Computing distances in a graph in constant time and with 0.793*n*-bit labels

Cyril Gavoille

(LaBRI, U. Bordeaux)

Stephen Alstrup (U. Copenhagen)

Esben Bistrup Halvorsen (U. Copenhagen)

Holger Petersen (-)

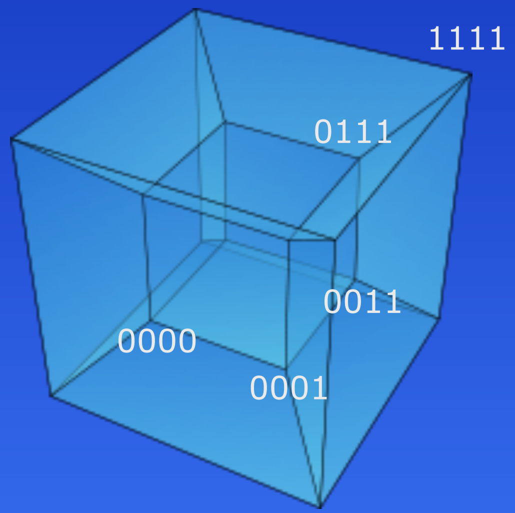
Aussois – ANR DISPLEXITY - March 2015

Agenda

1. The Problem
2. Squashed Cube Dimension
3. Dominating Set Technique
4. Another Solution
5. Conclusion

The Distance Labeling Problem

Given a graph, find a labeling of its nodes such that the distance between any two nodes can be computed by inspecting only their labels.

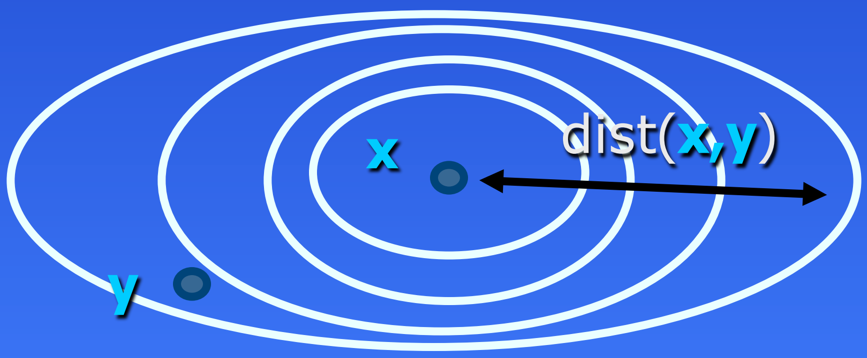


Subject to:

- label the nodes of every graph of a family (scheme),
- using short labels (size measured in bits), and
- with a fast distance decoder (algorithm)

Motivation
[Peleg '99]

If a short label (say of poly-logarithmic size) can be added to the address of the destination, then routing to any destination can be done without routing tables and with a "limited" number of messages.



message header=hop-count

A Distributed Data Structure

- Get the labels of nodes involved in the query
- Compute/decode the answer from the labels
- No other source of information is required

Label Size:
a trivial upper bound

There is a labeling scheme using labels of $O(n \log n)$ bits for every (unweighted) graph G with n nodes, and constant time decoding.

$$L_G(i)=(i, [\mathrm{dist}(i,1),\ldots,\mathrm{dist}(i,j),\ldots,\mathrm{dist}(i,n)])$$

➟ distance vector

Label Size:
a trivial lower bound

No labeling scheme can guarantee labels of less than $0.5n$ bits for all n-node graphs
(whatever the distance decoder complexity is)

Proof. The sequence $\langle L_G(1),\ldots,L_G(n)\rangle$ and the decoder $\delta(.,.)$ is a representation of G on $n \cdot k + O(1)$ bits if each label has size k: i adjacent to j iff $\delta(L_G(i),L_G(j))=1$.

