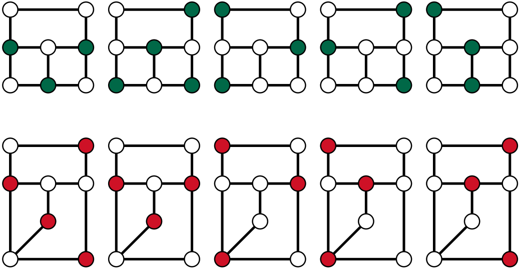
Local approximation algorithms for scheduling problems in sensor networks

Patrik Floréen, Petteri Kaski, Topi Musto, Jukka Suomela

Helsinki Institute for Information Technology HIIT
Department of Computer Science
University of Helsinki
Finland

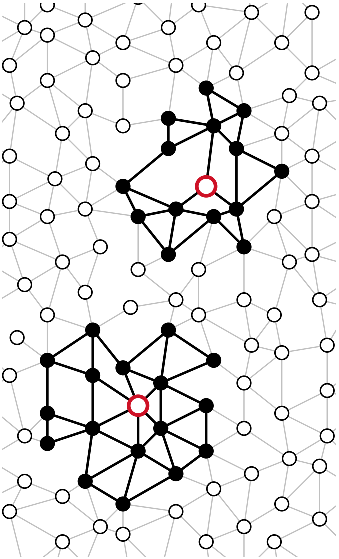
Algosensors
14 July 2007



Local algorithms

- Operation of a node only depends on **input within its constant-size neighbourhood**
- Extreme scalability: **constant** amount of communication, memory and computation per node
- Weak model: 3-colouring a cycle impossible (Linial 1992)

Our result: local algorithms can be used to approximate nontrivial **scheduling problems**



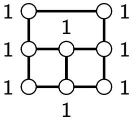
Sleep scheduling

Input: **redundancy graph**, battery capacities

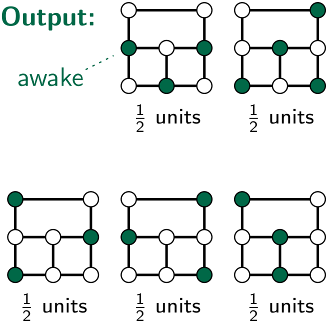
- Set of awake nodes = **dominating set** of redundancy graph
- Associate a time period with each dominating set
- **Maximise** total length
- Obey battery constraints

Motivation: maximising lifetime of a sensor network (pairwise redundancy)

Input:



Output:



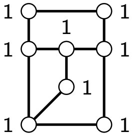
Activity scheduling

Input: **conflict graph**,
activity requirements

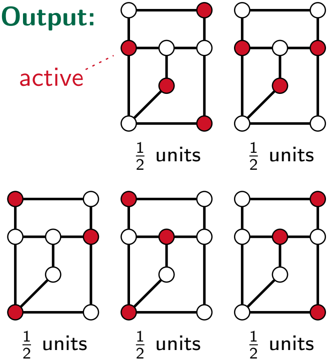
- Set of active nodes = **independent set** of conflict graph
- Associate a time period with each independent set
- **Minimise** total length
- Fulfil activity requirements

Motivation: minimising makespan of radio transmissions (pairwise interference)

Input:



Output:



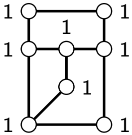
Scheduling problems

Sleep scheduling: generalisation of fractional domatic partition

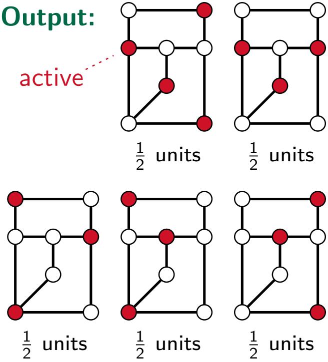
Activity scheduling: generalisation of fractional graph colouring

- Linear programs
- The size of the LP can be exponential in the size of the graph
- **Hard to solve and approximate** in general graphs

Input:



Output:

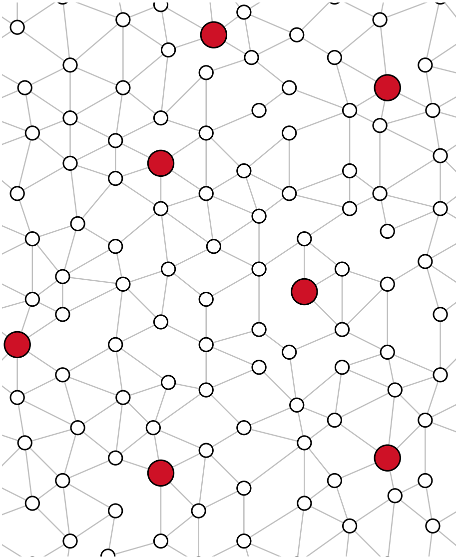


Solution

- Hard problems
- Weak model of computation

Solution: **markers**

1. Markers break symmetry
2. Characterisation of marker distribution constrains the family of graphs

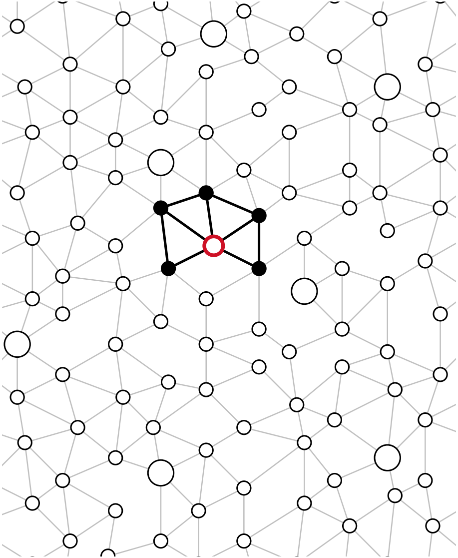


Marked graphs

$(\Delta, \ell_1, \ell_\mu, \mu)$-marked graph:

- Degree $\leq \Delta$
- ≥ 1 marker within ℓ_1 hops from any node
- $\leq \mu$ markers within ℓ_μ hops from any node

Intuition: bounded growth, symmetry-breakers nearby

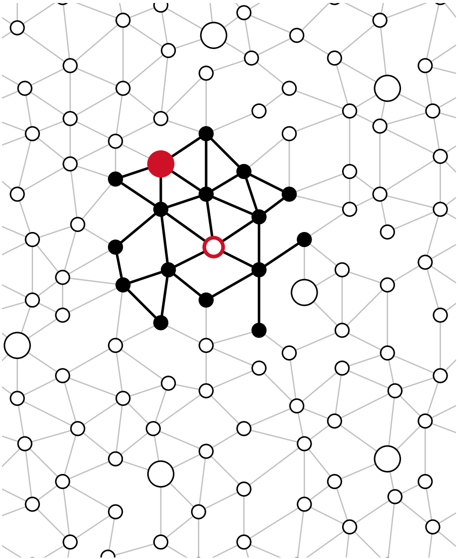


Marked graphs

$(\Delta, \ell_1, \ell_\mu, \mu)$-marked graph:

- Degree $\leq \Delta$
- ≥ 1 marker within ℓ_1 hops from any node
- $\leq \mu$ markers within ℓ_μ hops from any node

Intuition: bounded growth, symmetry-breakers nearby

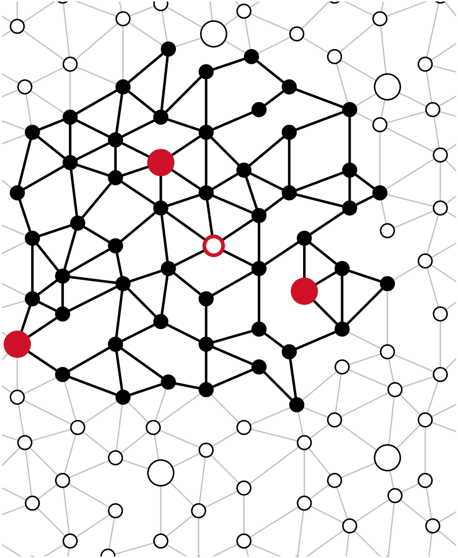


Marked graphs

$(\Delta, \ell_1, \ell_\mu, \mu)$-marked graph:

- Degree $\leq \Delta$
- ≥ 1 marker within ℓ_1 hops from any node
- $\leq \mu$ markers within ℓ_μ hops from any node

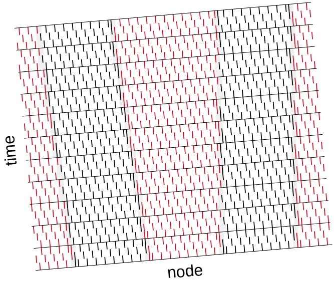
Intuition: bounded growth, symmetry-breakers nearby

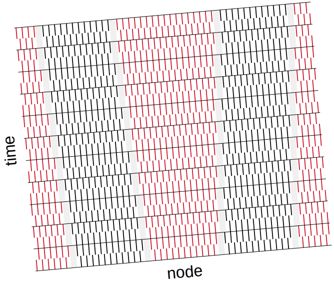


Main results

Local $(1+\epsilon)$-approximation
algorithm for sleep scheduling
in $(\Delta, \ell_1, \ell_\mu, \mu)$-marked graphs
for any $\epsilon > 4\Delta/\lfloor(\ell_\mu - \ell_1)/\mu\rfloor$

Local $1/(1-\epsilon)$-approximation
algorithm for activity scheduling
in $(\Delta, \ell_1, \ell_\mu, \mu)$-marked graphs
for any $\epsilon > 4/\lfloor(\ell_\mu - \ell_1)/\mu\rfloor$





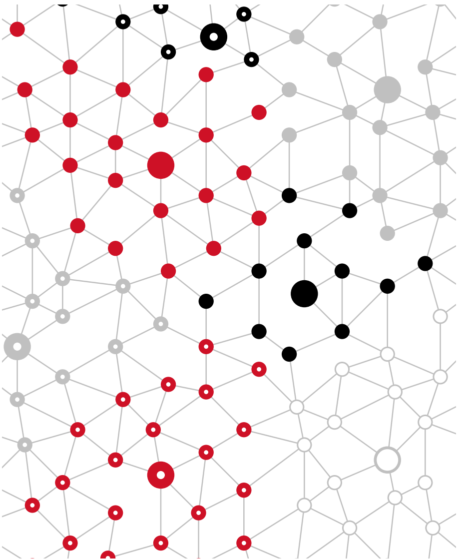
- Markers are enough:
  no coordinates needed
- Markers are necessary
- Cannot improve ϵ by factor 9

Algorithm sketch

Several partitions of communication graph

- Configuration 0: Voronoi cells for markers
- Configuration 1: shift cell borders
- Configuration *i*: shift *i* units

Solve the scheduling problem locally for each cell, interleave the solutions

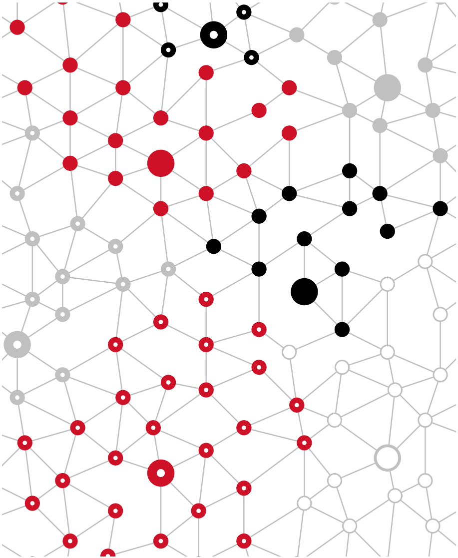


Algorithm sketch

Several partitions of communication graph

- Configuration 0: Voronoi cells for markers
- Configuration 1: shift cell borders
- Configuration i: shift i units

