Local approximation of the Maximum Cut in regular graphs

E. Bamas and L. Esperet

EPFL, Lausanne
Université Grenoble Alpes, France

WG 2019
Vall de Núria, June 19^{th}, 2019

Local/Distributed algorithms

The distributed models on graphs:

- Synchronous rounds, at each round each vertex can send/receive messages to/from its neighbors in the graph
- No failure
- Infinite local computational power

LOCAL model:

- Unique IDs
- Messages of unlimited size

CONGEST model:

- Unique IDs
- Messages of $O(\log n)$ bits

PO model:

- Port numbering+edge orientation
- Messages of $O(\log n)$ bits

Local/Distributed algorithms

The distributed models on graphs:

- Synchronous rounds, at each round each vertex can send/receive messages to/from its neighbors in the graph
- No failure
- Infinite local computational power

LOCAL model:

- Unique IDs
- Messages of unlimited size

CONGEST model:

- Unique IDs
- Messages of $O(\log n)$ bits

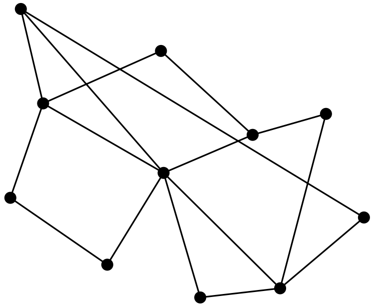
PO model:

- Port numbering+edge orientation
- Messages of $O(\log n)$ bits

How well can we approximate the Maximum Cut in any of these models ?

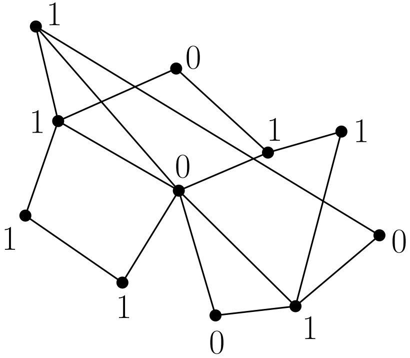
The Maximum (directed) Cut in the distributed model

Problem definition: partition the vertex set into 2 parts, A and B, maximizing the number of crossing edges (crossing specifically from A to B in the directed case)



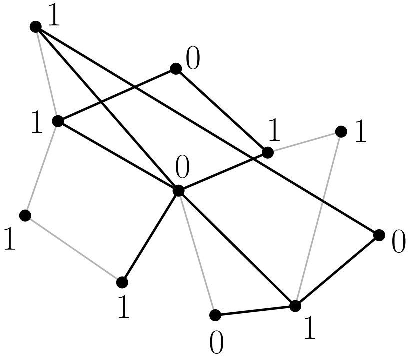
The Maximum (directed) Cut in the distributed model

Problem definition: partition the vertex set into 2 parts, A and B, maximizing the number of crossing edges (crossing specifically from A to B in the directed case)



The Maximum (directed) Cut in the distributed model

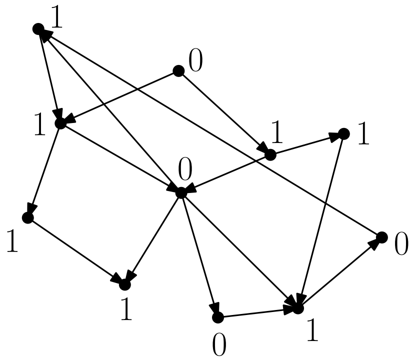
Problem definition: partition the vertex set into 2 parts, A and B, maximizing the number of crossing edges (crossing specifically from A to B in the directed case)



$\frac{1}{2}$-approximation in 0 communication rounds !

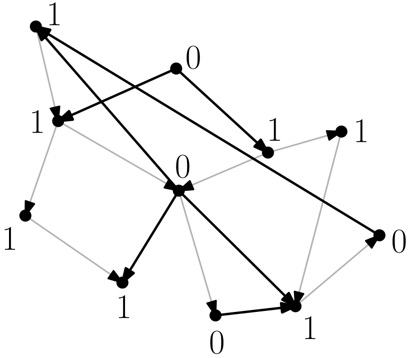
The Maximum (directed) Cut in the distributed model

Problem definition: partition the vertex set into 2 parts, A and B, maximizing the number of crossing edges (crossing specifically from A to B in the directed case)



The Maximum (directed) Cut in the distributed model

Problem definition: partition the vertex set into 2 parts, A and B, maximizing the number of crossing edges (crossing specifically from A to B in the directed case)



$\frac{1}{4}$-approximation in 0 communication rounds !

