Parallel and Distributed Graph Decomposition

Gary Miller (CMU)
Richard Peng (Geogia Tech)
Adrian Vladu (MIT)
Shen Chen Xu (CMU)

In Partial Fulfillment of the CSD Speaking Skills Requirement

Outline

- Graph decomposition.
- Exponential delay algorithm.
- Graph spanners with improved parameters in the parallel and distributed setting.

Graph Decomposition

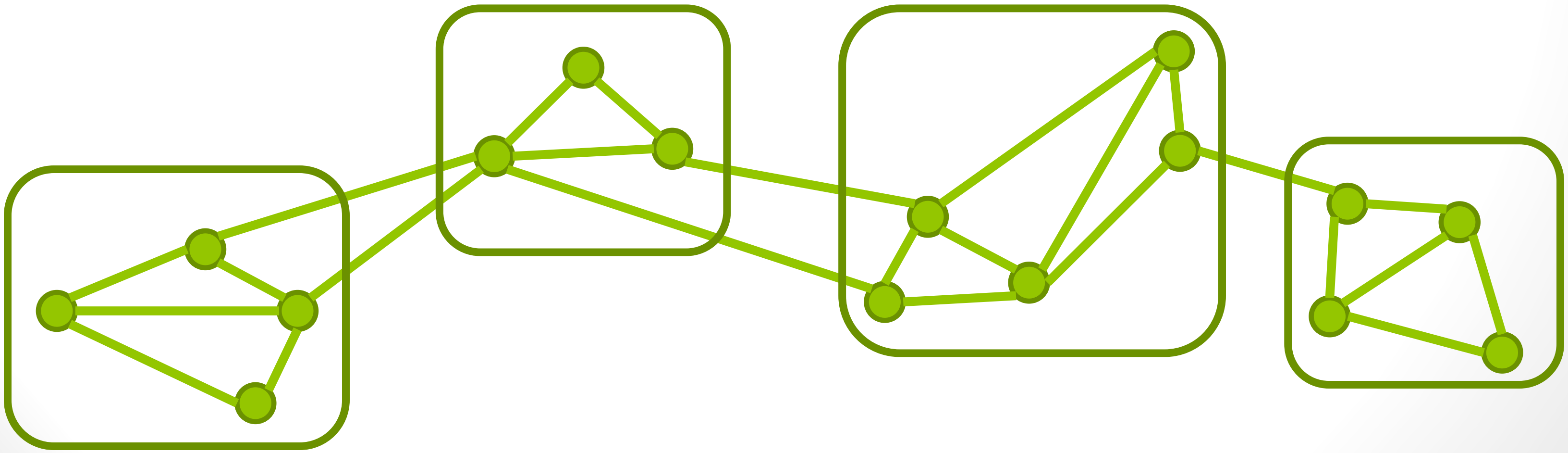
- Partition graph into smaller components.
- “Well connected” components.
- Different components are “loosely coupled”.

Graph Decomposition

- To enable divide and conquer algorithm.
  - Planar separator theorem.
- External memory computations.
  - Minimize disc access.
- Spectral clustering.
  - Cluster of graph => cluster of data.

Low Diameter Decomposition

- Partition of the graph:
  - Each component has small diameter.
  - Few edges between different components.



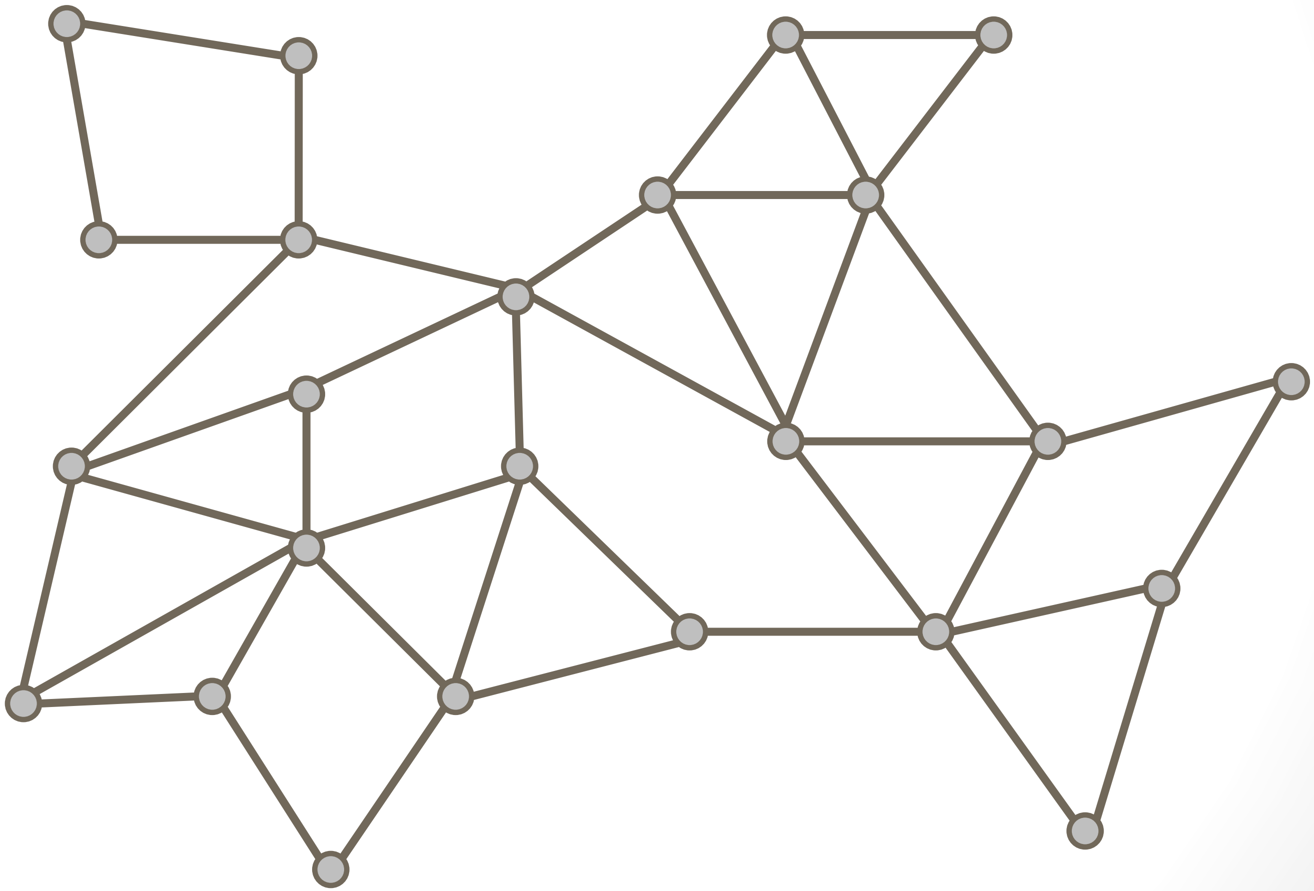
Low Diameter Decomposition

- Sequential algorithm by Awerbuch:
  - Each cluster has diameter $O\left(\frac{\log n}{\beta}\right)$.
  - β fraction of edges cut.
  - $O(m)$ runtime.

