6.S899 Distributed Graph Algorithms Lecture 8 MIT 11/07/2014

Material covered:

Distributed (2k-1)-Spanners as presented in chapters 1.2, 3, 4.1, 4.2 and 5.1 in "Surender Baswana and Sandeep Sen, A Simple and Linear Time Randomized Algorithm for Computing Sparse Spanners in Weighted Graphs, Random Structures & Algorithms (Wiley) volume 30, issue 4, pages 532-563, 2007."

LAST LECTURE:

- [ ] $1+o(1)$ - APPROXIMATE SINGLE SOURCE SHORTEST PATH IN $O(\sqrt{n}\, D^{1/4} + D)$ TIME.

TODAY:

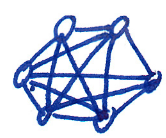
- [ ] $(2k-1)$ - SPANNER OF EXPECTED SIZE $O(k\, n^{1+1/k})$ IN $O(k^2)$ ROUNDS USING $O(k\, m)$ MESSAGES.

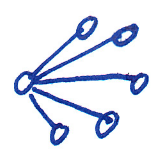
DEF: LET $G=(V,E,w)$ BE A WEIGHTED GRAPH

A (SPARSE) SUBGRAPH $S=(V,E_S,w) \subseteq G$ IS A t-SPANNER OF G IF $d_S(u,v) \le t \cdot d_G(u,v)$ FOR ALL $u,v \in V$

STRETCH t

EXAMPLE: A STAR IS A 2-SPANNER OF AN UNWEIGHTED CLIQUE





FOR ANY $x \neq y$: $d_G(x,y)=1$ $d_S(x,y) \le 2$

APPLICATIONS:

- COMPACT ROUTING TABLES
- SYNCHRONIZERS
- ALL PAIRS APPROXIMATE SHORTEST PATHS ← NOT AS GOOD AS $1+o(1)$

LOWER BOUND: $|E_S| = \Omega(n^{1+1/k})$ FOR STRETCH $t \in \{2k, 2k-1\}$

(BASED ON A CONJECTURE BY ERDŐS et al., PROVEN ONLY FOR $k \in \{1,2,3,5\}$)

DEF: $P_t(x,y)$: "$x,y \in V$ CONNECTED IN (V,E_S) BY PATH OF $\le t$ EDGES EACH OF WEIGHT AT MOST $\omega(x,y)$."

NOTE ☺: IF $P_t(x,y)$ HOLDS FOR EVERY $(x,y) \in E$, THEN (V,E_S) IS A t-SPANNER.

(AS P_t EXTENDS FROM EDGES TO PATHS IN G)

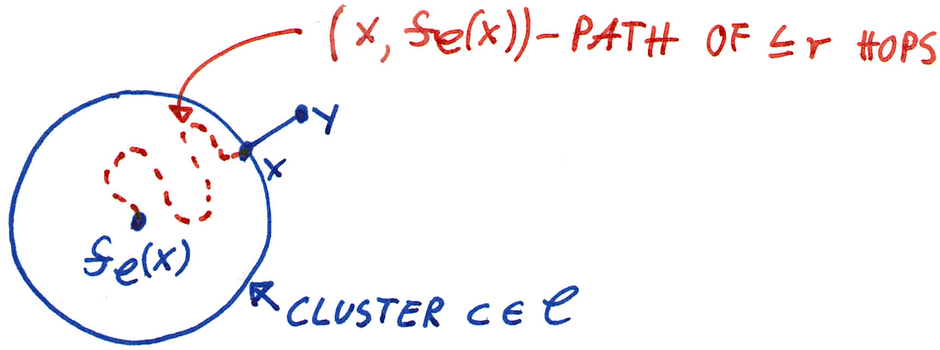
DEF:

- $C \subseteq V$ IS CLUSTER
- CLUSTERING $\mathcal{C}$ OF V' PARTITIONS $V' \subseteq V$ INTO CLUSTERS
- EACH CLUSTER HAS A CENTER.
- $\mathfrak{s}_{\mathcal{C}} : V \longrightarrow V$

  $u \longmapsto$ CENTER OF THE CLUSTER THAT v BELONGS TO.