The Maximum (directed) Cut in the distributed model

Problem definition: partition the vertex set into 2 parts, A and B, maximizing the number of crossing edges (crossing specifically from A to B in the directed case)

State-of-the-art in the CONGEST model:

Problem	approx.	rounds	rand/det	source
MaxCut	$1/2$	0	rand.	folklore
MaxDiCut	$1/4$	0	rand.	folklore
MaxCut d-regular graphs + triangle-free	$\approx 1/2 + 0.28/\sqrt{d}$	$O(1)$	rand.	[1]
MaxCut	$1/2 - \epsilon$	$O(\log^* n)$	det.	[2]
MaxDiCut	$1/3 - \epsilon$	$O(\log^* n)$	det.	[2]
MaxDiCut	$1/2 - \epsilon$	$O(\epsilon^{-1})$	rand.	[2]

[1] J. Hirvonen, J. Rybicki, S. Schmid, J. Suomela (Electron. J. of Combin. 2017)

[2] K. Kawarabayashi, G. Schwartzman (DISC 18)

The Maximum (directed) Cut in the distributed model

There appears to be a gap between randomized and deterministic running times

The Maximum (directed) Cut in the distributed model

There appears to be a gap between randomized and deterministic running times

We know randomized $\Omega(1)$-approximation in $O(1)$ rounds but only $O(\log^* n)$ deterministic rounds

The Maximum (directed) Cut in the distributed model

There appears to be a gap between randomized and deterministic running times

We know randomized $\Omega(1)$-approximation in $O(1)$ rounds but only $O(\log^* n)$ deterministic rounds

What is the deterministic complexity of $\Omega(1)$-approximation for MaxCut ?

The Maximum (directed) Cut in the distributed model

There appears to be a gap between randomized and deterministic running times

We know randomized $\Omega(1)$-approximation in $O(1)$ rounds but only $O(\log^* n)$ deterministic rounds

What is the deterministic complexity of $\Omega(1)$-approximation for MaxCut ?

Our negative results:

The Maximum (directed) Cut in the distributed model

There appears to be a gap between randomized and deterministic running times

We know randomized $\Omega(1)$-approximation in $O(1)$ rounds but only $O(\log^* n)$ deterministic rounds

What is the deterministic complexity of $\Omega(1)$-approximation for MaxCut ?

Our negative results:

Theorem

Let $d > 0$ be an integer.

- If d is even, any deterministic algorithm in the LOCAL model that guarantees a constant factor approximation for MaxCut on the class of bipartite d-regular graphs n-vertex graphs runs in $\Omega(\log^* n)$ rounds.
- If d is odd, then for any $\epsilon > 0$, any $\left(\frac{1}{d} + \epsilon\right)$-approximation requires $\Omega(\log^* n)$ rounds in the same setting.

The Maximum (directed) Cut in the distributed model

There appears to be a gap between randomized and deterministic running times

We know randomized $\Omega(1)$-approximation in $O(1)$ rounds but only $O(\log^* n)$ deterministic rounds

What is the deterministic complexity of $\Omega(1)$-approximation for MaxCut ?

Our negative results:

Theorem

Let $d > 0$ be an integer.

- If d is even, any deterministic algorithm in the LOCAL model that guarantees a constant factor approximation for MaxCut on the class of bipartite d-regular graphs n-vertex graphs runs in $\Omega(\log^* n)$ rounds.
- If d is odd, then for any $\epsilon > 0$, any $\left(\frac{1}{d} + \epsilon\right)$-approximation requires $\Omega(\log^* n)$ rounds in the same setting.

For the oriented cut problem (MaxDiCut), the results are the same except that $\frac{1}{d}$ is replaced by $\frac{2}{d}$ in the odd case.

The Maximum (directed) Cut in the distributed model

Our positive results:

The Maximum (directed) Cut in the distributed model

Our positive results:

Theorem

Let $d > 0$ be an odd integer.

