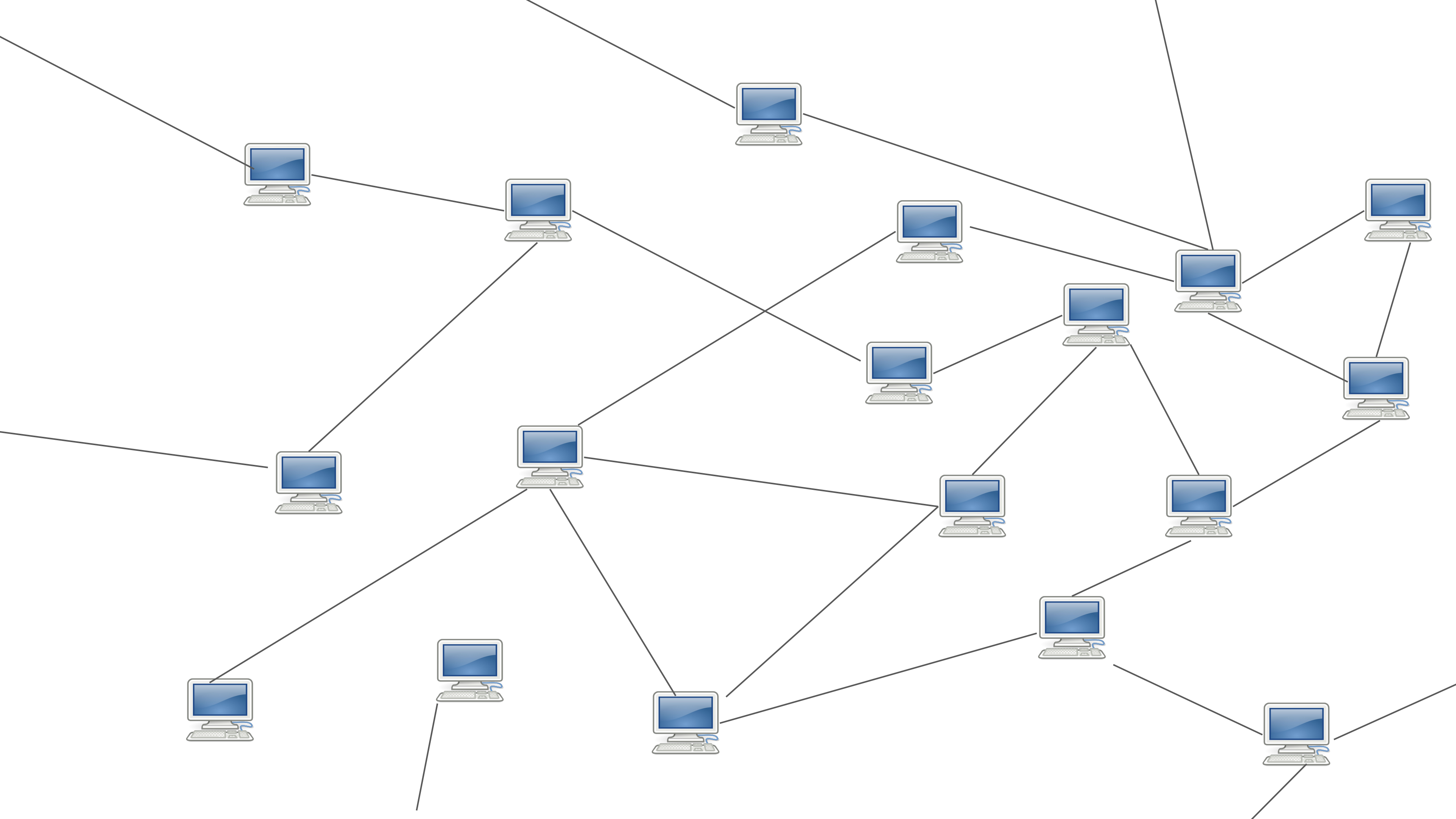
Local Complexity: New Results and Bridges to Other Fields

Vaclav Rozhon
Supervisors: Mohsen Ghaffari / Emo Welzl / Angelika Steger



Thesis based on the following papers

Václav Rozhoň ® Bernhard Haeupler ® Christoph Grunau. A simple deterministic distributed low-diameter clustering.

Sebastian Brandt, Christoph Grunau, and Václav Rozhoň. Generalizing the sharp threshold phenomenon for the distributed complexity of the Lovász local lemma.

Christoph Grunau ® Václav Rozhoň ® Sebastian Brandt. The landscape of distributed complexities on trees and beyond.

Sebastian Brandt ® Christoph Grunau ® Václav Rozhoň. The randomized local computation complexity of the Lovász local lemma.

Václav Rozhoň ® Michael Elkin ® Christoph Grunau ® Bernhard Haeupler. Deterministic low-diameter decompositions for weighted graphs and distributed and parallel applications.

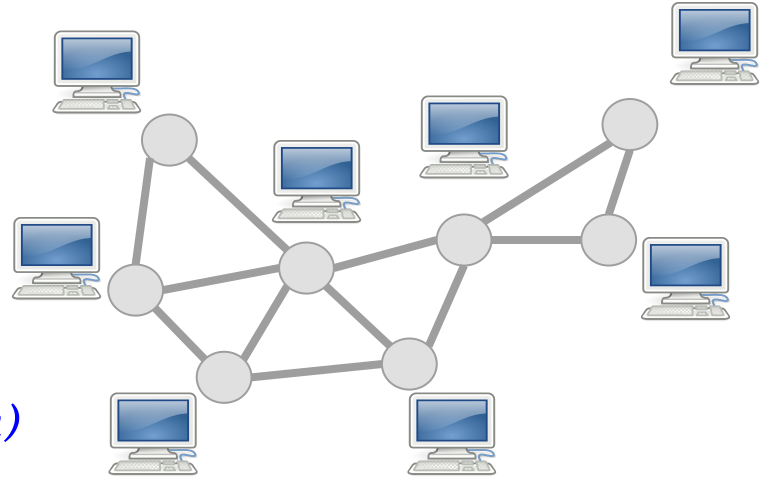
Sebastian Brandt, Yi-Jun Chang, Jan Grebík, Christoph Grunau, Václav Rozhoň, and Zoltán Vidnyánszky. Local problems on trees from the perspectives of distributed algorithms, finitary factors, and descriptive combinatorics.

Local Algorithms

- Undirected graph on n nodes
- One computer in each node

- Synchronous message passing rounds
- Unbounded message size and computation

- Initially, nodes know only n and their unique $O(\log n)$ bit identifier
- In the end, each node should know its part of output

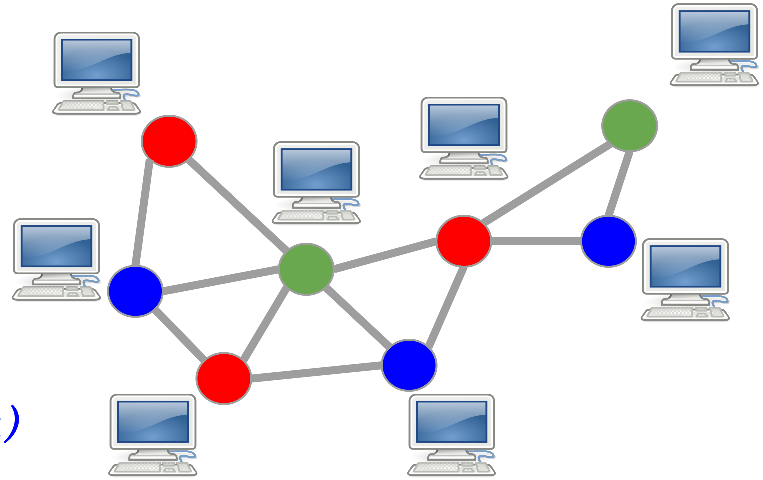


Local Algorithms

- Undirected graph on n nodes
- One computer in each node

- Synchronous message passing rounds
- Unbounded message size and computation

- Initially, nodes know only n and their unique $O(\log n)$ bit identifier
- In the end, each node should know its part of output

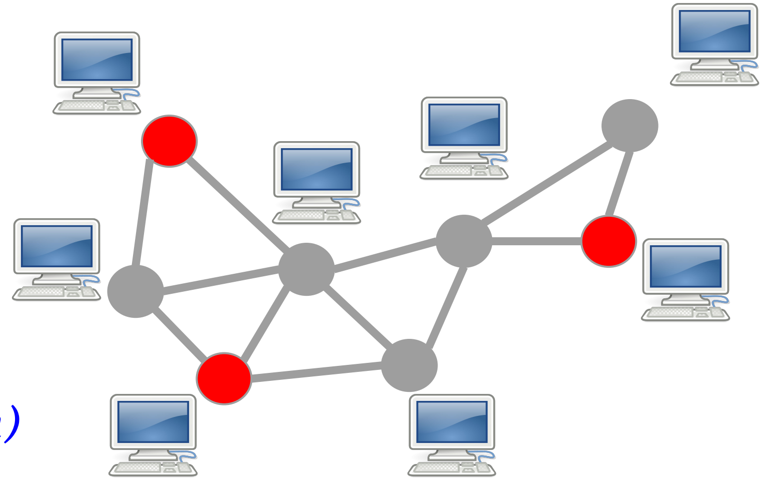


Local Algorithms

- Undirected graph on n nodes
- One computer in each node

- Synchronous message passing rounds
- Unbounded message size and computation

- Initially, nodes know only n and their unique $O(\log n)$ bit identifier
- In the end, each node should know its part of output



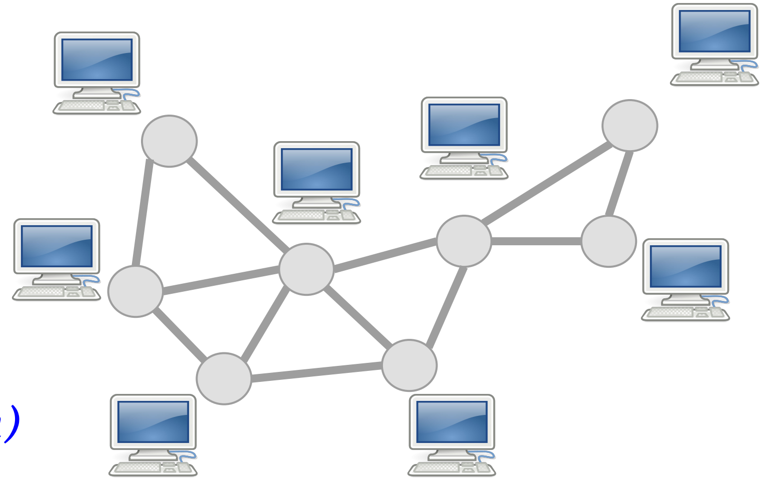
Local Algorithms

- Undirected graph on n nodes
- One computer in each node

- Synchronous message passing rounds
- Unbounded message size and computation

- Initially, nodes know only n and their unique $O(\log n)$ bit identifier
- In the end, each node should know its part of output

- Time complexity: number of rounds



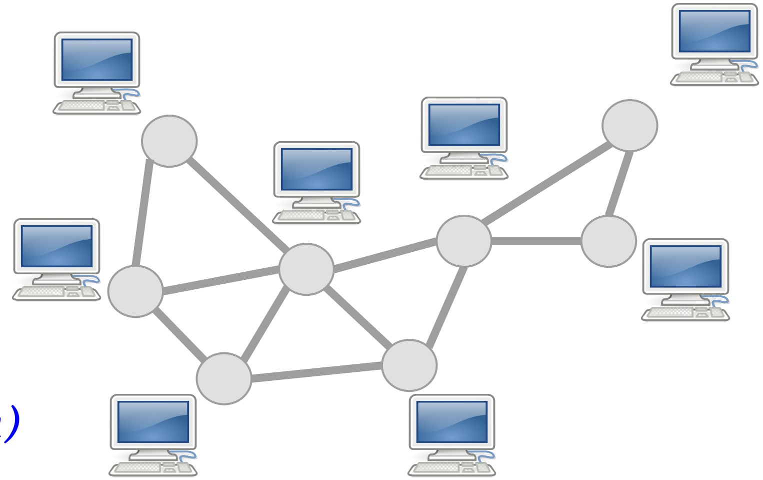
Local Algorithms

- Undirected graph on n nodes
- One computer in each node

- Synchronous message passing rounds
- Unbounded message size and computation

- Initially, nodes know only n and their unique $O(\log n)$ bit identifier
- In the end, each node should know its part of output

- Time complexity: number of rounds



Formalized by Nati Linial around 1990

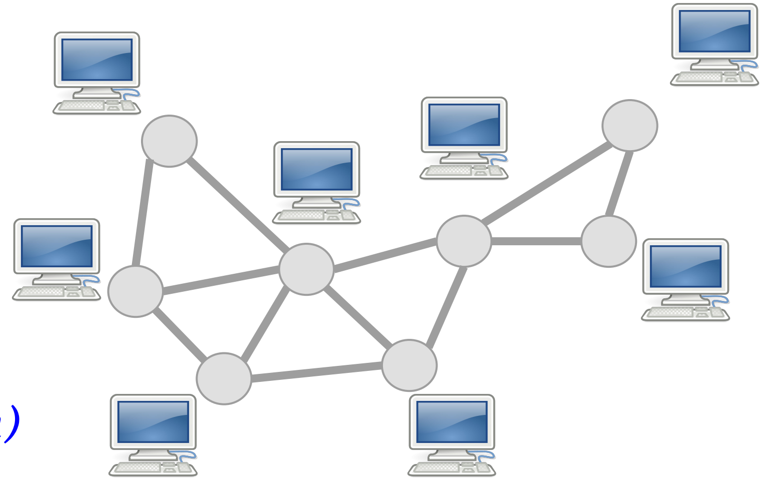
Local Algorithms

- Undirected graph on n nodes
- One computer in each node

- Synchronous message passing rounds
- Unbounded message size and computation

- Initially, nodes know only n and their unique $O(\log n)$ bit identifier
- In the end, each node should know its part of output

- Time complexity: number of rounds



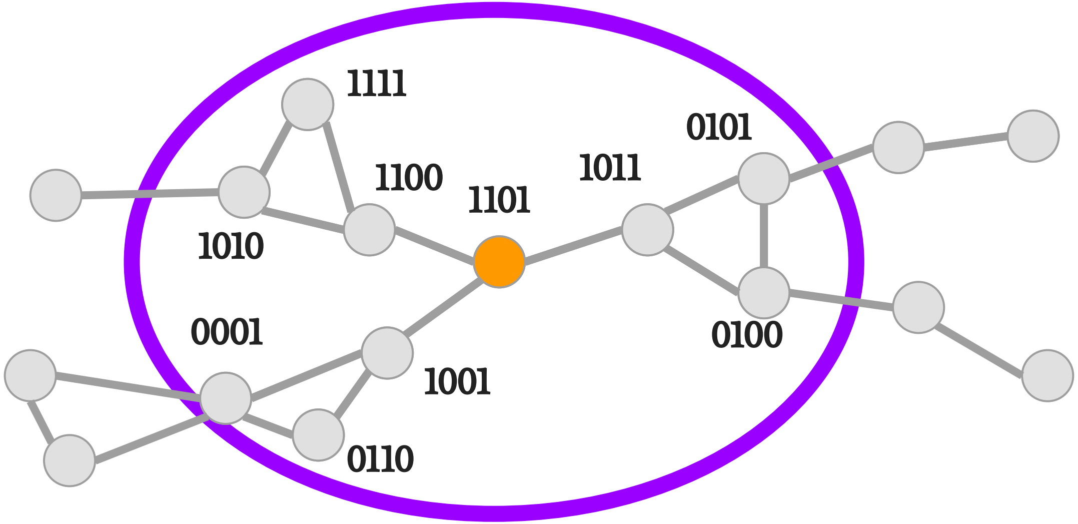
This is a very simple model!!!