Solve the scheduling problem locally for each cell, interleave the solutions

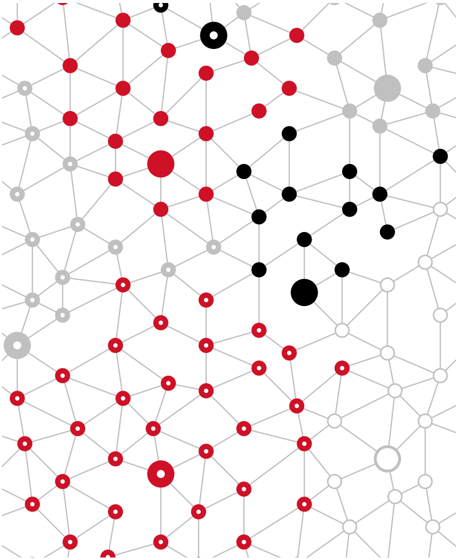


Algorithm sketch

Several partitions of communication graph

- Configuration 0: Voronoi cells for markers
- Configuration 1: shift cell borders
- Configuration *i*: shift *i* units

Solve the scheduling problem locally for each cell, interleave the solutions



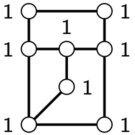
Summary

- **Local** approximation scheme: constant effort per node
- Fractional scheduling problems, both **packing** and **covering**
- Can be extended beyond pairwise redundancy/conflicts as long as there is "locality"
- **Markers** are enough, coordinates not needed
- Constants are not practical, more work needed

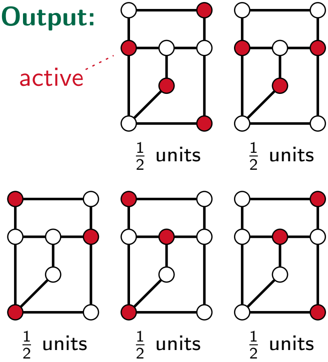
http://www.hiit.fi/ada/geru

jukka.suomela@cs.helsinki.fi

Input:



Output:



Appendix: Examples of marked graphs

- 2-dimensional grid of nodes
  - Use a sparser grid to place the markers
  - "Local approximation scheme":
    any approximation ratio by using a sparse enough grid
    (cost: higher computational complexity)
- "Coarse grids", graphs quasi-isometric to 2-dimensional grids
  - Arbitrary small-scale structure
- Cutting parts of coarse grids, with $L + 1$ hop margins
  - Arbitrary small-scale and large-scale structure
  - **Medium-scale structure has similarities with low-dimensional grids**

Appendix: Sleep scheduling LP

Input:

– communication graph $\mathcal{G}$
– redundancy graph $\mathcal{R}$, subgraph of $\mathcal{G}$
– battery capacity $b(v) \geq 0$ for each node $v \in V_\mathcal{R}$

Task:

maximise $\sum_D x(D)$

subject to $\sum_D D(v)x(D) \leq b(v)$ and $x(D) \geq 0$

v ranges over $V_\mathcal{R}$
D ranges over dominating sets of $\mathcal{R}$

$D(v) = 1$ if $v \in D$ and $D(v) = 0$ if $v \notin D$
$x(D) =$ the length of the time period associated with D

Appendix: Activity scheduling LP

Input:

– communication graph $\mathcal{G}$
– conflict graph $\mathcal{C}$, subgraph of $\mathcal{G}$
– activity requirement $a(v) \geq 0$ for each node $v \in V_{\mathcal{C}}$

Task:

minimise $\sum_I x(I)$

subject to $\sum_I I(v)x(I) \geq a(v)$ and $x(I) \geq 0$

v ranges over $V_{\mathcal{C}}$
I ranges over independent sets of $\mathcal{C}$

$I(v) = 1$ if $v \in I$ and $I(v) = 0$ if $v \notin I$
$x(I) =$ the length of the time period associated with I