Ramsey Spanning Trees and their Applications

Ittai Abraham (iabraham@vmware.com), Shiri Chechik (schechik@post.tau.ac.il), Michael Elkin (elkinm@cs.bgu.ac.il), Arnold Filtser (arnoldf@cs.bgu.ac.il), Ofer Neiman (neimano@cs.bgu.ac.il)

The *Metric Ramsey Problem*

Given a metric (X, d), what is the largest subset $M \subseteq X$ of a metric space that can be embedded into a Hilbert space?
Mendel and Naor [MN07], via a randomized algorithm, showed that every n-point metric (X, d) has a subset $M \subseteq X$ of size at least $n^{1-1/k}$ that embeds into an ultrametric (and thus also into Hilbert space) with distortion at most $128k$, for a parameter $k \geq 1$.

Theorem 1 *For every n-point metric space and $k \geq 1$, there exists a subset M of size $n^{1-1/k}$ that can be embedded into an ultrametric with distortion $8k-2$. Moreover, there is a deterministic polynomial algorithm that finds M and its embedding.*

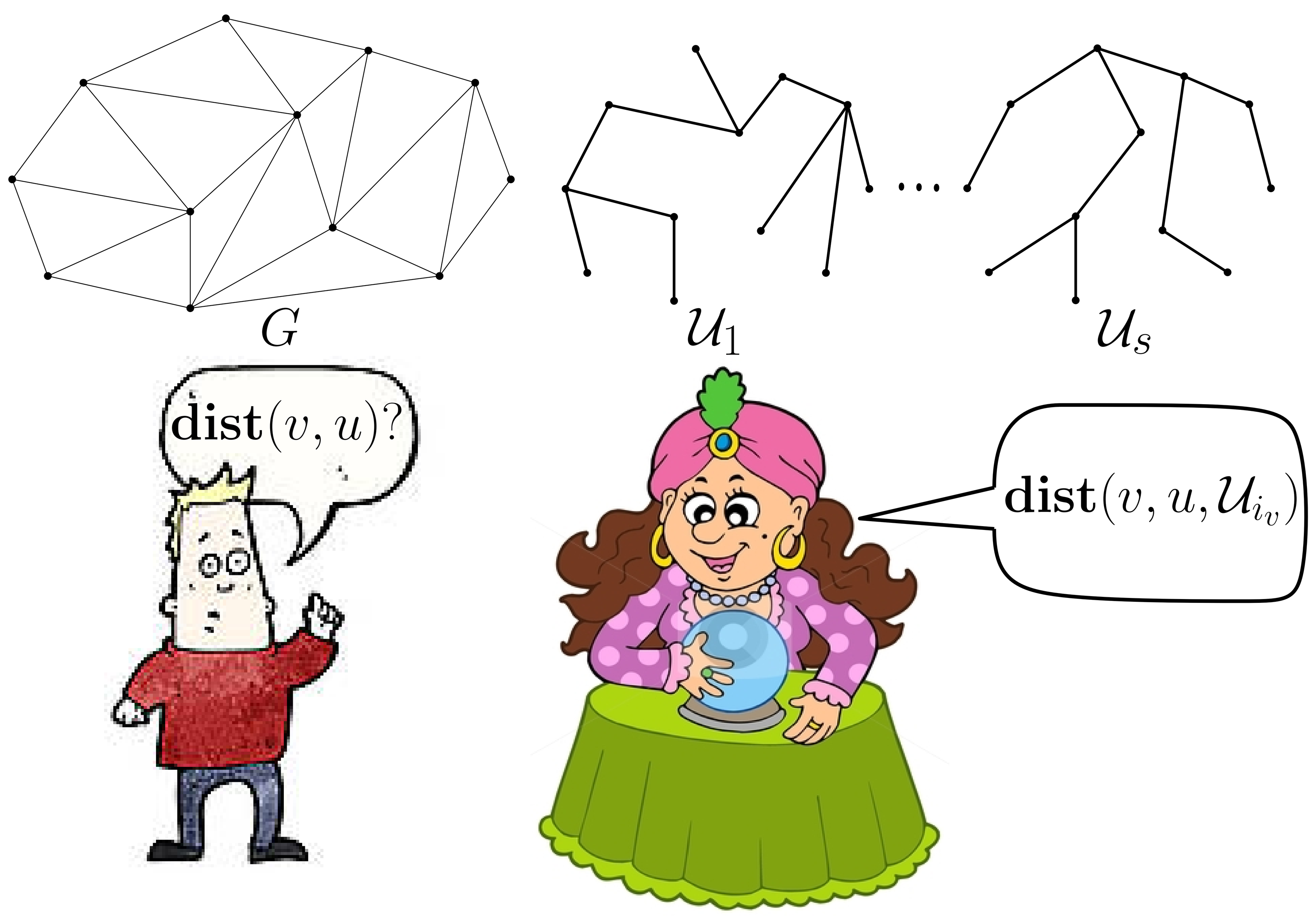
Application to Distance Oracles

A distance oracle is a succinct data structure that (approximately) answers distance queries. The properties of interest are size, stretch and query time.
Using Theorem 1 we can construct $O(kn^{1/k})$ ultrametrics $\{U_i\}_i$, such that for every $v \in X$, there is i_v, such that for every $u \in X$

$$\mathbf{dist}(v,u) \leq \mathbf{dist}(v,u,U_{i_v}) \leq 16k \cdot \mathbf{dist}(v,u)$$

