

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

Cyril Gavoille
(LaBRI, U. Bordeaux)

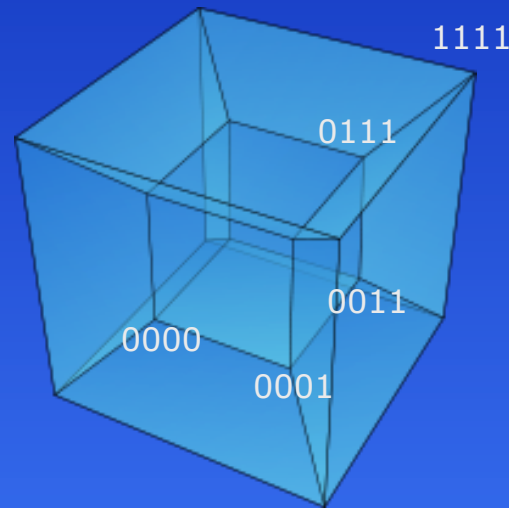
Stephen Alstrup (U. Copenhagen)
Esben Bistrup Halvorsen (U. Copenhagen)
Holger Petersen (-)

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.



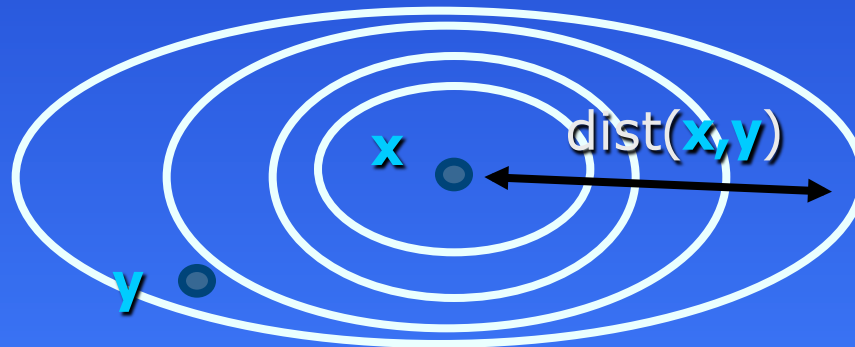
Subject to:

- 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.

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

⇒ 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 $\langle L_G(1), \dots, L_G(n) \rangle$ 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 $\delta(L_G(i), L_G(j)) = 1$.

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



Squashed Cube Dimension

[Graham, Pollack '71]