Outline

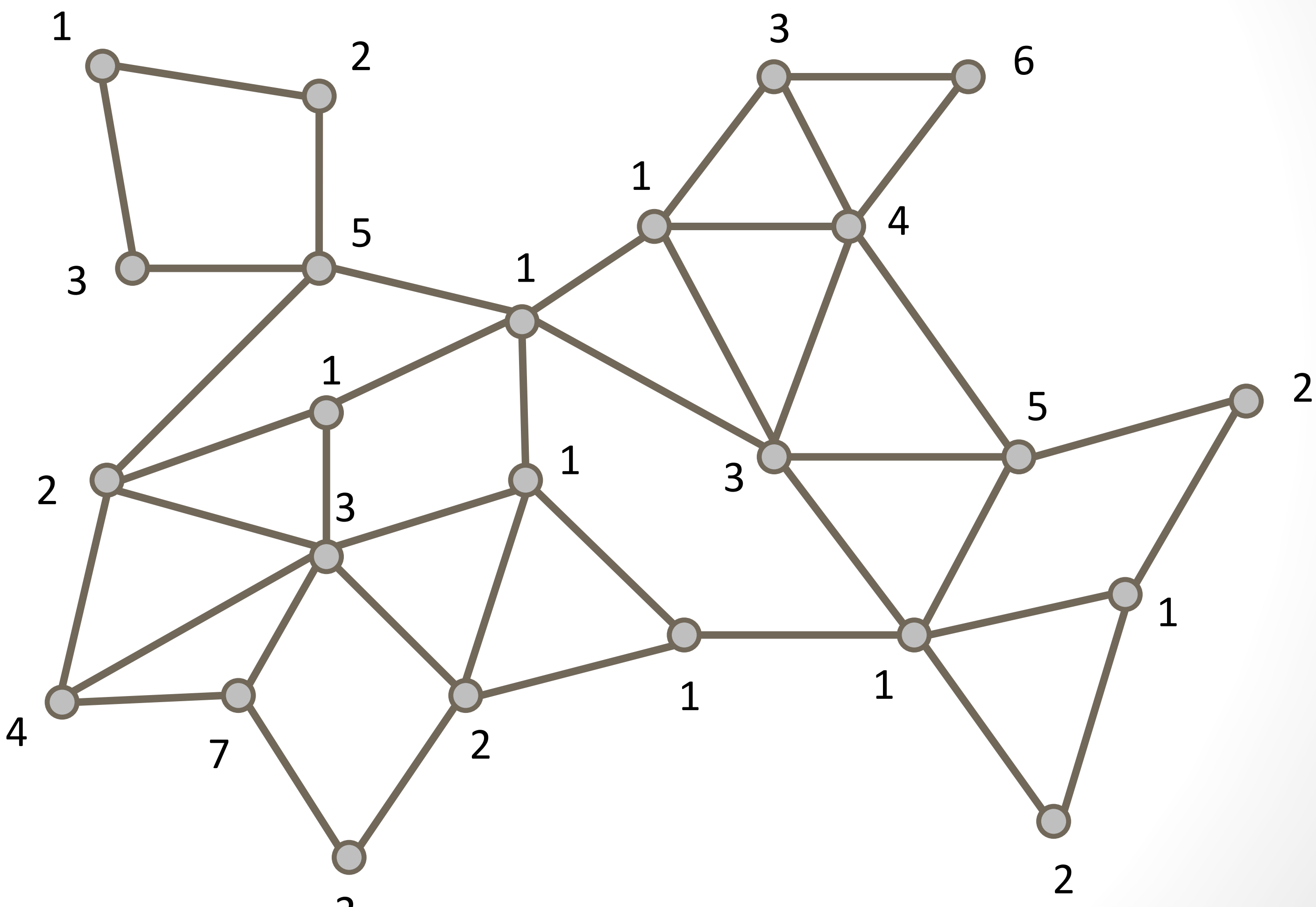
- Graph decomposition.
- Exponential delay algorithm.
- Graph spanners with improved parameters in the parallel and distributed setting.

An Example Run [MPX13]



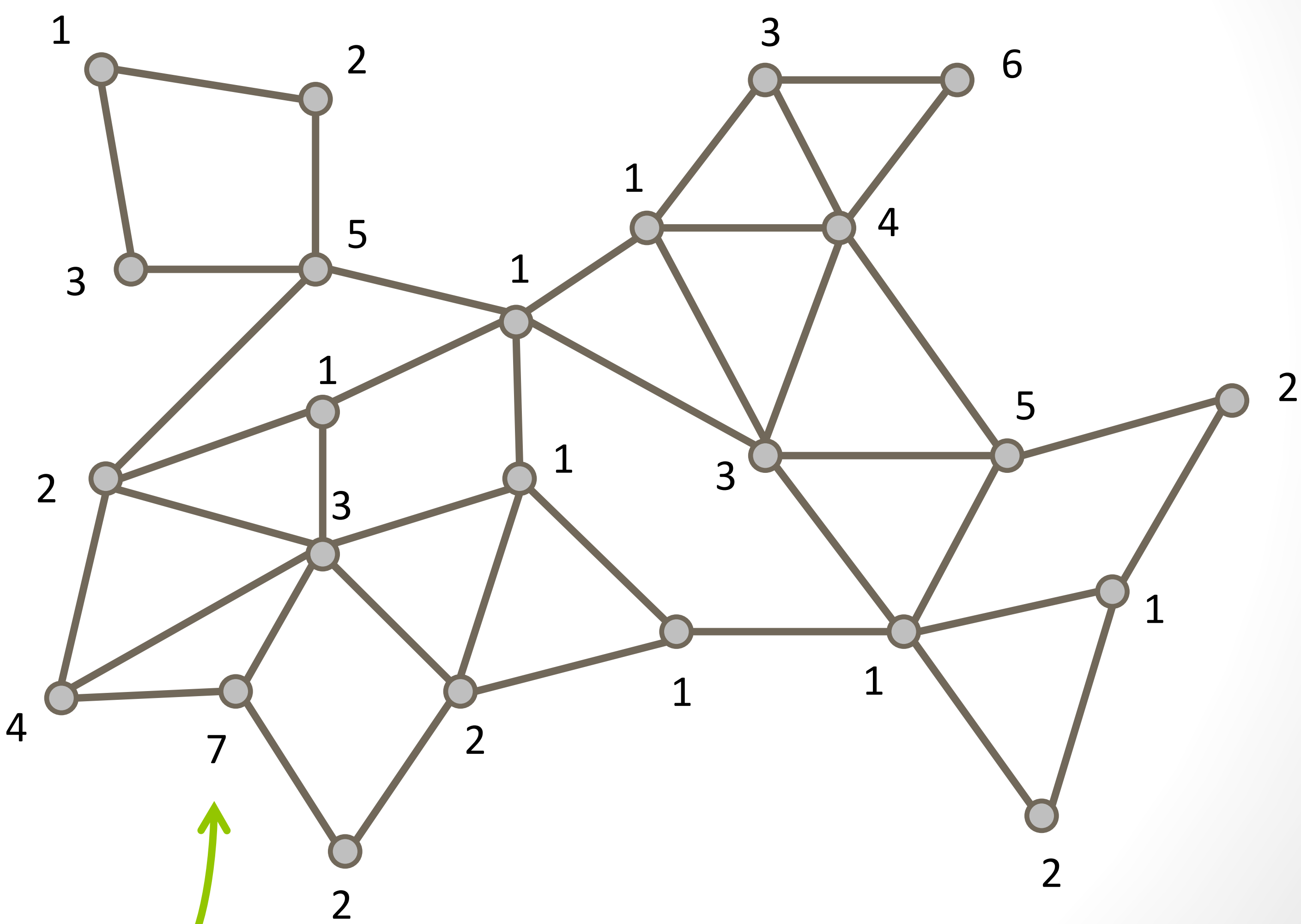
Vertex v draws $X_v \sim \mathrm{Exp}(\beta)$

An Example Run [MPX13]



Vertex v draws $X_v \sim \mathrm{Exp}(\beta)$

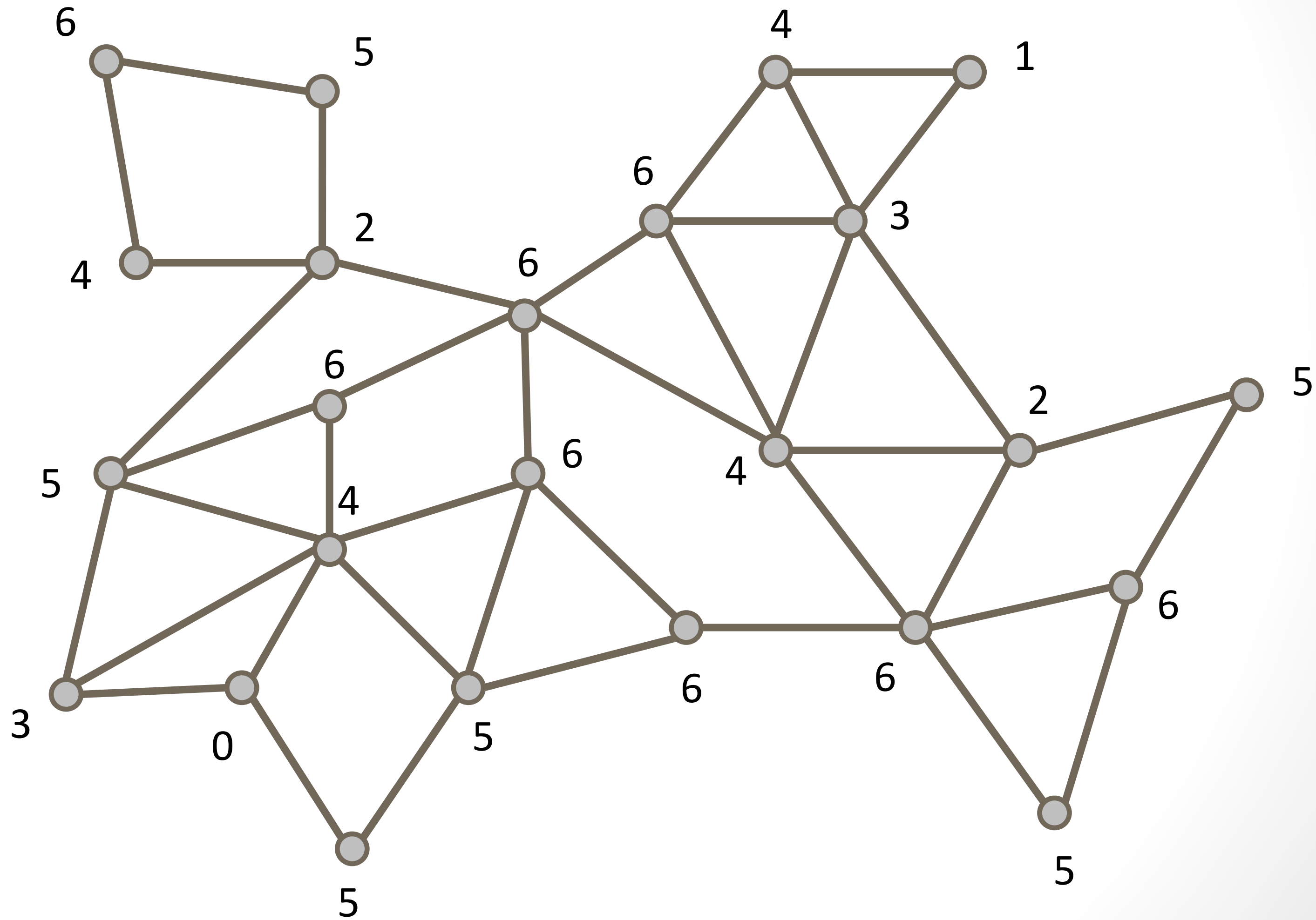
An Example Run [MPX13]