Given a query u, v, our distance oracle simply returns $\mathbf{dist}(v,u,U_{i_v})$.
The query time is constant. The required space is $O(kn^{1+\frac{1}{k}})$.

Distance Oracle	Stretch	Size	Query time	Is deterministic?
[TZ05]	$2k-1$	$O(k \cdot n^{1+1/k})$	$O(k)$	no
[MN07]	$128k$	$O(n^{1+1/k})$	$O(1)$	no
[WN13]	$(2+\epsilon)k$	$O(k \cdot n^{1+1/k})$	$O(1/\epsilon)$	no
[Che14]	$2k-1$	$O(k \cdot n^{1+1/k})$	$O(1)$	no
[Che15]	$2k-1$	$O(n^{1+1/k})$	$O(1)$	no
[RTZ05]	$2k-1$	$O(k \cdot n^{1+1/k})$	$O(k)$	yes
[WN13]	$2k-1$	$O(k \cdot n^{1+1/k})$	$O(\log k)$	yes
This paper	$8(1+\varepsilon)k$	$O(n^{1+1/k})$	$O(1/\epsilon)$	yes
This paper	$2k-1$	$O(k \cdot n^{1+1/k})$	$O(1)$	yes



Ramsey Spanning Trees (Main result)

Ramsey Spanning Trees is a natural extension of the metric Ramsey problem to graphs. Given a weighted graph $G = (V, E)$, what is the largest subset $M \subseteq V$ of vertices, such there is a spanning (sub-graph) tree T of G, with small stretch w.r.t all pairs in $M \times V$?

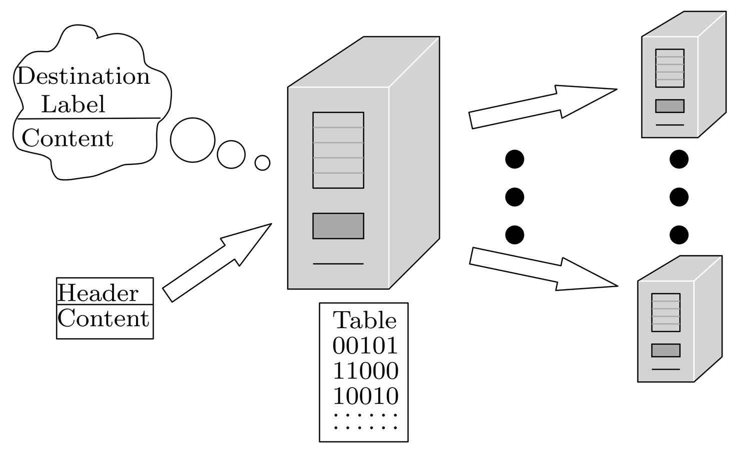
Theorem 2 *Let $G = (V, E)$ be a weighted graph, and a parameter $k \geq 1$. There exists a spanning tree T of G, and a set $M \subseteq V$ of size at least $n^{1-1/k}$, such that for every $v \in M$ and every $u \in V$ it holds that $\mathbf{dist}(u,v,T) \leq O(k \log\log m) \cdot \mathbf{dist}(v,u,G)$.*

Corollary 1 *Let $G = (V, E)$ be a weighted graph on n vertices, and fix a parameter $k \geq 1$. There is a polynomial time deterministic algorithm that finds a collection $\mathcal{T}$ of $k \cdot n^{1/k}$ spanning trees of G, and a mapping $\mathbf{home} : V \to \mathcal{T}$, such that for every $u, v \in V$ it holds that*

$$\mathbf{dist}(v,u,\mathbf{home}(v)) \leq O(k \log\log n) \cdot \mathbf{dist}(v,u,\mathbf{home}(v))$$

Compact Routing

A routing scheme in a network is a mechanism that allows packets to be delivered from any node to any other node. Each node can forward incoming data by using local information stored at its *routing table*, and the (short) packet's *header*. During preprocessing phase, each node is assigned a routing table and a short label. In the routing phase, each node receiving a packet should make a local decision, based on its own routing table and the packet's header (which usually contains the label of the destination) where to send the packet. The *routing decision time* is the time required for a node to make this local decision. The *stretch* of a routing scheme is the worst ratio between the length of a path on which a packet is routed, to the shortest possible path.



