Expanders via Local Edge Flips

Zeyuan Allen-Zhu
Princeton University

Aditya Bhaskara
Google

Silvio Lattanzi
Google

Vahab Mirrokni
Google

Lorenzo Orecchia
Boston Univesity

Big Data & Sublinear Algorithms Workshop, DIMACS

Outline

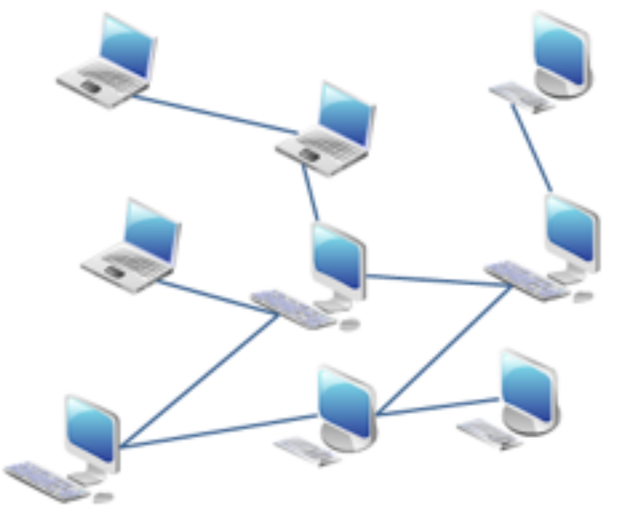
- How can we construct an expander locally?
  - Problem motivation and related works
- A simple distributed protocol
  - The switch and the flip protocols
- A new analysis for the two protocols
  - Obstacles in the analysis and new approach for the problem
- Conclusions and future directions
  - Open problems

How can we construct an expander locally?

Why is it interesting?

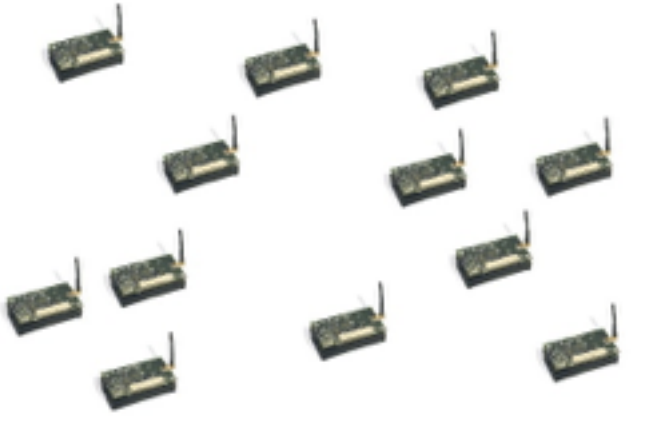
Distributed system

- P2P networks
- Sensor networks
- Asynchronous system



Benefits

- Efficient
- Robust



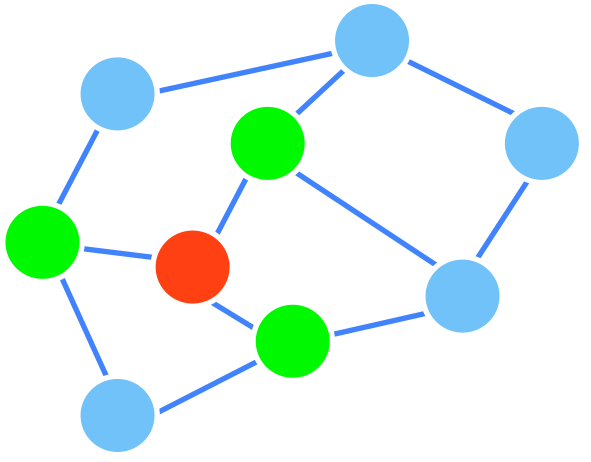
New challenges

- Important to construct quickly good network structure
- Only local communication

Local graph algorithms

Local algorithms

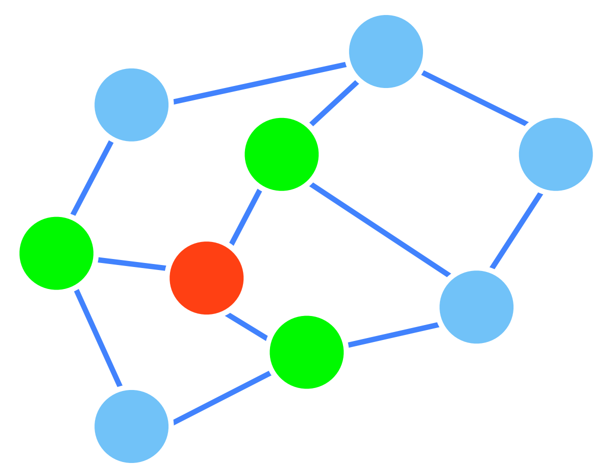
Algorithms based on *local* message passing among nodes



Local graph algorithms

Local algorithms

Algorithms based on *local* message passing among nodes



Advantages

Applicable to large scale graphs

Fast, easy to implement in parallel (MapReduce, Hadoop, Pregel...)

Problem

Starting from any connected graph is it possible to construct an expander locally?

Previous work

SKIP+: A Self-Stabilizing Skip Graph.

R. Jacob, A. W. Richa, C. Scheideler, S. Schmid and H. Täubig.

J. ACM 61(6): 36:1-36:26 (2014)

In the Local model it is possible to build an expander locally in $O(\log^2 n)$

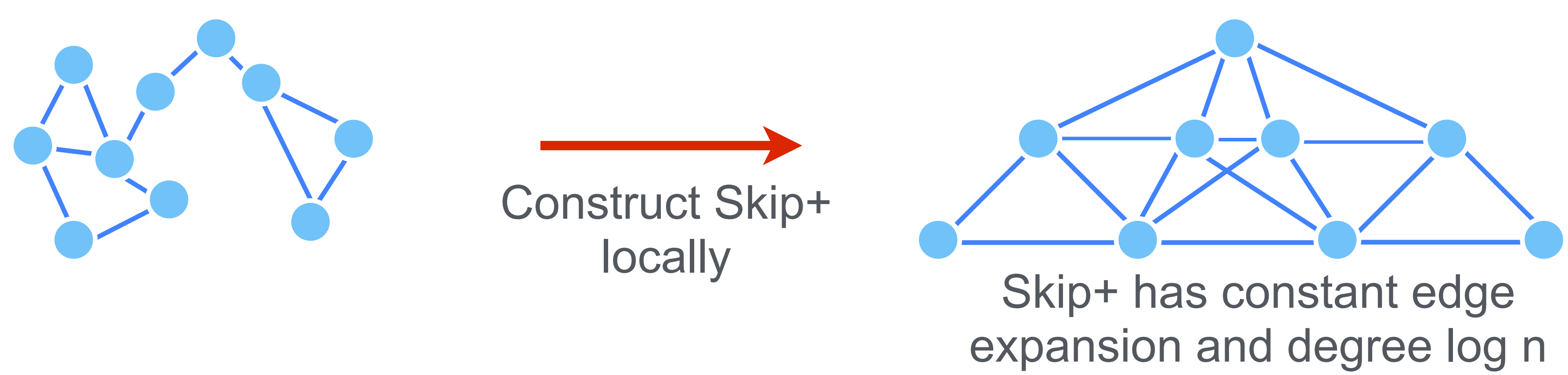
Previous work

SKIP+: A Self-Stabilizing Skip Graph.

R. Jacob, A. W. Richa, C. Scheideler, S. Schmid and H. Täubig.

J. ACM 61(6): 36:1-36:26 (2014)

In the Local model it is possible to build an expander locally in $O(\log^2 n)$



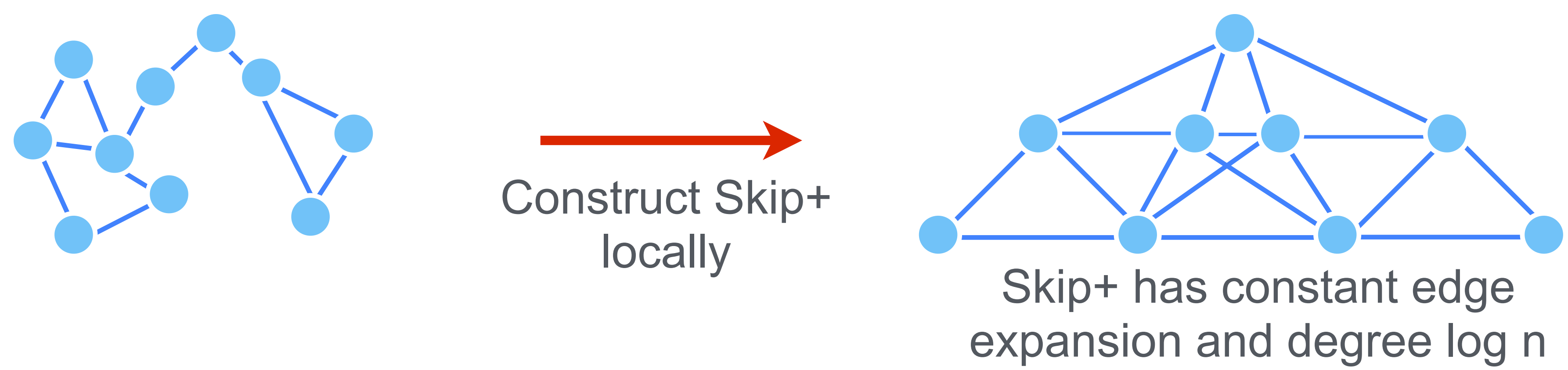
Previous work

SKIP+: A Self-Stabilizing Skip Graph.