$$n \cdot k + O(1) \geq \log_2(\#\text{graphs}(n)) = n \cdot (n-1)/2.$$

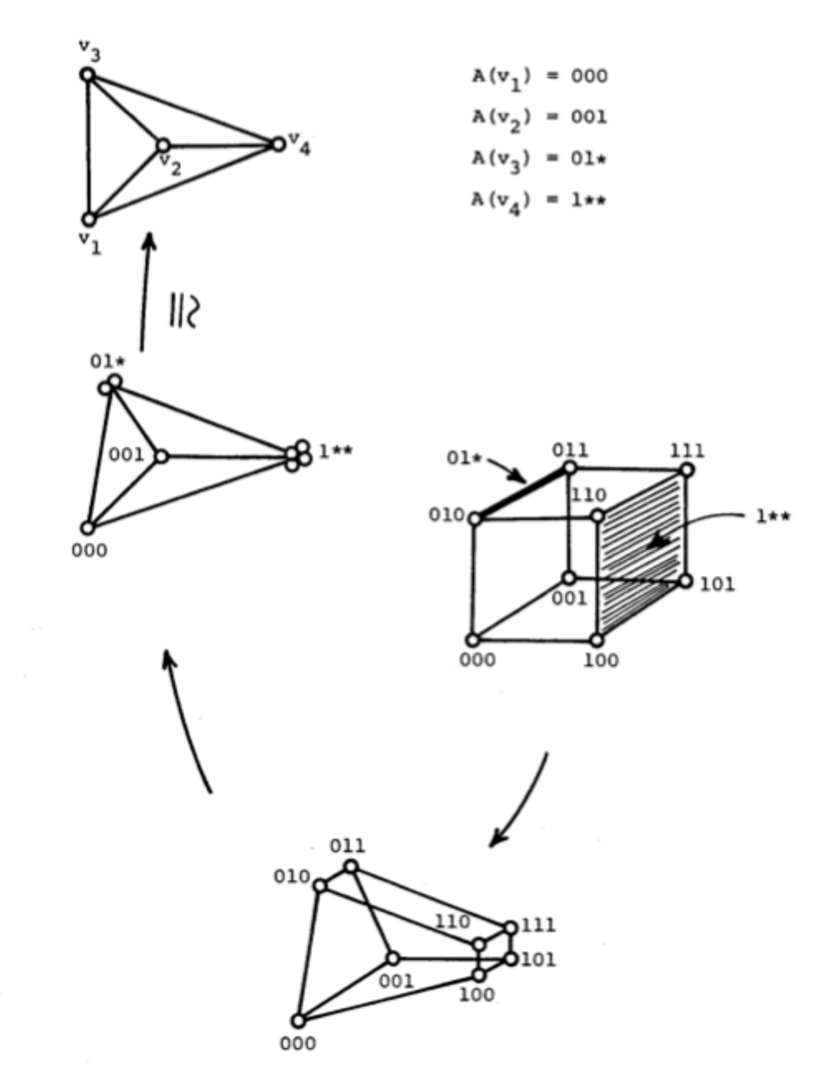
→ Distance Vector

Squashed Cube Dimension
[Graham,Pollack '71]

Labeling: word over {0,1,*}
Decoder: Hamming distance
(where *=don't care)
(graphs must be connected)

sc-dim(G)≥max{n^+,n^-}
$n^{+/-}$=#positive (negative) eigenvalues of the distance matrix of G

sc-dim(K_n)=n-1



Squashed Cube Dimension
[Winkler ’83]

Theorem. Every connected n-node graph has squashed cube dimension is at most n-1.

Therefore, for the family of all connected n-node graphs:

Label size: O(n) bits, in fact $n\log_2 3 \sim 1.58n$ bits
Decoding time: O($n/\log n$) in the RAM model

Rem: all graphs = connected graphs + O($\log n$) bits

Dominating Set Technique
[G.,Peleg,Pérennès,Raz ’01]

Definition. A k-dominating set for G is a set S such that, for every node $x \in G$, dist(x,S) $\leq k$.

Fact: Every n-node connected graph has a k-dominating set of size at most $n/(k+1)$.

