Triangle Finding and Listing in CONGEST Networks

Taisuke Izumi
Nagoya Institute of Technology

François Le Gall
Kyoto University

Paris, 21 February 2019

Recent Development

Our result (PODC 2017)

In the CONGEST model, there is a $\tilde{O}(n^{2/3})$-round classical algorithm for triangle finding and a $\tilde{O}(n^{3/4})$-round classical algorithm for triangle listing.

New development (Chang, Pettie and Zhang SODA 2019)

In the CONGEST model, there is $\tilde{O}(n^{1/2})$-round classical algorithms for triangle finding and listing.

They show how to partition the set of edges into three sets E_1, E_2, E_3 such that:

- ✓ each connected component of the graph induced by E_1 is well connected
- ✓ the graph induced by E_2 has small arboricity
- ✓ the graph induced by E_3 is sparse

Triangle Finding

three vertices u,v,w such that (u,v) ∈ E, (u,w) ∈ E and (v,w) ∈ E

unweighted (and undirected)

Given a graph G=(V,E), decide if it contains a triangle

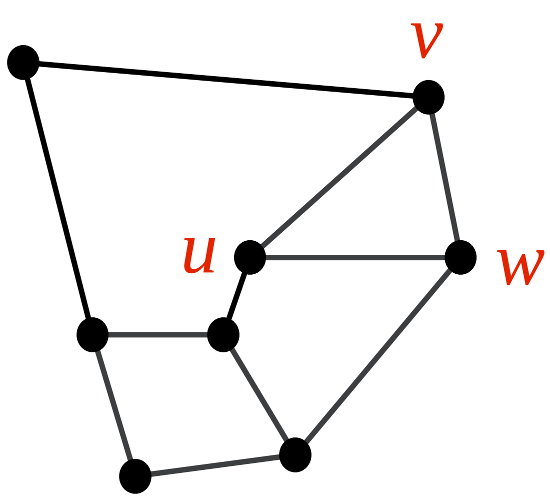
Triangle Finding

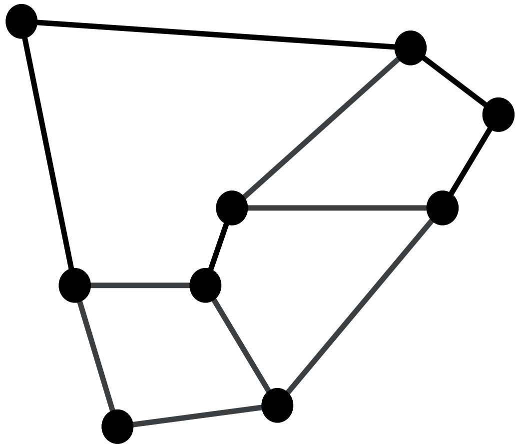
three vertices u,v,w such that (u,v) ∈ E, (u,w) ∈ E and (v,w) ∈ E

unweighted (and undirected)

Given a graph G=(V,E), decide if it contains a triangle

Examples:





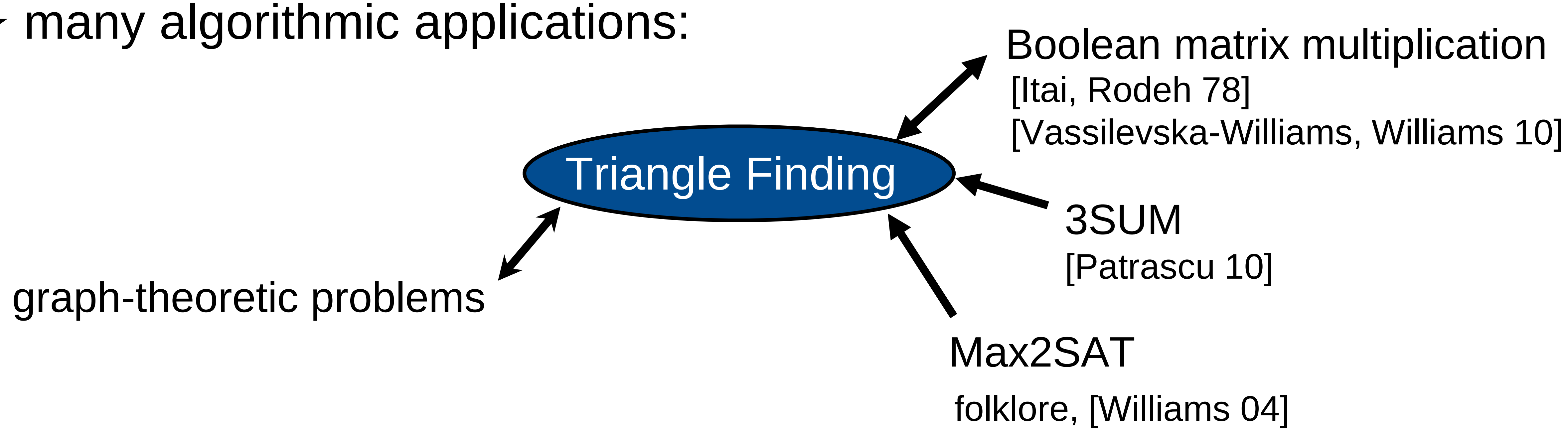
no triangle

Triangle Finding in Sequential Computing

Given a graph G=(V,E), decide if it contains a triangle

★ one of “most elementary” unsettled graph-theoretic problems

★ many algorithmic applications:



★ has become one of the central problems in the field of “fine-grained computational complexity”

Triangle Finding in Distributed Computing

In this work we consider the CONGEST model:

- ✓ network G=(V,E) of n nodes (all nodes have distinct identifiers)
- ✓ each node knows the identifiers of all its neighbors
- ✓ communication between adjacent nodes: one message of O(log n) bits per round

Triangle Finding:

if G has a triangle, then at least one node must output a triangle
(otherwise all nodes should output “not found”)

Triangle Listing:

each triangle of G is output by at least one node
(the same triangle may be output more than once)

- ✓ trivial algorithm using n rounds for both problems:
  “each node sends to each neighbor the list of all its neighbors”

- ✓ Related problem: property testing for triangle-freeness

[Fraigniaud et al. DISC’16] [Fischer et al. PODC’17]

Round Complexity of Triangle Finding/Listing

the tilde notation removes polylog(n) factors

communication allowed even between non-adjacent nodes

	Bounds	Problem	Model	Deterministic or randomized
Dolev et al. DISC'12	$\tilde{O}(n^{1/3})$	Listing	CONGEST clique	deterministic
Censor-Hillel et al. PODC'15	$O(n^{0.1572})$	Finding	CONGEST clique	deterministic
This work	$\tilde{O}(n^{2/3})$	Finding	CONGEST	randomized
This work	$\tilde{O}(n^{3/4})$	Listing	CONGEST	randomized

First algorithms with sublinear round complexity in the CONGEST model

Drucker et al. PODC'14	$\Omega\left(\frac{n}{\exp(\sqrt{\log n})}\right)$	Finding	CONGEST broadcast	deterministic
Pandurangan et al. 2016	$\Omega\left(\frac{n^{1/3}}{\log^3 n}\right)$	Listing	CONGEST clique	randomized
This work	$\Omega\left(\frac{n^{1/3}}{\log n}\right)$	Listing	CONGEST clique	randomized

Note: a lower bound for the CONGEST clique model implies a lower bound for the CONGEST model

Lower bound: Idea of the Proof

- ✓ A graph of n nodes can contain Ω(n^3) triangles (e.g., a random graph)
- ✓ Thus at least one node has to output Ω(n^2) triangles
- ✓ Fact: Ω($t^{2/3}$) edges are needed to form t triangles
- ✓ Thus at least one node have information about Ω($n^{4/3}$) edges
  It must then receive Ω($n^{4/3}$) bits, which requires Ω($n^{1/3}$ / log n) rounds

