Local Algorithms on Grids

Jukka Suomela · Aalto University
“LCL Problems on Grids”, joint work with:

- Janne H Korhonen, Tuomo Lempiäinen, Christopher Purcell, Patric RJ Östergård (Aalto)
- Sebastian Brandt, Przemysław Uznański (ETH)
- Juho Hirvonen (Paris Diderot)
- Joel Rybicki (Helsinki)
Introduction
Setting

• Distributed graph algorithms

• *Input graph = computer network*
  • node = computer, edge = communication link
  • unknown topology

• Each node outputs its own part of solution
  • e.g. graph colouring: node outputs its own colour
Setting

- Deterministic distributed algorithms, **LOCAL** model of computing
  - unique identifiers
  - synchronous communication rounds
  - time = number of rounds until all nodes stop
  - unlimited message size, unlimited local computation
Setting

• Deterministic distributed algorithms, LOCAL model of computing

• Time = distance

• Algorithm with running time $T$: mapping from radius-$T$ neighbourhoods to local outputs
LCL problems

- LCL = locally checkable labelling

- Valid solution can be detected by checking $O(1)$-radius neighbourhood of each node
  - maximal independent set, maximal matching, vertex colouring, edge colouring …
LCL problems

• All LCL problems can be solved with $O(1)$-round *nondeterministic* algorithms
  • guess a solution, verify it in $O(1)$ rounds

• Key question: how fast can we solve them with *deterministic* algorithms?
  • cf. P vs. NP
Traditional settings

- **Directed cycles**
  - well understood

- **General (bounded-degree) graphs**
  - lots of ongoing work...
  - typical challenge: *expander-like constructions*
Our setting today

- **Oriented grids** (2D)
  - toroidal grid, $n \times n$ nodes, unique identifiers
  - consistent orientations north/east/south/west

- **Generalisation of directed cycles** (1D)

- Closer to real-world systems than expander-like worst-case constructions?
1D grids

- Vertex colouring

- **2-colouring:** global, $\Theta(n)$ rounds

- **3-colouring:** local, $\Theta(\log^* n)$ rounds
Why is 3-colouring $\Theta(\log^* n)$?

• Upper bound: one-round colour reduction
  • input: colouring with $2^k$ colours
  • output: colouring with $2k$ colours

• Lower bound: speed-up lemma
  • given: algorithm for $k$-colouring in time $T$
  • construct: algorithm for $2^k$-colouring in time $T - 1$
1D grids

• Vertex colouring

• **2-colouring:** global, $\Theta(n)$ rounds

• **3-colouring:** local, $\Theta(\log^* n)$ rounds
  • Cole–Vishkin (1986), Linial (1992)
2D grids

- Vertex colouring

- **2-colouring**: global, $\Theta(n)$ rounds

- **3-colouring**: ???

- **4-colouring**: ???

- **5-colouring**: local, $\Theta(\log^* n)$ rounds
2D grids

• Vertex colouring

• **2-colouring**: global, $\Theta(n)$ rounds

• **3-colouring**: global, $\Theta(n)$ rounds

• **4-colouring**: local, $\Theta(\log^* n)$ rounds

• **5-colouring**: local, $\Theta(\log^* n)$ rounds
Classification of LCL problems
LCL problems on grids

- $O(1)$ time: “trivial”
  - $o(\log^* n)$ time implies $O(1)$ time (Naor–Stockmeyer)
- $\Theta(\log^* n)$ time: “local”
- $\Theta(n)$ time: “global”

- Why *nothing between local and global*?
Normalisation

- **Setting:** LCL problems, 2D grids

- **Theorem:** Any $o(n)$-time algorithm can be translated to a “normal form”:
  1. fixed $\Theta(\log^* n)$-time component
  2. problem-specific $O(1)$-time component
Normalisation in more detail...

- For any problem $P$ of complexity $o(n)$, there are constants $k$ and $r$ and function $f$ such that $P$ can be solved as follows:
  - input: 2D grid $G$ with unique identifiers
  - find a maximal independent set in $G^k$
  - discard unique identifiers
  - apply function $f$ to each $r \times r$ neighbourhood
Some proof ideas...

• Given: $A$ solves $P$ in time $o(n)$ in $n \times n$ grids

• Solving $P$ in time $O(\log^* N)$ in $N \times N$ grids:
  • pick suitable $n = O(1)$, $k = O(1)$
  • find a maximal independent set (MIS) in $G^k$
  • use MIS to find *locally unique identifiers* for $n \times n$ neighbourhoods
  • simulate $A$ in $n \times n$ local neighbourhoods
LCL problems on grids

- $O(1)$ time: “trivial”
  - $o(\log^* n)$ time implies $O(1)$ time (Naor–Stockmeyer)

- $\Theta(\log^* n)$ time: “local”
  - $o(n)$ time implies $O(\log^* n)$ time (normalisation)

- $\Theta(n)$ time: “global”
Vertex colouring

• Every LCL problem is trivial, local, or global
• Why is 4-colouring in 2D grids “local”? 
• Why is 3-colouring in 2D grids “global”? 
4-colouring on grids
4-colouring

- Lucky guess: maybe it is local?

- Try to use computers to find normal form
  - turns out it is enough to find an MIS in $G^3$, then consider $7 \times 5$ tiles
  - algorithm $\approx$ mapping $\{0, 1\}^{7 \times 5} \rightarrow \{1, 2, 3, 4\}$
  - only 2079 possible tiles, easy to find a solution
3-colouring on grids
3-colouring

• Inherently different from 4-colouring:
  • cannot be solved locally

• But also different from 2-colouring:
  • nontrivial to argue that the problem is global
Proof idea

• **Assume:** a local algorithm for 3-colouring in $n \times n$ grids

• **Implication:** a local algorithm for “sum coordination” in $n$-cycles

• But we can prove that this problem is global
Consider any feasible 3-colouring…
We can convert it into a greedy solution in constant time
(eliminate colour 2 whenever possible, then colour 3)
Greedy solution: *boundaries + 2-coloured regions*
Parity changes at each boundary
Parity changes at each boundary
even × even

Wrap around: **same** parity

odd × odd

Wrap around: **opposite** parity
Boundaries can be *oriented* with local rules

(keep orange on right, white on left)
Pick any row, label *boundary crossings* with +1 / −1

up = +1, down = −1
Sum of crossings:

- **even** × even
  - Sum of crossings: **even**

- **odd** × odd
  - Sum of crossings: **odd**
even $\times$ even

Sum of crossings: even

odd $\times$ odd

Sum of crossings: odd
Boundaries are closed curves: *constant sum*

up = +1, down = −1
Locality: sum only depends on *grid dimensions*, not on IDs

(otherwise we could construct one instance with non-constant sum)
Sum coordination

• What any 3-colouring algorithms has to solve for every row of the grid:
  • label nodes with \{+1, 0, −1\}
  • there is some function \(q\) so that the sum of labels is \(q(n)\) in any \(n\)-cycle, regardless of unique identifiers
  • \(q(n)\) odd iff \(n\) is odd: cannot label everything with 0
  • \(|q(n)|\) not too large: cannot label everything with +1
Sum coordination

• What any 3-colouring algorithms has to solve for every row of the grid

• Requires global coordination
Conclusions
2-colouring  

3-colouring  

4-colouring  

global  

global  

local
Conclusions: LCLs on grids

• Only three complexity classes in 2D grids: trivial $O(1)$, local $\Theta(\log^* n)$, global $\Theta(n)$

• 4-colouring is local: algorithm synthesis

• 3-colouring is global: sum coordination

• Can be generalised to $d$-dimensional grids!