Target:

Label size: O(n) bits

Decoding time: O(loglog$n$)

Part 1

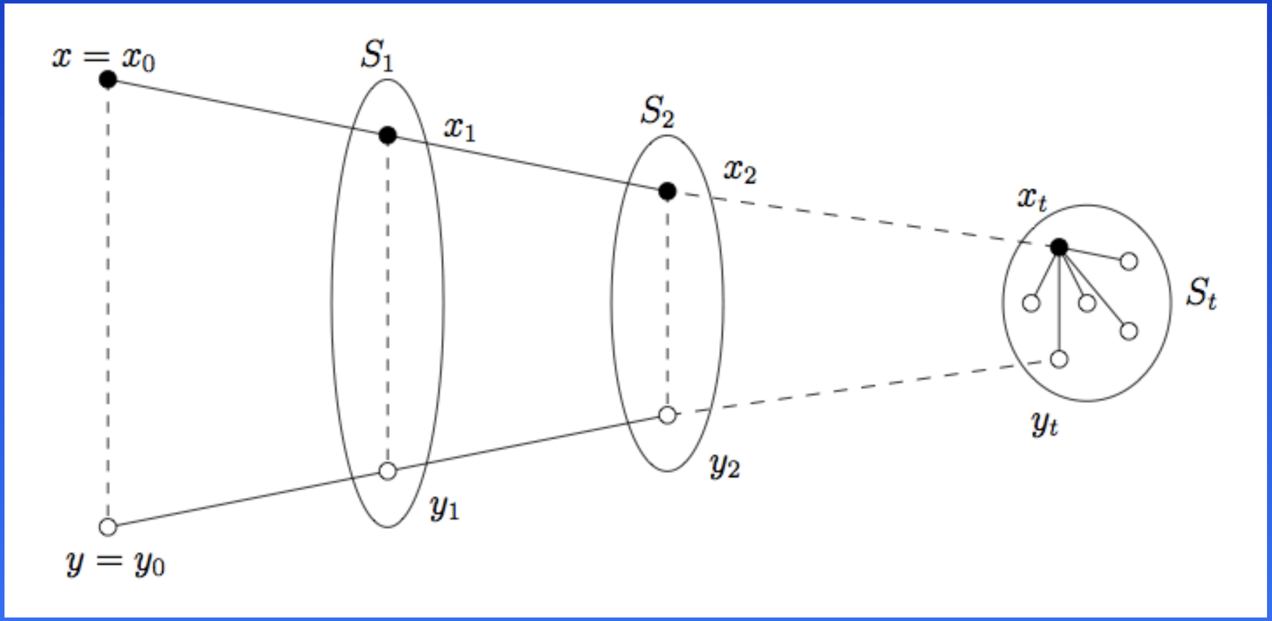
$\mathrm{dom}_S(x)$ = *dominator* of x, i.e., its closest node in S

Observation 1:
$\mathrm{dist}(x,y)=\mathrm{dist}(\mathrm{dom}_S(x),\mathrm{dom}_S(y))+e(x,y)$
where $e(x,y)\in[-2k,+2k]$

Observation 2:
One can be encoded $e(x,y)$ with $\log_2(4k+1)$ bits

Part 2

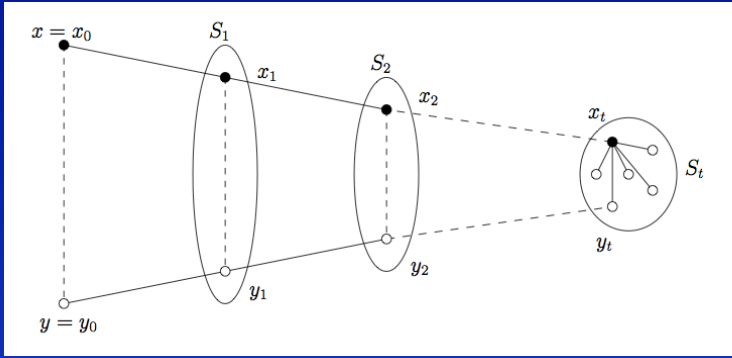
Idea: take a collection of k_i-dominating set S_i, i=0…t (may intersect) where $k_0=0$, $|S_i|=n/(k_i+1)=o(n/\log n)$. Denote $x_0=x$ and $x_i=\mathrm{dom}_{S_i}(x_{i-1})$.