$X_{max} = 7$

Vertex v computes $X_{max} - X_v$

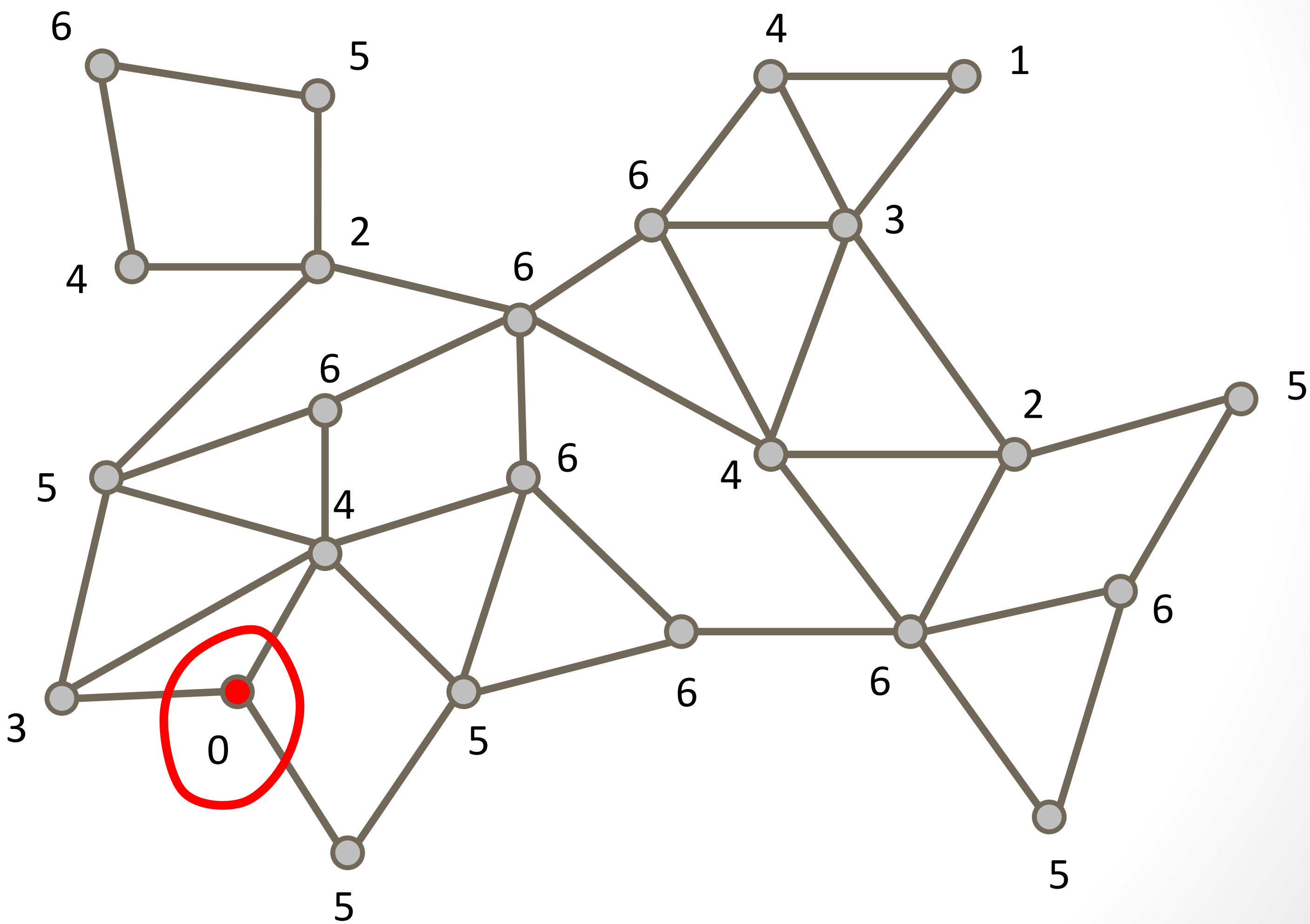
An Example Run [MPX13]



Vertex v computes $X_{max} - X_v$

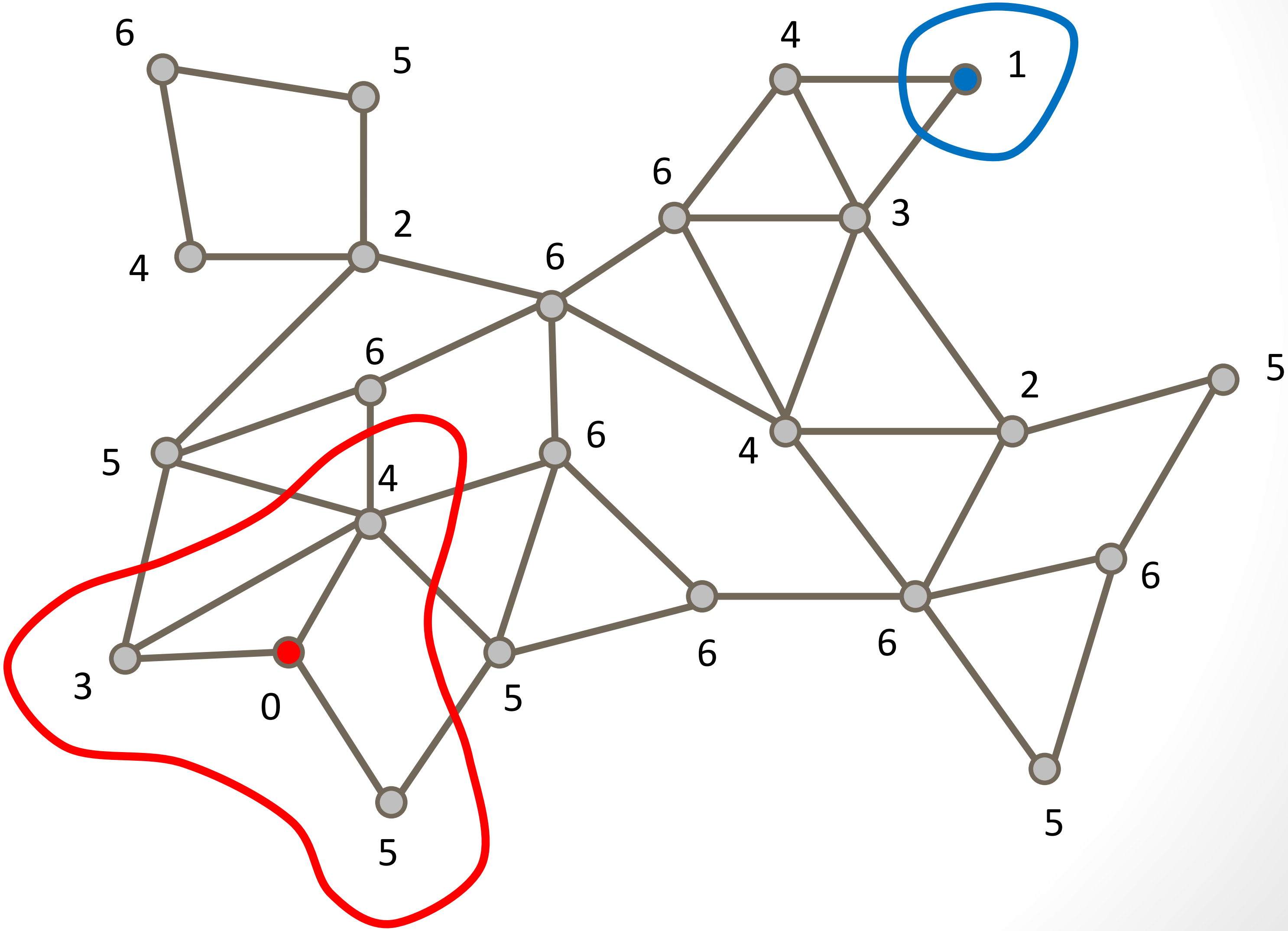
An Example Run [MPX13]