  REPRESENTS CLUSTERING $\mathcal{C}$.
- $\mathcal{E} \subseteq E$ INDUCES CLUSTERING VIA CONNECTED COMPONENTS IN $\mathcal{E}$.
- $\mathcal{E}(c) \subseteq \mathcal{E}$ DENOTE EDGES IN CLUSTER c (INDUCED BY $\mathcal{E}$)
- RADIUS OF CLUSTER c IN A CLUSTERING INDUCED BY $\mathcal{E}$:

  SMALLEST $r \in \mathbb{N}$ s.t. FOR EACH $(x,y) \in E \setminus \mathcal{E}$, WHERE $x \in c$, THERE IS A $(x, \mathfrak{s}_{\mathcal{C}}(x))$-PATH IN $\mathcal{E}(c)$ OF $\leq r$ HOPS, WHERE EACH EDGE HAS WEIGHT $\leq \omega(x,y)$.

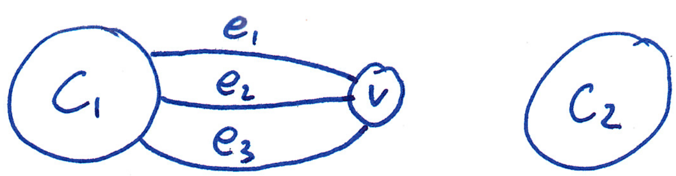


- CLUSTERING $\mathcal{C}$ INDUCED BY $\mathcal{E} \subseteq E$ IS OF RADIUS $\leq i$ IN G IF EACH $c \in \mathcal{C}$ HAS RADIUS $\leq i$.

NOTATION: IN ITERATION i (OF THE ALGORITHM BELOW.

- $E_i \subseteq E \setminus E_S$ ARE EDGES STILL CONSIDERED TO BE INCLUDED IN THE SPANNER. AFTER ITERATION i
- $E_i(x,c) \subseteq E_i$: EDGES BETWEEN $x \in V$ AND $c \in \mathcal{C}$
- $\min(F)$: LEAST WEIGHT EDGE FROM $F \subseteq E$.

EXAMPLE:



$E_i(v, C_1) = \{e_1, e_2, e_3\}$

$E_i(v, C_2) = \emptyset$

ALGO (OUTLINE):

$E_S := \emptyset$; // IS UPDATED TO BE THE SPANNER

$E_0 := E$; // EDGES FOR WHICH P_{2i-1} DOES NOT HOLD YET ARE STORED IN E_i

$\mathcal{E}_0 := \emptyset$; // EDGES IN $\mathcal{E}_i \subseteq E_S$ DEFINE $\mathcal{C}_i$ IN THE END

$V_0 := V$; // V_i ARE THE "ACTIVE" NODES (FORMING $\mathcal{C}_i$)

PHASE I: "FORM CLUSTERS", START WITH $\mathcal{C}_0 = \{\{v\} \mid v \in V\}$;

FOR $i = 1, \ldots, k-1$ DO

- GIVEN: CLUSTERING $\mathcal{C}_{i-1}$ OF RADIUS $i-1$
- MOVE SOME E_{i-1}-EDGES TO E_S (ADAPT CLUSTERING)
- REMOVE EDGES FROM E_{i-1} WHERE P_{2i-1} HOLDS

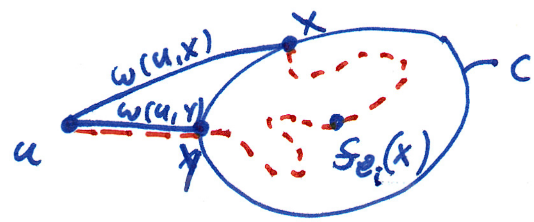
PHASE II: "VERTEX–CLUSTER JOINING"

- TAKE CARE OF REMAINING E_k-EDGES

LEM: Let • $\mathcal{C}_i$ BE A $\mathcal{E}_i$-INDUCED CLUSTERING OF RADIUS i IN $G=(V_i, \mathcal{E}_i \cup E_i$
• $c \in \mathcal{C}_i$

IF $\mathcal{E}_i \subseteq E_S$, THEN FOR ANY $u \notin c$, ADDING $\min(E_i(u,c))$ TO E_S ENSURES: $P_{2i+1}(e)$ HOLDS FOR ANY $e \in E_i(u,c)$.

PROOF: LET $(u,y) := \min(E_i(u,c))$



LET $(u,x) \in E_i(u,c)$

$\Rightarrow \exists$ $(y, S_e(y))$-PATH OF i HOPS IN c AND ALL EDGES HAVE WEIGHT $\leq \omega(u,y)$ (DUE TO THE DEFINITION OF RADIUS)

SIMILAR: $\exists$ $(x, S_e(x))$-PATH OF i HOPS IN c WITH WEIGHTS $\leq \omega(u,x)$

$\Rightarrow$ THE u-x PATH $(u, y, \dots, S_{e_i}(y) = S_{e_i}(x), \dots, x)$ HAS WEIGHT

$$\leq \omega(u,y) + i \cdot \omega(u,y) + i\,\omega(u,x)$$
$$\leq (2i+1)\,\omega(u,x)$$

