}$, the example is as follows. k nodes on a ring, with neighboring distance 1. For each node on the ring, connect k nodes directly to it via edges of distance 1. Complete the remaining metrics by the shortest path metrics. Now, the k nodes on the ring are the set of facilities, each with opening cost $\sqrt{n}$. The k^2 hanging nodes are the client cities. (So the graph is the product of a ring by a star.)

For any mapping p, consider any particular facility f and its “children” client cities $c_1, \cdots, c_k$. If all c_i get mapped to a facility at least s away from f (s to be specified below), then the stretch is at least

$$ks/2k = s/2 \quad \text{(service cost is at least } ks\text{, opt is } 2k\text{)}.$$

Otherwise, partition the ring into segments of size $2s$. For each segment's center facility, there is a “child” city being mapped to a facility within its corresponding segment.

The subset S of these cities is of size $k/2s$. The mapping p is 1-to-1 on this subset. Thus the cost is at least $(k/2s) \cdot k$ (facility cost). Let uss now put an upper bound on the optimal solution. We could group y (to be specified below) consecutive segments together, and assign their cities to one common facility. The cost is at most $(y \cdot 2sy + k) \cdot (k/2sy)$. Let $y \cdot 2sy = k$. The stretch is at least $y/2$.

Letting $s/2 = y/2$ and combining with the identity $y \cdot 2sy = k$, we conclude that $y/2$ (hence the stretch) is at least $\frac{1}{4}k^{1/3} = \frac{1}{4}n^{1/6}$. This completes the proof. □

8 On the complexity of UST, UTSP, and USC

In this section, we analyze the complexity of the universal problems studied in this chapter. We begin by establishing the co-NP-hardness of a slight generalization of the UST problem in which the input terminals are constrained to be selected from a specified subset of nodes.

Definition 6 (TCUST) *We are given a metric (V,d), a $r \in V$, a set $U \subseteq V$ (of allowed terminals), a bound $B \in Q^+$. Is there an undirected tree T that connects r, U, and possibly other vertices, such that*

$$\max_{W \subseteq U} \frac{\|T_{W \cup \{r\}}\|}{\|OptSt_{W \cup \{r\}}\|} \leq B?$$

(Recall that for any set X of edges, $\|X\|$ is the sum of the metric distances of the edges and for any set S, OptSt_S is an optimal Steiner tree for S.)

Theorem 9 TCUST *is coNP-hard.*

Proof: Our proof is by a reduction from the unweighted undirected minimum Steiner tree problem UNWEIGHTED-STEINER-TREE [12, page 208]. An instance of the UNWEIGHTED-STEINER-TREE problem consists of an undirected graph $G = (V,E)$, a subset $R \subseteq V$, and a positive integer k, and we are asked whether there exists a subtree of G that includes all the vertices of R and uses no more than k edges.

We now describe the reduction from UNWEIGHTED-STEINER-TREE to TCUST. Given an UNWEIGHTED-STEINER-TREE instance $G_s = (V_s, E_s)$ and an $R \subseteq V_s$ and a k (let $n = |V_s|, m = |R|$), we construct a metric space (V,d) of TCUST as follows. The vertex set V includes the set V_s, a new vertex r, and for each $r_i \in R, i = 1, \ldots, m$, a pair of new vertices, x_i, y_i. For convenience, we denote the collection of all x_i's and y_i's as X and Y respectively. We now describe the distance function d. We classify the edges of the complete graph over V into two categories.

- Physical edges: For each $(u,v) \in E_s$, we set $w(u,v) = 1$. We set $w(r, r_1) = L = n^4$. $w(r_i, x_i) = w(r_i, y_i) = 2C, w(x_i, y_i) = C$, where $C = n^2$.
- Virtual edges: For any other edge (u,v) whose weight has not been defined, we set $w(u,v)$ to the lightest path weight between u and v using only physical edges.

By construction, the function d is a metric. To complete the construction of an instance of TCUST we need to specify the root, the set U of allowable terminals, and the bound B. We take r as the root, and take $R \cup X \cup Y$ as the set U. Let the bound $B = \frac{L+k+1+4C \cdot m}{L+k+1+3C \cdot m}$. This completes the reduction, which is clearly polynomial time.