at each round a node receives at most O(n log n) bits

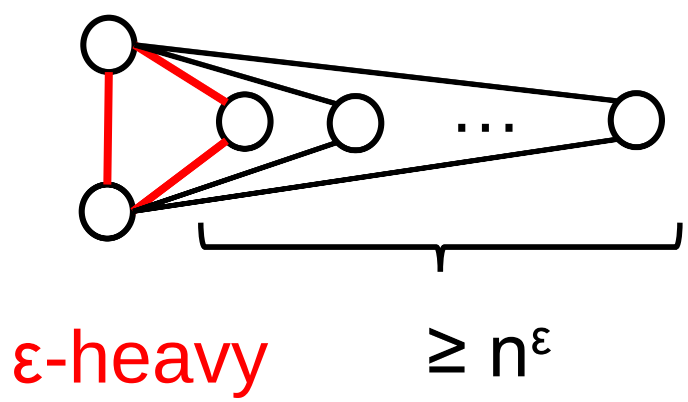
Pandurangan et al. 2016	$\Omega\left(\frac{n^{1/3}}{\log^3 n}\right)$	Listing	CONGEST clique	randomized
This work	$\Omega\left(\frac{n^{1/3}}{\log n}\right)$	Listing	CONGEST clique	randomized

Note: a lower bound for the CONGEST clique model implies a lower bound for the CONGEST model

Upper Bounds: Heavy and Light Triangles

Let ε be any constant such that $0 \leq \varepsilon \leq 1$

We say that a triangle is <u>ε-heavy</u> if one of its edges is shared by at least n^{ε} triangles. Otherwise we say that it is <u>ε-light</u>.



success probability at least 1-1/poly(n)

randomized algorithms

Finding one ε-heavy triangle	$\tilde{O}(n^{1-\varepsilon})$ rounds
Finding one ε-light triangle	$\tilde{O}(n^{1-\varepsilon} + n^{(1+\varepsilon)/2})$ rounds

Taking $\varepsilon = 1/3$ gives the claimed complexity $\tilde{O}(n^{2/3})$ for Finding

Listing all ε-heavy triangles	$\tilde{O}(n^{1-\varepsilon/2})$ rounds
Listing all ε-light triangles	$\tilde{O}(n^{1-\varepsilon} + n^{(1+\varepsilon)/2})$ rounds

same complexity for light triangles

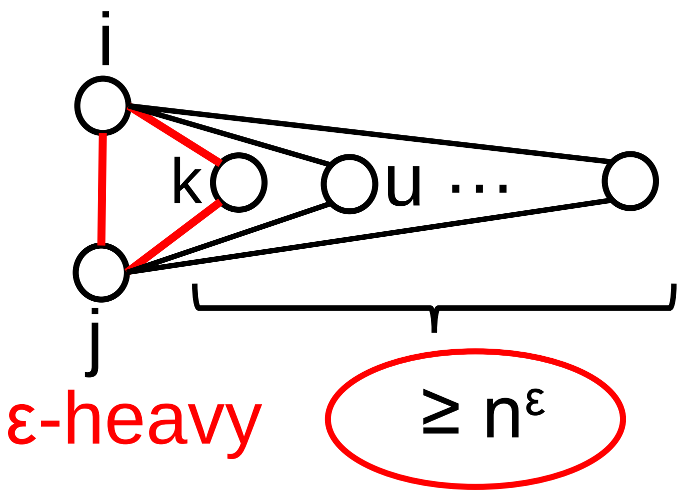
Taking $\varepsilon = 1/2$ gives the claimed complexity $\tilde{O}(n^{3/4})$ for Listing

Listing all ε-heavy triangles in $\tilde{O}(n^{1-\varepsilon/2})$ rounds

1. Each node u of the graph takes a pairwise independent hash function

$$h_u: V \to \{0,1,\dots,\lfloor n^{\varepsilon/2} \rfloor\}$$

(node u tells its neighbors which function it took)



$\Pr[h_u(j)=0 \text{ and } h_u(k)=0] = \Theta(1/n^{\varepsilon})$

re… …vy triangles with success probability 1-1/poly(n)

(expected) size of this set: $O(n^{1-\varepsilon/2})$

N(i) = set of neighbors of i

2. Each node i sends to each neighbor u the set $\{v \in N(i) \text{ such that } h_u(v)=0\}$.
Node u then outputs all the triangles he learns from what he receives.