$t = 0$



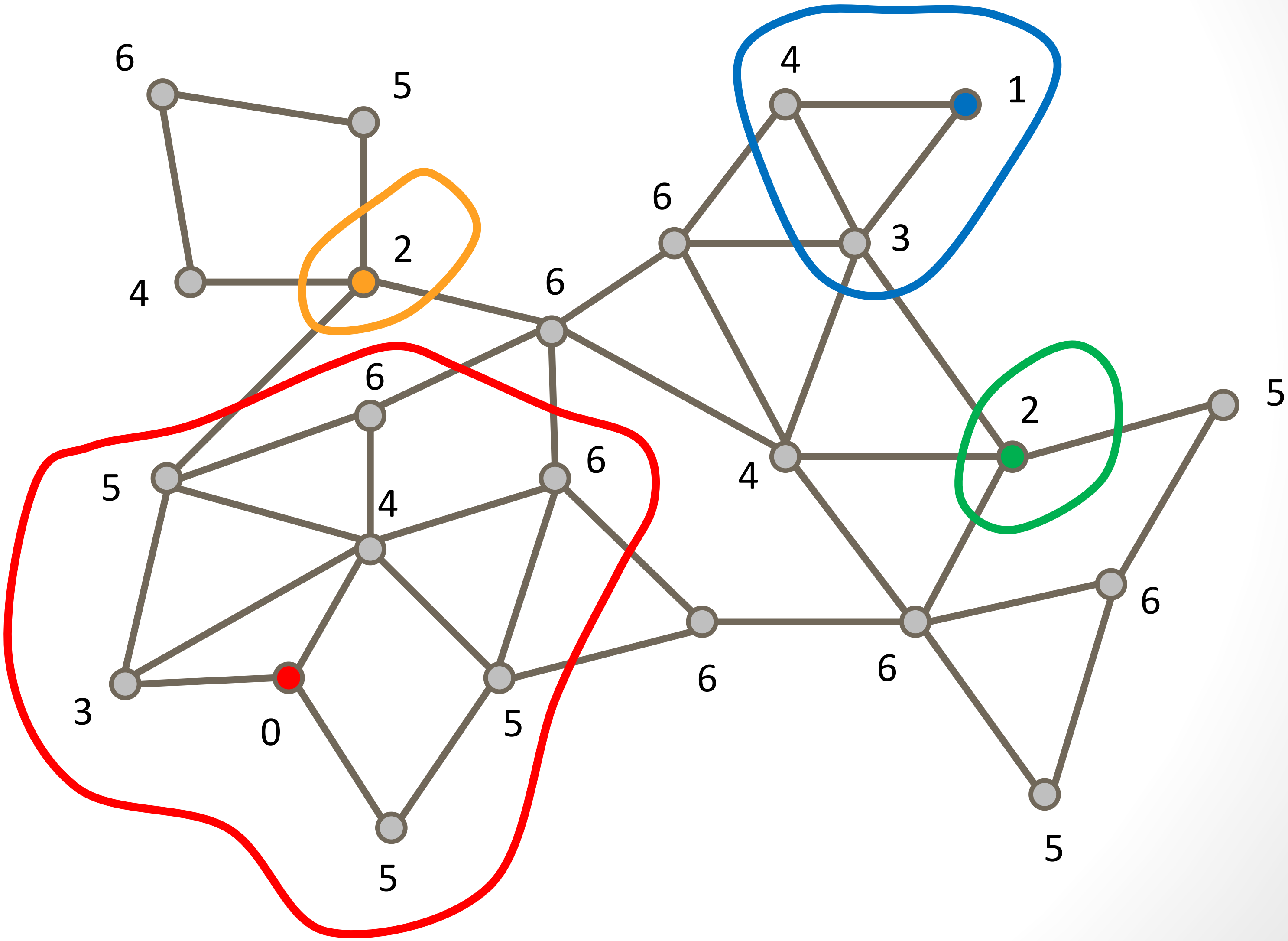
An Example Run [MPX13]

$t = 1$



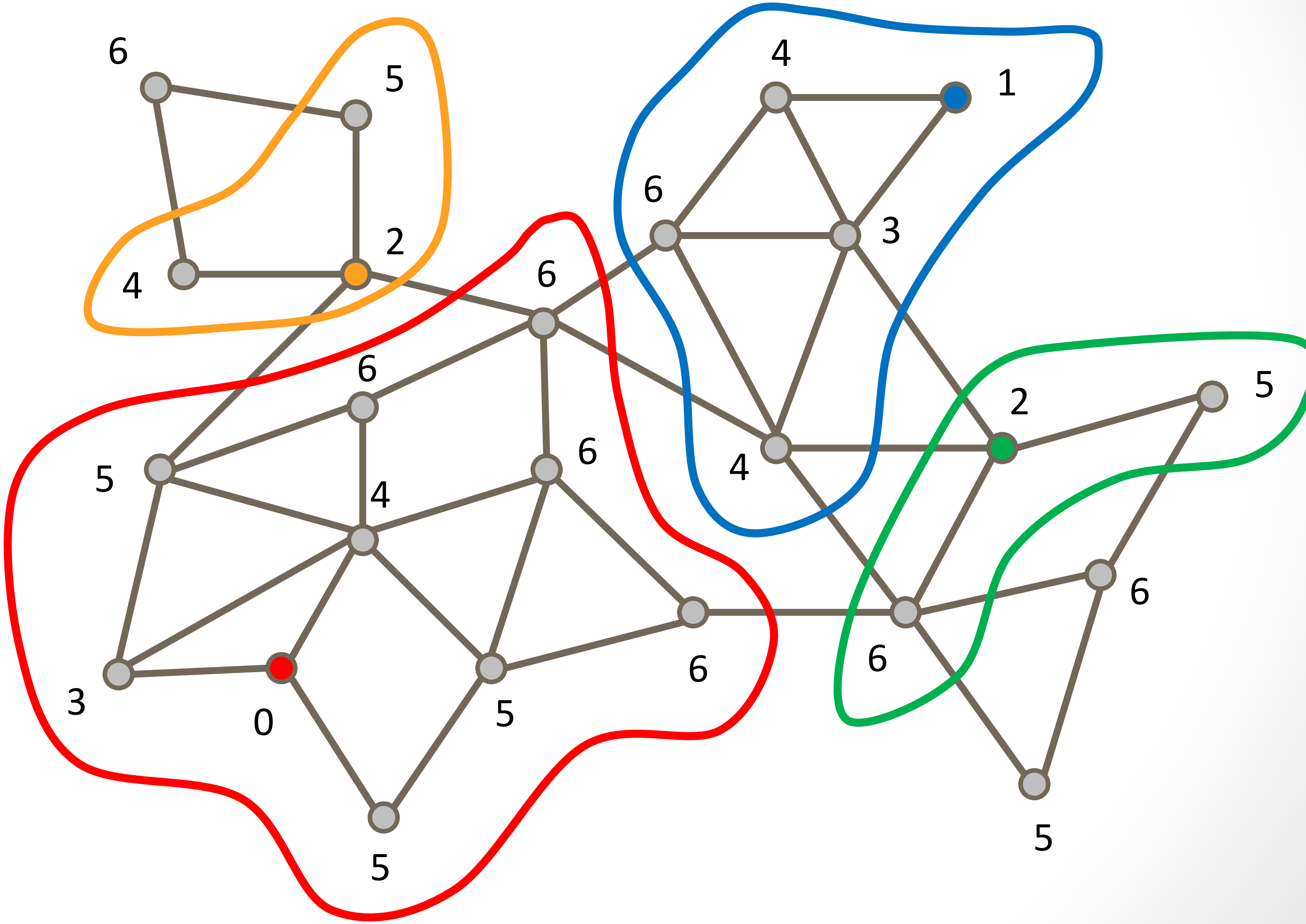
An Example Run [MPX13]

