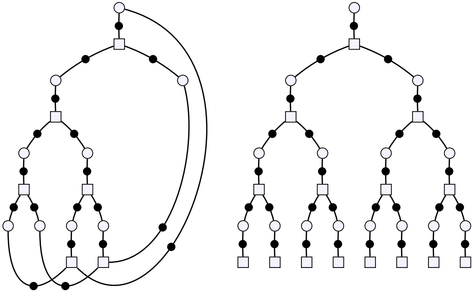
Local algorithms and max-min linear programs

Patrik Floréen,
Marja Hassinen,
Joel Kaasinen,
Petteri Kaski,
Topi Musto,
Jukka Suomela

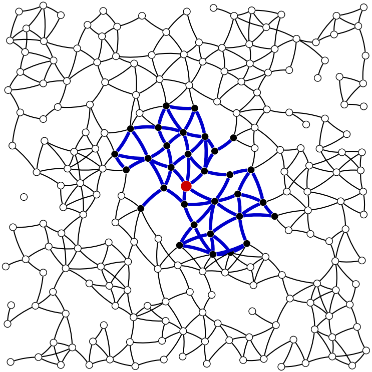
HIIT,
University of Helsinki,
Finland



TU Braunschweig
11 September 2008

Local algorithms

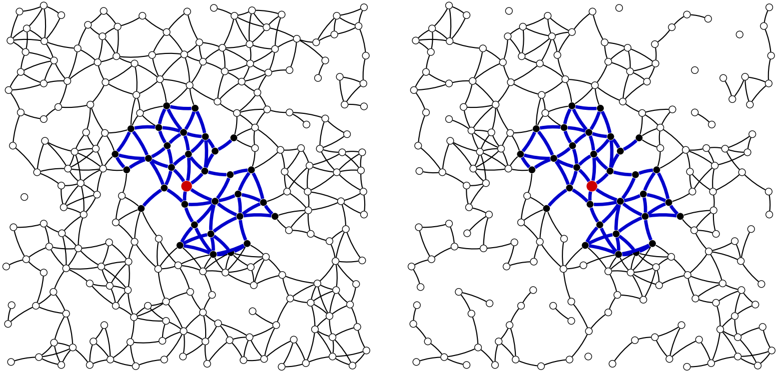
Local algorithm: output of a node is a function of input within its *constant-radius neighbourhood*



(Linial 1992; Naor and Stockmeyer 1995)

Local algorithms

Local algorithm: changes outside the *local horizon* of a node do not affect its output



(Linial 1992; Naor and Stockmeyer 1995)

Local algorithms

Local algorithms are efficient:

- Space and time complexity is constant per node
- Distributed constant time (even in an infinite network)

. . . and fault-tolerant:

- Topology change only affects a constant-size part
  (Naor and Stockmeyer 1995)
- Can be turned into self-stabilising algorithms
  (Awerbuch and Sipser 1988; Awerbuch and Varghese 1991)

(In this presentation, we assume bounded-degree graphs)

Local algorithms

Applications beyond distributed systems:

- Simple linear-time centralised algorithm
- In some cases randomised, approximate sublinear-time algorithms (Parnas and Ron 2007)

Consequences in theory of computing:

- Bounded-fan-in, constant-depth Boolean circuits: in NC^0
- Insight into algorithmic value of information (cf. Papadimitriou and Yannakakis 1991)

Local algorithms

Great, but do they exist? Fundamental hurdles:

1. Breaking the symmetry:
   e.g., colouring a ring of identical nodes
2. Non-local problems:
   e.g., constructing a spanning tree

Strong negative results are known:

- 3-colouring of n-cycle not possible,
  even if unique node identifiers are given (Linial 1992)
- No constant-factor approximation of vertex cover, etc.
  (Kuhn et al. 2004; Kuhn 2005)

Local algorithms

Side information

Many positive results are known,
if we assume some side information
(e.g., coordinates, clustering)

(Czyzowicz et al. 2008; Floréen et al. 2007; Hassinen et al. 2008;
Urrutia 2007; Wang and Li 2006; Wiese and Kranakis 2008; . . .)