\+ mathematically clean

\- further from practice

Equivalent Definition

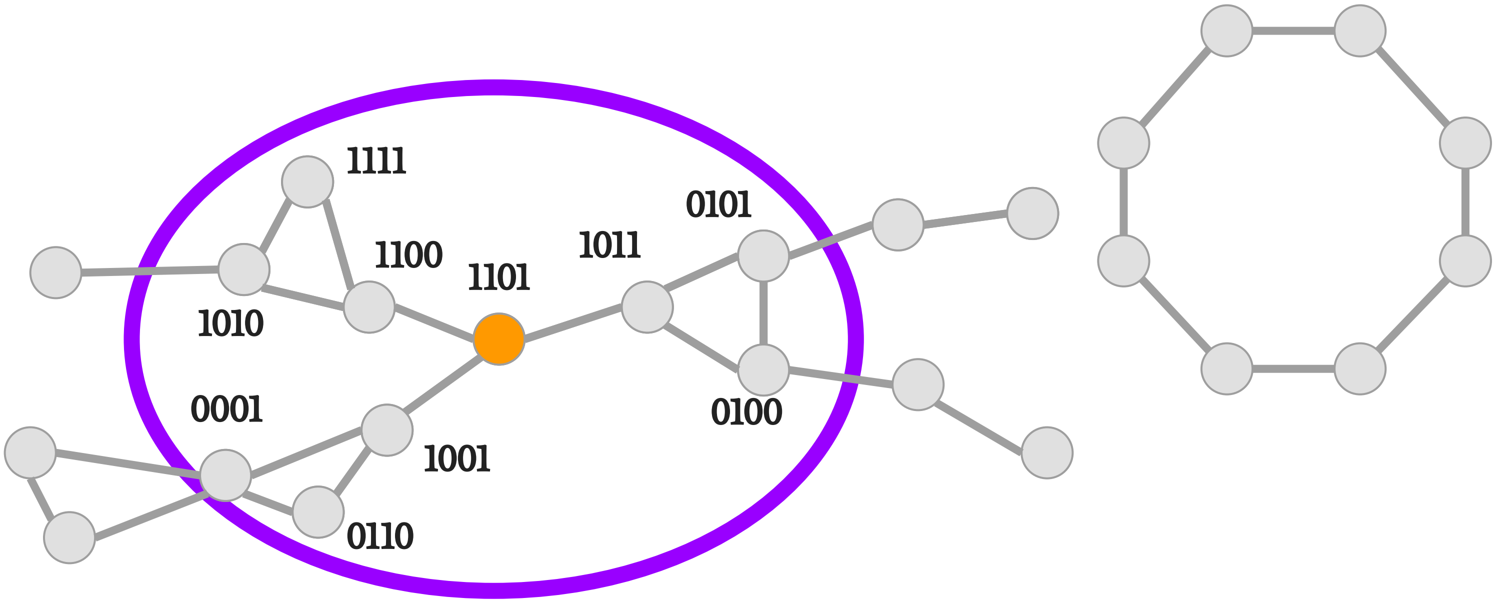
A t(n)-round local algorithm is a function mapping labeled t(n)-hop neighborhoods to the final output.



Equivalent Definition

A t(n)-round local algorithm is a function mapping labeled t(n)-hop neighborhoods to the final output.

this is why we need identifiers



What kind of problems can be solved efficiently with local algorithms?

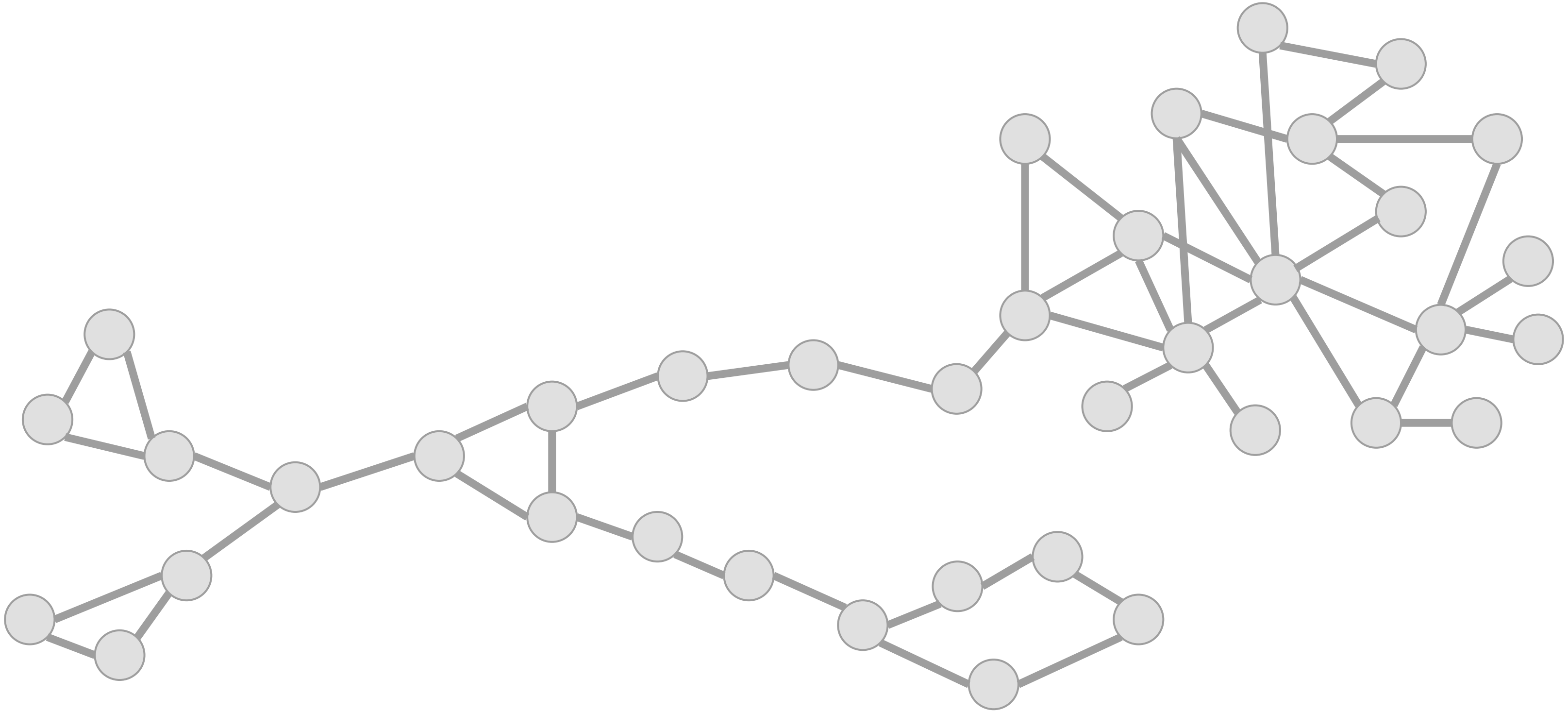
What kind of problems can be solved efficiently with local algorithms?

Actually, we have a pretty good understanding of this question!!!!!

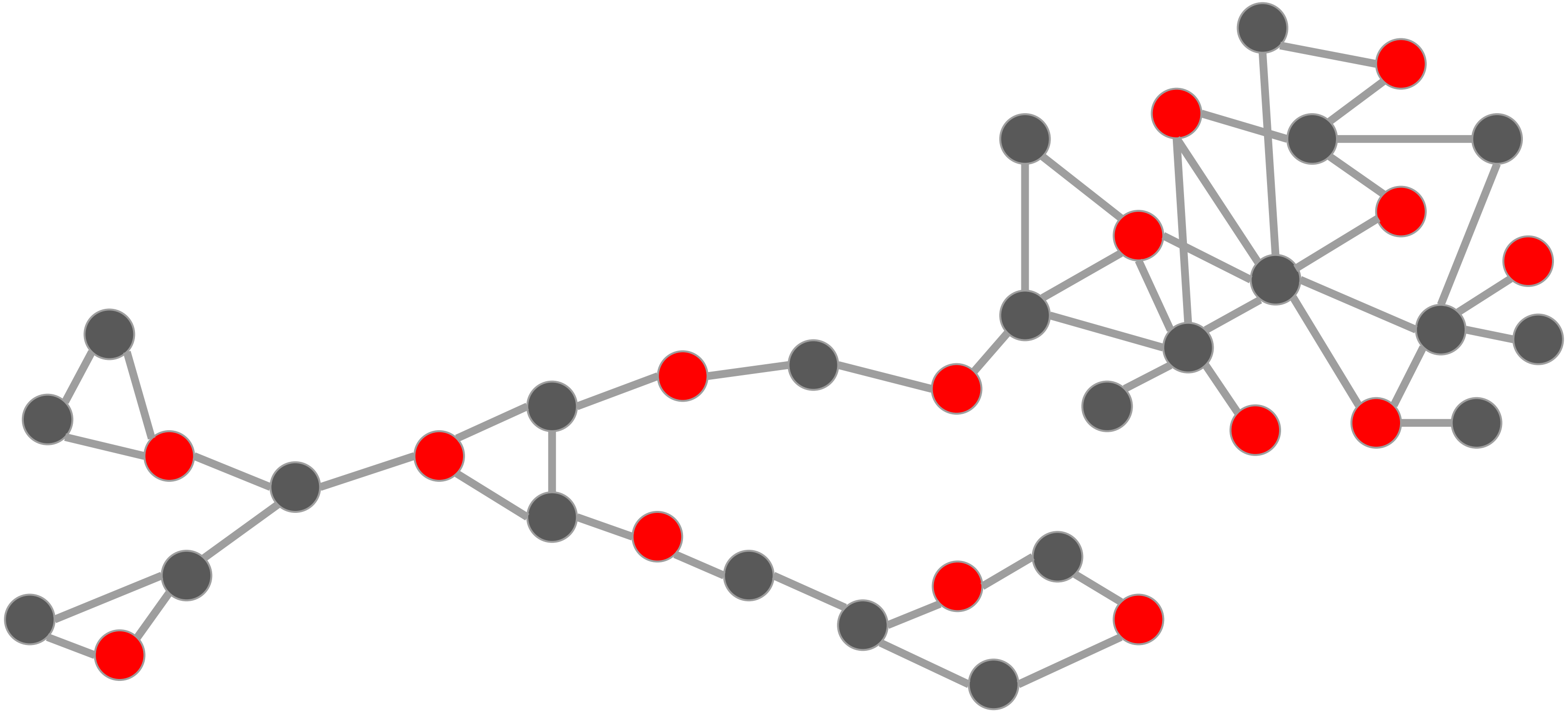
Two setups:

- Understand the complexities of problems “up to polylog(n)”
- Understand the problems with complexities below log(n)