We now prove that there exists a tree for the TCUST instance with stretch at most B if and only if the minimum Steiner tree for the UNWEIGHTED-STEINER-TREE instance has at least k edges, thus establishing the coNP-hardness of TCUST.

Let $ST(G_s)$ be a minimum Steiner tree for the UNWEIGHTED-STEINER-TREE instance. Consider the tree

$$T = ST(G_s) \cup \{(r, r_1)\} \cup \bigcup_{i=1}^{m} \{(r_i, x_i)\} \cup \bigcup_{i=1}^{m} \{(r_i, y_i)\}.$$

By construction, T connects all the terminals in U and the root r. In the following, we show that T is an optimal tree for the TCUST instance in the sense that it achieves minimum stretch. We first observe that for tree T, a subset $W \subseteq U$ that maximizes the stretch of T is $W = U$, and thus

$$p(T) = \frac{L + w(ST(G_s)) + 4C \cdot m}{L + w(ST(G_s)) + 3C \cdot m}$$

which is a decreasing function of n and equals 1 in the limit. We assume henceforth that n is large enough that $p(T) < 1.01$.

Now consider any tree T' that connects U and r. If T' contains two edges incident on r, then we let W consist of two vertices in U that belong to two different branches of T' rooted at r. The ratio $\|T'_{W \cup \{r\}}\|/\mathrm{OptSt}_{W \cup \{r\}}$ approaches 2 as n increases, implying that the stretch of T' is at least that of T.

In the remainder, we assume that T' contains only one edge adjacent to r. Consider some pair x_i, y_i, and let x'_i (resp., y'_i denote the first hop on the path in T' from x_i (resp., y_i) to r. Let x^*_i, (resp., y^*_i) denote the first hop on a lightest path, using only physical edges, from x_i to x'_i (resp., y_i to y'_i). If both x^*_i and y^*_i are equal to r_i, then we denote the scenario by $x_i \bowtie y_i$. If only $x^*_i = r_i$ (resp., $y^*_i = r_i$), then $y^*_i = x_i$ (resp., $x^*_i = y_i$); we denote such a scenario by $y_i \vdash x_i$ (resp., $x_i \vdash y_i$). It is easy to see that $x^*_i = y_i$ and $y^*_i = x_i$ can not happen at the same time.

We pick a subset $W \subseteq U$ for T' as follows. For each i: if $x_i \bowtie y_i$, we add both x_i and y_i to W; if $x_i \vdash y_i$, add y_i to W; otherwise ($y_i \vdash x_i$), we add x_i to W. Out of the m pairs of x_i, y_i, let t be the number of pairs such that $x_i \vdash y_i$ or $y_i \vdash x_i$. Note that $0 \leq t \leq m$. We now estimate the ratio of W on T'.

$$\begin{aligned} c_{T'}(W \cup \{r\}) &\geq L + w(ST(G_s)) + 4C \cdot (m-t) + 3C \cdot t \\ &= L + w(ST(G_s)) + 4C \cdot m - C \cdot t \\ \|\text{OptSt}_{W \cup \{r\}}\| &\leq L + w(ST(G_s)) + 3C \cdot (m-t) + 2C \cdot t \\ &= L + w(ST(G_s)) + 3C \cdot m - C \cdot t \end{aligned}$$

Hence $p(T') \geq p(T)$.

We have thus shown that the optimal stretch achievable for the TCUST instance is $p(T)$. Since $p(T)$ is a decreasing function of $w(ST(G_s))$, it follows that the optimal stretch for the TCUST instance is at most B if and only if the optimal Steiner tree for the UNWEIGHTED-STEINER-TREE instance has more than k edges. This completes the proof of coNP-hardness of TCUST. □

In studying the complexity of UST, a natural problem to consider is the following: given a spanning tree, determine the subset of vertices (containing the root) for which the tree has the worst performance, when compared with an optimal Steiner tree for the subset. The formal definition is as follows:

Definition 7 (Max Ratio Subset Problem (MRS)) *An instance of the MRS problem is a finite metric space (V, d), with vertex set V and metric function $d : (V, V) \to Q^+$, some spanning tree T, with edge weights specified by $d(\cdot, \cdot)$, a specified vertex $r \in V$ and a lower bound $B \in Q^+$. The decision question is whether there is a nonempty subset $W \subseteq V$, such that $\frac{c_T(W \cup \{r\})}{Opt(W \cup \{r\})} \geq B$, where $c_T(W \cup \{r\})$ is the cost of connecting W and r using only the edges of T, and $Opt(W \cup \{r\})$ is the cost of minimum spanning tree of $W \cup \{r\}$ in the sub-metric space $(W \cup \{r\}, d)$?*

Theorem 10 MRS *is NP-complete.*

Proof: Again, we prove the NP-completeness of this problem by reduction from UNWEIGHTED-STEINER-TREE problem [12]. Given an UNWEIGHTED-STEINER-TREE instance $G_s = (V_s, E_s)$ and an $R \subseteq V_s$ and a k (let $n = |V_s|$), we construct a metric space (V, d) of MRS as follows.

We introduce a new vertex r, and pick some arbitrary vertex $r_0 \in R$, connect r to any $v \in V$ by introducing new vertices, the number of which depends on v:

- if $v = r_0$, introduce $C - 1$ new vertices, use them to build a chain from r to v, such that $d(r, v) = C$. We choose $C = n^2$. For convenience, denote these new vertices introduced by S_1.
- if $v \neq r_0, v \in R$, introduce $L - 1$ new vertices, use them to build a chain from r to v, such that $d(r, v) = L$. We choose $L = 5n^5$. For convenience, denote the new vertices introduced for all $v \in R \setminus \{r_0\}$ by S_2.
- if $v \in V \backslash R$, introduce $D-1$ new vertices, use them to build a chain from r to v, such that $d(r, v) = D$. We choose $D = 2n^2$. For convenience, denote the new vertices introduced for all $v \in V \setminus R$ by S_3.

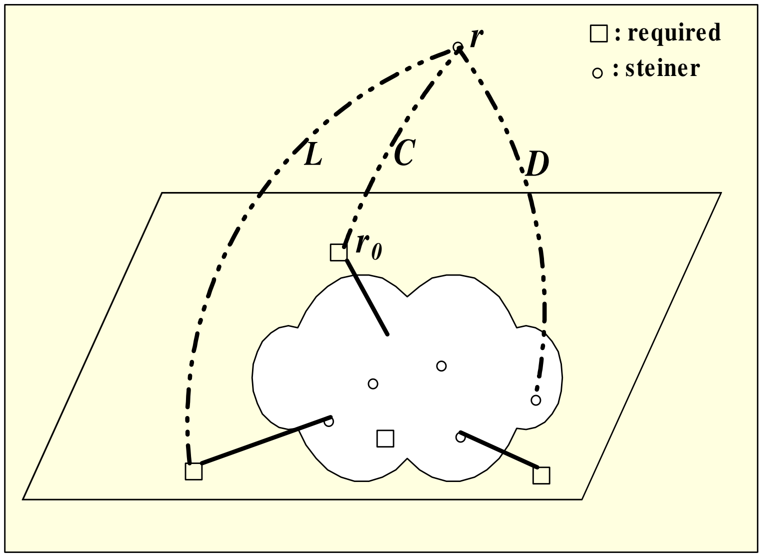

Figure 1: Reduction from Steiner tree to MRS

For the instance of MRS, we take the vertex set V as $V_s \cup S_1 \cup S_2 \cup S_3$, and the metric distance $d(v_1, v_2)$ is the length of the shortest path between them. r is the required vertex. The (star-like) tree T consists of the n chains from r to $v \in V_s$, and let the bound $B = \frac{C+L*(|R|-1)+D*(k+1-|R|)}{C+k}$.

It is obvious that the reduction can be performed within polynomial time.

We now argue that the vertices of the steiner tree connecting R in G_s is exactly one max ratio subset W of T.

It is an easy observation that the max ratio achievable is > 1. (consider a subset W consisting of the two end points of some edge $e \in E$.) We itemize our arguments below.