R. Jacob, A. W. Richa, C. Scheideler, S. Schmid and H. Täubig.

J. ACM 61(6): 36:1-36:26 (2014)

In the Local model it is possible to build an expander locally in $O(\log^2 n)$



Limitations:

- Using this technique it is not possible to obtain an algebraic expander
- In any round nodes can exchange arbitrary large messages
- Memory needed by a single node in any round is not bounded
- Synchronous model, complex algorithm

Problem

Starting from any connected graph is it possible to define a simple rule to construct an expander locally?

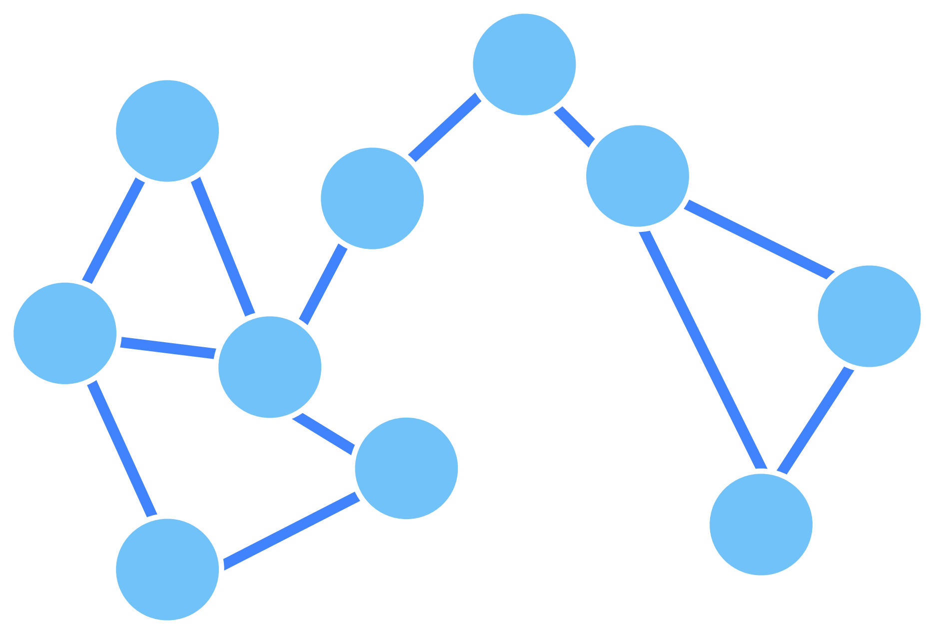
A simple distributed protocol

Switch protocol

[McKay, *Congressus Numerantium* 1981]

A simple protocol:

Pick two edges at random and invert their endpoints

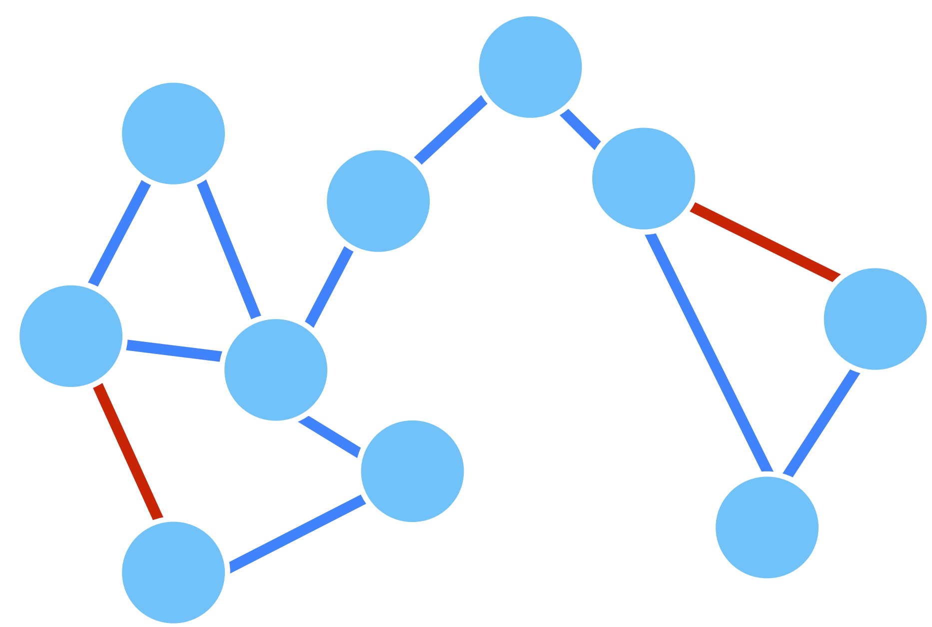


Switch protocol

[McKay, *Congressus Numerantium* 1981]

A simple protocol:

Pick two edges at random and invert their endpoints

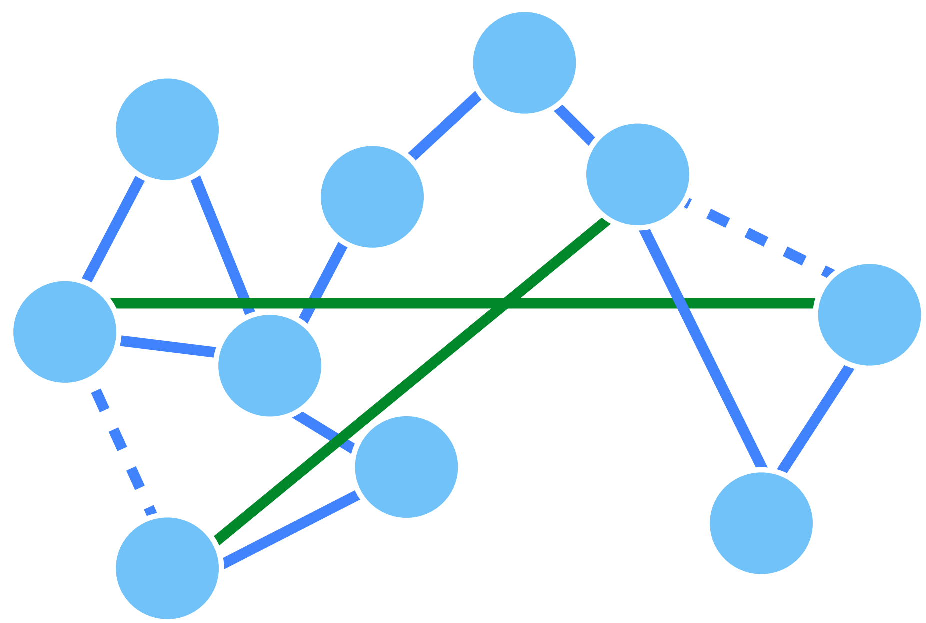


Switch protocol

[McKay, *Congressus Numerantium* 1981]

A simple protocol:

Pick two edges at random and invert their endpoints

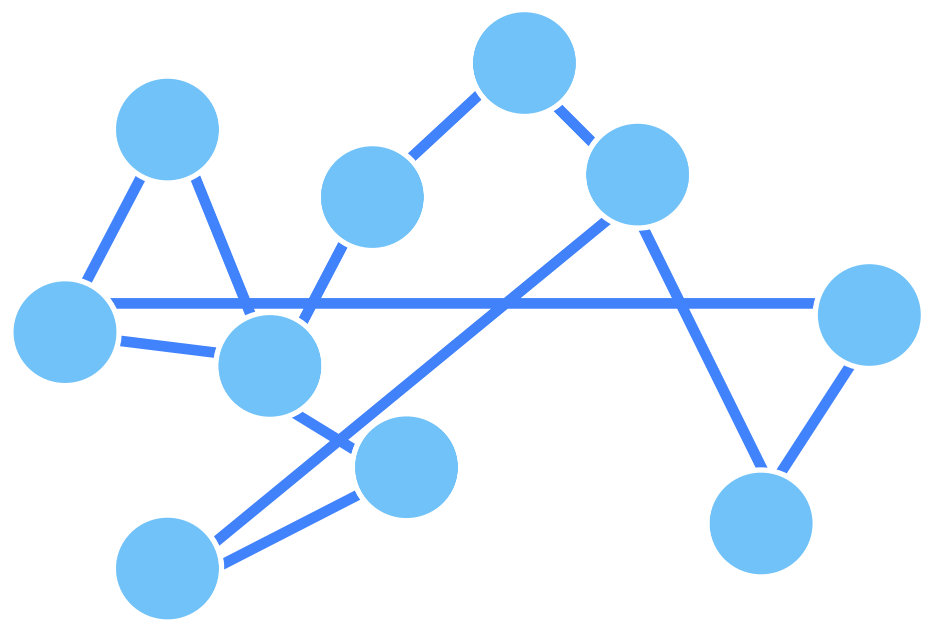


Switch protocol

[McKay, *Congressus Numerantium* 1981]

A simple protocol:

- Pick two edges at random and invert their endpoints
- Creation of parallel edges/self-loops is allowed



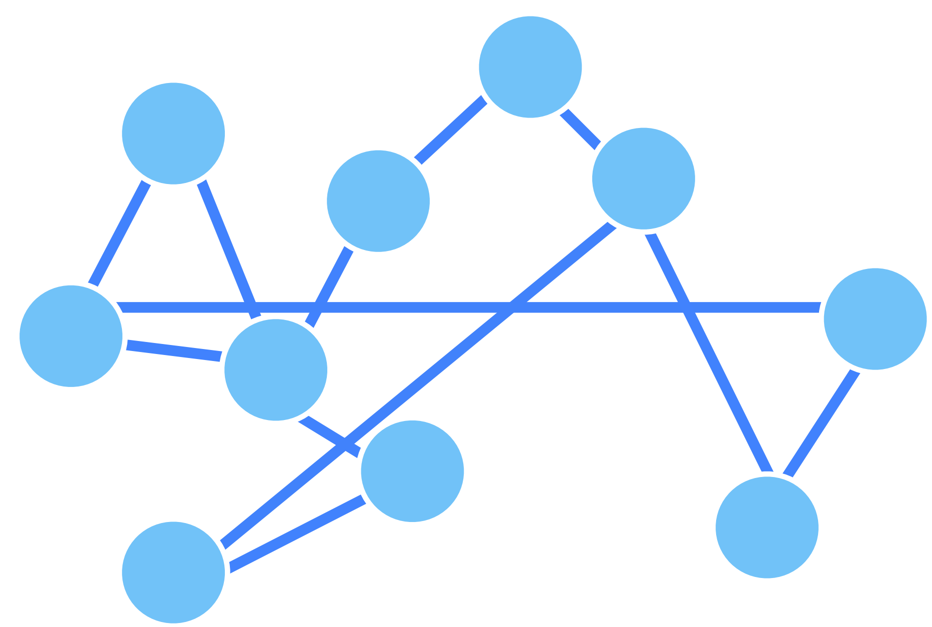
Switch protocol

[McKay, *Congressus Numerantium* 1981]

A simple protocol:

Pick two edges at random and invert their endpoints

Creation of parallel edges/self-loops is allowed



Limitation

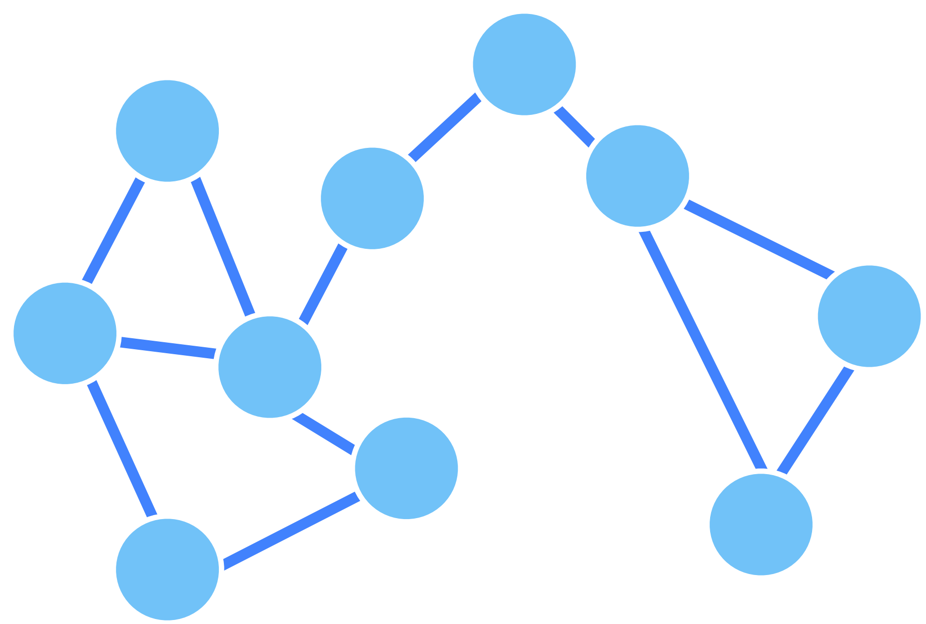
It is not local

It may disconnect the graph

Flip protocol

[Mahlmann and Schindelhauer, *SPAA* 2005]

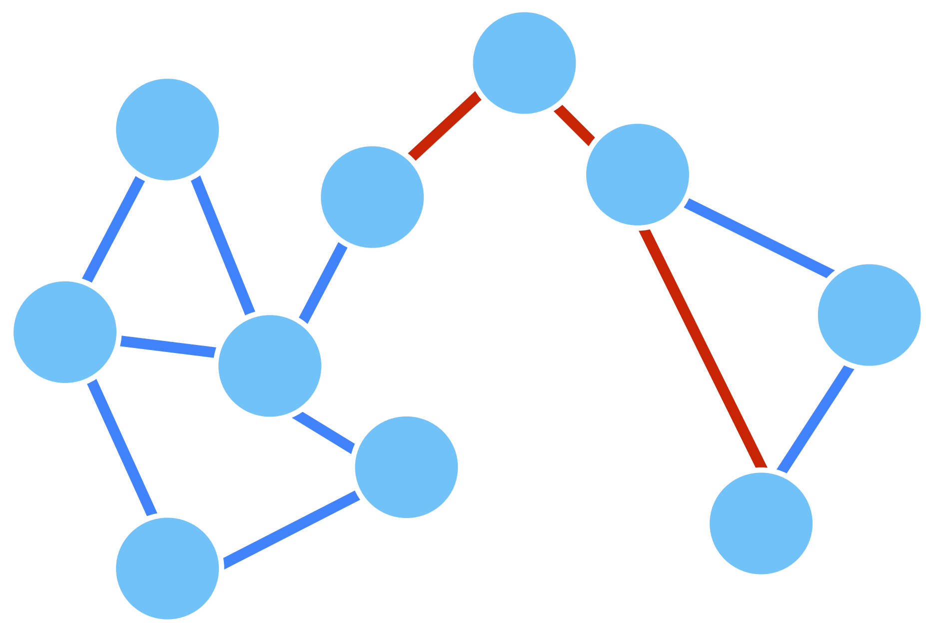
Pick a random length 3 path and invert its endpoints



Flip protocol

[Mahlmann and Schindelhauer, *SPAA* 2005]