Example: Maximal independent set

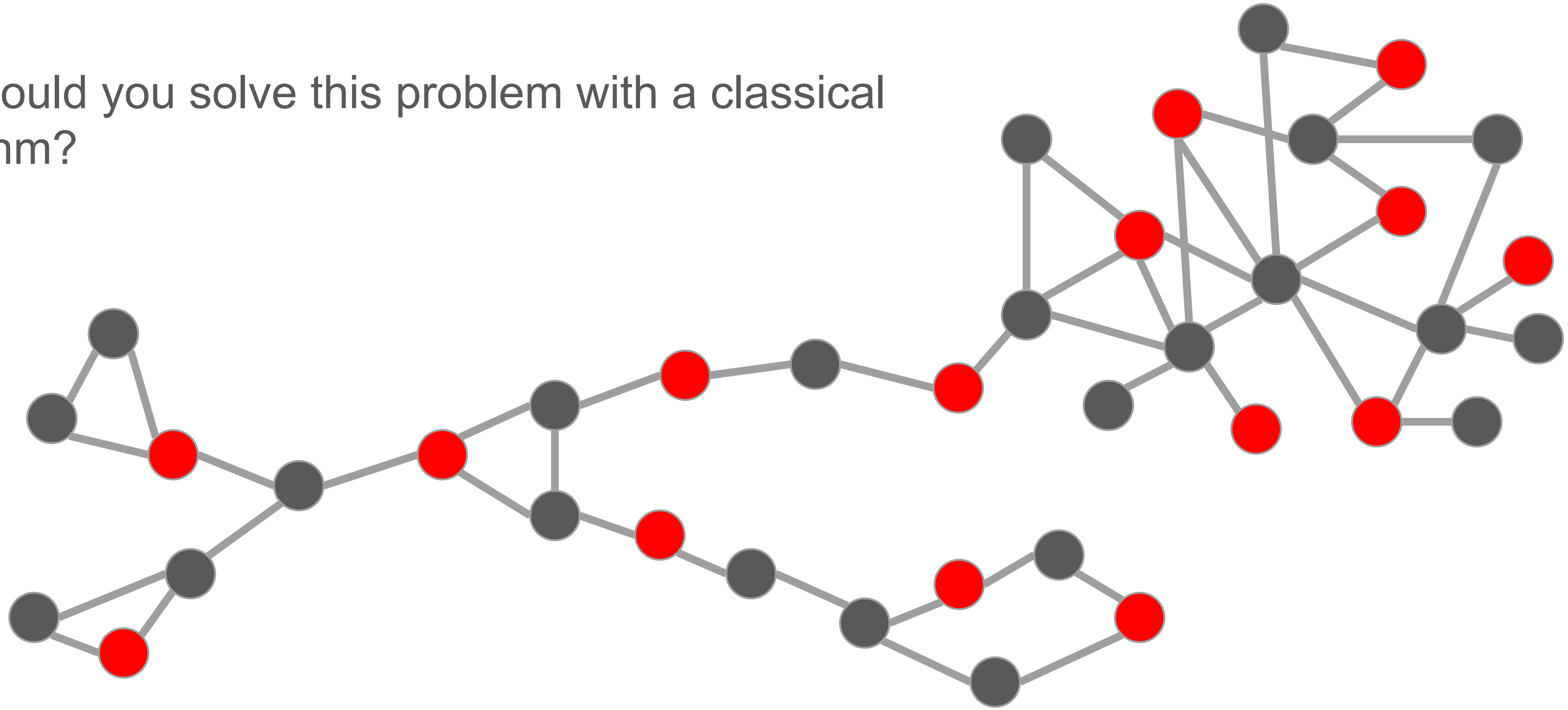


Example: Maximal independent set

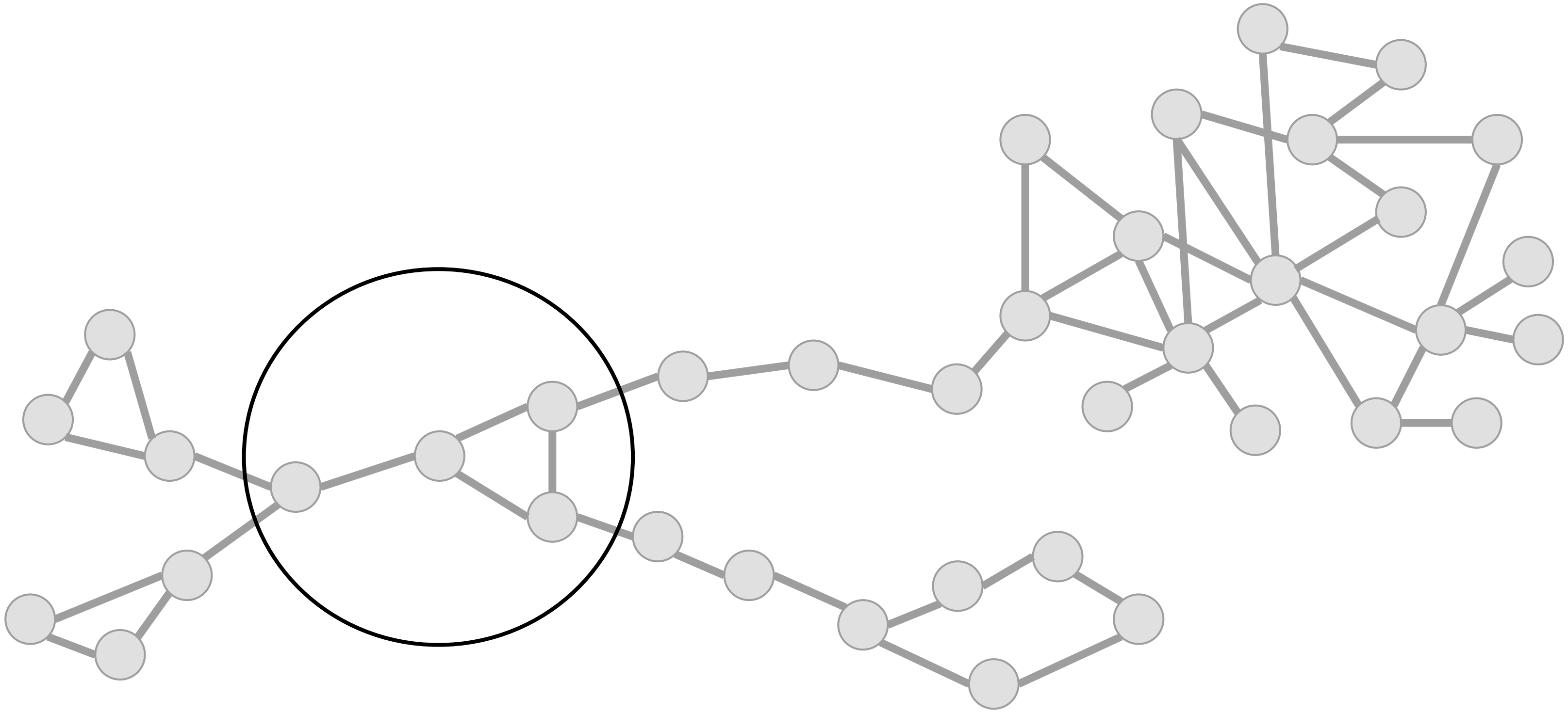


Example: Maximal independent set

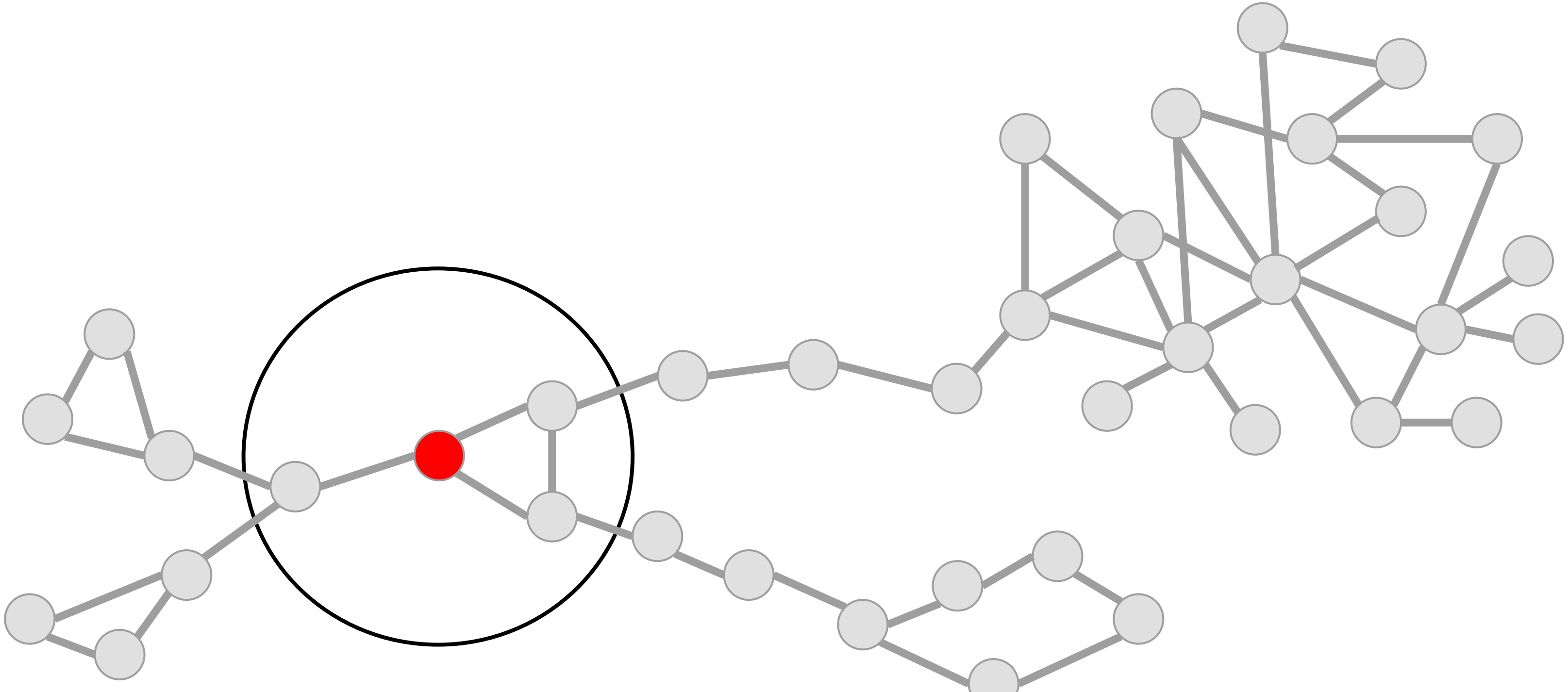
How would you solve this problem with a classical algorithm?