Correctness:

For any ε-heavy triangle (i,j,k), where {i,j} is the edge shared by at least n^{ε} triangles, the following happens with probability $\Theta(1/n^{\varepsilon})$:

- ✓ u receives j and k from i, and
- ✓ u receives k from j.

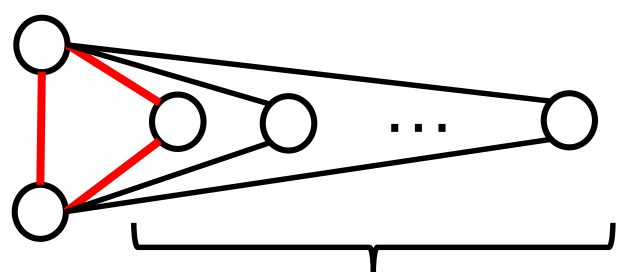
In this case u outputs the triangle (i,j,k).

Conclusion: with constant probability at least one node u will output (i,j,k)

Heavy Triangles

Let ε be any constant such that $0 \leq \varepsilon \leq 1$

We say that a triangle is ε-heavy if one of its edges is shared by at least n^{ε} triangles. Otherwise we say that it is ε-light.



ε-heavy $\geq n^{\varepsilon}$

randomized algorithms

Finding one ε-heavy triangle	$\tilde{O}(n^{1-\varepsilon})$ rounds
Finding one ε-light triangle	$\tilde{O}(n^{1-\varepsilon} + n^{(1+\varepsilon)/2})$ rounds

Taking ε = 1/3 gives the claimed complexity $\tilde{O}(n^{2/3})$ for Finding

Listing all ε-heavy triangles	$\tilde{O}(n^{1-\varepsilon/2})$ rounds
Listing all ε-light triangles	$\tilde{O}(n^{1-\varepsilon} + n^{(1+\varepsilon)/2})$ rounds

same algorithm

Taking ε = 1/2 gives the claimed complexity $\tilde{O}(n^{3/4})$ for Listing

Listing all ε-light triangles in $\tilde{O}(n^{1-\varepsilon}+n^{(1+\varepsilon)/2})$ rounds

1. Select randomly $\tilde{\Theta}(n^{1-\varepsilon})$ nodes. Let X be the set of selected nodes.
2. Each node tells its neighbors if it has been selected or not.
3. Each node k sends the set $N(k) \cap X$ to all its neighbors.

for instance: each node selects itself with probability $\approx 1/n^{\varepsilon}$

First key definition: $\Delta(X)$ = Set of all pairs of vertices of the graph that are not in the neighborhood of a same vertex in X

$\Delta(X)$ = all pairs of vertices except the green ones

Listing all ε-light triangles in $\tilde{O}(n^{1-\varepsilon}+n^{(1+\varepsilon)/2})$ rounds

1. Select randomly $\tilde{\Theta}(n^{1-\varepsilon})$ nodes. Let X be the set of selected nodes.
2. Each node tells its neighbors if it has been selected or not.
3. Each node k sends the set N(k) ∩ X to all its neighbors.

First key definition: $\Delta(X)$ = Set of all pairs of vertices of the graph that are not in the neighborhood of a same vertex in X

Claim 1: With high probability (on the choice of X), each ε-light triangle has its three edges in $\Delta(X)$.

with high probability, none of them is put in X, in which case {i,j} is in $\Delta(X)$.

each is put in X with probability ≈ $1/(n^{\varepsilon} \log n)$

Listing all ε-light triangles in $\tilde{O}(n^{1-\varepsilon}+n^{(1+\varepsilon)/2})$ rounds

1. Select randomly $\tilde{\Theta}(n^{1-\varepsilon})$ nodes. Let X be the set of selected nodes.
2. Each node tells its neighbors if it has been selected or not.
3. Each node k sends the set $N(k) \cap X$ to all its neighbors.

