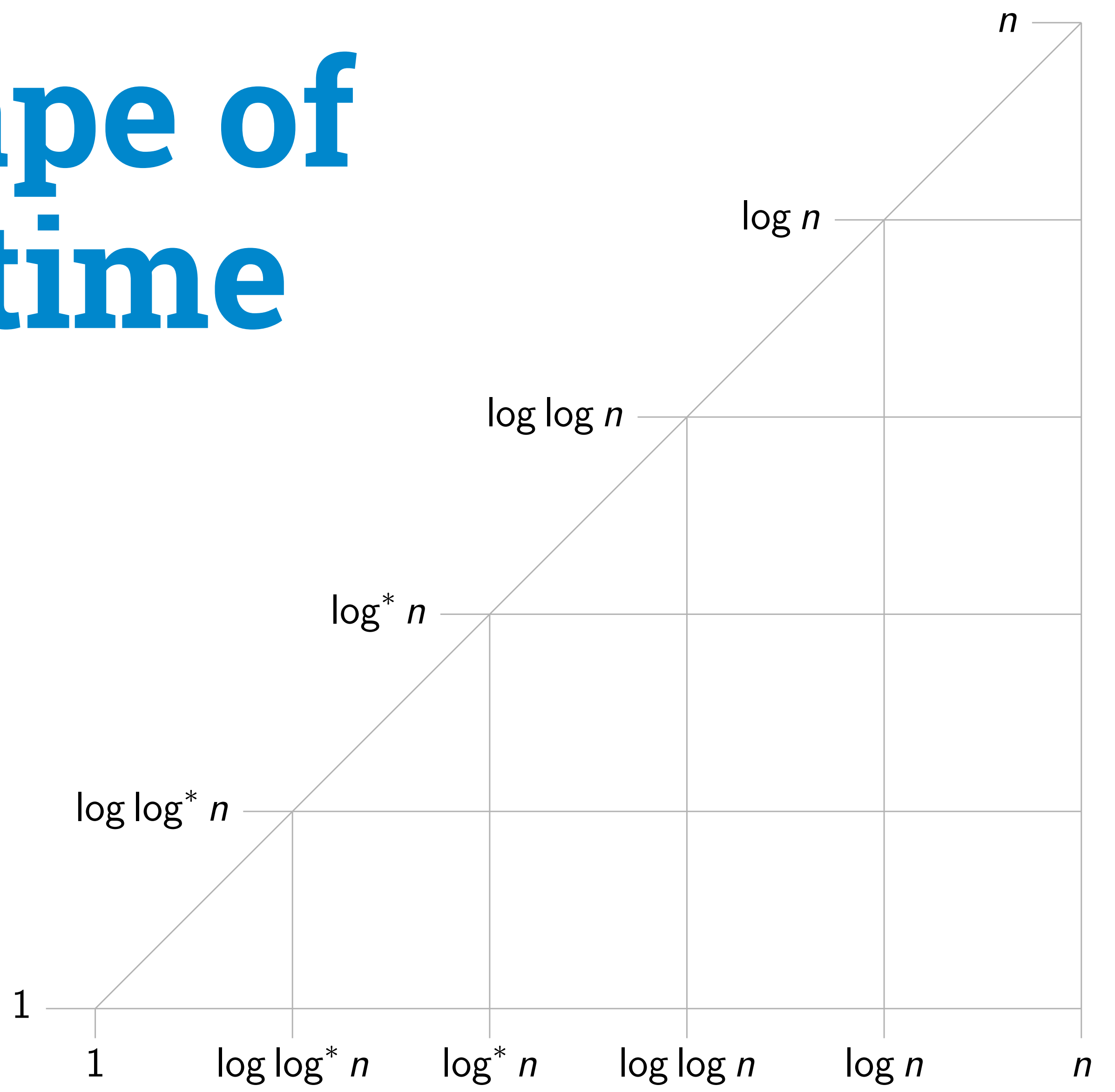
The landscape of distributed time complexity

Jukka Suomela
Aalto University, Finland



Graph problems

- Today's topic: ***graph problems***
- Lots of research activity related to ***individual graph problems***
  - this one can be solved in $O(nm)$ time, this one is NP-hard…
- What can we say about entire ***families of graph problems***?

Classifying graph problems

- **Combinatorial classifications**
  - e.g. **hereditary**, **monotone**, **minor-closed** graph properties
- **Logical classifications**
  - e.g. expressible by **existential second-order formulas**
- **Computational classifications**
  - deterministic time: e.g. in **P**
  - nondeterministic time: e.g. in **NP**
  - parallel time: e.g. in **NC**
  - space: e.g. in **PSPACE** …

Classifying graph problems

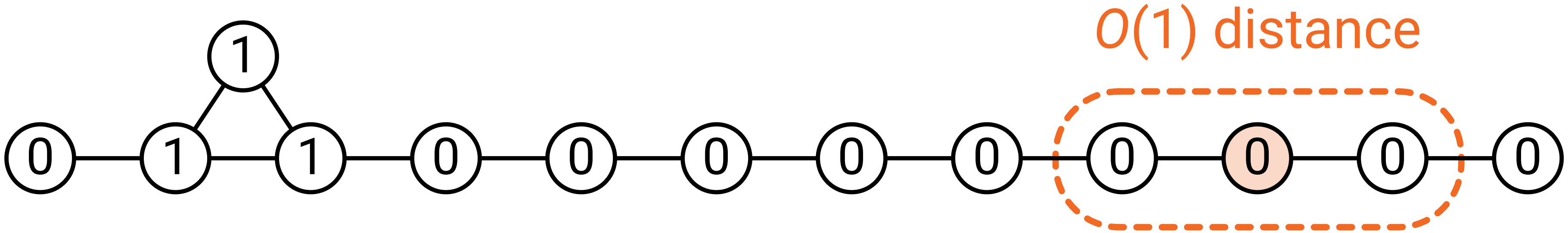
- **Combinatorial classifications**
- **Logical classifications**
- **Computational classifications**
- ***Information-theoretic classifications***
  - how **much** do you need to know about the graph to solve the problem?
  - how **many bits** do you need to communicate?
  - how **far** do you need to see?

Classifying graph problems

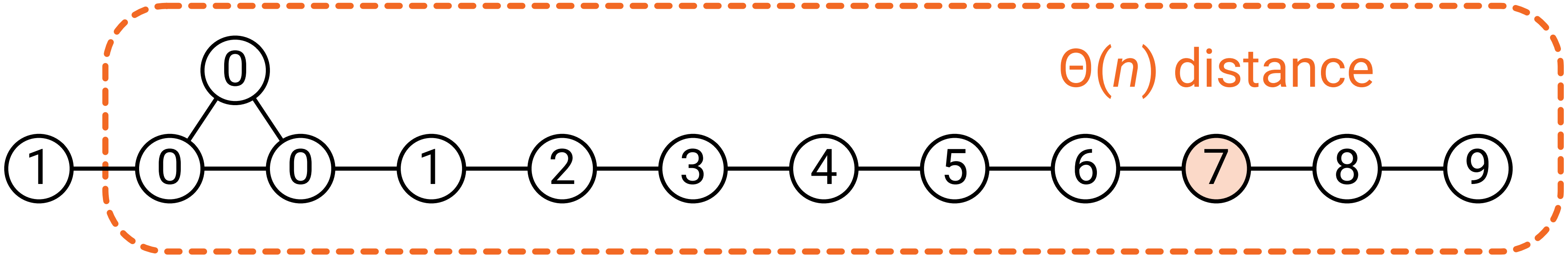
- **Combinatorial classifications**
- **Logical classifications**
- **Computational classifications**
- ***Information-theoretic classifications***
  - how **much** do you need to know about the graph to solve the problem?
  - how **many bits** do you need to communicate?
  - how **far** do you need to see?

Today's focus: locality

Local: am I part of a triangle?



Global: how far am I from the nearest triangle?



Why care about locality?

- **Problem is local:**
  - efficient ***distributed algorithms*** — few communication rounds
  - efficient ***parallel algorithms*** — graph can be decomposed in small components that can be solved independently from each other
  - ***property testing*** and sublinear-time algorithms — sample random nodes, compute their own part of the solution
- **Problem is global:**
  - ***understanding nature***, biological systems, social networks, computer networks — fundamental limitation for any system that consists of independent agents that exchange information with each other

Locality ≈ distributed time complexity

- “**LOCAL**” model of distributed computing:
  - ***graph = communication network***
    - **node** = processor
    - **edge** = communication link
    - all nodes have unique identifiers
  - ***time = number of communication rounds***
    - **round** = nodes exchange messages with all neighbors
    - 1 communication round: all nodes can learn everything within distance 1
    - T communication rounds: all nodes can learn everything within distance T
- ***Time = distance***

Towards distributed complexity theory

- Lots of work on ***specific graph problems***:
  - maximal independent set, maximal matching,
    vertex coloring, edge coloring, sinkless orientation …
  - upper bounds, lower bounds
  - reductions between problems
- What can we say about ***entire families of graph problems***?

We need the right definitions!

We have had the right concept since 1995 — didn't see it until recently

LCL: locally checkable labeling

- **Input:**
  - graph of maximum degree $\Delta = O(1)$
  - node (or edge) labels from set X, with $|X| = O(1)$
- **Output:**
  - node (or edge) labels from set Y, with $|Y| = O(1)$
- **Constraints:**
  - solution is globally feasible if it is locally feasible in all $O(1)$-radius neighborhoods

Naor & Stockmeyer 1995

LCL: locally checkable labeling

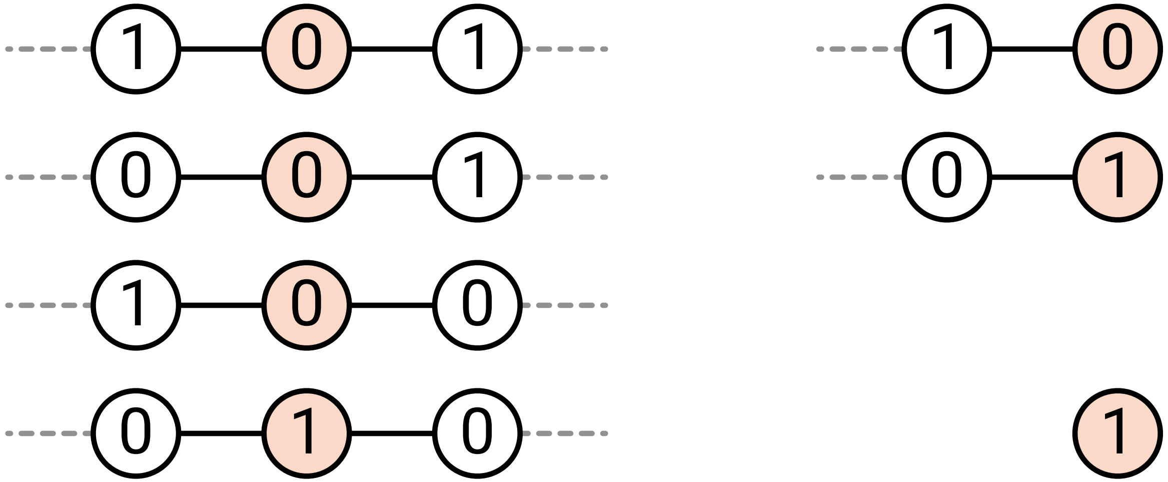
- Examples (in graphs of max degree Δ):
  - ***(Δ+1)-coloring***, ***Δ-coloring***, ***3-coloring*** …
  - ***maximal independent set***, ***maximal matching*** …
  - ***sinkless orientation***
    - orient all edges
    - all nodes of degree ≥ 3 have outdegree ≥ 1
  - ***locally optimal cut***
    - label nodes black/white
    - at least half of the neighbors have opposite color
  - ***SAT*** (when interpreted as a graph problem)
  - many other constraint satisfaction problems

LCL: locally checkable labeling

- Typically ***LCLs***:
  - finding a solution that satisfies local constraints
    - e.g. “maximal”, “minimal”, “equilibrium”
- Typically ***not LCLs***:
  - finding a solution that satisfies global constraints
    - e.g. “acyclic”, “connected”
  - optimization, approximation
  - decision, counting, enumeration

LCL: locally checkable labeling

- Any LCL has a trivial ***finite specification***:
  list all feasible local neighborhoods



Example: maximal independent set (Δ = 2)

LCL: locally checkable labeling

- LCL problems can be solved in $O(1)$ rounds with ***nondeterministic*** distributed algorithms
  - all nodes non-deterministically **guess** a solution
  - all nodes ***verify*** the solution in their local neighborhood
- Natural distributed analogue of class **NP**

LCL: locally checkable labeling

- Why is this a useful concept?
- **Good:** many problems that we study in distributed computing are LCLs
- **Not obvious:** can we prove any interesting theorems of the form: "***If P is any LCL problem, then …***"?