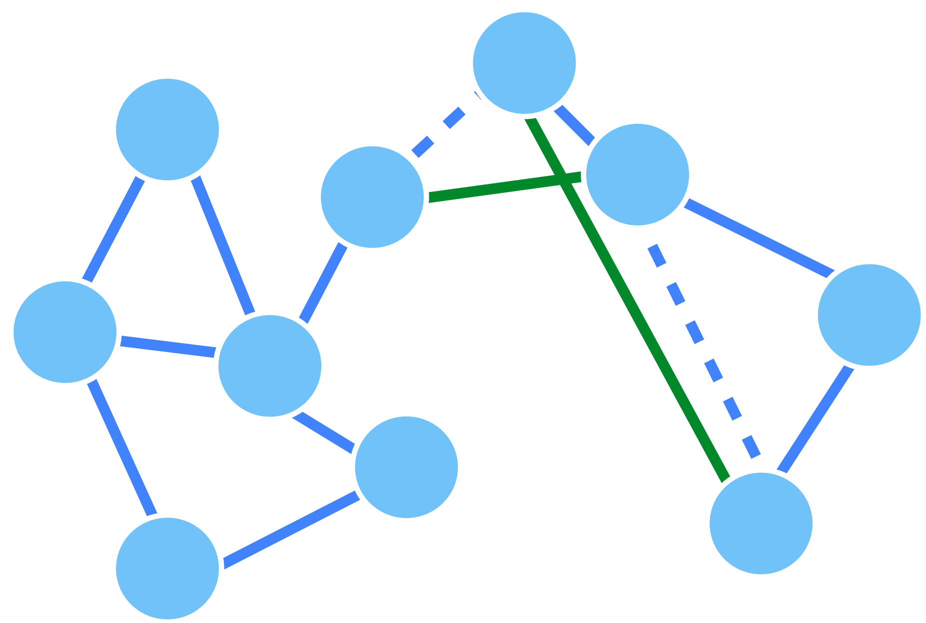
Pick a random length 3 path and invert its endpoints



Flip protocol

[Mahlmann and Schindelhauer, *SPAA* 2005]

Pick a random length 3 path and invert its endpoints

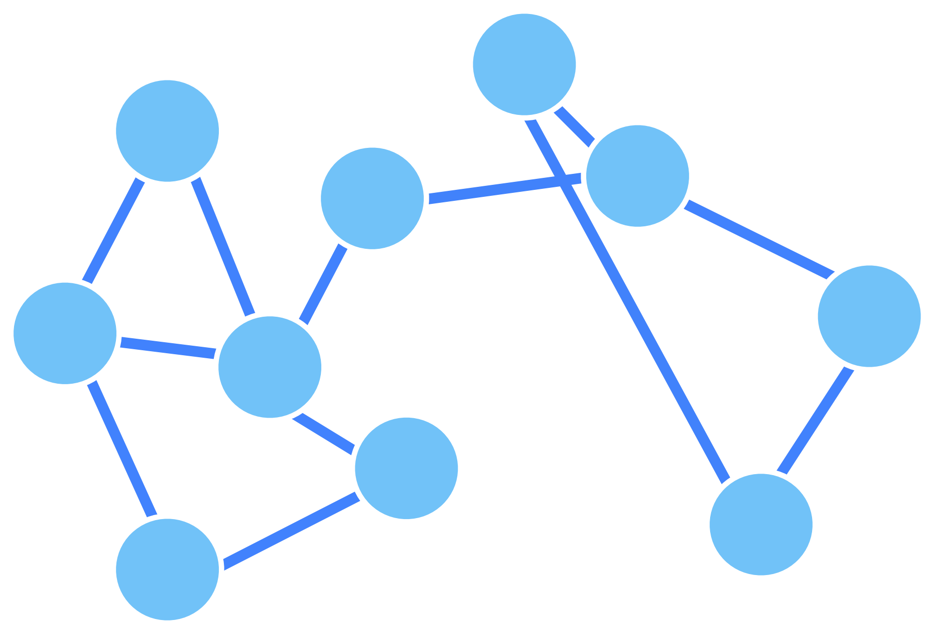


Flip protocol

[Mahlmann and Schindelhauer, *SPAA* 2005]

Pick a random length 3 path and invert its endpoints

Creation of parallel edges/self-loops is allowed

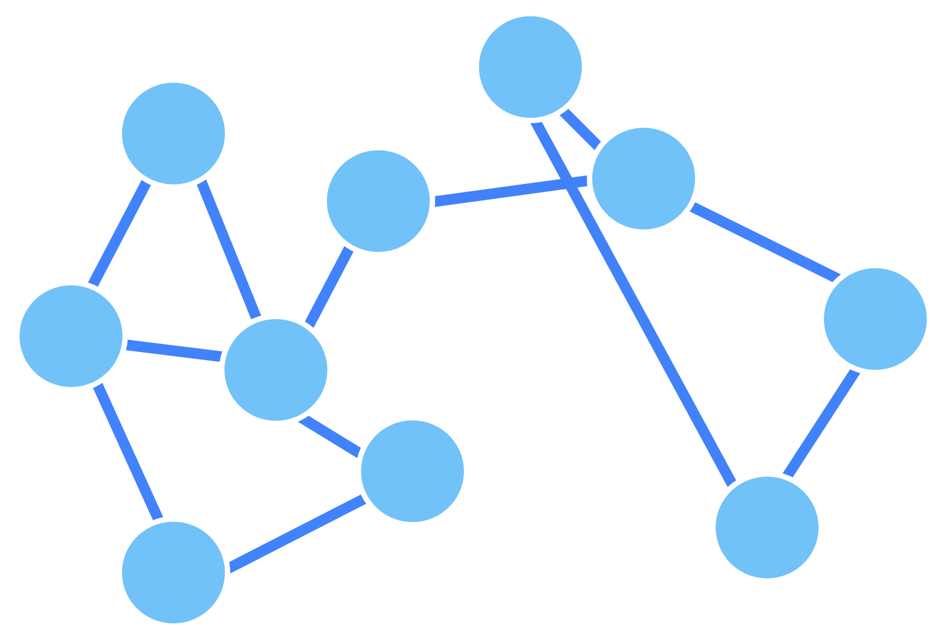


Flip protocol

[Mahlmann and Schindelhauer, *SPAA* 2005]

Pick a random length 3 path and invert its endpoints

Creation of parallel edges/self-loops is allowed



Experimentally it seems to be really fast

What is known about them?

[Cooper, Dyer and Greenhill, *SODA* 2005]

For d-regular graph the switch protocol converges to the configuration model in $\tilde{O}\left(n^8 d^{15}\right)$ steps.

[Greenhill, *SODA* 2015]

For non regular graph with max degree in $O\left(\sqrt{m}\right)$ the switch protocol converges to the configuration model in $\tilde{O}\left(m^{10} d_{max}^{14}\right)$ steps.

What is known about them?

[Cooper, Dyer and Greenhill, *SODA* 2005]

For d-regular graph the switch protocol converges to the configuration model in $\tilde{O}\left(n^8 d^{15}\right)$ steps.

[Greenhill, *SODA* 2015]

For non regular graph with max degree in $O\left(\sqrt{m}\right)$ the switch protocol converges to the configuration model in $\tilde{O}\left(m^{10} d_{max}^{14}\right)$ steps.

[Mahlmann and Schindelhauer, *SPAA* 2005]

For d-regular graph the flip protocol converges to the configuration model.

[Feder, Guetz, Mihail, and Saberi, *FOCS* 2006]

For d-regular graph the flip protocol converges to the configuration model in $\tilde{O}\left(d^{34} n^{36}\right)$ steps.

[Cooper and Dyer, *PODC* 2009]

For d-regular graph the flip protocol converges to the configuration model in $\tilde{O}\left(d^{23} n^{17}\right)$ steps.

How do they perform in practice?

[Mahlmann and Schindelhauer, *SPAA* 2005]

Experimentally switch and flips protocol transform any graph in an expander very quickly.

Conjectures:

Switch converges on d-regular graph in $O(nd)$ steps.

Flip converges on d-regular graph in $O(nd \log n)$ steps.

A new analysis for the two protocols

Results

Starting from any d-regular graph, with $d \in \Omega(\log n)$,

the switch protocol transforms the graph in an algebraic expander in $O\left(nd\right)$ steps.

the flip protocol transforms the graph in an algebraic expander in $O\left(n^2 d^2 \sqrt{\log n}\right)$ steps.

Results

Starting from any d-regular graph, with $d \in \Omega(\log n)$,

the switch protocol transforms the graph in an algebraic expander in $O\left(nd\right)$ steps.

the flip protocol transforms the graph in an algebraic expander in $O\left(n^2 d^2 \sqrt{\log n}\right)$ steps.