- For the MaxCut, there exists a deterministic $\frac{1}{d}$-approximation on d-regular graphs running in 1 round in the CONGEST model.
- For the MaxDiCut, there exists a deterministic $\frac{2}{d+1/d}$-approximation on d-regular graphs running in 0 round in the CONGEST model.
- For the MaxDiCut, there exists $\alpha > 0$ such that there is a deterministic $\left(\frac{2}{d+1/d} + \alpha\right)$-approximation on d-regular graphs running in 2 rounds in the CONGEST model.

The Maximum (directed) Cut in the distributed model

Our positive results:

Theorem

Let $d > 0$ be an odd integer.

- For the MaxCut, there exists a deterministic $\frac{1}{d}$-approximation on d-regular graphs running in 1 round in the CONGEST model.
- For the MaxDiCut, there exists a deterministic $\frac{2}{d+1/d}$-approximation on d-regular graphs running in 0 round in the CONGEST model.
- For the MaxDiCut, there exists $\alpha > 0$ such that there is a deterministic $\left(\frac{2}{d+1/d} + \alpha\right)$-approximation on d-regular graphs running in 2 rounds in the CONGEST model.

These algorithms are very simple and even do not require any communication if the orientation of edges is given (PO model for instance).

Proof of the lower bound

Theorem (Ramsey (1930))

Consider a hypergraph $G = (V, E)$ where edges are sets of r vertices. Fix $m > 0$ then for any mapping $f : E \mapsto \{0, 1\}$ there exists a n big enough such that there is a monochromatic induced subgraph of size m.

Proof of the lower bound

Theorem (Ramsey (1930))

Consider a hypergraph $G = (V, E)$ where edges are sets of r vertices. Fix $m > 0$ then for any mapping $f : E \mapsto \{0, 1\}$ there exists a n big enough such that there is a monochromatic induced subgraph of size m.

A deterministic algorithm is a mapping from every possible neighborhood to $\{0, 1\}$

Proof of the lower bound

Theorem (Ramsey (1930))

Consider a hypergraph $G = (V, E)$ where edges are sets of r vertices. Fix $m > 0$ then for any mapping $f : E \mapsto \{0, 1\}$ there exists a n big enough such that there is a monochromatic induced subgraph of size m.

A deterministic algorithm is a mapping from every possible neighborhood to $\{0, 1\}$

Here a neighborhood is given by the edge set, the set of IDs and the permutation of IDs on vertices

Proof of the lower bound

Find a graph where every vertex has the same local view (same edge set) and the permutation of IDs is easy to fix

Proof of the lower bound

Find a graph where every vertex has the same local view (same edge set) and the permutation of IDs is easy to fix

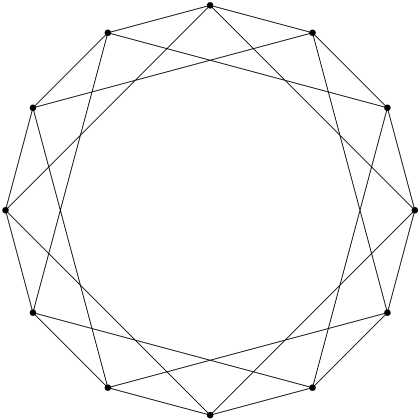


Figure: d even

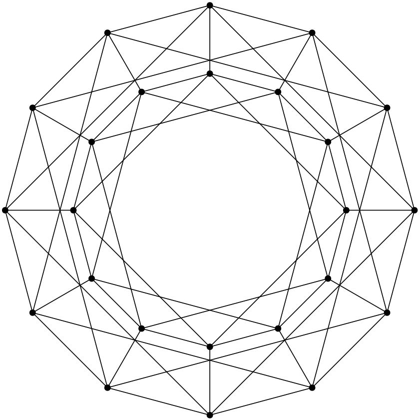


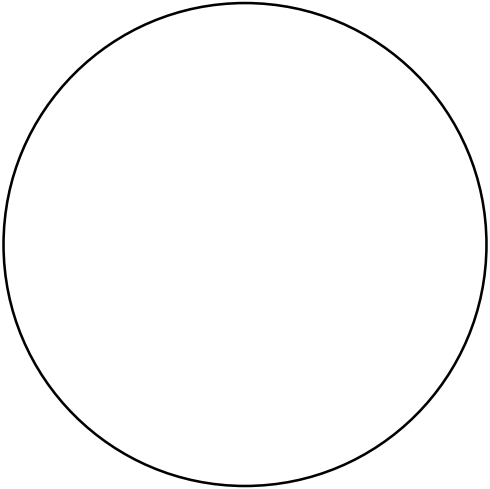
Figure: d odd

Proof of the lower bound

Iteratively apply Ramsey's theorem to partition the cycle into monochromatic slices