Side information helps to break the symmetry

But what if we have no side information?

Local algorithms

Some previous positive results:

- Locally checkable labellings (Naor and Stockmeyer 1995)
- Dominating set
  (Kuhn and Wattenhofer 2005; Lenzen et al. 2008)
- Packing and covering LPs
  (Papadimitriou and Yannakakis 1993; Kuhn et al. 2006)

Present work:

- Max-min LPs (Floréen et al. 2008a,b,c,d)

Max-min linear program

Let $A \geq 0$, $\mathbf{c}_k \geq \mathbf{0}$

Objective:

$$\begin{aligned} \text{maximise} \quad & \min_{k \in K} \mathbf{c}_k \cdot \mathbf{x} \\ \text{subject to} \quad & A\mathbf{x} \leq \mathbf{1}, \\ & \mathbf{x} \geq \mathbf{0} \end{aligned}$$

Generalisation of packing LP:

$$\begin{aligned} \text{maximise} \quad & \mathbf{c} \cdot \mathbf{x} \\ \text{subject to} \quad & A\mathbf{x} \leq \mathbf{1}, \\ & \mathbf{x} \geq \mathbf{0} \end{aligned}$$

Max-min linear program

Let $A \geq 0$, $C \geq 0$

Equivalent formulation:

$$\begin{aligned} \text{maximise} \quad & \omega \\ \text{subject to} \quad & A\mathbf{x} \leq \mathbf{1}, \\ & C\mathbf{x} \geq \omega\mathbf{1}, \\ & \mathbf{x} \geq \mathbf{0} \end{aligned}$$

Applications: mixed packing and covering, linear equations

$$\text{find } \mathbf{x} \text{ s.t. } \begin{aligned} A\mathbf{x} &\leq \mathbf{1}, \\ C\mathbf{x} &\geq \mathbf{1}, \\ \mathbf{x} &\geq \mathbf{0} \end{aligned} \qquad\qquad \text{find } \mathbf{x} \text{ s.t. } \begin{aligned} A\mathbf{x} &= \mathbf{1}, \\ \mathbf{x} &\geq \mathbf{0} \end{aligned}$$

Max-min linear program

Distributed setting:

- one node $v \in V$ for each variable x_v,
  one node $i \in I$ for each constraint $\mathbf{a}_i \cdot \mathbf{x} \leq 1$,
  one node $k \in K$ for each objective $\mathbf{c}_k \cdot \mathbf{x}$
- $v \in V$ and $i \in I$ adjacent if $a_{iv} > 0$,
  $v \in V$ and $k \in K$ adjacent if $c_{kv} > 0$

$$\begin{aligned} \text{maximise} \quad & \min_{k \in K} \mathbf{c}_k \cdot \mathbf{x} \\ \text{subject to} \quad & A\mathbf{x} \leq \mathbf{1}, \\ & \mathbf{x} \geq \mathbf{0} \end{aligned}$$

Max-min linear program

Distributed setting:

- one node $v \in V$ for each variable x_v,
  one node $i \in I$ for each constraint $\mathbf{a}_i \cdot \mathbf{x} \leq 1$,
  one node $k \in K$ for each objective $\mathbf{c}_k \cdot \mathbf{x}$
- $v \in V$ and $i \in I$ adjacent if $a_{iv} > 0$,
  $v \in V$ and $k \in K$ adjacent if $c_{kv} > 0$

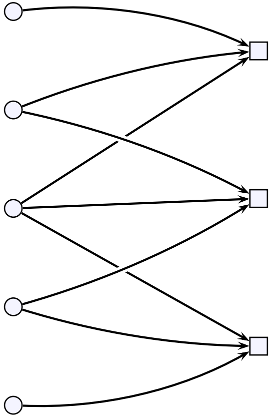
Key parameters:

- Δ_I = max. degree of $i \in I$
- Δ_K = max. degree of $k \in K$

Example

Task: Data gathering in a sensor network

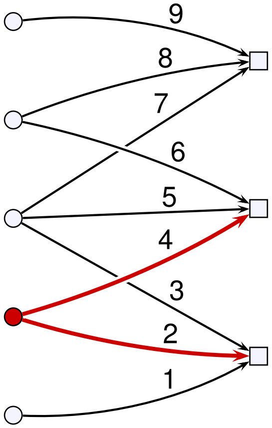
- circle = sensor
- square = relay
- edge = network connection



Example

Task: Maximise the minimum amount of data gathered from each sensor

$$\text{maximise} \quad \min \{ x_1,\ x_2 + x_4,\ x_3 + x_5 + x_7,\ x_6 + x_8,\ x_9 \}$$

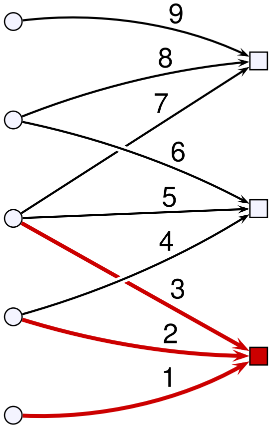


Example

Task: Maximise the minimum amount of data gathered from each sensor; each relay has a limited battery capacity

$$\text{maximise} \quad \min\{ x_1,\ x_2 + x_4,\ x_3 + x_5 + x_7,\ x_6 + x_8,\ x_9 \}$$

$$\text{subject to} \quad x_1 + x_2 + x_3 \leq 1,$$
$$x_4 + x_5 + x_6 \leq 1,$$
$$x_7 + x_8 + x_9 \leq 1,$$
$$x_1, x_2, \ldots, x_9 \geq 0$$

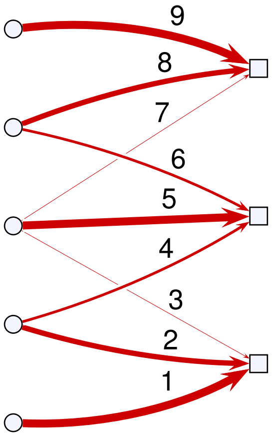


Example

Task: Maximise the minimum amount of data gathered from each sensor; each relay has a limited battery capacity

An optimal solution:

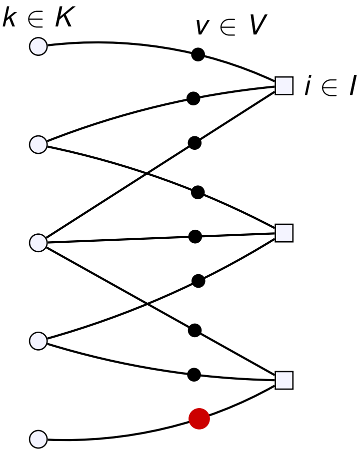
$$
\begin{aligned}
x_1 = x_5 = x_9 &= 3/5, \\
x_2 = x_8 &= 2/5, \\
x_4 = x_6 &= 1/5, \\
x_3 = x_7 &= 0
\end{aligned}
$$



Example

Communication graph: $\mathcal{G} = (V \cup I \cup K, E)$

$$
\begin{aligned}
\text{maximise} \quad & \min \{ \\
& \qquad x_1,\ x_2 + x_4, \\
& \qquad x_3 + x_5 + x_7, \\
& \qquad x_6 + x_8,\ x_9 \\
& \} \\
\text{subject to} \quad & x_1 + x_2 + x_3 \leq 1, \\
& x_4 + x_5 + x_6 \leq 1, \\
& x_7 + x_8 + x_9 \leq 1, \\
& \underline{x_1}, x_2, \ldots, x_9 \geq 0
\end{aligned}
$$