Decoder: $\mathrm{dist}(x,y)=\sum_{1\le i\le t} e_i(x_{i-1},y_{i-1}) + \mathrm{dist}(x_t,y_t)$

Time: $O(t)$

Part 3



Decoder: $\text{dist}(x,y)=\sum_{1\le i\le t} e_i(x_{i-1},y_{i-1}) + \text{dist}(x_t,y_t)$

Label size: $\sum_{1\le i\le t} |S_{i-1}|\cdot\log_2(4k_i+1) + |S_t|\cdot O(\log n)$ bits

$$\text{size} \le n \cdot \sum_{1\le i\le t} \log_2(4k_i+1)/(k_{i-1}+1) + o(n)$$

Choose: $k_i=4^i-1$ and $t=O(\log\log n)$

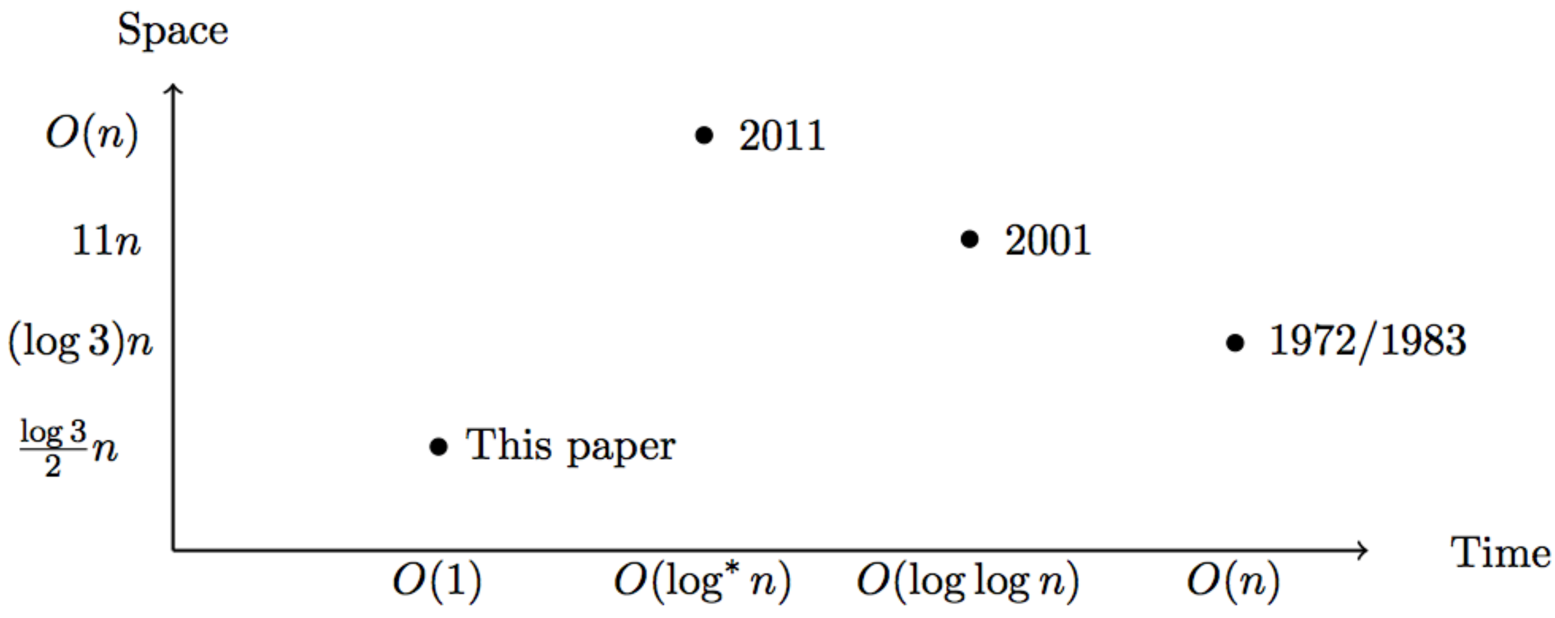
➡ $k_0=0$, $\log_2(4k_i+1)=2i+2$, and $|S_t|=o(n/\log n)$

➡ size $\le n \cdot \sum_{i\ge 1} (2i+2)/4^{i-1} = 56/9 \sim 6.22n$ bits

➡ time $O(t) = O(\log\log n)$

Another Solution

Label size: $n(\log_2 3)/2 \sim 0.793n$ bits
Decoding time: O(1) in the RAM model



Warm-up
(ignoring decoding time)

Assume G has a Hamiltonian cycle $x_0, x_1, \ldots, x_{n-1}$.