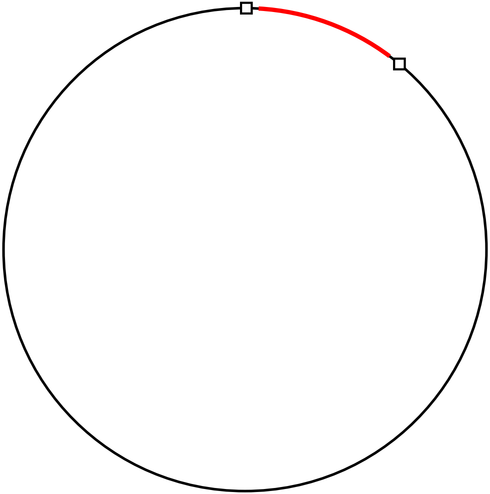
Proof of the lower bound

Iteratively apply Ramsey's theorem to partition the cycle into monochromatic slices



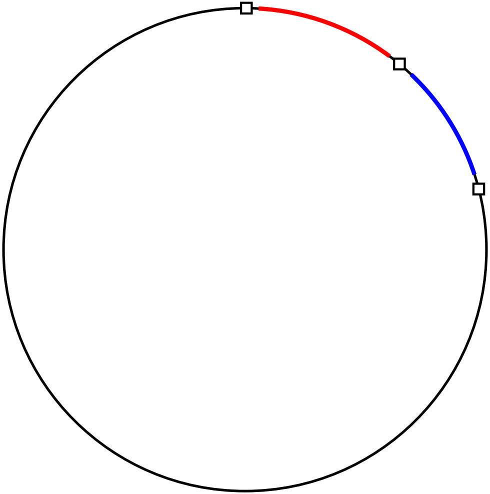
Proof of the lower bound

Iteratively apply Ramsey's theorem to partition the cycle into monochromatic slices



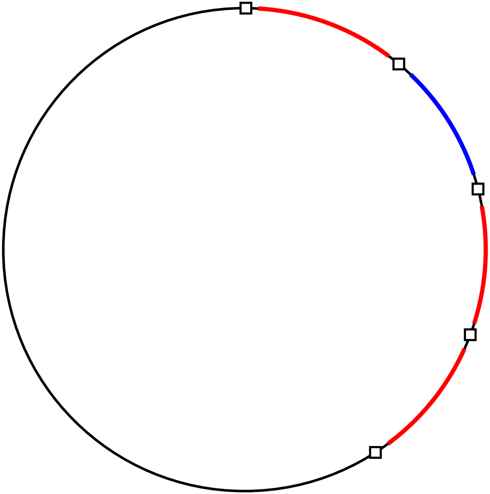
Proof of the lower bound

Iteratively apply Ramsey's theorem to partition the cycle into monochromatic slices



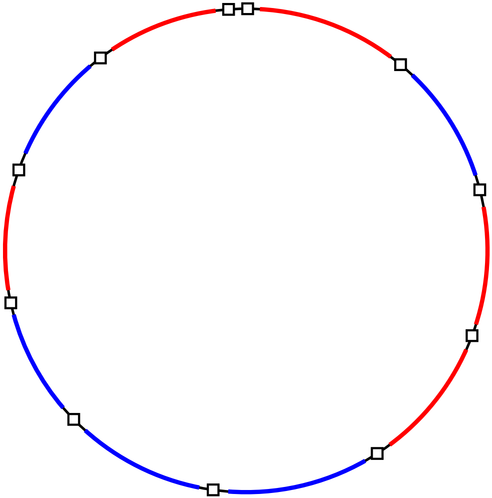
Proof of the lower bound

Iteratively apply Ramsey's theorem to partition the cycle into monochromatic slices



Proof of the lower bound

Iteratively apply Ramsey's theorem to partition the cycle into monochromatic slices



Proof of the lower bound

Iteratively apply Ramsey's theorem to partition the cycle into monochromatic slices

Theorem

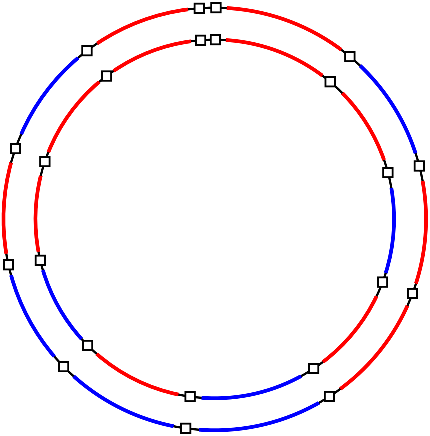
Let $d > 0$ be an integer.