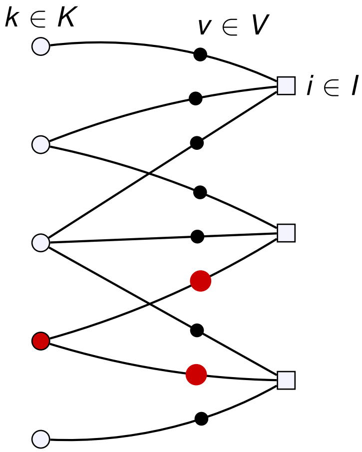


Example

Communication graph: $\mathcal{G} = (V \cup I \cup K, E)$

$$\text{maximise} \quad \min \{ x_1,\ \underline{x_2 + x_4},\ x_3 + x_5 + x_7,\ x_6 + x_8,\ x_9 \}$$

$$\text{subject to} \quad x_1 + x_2 + x_3 \leq 1,\quad x_4 + x_5 + x_6 \leq 1,\quad x_7 + x_8 + x_9 \leq 1,\quad x_1, x_2, \ldots, x_9 \geq 0$$



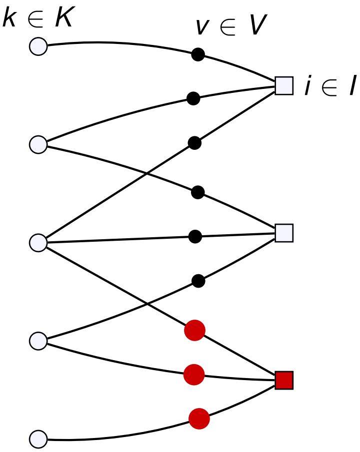
Example

Communication graph: $\mathcal{G} = (V \cup I \cup K, E)$

maximise $\min\{ x_1,\ x_2 + x_4,\ x_3 + x_5 + x_7,\ x_6 + x_8,\ x_9 \}$

subject to

$$\underline{x_1 + x_2 + x_3 \leq 1},$$
$$x_4 + x_5 + x_6 \leq 1,$$
$$x_7 + x_8 + x_9 \leq 1,$$
$$x_1, x_2, \ldots, x_9 \geq 0$$



Old results

"Safe algorithm":

Node v chooses

$$x_v = \min_{i\,:\,a_{iv}>0} \frac{1}{a_{iv}\,\left|\{u : a_{iu} > 0\}\right|}$$

(Papadimitriou and Yannakakis 1993)

Factor Δ_I approximation

Uses information only in radius 1 neighbourhood of v

A better approximation ratio with a larger radius?

New results, general case

The safe algorithm is factor Δ_I approximation

Theorem

For any $\epsilon > 0$, there is a local algorithm for max-min LPs with approximation ratio $\Delta_I(1 - 1/\Delta_K) + \epsilon$

Theorem

There is no local algorithm for max-min LPs with approximation ratio $\Delta_I(1 - 1/\Delta_K)$

Degree of a constraint $i \in I$ is at most Δ_I

Degree of an objective $k \in K$ is at most Δ_K

New results, bounded growth

Assume *bounded relative growth* beyond radius R:

$$\frac{|B(v, r+2)|}{|B(v, r)|} \leq 1 + \delta \qquad \text{for all } v \in V,\ r \geq R$$

where $B(v, r) =$ agents in radius r neighbourhood of v

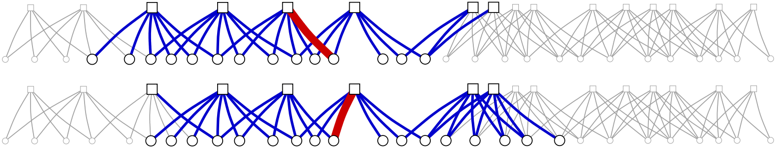
Theorem

There is a local algorithm for max-min LPs with approximation ratio $1 + 2\delta + o(\delta)$

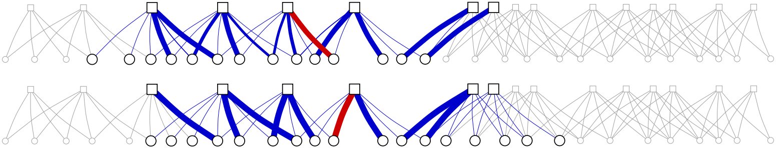
There is no local algorithm for max-min LPs with approximation ratio $1 + \delta/2$
(assuming $\Delta_I \geq 3$, $\Delta_K \geq 3$, $0.0 < \delta < 0.1$)