$t = 2$



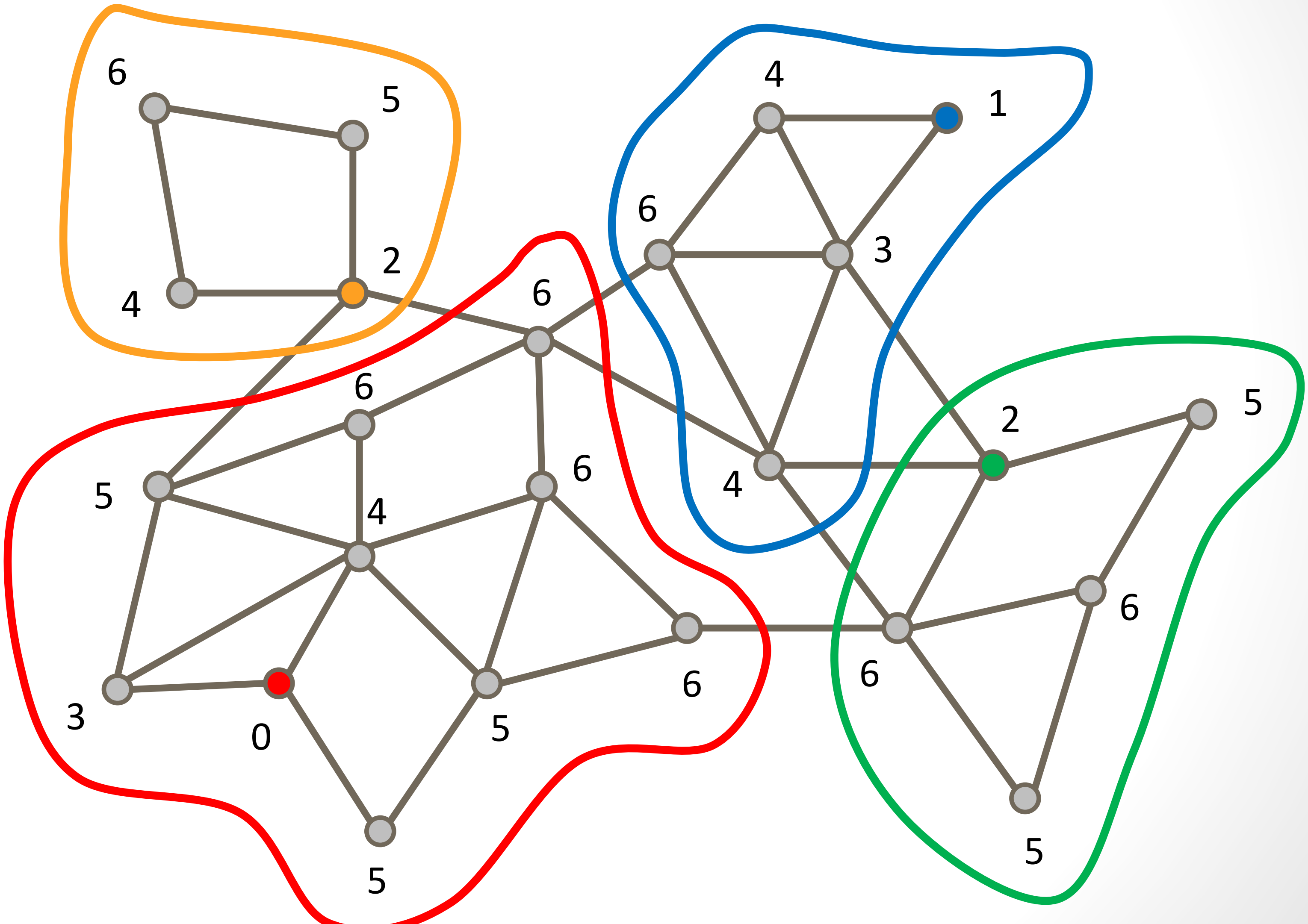
An Example Run [MPX13]

$t = 3$



An Example Run [MPX13]

$t = 4$



Exponential Delay Algorithm

- Each vertex v draws $X_v \sim \mathrm{Exp}(\beta)$.
- Each vertex v computes $X_{max} - X_v$.
- Each vertex v starts a BFS at time $X_{max} - X_v$.
  - Only search unexplored vertices.
- Alternatively: vertex v is assigned to $\mathrm{argmin}_u \, dist(u, v) + X_u$

Low Diameter Decomposition

- Sequential algorithm by Awerbuch:
  - Each cluster has diameter $O\left(\frac{\log n}{\beta}\right)$.
  - β fraction of edges cut.

Low Diameter Decomposition

- Our exponential delay algorithm:
  - Each cluster has diameter $O\left(\frac{\log n}{\beta}\right)$ *with high probability*.
  - β fraction of edges cut *in expectation*.

Low Diameter Decomposition

- Our exponential delay algorithm:
  - Each cluster has diameter $O\left(\frac{\log n}{\beta}\right)$ *with high probability*.
  - β fraction of edges cut *in expectation*.

Exponential Delay Algorithm

- Each vertex v draws $X_v \sim \text{Exp}(\beta)$.
- Each vertex v computes $X_{max} - X_v$.
- Each vertex v starts a BFS at time $X_{max} - X_v$.
  - Only search unexplored vertices.

The search has at most X_{max} levels!

Exponential Distribution

- Parmeter β
- Density function:

$$f(x) = \beta e^{-\beta x}$$

- Cumulative function:

$$F(x) = 1 - e^{-\beta x}$$

Bounding X_{max} and Diameter

- By union bound:

$$\Pr\left[X_{max} > \frac{k\log n}{\beta}\right] \leq n^{-(k-1)}$$

- Diameter $\leq 2X_{max} = O\left(\frac{\log n}{\beta}\right)$ w.h.p.!

Low Diameter Decomposition

- Our exponential delay algorithm:
  - Each cluster has diameter $O\left(\frac{\log n}{\beta}\right)$ *with high probability*.
  - β fraction of edges cut *in expectation*.

Exponential Distribution

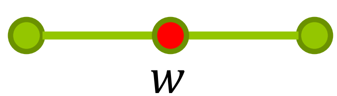
- *Memoryless property*:
  $\Pr(X > s + t \mid X > s) = \Pr(X > t)$
- Discrete analogue: geometric distribution.
  - Flipped s tails, when will I get a head?

Bounding # Edges Cut

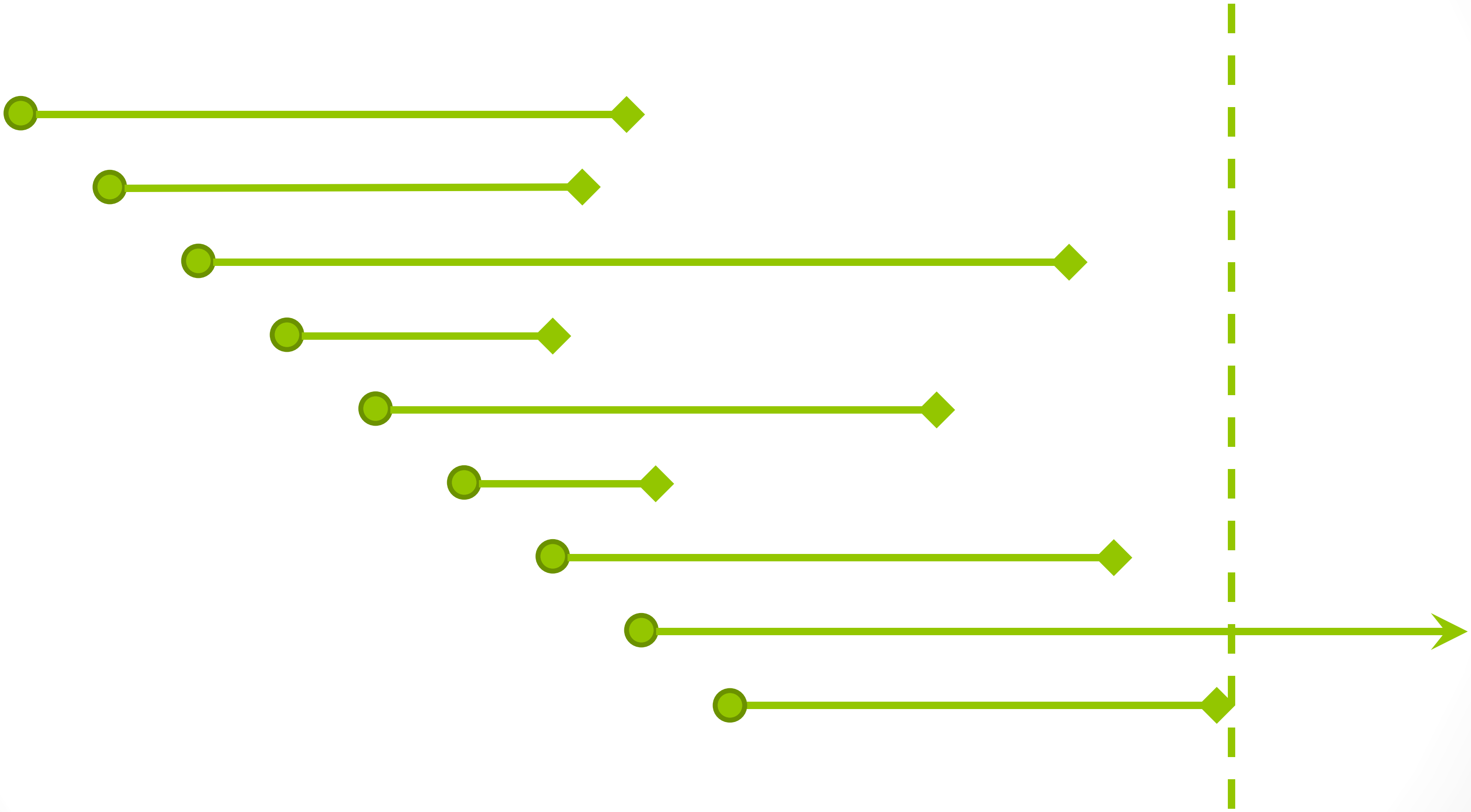
- Bound the cut probability for any edge by β.
- Linearity of expectation => β fraction of edges cut in expectation.