LCL: locally checkable labeling

- Why is this a useful concept?
- **Good:** many problems that we study in distributed computing are LCLs
- **Not obvious:** can we prove any interesting theorems of the form: "***If P is any LCL problem, then …***"?

Yes!

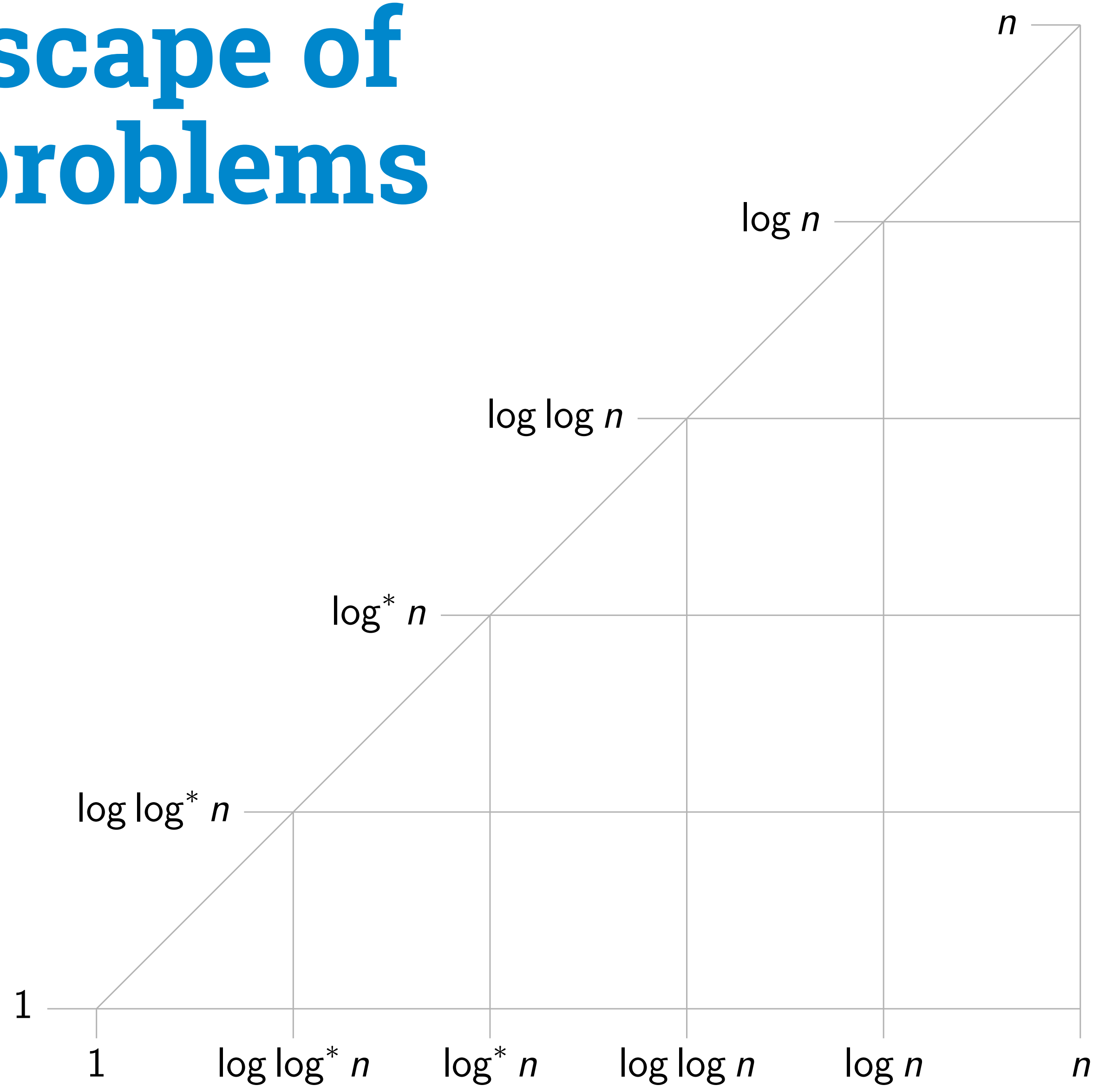
How local are LCL problems?

When does local checkability imply local solvability?

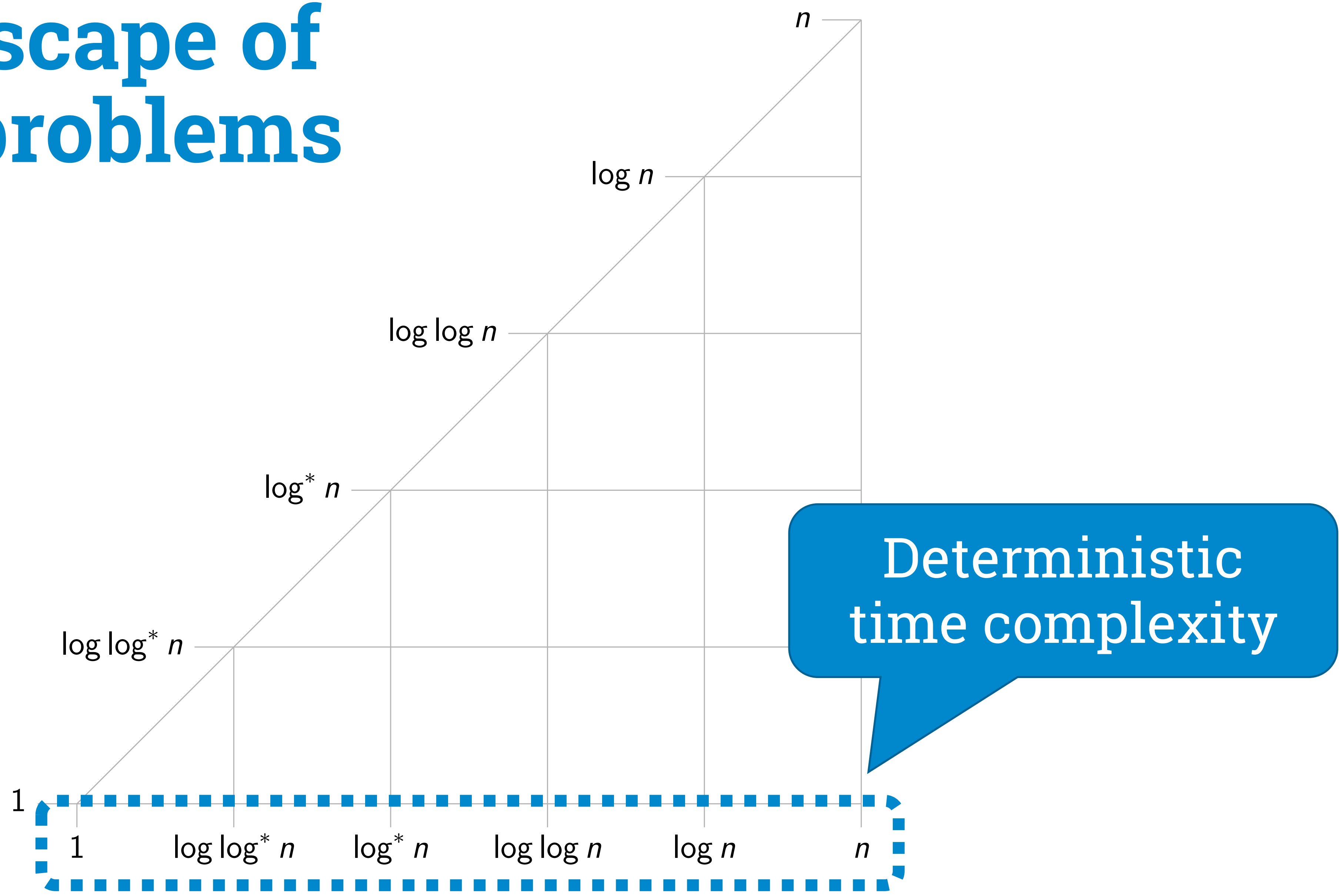
Locality of LCL problems

- Some LCLs are trivially ***local***
  - e.g. "label all nodes with 1"
- Some LCLs are trivially ***global***
  - e.g. 2-coloring: just telling if a solution exists requires $\Theta(n)$ distance
- Can we have LCLs that are ***"intermediate"***?
  - e.g. $\Theta(n^{1/2})$, $\Theta(\log n)$, $\Theta(\log \log n)$?
- Does ***randomness*** ever help with LCLs?
  - e.g. global deterministically, local with randomness?

