Approximation Algorithms Workshop June 13-17, 2011, Princeton

Universal Approximations in Network Design — 17 June, 2011

Rajmohan Rajaraman *Scribe: Scott Roche*

1 Overview

This lecture provides an overview of problems and results in the space of universal approximations. We provide some insight into current techniques, connect the problems to other related areas, and introduce open problems in the field. The primary motivation for the problems considered in this lecture is efficient network design under dynamic, adversarial network conditions.

2 Universal Approximations

The notion of universal approximations was formally introduced in [15], with a focus on three optimization problems: Traveling Salesman (TSP), Minimum Steiner Tree, and Minimum Set Cover.

2.1 Universal Traveling Salesman

Suppose there are two couriers, a ”standard” courier and and a ”lazy” courier. Every day, each courier is given a list of addresses to deliver packages to. Each day, however, the list each courier receives is only a subset of the set of all the addresses in the company database. Assume also that there is also a table which lists pairwise distances between every address.

The standard courier does the following: every day he receives a fresh list of addresses, from which he computes a TSP tour (only using the addresses he received that day). The lazy courier, being lazy, would like to avoid the work of having to compute a new tour every day. He decides to use all of the addresses in the database to compute a single TSP tour. He will then follow that tour every day, stopping only at the addresses in that day's assignment, and shortcutting wherever possible. For instance, if the single TSP tour computed over the set $\{A, B, C, D, E, F, G, H\}$ of addresses is $A \to B \to C \to D \to E \to F \to G \to H \to A$, and the list of addresses to visit on a given day is $\{A, D, E, G\}$, the lazy courier will travel along the route $A \to D \to E \to G$. Clearly, the computation done by the lazy courier each day is minimal, while the standard courier needs to solve a (possibly new) TSP problem every day. On the other hand, it is not hard to see that on some days the lazy courier will travel more than the standard courier for a given address list. This raises the following question: how good of an approximation is this universal tour (computed by the lazy courier) as compared to the method of computing a separate tour for each day's delivery assignment?

Formally, given a metric space (V, d), can we design a tour T over V that has minimum stretch, which is defined as

$$max_{S \subseteq V} \frac{\text{Cost}(T_S)}{\text{OPT}(S)}$$

where T_S is the subtour of T induced by S, Cost(X) is the length of tour X, and OPT(S) is the cost of an optimal tour for S.

2.2 Universal Steiner Tree (UST)

We can similarly formulate a universal approximation version of the Minimum Steiner Tree problem. We are given a weighted undirected graph $G(V, E)$ and a root vertex, r. The goal is to find a spanning tree T of G of minimum *stretch*, which is defined as

$$max_{S \subseteq V} \frac{\text{Cost}(T_S)}{\text{OPT}(S)}$$

where T_S is the subtree of T induced by S and root r, OPT(S) is the cost of an optimal tree connecting S to r, and $\text{Cost}(X) = \Sigma_{e \in X} w(e)$.

There are two main versions of this problem: *graphical UST* and the *metric UST*. In the graphical version, the tree T can only draw its edges from G. In the metric version, we work with the metric completion of any underlying graph; i.e., we can assume that G is the complete graph with the weight of edge (u, v) being the shortest path distance between u and v in the original graph. We can easily see that the best stretch achievable in graphical UST is at least as large as that achieved in the metric version.

2.3 Universal Set Cover

Here is a universal approximation version of minimum set cover. Given a set V of elements, a collection C of subsets of V, determine $f : V \mapsto C$ that satisfies the constraint that for all $x \in V$, $f(x)$ contains x, and minimizes stretch, which is defined as

$$max_{S \subseteq V} \frac{\text{Cost}(f(S))}{\text{OPT}(S)},$$

where Cost(X) is the sum of the costs of the sets in X and OPT(S) is the cost of a minimum-cost set cover for S.

