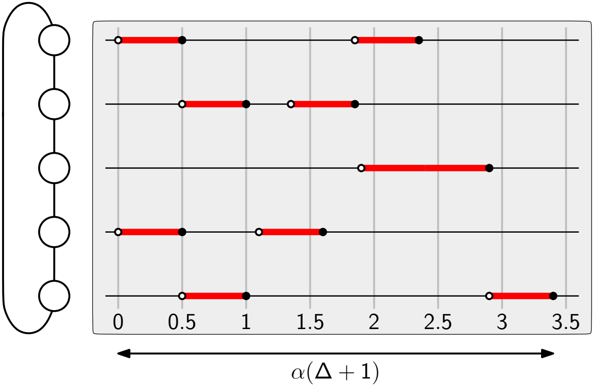
Deterministic Local Algorithms, Unique Identifiers, and Fractional Graph Colouring

Henning Hasemann, Juho Hirvonen,
Joel Rybicki, and Jukka Suomela

TU Braunschweig
University of Helsinki

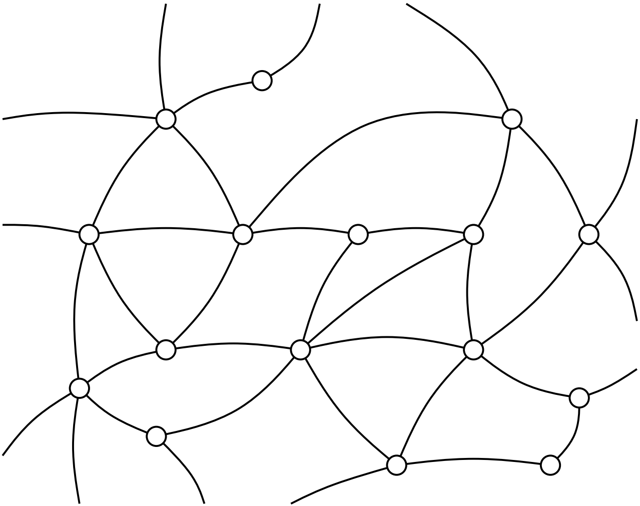
SIROCCO 2012
30 June 2012

Our Result



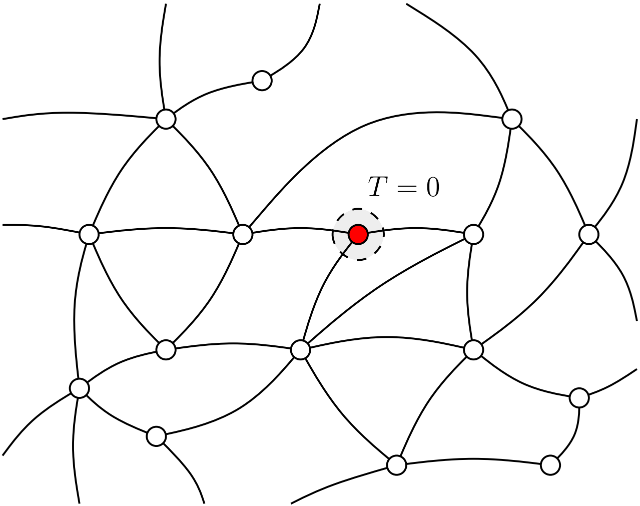
There is a deterministic distributed algorithm that runs in 1 communication round that, for any $\alpha > 1$, finds a fractional graph colouring of length at most $\alpha(\Delta + 1)$