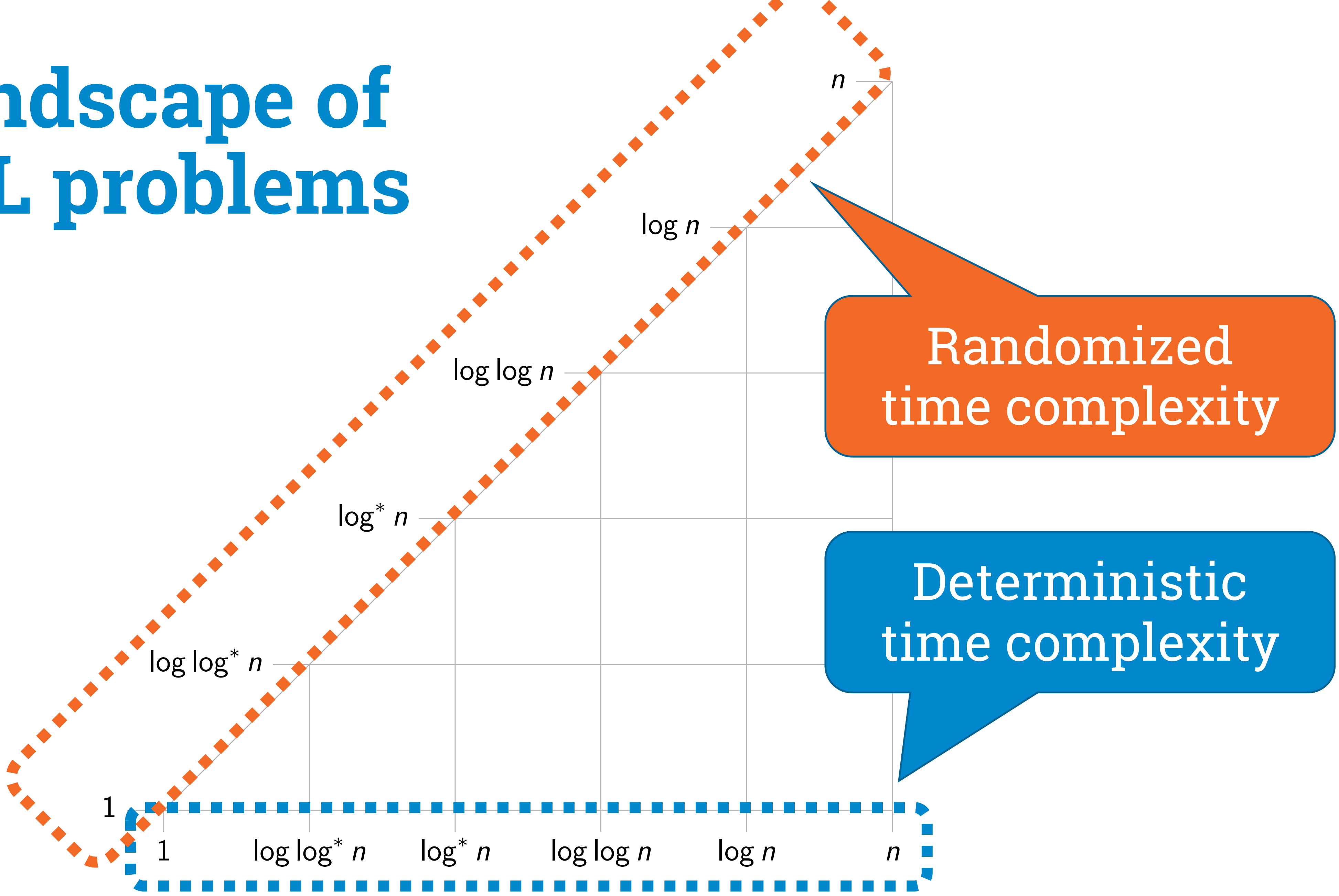
Landscape of LCL problems



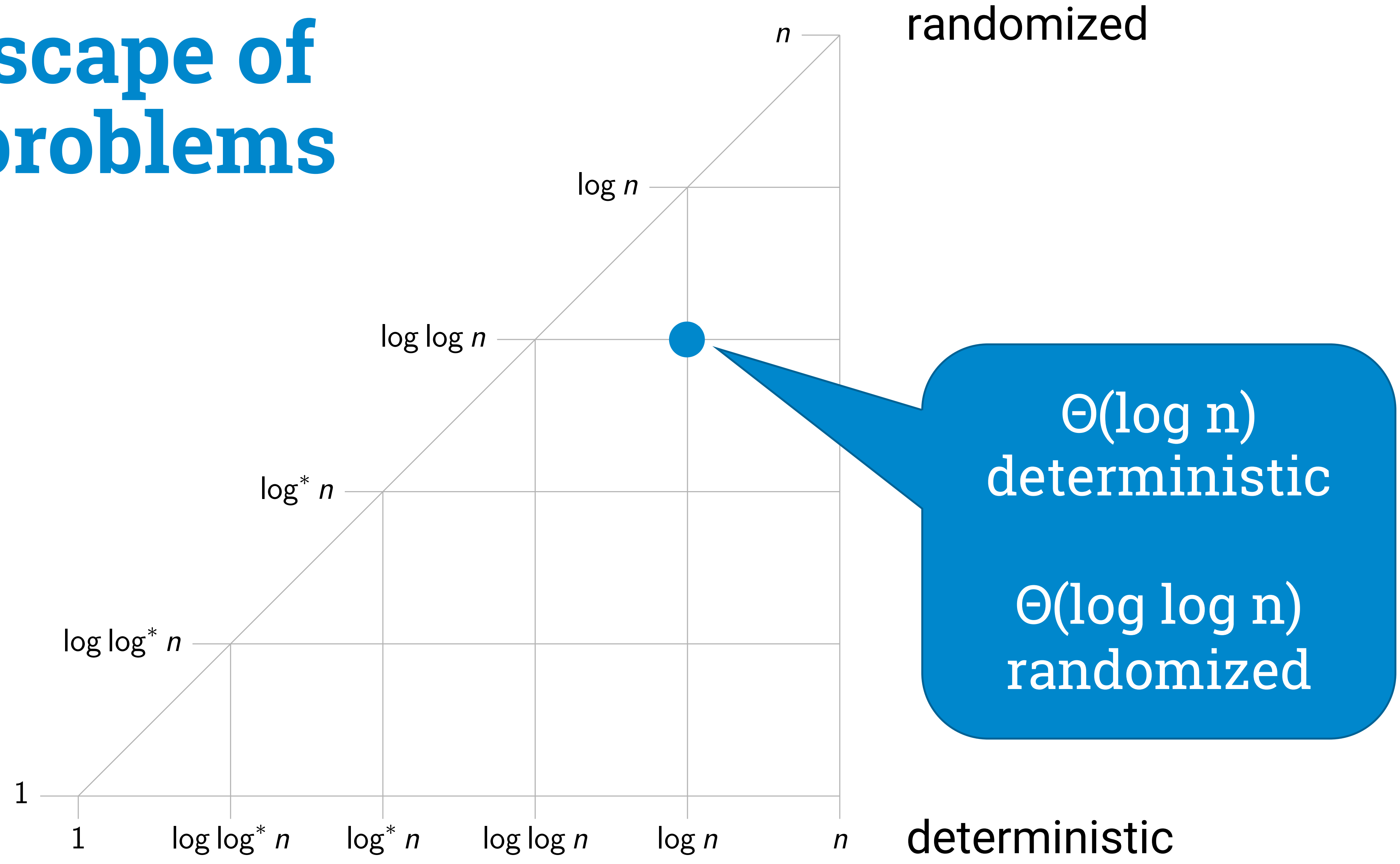
Landscape of LCL problems



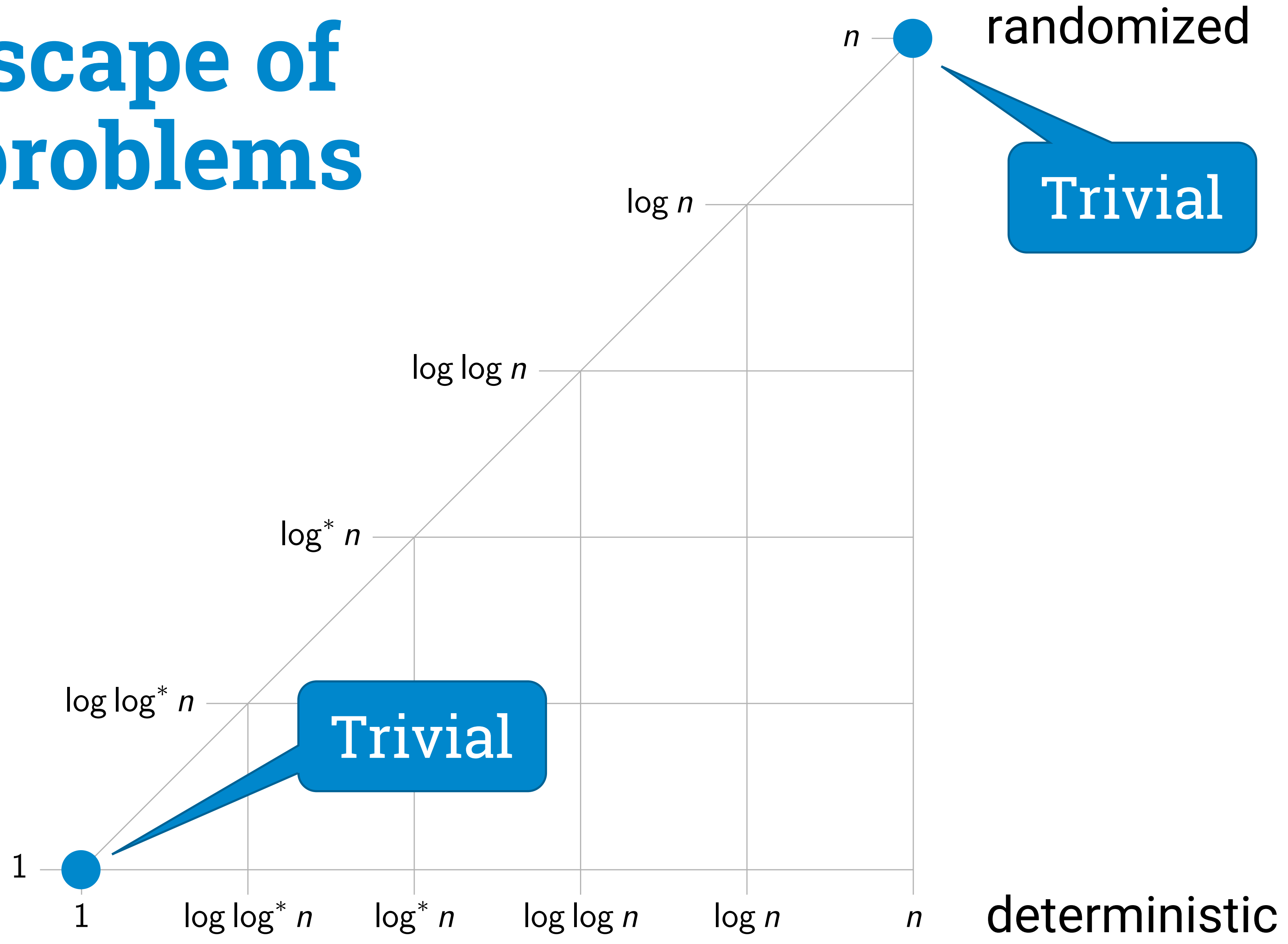
Landscape of LCL problems



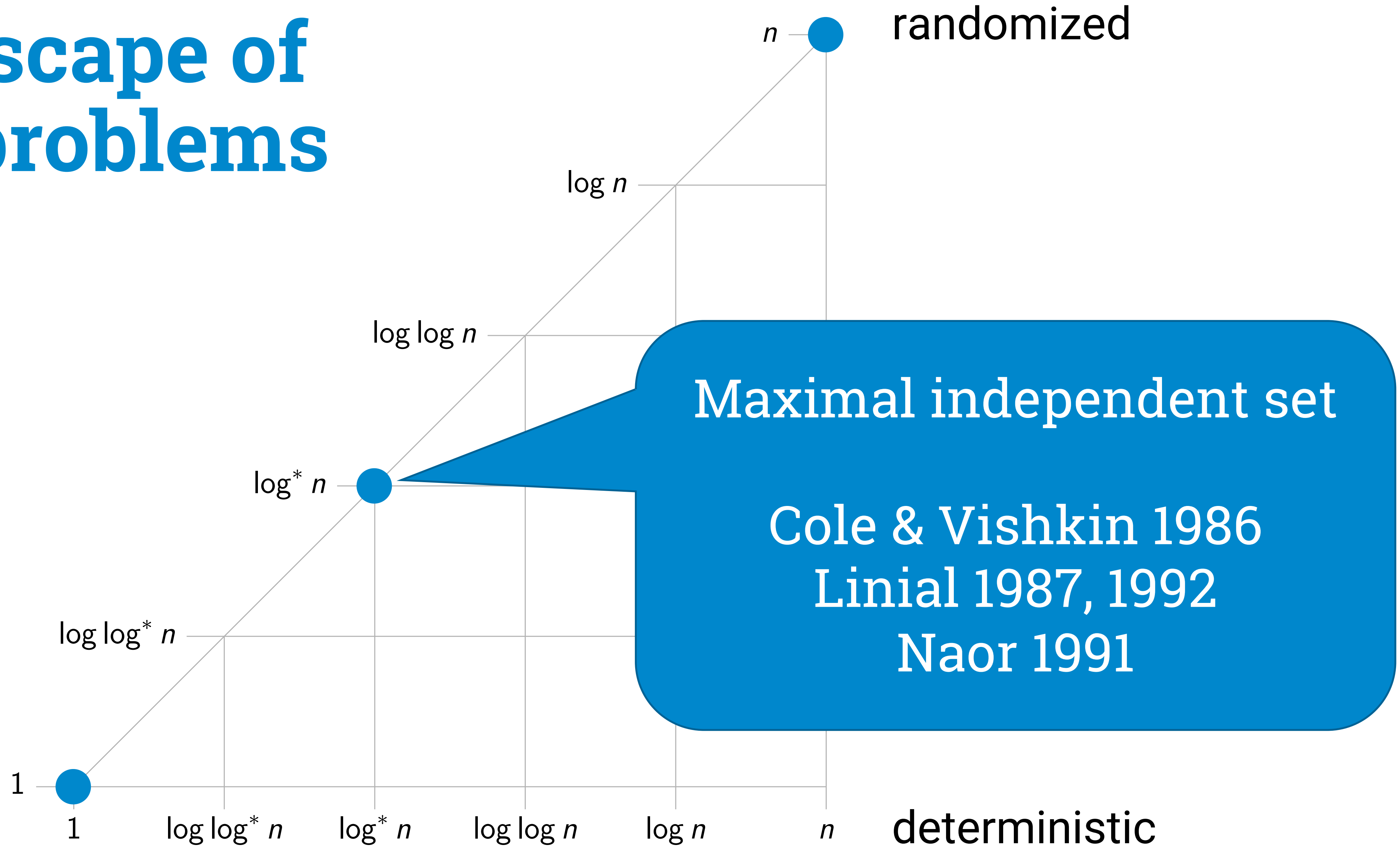
Landscape of LCL problems



Landscape of LCL problems

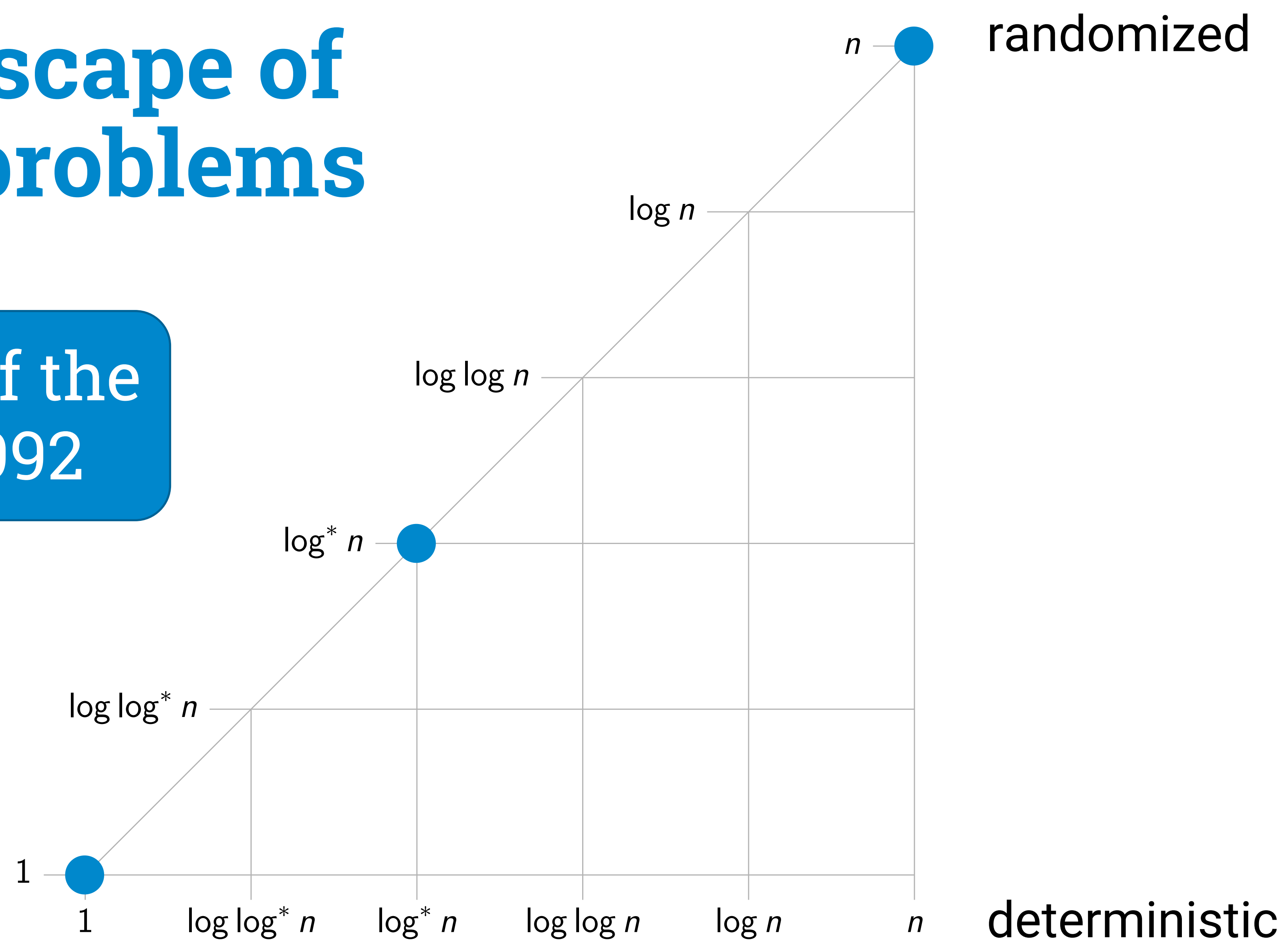


