Related to...

Brandt et al.: “A lower bound for the distributed Lovász local lemma”, STOC 2016

arxiv.org/abs/1511.00900
Big picture

• Tuomo presented details related to LLL in December

• Now: *why does all this matter?*
Big picture

• Our focus:
  • distributed computing
  • distributed time complexity

• Compare with:
  • Turing machines
  • classes $P$, $NP$-intermediate, $NP$-complete
Classic setting

- **Algorithm:** Turing machine
- **Input:** string on a tape
- **Output:** string on a tape
- **Time:** elementary steps
Classic setting

- **P**
  - lots of things: *easy to compute*

- **NP-intermediate**
  - some candidates: factoring, graph isomorphism

- **NP-complete**
  - lots of things: *easy to check*, hard to compute?
Distributed setting

Computer network + message passing
  • input: network topology + unique node identifiers
  • output: local output for each computer
  • e.g. LOCAL model

Time step:
  • all computers in parallel: send + receive + compute
Distributed setting

$t = 0$

$t = 1$

$t = 2$

$t = 3$
Distributed setting

- Fast algorithm = localised algorithm
- *Time* = *distance*
Distributed setting

- Time $O(1) =$ “fast”
  - typically fairly trivial problems

- Time $\Theta(n) =$ “slow”
  - brute-force algorithms
  - everything is trivial
Distributed setting

• Time $O(1) = \text{“fast”}$
  • typically fairly trivial problems

• Intermediate time complexity?

• Time $\Theta(n) = \text{“slow”}$
  • brute-force algorithms
  • everything is trivial
Distributed setting

• Fairly trivial to construct contrived problems of any time complexity $T$
  • cheat with a promise on input: “trees of diameter $T$”
  • cheat with problem definition: “detect if there are any red nodes within distance $T$”
Distributed setting

- Fairly trivial to construct contrived problems of any time complexity $T$
- Cf. time hierarchy theorems
- Cf. class $\text{EXP}$
- What could be the analogue of $\text{NP}$?
Idea: easy to check

• **LCL: locally checkable labelling**

• Everything bounded:
  • $O(1)$ bits of output / node
  • $O(1)$ bits of input / node
  • maximum degree $\Delta = O(1)$
Idea: easy to check

• **LCL:** locally checkable labelling

• Everything bounded

• Correct solution can be locally verified:
  • check that *radius*-$O(1)$ *neighbourhoods* of all nodes look good
Idea: easy to check

• These are locally checkable labellings:
  • vertex colouring with \( k \) colours, edge colouring …
  • maximal independent set, minimal dominating set, maximal matching, perfect matching, SAT, …

• These are not:
  • spanning trees, Eulerian cycles …
  • maximum independent set, maximum matching …
LCL problems

- **Time $O(1)$**
  - easy to compute
  - cf. $P$

- **Time $\Theta(n)$**
  - easy to check, hard to compute
  - cf. $NP$
LCL problems

- **Time $O(1)$ — *a bit too strict!***
  - easy to compute
  - cf. $P$

- **Time $\Theta(n)$**
  - easy to check, hard to compute
  - cf. $NP$
\[ \log^* n \]

\[ \log \log \ldots \log n \leq 1 \]

\[ \log^* 10^{10000} = 5 \]
**log* \(n\)**

- Cole–Vishkin (1986) technique:
  - from \(x\) colours to \(O(\log x)\) colours in one round
  - paths: compare my colour with my successor
  - (value, index) of the *first bit that differs*

- Unique identifiers: \(\text{poly}(n)\) colours

- After \(O(\log^* n)\) steps: \(O(1)\) colours
$\log^* n$

- Lots of LCL problems in time $\Theta(\log^* n)$
  - typically: problems that are *easy to solve greedily*

- Examples:
  - vertex colouring with $\Delta+1$ colours, edge colouring with $2\Delta-1$ colours
  - maximal independent set, maximal matching, minimal dominating set
**LCL problems**

- **Time $O(\log^* n)$**
  - easy to compute, cf. $P$

- **Time $\Theta(n)$**
  - easy to check, hard to compute, cf. NP-complete
LCL problems

- **Time** $O(\log^* n)$
  - easy to compute, cf. $\mathsf{P}$

- *Intermediate problems?*
  - cf. $\mathsf{NP}$-intermediate?

- **Time** $\Theta(n)$
  - easy to check, hard to compute, cf. $\mathsf{NP}$-complete
Intermediate LCL problems

• Try to construct one!
  • without resorting to a promise…

• Not so easy to cheat any more

• Perhaps everything is either strictly local or strictly global?
  • $O(\log^* n)$ or $\Theta(n)$, nothing else?
Intermediate LCL problems

• First proper examples discovered this year!

• *Sinkless orientation*:
  • orient all edges so that all nodes have outdegree $\geq 1$
Intermediate LCL problems

• First proper examples discovered this year!

• Sinkless orientation:
  • orient all edges so that all nodes have outdegree $\geq 1$

• 2-regular graphs: boring…
  • upper bound $O(n)$, trivial
  • lower bound $\Omega(n)$, easy
Intermediate LCL problems

• First proper examples discovered this year!

• **Sinkless orientation**:
  • orient all edges so that all nodes have outdegree \( \geq 1 \)

• 3-regular graphs: more interesting!
  • upper bound \( O(\log n) \), e.g. using LLL
  • lower bound \( \Omega(\log \log n) \)
Intermediate LCL problems

• One problem found, more with reductions!

• Natural example: 
  \textit{d-colouring in d-regular graphs}, \( d \geq 3 \)
  
  • at least as hard as sinkless orientation, \( \Omega(\log \log n) \)
  
  • upper bounds from prior work, e.g. \( \text{polylog}(n) \)
  
  • (recall Brook’s theorem)
## Intermediate LCL problems

<table>
<thead>
<tr>
<th></th>
<th>2-regular graphs</th>
<th>3-regular graphs</th>
</tr>
</thead>
<tbody>
<tr>
<td>2-colouring</td>
<td>( \Theta(n) )</td>
<td>( \Theta(n) )</td>
</tr>
<tr>
<td>3-colouring</td>
<td>( O(\log^* n) )</td>
<td><em>intermediate</em></td>
</tr>
<tr>
<td>4-colouring</td>
<td></td>
<td>( O(\log^* n) )</td>
</tr>
</tbody>
</table>