Model of Computation: LOCAL



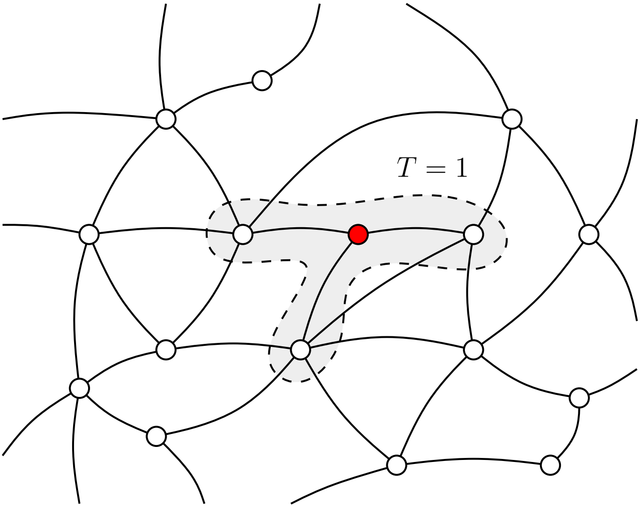
- Communication graph

Model of Computation: LOCAL



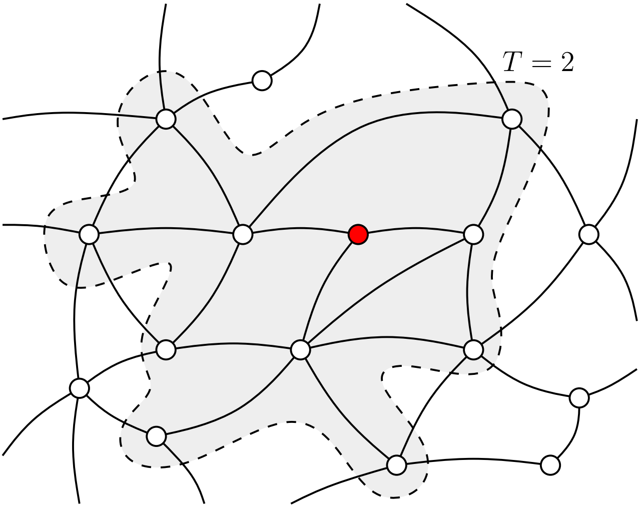
- Synchronous communication

Model of Computation: LOCAL



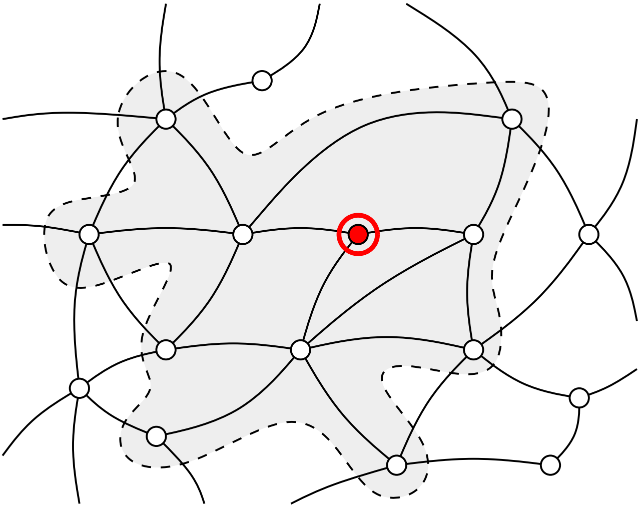
- Synchronous communication

Model of Computation: LOCAL



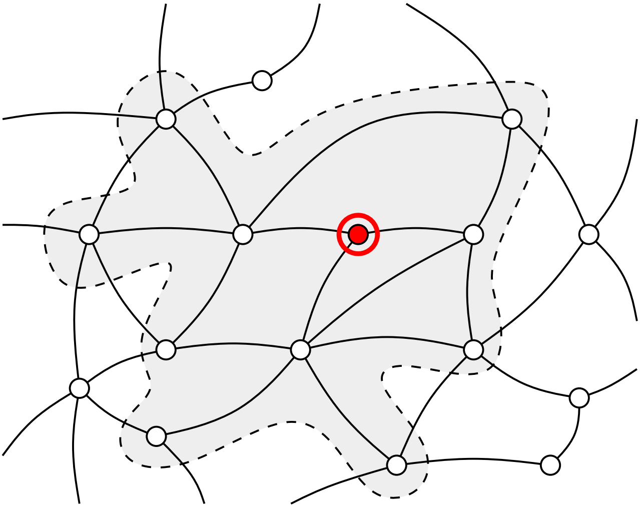
- In T rounds gather radius-T neighbourhood