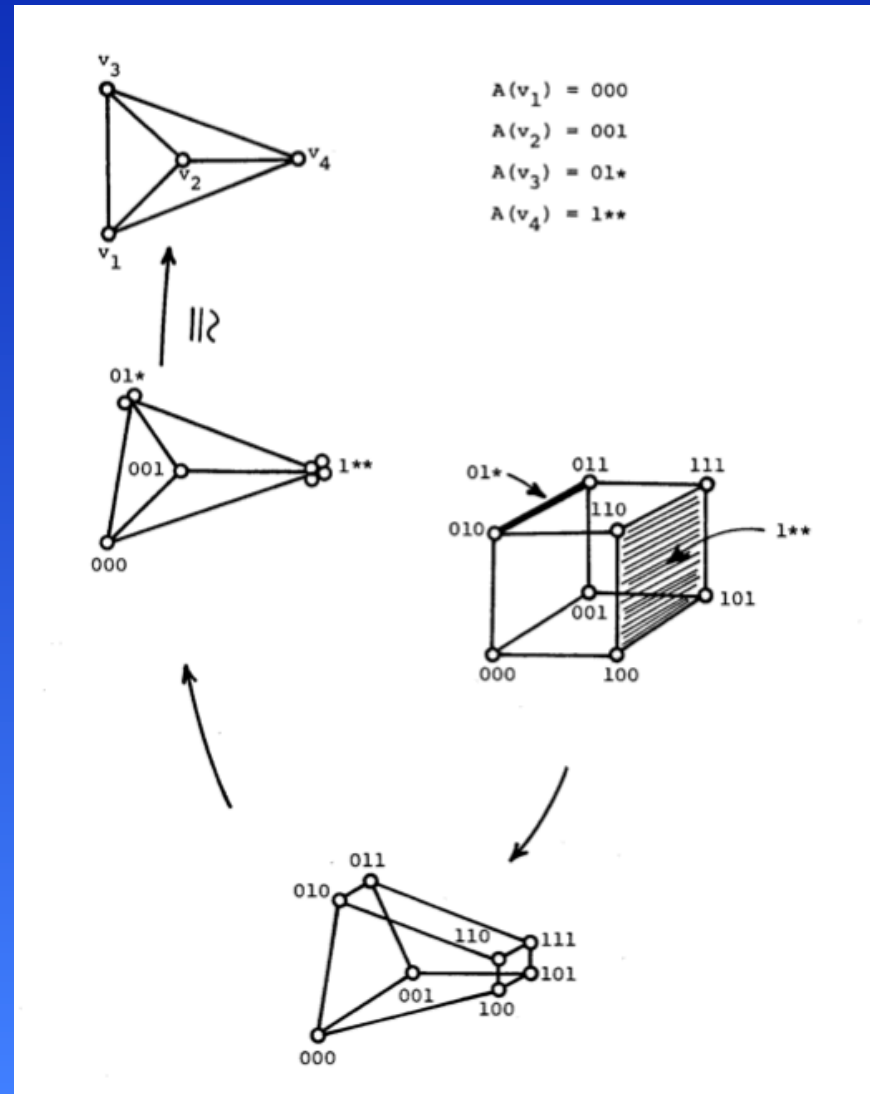
Labeling: word over $\{0,1,*\}$

Decoder: Hamming distance
(where $*$ = don't care)
(graphs must be connected)

$$SC_{\dim}(G) \geq \max\{n^+, n^-\}$$

$n^{\pm} = \#$ 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 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 $x \in G$, $\text{dist}(x, S) \leq k$.

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: $O(\log \log n)$

Part 1

$\text{dom}_S(x)$ = *dominator* of x , its closest node in S

Observation 1:

$\text{dist}(x,y) = \text{dist}(\text{dom}_S(x), \text{dom}_S(y)) + e(x,y)$

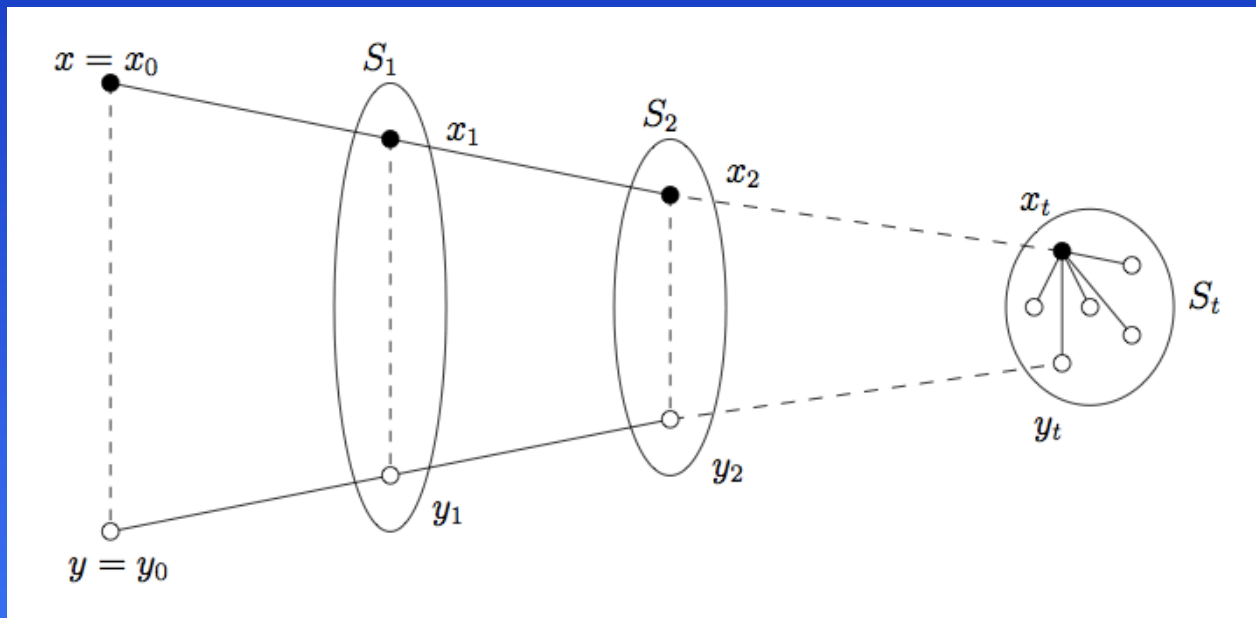
where $e(x,y) \in [-2k, +2k]$

Observation 2:

One can encode $e(x,y)$ with $\lceil \log_2(4k+1) \rceil$ bits

Part 2

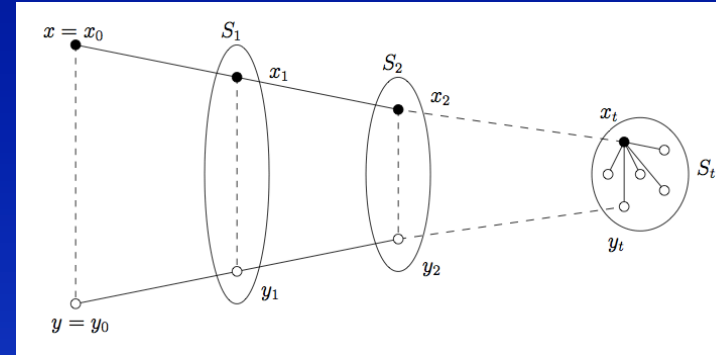
Idea: take a collection of k_i -dominating set S_i , $i=0\dots t$ (may intersect) where $k_0=0$, $|S_i|=n/(k_i+1)$, and $|S_t|=o(n/\log n)$. Denote $x_0=x$ and $x_i=\text{dom}_{S_i}(x_{i-1})$.



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