1. It is wlog to assume that W contains at most 1 vertex (excluding r) in each branch in T. Otherwise we can obtain a proper subset of W with no smaller ratio by only keeping the farthest vertices of W in each chain of T.

2. It can be assumed that $MST(W \cup \{r\})$ is of the form: first connect W by some $MST(W)$ (without using r) and then connect $MST(W)$ to r. Otherwise, there are multiple branches from r to the components of W. By averaging argument, the branch with the highest ratio has a ratio at least as large as that of W.

3. $MST(W)$ consists of (physical) unit edges only (i.e., no *induced path* involved). Otherwise, say $v_1, v_2 \in W$ are connected by some *induced path* with length ≥ 2, then including all the vertices on the path into W will increase $cost_T(\cdot)$ without increasing $Opt(\cdot)$, a contradiction.

4. $W \subseteq V$. Otherwise, say $w \in W$, and $w \in S_1 \cup S_2 \cup S_3$. Now consider the way w is connected to the other members of W in $MST(W)$ (Recall item 2). Note also that there exists at least one other vertex, otherwise, ratio $= 1$. w has to go down its chain according to items 2 and 3. Hence W will include at least one vertex in this chain that is farther than w from r, a contradiction to item 1.

5. Recall that $C = n^2, D = 2n^2, L = 5n^5$. For $MST(W \cup \{r\})$, $MST(W)$ will connect to r via a metric path through r_0. Using a similar argument as in item 3, we conclude that $r_0 \in W$.

6. $R \subseteq W$. If W contains m required vertices, where $m < |R|$, and b Steiner vertices (subset of $V \setminus R$). The ratio of W is at most
$$\frac{C + (m-1)L + bD}{C + m - 1 + b} < \frac{L}{n}$$
which can be verified by plugging in the values of C, D, L. This implies that by adding more required vertices to W, *Opt* (denominator) increased by at most n, while *cost* (numerator) increased by at least L, the ratio improves, a contradiction!

7. W includes as few vertices from $V \setminus R$ as possible. Again, it can be checked that
$$\frac{C + (m-1)L + bD}{C + m - 1 + b} > \frac{n \times D}{1}$$
So removing unnecessary Steiner vertices would raise the ratio. Hence we want to *remove* as many Steiner vertices as possible.

In conclusion, the max ratio subset W is a subset of V and contains the required vertex set R and as few Steiner vertices as possible, i.e., W is the minimum Steiner tree vertices connecting R in G_s. Let x be the cost of the minimum Steiner tree. W's ratio is
$$\frac{C + L * (|R| - 1) + D * (x + 1 - |R|)}{C + x}$$
Since the ratio is a decreasing function of x (see also item 7), the ratio is at least B if and only if $x \leq k$. This proves that MRS is NP-hard. It is not hard to see that MRS $\in$ NP. We conclude that MRS is NP-complete. □

Conjecture 1 *On the basis of Theorems 9 and 10, we conjecture that* UST *is* Σ_2^P*-hard.*

For the UTSP problem, our preliminary work suggests that the strategy of the coNP-hardness proof for the UST problem can be applied to a variant of UTSP in which a distinguished vertex has to be on every tour.

We finally show that USC is in NP. Consider the decision version of USC in which we are asked whether there exists a feasible assignment for a USC instance with stretch at most B, for a given number B. The upper bound proof for USC (Theorem 6) shows that the stretch for any assignment is, in fact, achieved on a set in C. Thus, the decision version of USC can be solved in non-deterministic polynomial time by first guessing the assignment and then verifying that it achieves the desired bound for each of the sets in C.

9 Open Problems

In this paper, we have introduced universal approximations, a new paradigm for approximation algorithms, and have studied universal approximations for three classic optimization problems: TSP, (rooted) Steiner trees, and set cover. There are a number of research directions that merit further study.

- *Tight bound for metric* UST: An immediate open problem for UST is to resolve the $\Omega(\log^3 n)$ factor gap between our upper and lower bounds, for general metric spaces.

- *Lower bound for* UTSP: We believe that the best stretch achievable for UTSP is at least logarithmic in the number of nodes, even for the Euclidean case. The best lower bound we have thus far, however, is a constant. In this regard, M. Grigni has posed a very interesting conjecture (presented here in terms of the notion of universality): Given n^2 points forming an $n \times n$ grid on the plane, every universal tour has a stretch of $\Omega(\log n)$.