Approximability, bounded growth

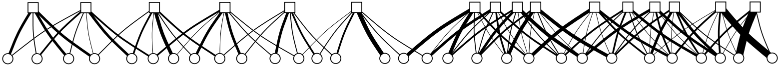
Step 1: Choose local constant-size subproblems



Step 3: Solve them optimally



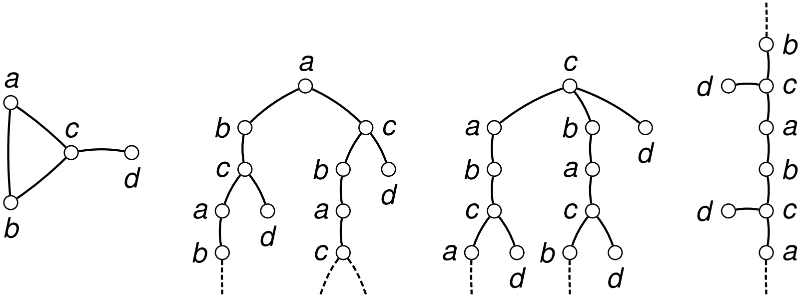
Step 3: Take averages of local solutions, add some slack



Approximability, general case

Preliminary step 1:

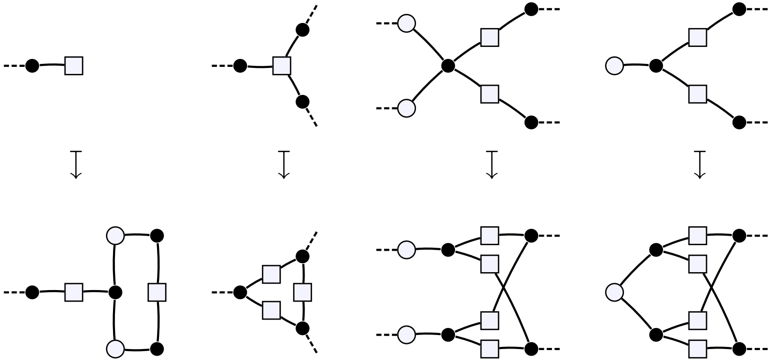
Unfold the graph into an infinite tree



Approximability, general case

Preliminary step 2:

Apply a sequence of local transformations (and unfold again)

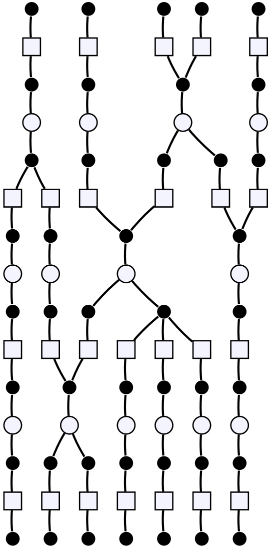


Approximability, general case

It is enough to design a local approximation algorithm for the following special case:

- Communication graph $\mathcal{G}$ is an (infinite) tree
- Degree of each constraint $i \in I$ is exactly 2
- Degree of each objective $k \in K$ is at least 2
- Each agent $v \in V$ adjacent to at least one constraint
- Each agent $v \in V$ adjacent to exactly one objective
- $c_{kv} \in \{0, 1\}$

Approximability, general case

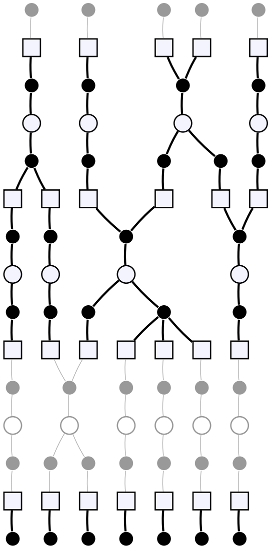


After the local transformations,
we have an infinite tree
with a fairly regular structure

In a centralised setting,
we could organise
the nodes into *layers*

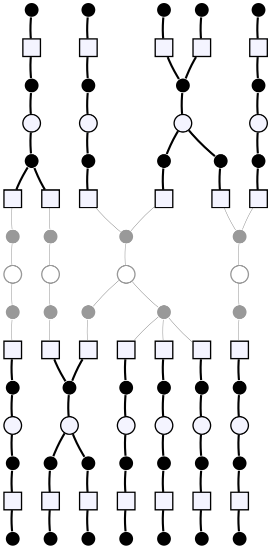
Then we could design
an approximation algorithm. . .

