The distributed complexity of locally checkable problems on paths is decidable

Alkida Balliu, Sebastian Brandt, Yi-Jun Chang, Dennis Olivetti, Mikaël Rabie, Jukka Suomela
Given a graph problem, can we decide its distributed time complexity?
LOCAL model

- Entities = nodes
- Communication links = edges
- Input graph = communication graph
LOCAL model

- Each node has a **unique identifier** from 1 to $\text{poly}(n)$
- **No bounds** on the computational power of the entities
- **No bounds** on the bandwidth
LOCAL model

• Round 0
LOCAL model

- Round 1
LOCAL model

- Round 2
LOCAL model

• After $t$ rounds: knowledge of the graph up to distance $t$
• Focus on locality
Locally Checkable Labelings (LCLs)

- **Input**
  - Graph of **constant** maximum degree $\Delta$
  - 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 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

- **Correctness**
  - A solution is globally correct if it is correct in all constant-radius neighborhoods

[Naor and Stockmeyer, 1995]
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
Objective of this work

Given an LCL $\Pi = (\text{input}, \text{output}, \text{constraints})$ we want to:

- **Decide** the distributed complexity of $\Pi$
- **Synthesize** an asymptotically optimal algorithm for $\Pi$
State of the art
State of the art

• Paths/Cycles with **NO** input:
  • the time complexity is always decidable, and
  • it can be either \(O(1)\), \(\Theta(\log^* n)\), or \(\Theta(n)\)

[Naor and Stockmeyer 1995] [Chang et al. 2016] [Brandt et al. 2017]
State of the art

• Paths/Cycles with NO input:
  • the time complexity is always decidable, and
  • it can be either $O(1)$, $\Theta(\log* n)$, or $\Theta(n)$
    
    [Naor and Stockmeyer 1995] [Chang et al. 2016] [Brandt et al. 2017]

• $\sqrt{n} \times \sqrt{n}$ Grids:
  • the time complexity is undecidable, but
  • if the grid has no input, it can be either $O(1)$, $\Theta(\log* n)$, or $\Theta(\sqrt{n})$
  • if the grid has no input and is toroidal, it is decidable if there is a $O(1)$ algorithm
    
    [Naor and Stockmeyer 1995] [Brandt et al. 2017]
State of the art

• **Paths/Cycles with NO input:**
  - the time complexity is always decidable, and
  - it can be either $O(1)$, $\Theta(\log^* n)$, or $\Theta(n)$
    
    [Naor and Stockmeyer 1995] [Chang et al. 2016] [Brandt et al. 2017]

• $\sqrt{n} \times \sqrt{n}$ Grids:
  - the time complexity is undecidable, but
  - if the grid has no input, it can be either $O(1)$, $\Theta(\log^* n)$, or $\Theta(\sqrt{n})$
  - if the grid has no input and is toroidal, it is decidable if there is a $O(1)$ algorithm
    
    [Naor and Stockmeyer 1995] [Brandt et al. 2017]

• **Trees:**
  - it is decidable if the LCL requires $O(\log n)$ or $n^{\Omega(1)}$
    
    [Chang and Pettie 2017]
Unlabeled Directed Cycles

Independent Set

[Brandt et al. 2017]
Unlabeled Directed Cycles

Independent Set

[Brandt et al. 2017]
Unlabeled Directed Cycles

Independent Set

[Brandt et al. 2017]
Unlabeled Directed Cycles

[Brandt et al. 2017]
Unlabeled Directed Cycles

Independent Set

[Brandt et al. 2017]
Unlabeled Directed Cycles

[Brandt et al. 2017]
Unlabeled Directed Cycles

Independent Set

[Brandt et al. 2017]
Unlabeled Directed Cycles

Independent Set

[Brandt et al. 2017]
Unlabeled Directed Cycles

Independent Set

Self loop: $O(1)$

[Brandt et al. 2017]
Unlabeled Directed Cycles

Independent Set

Maximal Independent Set

Self loop: $O(1)$

[Brandt et al. 2017]
Unlabeled Directed Cycles

Independent Set

Self loop: $O(1)$

Maximal Independent Set

Flexible state: $\Theta(\log^* n)$

"1 0" is flexible:

\[ \forall k \geq 3, \exists \text{ cycle of length } k \text{ that starts and ends at "1 0"} \]

[Brandt et al. 2017]
Unlabeled Directed Cycles

Independent Set

Self loop: $O(1)$

Maximal Independent Set

Flexible state: $\Theta(\log^* n)$

"1 0" is flexible:
\[ \forall k \geq 3, \exists \text{ cycle of length } k \text{ that starts and ends at } "1 0" \]