Bounding # Edges Cut

- Arrival time of v at midpoint w:
  - $dist(v, w) + (X_{max} - X_v)$
- Edge cut => First two arrivals within 1 unit of time.
- Early arrival <=> $(-dist(v, w) + X_v)$ is large.



Memorylessness



Memorylessness

Memorylessness

Memorylessness

$$F(x) = 1 - \exp(-\beta x) \leq \beta$$

Bounding # Edges Cut

- Each edge is cut with probability at most β.
- Linearity of expectation => β fraction of edges cut in expectation.

Implementation

- Inherently parallel.
- Can be implemented using BFS.
- Use only integer part with random tie breaking.
- Parallel/distributed time: $\tilde{O}\left(\frac{\log n}{\beta}\right)$.
- Work efficient.

Outline

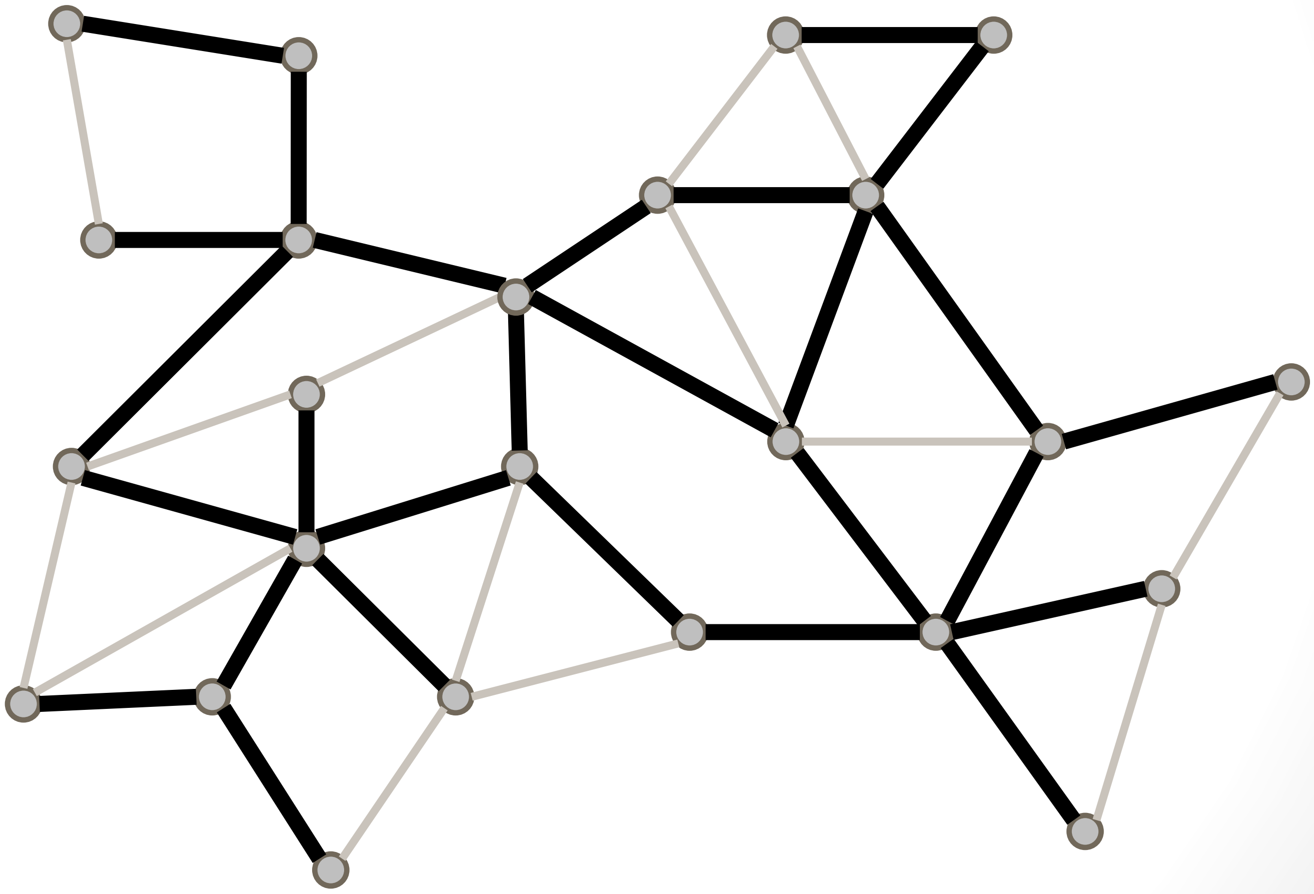
- Graph decomposition.
- Exponential delay algorithm.
- Graph spanners with improved parameters in the parallel and distributed setting.

Spanners

- Given G, a k-spanner H is a subgraph s.t. $dist_H(u, v) \leq k \cdot dist_G(u, v)$ for all $u, v \in V(G)$.

Spanners

3-spanner



Spanners

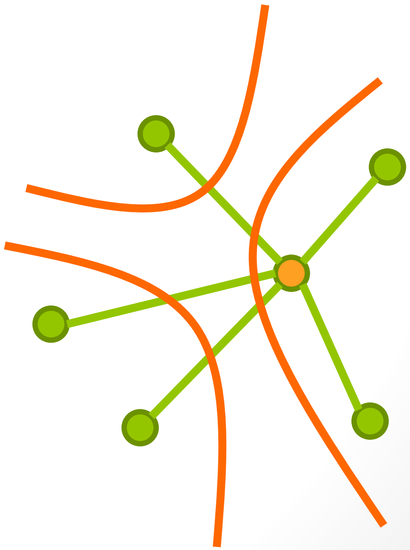
- Given G, a k-spanner H is a subgraph s.t. $dist_H(u, v) \leq k \cdot dist_G(u, v)$ for all $u, v \in V(G)$.
- k is called the stretch factor.
- Goal: given k, find spanner of small size.

Spanners

- $\exists(2k-1)$-spanner with $\frac{1}{2}n^{1+1/k}$ edges [PS89,ADD+93].
- Tight up to the Erdös girth conjecture.

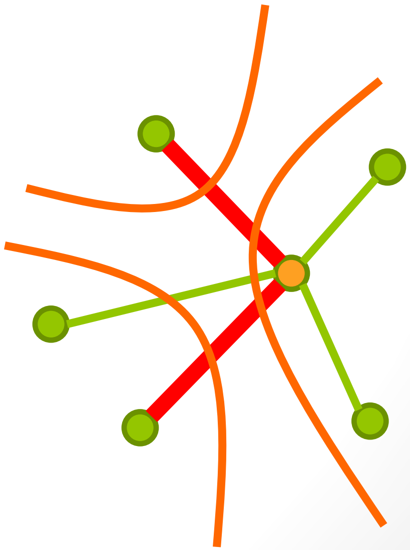
Spanners via Graph Decomposition

- Set $\beta = \log n / k$.
- Compute a low diameter decomposition.
- Include the spanning forest.
- Each boundary vertex connects to each adjacent cluster.