$E_i(u,c)$

AS $\omega(u,y) = \min(E_i(u,c)) \leq \omega(u,x)$

ETAILS OF PHASE I:

FOR $i = 1, \dots, k-1$ DO

SELECTS SOME CLUSTERS TO GROW

$E_i := E_{i-1}$;

EACH $c \in \mathcal{C}_{i-1}$ JOINS R_i WITH PROBABILITY $n^{-1/k}$;

$\mathcal{E}_i :=$ EDGES IN $\mathcal{E}_{i-1}$ THAT INDUCE R_i;

IF $v \notin c$ FOR ALL $c \in R_i$ THEN

IF $v \in T'(c)$ FOR SOME $c \in R_i$ THEN

$$e_v := \min\Big(\bigcup_{c \in R_i} E_{i-1}(v,c)\Big) \;;$$

ADD e_v TO E_S AND TO $\mathcal{E}_i$;

REMOVE $E_{i-1}(c,v)$ FROM E_i, WHERE c IS THE CLUSTER INCIDENT TO e;

FOR EACH $c' \in \mathcal{C}_{i-1}$ ADJACENT TO v DO

$e' := \min(E_{i-1}(v,c'))$;

IF $\omega(e') < \omega(e)$ THEN

ADD e' TO E_S ; (NOT TO $\mathcal{E}_i$!)

REMOVE $E_{i-1}(v,c')$ FROM E_i ;

ELSE

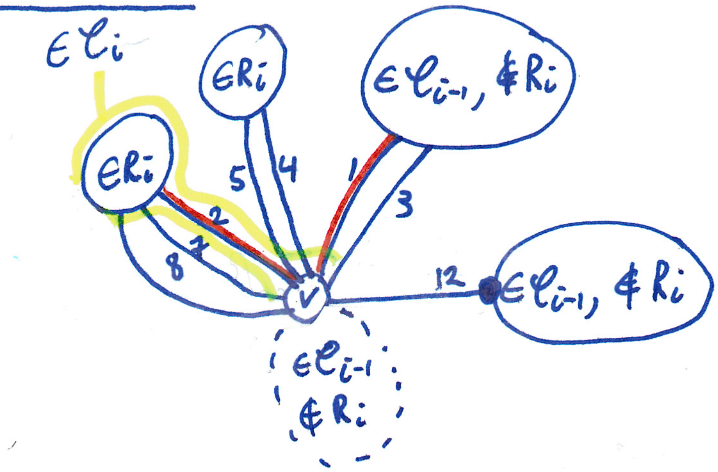
(CONTINUED LATER)

NOTE:

- GROWS CLUSTERS IN R_i WITHOUT MERGING THEM
- INCREASES HOP-RADIUS OF CLUSTERS BY ≤ 1
- REMOVES ALL EDGES INCIDENT TO SPANNER-EDGE e_v OF LOWER WEIGHT

TRIES TO ADD ONLY FEW EDGES TO E_S

EXAMPLE:



- EDGE 3 TO 1'S CLUSTER IS REMOVED FROM E_i.
- 12 IS NOT REMOVED, AS IT IS LARGER THAN 2 AND TO AN NON-R_i-CLUSTER.

V ADJACENT TO A CLUSTER $\in R_i$

$\Rightarrow$
- 2 IS LIGHTEST EDGE TO (ALL) R_i-CLUSTERS AND THUS JOINS E_S AND $\mathcal{E}_i$.
- ALL EDGES (7,8) TO 2'S CLUSTER ARE REMOVED FROM E_i.
- EDGES 5,4 TO OTHER R_i-CLUSTER(S) ARE NOT REMOVED (SO FAR)
- 1 IS ADDED TO E_S (BUT NOT $\mathcal{E}_i$) AS IT IS AN EDGE LIGHTER THAN 2 TO A

ALGORITHM CONTINUED:

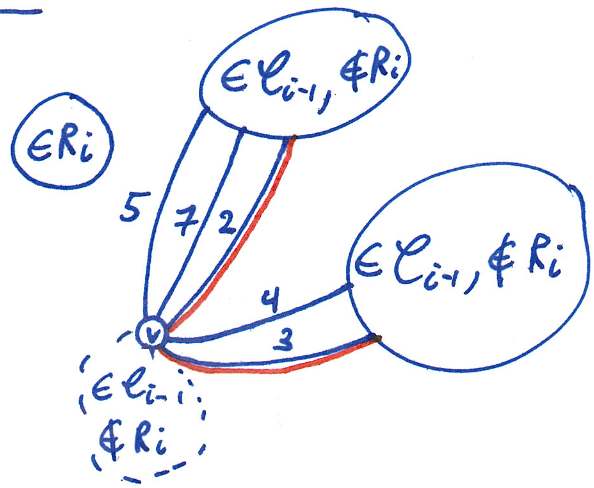
- REMOVE ALL EDGES TO $\mathcal{C}_{i-1}\setminus R_i$-CLUSTERS
- ADD AS FEW EDGES TO E_S AS NECESSARY