3 General Framework, Motivation, and Summary of Relevant Results

3.1 Framework

We can now specify a general framework for universal approximations. The universal version of an optimization problem $\mathcal{P}$ has two additional notions :

- A **sub-instance** relation $\leq$

- A **restriction** function R, which takes a solution S for instance I, $I \leq I'$ and returns a solution $R(S, I, I')$ for I'

For a given instance I, the goal is to determine a solution T_I for instance I that minimizes the stretch, which is defined as

$$max_{I' \leq I} \frac{\text{Cost}(R(S, I, I'))}{\text{OPT}(I')},$$

where Cost(X) is the cost of the solution X and OPT(I') is the cost of a minimum-cost solution for instance I'.

3.2 Motivation

Under what circumstances would finding a universal approximation be a good strategy? A primary example is when one is trying to optimize under uncertain inputs; universal solutions are particularly robust against adversarial inputs. For example, consider the problem of trying to aggregate data in a sensor network. Data is being generated at multiple sensors and must be be aggregated at a single sink. In each "round" the input sources change, as well as the type of query. Dynamically setting up an aggregation tree on this network as all input factors change may be expensive or impractical. On the other hand, a Universal Steiner Tree can provide a good approximation across all query and input patterns. Universal solutions are also useful in the privacy and security, since they are differentially private [3].

3.3 Current Results

	Upper Bound	Lower Bound
UTSP	$O(\log^4 n/ \log\log n)$ [15] $O(\log^2 n)$ [12] $O(\log n)$ for doubling metrics [15]	$\Omega(\log n)$ [10, 3] $\Omega(\log^{\frac{1}{6}} n)$ Euclidean [13]
Metric UST	$O(\log^4 n/ \log\log n)$ [15] $O(\log^2 n)$ [12] $O(\log n)$ for doubling metrics [15] $O(\log n)$ for planar [13]	$\Omega(\log n)$ [15] $\Omega(\log n/ \log\log n)$ Euclidean [15]
Graphical UST	$2^{\tilde{O}(\log^{\frac{3}{4}} n)}$ [6] $2^{\tilde{O}(\sqrt{\log n})}$ for doubling graphs	Same as for metric UST
USC	Weighted: $O(\sqrt{n}\log n)$ [15] Unweighted: $O(\sqrt{n})$	$\Omega(\sqrt{n})$ [15]

Table 1: Universal Approximation Results

3.4 The "universal" landscape

The following is a summary of progress on universal approximation problems and related results.

- Perhaps, the earliest result in the space of universal approximations is due to Platzman and Bartholdi, who showed that space-filling curves satisfying certain technical conditions provide an $O(\log n)$-stretch universal TSP in the Euclidean plane [17].

- An early result in the area of network design that is similar in spirit to universal approximations is due to Goel and Estrin [8]. They studied the single-sink, buy-at-bulk problem: Given a graph G, a set of demands to be routed to a sink s, and a cost for each edge, the goal is to route all demands to minimize total cost. They gave a randomized construction of a tree that achieves expected $O(\log n)$-approximation for all concave cost functions. This has subsequently been improved to $O(1)$ in recent work by Goel and Post [9].

- The UST problem seeks a single tree that simultaneously approximates all minimum Steiner trees for a given root. A series of very important results on metric embeddings have shown that for every metric space, there exist tree metrics that well-approximate all pairwise distances in the original metric space. Here is a very brief summary of results in tree embeddings.

  – Results of [7] yield metric tree whose expected stretch for each set is $O(\log n)$.
  – An $O(\log n \log \log n)$ expected stretch is achieved using distribution over spanning trees [1, 5].

  Also related are the cut-based decompositions of [18, 19] that aim for a distribution over trees or a tree with a distribution over embeddings, which well-approximate all the cuts in the original graph.

- Gupta, Hajiaghayi, and Räcke formulated problems in oblivious routing and oblivious network design that share the notion of "universality": Given a graph, source-sink pairs, and a per-edge routing cost, determine routes that are oblivious to demand pairs and cost function [12].

Other related results include a-priori approximations for the TSP, in which a set of vertices to be visited is drawn from a probability distribution [20, 21], and stochastic set covering, in which the goal is to find a single mapping of elements to sets to minimize the expected cost of covering a randomly chosen subset of elements [11].