First key definition: $\Delta(X)$ = Set of all pairs of vertices of the graph that are not in the neighborhood of a same vertex in X

Claim 1: With high probability (on the choice of X), each ε-light triangle has its three edges in $\Delta(X)$.

Goal: list all triangles with three edges in $\Delta(X)$.

Listing all ε-light triangles in $\tilde{O}(n^{1-\varepsilon}+n^{(1+\varepsilon)/2})$ rounds

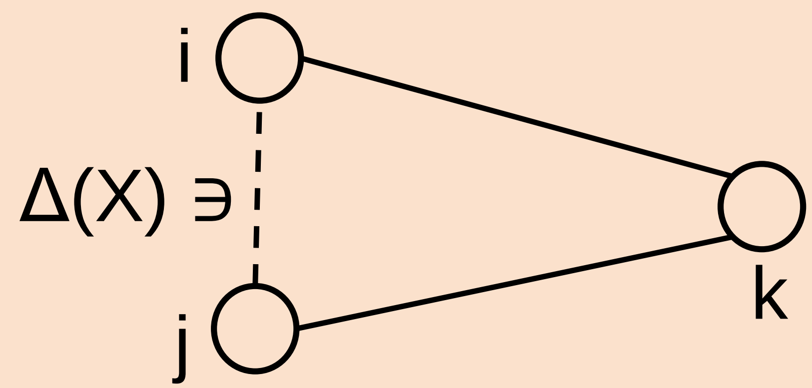
1. Select randomly $\tilde{\Theta}(n^{1-\varepsilon})$ nodes. Let X be the set of selected nodes.
2. Each node tells its neighbors if it has been selected or not.
3. Each node k sends the set N(k) ∩ X to all its neighbors.

Second key definition:

Let {i,k} be any edge.

Consider the set S(i,k) of all vertices j such that {j,k} is an edge and {i,j} ∈ Δ(X).

It can be computed by k without communication from the information received at Step 3.



Claim 2: With high probability (on the choice of X), the average value of |S(i,k)|, over all edges {i,k} of the graph, is $O(n^{\varepsilon})$.

Proof: proof of Claim 1 + a counting argument

Listing all ε-light triangles in $\tilde{O}(n^{1-\varepsilon}+n^{(1+\varepsilon)/2})$ rounds

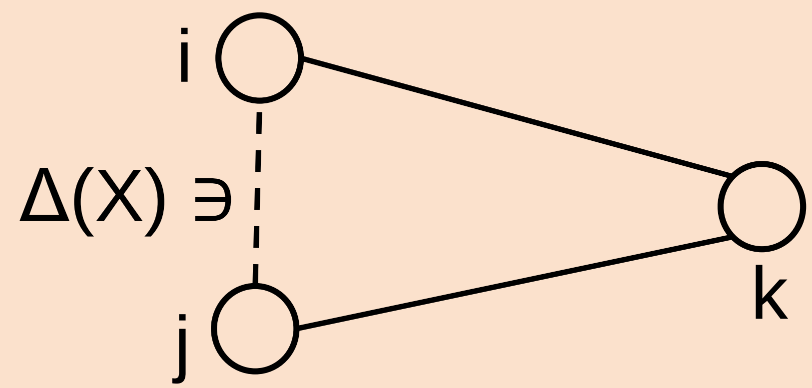
1. Select randomly $\tilde{\Theta}(n^{1-\varepsilon})$ nodes. Let X be the set of selected nodes.
2. Each node tells its neighbors if it has been selected or not.
3. Each node k sends the set $N(k) \cap X$ to all its neighbors.
4. Each node k sends to each of its neighbor i the set S(i,k).
   The neighbor i outputs all triangles consisting of i, k and a vertex in S(i,k)

Second key definition:

Let {i,k} be any edge.

Consider the set S(i,k) of all vertices j such that {j,k} is an edge and $\{i,j\} \in \Delta(X)$.

It can be computed by k without communication from the information received at Step 3.



Claim 2: With high probability (on the choice of X), the average value of |S(i,k)|, over all edges {i,k} of the graph, is $O(n^{\varepsilon})$.