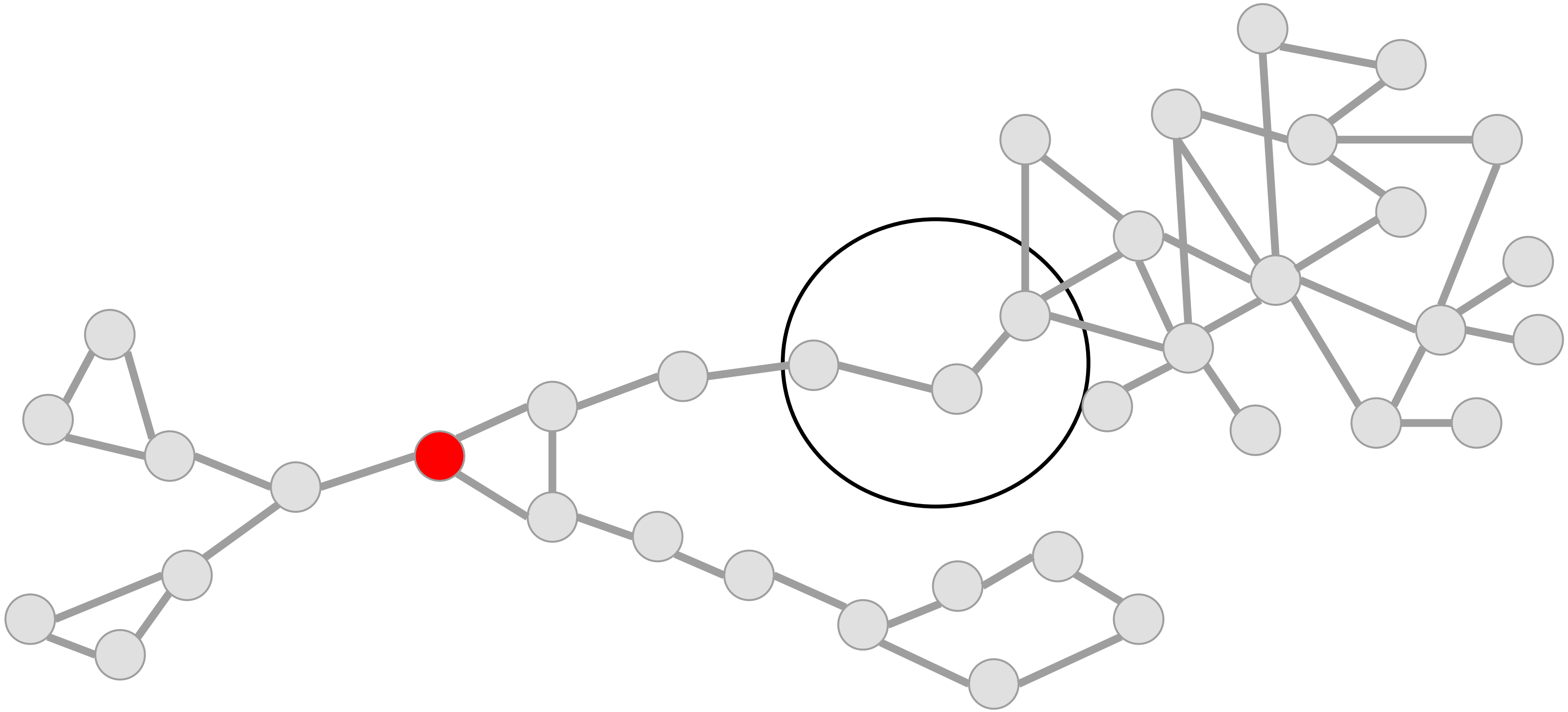
Example: Maximal independent set



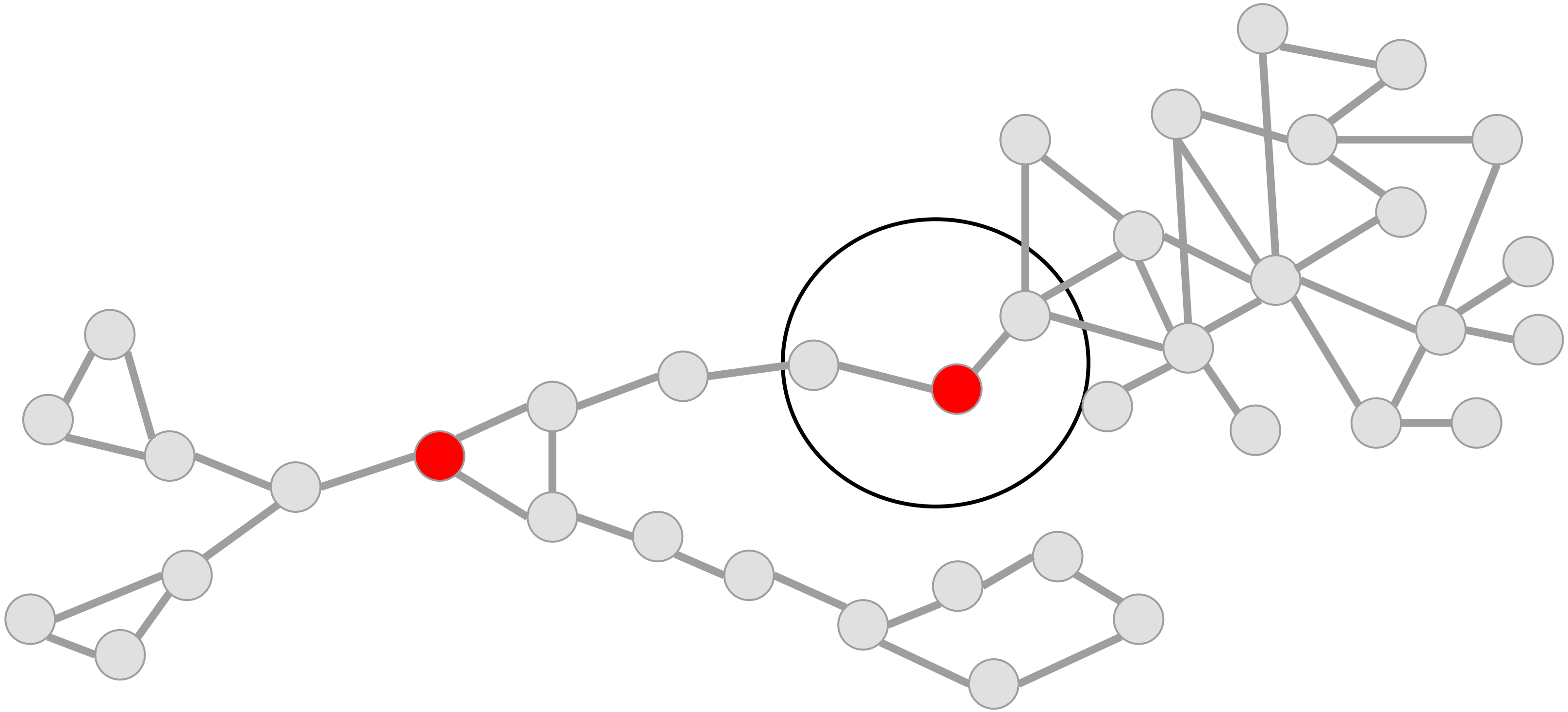
Example: Maximal independent set



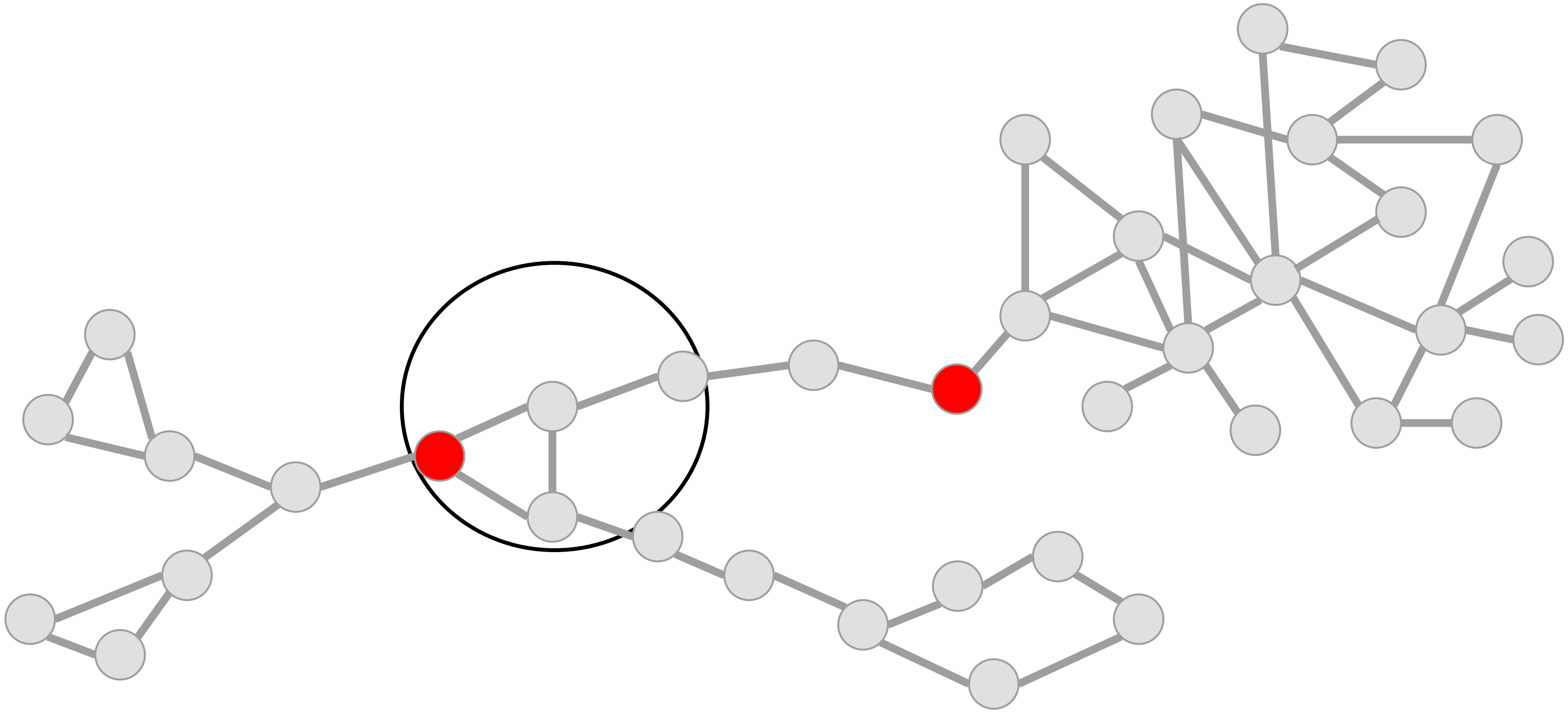
Example: Maximal independent set



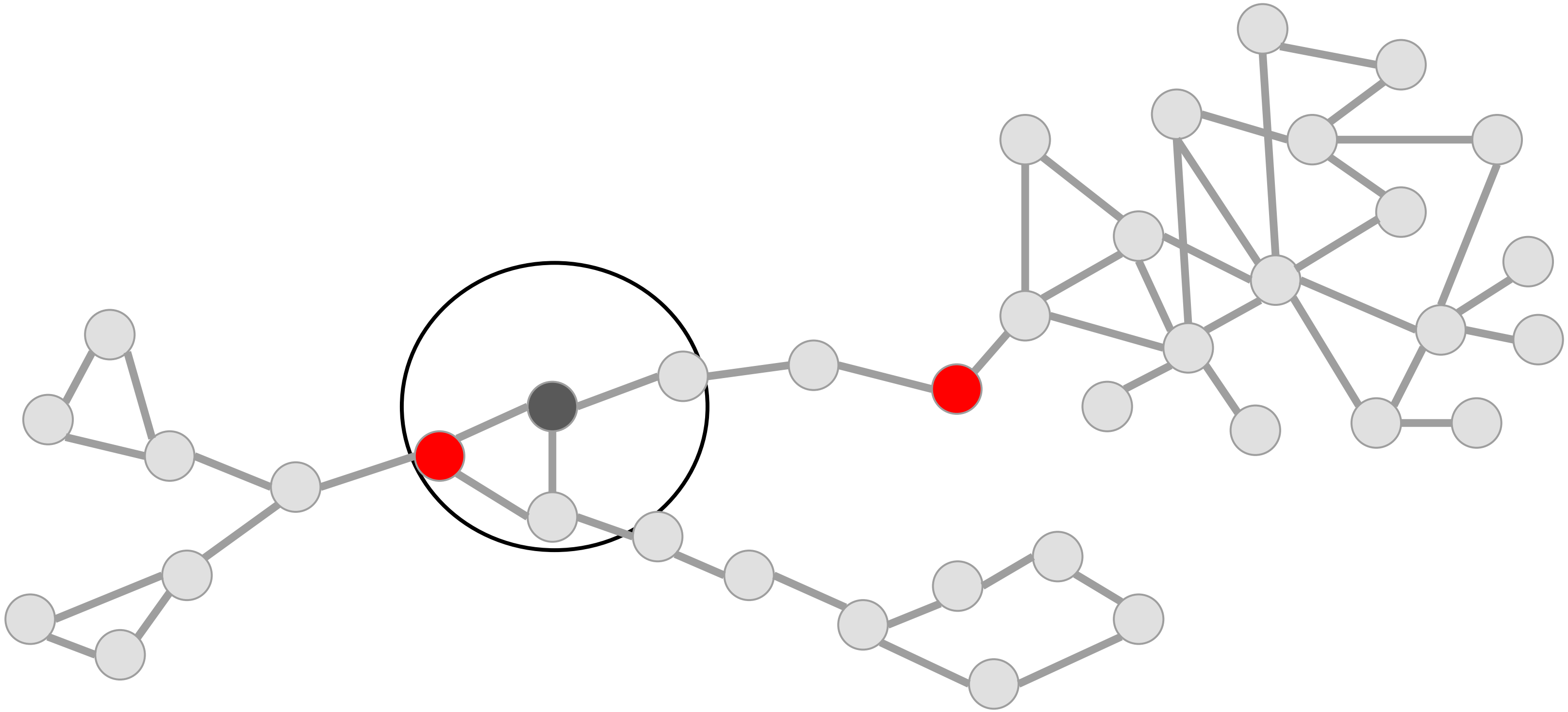
Example: Maximal independent set



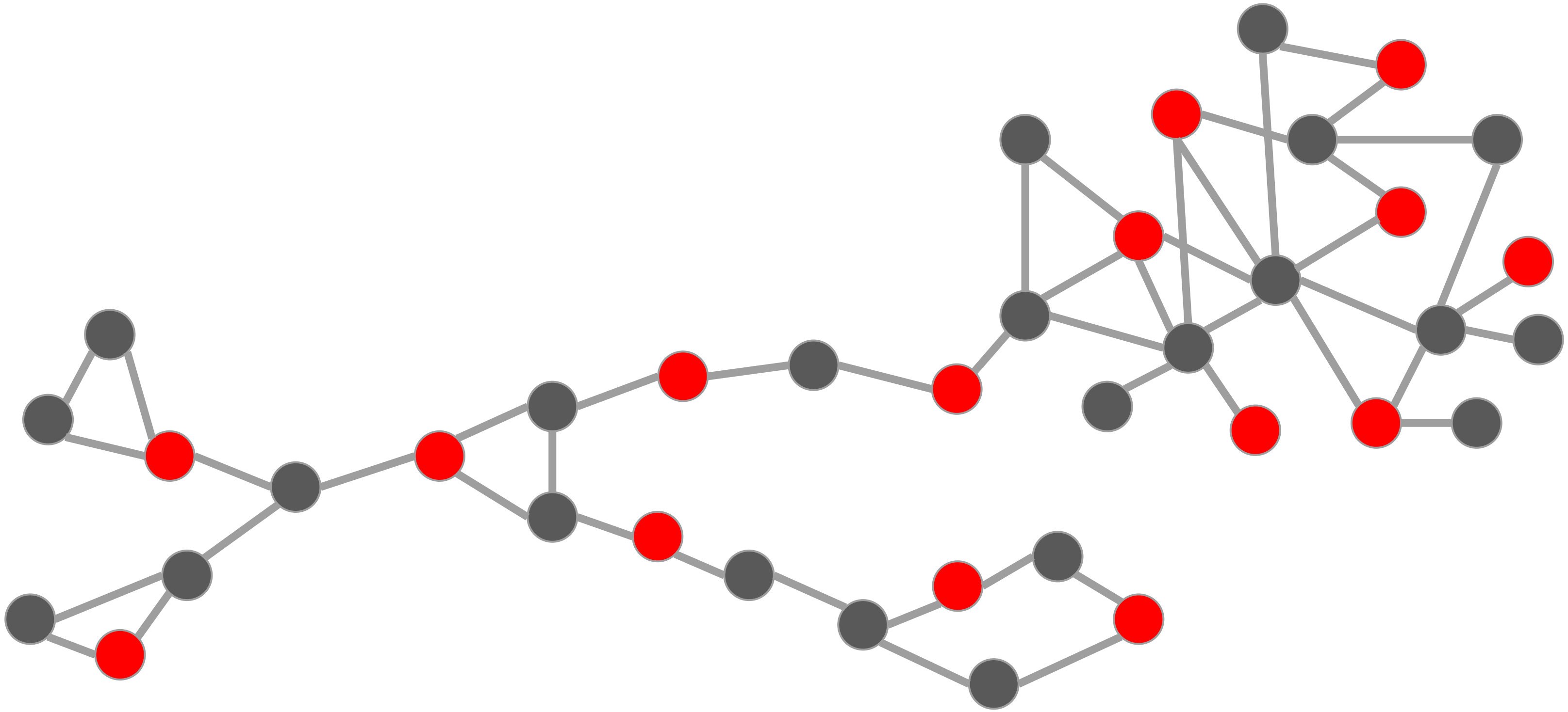
Example: Maximal independent set



Example: Maximal independent set

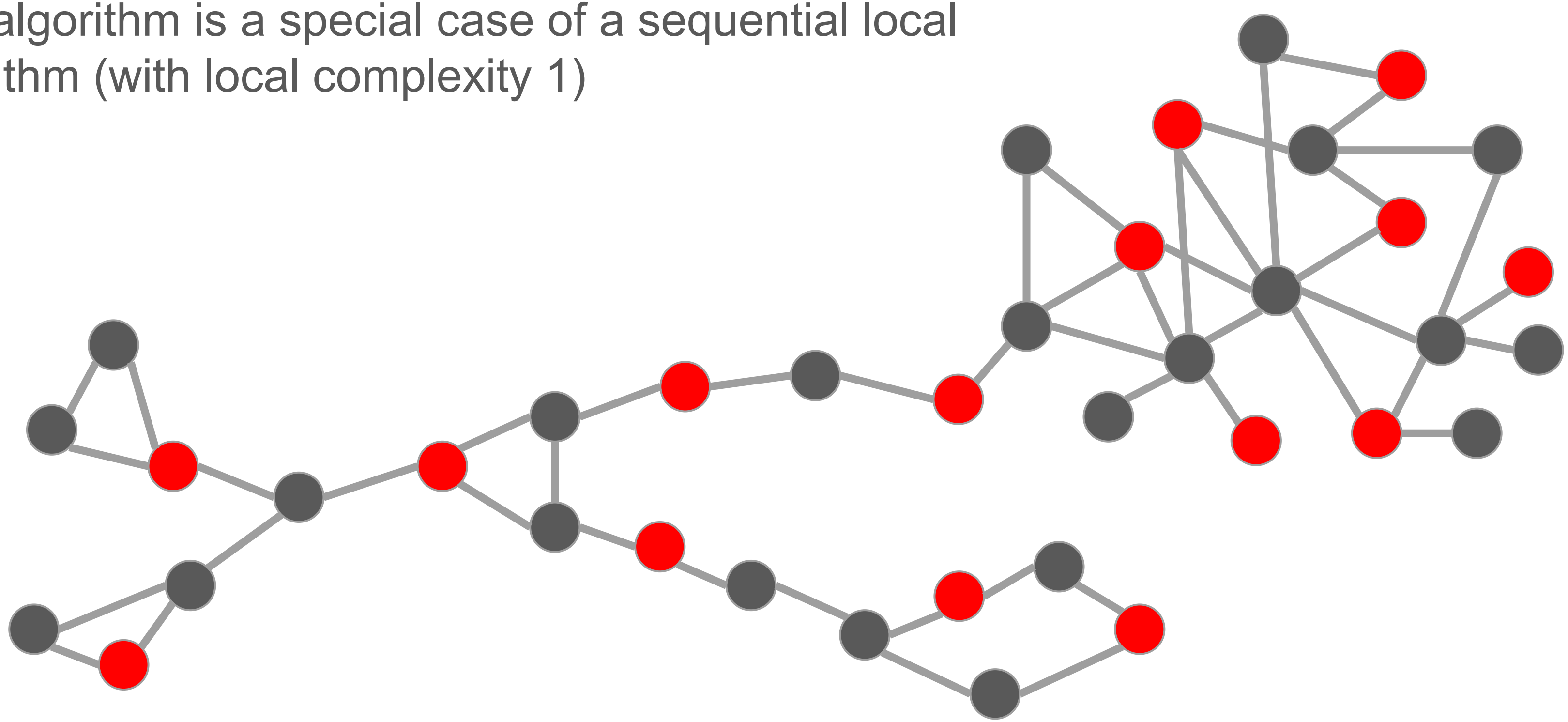


Example: Maximal independent set



Example: Maximal independent set

This algorithm is a special case of a sequential local algorithm (with local complexity 1)