Model of Computation: LOCAL



- Constant-time algorithms

Model of Computation: LOCAL

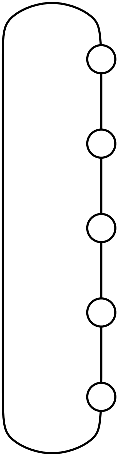


- Each node maps neighbourhood to output

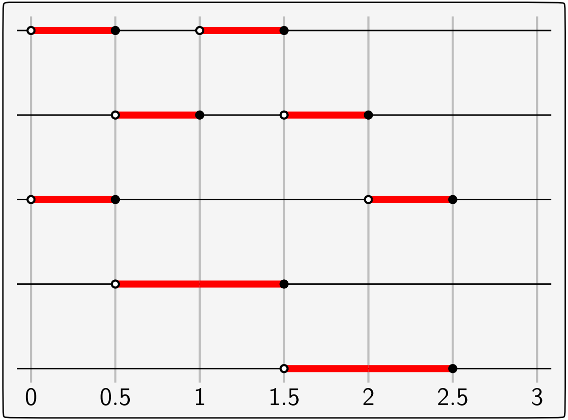
Fractional Graph Colouring

Fractional Graph Colouring

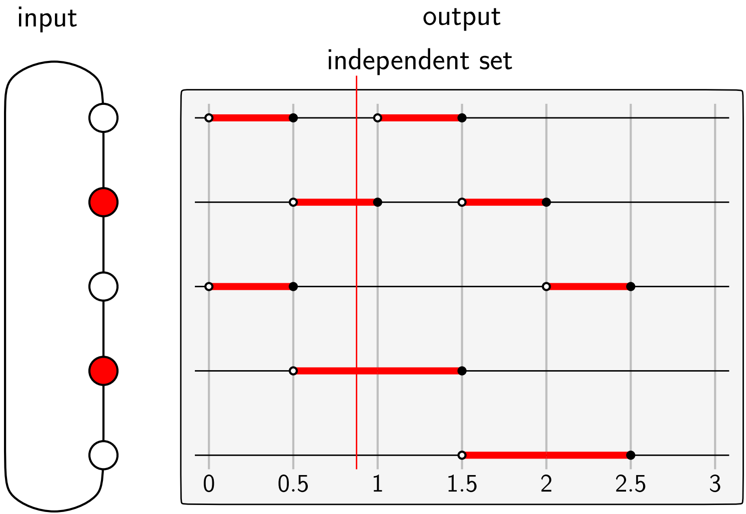
input



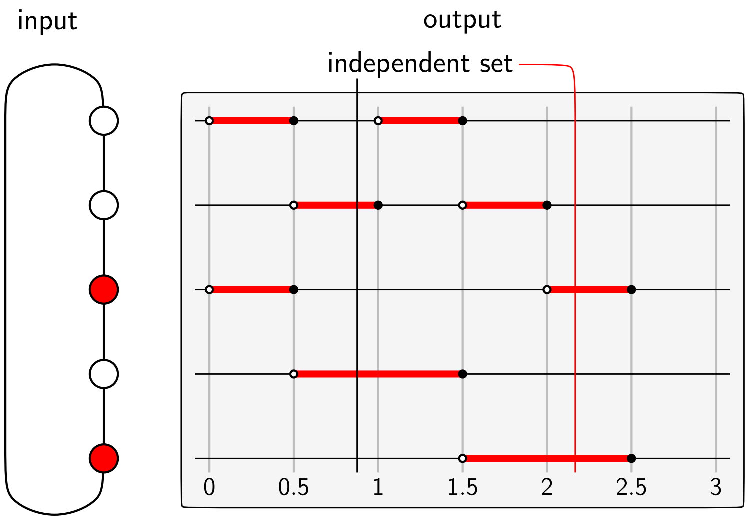
output



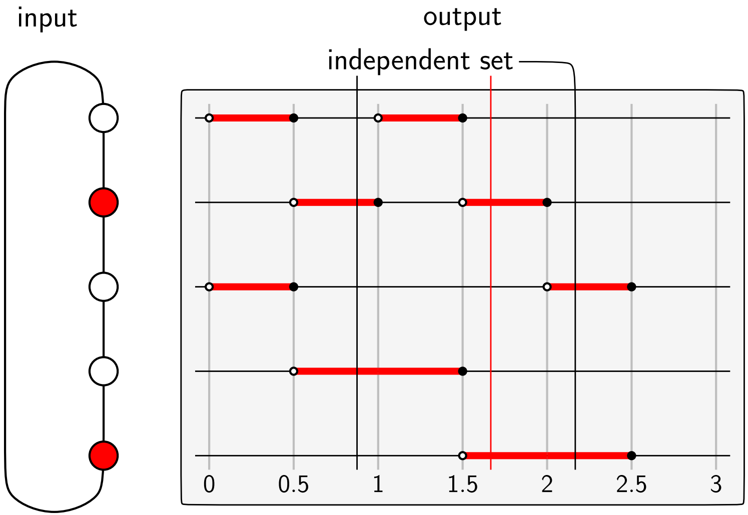
Fractional Graph Colouring



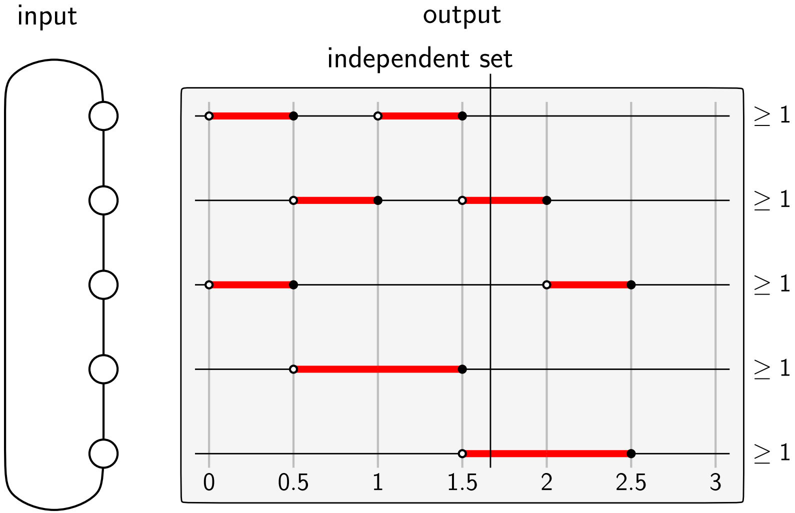
Fractional Graph Colouring