[Brandt et al. 2017]
Unlabeled Directed Cycles

**Independent Set**

- 10 → 01
- Self loop: $O(1)$

**Maximal Independent Set**

- 10 → 01
- Flexible state: $\Theta(\log^* n)$

"1 0" is flexible:

$\forall k \geq 3, \exists$ cycle of length $k$ that starts and ends at "1 0"

[Brandt et al. 2017]
Unlabeled Directed Cycles

Independent Set

Maximal Independent Set

2-Coloring

Self loop: $O(1)$

Flexible state: $\Theta(\log^* n)$

"1 0" is flexible:
\[ \forall k \geq 3, \exists \text{ cycle of length } k \text{ that starts and ends at } "1 0" \]

Otherwise: $\Omega(n)$

[Brandt et al. 2017]
Grids

[Naor and Stockmeyer 1995]
Grids

[Naor and Stockmeyer 1995]
Grids

[Naor and Stockmeyer 1995]
Grids

[Naor and Stockmeyer 1995]
Grids

• Define an LCL that requires to output the execution of a Turing machine
  
• If the machine *terminates*, the LCL can be solved in $O(1)$

• If the machine *does not terminate*, the LCL requires $\Omega(\sqrt{n})$

[Naor and Stockmeyer 1995]
General Picture
General Picture

- Grids allow to propagate too much information
General Picture

• Grids allow to propagate too much information

• On trees/bounded treewidth graphs it should not be possible
General Picture

• Grids allow to propagate too much information

• On trees/bounded treewidth graphs it should not be possible

• Let us prove that the complexity of LCLs is decidable on trees!
General Picture

- Grids allow to propagate too much information
- On trees/bounded treewidth graphs it should not be possible
- Let us prove that the complexity of LCLs is decidable on trees!
  - It seems too hard, let us try with trees with NO input
General Picture

- Grids allow to propagate too much information
- On trees/bounded treewidth graphs it should not be possible
- Let us prove that the complexity of LCLs is decidable on trees!
  - It seems too hard, let us try with trees with NO input
  - The tree structure can be used to encode inputs!
General Picture

• Grids allow to propagate too much information

• On trees/bounded treewidth graphs it should not be possible

• Let us prove that the complexity of LCLs is decidable on trees!
  ‣ It seems too hard, let us try with trees with NO input
  ‣ The tree structure can be used to encode inputs!
  ‣ Let us just try to understand inputs, on cycles
Given an LCL Π on cycles/path with input, it is possible to decide its distributed time complexity, and synthesize an asymptotically optimal algorithm for Π.
Results

It is PSPACE-hard to distinguish whether an LCL \( \Pi \) on cycles/paths with input labels can be solved in \( O(1) \) time or it needs \( \Omega(n) \) time.

\[ \Pi \]
Input = \{0,1\}
Output = \{0,1,2\}
Constraints = {...}

PSPACE hard

Complexity of \( \Pi \)
Algorithm for \( \Pi \)
Hardness

Time

Tape

Hardness
Hardness

Tape

Time
Hardness
Hardness

Input: 

Locally checkable proof
Hardness

Input:

Copy the special symbol

OR

Prove that there is an error in the locally checkable proof

Output:
Hardness

The obtained LCL has binary input and it is radius 1 checkable.
Conclusions
Conclusions

• We can automatically obtain the complexity of any LCL on paths with input
Conclusions

• We can automatically obtain the complexity of any LCL on paths with input
• We can automatically obtain an optimal algorithm
Conclusions

• We can automatically obtain the complexity of any LCL on paths with input

• We can automatically obtain an optimal algorithm

• What about trees?
Conclusions

• We can automatically obtain the complexity of any LCL on paths with input

• We can automatically obtain an optimal algorithm

• What about trees?

  • Can we decide if an LCL is $O(\log^* n)$ or $\Omega(\log n)$?
Conclusions

• We can automatically obtain the complexity of any LCL on paths with input

• We can automatically obtain an optimal algorithm

• What about trees?
  • Can we decide if an LCL is $O(\log^* n)$ or $\Omega(\log n)$?
  • What about regular balanced trees with no input?
Conclusions

• We can automatically obtain the complexity of any LCL on paths with input

• We can automatically obtain an optimal algorithm

• What about trees?

  • Can we decide if an LCL is $O(\log^* n)$ or $\Omega(\log n)$?

  • What about regular balanced trees with no input?

Thank you!