Sequential vs Distributed Locality

Theorem: [Linial, Saks; Ghaffari, Kuhn, Maus; R., Ghaffari]

Every sequential local algorithm with local complexity t(n) can be converted to a local algorithm with round complexity t(n) * polylog(n).

Application: there exists a local algorithm for the maximal independent set problem with polylog(n) rounds.

This thesis:

- an alternative proof of a part of this theorem [R., Haeupler, Grunau]
- applications to parallel algorithms: parallel algorithms for various graph clustering problems [R., Elkin, Grunau, Haeupler] that in turn lead to faster parallel algorithms for e.g. the shortest path problem [R., Grunau, Haeupler, Zuzic, Li]

Sequential vs Distributed Locality

Upshot of the theorem: If we care about complexities up to polylog(n), we have a pretty good understanding of what problems can be solved!

But polylog(n) is maybe not so small…

Can we get some understanding about which problems can be solved in, say, less than log(n) rounds?

Yes, if we make some assumptions!

What about problems with local complexity o(log n)?

Theorem [Naor, Stockmeyer; Chang, Pettie; Chang, Kopelowitz, Pettie; Brandt, Fischer, Hirvonen, Keller, Lempiäinen, Rybicki, Suomela, Uitto; R., Ghaffari; Fischer, Ghaffari] includes a small lie

Any local problem with local complexity o(log n) on bounded-degree graphs falls into one of three classes based on its local complexity:

[Trivial problems]	O(1) local complexity
[Basic symmetry-breaking problems]	Θ(log*n) local complexity

What about problems with local complexity o(log n)?

Theorem [Naor, Stockmeyer; Chang, Pettie; Chang, Kopelowitz, Pettie; Brandt, Fischer, Hirvonen, Keller, Lempiäinen, Rybicki, Suomela, Uitto; R., Ghaffari; Fischer, Ghaffari]

Any local problem with local complexity o(log n) on bounded-degree graphs falls into one of three classes based on its local complexity:

[Trivial problems] O(1) local complexity

[Basic symmetry-breaking problems] $\Omega(\log \log^* n)$ – $O(\log^* n)$ local complexity

This thesis:
Just $\Theta(\log^* n)$ on trees & grids [Brandt, Grunau, R.]

What about problems with local complexity o(log n)?

Theorem [Naor, Stockmeyer; Chang, Pettie; Chang, Kopelowitz, Pettie; Brandt, Fischer, Hirvonen, Keller, Lempiäinen, Rybicki, Suomela, Uitto; R., Ghaffari; Fischer, Ghaffari] includes a small lie

Any local problem with local complexity o(log n) on bounded-degree graphs falls into one of three classes based on its local complexity:

[Trivial problems]	O(1) local complexity
[Basic symmetry-breaking problems]	Θ(log*n) local complexity
[Instances of Lovász-local-lemma]	Θ(polyloglog(n)) rand. & Θ(polylog(n)) det. complexity

What about problems with local complexity o(log n)?

Theorem [Naor, Stockmeyer; Chang, Pettie; Chang, Kopelowitz, Pettie; Brandt, Fischer, Hirvonen, Keller, Lempiäinen, Rybicki, Suomela, Uitto; R., Ghaffari; Fischer, Ghaffari] includes a small lie

Any local problem with local complexity o(log n) on bounded-degree graphs falls into one of three classes based on its local complexity:

[Trivial problems]	O(1) local complexity
[Basic symmetry-breaking problems]	Θ(log*n) local complexity
[Instances of Lovász-local-lemma]	Θ(polyloglog(n)) rand. & Θ(polylog(n)) det. complexity

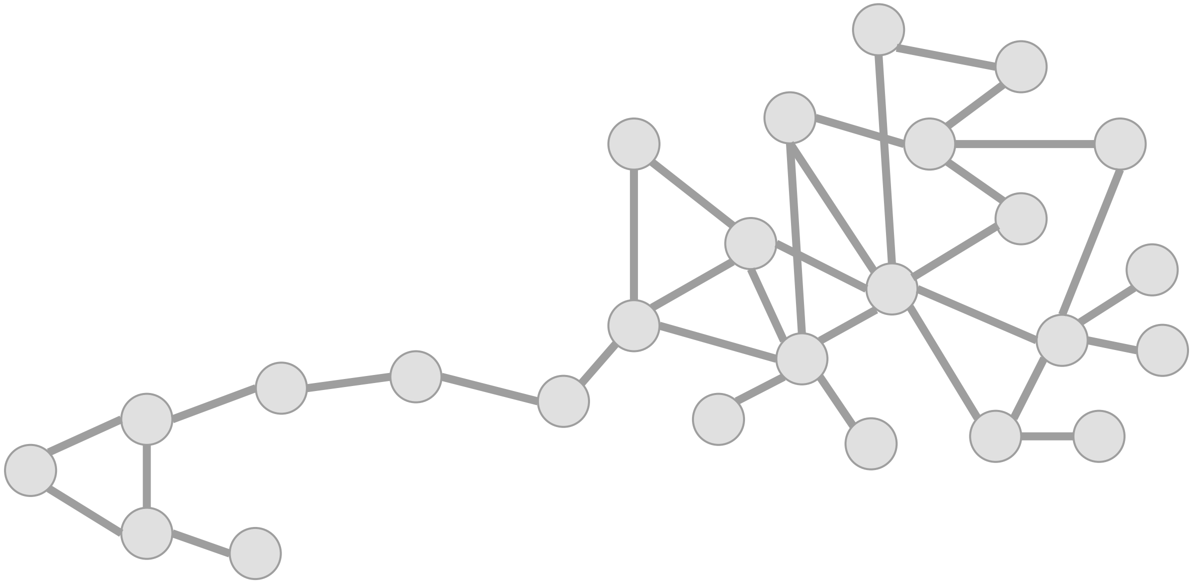
Can we extend this theorem beyond local algorithms?

The rest of the talk: Two extensions

- sublinear algorithms
- measurable combinatorics

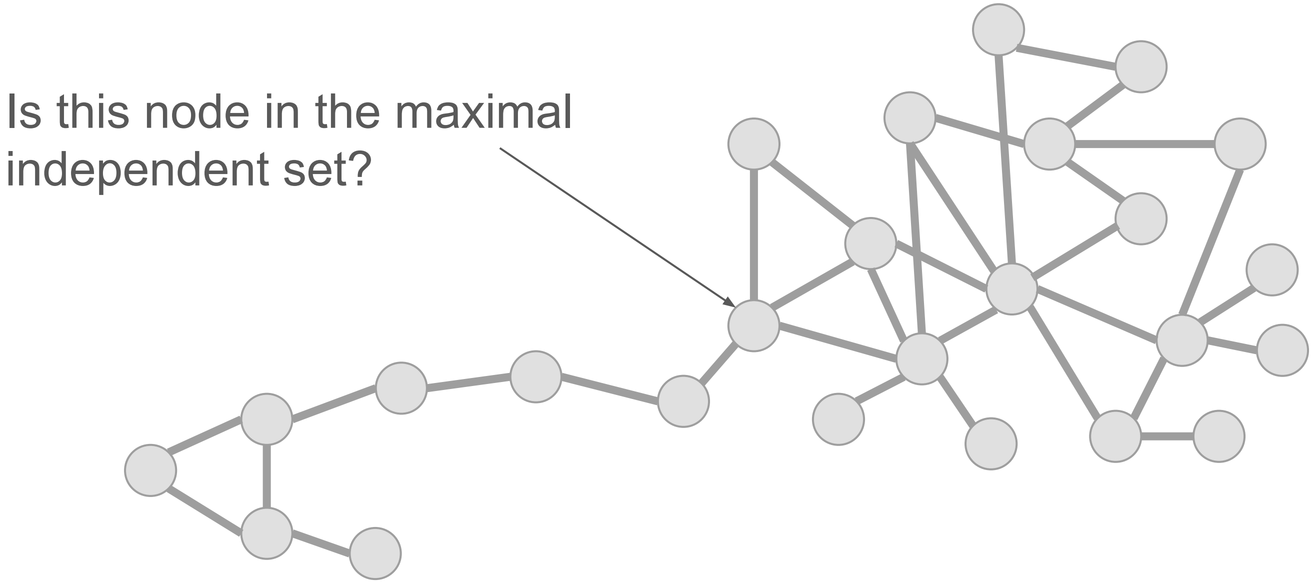
Extension #1: Sublinear algorithms

Local computation algorithms (model of graph sublinear algorithms)



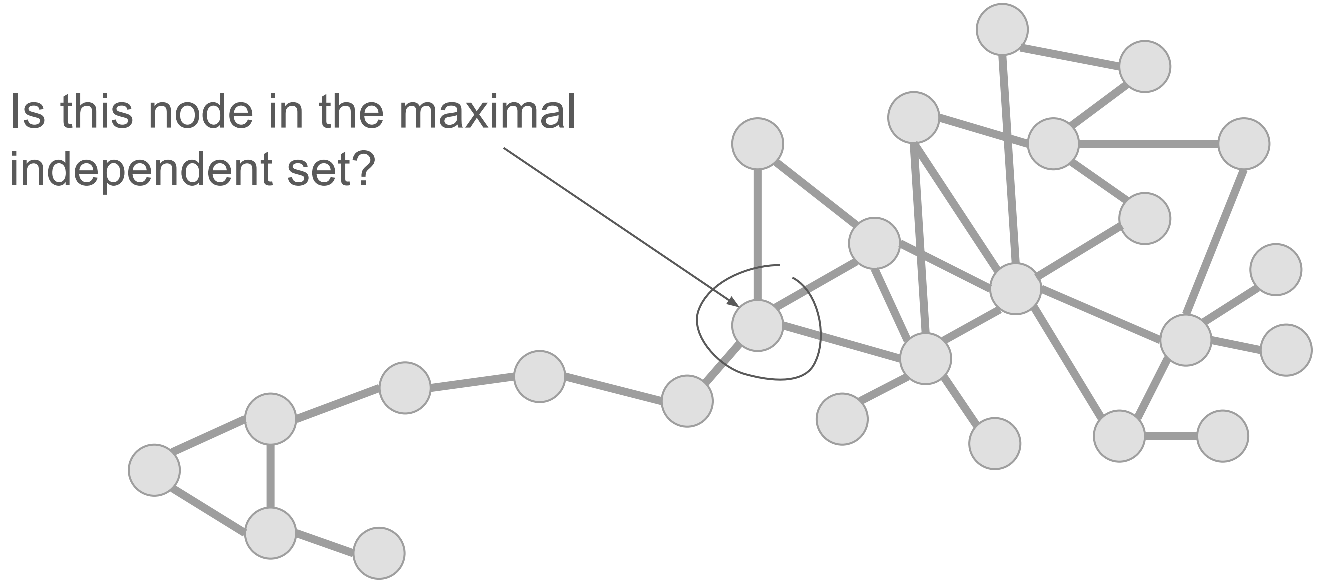
Extension #1: Sublinear algorithms

Local computation algorithms (model of graph sublinear algorithms)



