The Landscape of Distributed Time Complexity

Dennis Olivetti
Aalto University, Finland
Topic: distributed graph problems

- Family of graph problems: LCLs
- Focus on locality
  - How much does an entity need to know about the graph in order to solve a graph problem?
  - How local can these problems be?
  - When can randomness help?
LOCAL model

- **Graph** = communication network
- **Synchronous** rounds
- Time complexity = *number of rounds* required to solve the problem
- Nodes have **IDs**
- **No bounds** on the computational power of the entities and on the bandwidth
LOCAL model

• **Initial knowledge** of a node:
  • $n =$ the total number of nodes in the graph
  • $\Delta =$ the maximum degree of the graph
  • Its unique *ID*
  • A *port numbering* of its incident edges
  • Sequence of *random bits*
LOCAL model

• After \( t \) rounds:
  • knowledge of the graph up to distance \( t \)
• Focus on locality:
  • time = number of rounds = distance
LOCAL model

• After $t$ rounds:
  • knowledge of the graph up to distance $t$

• Focus on locality:
  • time = number of rounds = distance

Everything can be solved in Diameter time!
Locally Checkable Labelings (LCLs)

A family of graph problems that includes many important problems:
Maximal Independent Set, Maximal Matching, vertex coloring, edge coloring...
Locally Checkable Labelings (LCLs)

• Input
  • Graph of constant maximum degree Δ
  • Node labels from a constant-size set $X$

[Naor and Stockmeyer, 1995]
Locally Checkable Labelings (LCLs)

- **Input**
  - Graph of *constant* maximum degree $\Delta$
  - Node labels from a *constant-size* set $X$

- **Output**
  - Node labels from a *constant-size* set $Y$, such that each node satisfies some *local constraints*

[Naor and Stockmeyer, 1995]
Locally Checkable Labelings (LCLs)

- **Input**
  - Graph of \textit{constant} maximum degree $\Delta$
  - Node labels from a \textit{constant-size} set $X$

- **Output**
  - Node labels from a \textit{constant-size} set $Y$, such that each node satisfies some \textit{local constraints}

- **Correctness**
  - A solution is globally correct if it is correct in \textit{all constant-radius} neighborhoods

[Naor and Stockmeyer, 1995]
Locally Checkable Labelings (LCLs)

Two algorithms:
**Locally Checkable Labelings (LCLs)**

Two algorithms:

- **Prover**: runs for some time and produces an output, plus a certificate of correctness
Two algorithms:

- **Prover**: runs for some time and produces an output, plus a certificate of correctness
- **Verifier**: checks, in constant time, if the certificate is correct
Example: weak 2-coloring

- **Output**: color nodes from a palette of 2 colors
- **Constraint**: each node must have a **different color** from at least 1 neighbor
Landscape of LCLs

• Which time complexities are possible for LCLs?
• How local are LCLs?
• Does randomness help in solving an LCL faster?
Randomised

Deterministic
Randomised

Θ(log n) deterministic
Θ(log* n) randomized
Paths/Cycles
Paths/Cycles

- Trivial
- $\log^* n$
- $\log n$
- $n$

Cole & Vishkin 1986
Linial 1992

Deterministic
Randomised
Paths/Cycles

Cole & Vishkin 1986
Linial 1992

Naor & Stocmayer 1995

Chang et al 2016
Gaps

• \(\omega(1) - o(\log^* n)\) gap:
  
  • Every algorithm A that solves an LCL P in \(o(\log^* n)\) rounds can be automatically sped up into an algorithm A' that solves P in \(O(1)\) rounds

• \(\omega(\log^* n) - o(n)\) gap:
  
  • Every algorithm A that solves an LCL P in \(o(n)\) rounds can be automatically sped up into an algorithm A' that solves P in \(O(\log^* n)\) rounds
Using gaps for proving lower bounds

• The $\omega(\log^* n) - o(n)$ gap gives also a normalized way to solve LCLs:
  • Find a distance-k coloring in $O(\log^* n)$ time
  • Use the coloring in constant time
• Easy way to prove an $\Omega(n)$ lower bound:
  • Prove that, for any $k$, a problem can not be solved in $O(1)$ rounds given a distance-$k$ coloring!
Trees and general graphs?
Trees and general graphs?
Trees and general graphs?
Lots of progress since 2016

- Brandt, Fischer, Hirvonen, Keller, Lempiäinen, Rybicki, Suomela, Uitto [STOC 2016]
- Chang, Kopelowitz, Pettie [FOCS 2016]
- Ghaffari, Su [SODA 2017]
- Brandt, Hirvonen, Korhonen, Lempiäinen, Östergård, Purcell, Rybicki, Suomela, Uznański [PODC 2017]
- Fischer, Ghaffari [DISC 2017]
- Chang, Pettie [FOCS 2017]
- Chang, He, Li, Pettie, Uitto [SODA 2018]
- Balliu, Hirvonen, Korhonen, Lempiäinen, O., Suomela [STOC 2018]
- Ghaffari, Hirvonen, Kuhn, Maus [PODC 2018]
- Balliu, Brandt, O., Suomela [DISC 2018]
- Balliu, Brandt, O., Suomela [Unpublished 2019]
Trees
Trees

Randomised

Deterministic

1

1

log log* n

log*n

log log n

log n

n^{1/k}

n^{1/3}

n^{1/2}

n

n^{1/3}

n^{1/2}

n^{1/3}

n^{1/2}
Trees

- Randomised
- Deterministic

$\log n$

- $n^{1/2}$
- $n^{1/k}$
- $n^{1/3}$
- ... $n^{1/k}$

- 1
- $\log \log n$
- $\log* n$
- $\log \log* n$

- Gap: Chang & Pettie 2017
- Gap: Chang & Pettie 2017
- Gap: Chang et al. 2018
- Gap: Chang (unpublished)

- Chang (unpublished)
- Balliu et al. 2018
- Chang & Pettie 2017
- Chang et al. 2016
- Ghaffari & Su 2017

- Gap: Chang & Pettie 2017
- Gap: Chang (unpublished)

- Cole & Vishkin 1986
- Linial 1992

- Gap: Chang & Pettie 2017
- Naor & Stocmayer 1995

- Gap: Chang & Pettie 2017

- Gap: Chang & Pettie 2017
- Gap: Chang (unpublished)
Trees

Open problem
Trees

Homogeneous LCLs

Randomised

Deterministic

[Balliu, Hirvonen, O., Suomela 2019]
General graphs
General graphs
General graphs
Randomised

Deterministic

Randomness may help, but not by much

General graphs
Artificial problems

- How to get an LCL with complexity $\Theta(n^{3/5})$?
Artificial problems

• How to get an LCL with complexity $\Theta(n^{3/5})$?

• Define the following problem:
  • Solve some global problem if the diameter is $O(n^{3/5})$, or
  • Prove that the diameter is $\omega(n^{3/5})$
Artificial problems

• How to get an LCL with complexity $\Theta(n^{3/5})$?

• Define the following problem:
  
  • Solve some global problem if the diameter is $O(n^{3/5})$, or
  
  • Prove that the diameter is $\omega(n^{3/5})$

• Challenge:
  
  • It should not be possible to prove that the diameter is too high when it is not
  
  • It must always be possible to prove that the diameter is too high if it is true, no matter what the graph is
  
  • The number of labels must be constant
Randomised vs Deterministic complexity for various graph types:

- **Paths/Cycles**: Completely understood.
- **Trees**: Gap?
- **General graphs**: LLL?

Complexities include:
- $\log n$, $\log^* n$, $\log \log n$, $\log \log \log n$, $\log \log \log \log n$, ...