- *A graph version of* UST: Our formulation of the UST problem assumes that the universal tree can include an edge between any two nodes of the underlying metric space. A natural variant that we are currently investigating is where the metric space is induced by an undirected weighted graph and the universal tree is required to include graph edges only. A plausible approach to solving this graph version of UST is to extend our partitioning scheme to graphs, a challenging problem that is of independent interest.
- *Complexity:* We have shown that USC is in NP, and have provided preliminary evidence that the UST and UTSP may be Σ_2^P-hard. Resolving the complexity of UST and UTSP is an important problem.
- *Universal approximations for other problems*: Finally, we believe that the universal approximations framework has the potential to yield insightful results on the approximability of diverse optimization problems, and plan to explore this line of research.

Acknowledgments

We would like to thank R. Ravi and A. Frieze for directing us to the work on Euclidean TSP tours using space-filling curves, and M. Goemans for pointers to some work on stochastic optimization. The authors Jia, Lin, and Rajaraman were partially supported by NSF awards Career CCR-9983901 and IIS-0330201, Noubir was partially supported by NSF Career award CNS-0448330, and Sundaram was partially supported by a grant from DARPA.

References

[1] N. Alon and Y. Azar. On-line Steiner trees in the Euclidean plane. In *Proceedings of the Eighth Annual ACM Symposium on Computational Geometry*, pages 337–343, 1992.

[2] B. Awerbuch and D. Peleg. Sparse partitions. In *Proceedings of the Thirty-First IEEE Symposium on Foundations of Computer Science (FOCS)*, pages 503–513, 1990.

[3] Y. Azar, E. Cohen, A. Fiat, H. Kaplan, and H. Racke. Optimal oblivious routing in polynomial time. In *Proceedings of the Thirty-Fifth ACM Symposium on Theory of Computing (STOC)*, pages 383–388, 2003.

[4] Y. Bartal. Probabilistic approximations of metric spaces and its algorithmic applications. In *Proceedings of the Thirty-Seventh IEEE Symposium on Foundations of Computer Science (FOCS)*, pages 184–193, 1996.

[5] Y. Bartal. On approximating arbitrary metrics by tree metrics. In *Proceedings of the Thirtieth ACM Symposium on Theory of Computing (STOC)*, pages 161–168, 1998.

[6] D. Bertsimas and M. Grigni. On the space-filling curve heuristic for the euclidean traveling salesman problem. *Operations Research Letters*, 8:241–244, October 1989.

[7] A. Borodin and R. El-Yaniv. *Online Computation and Competitive Analysis*. Cambridge University Press, Cambridge, UK, 1998.

[8] J. L. Carter and M. N. Wegman. Universal classes of hash functions. *Journal of Computer Systems and Sciences (JCSS)*, 18:143–154, 1979.

[9] B. Dean, M. Goemans, and J. Vondrak. Approximating the stochastic knapsack problem: the benefit of adaptivity. In *Proceedings of the Forty-Fifth IEEE Symposium on Foundations of Computer Science (FOCS)*, 2004.

[10] J. Fakcheroenphol, S. Rao, and K. Talwar. A tight bound on approximating arbitrary metrics by tree metrics. In *Proceedings of the Thirty-Fifth ACM Symposium on Theory of Computing (STOC)*, pages 448–455, 2003.

[11] A. Fiat and G. J. Woeginger, editors. *Online Algorithms: The State of the Art*. Springer, 1998.

[12] M. R. Garey and D. S. Johnson. *Computers and Intractability: A guide to the theory of NP-completeness*. W. H. Freeman, San Francisco, 1979.

[13] A. Goel and D. Estrin. Simultaneous optimization for concave costs: single sink aggregation or single source buy-at-bulk. In *Proceedings of the Fourteenth Annual ACM-SIAM Symposium on Discrete Algorithms (SODA-03)*, pages 499–505, 2003.

[14] F. Grandoni, A. Gupta, S. Leonardi, P. Miettinen, P. Sankowski, and M. Singh. Set covering with our eyes closed. In *FOCS*, pages 347–356, 2008.