Landscape of LCL problems



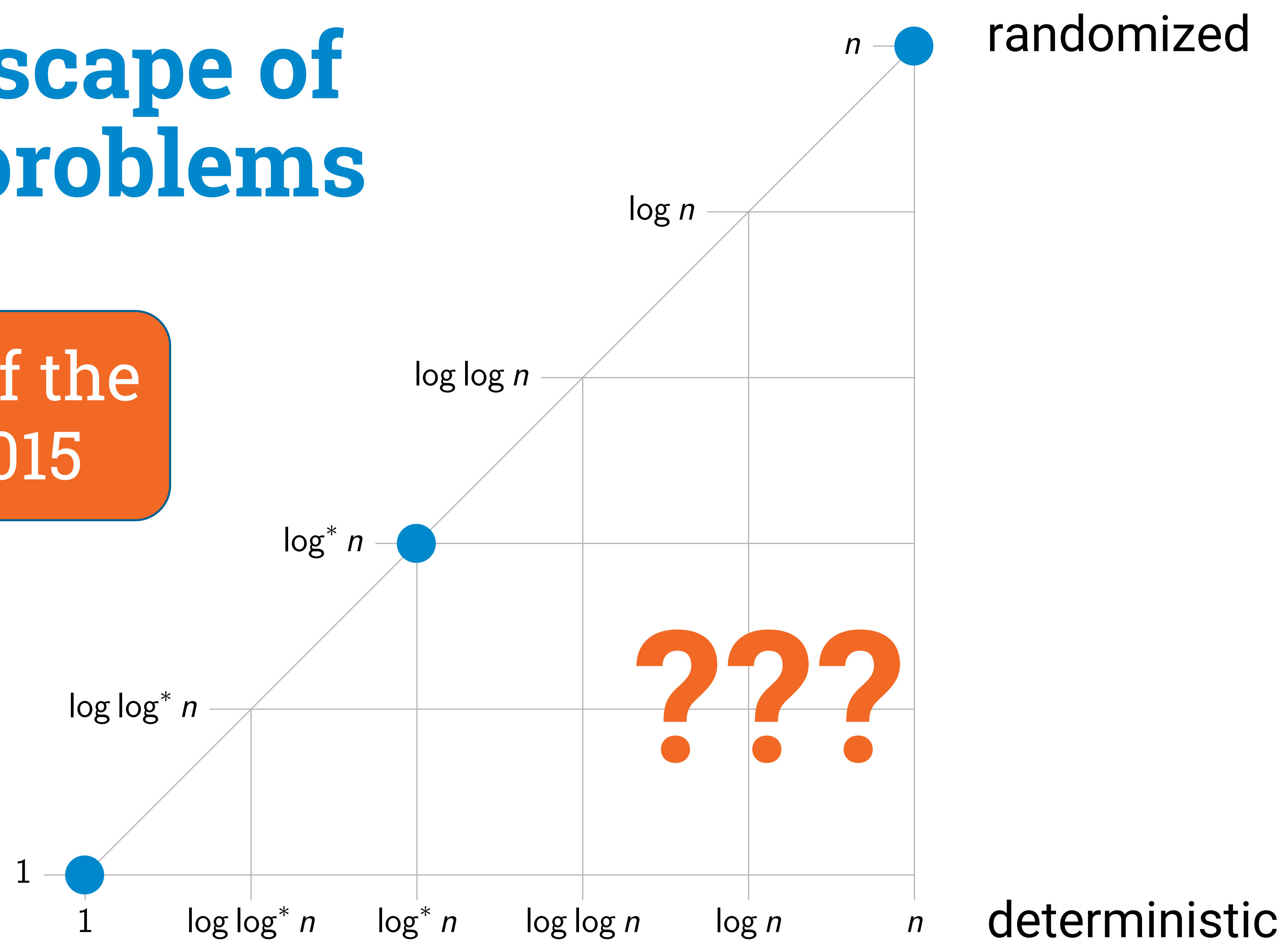
Landscape of LCL problems

State of the art 1992



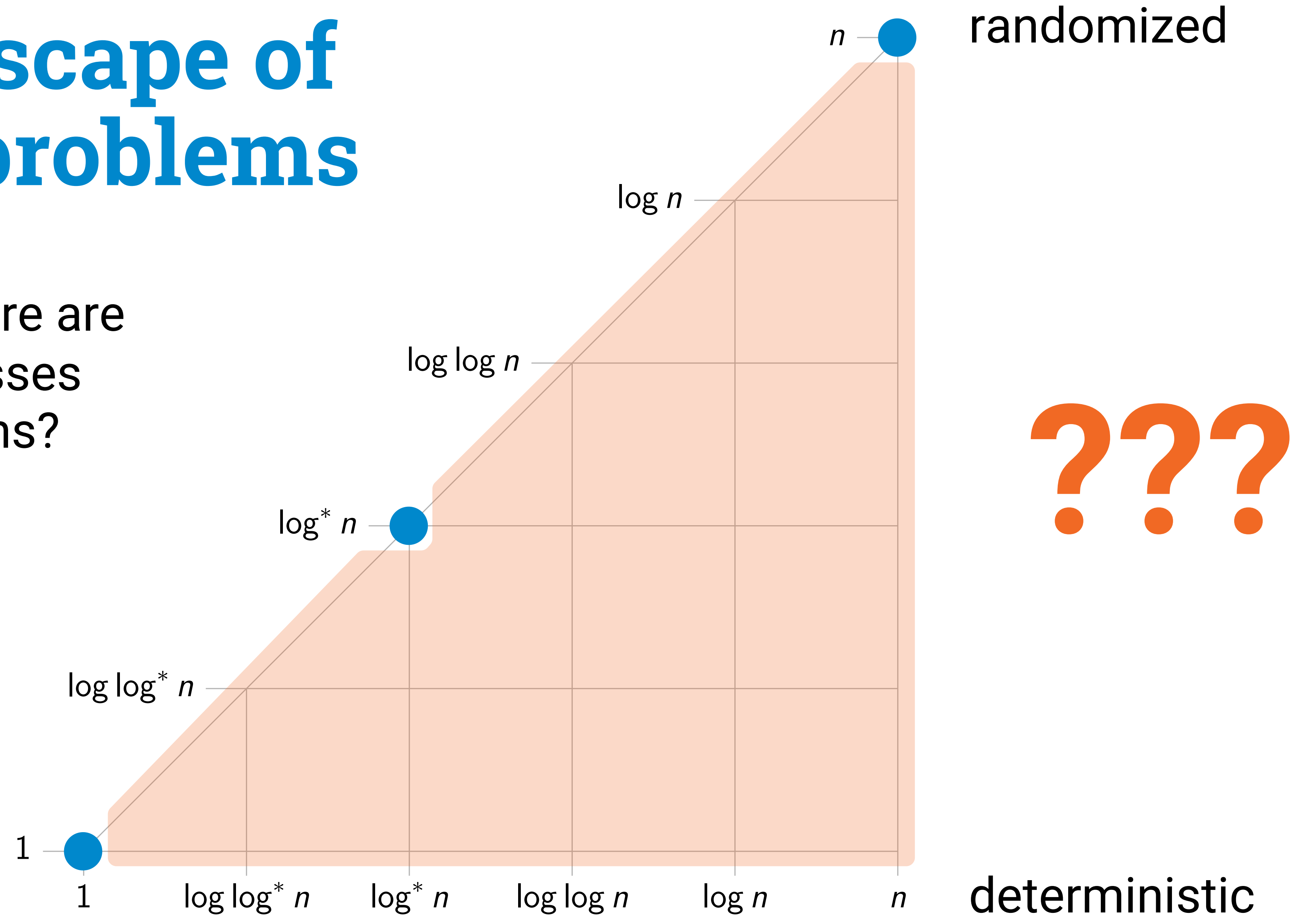
Landscape of LCL problems

State of the art 2015



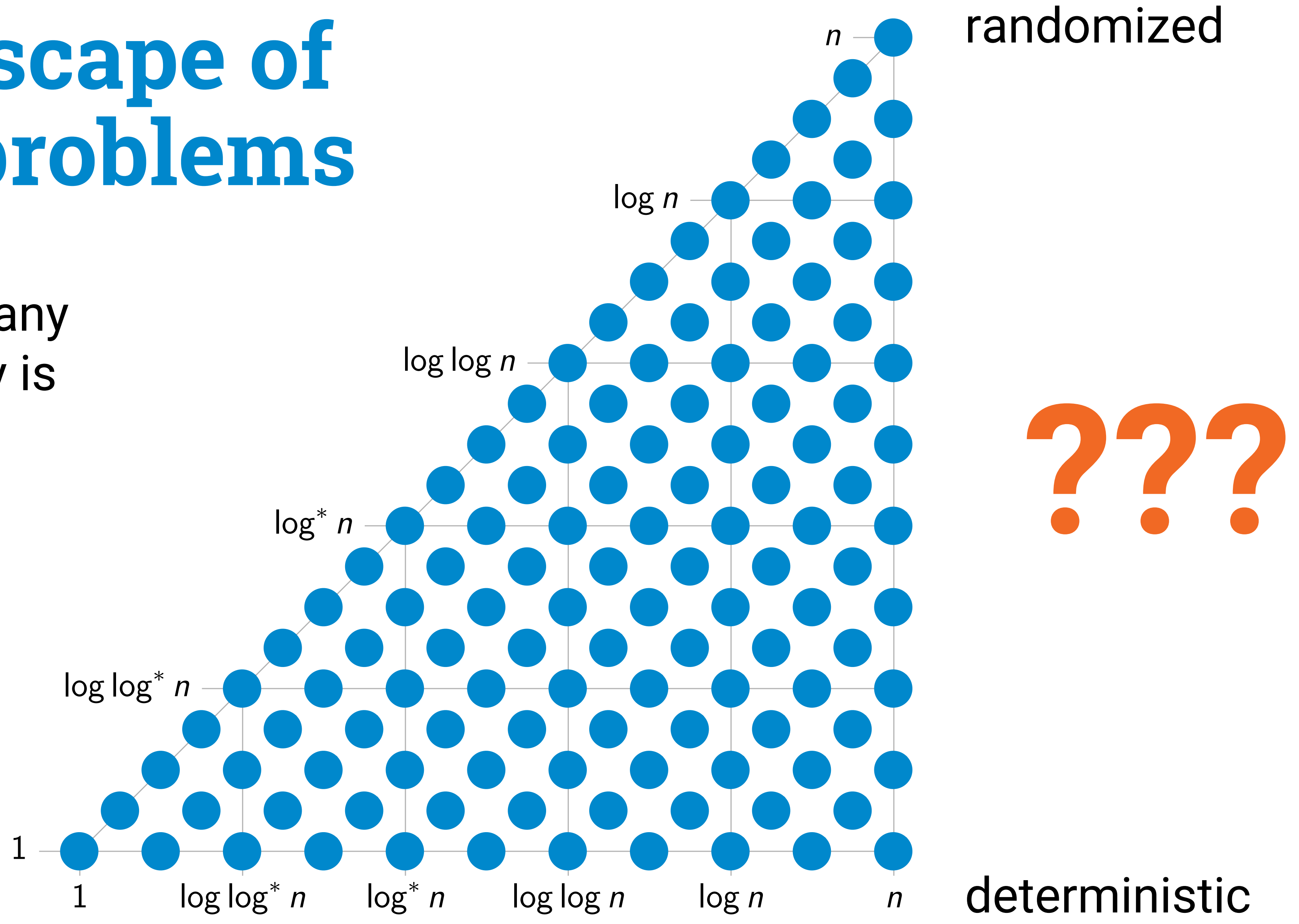
Landscape of LCL problems

Maybe there are only 3 classes of problems?