Obstacles

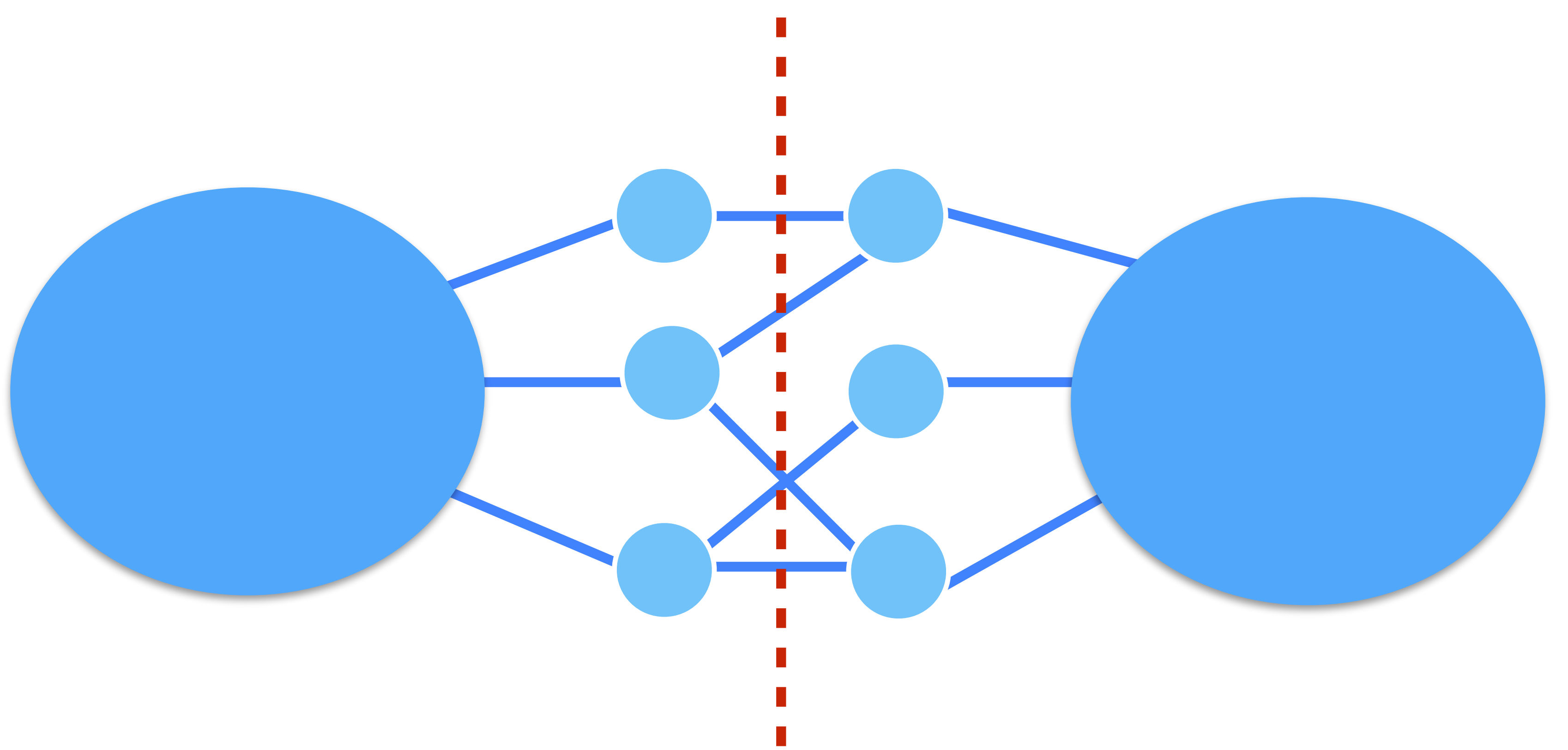
Dependencies.

Small cuts may first become smaller and only later increase.

Obstacles

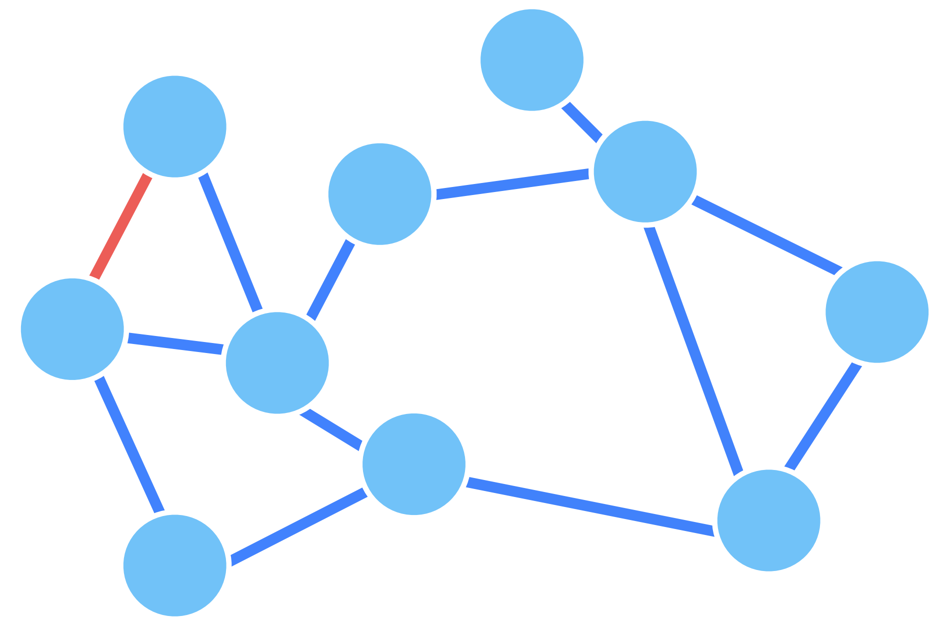
Dependencies.

Small cuts may first become smaller and only later increase.



Flip definition

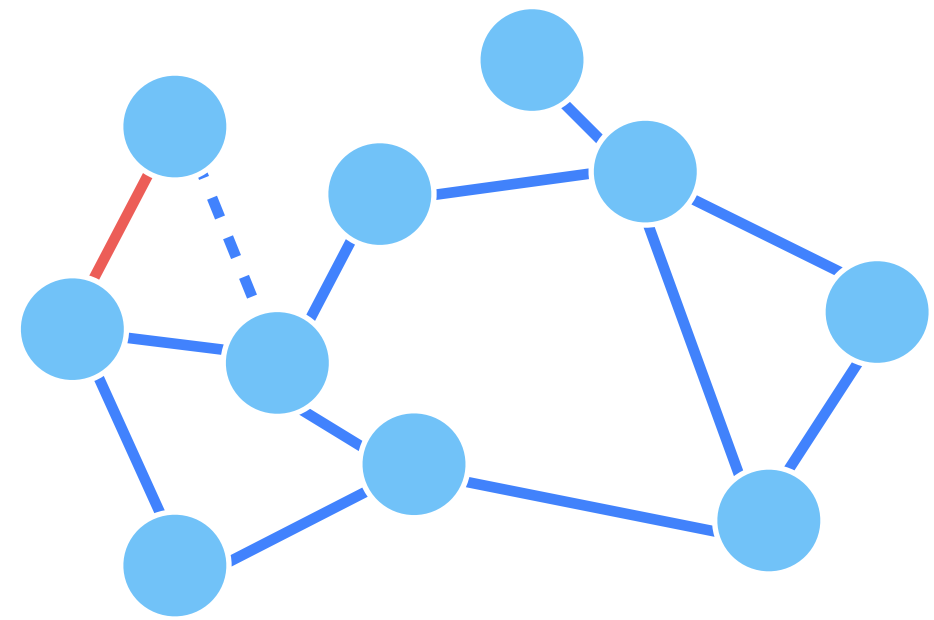
Pick a random edge.



Flip definition

Pick a random edge.

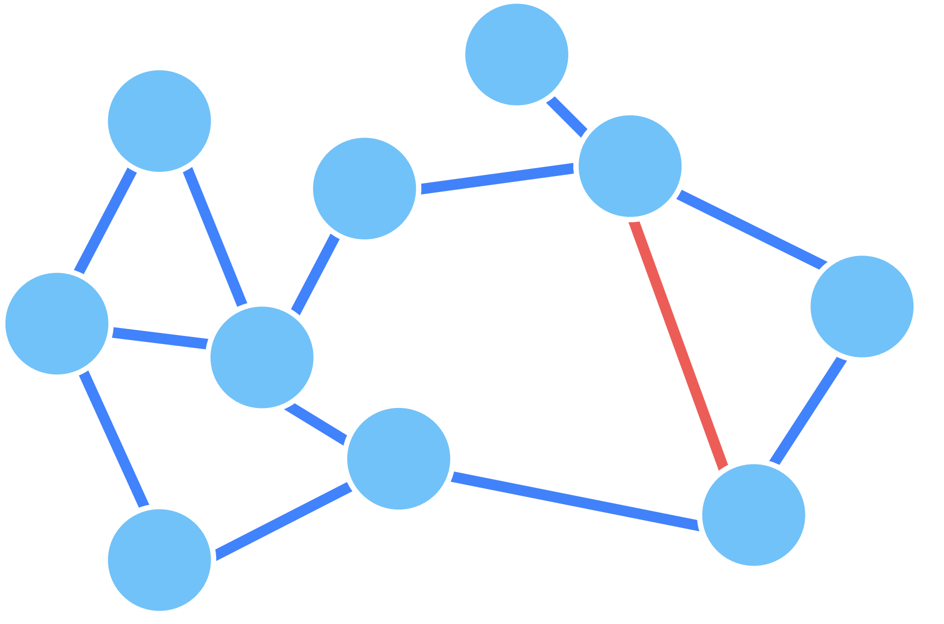
One of the endpoints picks a neighbor at random(if in common, abort).