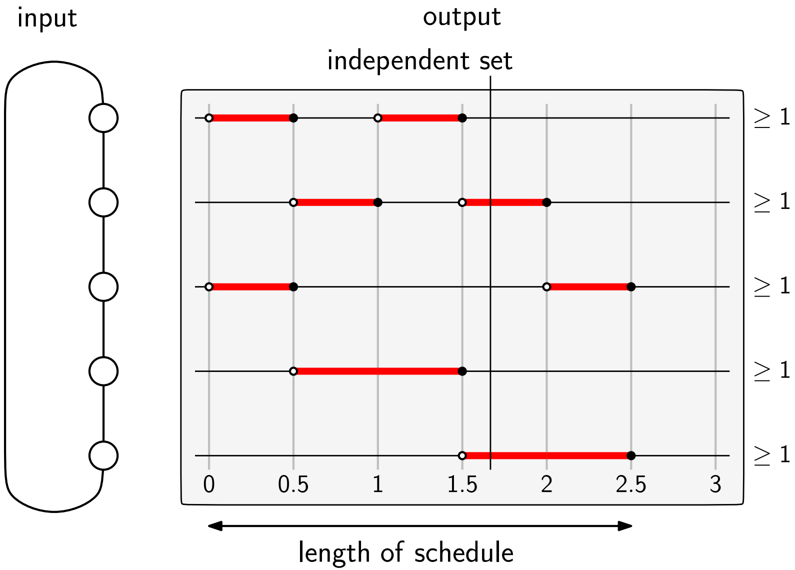
Fractional Graph Colouring



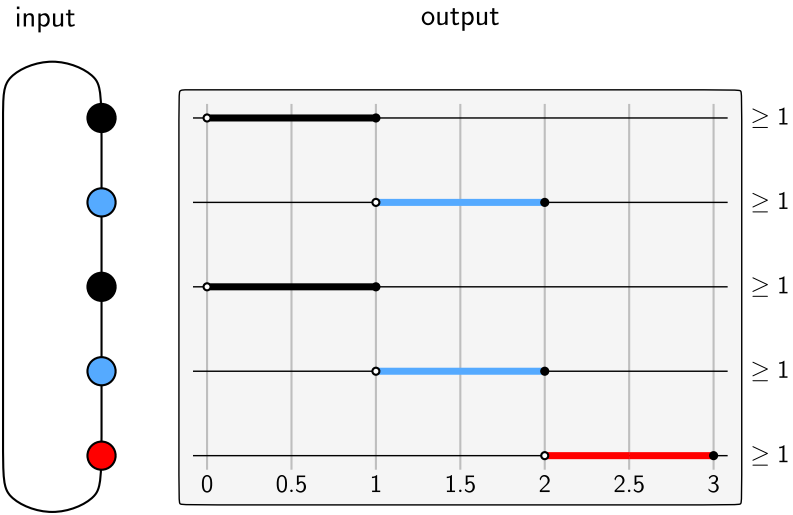
Fractional Graph Colouring



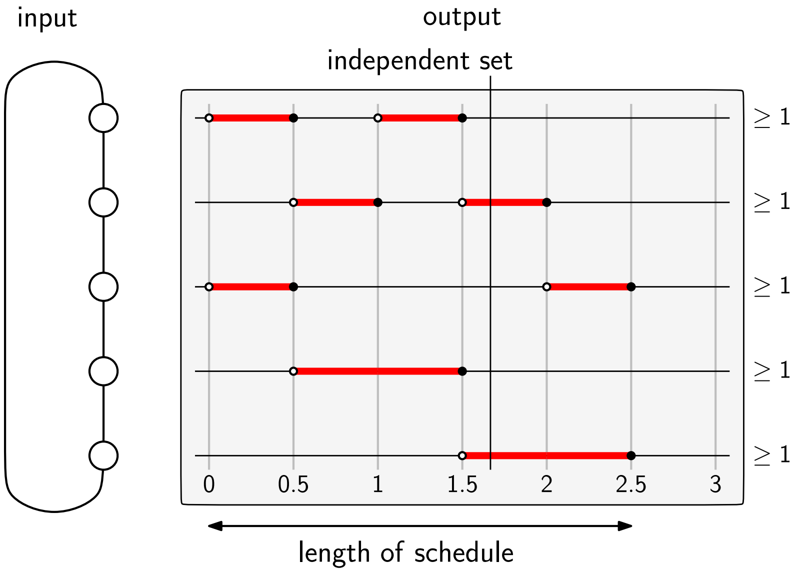
Fractional Graph Colouring



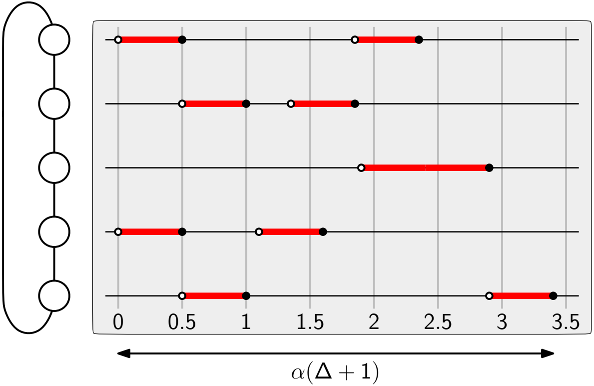
Fractional Graph Colouring



Fractional Graph Colouring

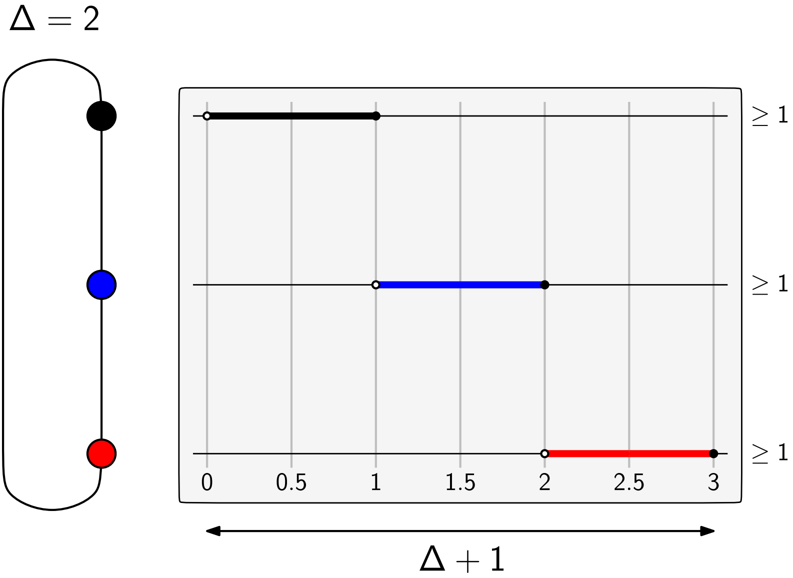


Our Result Again



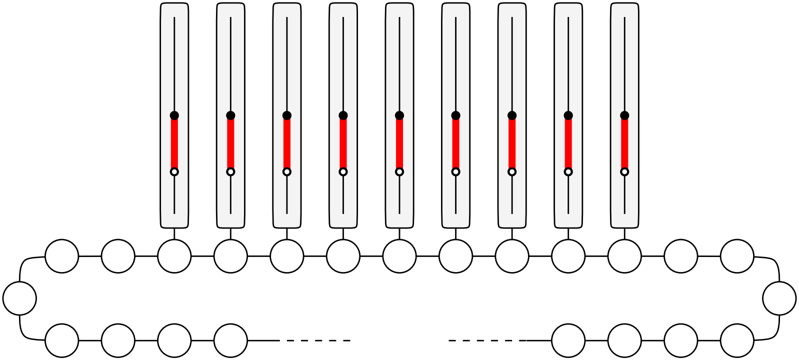
There is a deterministic distributed algorithm that runs in 1 communication round that, for any $\alpha > 1$, finds a fractional graph colouring of length at most $\alpha(\Delta + 1)$