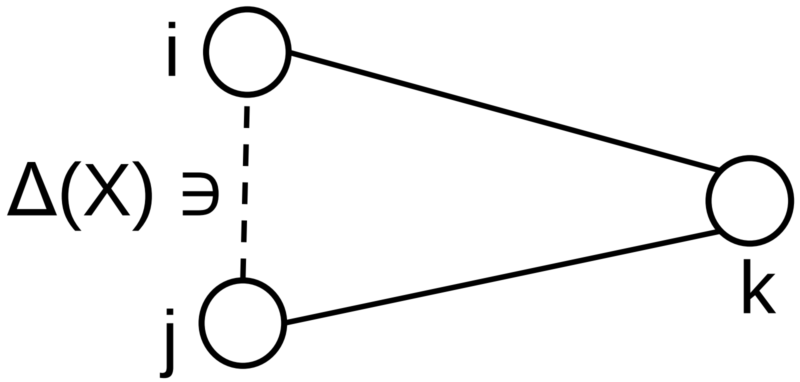
Proof: proof of Claim 1 + a counting argument

Listing all ε-light triangles in $\tilde{O}(n^{1-\varepsilon}+n^{(1+\varepsilon)/2})$ rounds

1. Select randomly $\tilde{\Theta}(n^{1-\varepsilon})$ nodes. Let X be the set of selected nodes.
2. Each node tells its neighbors if it has been selected or not.
3. Each node k sends the set N(k) ∩ X to all its neighbors.
4. Each node k sends to each of its neighbor i the set S(i,k).
   The neighbor i outputs all triangles consisting of i, k and a vertex in S(i,k)

Correctness: at Step 4 each triangle with at least one edge in Δ(X), and thus each ε-light triangle, is output.

Round complexity of Step 4:
maximum value of |S(i,k)|, not its average!



Claim 2: With high probability (on the choice of X), the average value of |S(i,k)|, over all edges {i,k} of the graph, is $O(n^{\varepsilon})$.

Proof: proof of Claim 1 + a counting argument

Listing all ε-light triangles in $\tilde{O}(n^{1-\varepsilon}+n^{(1+\varepsilon)/2})$ rounds

1. Select randomly $\tilde{\Theta}(n^{1-\varepsilon})$ nodes. Let X be the set of selected nodes.
2. Each node tells its neighbors if it has been selected or not.
3. Each node k sends the set $N(k) \cap X$ to all its neighbors.
4. Each node k sends to each of its neighbor i the set S(i,k) if $|S(i,k)| \leq n^{\varepsilon+\delta}$. The neighbor i outputs all triangles consisting of i, k and a vertex in S(i,k)
5. Deal with the triangles involving edges {i,k} such that $|S(i,k)| \geq n^{\varepsilon+\delta}$. (Details omitted.)

Trick: send S(i,k) only if its size does not exceed too much the average

It remains to deal with the S(i,k) that exceed the average by a factor n^{δ}

idea: there is only a small number of such edges, so we can apply recursively the algorithm on a sparser graph

Listing all ε-light triangles in $\tilde{O}(n^{1-\varepsilon}+n^{(1+\varepsilon)/2})$ rounds

1. Select randomly $\tilde{\Theta}(n^{1-\varepsilon})$ nodes. Let X be the set of selected nodes.
2. Each node tells its neighbors if it has been selected or not.
3. Each node k sends the set $N(k) \cap X$ to all its neighbors.
4. Each node k sends to each of its neighbor i the set S(i,k) if $|S(i,k)| \leq n^{\varepsilon+\delta}$. The neighbor i outputs all triangles consisting of i, k and a vertex in S(i,k)
5. Deal with the triangles involving edges {i,k} such that $|S(i,k)| \geq n^{\varepsilon+\delta}$. (Details omitted.)