Flip definition

Pick a random edge.

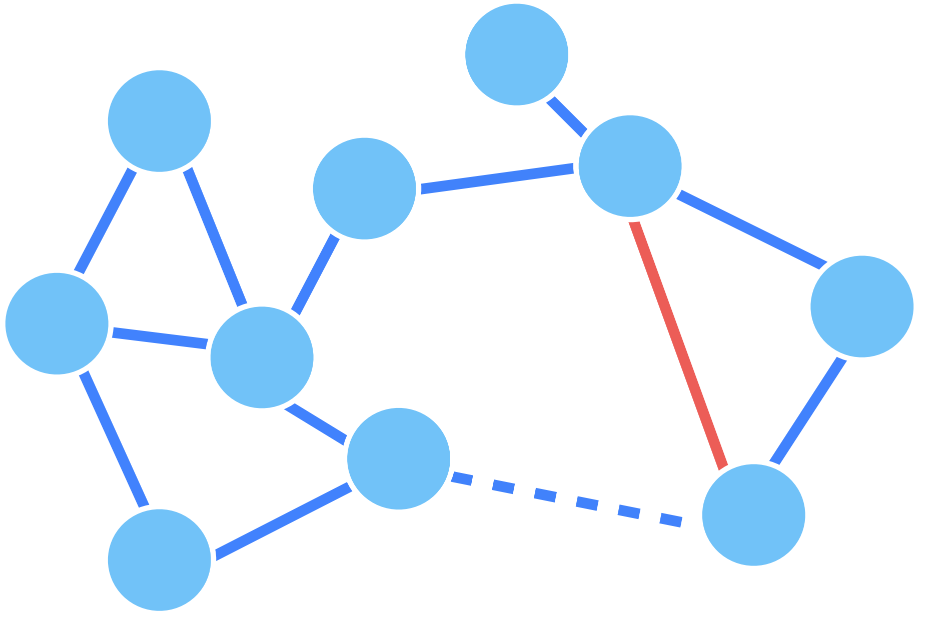
One of the endpoints picks a neighbor at random(if in common, abort).



Flip definition

Pick a random edge.

One of the endpoints picks a neighbor at random(if in common, abort).

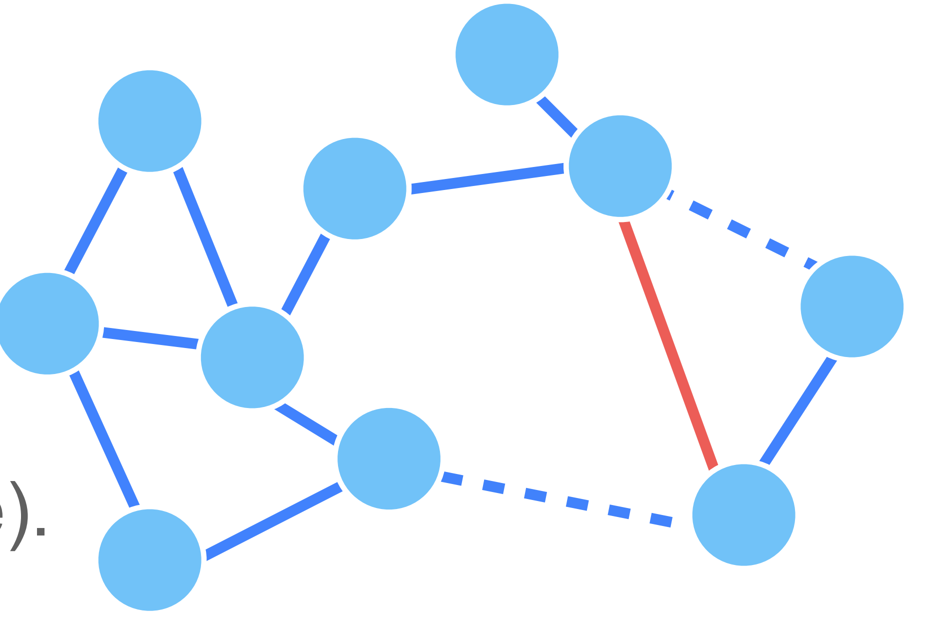


Flip definition

Pick a random edge.

One of the endpoints picks a neighbor at random(if in common, abort).

The other endpoint picks a random neighbor(if in common, picks a new one).

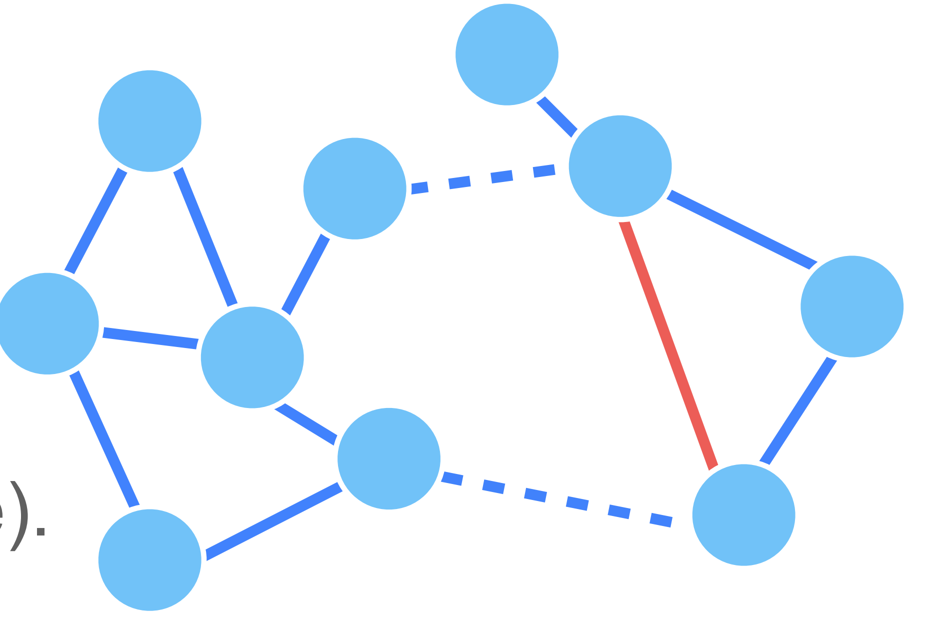


Flip definition

Pick a random edge.

One of the endpoints picks a neighbor at random(if in common, abort).

The other endpoint picks a random neighbor(if in common, picks a new one).



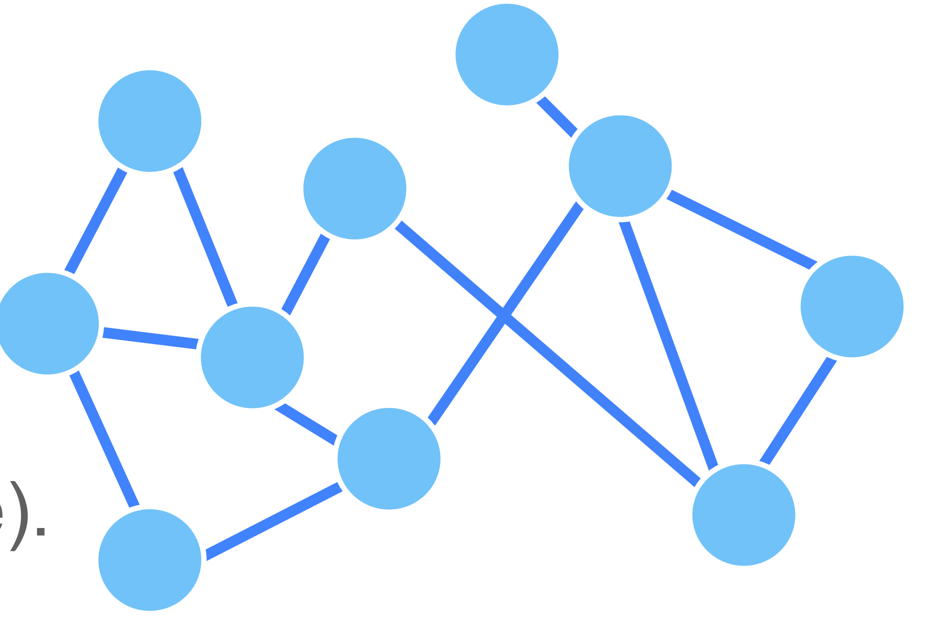
Flip definition

Pick a random edge.