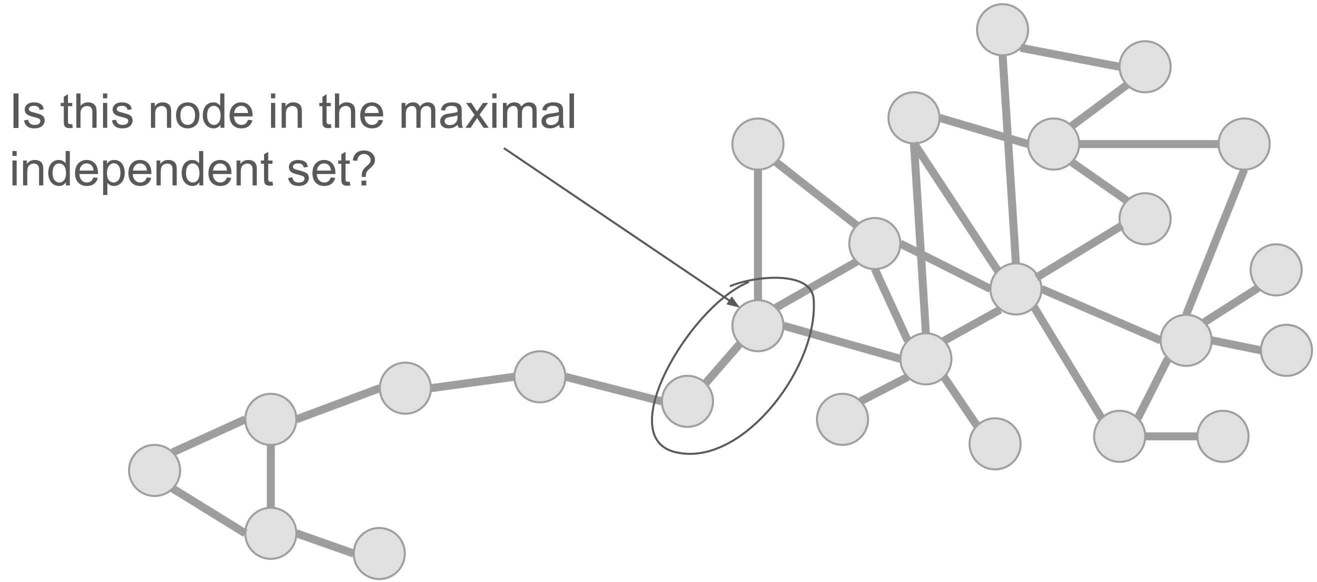
Extension #1: Sublinear algorithms

Local computation algorithms (model of graph sublinear algorithms)



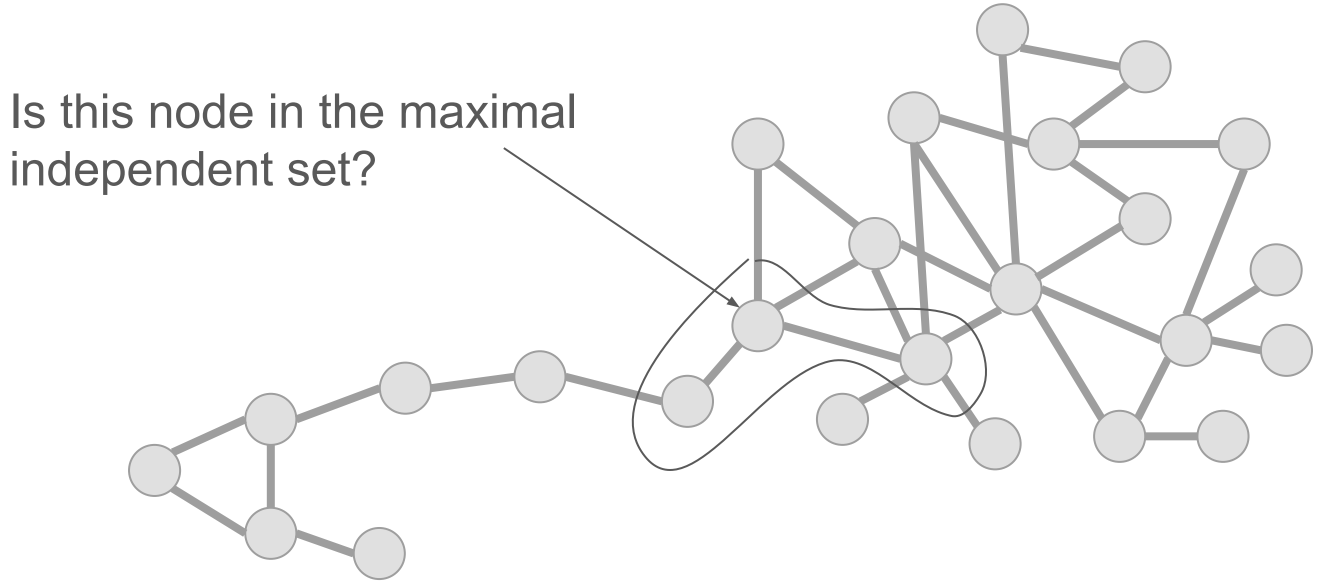
Extension #1: Sublinear algorithms

Local computation algorithms (model of graph sublinear algorithms)



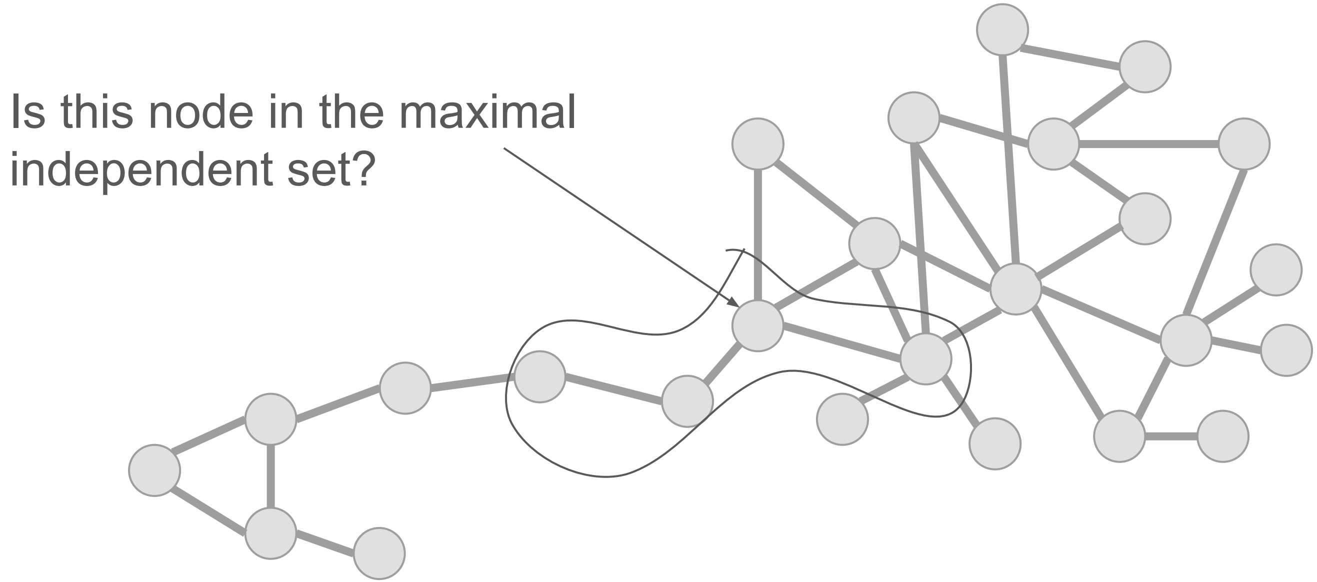
Extension #1: Sublinear algorithms

Local computation algorithms (model of graph sublinear algorithms)



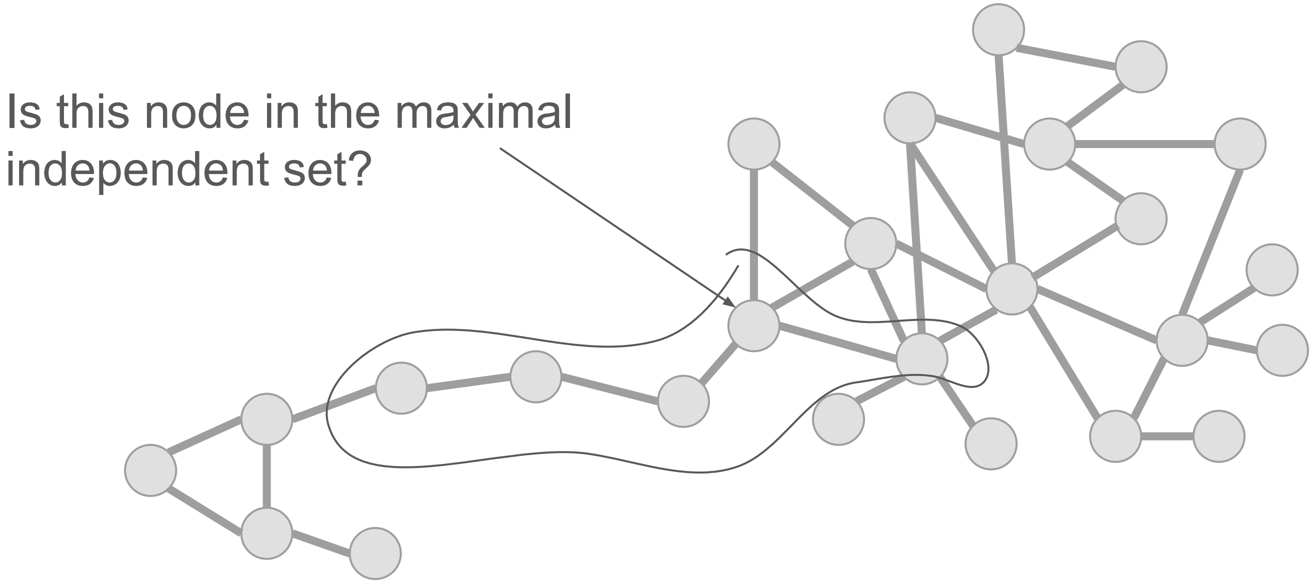
Extension #1: Sublinear algorithms

Local computation algorithms (model of graph sublinear algorithms)



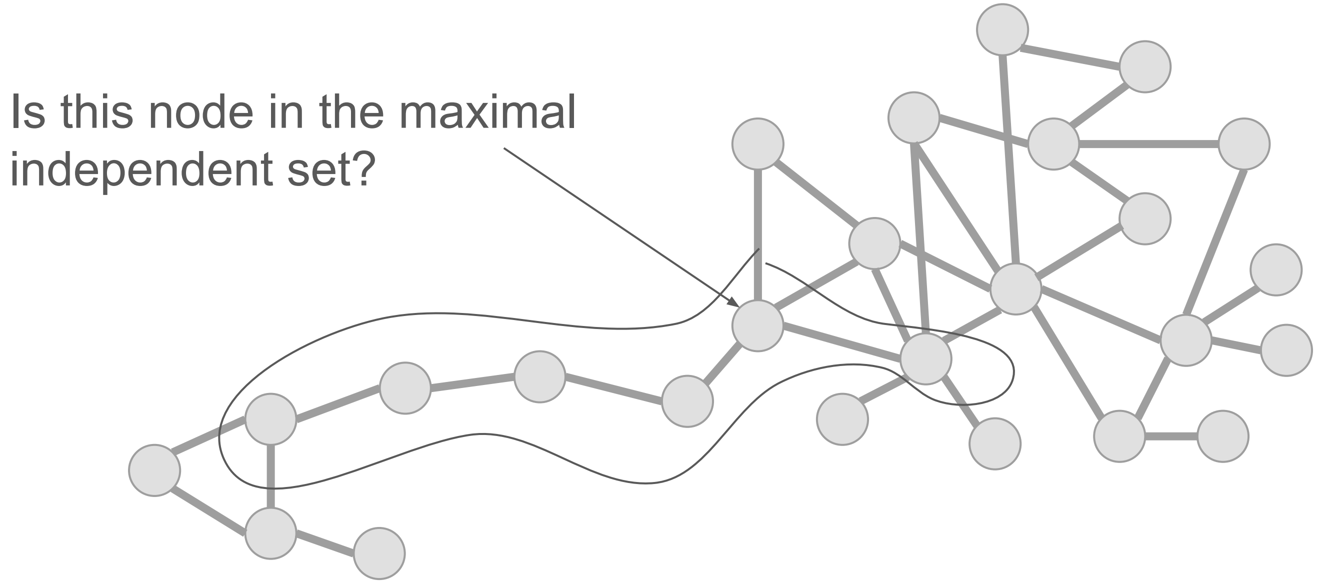
Extension #1: Sublinear algorithms

Local computation algorithms (model of graph sublinear algorithms)



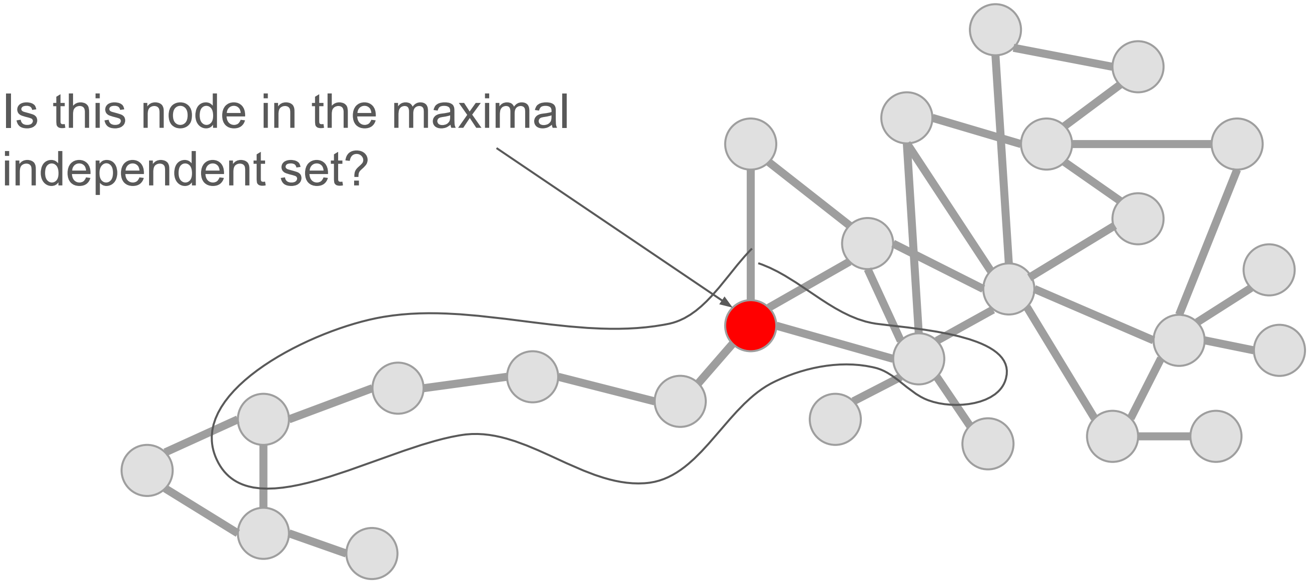
Extension #1: Sublinear algorithms

Local computation algorithms (model of graph sublinear algorithms)