Time: $O(t)$

Part 3



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

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

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

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

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

$\Rightarrow \text{size} \leq n \cdot \sum_{i \geq 1} (2i + 2) / 4^{i-1} = n \cdot (56/9) \sim 6.22n$ bits

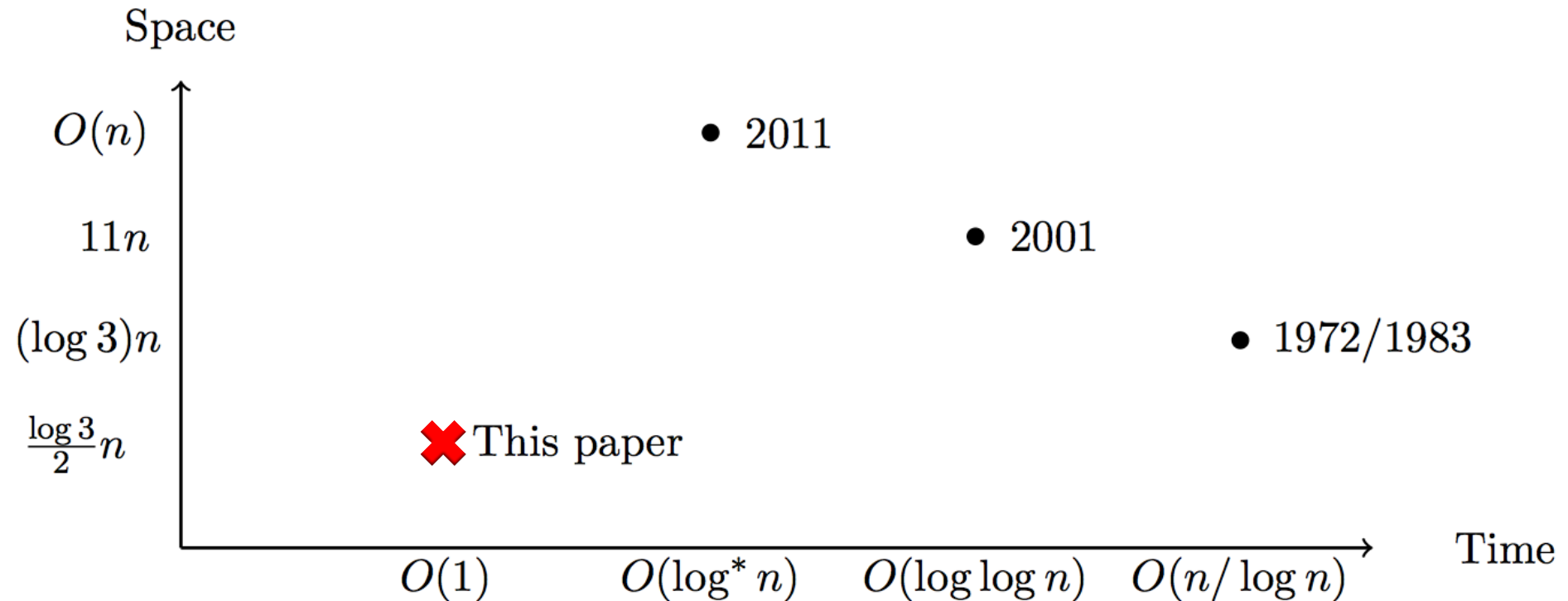
$\Rightarrow \text{time } O(t) = O(\log \log n)$