4 A closer look at Universal Steiner Tree (UST)

4.1 Metric UST

Recall the definition of the metric UST problem from above. Our goal is to find a spanning tree of the complete graph G (whose edge weights form a distance metric) with minimum stretch. Intuitively, what would make a good UST? Proceeding from root r, at each distance level from r we can view T as providing a clustering of G. If an adversary were to try and find the worst-possible nodes nodes to pick as the subset, S, it would look for notes that are "well separated" in T but "close" in G - that is nodes that appear in totally different clusters through T but are not far apart in G. Thus, to avoid this problem, the UST should cluster nodes in T so that each node's neighborhood does not intersect many clusters. Otherwise, the adversary could select several nodes from this neighborhood that lie in different clusters.

4.1.1 Partition using bounded, locally consistent partitions

Motivated by the above discussion, we introduce the notion of **bounded, locally consistent partitions**. We define an (α, β, R)-partition as a partition of the metric space such that the diameter of every cluster in the partition is at most αR and every ball of radius R around any node intersects at most β clusters. It can be shown that every n-node metric space has an $(O(\log n), O(\log n), R)$-partition for every R (see [4, 16] on sparse partitions).

We can put bounded locally consistent partitions together to form **hierarchical partitions**: We create a collection of partitions $\{P_i\}$ of G with the following properties:

- Partition P_i is an (α, β, R^i)-partition.
- Hierarchy: P_i is a refinement of P_{i+1} .
- Root padding: The cluster in P_i containing the root contains a ball of radius R^i around the root.

Using the sparse partitions of Awerbuch-Peleg, one can obtain a hierarchical $(O(\log n), O(\log n), O(\log n))$-partition for every metric space. We then have a metric UST algorithm:

Metric UST Algorithm [15]

- Compute a hierarchical $(O(\log n), O(\log n), O(\log n))$-partition
- For each level i, from lowest to highest:
  - For each level-i cluster
    - Select leader from leaders of its constituent level $i-1$ clusters (root is always made leader of its cluster)
    - Connect level i leader to level $i-1$ leaders

We sketch a proof that the stretch for the UST using this algorithm is at most polylog(n). Note that for level j, the cost in UST is $O(n_j \log^{j+1} n)$, where n_j is the number of level-j ancestors of nodes in S. To do this we need to prove the following main lemma:

Main Lemma. If P_j is a maximal set of nodes in S pair-wise separated by $\log^{j-1} n$, then $n_j = O(|P_j| \log^{j-1} n)$.

Proof Sketch. n_j is the number of nodes at level j of the induced tree, and P_j is the maximal set of nodes in S pair-wise separated by $\log^{j-1} n$. Any node v's ancestor at level j is within $O(\log^j n)$ cost of v. Therefore, an $O(\log^j n)$-ball around the ancestors of P_j at level j covers all n_j ancestors of S at level j. Thus by the partitioning scheme, it follows that n_j is $O(|P_j| \log n)$.

Since $\text{Cost}(\text{OPT}(S))$ is $\Omega(|P_j| \log^{j-1} n)$, the cost at level j in the UST is thus $O(\log^3 n)\text{Cost}(\text{OPT}(S))$. Since the number of levels in the hierarchical decomposition is $O(\log n/\log\log n)$, we obtain an $O(\log^4 n/\log logn)$-stretch UST.

4.1.2 An $O(\log^2 n)$ stretch for metric UST.

Gupta, Hajiaghayi, and Räcke obtain an improved bound for metric UST in [12]. We say that a node v is *α-padded* in a hierarchical decomposition if, at level i, the ball of radius $2\alpha^i$ around v is fully contained within its cluster at level i. Then we have the following theorem:

Theorem 1. *[12] For any v, in any tree drawn from the FRT distribution [7], probability that v is is $\Omega(1/\log n)$-padded is at least $3/4$.*

Using this theorem we can create a simple metric UST construction to give us an $O(\log^2 n)$ stretch:

1. Sample $O(\log n)$ trees from the FRT distribution.
2. For each vertex v select a tree where v is $\Omega(1/\log n)$-padded.
3. In each tree, build the sub-tree induced by the root and vertices that selected the tree (using metric completion).
4. Return the union of the $O(\log n)$ sub-trees computed above.