Approximability, general case



“Switch off” every
*R*th layer of objectives

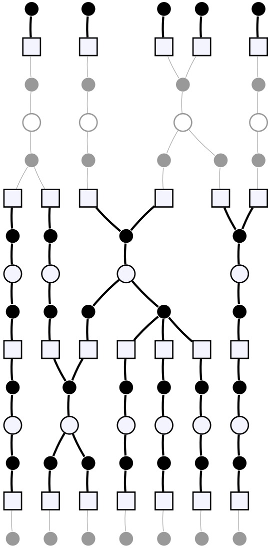
Approximability, general case



“Switch off” every
Rth layer of objectives

Consider all possible locations
(shifting strategy)

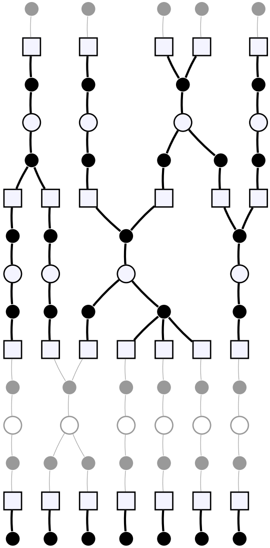
Approximability, general case



“Switch off” every
Rth layer of objectives

Consider all possible locations
(shifting strategy)

Approximability, general case



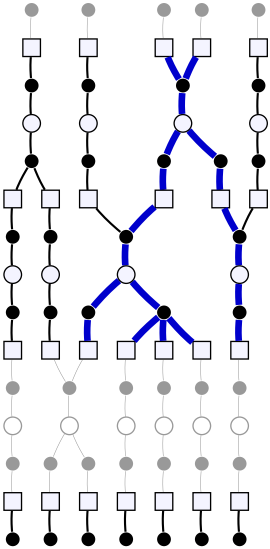
"Switch off" every
Rth layer of objectives

Consider all possible locations
(shifting strategy)

Solve the LP for the "active" layers,
take averages

Factor $R/(R-1)$ approximation

Approximability, general case



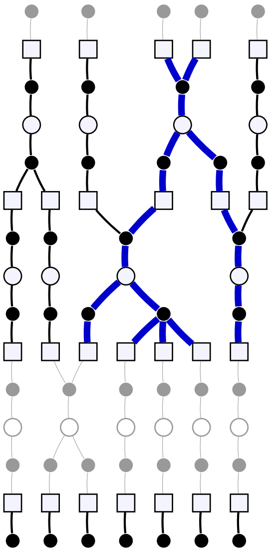
We could solve the LP simply by propagating information upwards between a pair of “passive” layers

But we cannot choose the layers by any local algorithm!

Two fundamentally different roles for agents: “up” and “down”

How to choose the roles?
How to break the symmetry?

Approximability, general case



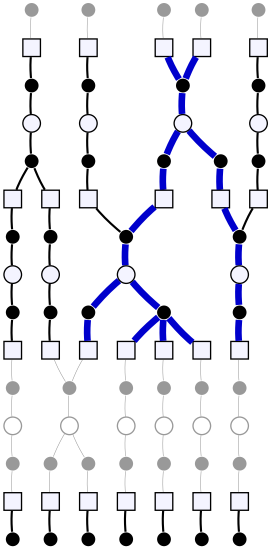
Trick: consider both possible roles for each agent, "up" an "down"

Compute locally two candidate solutions, one for each role

Take averages

Surprise: factor $\Delta_I(1 - 1/\Delta_K) + \epsilon$ approximation, best possible!