Lower Bound



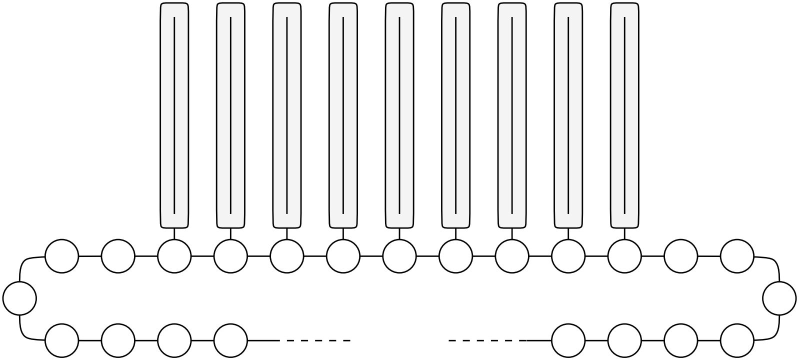
Finding a Fractional Graph Colouring

Finding a Fractional Graph Colouring



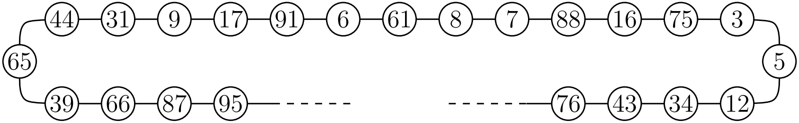
- Impossible to break symmetry in an anonymous cycle

Finding a Fractional Graph Colouring



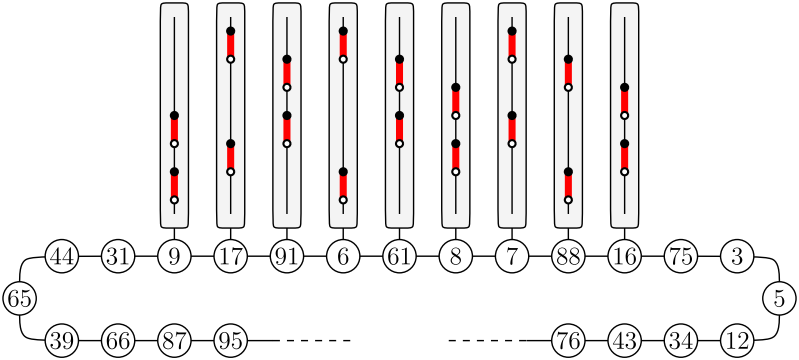
- Nodes must produce an empty schedule

Finding a Fractional Graph Colouring



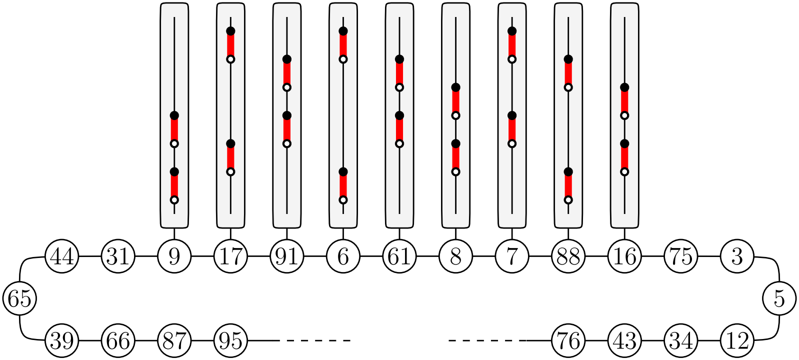
- Standard assumption: numeric identifiers