4.1.3 Improved bounds for doubling metrics

We define the *doubling dimension* of a metric to be the smallest σ such that every ball of radius $2r$ can be covered by 2^σ balls of radius r, for every r. A *doubling metric* is a metric that has constant doubling dimension. Euclidean metrics are the simplest examples of doubling metrics. For doubling metrics, the algorithm of [15] achieves a stretch of $O(\log n)$ through a hierarchical $(O(1), O(1), O(1))$ partition.

4.2 Graphical UST

Recall that in the graphical version of universal UST, we require that T can only draw its edges from G. How does this change the bounds achievable on the stretch? In the graphical case, we can natural extend the notion of bounded, locally consistent partition: we want to partition G into clusters of **strong diameter** at most αR and such that each R-ball intersects at most β clusters. We now ask, how small can α and β be? An intriguing open question is if $(\text{polylog}(n), \text{polylog}(n), 1)$-partition is achievable.

The following lemma states a necessary condition for a certain partition bound:

Lemma 2. *[6] If a σ-stretch is achievable for a graphical UST, then an $(O(\sigma), O(\sigma^2), R)$-partition exists for all R.*

We face two major challenges in applying the partition-based approach that was successful for metric MST. First, as mentioned above, we do not know yet how to obtain the desired partitions with polylogarithmic parameters. Second, we also run in to problems when we try to create a hierarchical partition. Unlike in the metric case we cannot simply choose leaders from each subcluster and connect directly, because connecting lower level partition arbitrarily may introduce huge blow-up costs. One approach following [12] is to replace $O(\log n)$ FRT trees by spanning trees drawn from the distribution of [5]. However, even with this, it is not clear how to combine paths drawn from these trees into a single spanning tree.

Dutta et al. propose the following UST construction [6].

1. Construct $(2^{\tilde{O}(\sqrt{\log n})}, 2^{\tilde{O}(\sqrt{\log n})}, R)$ partition for all R; for graphs with constant doubling dimension, we can achieve $(O(1), O(1), R)$-partitions.
2. Construct a hierarchical $(2^{\tilde{O}(\sqrt{\log n})}, 2^{\tilde{O}(\sqrt{\log n})}, 2^{\tilde{O}(\sqrt{\log n})})$-partition; for graphs with constant doubling dimension, a hierarchical $(O(1), O(1), O(\log^2 n))$-partition can be achieved.
3. Then build a UST from the hierarchical partition by connecting lower-level trees using shortest paths and invoke the properties of partitioning to bound the stretch, obtaining a $2^{\tilde{O}(\log^{3/4} n)}$ bound for general graphs and a $2^{\tilde{O}(\sqrt{\log n})}$ for doubling graphs.

4.3 Lower bound for UST

We find lower bounds for UST by looking at lower-bounds for on-line Steiner tree problems. Over n-nodes, the on-line Steiner Tree problem has a competitive ratio of $\Theta(\log n)$ for general metric spaces [14] and $\Omega(\log n / \log\log n)$ for Euclidean metric spaces [2]. Every UST for an n-node metric space with stretch $s(n)$ can be transformed into an on-line algorithm with a competitive ratio of $s(n)$. Hence, every UST has a stretch of $\Omega(\log n)$ for n-node metrics and $\Omega(\log n / \log\log n)$ for Euclidean metrics.

5 Conclusion and Open Problems

5.1 Complexity of Universal Problems

Consider again the stretch function:

$$max_{S \subseteq V} \frac{\text{Cost}(T_S}{\text{OPT}(S)}$$

We know that for a given terminal set S, finding OPT(S) is hard, although there are polynomial-time $O(1)$ approximations are known (MST, for example, in the UST case). For a candidate UST, finding the the worst-case set is also NP-hard. Finding whether there exists a UST a stretch with at most σ is coNP-hard. The problem of finding the minimum stretch UST is in Σ_2, and we conjecture that it is Σ_2-hard.