FOR EACH $c \in \mathcal{C}_{i-1}$ ADJACENT TO v DO

ADD $\min(E_{i-1}(v,c))$ TO E_S; (NOT TO $\mathcal{E}_i$!)

REMOVE $E_{i-1}(v,c)$ FROM E_i;

EXAMPLE:



v NOT ADJACENT TO ANY CLUSTER $\in R_i$.

$\Rightarrow$ ADD LEAST WEIGHT EDGES TO ADJACENT C_{i-1}-CLUSTERS TO E_S (BUT NOT $\mathcal{E}_i$) AND DELETE OTHERS FROM E_i.

HERE 2,3 ARE ADDED TO E_S, 5,7,4 ARE REMOVED FROM E_i.

NOTE THAT THE TWO ADJACENT CLUSTERS DO NOT GROW (AS THEY ARE NOT IN R_i).

FINAL PART OF PHASE I OF THE ALGORITHM:

$\mathcal{C}_i :=$ THE CLUSTERING INDUCED BY $\mathcal{E}_i$;

REMOVE ALL INTRA-CLUSTER EDGES FROM E_i;

$V_i :=$ ENDPOINTS OF EDGES IN $\mathcal{E}_i \cup E_i$;

THM: FOR EACH ITERATION $i \geq 1$ IT IS TRUE THAT

"$\mathcal{C}_i$ HAS RADIUS i IN $(V_i, \underbrace{\mathcal{E}_i \cup E_i}) \subseteq G$" $(A(i))$

CONSIDERS ONLY $\mathcal{E}_i$-EDGES AND THOSE WHERE P_{2k-1} DOES NOT HOLD YET (WHICH ARE EDGES BETWEEN $\mathcal{C}_i$)

PROOF: INDUCTION ON i (BUT START AT $i=0$):

$i=0$: $\mathcal{C}_0$ IS CLUSTERING OF RADIUS 0. ✓
$\mathbb{L}:\{\{v\}\mid v\in V\}$

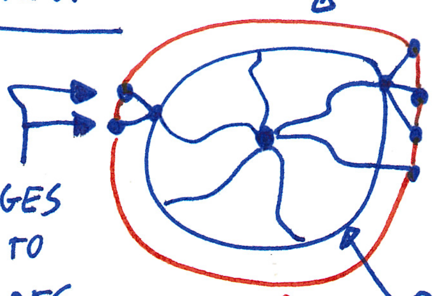
$i-1 \to i,\ i>0$: LET $A(i-1)$ BE TRUE.

A CLUSTER $c \in R_i$ GROWS ONLY BY EDGES e_v ADDED TO $\mathcal{E}_i$ (WHICH DEFINES $\mathcal{C}_i$ IN THE END). ALL EDGES INCIDENT TO v OF WEIGHT $\leq \omega(e_v)$ ARE REMOVED. (ALSO THESE TO NON-R_i-CLUSTERS IN $\mathcal{C}_{i-1}$).

USES $A(i)$ $\Rightarrow$ ALL EDGES IN $\mathcal{E}_i(c)$, AFTER ITERATION i, ARE LIGHTER THAN ALL REMAINING E_i-EDGES INCIDENT TO c.

NEW RADIUS IS $\leq \underbrace{(i-1)}_{A(i-1)} + \underbrace{1}_{\text{EDGE } e_v} = i.$ $\square$

ILLUSTRATION OF PROOF:



THM II: IF $(u,v) \in E$ ELIMINATED IN PHASE I, THEN P_{2k-2} HOLDS FOR (u,v).

PROOF: **CASE I:** (u,v) IS ELIMINATED IN PART I.
(ASSUME w.o.l.g. DUE TO u'S DECISSION)
IT IS: 1) $\omega(\min(E_{i-1}(u, c_v))) \leq \omega(u,v)$
(WHEN v BELONGED TO $c_v \in \mathcal{C}_{i-1}$ AND (u,v) WAS ELIMINATED IN ROUND i)
2) $c \in \mathcal{C}_{i-1}$ HAS RADIUS $i-1$

COMBINE THIS WITH LEM I.
$\Rightarrow P_{2i-1}$ HOLDS FOR (u,v).