Trick: send S(i,k) only if its size does not exceed too much the average

It remains to deal with the S(i,k) that exceed the average by a factor n^{δ}

idea: there is only a small number of such edges, so we can apply recursively the algorithm on a sparser graph

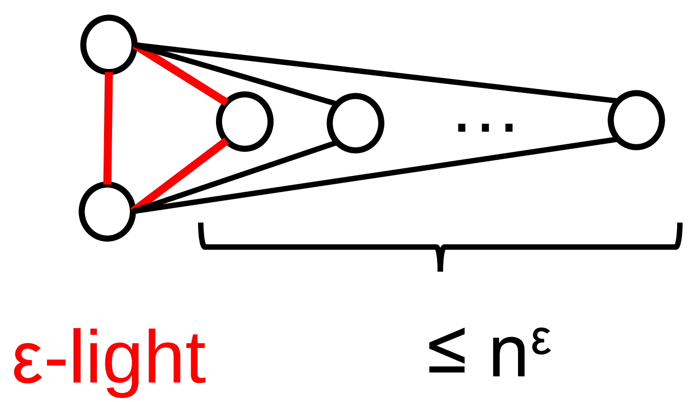
Conclusion: The round complexity of Steps 4-5 is $\tilde{O}(n^{(1+\varepsilon)/2})$

(the optimal choice for δ is $\delta=(1-\varepsilon)/2$)

Light Triangles

Let ε be any constant such that $0 \le \varepsilon \le 1$

We say that a triangle is <u>ε-heavy</u> if one of its edges is shared by at least n^{ε} triangles. Otherwise we say that it is <u>ε-light</u>.



randomized algorithms

Finding one ε-heavy triangle	$\tilde{O}(n^{1-\varepsilon})$ rounds
Finding one ε-light triangle	$\tilde{O}(n^{1-\varepsilon} + n^{(1+\varepsilon)/2})$ rounds

Taking $\varepsilon = 1/3$ gives the claimed complexity $\tilde{O}(n^{2/3})$ for finding

Listing all ε-heavy triangles	$\tilde{O}(n^{1-\varepsilon/2})$ rounds
Listing all ε-light triangles	$\tilde{O}(n^{1-\varepsilon} + n^{(1+\varepsilon)/2})$ rounds

same algorithm

Taking $\varepsilon = 1/2$ gives the claimed complexity $\tilde{O}(n^{3/4})$ for listing

Conclusion

We constructed the first sublinear-time algorithms for Triangle Finding and Listing in the CONGEST model:

	Bounds	Problem	Model	Deterministic or randomized
Dolev et al. DISC'12	$\tilde{O}(n^{1/3})$	Listing	CONGEST clique	deterministic
Censor-Hillel et al. PODC'15	$O(n^{0.1572})$	Finding	CONGEST clique	deterministic
This work	$\tilde{O}(n^{2/3})$	Finding	CONGEST	randomized
This work	$\tilde{O}(n^{3/4})$	Listing	CONGEST	randomized

Pandurangan et al. 2016	$\Omega\left(\frac{n^{1/3}}{\log^3 n}\right)$	Listing	CONGEST clique	randomized
This work	$\Omega\left(\frac{n^{1/3}}{\log n}\right)$	Listing	CONGEST clique	randomized

Open problem:

What about quantum algorithms?

Conclusion

We constructed the first sublinear-time algorithms for Triangle Finding and Listing in the CONGEST model:

	Bounds	Problem	Model	Deterministic or randomized
Dolev et al. DISC'12	$\tilde{O}(n^{1/3})$	Listing	CONGEST clique	deterministic
Censor-Hillel et al. PODC'15	$O(n^{0.1572})$	Finding	CONGEST clique	deterministic
This work	$\tilde{O}(n^{2/3})$	Finding	CONGEST	randomized
This work	$\tilde{O}(n^{3/4})$	Listing	CONGEST	randomized
Chang et al. SODA 2019	$\tilde{O}(n^{1/2})$	Listing	CONGEST	randomized

They show how to partition the set of edges into three sets E_1, E_2, E_3 such that:

- ✓ each connected component of the graph induced by E_1 is well connected
- ✓ the graph induced by E_2 has small arboricity
- ✓ the graph induced by E_3 is sparse