- If d is even, any deterministic algorithm in the LOCAL model that guarantees a constant factor approximation for MaxCut on the class of bipartite d-regular graphs n-vertex graphs runs in $\Omega(\log^* n)$ rounds.
- If d is odd, then for any $\epsilon > 0$, any $\left(\frac{1}{d} + \epsilon\right)$-approximation requires $\Omega(\log^* n)$ rounds in the same setting.

Proof of the lower bound

Theorem

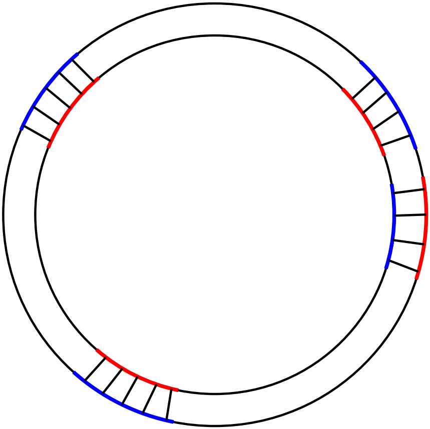
If d is an odd integer, then for any $\epsilon > 0$, any $\left(\frac{1}{d} + \epsilon\right)$-approximation requires $\Omega(\log^* n)$ rounds in the same setting.



Proof of the lower bound

Theorem

If d is an odd integer, then for any $\epsilon > 0$, any $\left(\frac{1}{d} + \epsilon\right)$-approximation requires $\Omega(\log^* n)$ rounds in the same setting.



Proof of the positive results

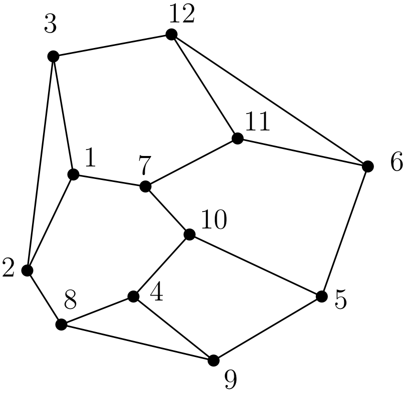
Theorem

Let $d > 0$ be an odd integer.

- For the MaxCut, there exists a deterministic $\frac{1}{d}$-approximation on d-regular graphs running in 1 round in the CONGEST model.
- For the MaxDiCut, there exists a deterministic $\frac{2}{d+1/d}$-approximation on d-regular graphs running in 0 round in the CONGEST model.
- For the MaxDiCut, there exists $\alpha > 0$ such that there is a deterministic $\left(\frac{2}{d+1/d} + \alpha\right)$-approximation on d-regular graphs running in 2 rounds in the CONGEST model.

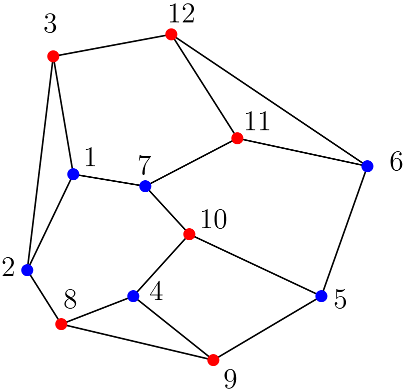
Proof of the positive results

Unoriented case with d odd: every vertex v collects the list $L(v)$ of IDs of its neighbors. Let $m(v)$ be the median value of $L(v)$