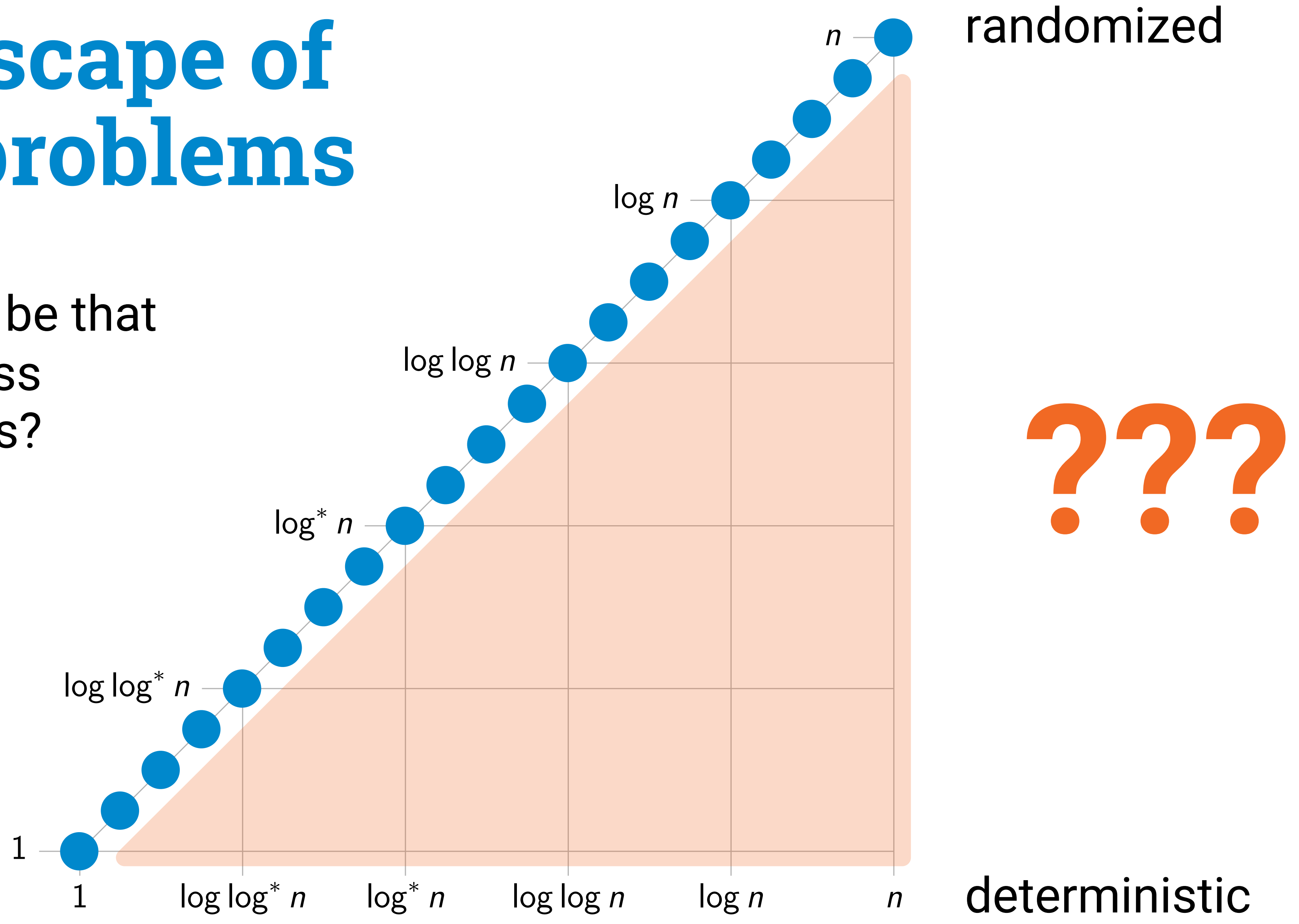
Landscape of LCL problems

Or what if any complexity is possible?



Landscape of LCL problems

Or could it be that randomness never helps?



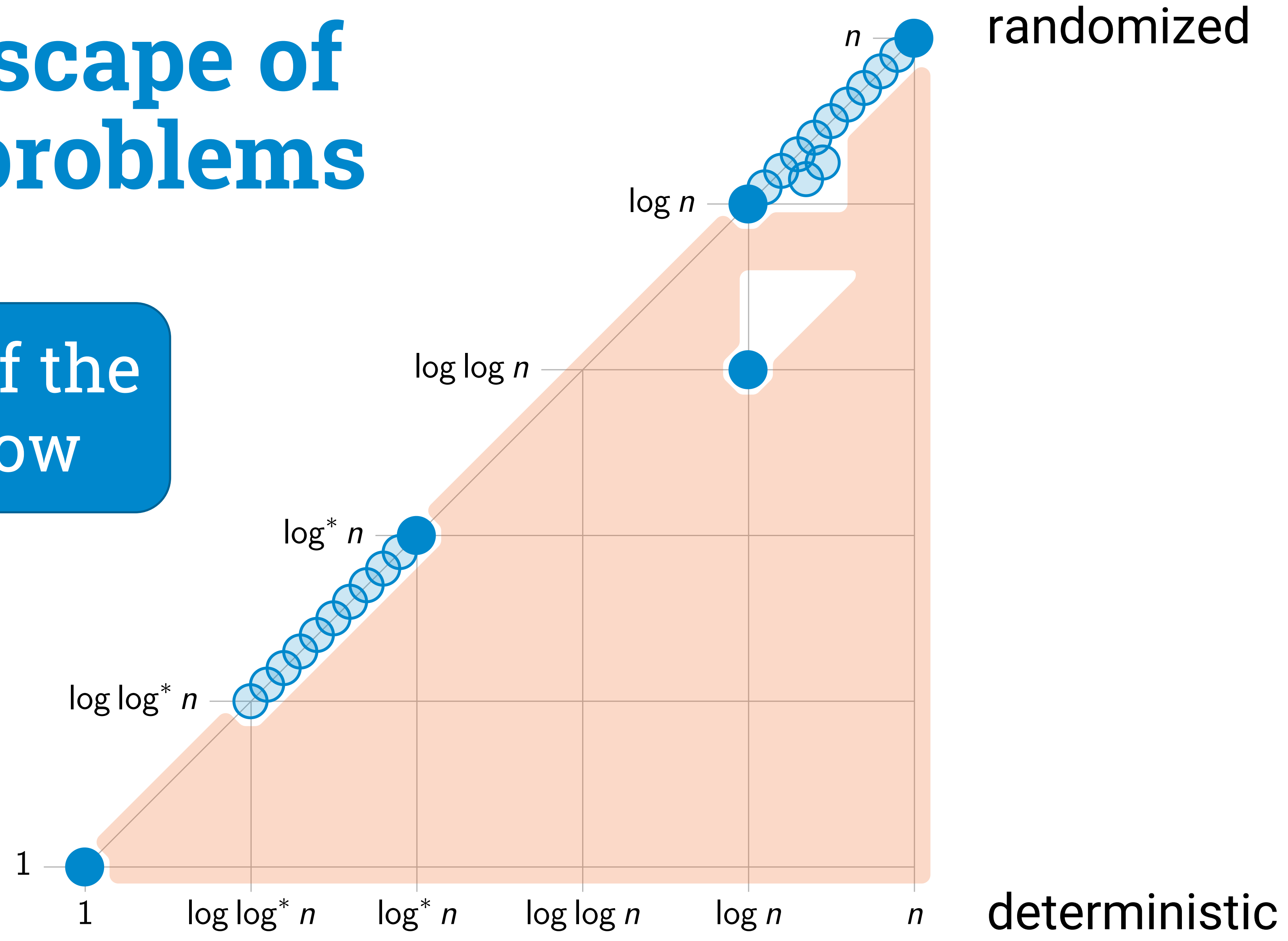
Landscape of LCL problems

Progress in 2016–

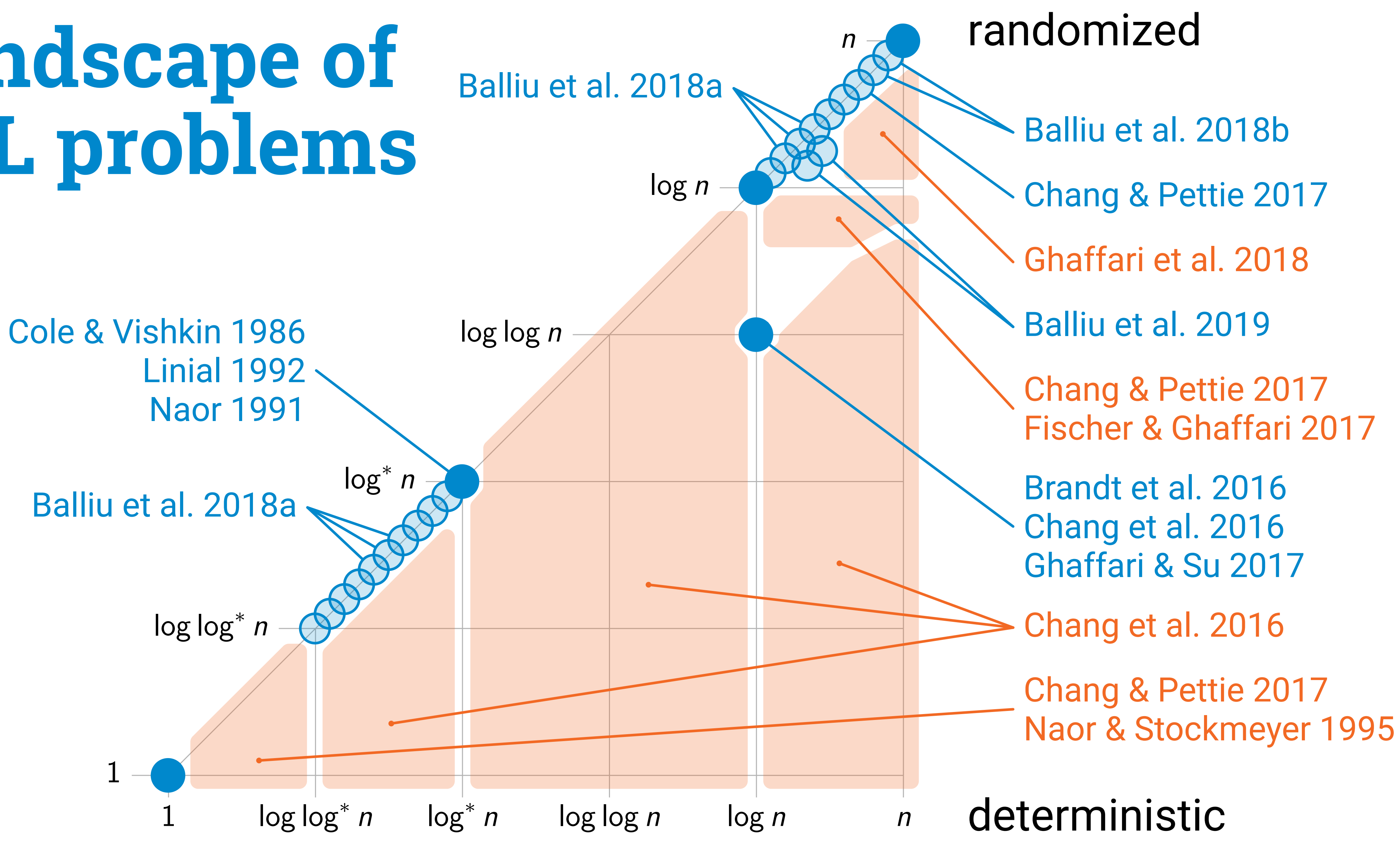
- Brandt, Fischer, Hirvonen, Keller, Lempiäinen, Rybicki, S, Uitto **(STOC 2016)**
- Chang, Kopelowitz, Pettie **(FOCS 2016)**
- Ghaffari, Su **(SODA 2017)**
- Brandt, Hirvonen, Korhonen, Lempiäinen, Östergård, Purcell, Rybicki, S, Uznański **(PODC 2017)**
- Fischer, Ghaffari **(DISC 2017)**
- Chang, Pettie **(FOCS 2017)**
- Chang, He, Li, Pettie, Uitto **(SODA 2018)**
- Balliu, Hirvonen, Korhonen, Lempiäinen, Olivetti, S **(STOC 2018)**
- Ghaffari, Hirvonen, Kuhn, Maus **(PODC 2018)**
- Balliu, Brandt, Olivetti, S **(DISC 2018)**