➟ $\text{dist}(x_0,x_i) = \text{dist}(x_0,x_{i-1}) + e_0(x_i,x_{i-1})$ where $e_0(x_i,x_{i-1}) \in \{-1,0,+1\}$

➟ $\text{dist}(x_0,x_i) = \sum_{1 \le j \le i} e_0(x_j,x_{j-1})$

Label: $L_G(x_i)=(i, [e_i(x_{i+1},x_i), e_i(x_{i+2},x_{i+1}), \ldots])$
Size: $n(\log_2 3)/2$ bits by storing only half $e_i()$'s

Applications

Assume G is 2-connected. By Fleischner's Theorem [1974], G^2 has a Hamiltonian cycle $x_0,x_1,\ldots,x_{n-1}$. So, $e_0(x_i,x_{i-1})=\text{dist}(x_0,x_i)-\text{dist}(x_0,x_{i-1})\in\{-2,-1,0,1,2\}$.

Label size: $n(\log_2 5)/2 \sim 1.16n$ bits

For every tree T, T^3 is Hamiltonian. So, if G is connected, then G^3 is Hamiltonian and has a labeling scheme with $n(\log_2 7)/2 \sim 1.41n$ bits.

Constant Time Solution

T = any rooted spanning tree of G

$e_x(u)$ = dist(x,u)-dist(x,parent(u))$\in\{-1,0,+1\}$

Telescopic sum: if z ancestor of y, then

dist(x,y)=dist(x,z)+$\sum_{u\in T[y,z)} e_x(u)$

Dominating Set & Edge-Partition

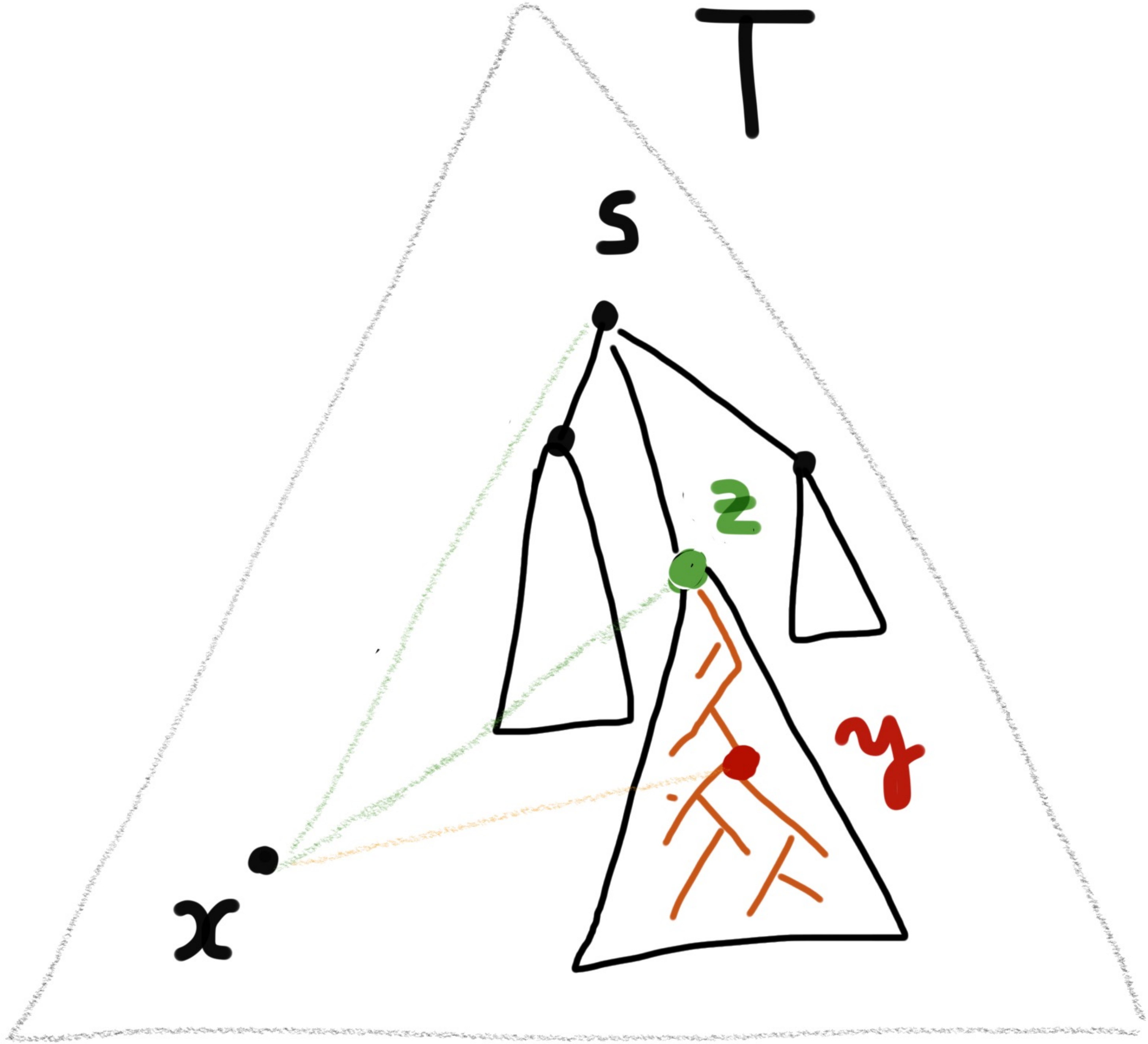
Select an α-dominating $S=\{s_i\}$ with $root(T) \in S$ and $|S| \leq n/\alpha+1$