One of the endpoints picks a neighbor at random(if in common, abort).

The other endpoint picks a random neighbor(if in common, picks a new one).

Perform swap.



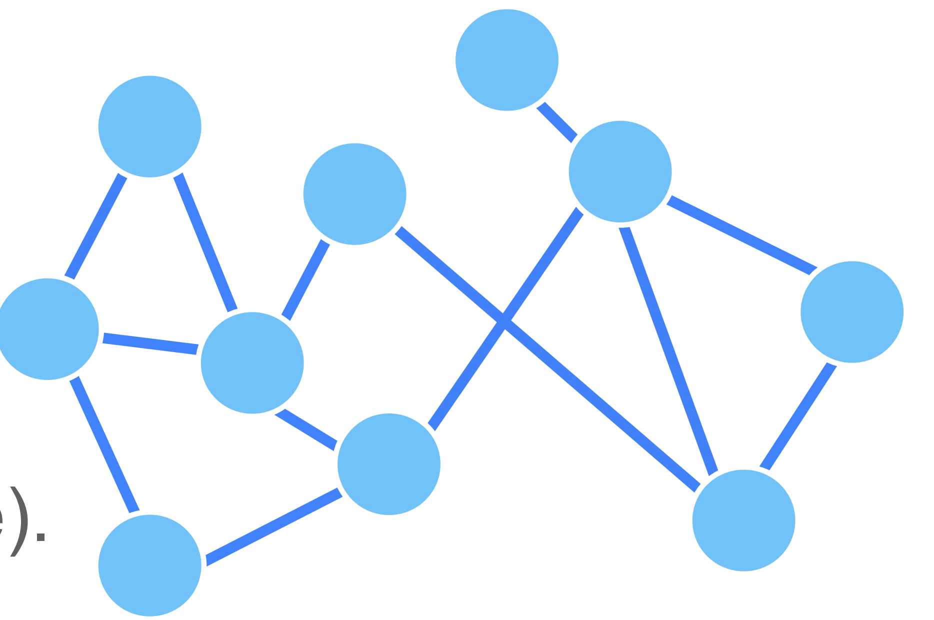
Expected change of laplacian

Pick a random edge.

One of the endpoints picks a neighbor at random(if in common, abort).

The other endpoint picks a random neighbor(if in common, picks a new one).

Perform swap.



Let $\Delta^{(t)} = L\left(G^{(t+1)}\right) - L\left(G^{(t)}\right)$

$$E\left[\Delta^{(t)} | G^{(t)}\right] = \frac{4}{d^2 n}\left((d+1)L^{(t)} - \left(L^{(t)}\right)^2\right)$$

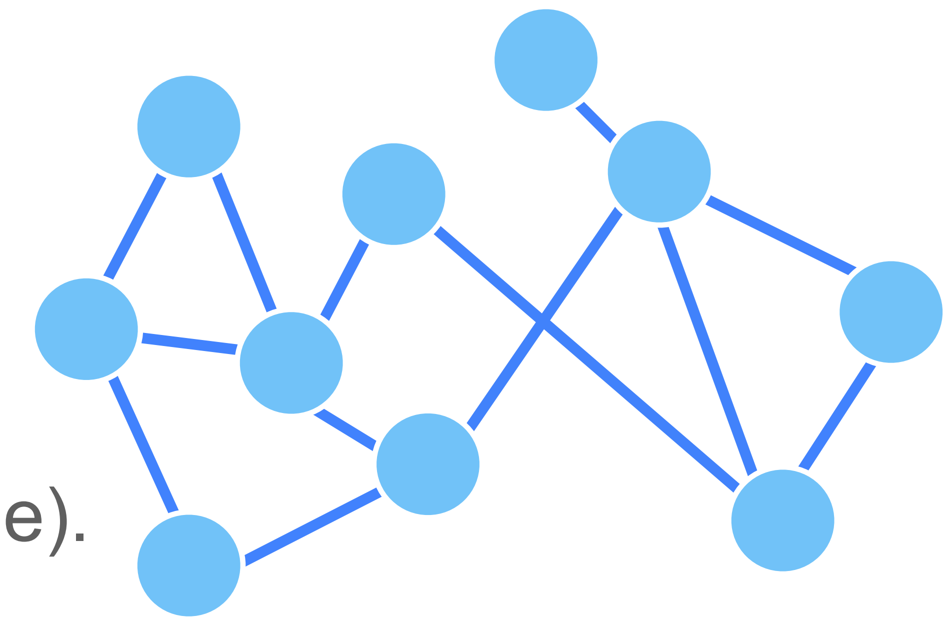
Expected change of laplacian

Pick a random edge.

One of the endpoints picks a neighbor at random(if in common, abort).

The other endpoint picks a random neighbor(if in common, picks a new one).

Perform swap.



Let $\Delta^{(t)} = L\left(G^{(t+1)}\right) - L\left(G^{(t)}\right)$

$$E\left[\Delta^{(t)} | G^{(t)}\right] = \frac{4}{d^2 n}\left((d+1)L^{(t)} - \left(L^{(t)}\right)^2\right)$$

Nice term. $\left(G^{(t)}\right)^2$ has better expansion.

Potential

Unfortunately we cannot argue directly on the expectation of the matrix after t step.

$$E\left[\Delta^{(t)}|G^{(t)}\right] = \frac{4}{d^2 n}\left((d+1)L^{(t)} - \left(L^{(t)}\right)^2\right)$$

Potential

Unfortunately we cannot argue directly on the expectation of the matrix after t step.

$$E\left[\Delta^{(t)} | G^{(t)}\right] = \frac{4}{d^2 n}\left((d+1)L^{(t)} - \left(L^{(t)}\right)^2\right)$$

We use a classic potential used for matrix concentration:

$$\Phi^{(t)} = \hat{tr}\left(e^{-\frac{20\log n}{d}L^{(t)}}\right)$$

where $\hat{tr}\left(e^{A}\right) = e^{\lambda_1} + e^{\lambda_2} + \ldots$

Potential

Unfortunately we cannot argue directly on the expectation of the matrix after t step.

$$E\left[\Delta^{(t)} | G^{(t)}\right] = \frac{4}{d^2 n}\left((d+1)L^{(t)} - \left(L^{(t)}\right)^2\right)$$

We use a classic potential used for matrix concentration:

$$\Phi^{(t)} = \hat{tr}\left(e^{-\frac{20 \log n}{d} L^{(t)}}\right)$$

where $\hat{tr}\left(e^{A}\right) = e^{\lambda_1} + e^{\lambda_2} + \ldots$

Note that in order to have $\Phi^{(t)}$ very small all the eigenvalues need to be large.