Finding a Fractional Graph Colouring



- Standard assumption: numeric identifiers

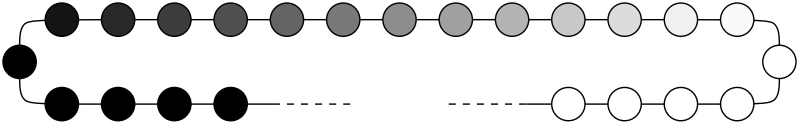
Finding a Fractional Graph Colouring

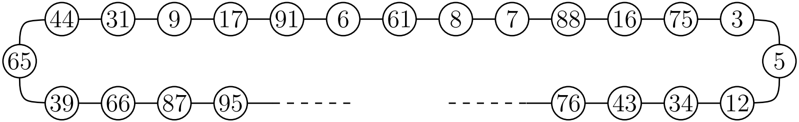


- Standard assumption: numeric identifiers
- FGC is the first example where numeric identifiers give a constant-time algorithm

Why Numeric Identifiers Do Not Help?

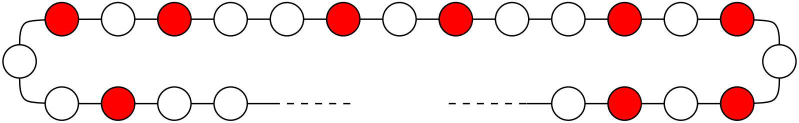
Numeric Identifiers Not Needed





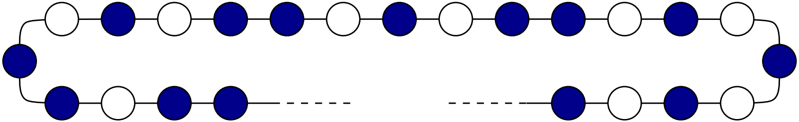
- Naor & Stockmeyer (1995) studied when numeric identifiers are necessary
- LCL-problems