5.2 Open Problems

1. Close the gaps for UTSP and UST
   - Euclidean UTSP: $\Omega(\log^{1/6} n)$ vs $O(\log n)$
   - UTSP: $\Omega(\log n)$ vs $O(\log^2 n)$
   - Metric UTSP: $\Omega(\log n)$ vs $O(\log^2 n)$
2. Is there a polylog(n)-stretch graphical UST?
3. Strong diameter-partitions: Can we partition any graph into components of strong diameter polylog(n) such that each vertex has neighbors in polylog(n) components? (See [16] for similar related questions.)
4. Explore universal approximations for other optimization problems.

References

[1] Ittai Abraham, Yair Bartal, and Ofer Neiman. Nearly tight low stretch spanning trees. In *FOCS*, pages 781–790, 2008.

[2] 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.

[3] Deeparnab Chakrabarty Anand Bhalgat and Sanjeev Khanna. Optimal lower bounds for universal and differentially private steiner trees and TSPs. In *Proceedings of APPROX-RANDOM*, 2011.

[4] B. Awerbuch and D. Peleg. Sparse partitions. In *Proceedings of the 31st Annual IEEE Symposium on Foundations of Computer Science*, pages 503–513, 1990.

[5] Michael Elkin, Yuval Emek, Daniel A. Spielman, and Shang-Hua Teng. Lower-stretch spanning trees. In *STOC '05: Proceedings of the thirty-seventh annual ACM symposium on Theory of computing*, pages 494–503, New York, NY, USA, 2005. ACM Press.

[6] Dutta et al. Universal steiner trees for graphs. Unpublished manuscript, 2011.

[7] J. Fakcharoenphol, S. Rao, and K. Talwar. A tight bound on approximating arbitrary metrics by tree metrics. In *Proceedings of the 35th Annual ACM Symposium on Theory of Computing*, pages 448–455, June 2003.

[8] A. Goel and D. Estrin. Simultaneous optimization for concave costs: Single sink aggregation or single source buy-at-bulk. In *Proceedings of the 14th Annual ACM-SIAM Symposium on Discrete Algorithms*, January 2003.

[9] Ashish Goel and Ian Post. One tree suffices: A simultaneous O(1)-approximation for single-sink buy-at-bulk. In *FOCS*, pages 593–600, 2010.

[10] Igor Gorodezky, Robert D. Kleinberg, David B. Shmoys, and Gwen Spencer. Improved lower bounds for the universal and *priori* tsp. In *Proceedings of APPROX-RANDOM*, pages 178–191, 2010.

[11] Fabrizio Grandoni, Anupam Gupta, Stefano Leonardi, Pauli Miettinen, Piotr Sankowski, and Mohit Singh. Set covering with our eyes closed. In *FOCS*, pages 347–356, 2008.

[12] A. Gupta, M. Hajiaghayi, and H. Räcke. Oblivious network design. In *Proceedings of the 17th Annual ACM-SIAM Symposium on Discrete Algorithms*, January 2006.

[13] Mohammad Taghi Hajiaghayi, Robert D. Kleinberg, and Frank Thomson Leighton. Improved lower and upper bounds for universal TSP in planar metrics. In *SODA*, pages 649–658, 2006.

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

[15] L. Jia, G. Lin, G. Noubir, R. Rajaraman, and R. Sundaram. Universal approximations for TSP, steiner tree, and set cover. In *Proceedings of the 37th Annual ACM Symposium on Theory of Computing (STOC)*, 2005.

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

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

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

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

[20] Frans Schalekamp and David B. Shmoys. Algorithms for the universal and a priori TSP. *Oper. Res. Lett.*, 36(1):1–3, 2008.

[21] David B. Shmoys and Kunal Talwar. A constant approximation algorithm for the a priori Traveling Salesman Problem. In *IPCO*, pages 331–343, 2008.