Application to Compact Routing

For any tree $T = (V, E)$ (where $|V| = n$), there is a routing scheme with stretch 1 that has routing tables of size $O(b)$ and labels of size $(1 + o(1)) \log_b n$. The decision time in each vertex is $O(1)$.
Thus, given the collection $\mathcal{T}$ of trees from Corollary 1, the label of each vertex v consist of $\mathbf{home}(v)$ and the label of v in $\mathbf{home}(v)$. The table is the union of the tables in all the $k \cdot n^{1/k}$ trees. We conclude,

Theorem 3 *Given a weighted graph $G = (V, E)$ on n vertices and integer parameters $k, b > 1$, there is a routing scheme with stretch $O(k \log\log n)$ that has routing tables of size $O(k \cdot b \cdot n^{1/k})$ and labels of size $(1 + o(1)) \log_b n$. The decision time in each vertex is $O(1)$.*

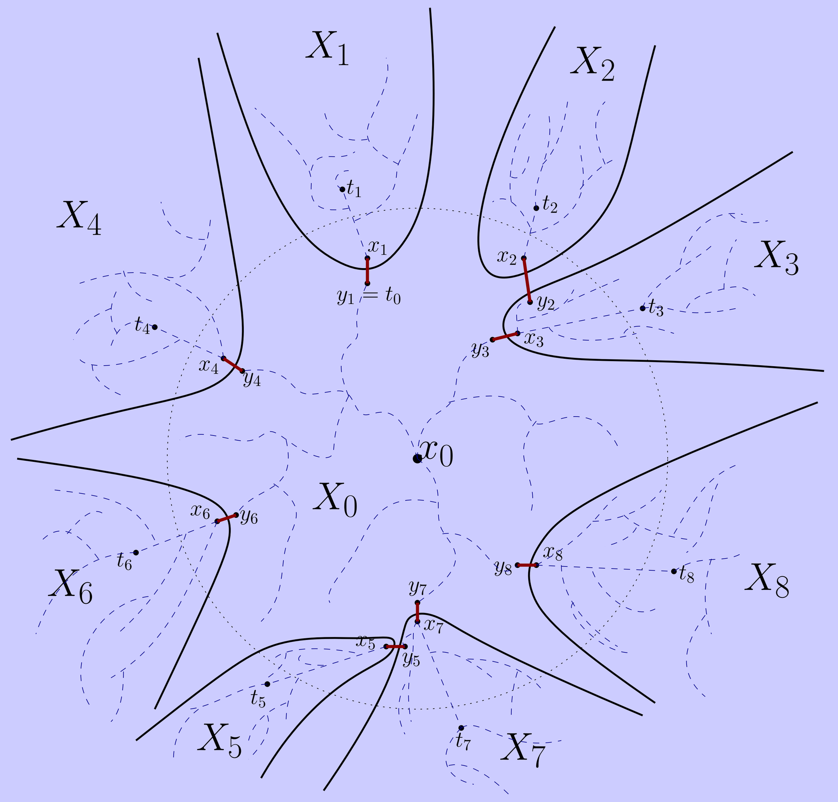
	Parameter range	Label	Table	Stretch
[TZ01]	$k \geq 1$	$O(k \log n)$	$O(k \cdot n^{\frac{1}{k}})$	$4k-3$
Theorem 3	$k \geq 1$	$(1+o(1)) \log n$	$O(k \cdot n^{\frac{1}{k}})$	$O(k \log\log n)$
[TZ01]	$k = \log n$	$O(\log^2 n)$	$O(\log n)$	$O(\log n)$
Theorem 3	$k = \frac{\log n}{\log\log n}$	$(1+o(1)) \log n$	$O(\frac{\log^2 n}{\log\log n})$	$O(\log n)$
Theorem 3	$k = \log n$	$(1+o(1)) \log n$	$O(\log n)$	$O(\log n \log\log n)$

Our scheme is arguably simpler then [TZ01], and has extremely small label size.

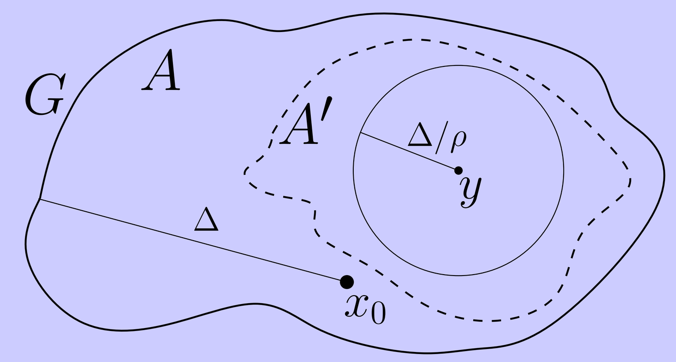
Technical ideas in Theorem 2

0.1 Petal Decomposition

Petal decomposition is an iterative method to build a spanning tree. In each level, the current graph is partitioned into smaller diameter pieces, called petals, and a single central piece, which are then connected by edges in a tree structure. Each of the petals is a ball in a certain metric. The produced spanning tree has diameter ($O(\Delta)$) which is proportional to the diameter (Δ) of the graph, while allowing large freedom for the choice of radii of the petals ($\Omega(\Delta)$).



