

# On Approximate Distance Labels and Routing Schemes with Affine Stretch

Ittai Abraham – **Cyril Gavoille**

Microsoft, Mountview – University of Bordeaux

DISC '11 – Rome

September 2011

# Topic

Given a graph  $G$  with  $n$  nodes compute an **efficient** data structure supporting **approximate** distance and/or routing queries in  $G$ .

- **Efficient**: fast pre-processing and low query time
- **Approximate**: guarantees on the distance  $\hat{d}$  or route length returned w.r.t. the shortest path in  $G$ .

$$\text{Ex: } d_G(u, v) \leq \hat{d}(u, v) \leq f(d_G(u, v))$$

Affine stretch if  $f(d) = \alpha \cdot d + \beta$

# Goals

- data structures of  $o(n^2)$  space
- constant time query
- polynomial time pre-processing
- distributed data structure: split into  $n$  balanced labels



# State-of-the-art

**Thorup-Zwick distance oracle (J.ACM '05)** Every weighted graph has a stretch  $(2k - 1)d$  distance oracle of size  $\tilde{O}(n^{1+1/k})$  with query time  $O(k)$ , and polynomial pre-processing. Moreover, the oracle can be represented as a distance labeling.

**For  $k=2$ :**

$\Rightarrow$  stretch  $3d$ , space  $n^{3/2}$ , constant query time

## Can we do better for unweighted graphs?

**Pătraşcu-Roditty (FOCS '10)** Every unweighted graph has stretch  $2d + 1$  distance oracle of size  $\tilde{O}(n^{5/3})$  with constant query time. (Label approach fails because of use a global hash table of size  $n^{5/3}$ )

# Can we do better for unweighted graphs?

**Pătraşcu-Roditty (FOCS '10)** Every unweighted graph has stretch  $2d + 1$  distance oracle of size  $\tilde{O}(n^{5/3})$  with constant query time. (Label approach fails because of use a global hash table of size  $n^{5/3}$ )

## Theorem (1st contribution)

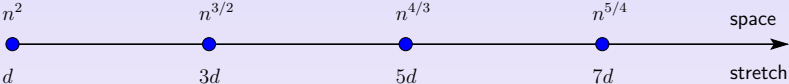
*Let  $k \geq 2$ . Every unweighted graph has stretch  $(2k - 2)d + 1$  distance oracle of size  $\tilde{O}(n^{1+2/(2k-1)})$  with query time  $O(k)$ . Moreover it can be represented as a distance labeling.*

**For  $k=2$ :**

$\Rightarrow$  stretch  $2d + 1$ , space  $n^{1+2/3} = n^{5/3}$ , constant query time

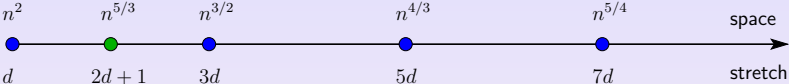
# Different trade-offs

[TZ05]



# Different trade-offs

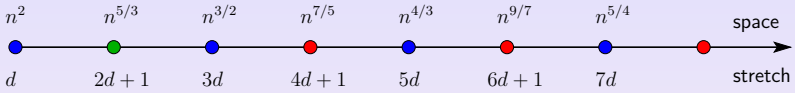
[TZ05][PR10]





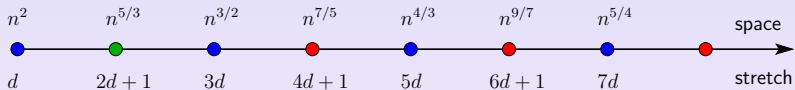
# Different trade-offs

[TZ05][PR10][us]



# Different trade-offs

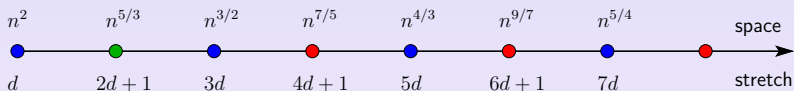
[TZ05][PR10][us]



The **Best Solution** depends on the question:

# Different trade-offs

[TZ05][PR10][us]

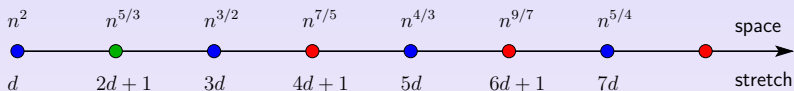


The **Best Solution** depends on the question:

- 1 What's the lowest space complexity with stretch  $\leq 3d$ ?  
 $\Rightarrow n^{3/2}$  [TZ05]

# Different trade-offs

[TZ05][PR10][us]



The **Best Solution** depends on the question:

- 1 What's the lowest space complexity with stretch  $\leq 3d$ ?  
 $\Rightarrow n^{3/2}$  [TZ05]
- 2 What is the smallest stretch with space complexity  $o(n^2)$ ?  
 $\Rightarrow$  at most  $2d+1$  [PR10][us]

## Proof for $k = 2$

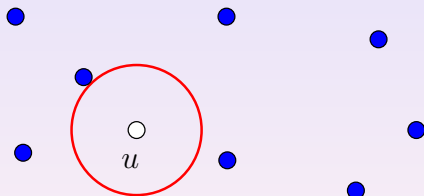
(we want  $n$  labels of  $\tilde{O}(n^{2/3})$  bits and stretch  $2d + 1$ )

## Proof for $k = 2$

(we want  $n$  labels of  $\tilde{O}(n^{2/3})$  bits and stretch  $2d + 1$ )

**Definitions:** Given a set of landmarks  $L \subset V$ :

$$B_L(u) = \{v \in V : d(u, v) < d(u, L)\}$$



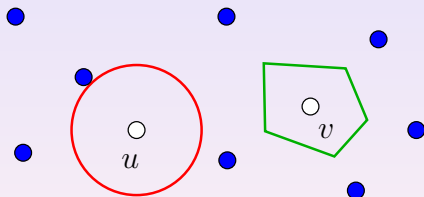
# Proof for $k = 2$

(we want  $n$  labels of  $\tilde{O}(n^{2/3})$  bits and stretch  $2d + 1$ )

**Definitions:** Given a set of landmarks  $L \subset V$ :

$$B_L(u) = \{v \in V : d(u, v) < d(u, L)\}$$

$$C_L(v) = \{u \in V : d(u, v) < d(u, L)\}$$



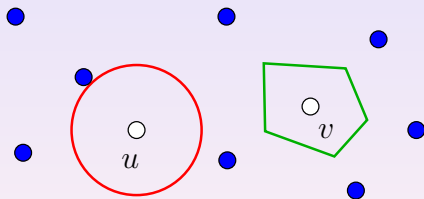
# Proof for $k = 2$

(we want  $n$  labels of  $\tilde{O}(n^{2/3})$  bits and stretch  $2d + 1$ )

**Definitions:** Given a set of landmarks  $L \subset V$ :

$$B_L(u) = \{v \in V : d(u, v) < d(u, L)\} \Rightarrow u \in C_L(v)$$

$$C_L(v) = \{u \in V : d(u, v) < d(u, L)\} \Rightarrow v \in B_L(u)$$



👉  $v \in B_L(u)$  iff  $u \in C_L(v)$



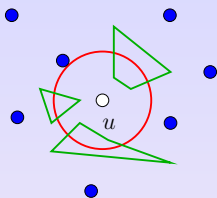
# Choosing the Landmarks

(Sampling Lemma from TZ [SPAA '01])

**Select  $L$  such that**

- 1  $|L| \sim n^{2/3}$
- 2  $\forall u, |B_L(u)| \ \& \ |C_L(u)| \sim n^{1/3}$

# Storage

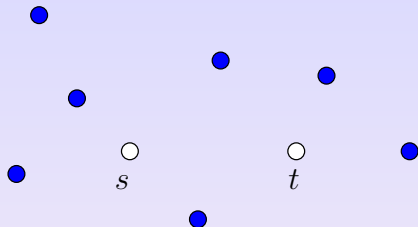


$$S(u) := L \cup B_L(u) \cup \left( \bigcup_{v \in B_L(u)} C_L(v) \right)$$

**Storage for  $u$ :**  $S(u)$  and the distance from  $u$  to every  $v \in S(u)$ , plus its closest landmarks  $l_v$ .

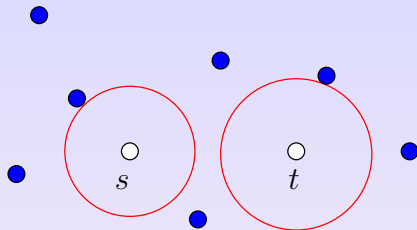
[  $|S(u)| = \tilde{O}(n^{2/3})$ , storage ✓ ]