Approximability, general case



Some complications:

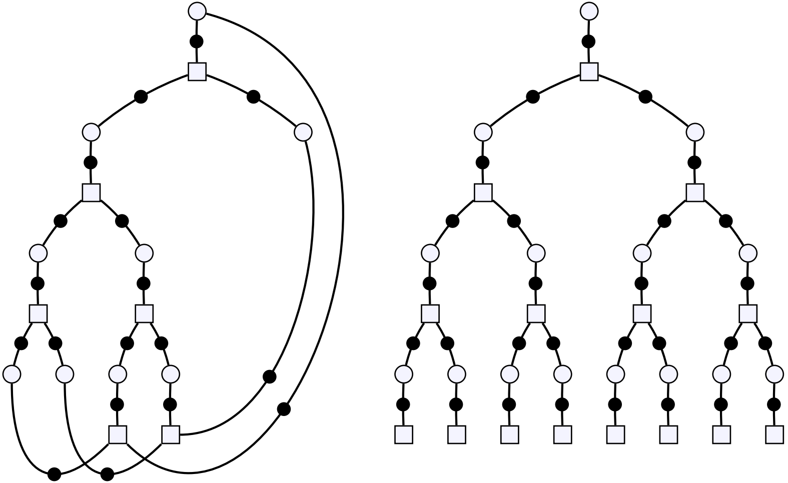
- several optimal solutions
- how to make sure that the local choices are "compatible" with each other?

Key idea:

- "down" nodes choose as large values as possible
- "up" nodes choose as small values as possible

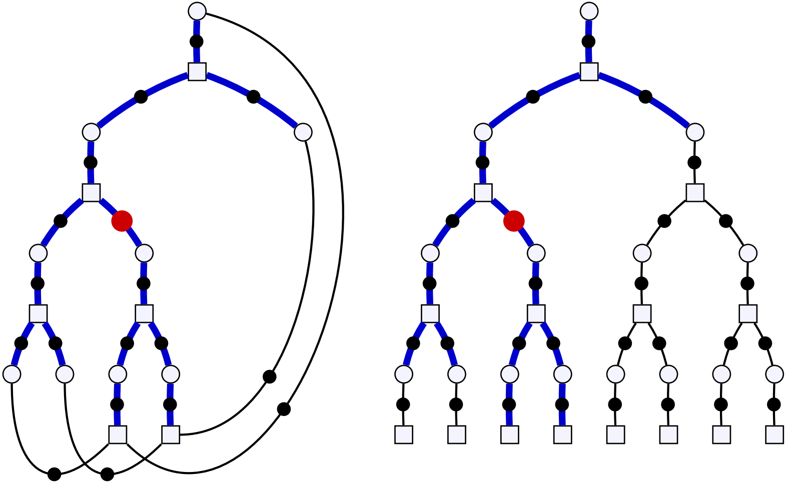
Inapproximability

Regular high-girth graph or regular tree?