CASE II: (u,v) IS ELIMINATED IN PART II.

$\Rightarrow u, v$ JOINED THE SAME CLUSTER $c \in R_i$
$\omega(u,v) \geq \min(E_{i-1}(u,c) \cup E_{i-1}(v,c))$ AS OTHERWISE (u,v) WOULD HAVE BEEN ELIMINATED IN PART I.

$\Rightarrow$ PATH $(u, \underbrace{x, \dots, s_{\mathcal{C}_{i-1}}(x)}_{i-1 \text{ HOPS}} = \underbrace{s_{\mathcal{C}_{i-1}}(y), \dots, y}_{i-1 \text{ HOPS}}, v)$

HAS $1 + i-1 + i-1 + 1 = 2i$ HOPS, AS c HAS RADIUS $i-1$, AND ALL THESE EDGES HAVE WEIGHT LESS THAN (u,v).
$\Rightarrow P_{2i}$ HOLDS FOR (u,v).

AS $i \leq k-1$, THE THM. FOLLOWS.

$\square$

LEM: $\mathbb{E}[|E_S|] = O(k\, n^{1+1/k})$ AFTER PHASE I.

PROOF: LET $v \in V_{i-1}$ IN ITERATION i.

CONSIDER v'S ADJACENT CLUSTERS $c_1, \dots, c_\ell$ S.T.

$$e_1 < e_2 < \dots < e_\ell \,, \quad \text{WHERE } e_j := \min(E_{i-1}(v, c_j)).$$

e_j ADDED TO E_S IFF $c_1, \dots, c_{j-1} \notin R_i$

$$\Rightarrow PR[e_j \in E_S] = (1 - n^{-1/k})^{j-1}$$

$\Rightarrow \mathbb{E}[\# e_j$ ADDED TO E_S IN ITERATION $i]$

$$= \sum_{j=1}^{\ell} (1 - n^{-1/k})^{j-1} \leq n^{1/k}$$

MULTIPLY THIS BY $O(n)$ VERTICES AND $k-1$ ITERATIONS. $\square$

NOTE: v MIGHT BE ADJACENT TO DIFFERENT CLUSTERS IN EACH ITERATION.

LEM III RUNTIME OF PHASE I: $O(k^2)$ MESSAGES: $O(m \cdot k)$

PROOF: IN EACH OF $k-1$ ITERATIONS $i = 1, \dots, k-2$ WE HAVE:

	TIME	MSG
1) SAMPLING + MEMBER NOTIFICATION	$i-1$	m
2) FIND NEAREST R_i-CLUSTER	1	m
3) ADD EDGES TO E_S	0	0
4) REMOVE EDGES FROM E_i	1	m
	$O(i)$	$O(m)$

TOTAL: $\sum_{i=1}^{k-1} O(i) = O(k^2)$ TIME AND $O(k \cdot m)$ MSG.

DETAILS OF PHASE II:

IF $v \in V_{k-1}$ THEN

FOR EACH $c \in \mathcal{C}_{k-1}$ DO

ADD $\min(E_{k-1}(v,c))$ TO E_S;

REMOVE $E_{k-1}(v,c)$ FROM E_{k-1};

• REMOVES ALL REMAINING CLUSTERS. NO R_k INVOLVED!

LEM: AFTER PHASE II, P_{2k-1} HOLDS FOR ALL $e \in E$.

PROOF:

- AS IN CASE I OF THM II'S PROOF, P_{2k-1} HOLDS FOR ALL EDGES ELIMINATED IN PHASE II. $(i=k)$
- ALL EDGES IN E_{k-1} ARE ELIMINATED. $\square$

LEM IV PHASE II ADDS $\leq n^{1+1/k}$ EDGES TO E_S IN TIME $O(k)$ USING $O(m)$ MESSAGES.

PROOF: IT IS $|\mathcal{C}_i| \approx n^{1-i/k}$ AND THUS $|\mathcal{C}_{k-1}| \approx n^{1/k}$

$\Rightarrow \leq |V_{k-1}| \cdot |\mathcal{C}_{k-1}| \approx n^{1+1/k}$ EDGES ADDED.

RUNTIME / MSG CLEAR. $\square$

THM: THE ALGORITHM COMPUTES A $(2k-1)$-SPANNER (V, E_S) IN TIME $O(k^2)$ USING $O(km)$ MESSAGES.

PROOF: COMBINE LEM III + IV AND NOTE σ. $\square$