Landscape of LCL problems

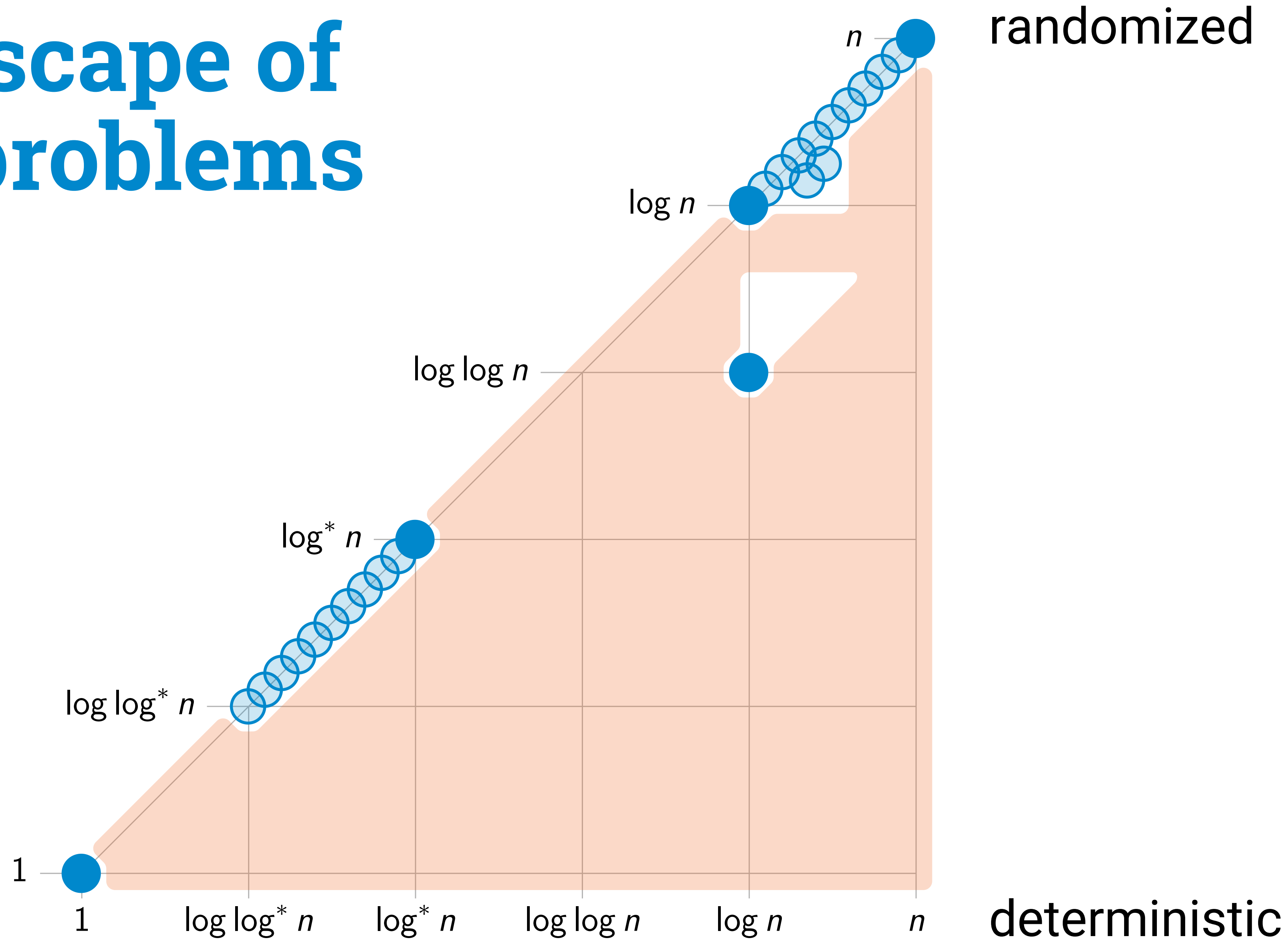
State of the art now



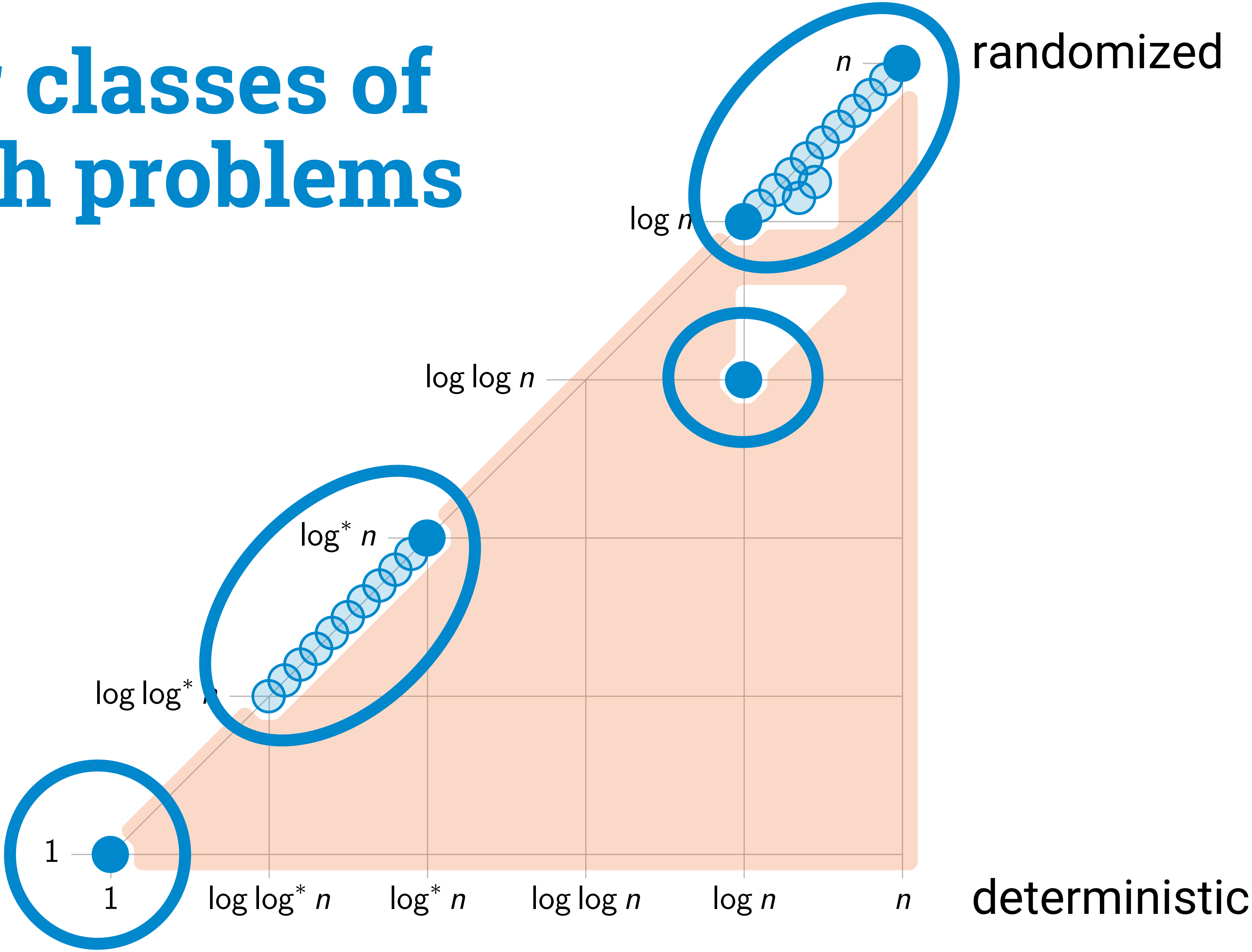
Landscape of LCL problems



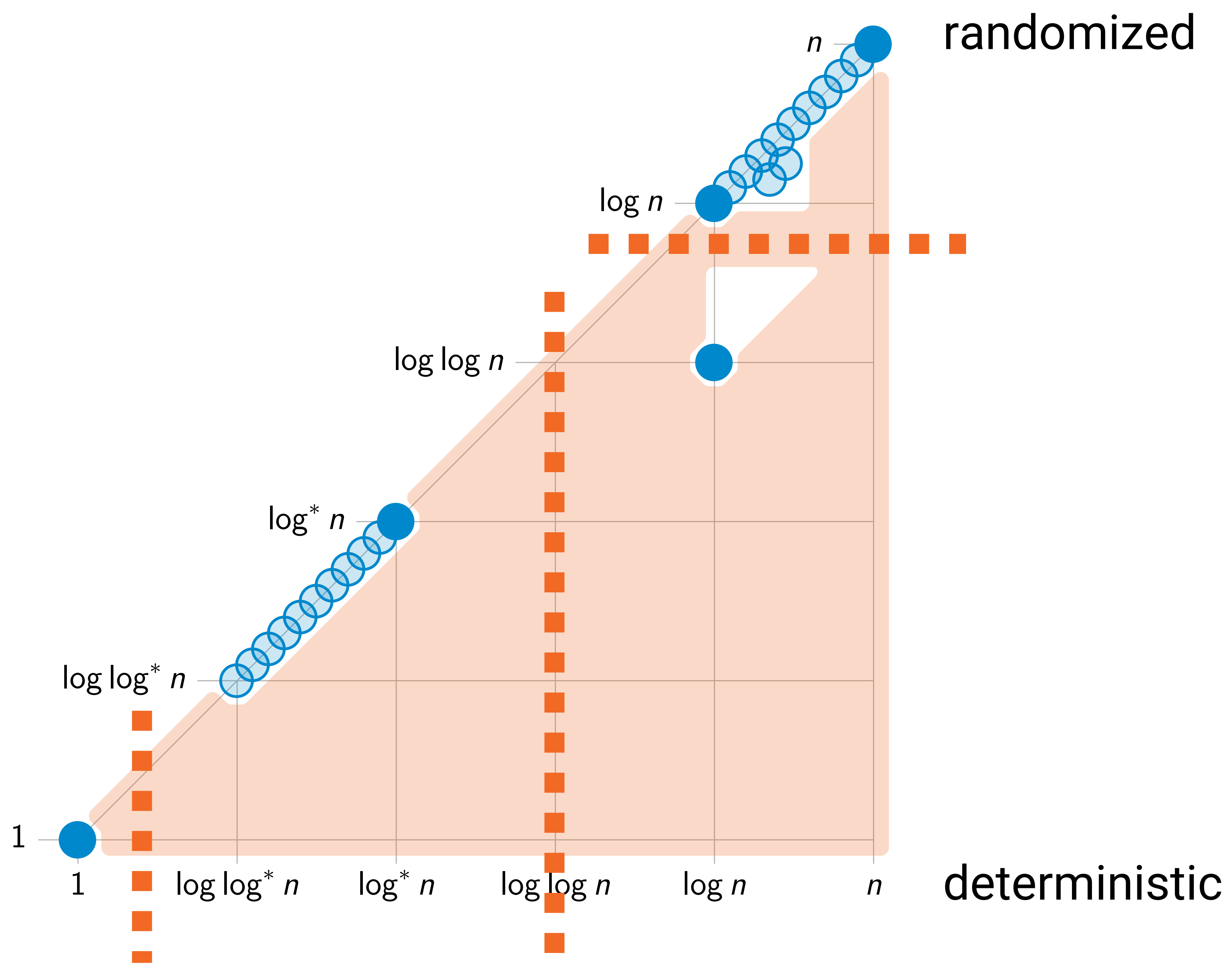
Landscape of LCL problems



Four classes of graph problems



Gaps



randomized
n
log n
log log n
log* n
log log* n
1
1
log log* n
log* n
log log n
log n
n
deterministic