Edge-partition: T has an edge-partition into at most $\lceil n/\beta \rceil$ sub-trees $\{T_i\}$ of $\leq 2\beta$ edges.

Choose:

$\alpha = \lceil \log n \cdot \log\log n \rceil$

$\beta = \lceil \log \alpha \cdot \log\log n \rceil \sim (\log\log n)^2$



T
s
z
y
x

Distance decoder:

dist(x,y)=dist(x,z)+$\sum_{u\in T[y,z)} e_x(u)$ = A + B

Storage for computing A:

#z x log$n$=(n/β)·log$n$=ω(n) ✖

But, A = dist(x,s) + A′, where

A′=(dist(x,z)-dist(x,s))∈[-2α,+2α]

➟ only $\lceil \log_2(2\alpha+1)\rceil$ bits to store A′.

#s x log$n$ =(n/α)·log$n$ = o(n) ✔

#z x $\lceil \log(2\alpha+1)\rceil$=($n$/$\beta$)·log$\alpha$ = o(n) ✔

Distance decoder:

$\mathrm{dist}(x,y)=\mathrm{dist}(x,z)+\sum_{u\in T[y,z)} e_x(u) = A + B$

Storage for computing B: let $y \in T_i$

$B=h(T_i,e_x(T_i),rk(y))$ for some $h(\ldots)$

But: the input $\langle T_i,e_x(T_i),rk(y)\rangle$ writes on $O(\beta)=O(\log^2\log n)$ bits!

➟ one can tabulate $h(\ldots)$ once for all its possible inputs with $2^{O(\beta)} \times \log_2 B = o(n)$ bits to have constant time.

Storage for *u* (label):

storage for all A,A′,B ➟ $o(n)$ bits

storage also for:

1. i st. $u \in T_i$, $rk_i(u)$ in T_i, and $z = root(T_i)$
2. closest $s \in S$ ancestor of z
3. coding of T_i with $O(\beta)$ bits
4. $e_u(T_i),\ldots,e_u(T_{i+k})$ for half the total information (this latter costs $n(\log_2 3)/2 + o(n)$ bits)

Decoder: [x "knows" y, otherwise swap x,y]

1. y sends to x: i, T_i, $rk_i(y)$, z, s ($O(\log n)$ bits)
2. x computes and returns A+A′+B

Conclusion

Main question: Design a labeling scheme with $0.5n+o(n)$-bit labels and constant time decoder?

Bonus:

- The technique extends to weighted graphs. We show $n \cdot \log_2(2w+1)/2$ bits for edge-weight in [1,w]. We show a lower bound of $n \cdot \log_2(w/2+1)/2$.
- We also show a 1-additive (one sided error) labeling scheme of $n/2$ bits, and a lower bound of $n/4$.