Proof of the positive results

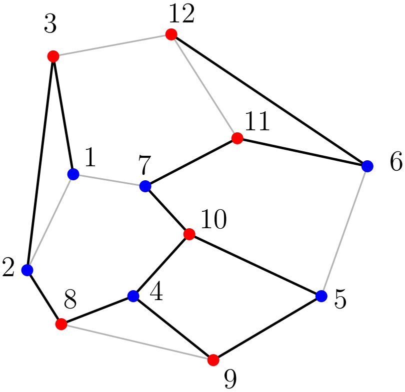
Unoriented case with d odd: every vertex v collects the list $L(v)$ of IDs of its neighbors. Let $m(v)$ be the median value of $L(v)$



v chooses the left side if and only if $ID(v) > m(v)$

Proof of the positive results

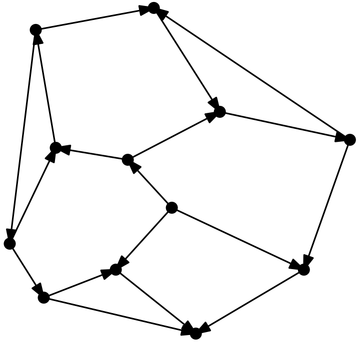
Unoriented case with d odd: every vertex v collects the list $L(v)$ of IDs of its neighbors. Let $m(v)$ be the median value of $L(v)$



v chooses the left side if and only if $ID(v) > m(v) \implies \frac{1}{d}$-approximation

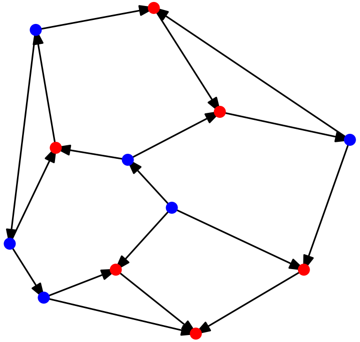
Proof of the positive results

Oriented case with d odd: a vertex chooses the right side if and only if it has more ingoing than outgoing edges



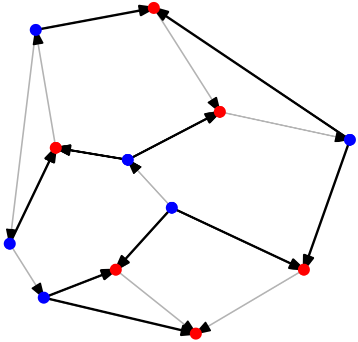
Proof of the positive results

Oriented case with d odd: a vertex chooses the right side if and only if it has more ingoing than outgoing edges



Proof of the positive results

Oriented case with d odd: a vertex chooses the right side if and only if it has more ingoing than outgoing edges



$\Longrightarrow$ $\frac{2}{d+1/d}$-approximation (the ratio is tight).

Proof of the positive results

How to match the $\frac{2}{d}$ upper bound ?

Proof of the positive results

How to match the $\frac{2}{d}$ upper bound ?

In a given cut, a vertex v is said to be unstable if and only if all its neighbors are on the same side of the cut as v

Proof of the positive results

How to match the $\frac{2}{d}$ upper bound ?

In a given cut, a vertex v is said to be unstable if and only if all its neighbors are on the same side of the cut as v

For 2 rounds, every unstable vertex changes side. $\implies \left(\frac{2}{d+1/d} + \alpha\right)$ approximation ratio

Conclusion

Is the upper bound on approximation ratio $\frac{2}{d}$ matched ?

Conclusion

Is the upper bound on approximation ratio $\frac{2}{d}$ matched ?

MaxCut is not a locally checkable problem which makes it possible to get this gap between randomized and deterministic algorithms

Conclusion

Is the upper bound on approximation ratio $\frac{2}{d}$ matched ?

MaxCut is not a locally checkable problem which makes it possible to get this gap between randomized and deterministic algorithms

A related problem that is also locally checkable ?

Conclusion

Is the upper bound on approximation ratio $\frac{2}{d}$ matched ?

MaxCut is not a locally checkable problem which makes it possible to get this gap between randomized and deterministic algorithms

A related problem that is also locally checkable ?

Yes ! Maximal cut

Conclusion

Is the upper bound on approximation ratio $\frac{2}{d}$ matched ?

MaxCut is not a locally checkable problem which makes it possible to get this gap between randomized and deterministic algorithms

A related problem that is also locally checkable ?

Yes ! Maximal cut

Maximal cut behaves very differently: $\Omega(\log n)$ deterministic rounds and $\Omega(\log \log n)$ randomized rounds (A. Balliu, J. Hirvonen, C. Lenzen, D. Olivetti, J. Suomela (SIROCCO 2019))