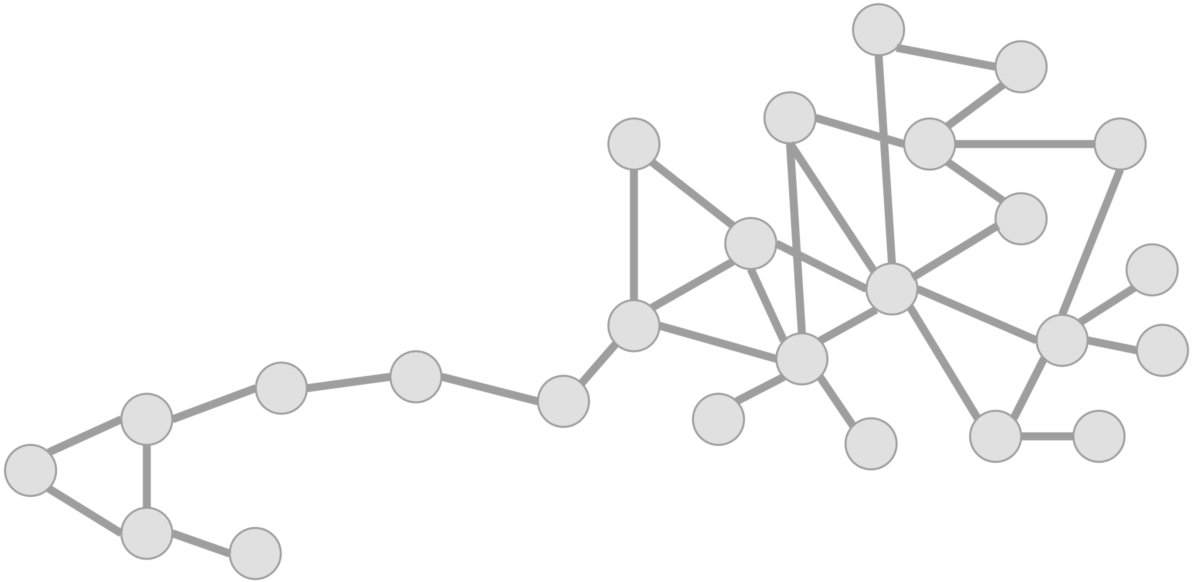
Extension #1: Sublinear algorithms

Local computation algorithms (model of graph sublinear algorithms)



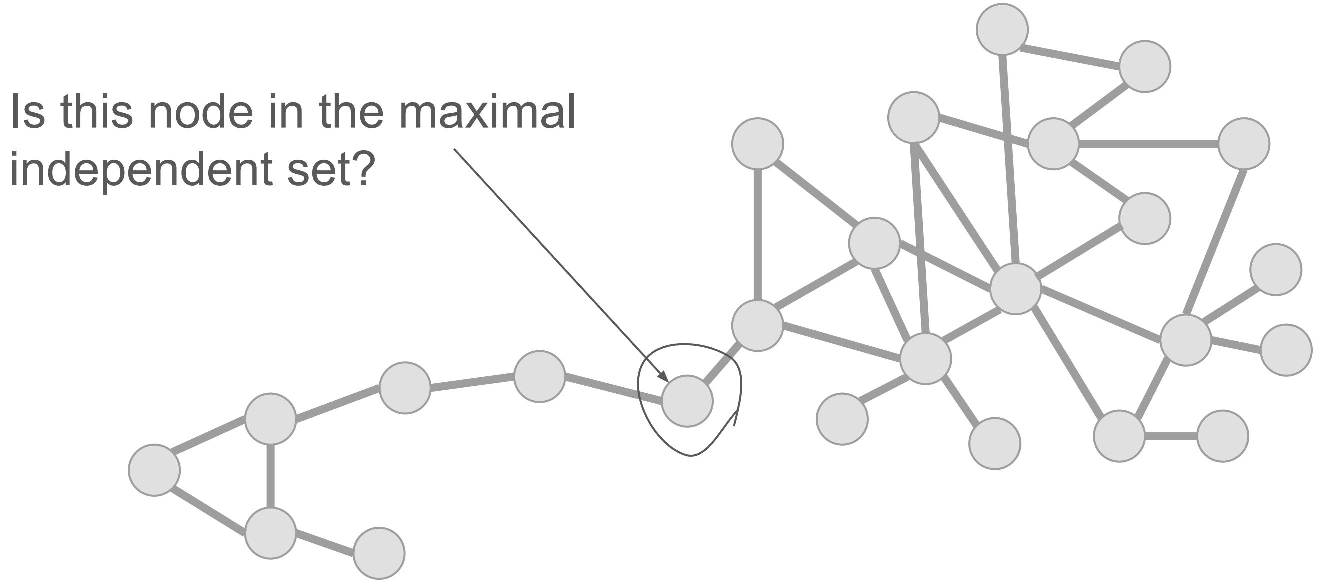
Extension #1: Sublinear algorithms

Local computation algorithms (model of graph sublinear algorithms)



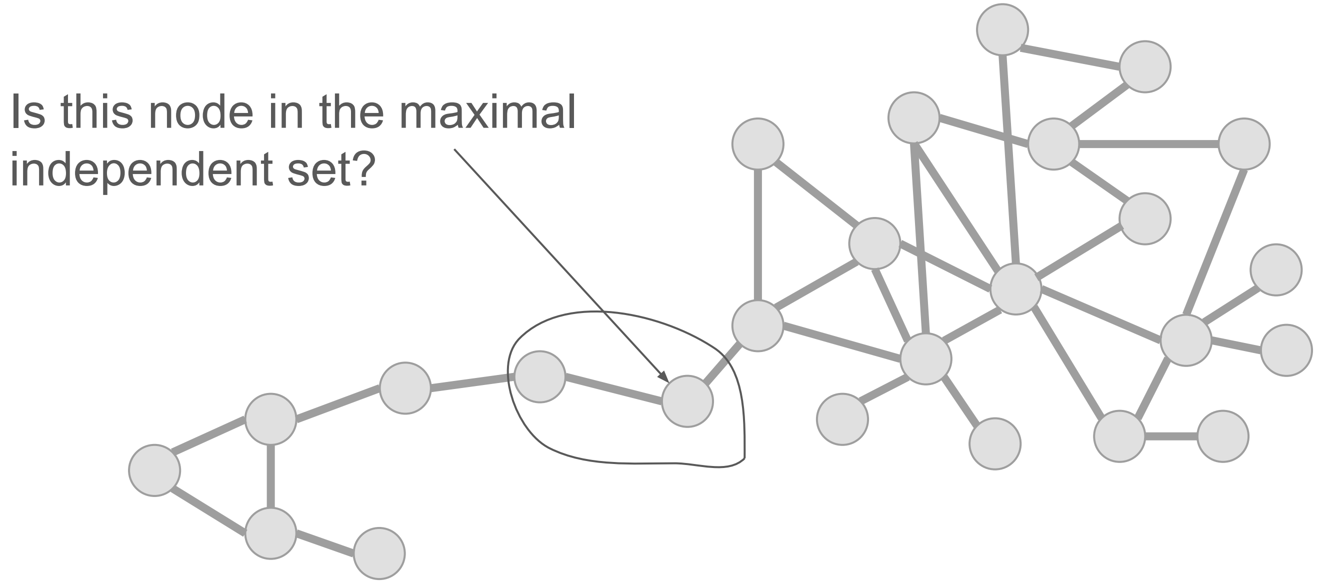
Extension #1: Sublinear algorithms

Local computation algorithms (model of graph sublinear algorithms)



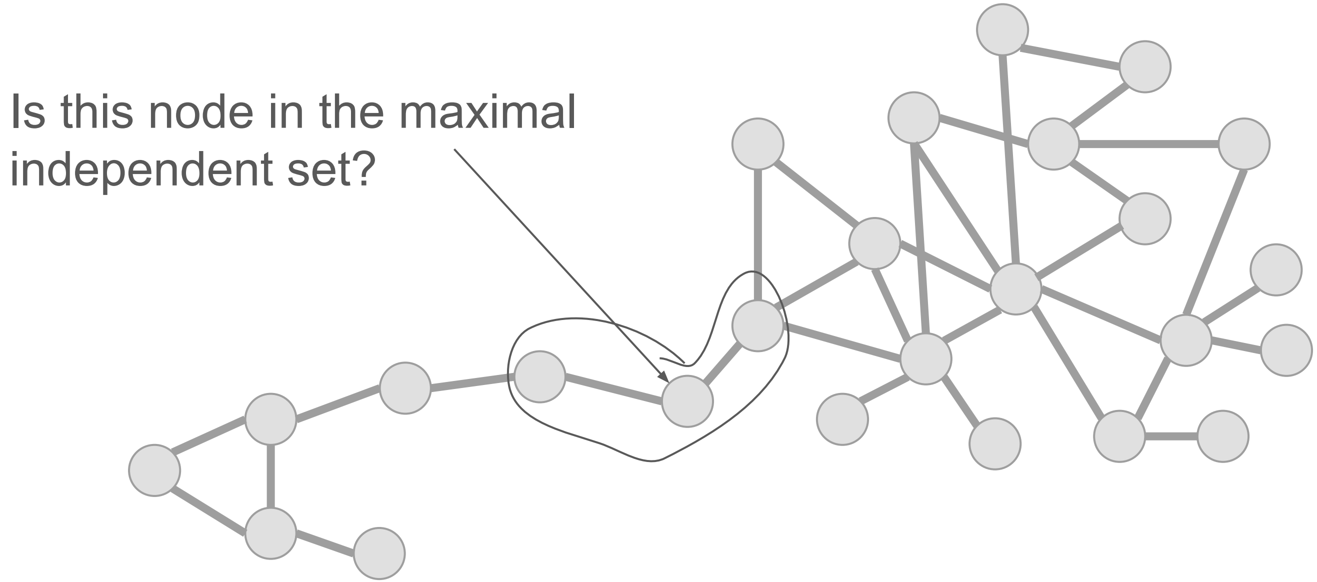
Extension #1: Sublinear algorithms

Local computation algorithms (model of graph sublinear algorithms)



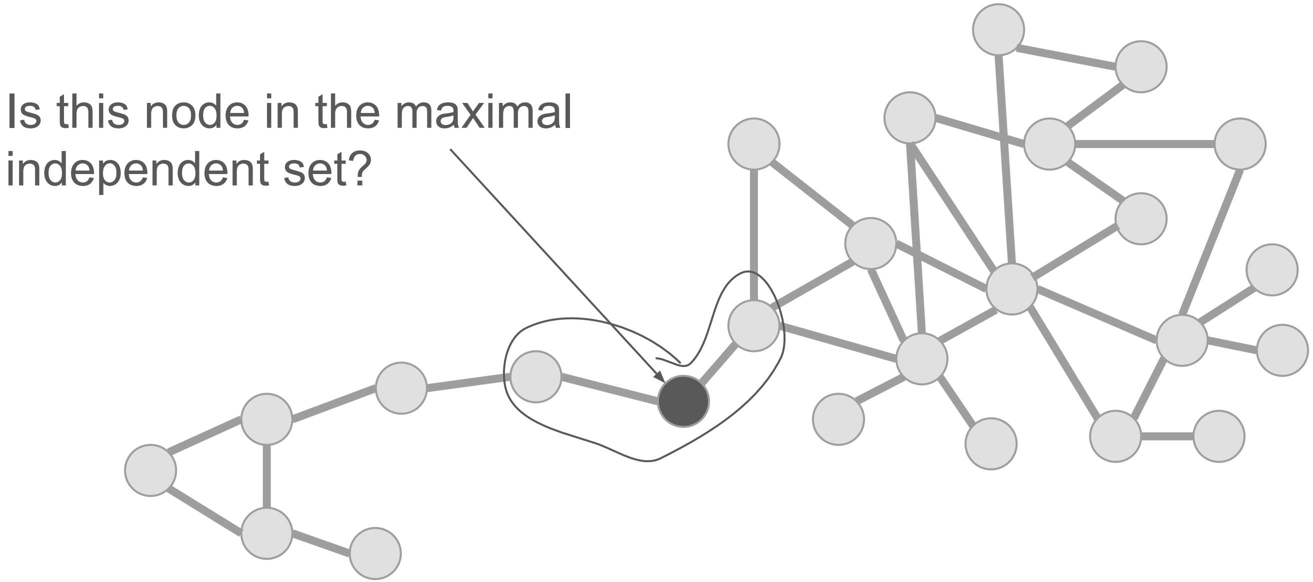
Extension #1: Sublinear algorithms

Local computation algorithms (model of graph sublinear algorithms)



Extension #1: Sublinear algorithms

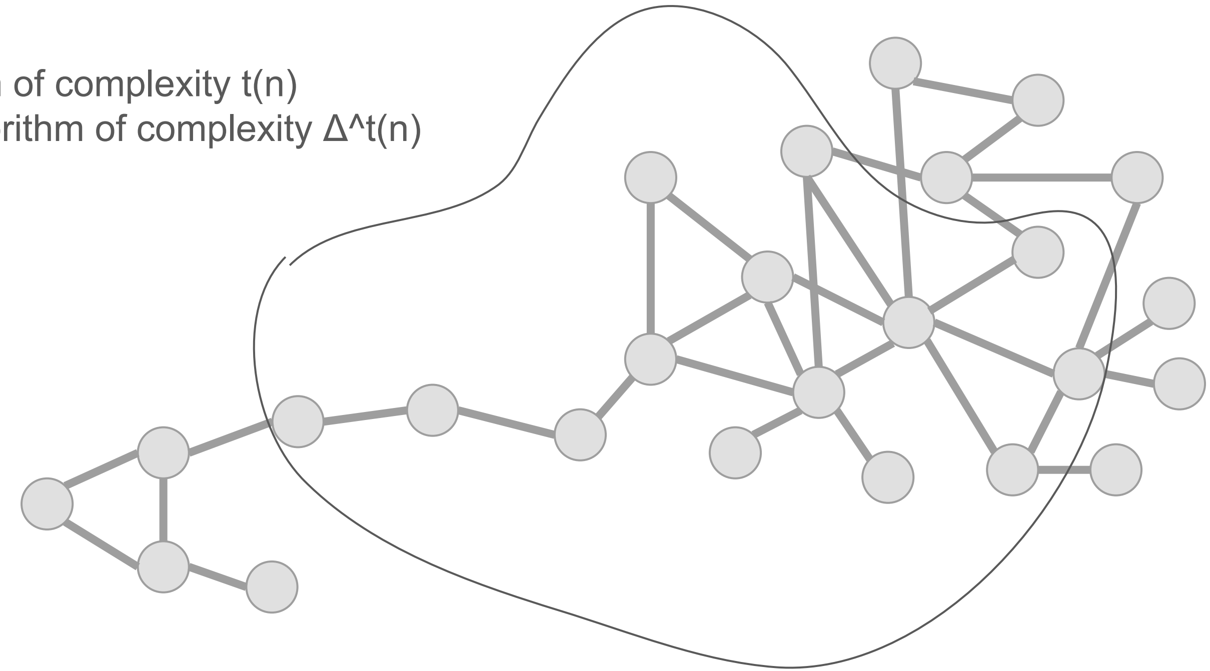
Local computation algorithms (model of graph sublinear algorithms)



Extension #1: Sublinear algorithms

Local computation algorithms (model of graph sublinear algorithms)

[Parnas, Ron] Local algorithm of complexity t(n)
=> local computation algorithm of complexity Δ^t(n)



Extension #1: Sublinear algorithms

This thesis: Adding local computation algorithms to the picture:

Theorem:[Even, Medina, Ron; Rosenbaum, Suomela; Grunau, R., Brandt; Brandt, Grunau, R.]includes a small lie We can extend the three local complexity classes to local computation algorithms as follows:

	Local complexity	Local computation algorithms
[Trivial problems]	O(1)	O(1)
[Basic symmetry-breaking problems]	$\Theta(\log^* n)$	$\Theta(\log^* n)$
[Instances of Lovász-local-lemma]	$\Theta(\text{polyloglog}(n))$ / $\Theta(\text{polylog}(n))$	$\Theta(\text{polylog } n)$ / $O(n)$

Extension #2: Measurable combinatorics

[Bernshteyn]: There is a close connection between local algorithms and measurable combinatorics.