LCL-problems



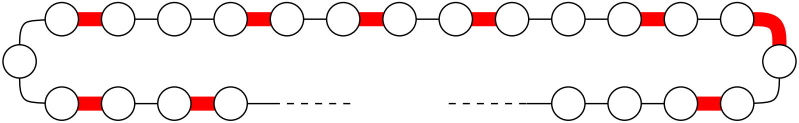
- Maximal Independent Set

LCL-problems



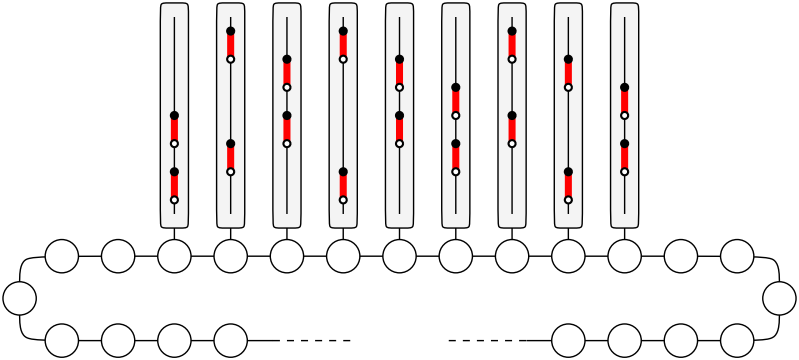
- Vertex Cover

LCL-problems



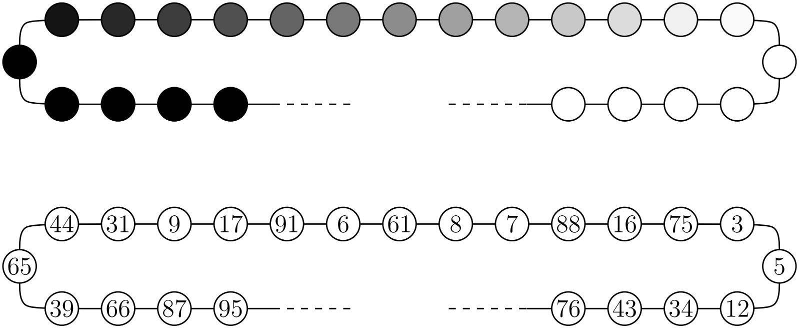
- Maximal Matching

LCL-problems



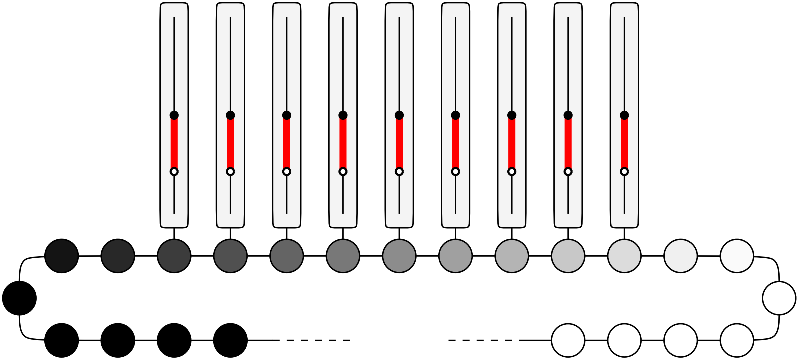
- Fractional Graph Colouring

Numeric Identifiers Not Needed



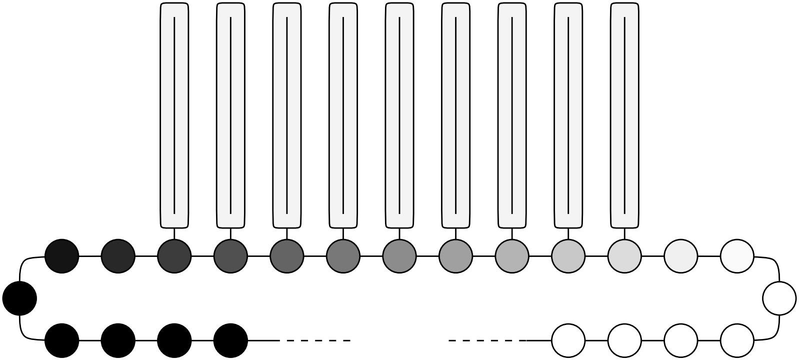
- Naor & Stockmeyer (1995): In LCL-problems numeric identifiers not necessary
  - Technicality: applies if output bounded

No FGC with Comparisons



- Identifiers arranged in an ascending order

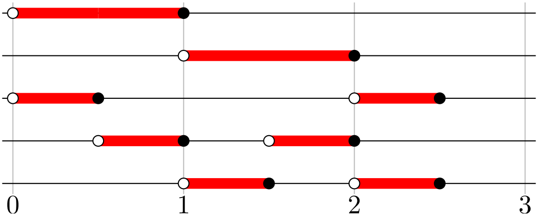
No FGC with Comparisons



- Some nodes must produce an empty schedule

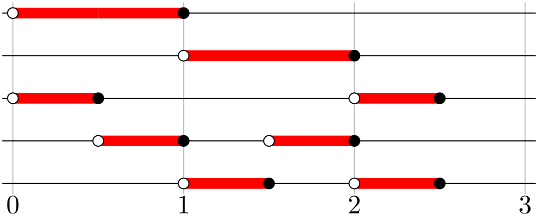
Why Numeric Identifiers Help with FGC?

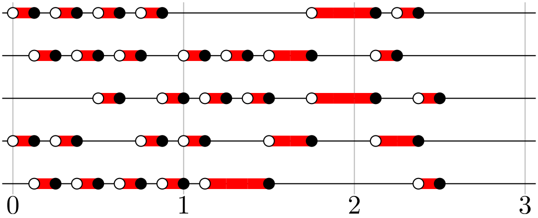
Non-constant output



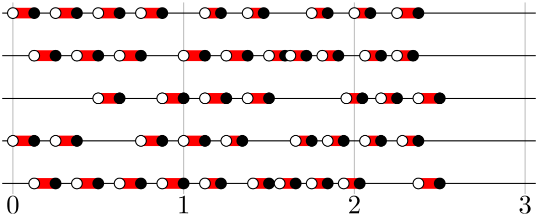
- In FGC natural encoding of solution not bounded in size

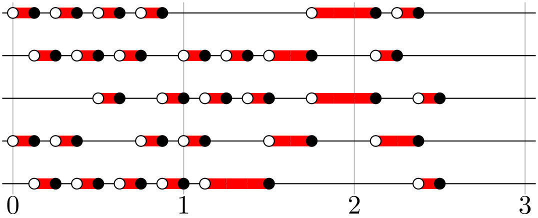
Non-constant output





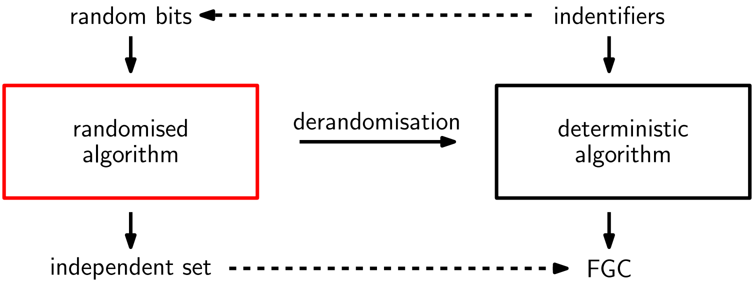
Non-constant output





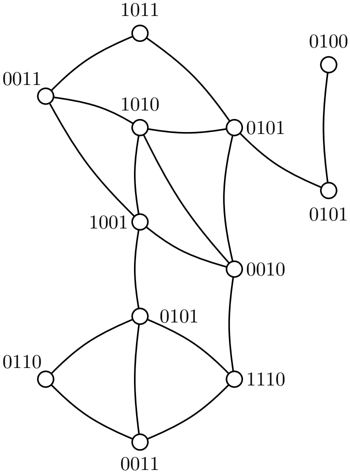
The Algorithm

Algorithm Design Idea



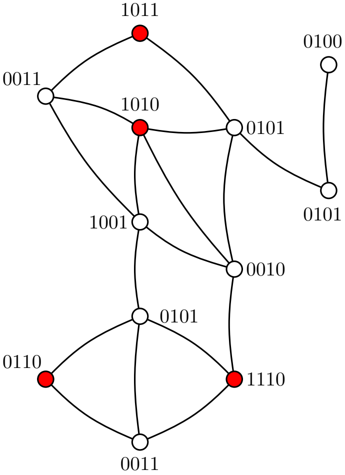
- Use a randomised independent set algorithm as a black box
- Iterate over possible random bit strings for the black box to get a deterministic algorithm