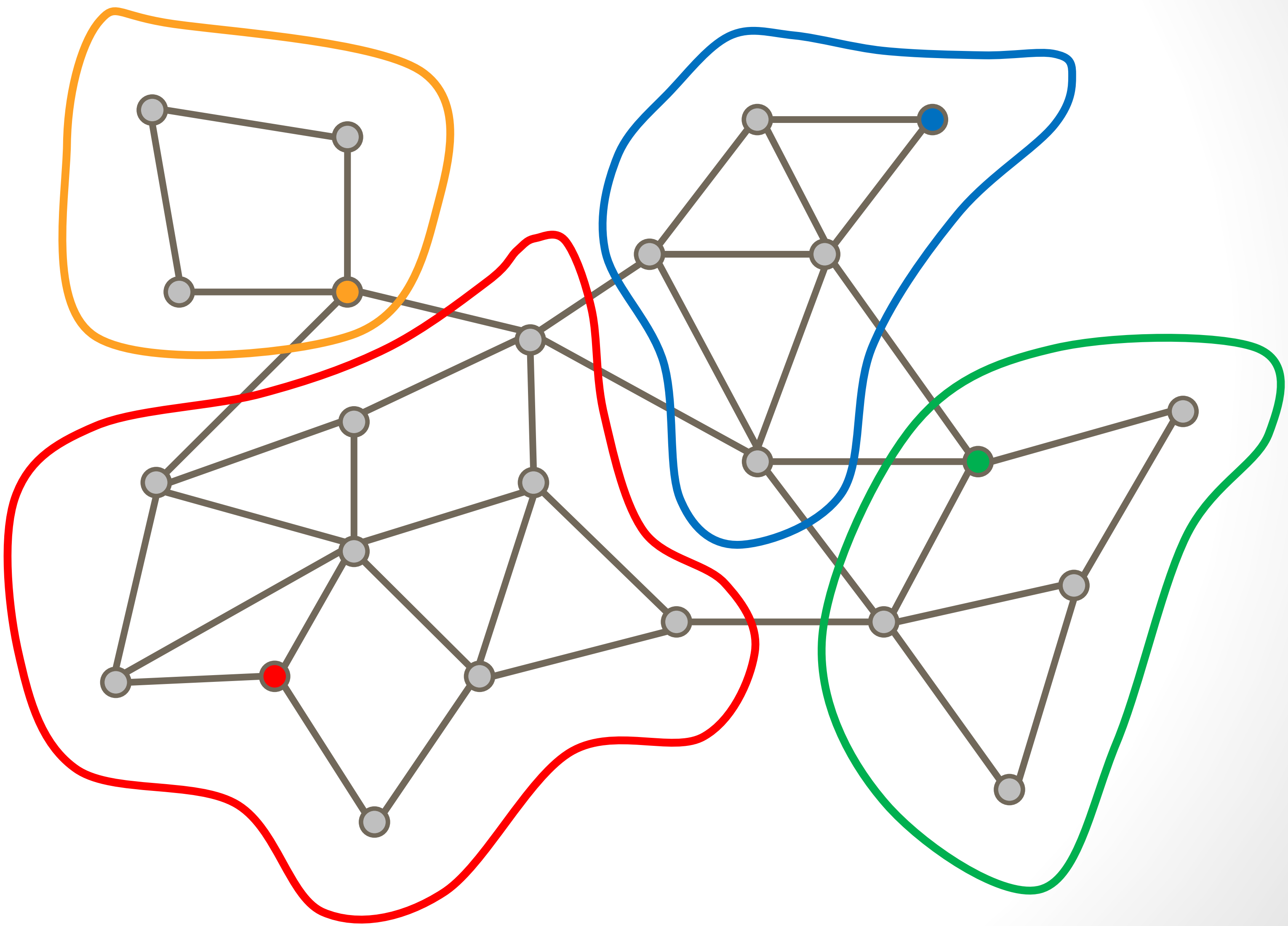


Spanners via Graph Decomposition

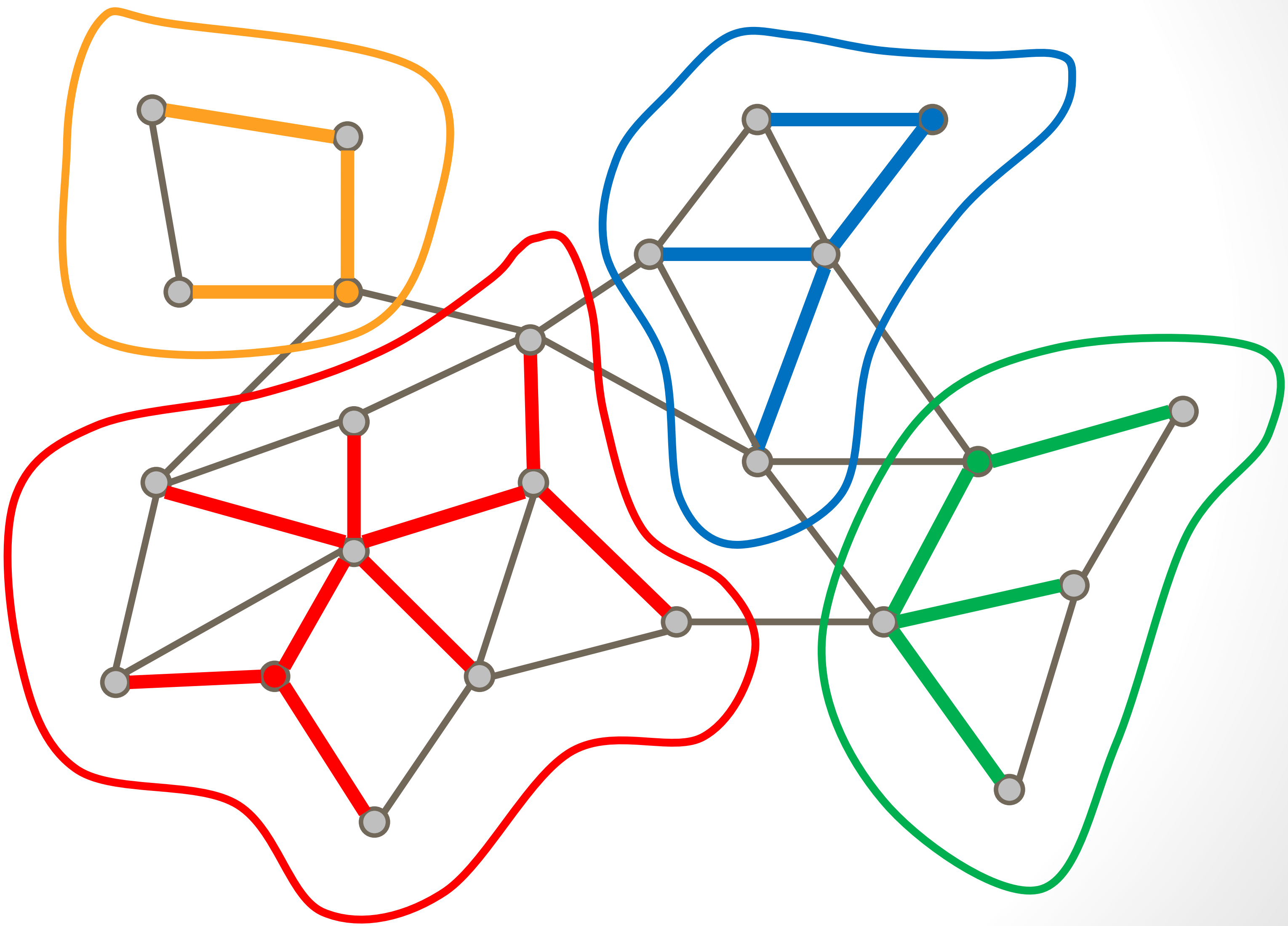
- Set $\beta = \log n / k$.
- Compute a low diameter decomposition.
- Include the spanning forest.
- Each boundary vertex connects to each adjacent cluster.



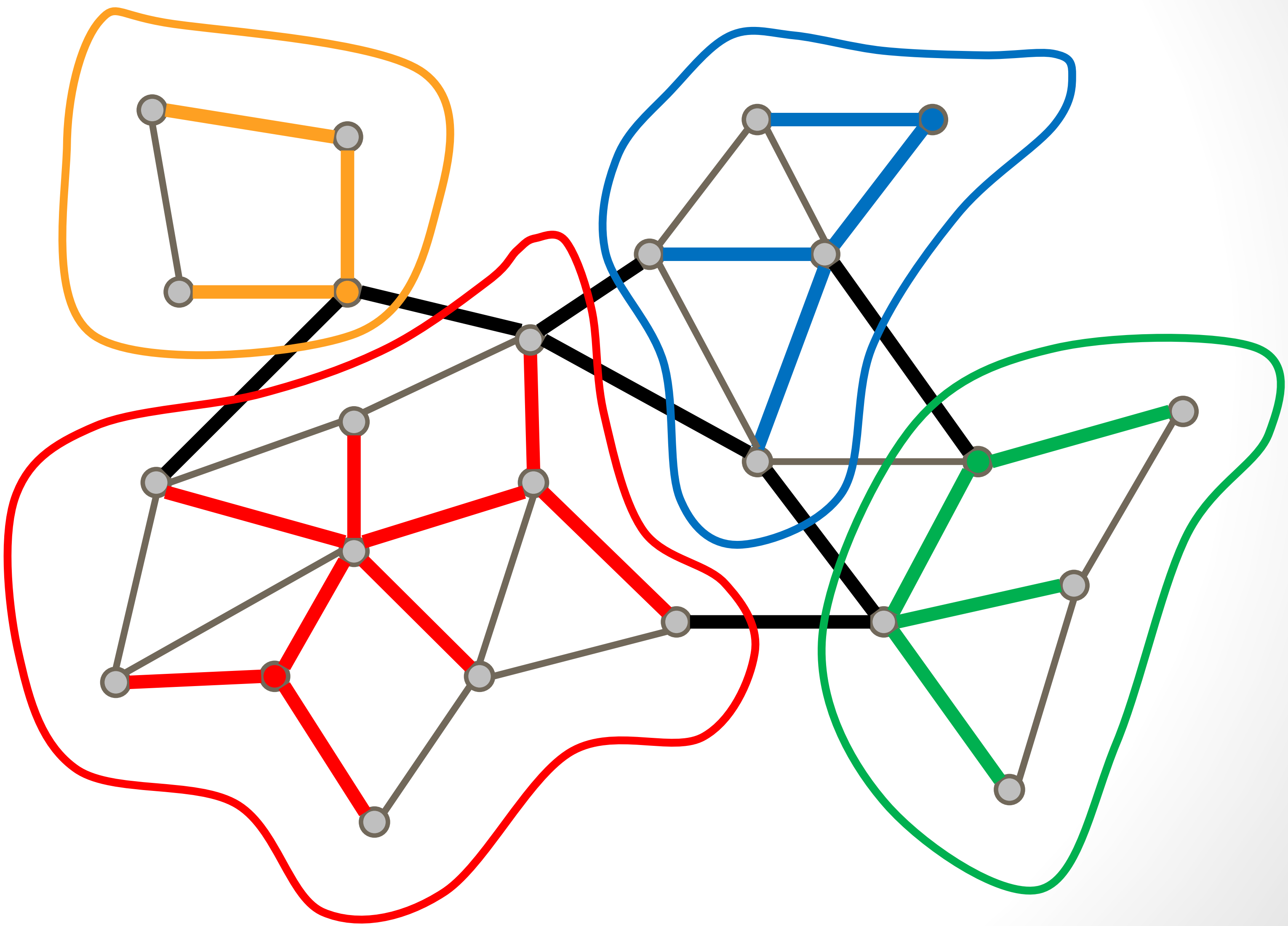
An Example Run [MPVX15]