## Querying between $s$ and $t$



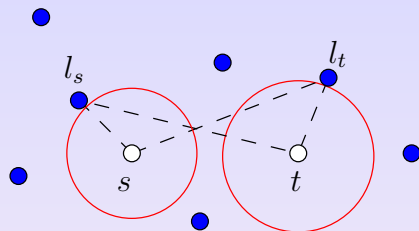
**If**  $t \in S(s)$ , **then** returns  $d(s, t)$

## Querying between $s$ and $t$



**If  $t \in S(s)$ , then** returns  $d(s, t)$

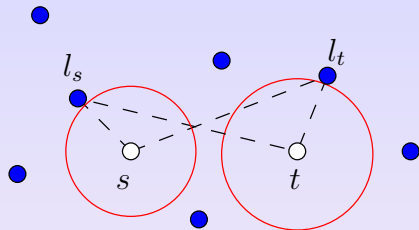
## Querying between $s$ and $t$



**If**  $t \in S(s)$ , **then** returns  $d(s, t)$

**else** returns  $\min \{d(s, l_s) + d(l_s, t), d(t, l_t) + d(l_t, t)\}$

## Querying between $s$ and $t$

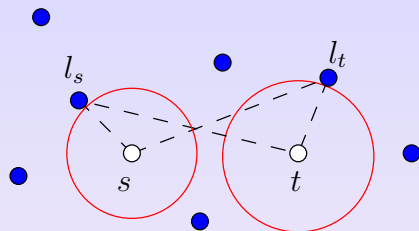


**If**  $t \in S(s)$ , **then** returns  $d(s, t)$

**else** returns  $\min \{d(s, l_s) + d(l_s, t), d(t, l_t) + d(l_t, t)\}$

[dictionary and 2-level hash table, query time ✓]

## Querying between $s$ and $t$



**If**  $t \in S(s)$ , **then** returns  $d(s, t)$

**else** returns  $\min \{d(s, l_s) + d(l_s, t), d(t, l_t) + d(l_t, t)\}$

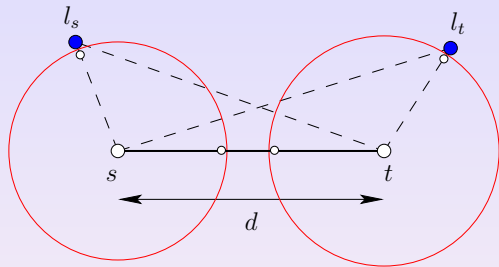
[stretch  $2d + 1$ ?]

☞ If  $t \notin S(s)$ , then  $B_L(s) \cap B_L(t) = \emptyset$

[otherwise  $\exists w \in B_L(t) \cap B_L(s) \Rightarrow t \in C_L(w)$  and  $w \in B_L(s) \Rightarrow t \in S(s)$ ]