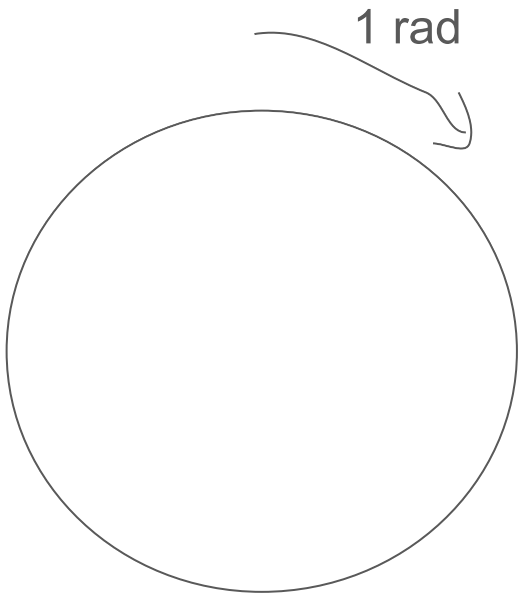
A riddle:

Color every point of a unit cycle such that

1) no two vertices 1 radian apart have the same color
2) vertices of the same color are finite union of intervals

How many colors do you need?

This is an example of solving a local problem on measurable graphs



Extension #2: Measurable combinatorics

[Bernshteyn]: There is a close connection between local algorithms and measurable combinatorics.

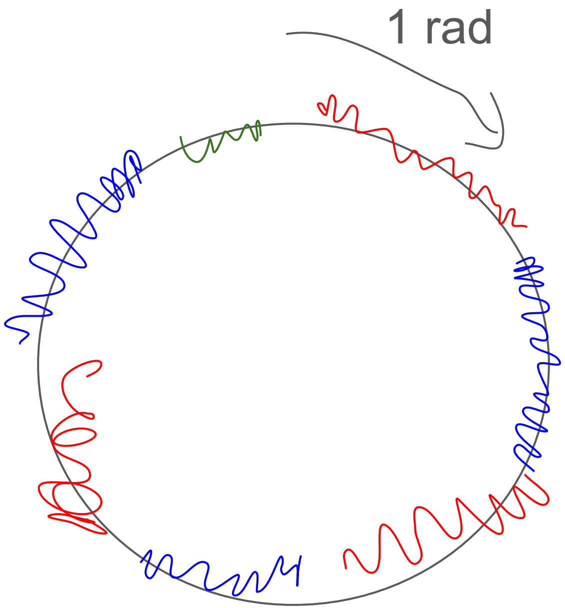
A riddle:

Color every point of a unit cycle such that

1) no two vertices 1 radian apart have the same color
2) vertices of the same color are finite union of intervals

How many colors do you need?

This is an example of solving a local problem on measurable graphs

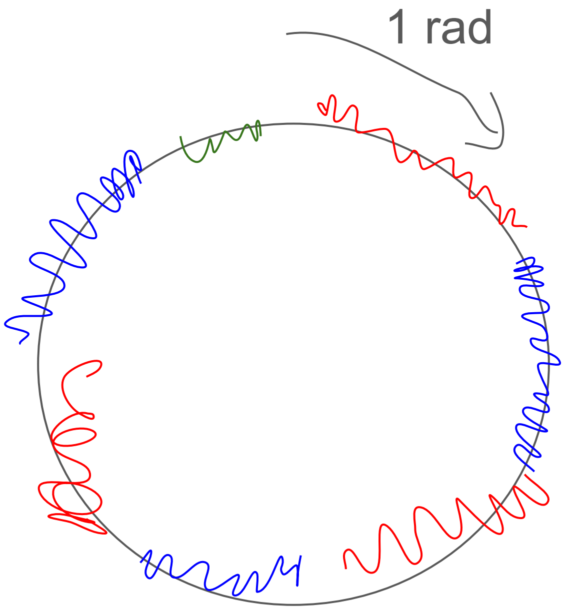


Extension #2: Measurable combinatorics

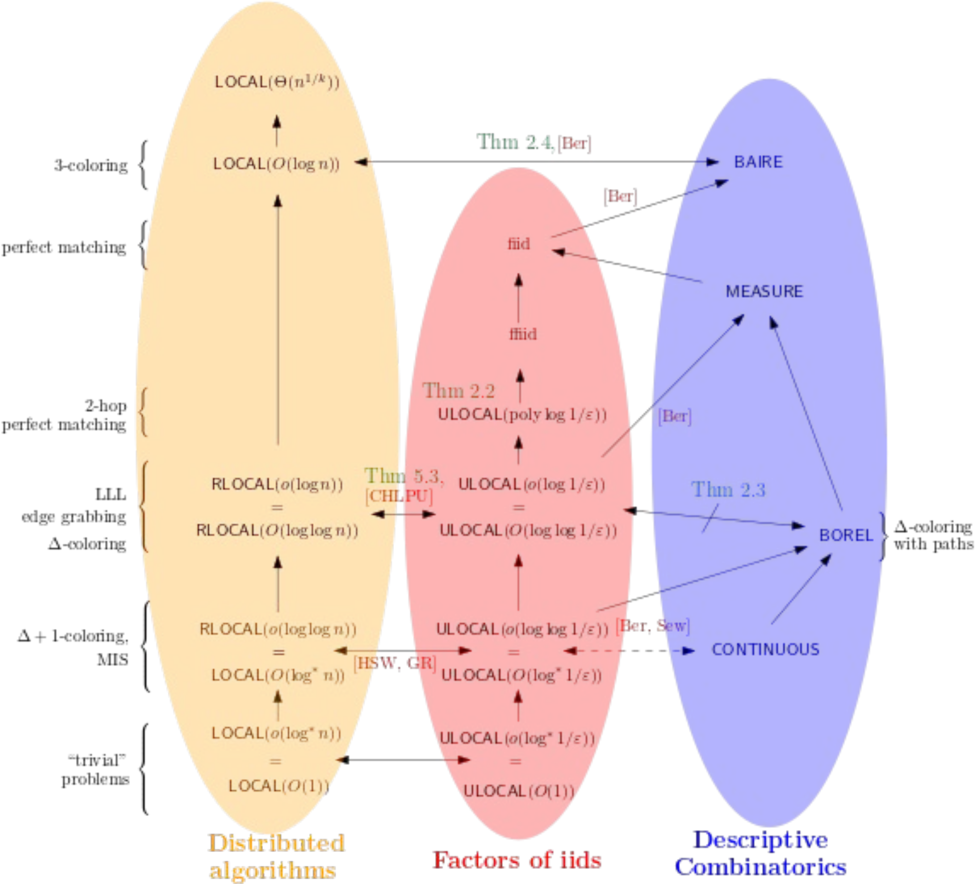
In general, measurable combinatorics is a combinatorics of uncountable graphs equipped with a measure.

[Bernhsteyn]: Constructions by local algorithms often automatically imply results in measurable combinatorics.

Fun Fact: [Grebík, R.]:
One can solve a local problem on the circle if and only if there is a o(n)-round local algorithm for the problem on path graphs.

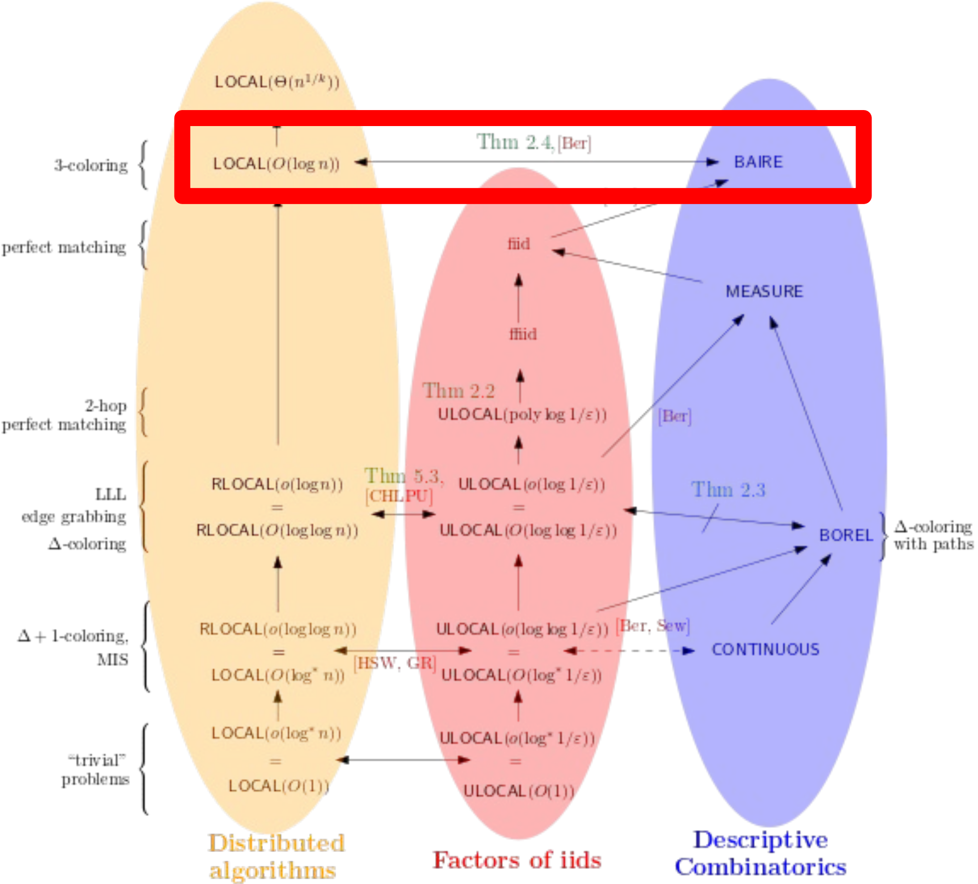


Extension #2: Measurable combinatorics



This thesis: investigation of the connections between local algorithms and descriptive combinatorics on regular trees

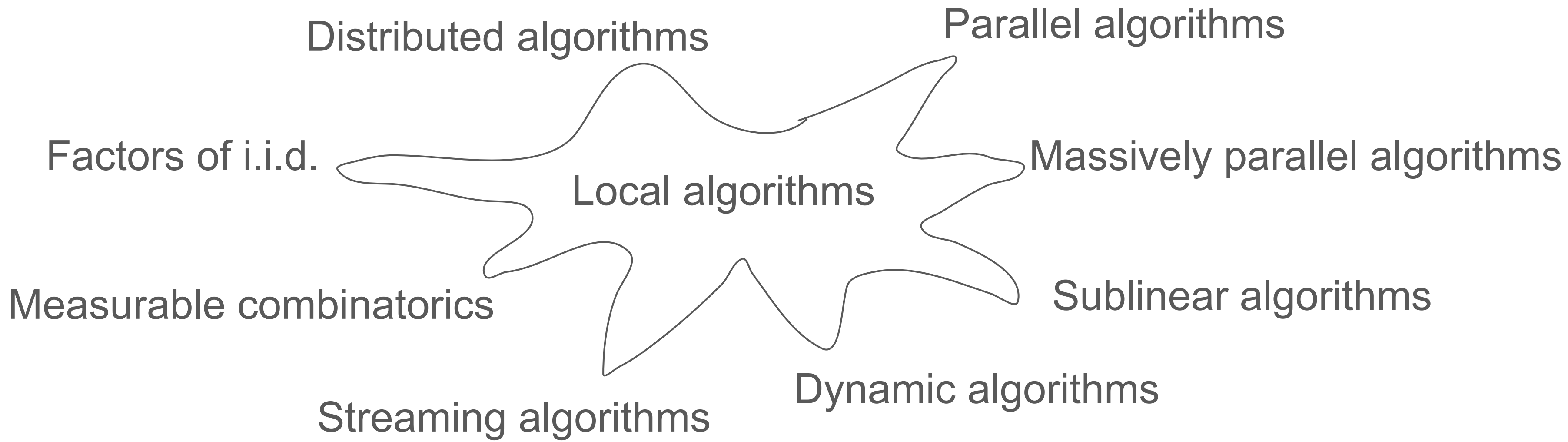
Extension #2: Measurable combinatorics



This thesis: investigation of the connections between local algorithms and descriptive combinatorics on regular trees

My guess for what "Baire solution" means:
For any measurable graph where each component is a regular tree and any compatible reasonable topology on it, there exists a solution to the problem that is correct on all vertices except for a set that is a countable union of sets whose closure has empty interior.

A view of local complexity




Thank you!