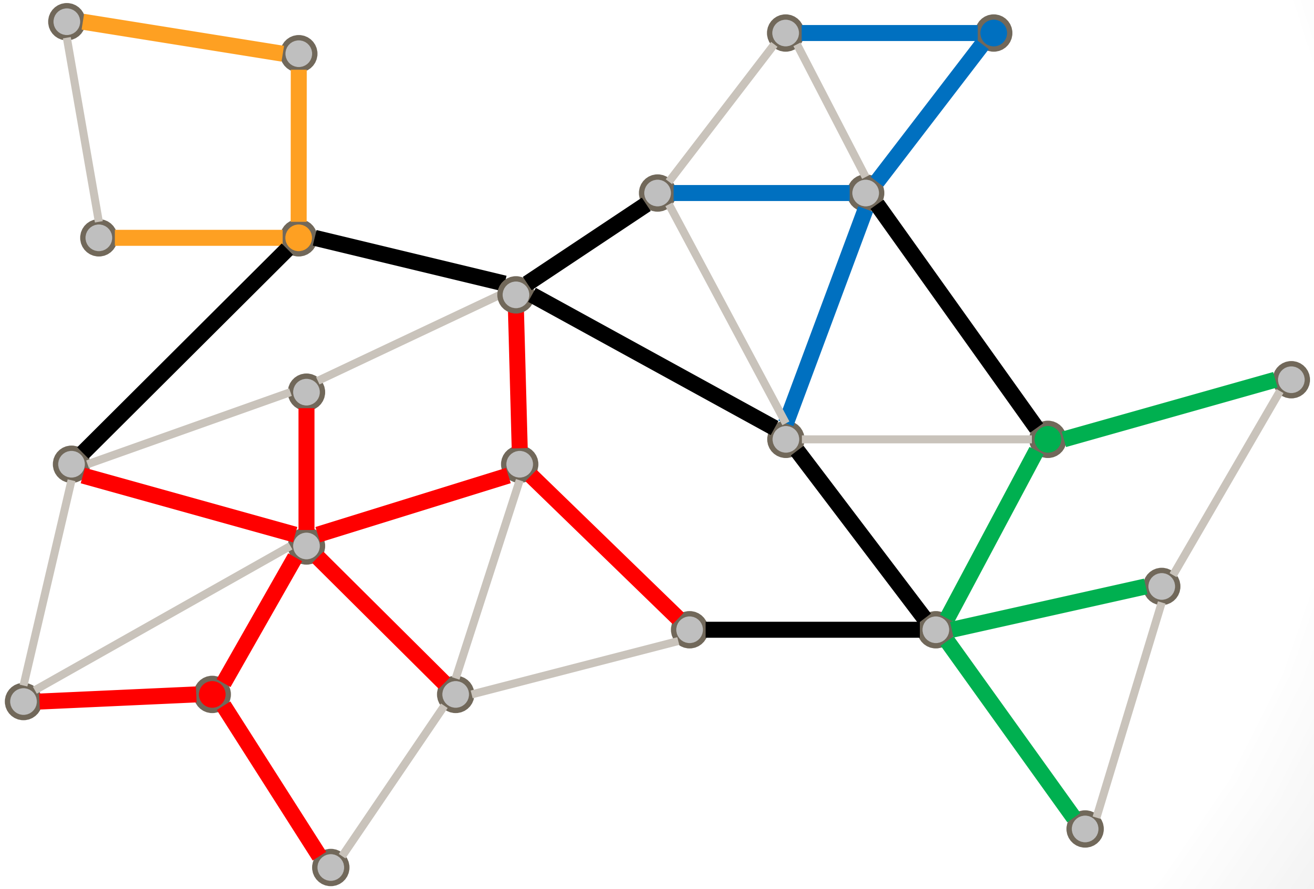
An Example Run [MPVX15]



An Example Run [MPVX15]

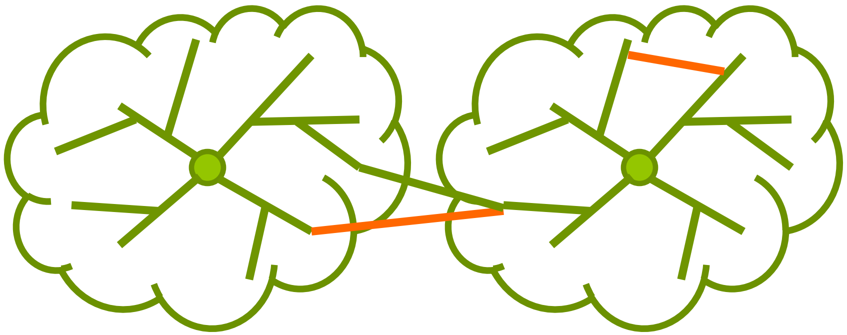


An Example Run [MPVX15]



Bounding Stretch

- Stretch bounded by diameter:
  - $O(\log n/\beta) = O(k)$ w.h.p.
  - $2\log n/\beta + 1 = 4k + 1$ in expectation.

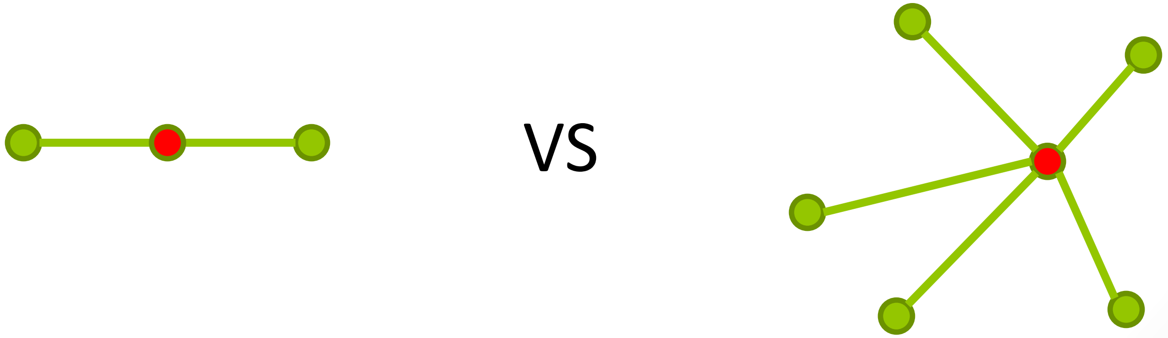


Bounding Spanner Size

- Goal: $O(n^{1+1/k})$ size.
- $|\text{spanning forest}| \leq n - 1$.
- Inter-cluster edge:
  - Bound contribution of each vertex.
  - # of cluster intersecting neighborhood.

Contribution of a Vertex

- How many cluster does a neighborhood intersect?
- Previously: how many cluster does an edge intersect?



Memorylessness

$$F(x) = (1 - \exp(-2\beta))^{-3}$$

Bounding Spanner Size

- Let L be the # of clusters intersecting the neighborhood.
- $\mathrm{E}[L] = \sum_{\ell=1}^{\infty} \Pr[L \geq \ell]$

$$= \sum_{\ell=1}^{\infty} (1 - \exp(-2\beta))^{\ell-1}$$

$$= \frac{1}{1-(1-\exp(-2\beta))}$$

$$= n^{1/k}$$

Bounding Spanner Size

- $|\text{spanning forest}| \leq n - 1$.
- Inter-cluster edge:
  - Contribution per vertex: $n^{1/k}$
- Overall $O(n^{1+1/k})$.

Comparison to Previous Works

Stretch	Expected Size	Work	Parallel/Distributed time	Notes
$2k-1$	$\frac{1}{2}n^{1+1/k}$	$O(m)$	$O(m)$	[PS98]
$2k-1$	$O(kn^{1+1/k})$	$O(km)$	$\tilde{O}(k)$	[BS07]
$4k+1$	$O(n^{1+1/k})$	$O(m)$	$\tilde{O}(k)$	New

Other Applications

- Spanners [PS89].
- Low stretch tree metrics/spanning tree [AKPW95, Bar96, FRT03, EEST08, AN12].
- SDD linear system solver [ST04, KMP11, BGK+13, CKM+14].
- Parallel graph connectivity [SDB14].
- Work efficient parallel shortest paths [MPVX15].

Conclusion

- Parallel/distributed low diameter decomposition.
- Memoryless property key to our analysis.
- Parallel and distributed spanners with improved parameters.
- Future direction: spanner for weighted graphs.