0.2 Padding



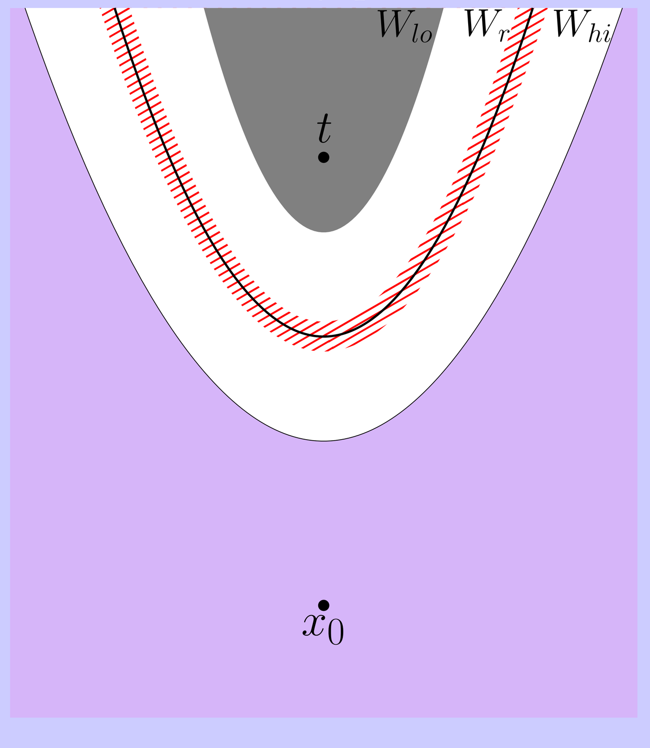
Consider a vertex $y \in A' \subseteq A$ A is a subset of vertices in the graph G with center x_0. The radius of A (w.r.t x_0) is Δ. A' is a subset of A (denoted by the dashed line). If $B(y, \Delta/\rho, G) \subseteq A'$, we say that the vertex y is padded by A' w.r.t A. y is fully padded, if it is padded in all the levels.

0.3 Choosing a Radius

We are given range $[lo, mid]$ of length $\Delta/8$. For each $r \in [lo, mid]$, W_r denotes the petal with radius r. w_r denotes the number of vertices in W_r which are fully padded so far.
For radius r, all vertices in $W_{r+\frac{\Delta}{c \cdot k \cdot \log\log n}} \setminus W_{r-\frac{\Delta}{c \cdot k \cdot \log\log n}}$ ceased being padded. We will ensure that large fraction of the vertices will be "saved".
First choose $[a, b] \subseteq [lo, mid]$ such that $b - a = \frac{R}{2L}$ and $w_a \geq w_b^2/|S|$ ($|S|$ is the number of active vertices). Then we pick $r \in [a, b]$ such that

$$w_{r+\frac{b-a}{2k}} \leq w_{r-\frac{b-a}{2k}} \cdot \left(\frac{w_b}{w_a}\right)^{\frac{1}{k}}.$$

In our setting we are very sensitive to constant factors in this charging scheme, because these constant are multiplied throughout the recursion. In particular, we must avoid a range in $[lo, hi]$ that contains more than half of the marked vertices. To this end, we sometimes "cut backwards", that is, the "saved" vertices are those out of $W_{r+\frac{\Delta}{c \cdot k \cdot \log\log n}}$.



References

[Che14] Shiri Chechik. Approximate distance oracles with constant query time. In *STOC*, 2014.
[Che15] Shiri Chechik. Approximate distance oracles with improved bounds. In *STOC*, 2015.
[MN07] Manor Mendel and Assaf Naor. Ramsey partitions and proximity data structures. *JEMS*, 2007.
[RTZ05] Liam Roditty, Mikkel Thorup, and Uri Zwick. Deterministic constructions of approximate distance oracles and spanners. In *ICALP*, 2005.
[TZ01] M. Thorup and U. Zwick. Compact routing schemes. In *SPAA*, 2001.
[TZ05] Mikkel Thorup and Uri Zwick. Approximate distance oracles. *J. ACM*, 2005.
[WN13] Christian Wulff-Nilsen. Approximate distance oracles with improved query time. SODA, 2013.