[$k_{0..4} = (0, 2, 15, 145, 940)$, $k_i = \exp(k_{i-1})$, $t = \log^* n$, $\text{size} \sim 5.90n$]

Another Solution

Label size: $n(\log_2 3)/2 \sim 0.793n$ bits

Decoding time: $O(1)$



Warm-up

(ignoring decoding time)

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

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

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

Label: $L_G(x_i) = (i, [e_i(x_{i+1}, x_i), e_i(x_{i+2}, x_{i+1}), \dots])$

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, \dots, 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

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

Telescopic sum: if z ancestor of y , then
 $\text{dist}(x,y) = \text{dist}(x,z) + \sum_{u \in T[y,z)} e_x(u)$

Dominating Set & Edge-Partition

Select an α -dominating $S = \{s_i\}$ with $\text{root}(T) \in S$
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:

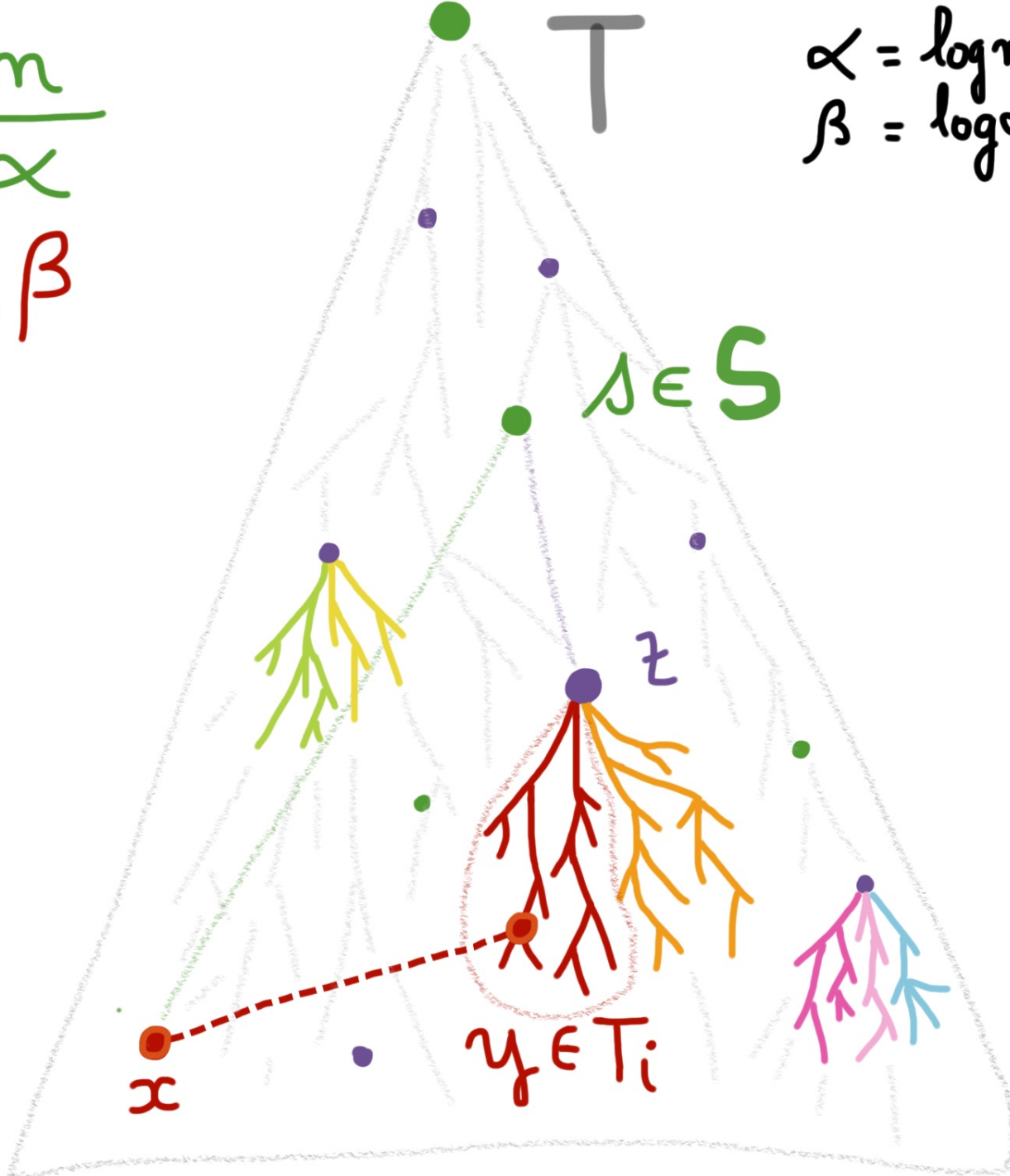
$$\alpha = \lceil \log n \cdot \log \log n \rceil$$

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

$$|S| \leq \frac{n}{\alpha}$$

$$|T_i| \leq 2\beta$$

$$\alpha = \log n \cdot \log \log n$$
$$\beta = \log \alpha \cdot \log \log n$$



Distance decoder:

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

Storage for computing A:

$$\#z \times \log n = (n/\beta) \cdot \log n = \omega(n) \quad \times$$

But, $A = \text{dist}(x,s) + A'$, where

$$A' = (\text{dist}(x,z) - \text{dist}(x,s)) \in [-2\alpha, +2\alpha]$$

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

$$\#s \times \log n = (n/\alpha) \cdot \log n = o(n) \quad \checkmark$$

$$\#z \times \lceil \log(2\alpha+1) \rceil = (n/\beta) \cdot \log \alpha = o(n) \quad \checkmark$$

Distance decoder:

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

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

$$B = f(T_i, \vec{e}_x(T_i), \text{rk}(y)) \text{ for some } f(\dots)$$

$$\text{where } \vec{e}_x(T_i) = [-1, 0, +1, +1, 0, -1, \dots]$$

But, the input $\langle T_i, \vec{e}_x(T_i), \text{rk}(y) \rangle$ writes on $O(\beta) = O(\log^2 \log n)$ bits!

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

Storage for u (in its label):

storage for all $A, A', B \rightsquigarrow o(n)$ bits

storage also for:

1. i st. $u \in T_i$, $\text{rk}(u)$ in T_i , and $z = \text{root}(T_i)$
2. closest $s \in S$ ancestor of z
3. coding of T_i with $O(\beta)$ bits
4. $\vec{e}_u(T_i), \vec{e}_u(T_{i+1}), \dots$ 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 tells to x : $\langle i, T_i, \text{rk}(y), z, s \rangle$
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 $n/4$.