On the power of preprocessing in decentralized network optimisation

Klaus-Tycho Foerster, University of Vienna
Juho Hirvonen, Aalto University
Stefan Schmid, University of Vienna
Jukka Suomela, Aalto University

INFOCOM 2019, Paris, France
Distributed systems
and locality
Locality

Everybody’s favourite network topology, the ring
Locality

Problem: 2-coloring
Locality: 2-coloring

Each computer must decide its own color
Locality: 2-coloring

Each computer must decide its own color
Locality: 2-coloring

Once a color is fixed, it is propagated

same color!
Locality: 3-coloring

What if we have one extra color?

use the third color!
Locality

- **2-coloring** a ring is inherently *global*: each node must see the *whole network* in order to decide its color.

- **3-coloring** a ring is inherently *local*: a *greedy approach* works, nodes need only to avoid the colors of their neighbors.

We want to understand the *locality* of problems.
This talk

Theory warning!

1. Modelling the concept of **locality**

2. **Recent developments** in theoretical understanding

3. **Transferring** the understanding to the context of networking (e.g. *distributed SDN control plane*)
Modelling and understanding locality
Modeling locality

- LOCAL model of Linial (SICOMP, 1992)
- Model locality by *abstracting away* other aspects of distributed computing
  - Synchronous communication rounds
  - Unbounded messages
  - Free of faults, crashes, byzantine behavior
  - Static network, no dynamic changes
Locality and time

- In $T$ synchronous rounds flooding collects all information inside $T$-hop neighbourhood.
- In particular, no information outside the $T$-hops!

$\text{time} = \text{distance}$
Locality and time

Complexity
= number of communication rounds (time)
= radius of each node’s view (distance)

time = distance
Locality of some problems

- Classic symmetry breaking problems are local: MIS, MM, \((\Delta+1)\)-coloring in \(O(\Delta + \log^* n)\) rounds*

- 2-coloring, MST, spanners, leader election are global, require diameter time

- optimization, new "intermediate" problems in polylog in \(n\) time

- everything in diameter time

\[\Delta = \text{maximum degree}\]
\[\log^* (\text{number of atoms in the observable universe}) = 5\]
Algorithmic model?

Asynchronous: use synchronisers

Limited bandwidth: algorithms often don’t abuse this (e.g. coloring, network decomposition with $O(\log n)$-bit messages)

Fault-tolerance: efficient distributed algorithms stabilise quickly after faults, dynamic changes

However, e.g. triangle detection trivial in LOCAL
Impossibility results

• Powerful model implies very general negative results

• Results apply in the presence of congestion, faults, asynchrony, byzantine behaviour, ...

• Upper bounds show whether tasks are locality constrained
Impossibility results

- Powerful model implies very general negative results
- A number of recent developments
  - Simulation speedup for intermediate problems (Brandt et al., STOC 2016)
  - Simulation gap and derandomization (Chang et al., FOCS 2016)
  - SLOCAL-completeness (Ghaffari et al., STOC 2017)
  - Derandomization (Ghaffari et al., FOCS 2018)
  - Simulation speedup for maximal matching (Balliu et al., 2019)
Locally checkable labelings

- Balliu et al. 2018a
- Balliu et al. 2018b
- Chang & Pettie 2017
- Ghaffari et al. 2018
- Balliu et al. 2019
- Chang & Pettie 2017
- Fischer & Ghaffari 2017
- Brandt et al. 2016
- Chang et al. 2016
- Ghaffari & Su 2017
- Chang et al. 2016
- Chang & Pettie 2017
- Naor & Stockmeyer 1995
- Cole & Vishkin 1986
- Linial 1992
- Naor 1991

- \( \log \log n \)
- \( \log n \)
- \( n \)
Locality, networking, preprocessing
Modelling locality in networking

• We study a particular model, the supported LOCAL model of Schmid and Suomela (HotSDN, 2013)

• Inspiration e.g. a distributed control plane in SDN
  • The physical network known in advance
  • The global logical state of the network unknown

• Also a study on the power of preprocessing
Supported LOCAL

support = graph known to all nodes
Supported LOCAL

network, logical graph = subgraph of the support
Supported LOCAL

- **At least as powerful** as the LOCAL model
  - The input is a subgraph of the globally and consistently known support

- Are the "removed" edges available for communication?
  - Affects computational power
  - *active / passive* model
The Bad, The Good

- Support is not useful in some corner cases
- Let’s make a wild assumption: we have some degree of control over the network…
  - We can actually **design** the network?
  - The switches have a **finite** number of ports?
Our work

• The support can be used to **precompute** various useful **primitives**, e.g.
  • *coloring*
  • *network decomposition*
  • *spanning tree*

• Support particularly useful if it has nice structure
Coloring

- In networks of e.g. bounded maximum degree, colorings are a useful primitive

- many problems solvable in constant time given a coloring (i.e. independent of the network size)
Coloring
Coloring

Coloring of the support is coloring of the input!
Colorings

- **Coloring** → greedy algorithms (e.g. maximal matching, maximal independent set, (Δ+1)-coloring)
- **Distance-T coloring** → simulate and speed up **LOCAL**
- **Distance-T coloring** → simulate **SLOCAL**
Special graph classes

- Support with **small chromatic number** is useful.

- **Planar graphs** are particularly useful (4-colorable, large degree)
  - Case study: approximation of *minimum dominating set*
  - Use preprocessing to **speed up subroutines** in existing distributed algorithms
  - **(1+\(\epsilon\))-approximation** in **constant time**
Network decomposition

$O(\log n)$ colors, $O(\log n)$ diameter
Network decomposition

• Useful primitive in the case of large degrees (coloring a special case!)

• All edges must be available for communication to be useful (removing edges affects cluster diameter)

• Simulation of the SLOCAL model of Ghaffari et al. (STOC 2016)
  • PSLOCAL-completeness: supported LOCAL closes the gap between randomised and deterministic
  • Symmetry breaking in polylog time
Impossibility in the supported LOCAL

- Example: **Hardness of approximation** for *maximum cut*
  - 2 vertex labels, edge is *cut*, if endpoints have different labels
  - optimum cannot be found in $o(\log n)$ rounds
  - hard even in the **active model** with **bounded degrees**
Proof sketch: "hide" subgraphs with large and small optima in the support s.t. locally you cannot know which one has been selected.
Concluding

• **Understanding of locality** in distributed message passing has developed significantly in recent years.

• This understanding can be extended to **models of networking**
  • Lot of work **still left!**

• Network topology **can be designed** to improve the locality of distributed algorithms