If  $t \notin S(s)$ , i.e.,  $B_L(s) \cap B_L(t) = \emptyset$

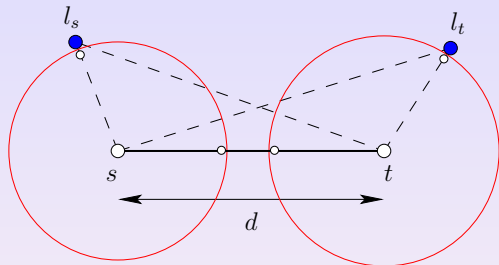
W.l.o.g.  $d(s, l_s) \leq d(t, l_t)$





If  $t \notin S(s)$ , i.e.,  $B_L(s) \cap B_L(t) = \emptyset$

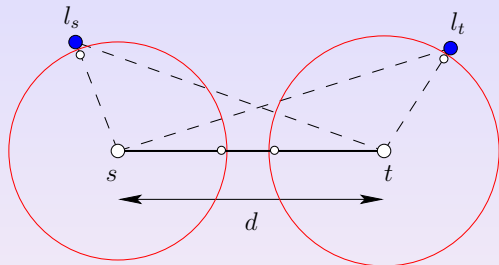
W.l.o.g.  $d(s, l_s) \leq d(t, l_t)$



$$\Rightarrow [d(s, l_s) - 1] + 1 + [d(t, l_t) - 1] \leq d$$

If  $t \notin S(s)$ , i.e.,  $B_L(s) \cap B_L(t) = \emptyset$

W.l.o.g.  $d(s, l_s) \leq d(t, l_t)$

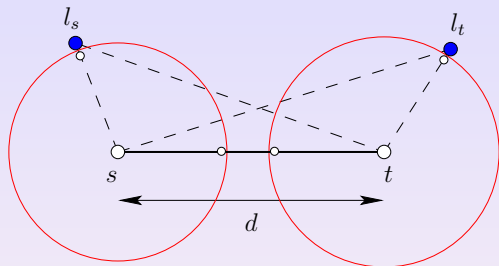


$$\Rightarrow [d(s, l_s) - 1] + 1 + [d(t, l_t) - 1] \leq d$$

$$\Rightarrow 2d(s, l_s) \leq d + 1$$

If  $t \notin S(s)$ , i.e.,  $B_L(s) \cap B_L(t) = \emptyset$

W.l.o.g.  $d(s, l_s) \leq d(t, l_t)$



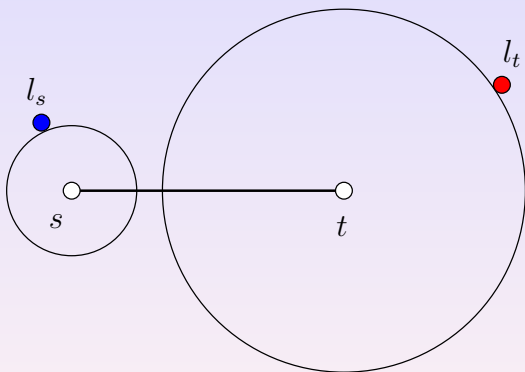
$$\Rightarrow [d(s, l_s) - 1] + 1 + [d(t, l_t) - 1] \leq d$$

$$\Rightarrow 2d(s, l_s) \leq d + 1$$

$$\Rightarrow \hat{d} \leq 2d(s, l_s) + d \leq 2d + 1$$

## Observation

Routing with  $n^{2/3}$ -bit routing tables, **polylog** addresses and stretch  $< 3d$  is not known. Routing query is not symmetric!



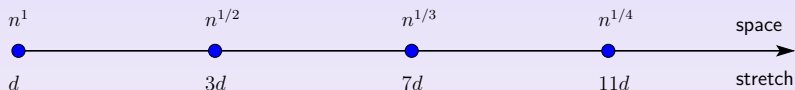
The route  $l_s \rightarrow t$  is space consuming: either  $l_s$ 's table or  $t$ 's address is large.

# What about Compact Routing?

**Best routing scheme [Thorup-Zwick (SPAA '01)]**

achieves stretch  $(4k - 5)d$  and routing tables of size  $\tilde{O}(n^{1/k})$ .

[TZ01]

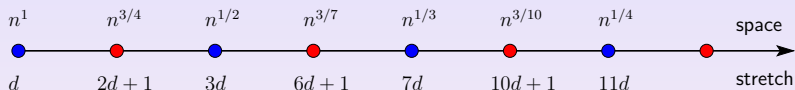


# What about Compact Routing?

**Best routing scheme [Thorup-Zwick (SPAA '01)]**

achieves stretch  $(4k - 5)d$  and routing tables of size  $\tilde{O}(n^{1/k})$ .

[TZ01][us]



## Theorem (2nd contribution)

Let  $k \geq 2$ . Every unweighted graph has a routing scheme with tables of size  $\tilde{O}(n^{3/(3k-2)})$ , stretch  $(4k - 6)d + 1$ , and polylog addresses.

# Conclusion

Not clear if same space-stretch trade-offs can be achieved for both problems, even for  $k = 2$ .

	stretch	size
Distance Labeling	$2d + 1$	$\tilde{O}(n^{2/3})$
Compact Routing	$2d + 1$ $d + O(1)$	$\tilde{O}(n^{3/4})$ $\tilde{\Omega}(n)$

## Conclusion

Not clear if same space-stretch trade-offs can be achieved for both problems, even for  $k = 2$ .

	stretch	size
Distance Labeling	$2d + 1$	$\tilde{O}(n^{2/3})$
Compact Routing	$2d + 1$ $d + O(1)$	$\tilde{O}(n^{3/4})$ $\tilde{\Omega}(n)$

**Thank You!**