[15] A. Gupta, M. T. Hajiaghayi, and H. Räcke. Oblivious network design. In *SODA*, pages 970–979, 2006.

[16] A. Gupta, R. Krauthgamer, and J. R. Lee. Bounded geometries, fractals, and low-distortion embeddings. In *Proceedings of the Forty-Fourth IEEE Symposium on Foundations of Computer Science (FOCS)*, 2003.

[17] A. Gupta, M. Pal, R. Ravi, and A. Sinha. Boosted sampling: Approximation algorithms for stochastic optimization. In *Proceedings of the Thirty-Sixth ACM Symposium on Theory of Computing (STOC)*, pages 417–426, 2004.

[18] M. T. Hajiaghayi, R. D. Kleinberg, and F. T. Leighton. Improved lower and upper bounds for universal tsp in planar metrics. In *SODA*, pages 649–658, 2006.

[19] M. Imase and B. Waxman. Dynamic steiner tree problem. *SIAM J. Discrete Math.*, 4:369–384, 1991.

[20] N. Immorlica, D. R. Karger, M. Minkoff, and V. S. Mirrokni. On the costs and benefits of procrastination: Approximation algorithms for stochastic combinatorial optimization problems. In *Proceedings of the Fourteenth Annual ACM-SIAM Symposium on Discrete Algorithms (SODA-04)*, pages 691–700, 2004.

[21] P. Jaillet. A priori solution of a traveling salesman problem in which a random subset of the customers are visited. *Operations Research*, 36:929–936, 1988.

[22] D. R. Karger and M. Minkoff. Building steiner trees with incomplete global knowledge. In *Proceedings of the Forty-First IEEE Symposium on Foundations of Computer Science (FOCS)*, pages 613–623, 2000.

[23] R. R. Mettu and C. G. Plaxton. The online median problem. In *Proceedings of the Forty-First IEEE Symposium on Foundations of Computer Science (FOCS)*, pages 339–348, 2000.

[24] A. Meyer and L. Stockmeyer. The equivalence problem for regular expresions with squaring requires exponential space. In *13th IEEE Symposium on Switching and Automata Theory*, pages 12–129, 1972.

[25] A. Meyerson. Online facility location. In *Proceedings of the Forty-Second IEEE Symposium on Foundations of Computer Science (FOCS)*, pages 426–433, 2001.

[26] M. Minkoff. *Approximation Algorithms for Combinatorial Optimization Under Uncertainty*. PhD thesis, M.I.T., Cambridge, MA, 2003.

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

[28] L. K. Platzman and I. J. J. Bartholdi. Spacefilling curves and the planar travelling salesman problem. *Journal of the ACM (JACM)*, 36(4):719–737, October 1989.

[29] C. G. Plaxton. Approximation algorithms for hierarchical location problems. In *Proceedings of the Thirty-Fifth ACM Symposium on Theory of Computing (STOC)*, pages 40–49, 2003.

[30] H. Racke. Minimizing congestion in general networks. In *Proceedings of the Forty-Third IEEE Symposium on Foundations of Computer Science (FOCS)*, pages 43–52, 2002.

[31] H. Räcke. Optimal hierarchical decompositions for congestion minimization in networks. In *STOC*, pages 255–264, 2008.

[32] F. Schalekamp and D. B. Shmoys. Algorithms for the universal and a priori tsp. *Oper. Res. Lett.*, 36(1):1–3, 2008.

[33] D. D. Sleator and R. E. Tarjan. Amortized efficiency of list update and paging rules. *Communications of the ACM*, 28(2):202–208, 1985.

[34] L. Stockmeyer. The polynomial-time hierarchy. *Theoretical Computer Science*, 3:1–22, 1977.

[35] C. Umans. The hardness of approximating Σ_2^P minimization problems. In *Proceedings of the Fortieth IEEE Symposium on Foundations of Computer Science (FOCS)*, pages 465–474, 1999.

[36] L. G. Valiant and G. Brebner. Universal schemes for parallel communication. In *Proceedings of the Thirteenth ACM Symposium on Theory of Computing (STOC)*, pages 263–277, 1981.