Gaps have direct algorithmic implications

If you can solve an LCL problem

- in $o(\log n)$ rounds with a ***deterministic*** algorithm ***or***
- in $o(\log \log n)$ rounds with a ***randomized*** algorithm

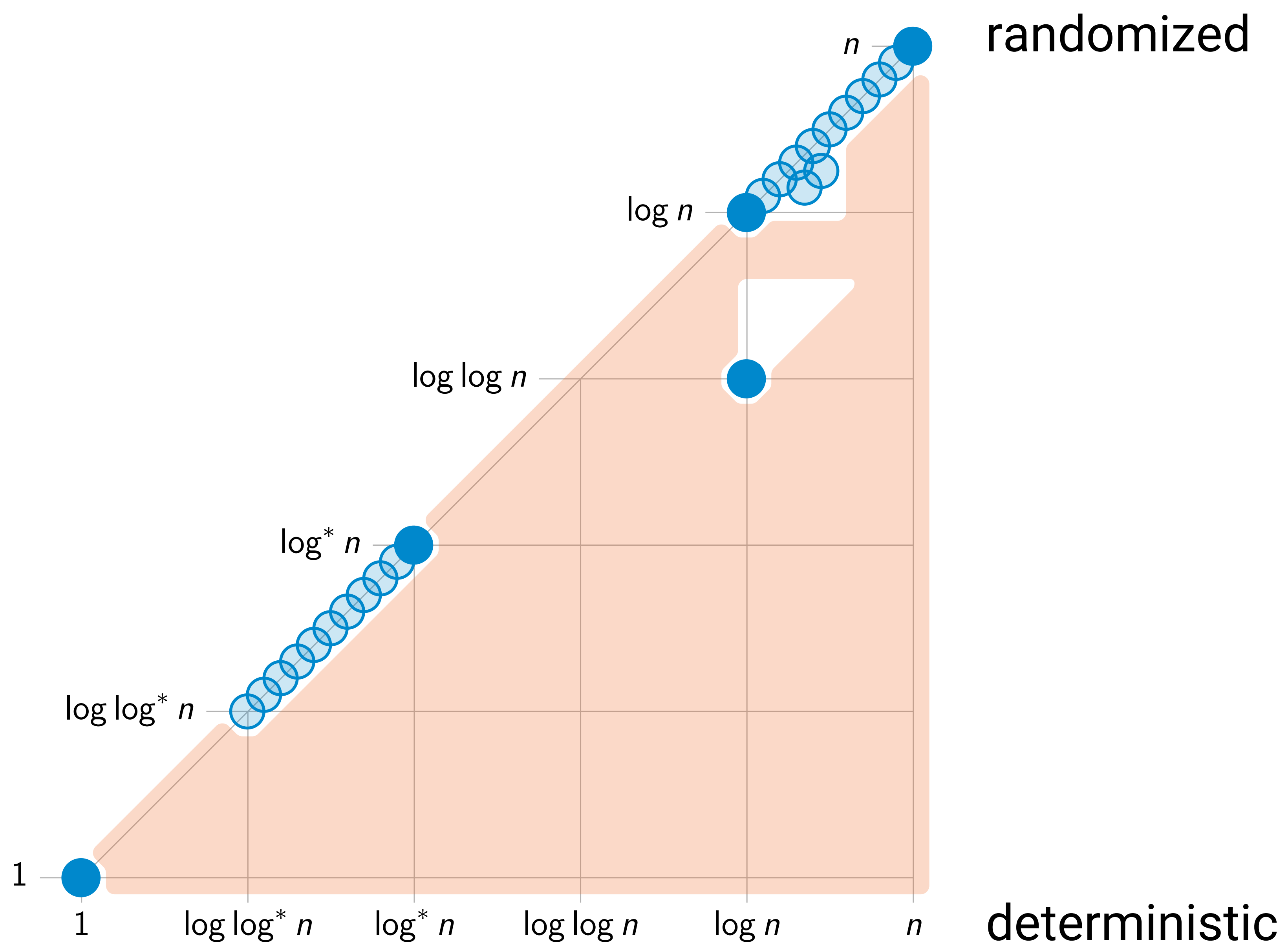
then you can also solve it

- in $O(\log^* n)$ rounds with a ***deterministic*** algorithms

Gaps have direct complexity-theoretic implications

If you can show that there is no $O(\log^* n)$-time deterministic algorithm then:

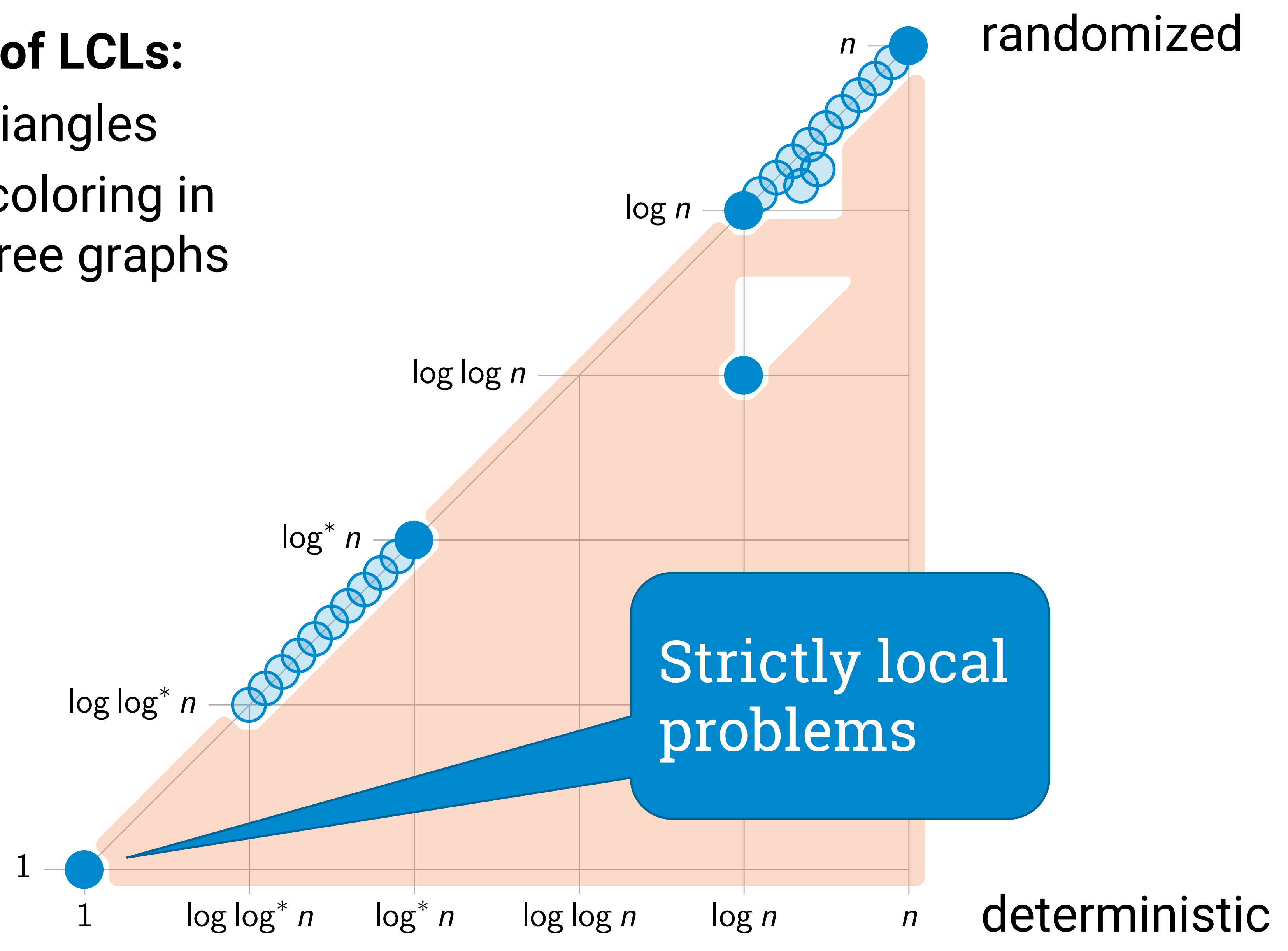
- ***deterministic*** complexity is at least $\Omega(\log n)$
- ***randomized*** complexity is at least $\Omega(\log \log n)$



randomized
n
$\log n$
$\log \log n$
$\log^* n$
$\log \log^* n$
1
1
$\log \log^* n$
$\log^* n$
$\log \log n$
$\log n$
n
deterministic

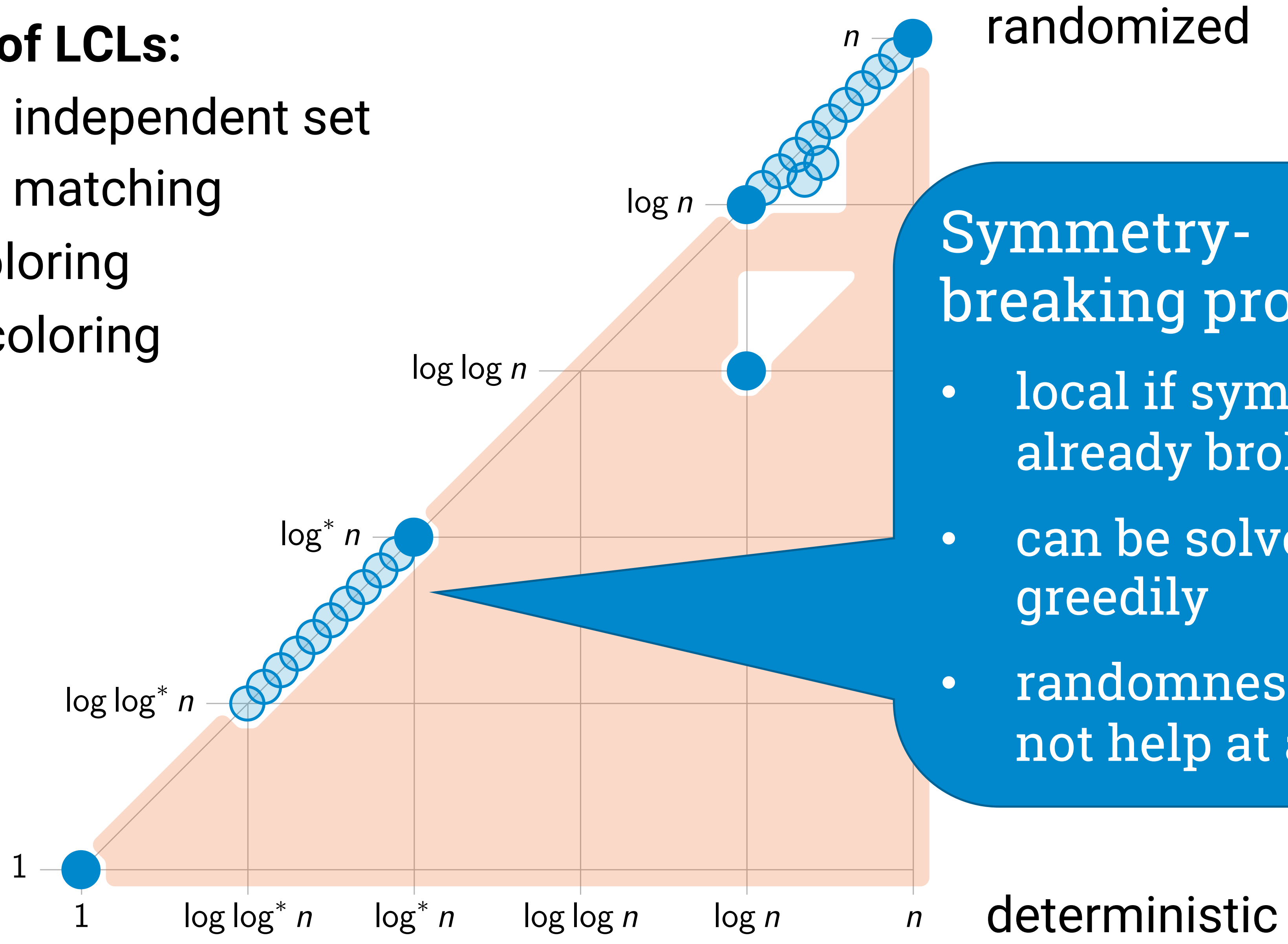
Examples of LCLs:

- detect triangles
- weak 2-coloring in odd-degree graphs



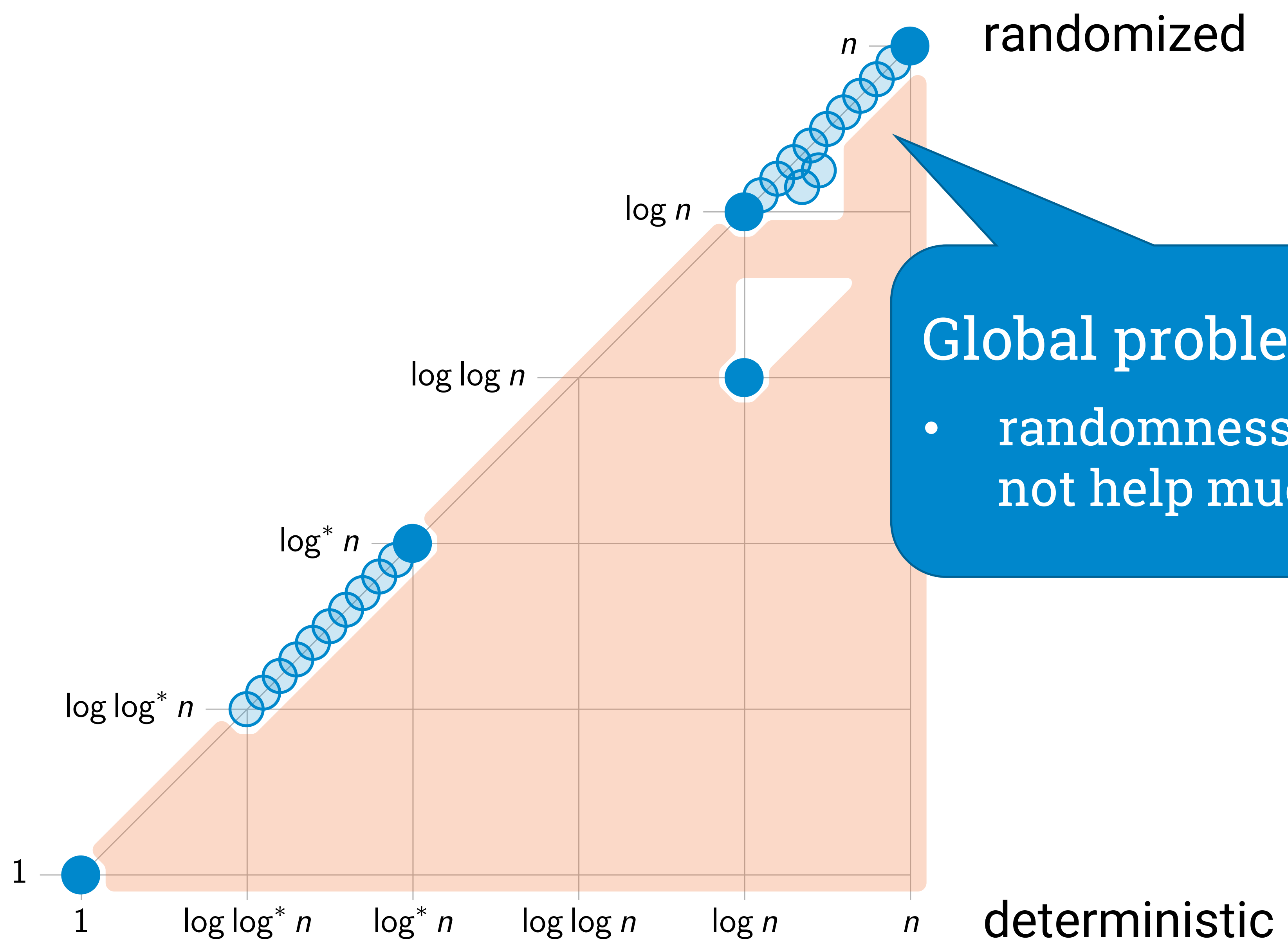
Examples of LCLs:

- maximal independent set
- maximal matching
- (Δ+1)-coloring
- weak 2-coloring



Symmetry-breaking problems

- local if symmetry already broken
- can be solved greedily
- randomness does not help at all

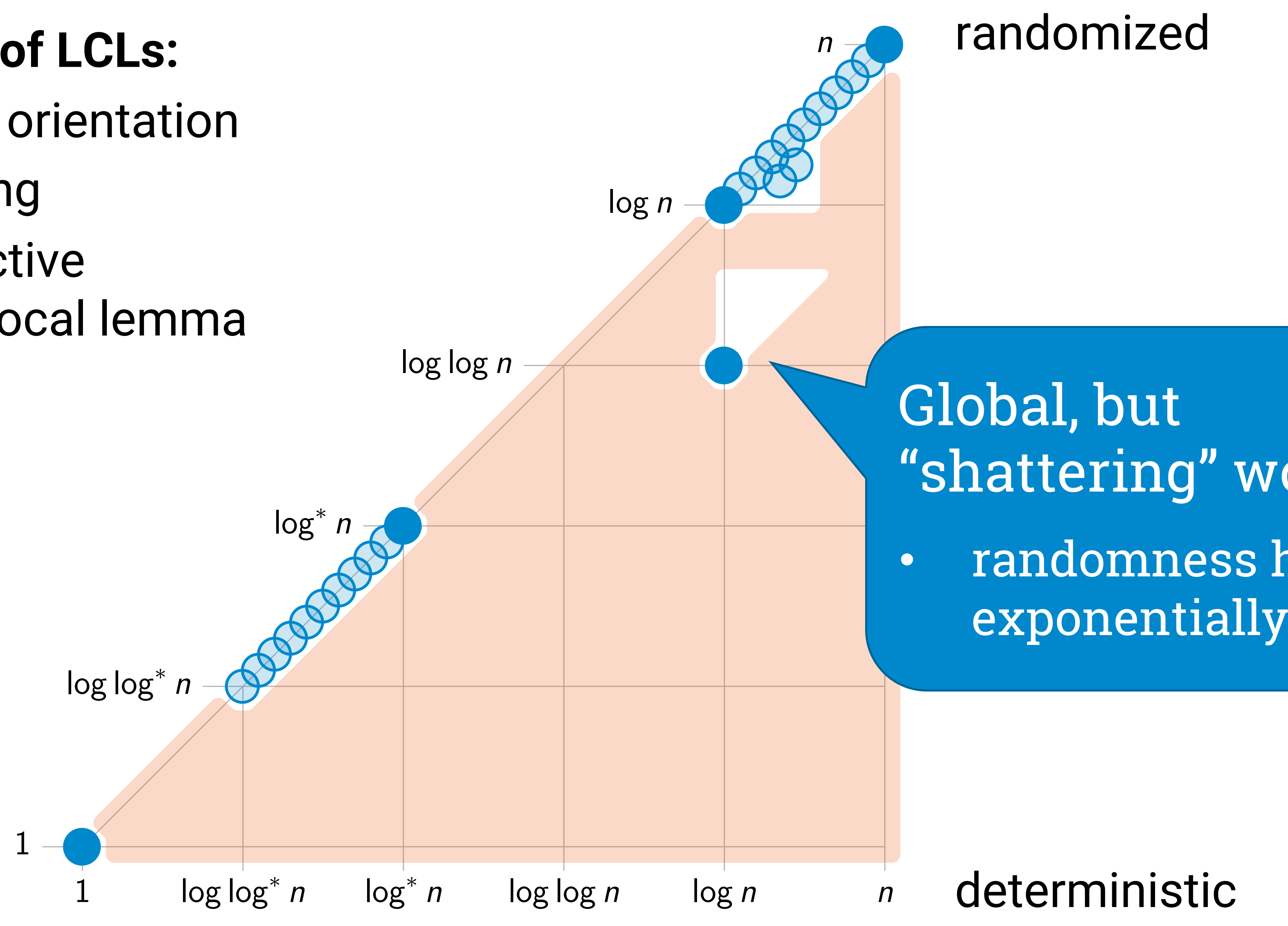


Global problems

- randomness does not help much

Examples of LCLs:

- sinkless orientation
- Δ-coloring
- constructive Lovász local lemma

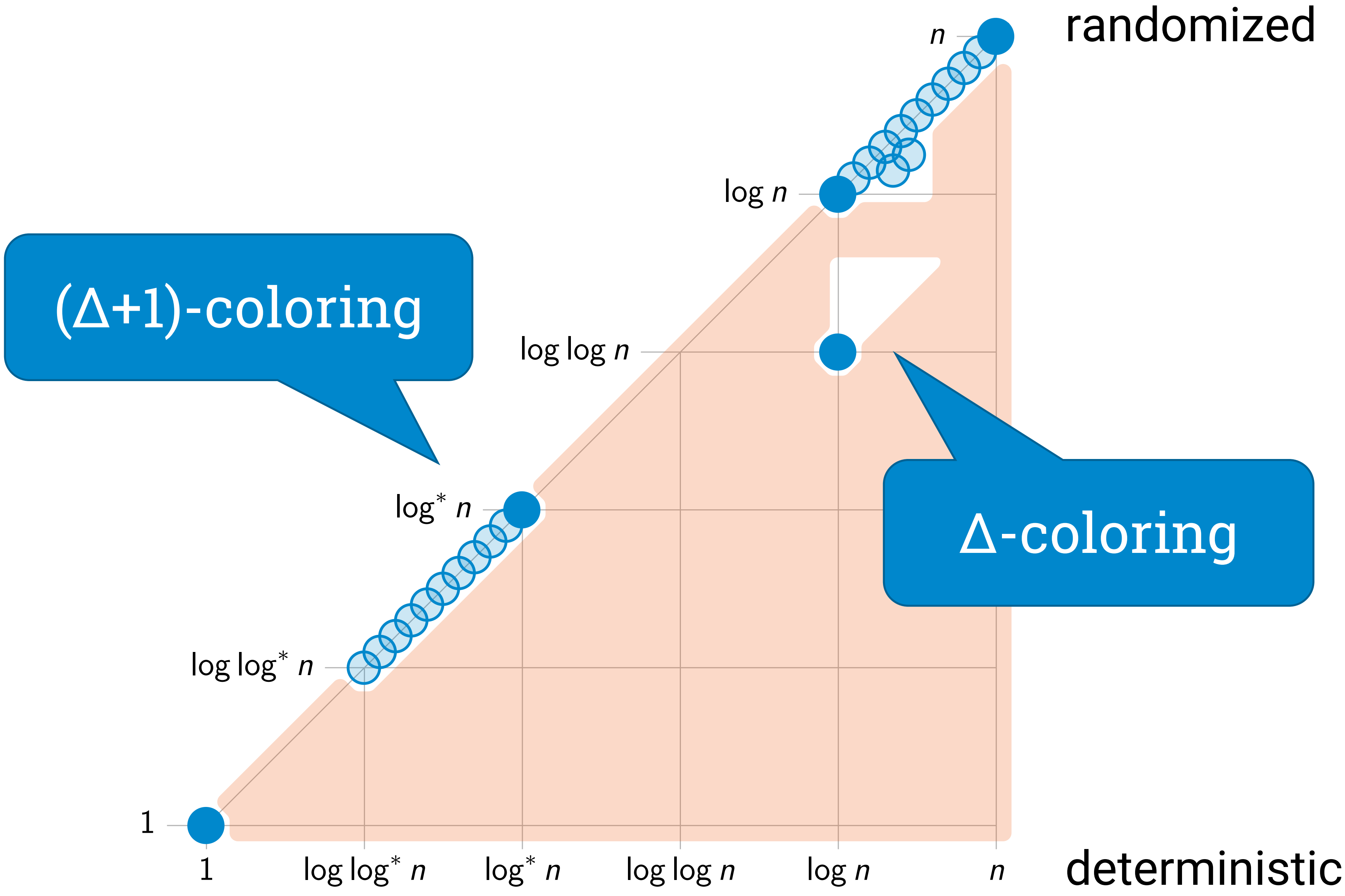


Global, but "shattering" works

- randomness helps exponentially

Shattering technique

- Example: ***sinkless orientation*** (in high-degree graphs)
- Orient half of the edges randomly
  - runs in $O(1)$ time
- Most nodes are ***happy***
  - there is at least one oriented outgoing edge
- ***Unhappy*** connected components have $O(\log n)$ nodes w.h.p.
  - apply $O(\log n)$-time deterministic algorithm in unhappy components
  - runs in $O(\log \log n)$ time


randomized
n
log n
log log n
log* n
log log* n
1
(Δ+1)-coloring
Δ-coloring
1
log log* n
log* n
log log n
log n
n
deterministic

Summary

- Key concept: ***locally checkable labeling*** (LCL)
  - bounded degrees, bounded inputs, bounded outputs
  - constant-radius checkable
- **Four distinct classes of LCL problems**
  - wide ***gaps*** → automatic speedups
  - robust classes, relevant also beyond the LOCAL model
- Key open challenge: finding useful generalizations