A Randomised Algorithm



- A randomised algorithm for the independent set problem
- Each node gets a random bit string

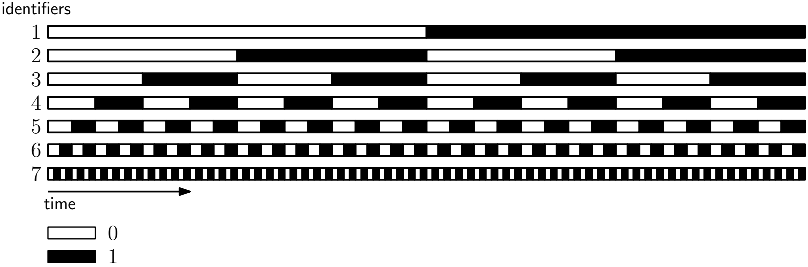
A Randomised Algorithm



- Local maxima join the independent set
- Guarantee: each node v joins with probability at least

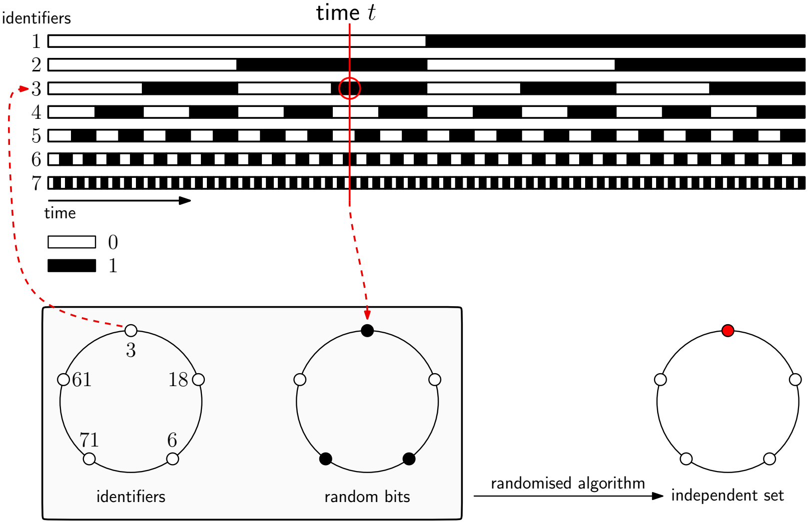
$$\frac{1-\varepsilon}{\deg(v)+1}$$

Deterministic Algorithm (Oversimplified)



- Simulate the random algorithm by iterating over all combinations of inputs
- Encoding of the output grows with size of the network
- By Naor & Stockmeyer, dependence on n is necessary

Deterministic Algorithm (Oversimplified)



Tradeoffs

Granularity Tradeoff

granularity	unbounded
length	$O(1)$
running time	$O(1)$

- Any two can be kept constant in bounded degree graphs
- Constant running time and length of schedule
  - This work
  - granularity of schedule grows with size of the network

Running Time Tradeoff

granularity	$O(1)$
length	$O(1)$
running time	$\log^* n$

- Any two can be kept constant in bounded degree graphs
- Constant length of schedule and granularity
  - find a $(\Delta + 1)$-colouring in $O(\log^* n)$ rounds

Length of Schedule Tradeoff

granularity	$O(1)$
length	$\text{poly}(n)$
running time	$O(1)$

- Any two can be kept constant in bounded degree graphs
- Constant running time and granularity
  - node of colour $c(v)$ is active during time interval

    $$\big(c(v) - 1, c(v)\big]$$

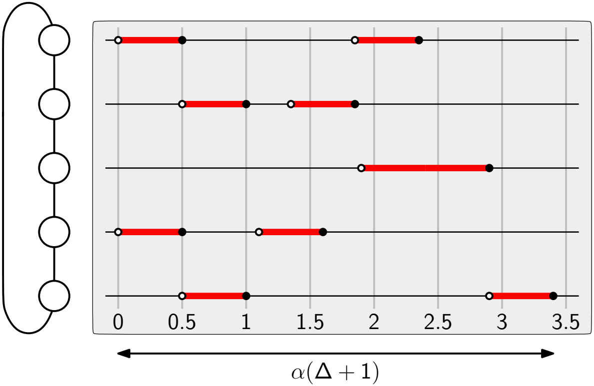
  - length of schedule $\text{poly}(n)$

Time-Length-Granularity Tradeoff—Summary

granularity	$O(1)$
length	$O(1)$
running time	$O(1)$

- Impossible to have constant running time, length and granularity at the same time
  - Naor & Stockmeyer

Our Result



There is a deterministic distributed algorithm that runs in 1 communication round that, for any $\alpha > 1$, finds a fractional graph colouring of length at most $\alpha(\Delta + 1)$