Potential

We want to show that the potential decreases

$$\Phi^{(t+1)} = \hat{tr}\left(e^{-\frac{20\log n}{d}(L^{(t)}+\Delta^{(t)})}\right)$$

Potential

We want to show that the potential decreases

$$\Phi^{(t+1)} = \hat{tr}\left(e^{-\frac{20\log n}{d}(L^{(t)}+\Delta^{(t)})}\right)$$

by Golden-Thompson inequality $$= \hat{tr}\left(e^{-\frac{20\log n}{d}L^{(t)}}e^{-\frac{20\log n}{d}\Delta^{(t)}}\right)$$

Potential

We want to show that the potential decreases

$$\Phi^{(t+1)} = \hat{tr}\left(e^{-\frac{20\log n}{d}(L^{(t)}+\Delta^{(t)})}\right)$$

by Golden-Thompson inequality
$$= \hat{tr}\left(e^{-\frac{20\log n}{d}L^{(t)}}e^{-\frac{20\log n}{d}\Delta^{(t)}}\right)$$

by $e^{-A} = I - A + A^2$
$$= \hat{tr}\left(e^{-\frac{20\log n}{d}L^{(t)}}\left(I - \frac{20\log n}{d}\Delta^{(t)} + \left(\frac{20\log n}{d}\Delta^{(t)}\right)^2\right)\right)$$

Potential

We want to show that the potential decreases

$$\Phi^{(t+1)} = \hat{tr}\left(e^{-\frac{20\log n}{d}(L^{(t)}+\Delta^{(t)})}\right)$$

by Golden-Thompson inequality
$$= \hat{tr}\left(e^{-\frac{20\log n}{d}L^{(t)}}e^{-\frac{20\log n}{d}\Delta^{(t)}}\right)$$

by $e^{-A} = I - A + A^2$
$$= \hat{tr}\left(e^{-\frac{20\log n}{d}L^{(t)}}\left(I - \frac{20\log n}{d}\Delta^{(t)} + \left(\frac{20\log n}{d}\Delta^{(t)}\right)^2\right)\right)$$

Taking expectation:

$$E\left[\Phi^{(t+1)}|G^t\right] = \Phi^{(t)} - \frac{4\log n}{d^3 n}\hat{tr}\left(e^{-\frac{20\log n}{d}L^{(t)}}\left(L^{(t)}\left(\frac{d}{2}\hat{I} - L^{(t)}\right)\right)\right)$$

Potential

We want to show that the potential decreases

$$\Phi^{(t+1)} = \hat{tr}\left(e^{-\frac{20\log n}{d}(L^{(t)}+\Delta^{(t)})}\right)$$

by Golden-Thompson inequality

$$= \hat{tr}\left(e^{-\frac{20\log n}{d}L^{(t)}}e^{-\frac{20\log n}{d}\Delta^{(t)}}\right)$$

by $e^{-A} = I - A + A^2$

$$= \hat{tr}\left(e^{-\frac{20\log n}{d}L^{(t)}}\left(I - \frac{20\log n}{d}\Delta^{(t)} + \left(\frac{20\log n}{d}\Delta^{(t)}\right)^2\right)\right)$$

Taking expectation:

$$E\left[\Phi^{(t+1)}|G^t\right] = \Phi^{(t)} - \frac{4\log n}{d^3 n}\hat{tr}\left(e^{-\frac{20\log n}{d}L^{(t)}}\left(L^{(t)}\left(\frac{d}{2}\hat{I} - L^{(t)}\right)\right)\right)$$

Potential

Using common diagonalization

$$\sum_{1 \le i \le n} e^{-\frac{20 \log n}{d} \lambda_i} \lambda_i (d/2 - \lambda_i)$$

Potential

Using common diagonalization

$$\sum_{1 \le i \le n} e^{-\frac{20 \log n}{d} \lambda_i} \lambda_i (d/2 - \lambda_i)$$

Two interesting cases:

$$\forall i : \lambda_i \ge \frac{d}{4}$$

$$\sum_{1 \le i \le n} e^{-\frac{20 \log n}{d} \lambda_i} \lambda_i (d/2 - \lambda_i) \in O(n^{-3})$$

Potential

Using common diagonalization

$$\sum_{1\leq i\leq n} e^{-\frac{20\log n}{d}\lambda_i}\lambda_i(d/2-\lambda_i)$$

Two interesting cases:

$$\exists i : \lambda_i < \frac{d}{4}$$

We look at:

$$\frac{\sum_{1\leq i\leq n} e^{-\frac{20\log n}{d}\lambda_i}\lambda_i(d/2-\lambda_i)}{\Phi^{(t)}} = \frac{\sum_{1\leq i\leq n} e^{-\frac{20\log n}{d}\lambda_i}\lambda_i(d/2-\lambda_i)}{\sum_{1\leq i\leq n} e^{-\frac{20\log n}{d}\lambda_i}}$$

Potential

Using common diagonalization

$$\sum_{1\le i\le n} e^{-\frac{20\log n}{d}\lambda_i}\lambda_i(d/2-\lambda_i)$$

Two interesting cases:

$$\exists i : \lambda_i < \frac{d}{4}$$

We look at:

$$\frac{\sum_{1\le i\le n} e^{-\frac{20\log n}{d}\lambda_i}\lambda_i(d/2-\lambda_i)}{\Phi^{(t)}} = \frac{\sum_{1\le i\le n} e^{-\frac{20\log n}{d}\lambda_i}\lambda_i(d/2-\lambda_i)}{\sum_{1\le i\le n} e^{-\frac{20\log n}{d}\lambda_i}}$$

$$\approx \frac{\sum_{1\le i\le k} e^{-\frac{20\log n}{d}\lambda_i}\lambda_i}{\sum_{1\le i\le k} e^{-\frac{20\log n}{d}\lambda_i}}$$

Potential

Using common diagonalization

$$\sum_{1 \le i \le n} e^{-\frac{20 \log n}{d} \lambda_i} \lambda_i (d/2 - \lambda_i)$$

Two interesting cases:

$$\exists i : \lambda_i < \frac{d}{4}$$

We look at:

$$\frac{\sum_{1 \le i \le n} e^{-\frac{20 \log n}{d} \lambda_i} \lambda_i (d/2 - \lambda_i)}{\Phi^{(t)}} = \frac{\sum_{1 \le i \le n} e^{-\frac{20 \log n}{d} \lambda_i} \lambda_i (d/2 - \lambda_i)}{\sum_{1 \le i \le n} e^{-\frac{20 \log n}{d} \lambda_i}}$$

$$\approx \frac{\sum_{1 \le i \le k} e^{-\frac{20 \log n}{d} \lambda_i} \lambda_i}{\sum_{1 \le i \le k} e^{-\frac{20 \log n}{d} \lambda_i}} \in \Omega\left(\frac{d}{n\sqrt{\log n}}\right)$$

Potential

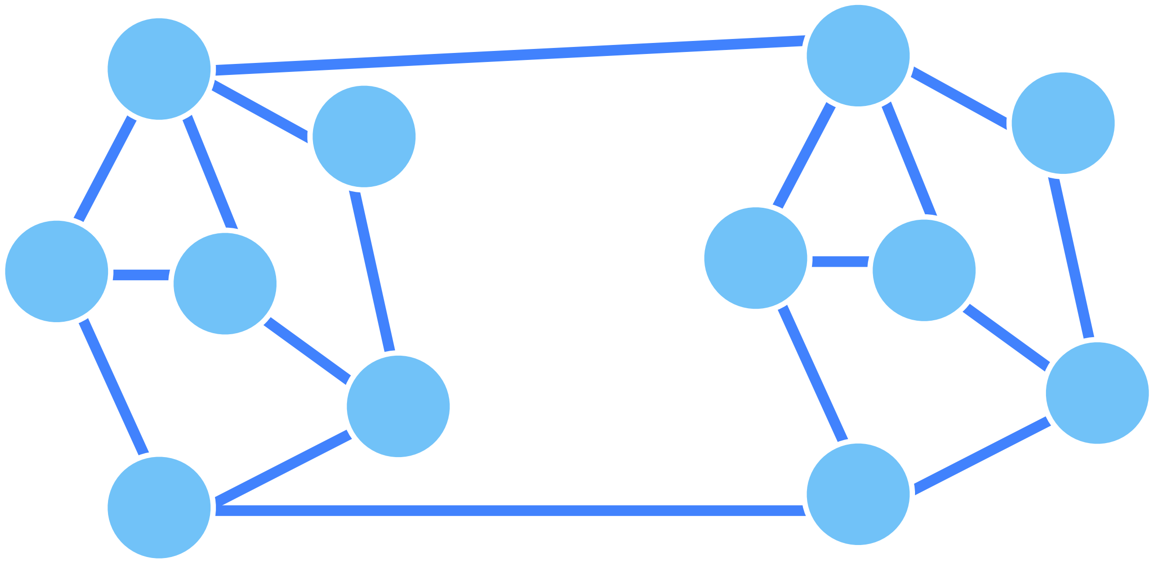
Thus:

$$E\left[\Phi^{(t+1)} | G^{(t)}\right] = \left(1 - \Omega\left(\frac{\sqrt{\log n}}{n^2 d^2}\right)\right)\Phi^{(t)} + O(n^{-3})$$

So in expectation $\Phi^{(t)}$ is in $O(n^{-3})$ after $O(n^2 d^2 \log n)$ steps, hence using Markov inequality we get the result.

Limit of our analysis

Expected additive improvement in a round can be $O\left(\frac{1}{n^2 d^2}\right)$



Conclusions and future directions

Conclusions

- New technique to analyze distribute protocol

- New convergence time analysis for flip and switch protocol

Future works

- Improve analysis of the flip
- Study parallelized version of the protocol
- Study node addition or deletion

Thanks!