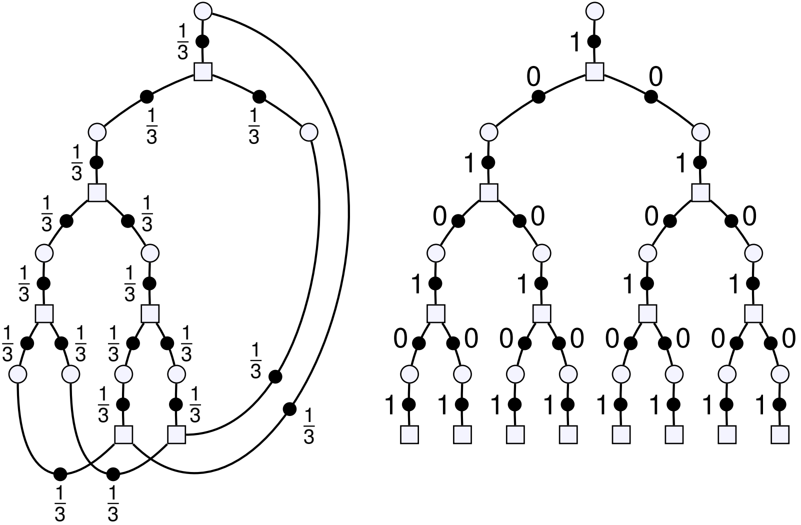
Inapproximability

Locally indistinguishable



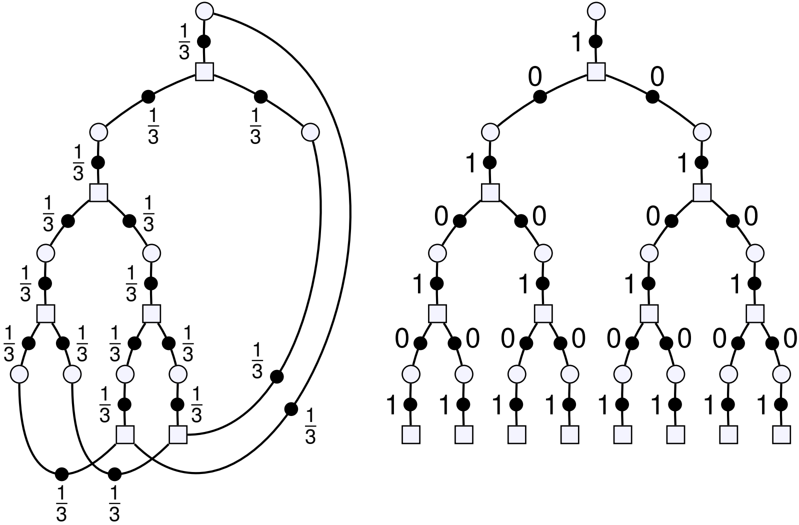
Inapproximability

Optimum $\leq 2/3$ vs. optimum ≥ 1



Inapproximability

Approx. ratio $\geq 1/(2/3) = 3\,(1 - 1/2) = \Delta_I\,(1 - 1/\Delta_K)$



Summary

Max-min linear programs: given $A, \mathbf{c}_k \geq 0$,

$$\begin{aligned} \text{maximise} \quad & \min_{k \in K} \mathbf{c}_k \cdot \mathbf{x} \\ \text{subject to} \quad & A\mathbf{x} \leq \mathbf{1}, \\ & \mathbf{x} \geq \mathbf{0} \end{aligned}$$

Local algorithms: output of a node is a function of input within its constant-radius neighbourhood

Main result: tight characterisation of local approximability

http://www.hiit.fi/ada/geru — jukka.suomela@cs.helsinki.fi

References (1)

B. Awerbuch and M. Sipser (1998). Dynamic networks are as fast as static networks. *FOCS 1988*. [DOI]

B. Awerbuch and G. Varghese (1991). Distributed program checking: a paradigm for building self-stabilizing distributed protocols. *FOCS 1991*. [DOI]

J. Czyzowicz, S. Dobrev, T. Fevens, H. González-Aguilar, E. Kranakis, J. Opatrny, and J. Urrutia (2008). Local algorithms for dominating and connected dominating sets of unit disk graphs with location aware nodes. *LATIN 2008*. [DOI]

P. Floréen, P. Kaski, T. Musto, and J. Suomela (2007). Local approximation algorithms for scheduling problems in sensor networks. *ALGOSENSORS 2007*. [DOI]

P. Floréen, P. Kaski, T. Musto, and J. Suomela (2008a). Approximating max-min linear programs with local algorithms. *IPDPS 2008*. [DOI]

References (2)

P. Floréen, M. Hassinen, P. Kaski, and J. Suomela (2008b). Local approximation algorithms for a class of 0/1 max-min linear programs. Manuscript, arXiv:0806.0282 [cs.DC].

P. Floréen, M. Hassinen, P. Kaski, and J. Suomela (2008c). Tight local approximation results for max-min linear programs. *ALGOSENSORS 2008*.

P. Floréen, J. Kaasinen, P. Kaski, and J. Suomela (2008d). An optimal local approximation algorithm for max-min linear programs. Manuscript, arXiv:0809.1489 [cs.DC].

M. Hassinen, V. Polishchuk, and J. Suomela (2008). Local 3-approximation algorithms for weighted dominating set and vertex cover in quasi unit-disk graphs. *LOCALGOS 2008*.

F. Kuhn (2005). *The Price of Locality: Exploring the Complexity of Distributed Coordination Primitives*. PhD thesis.

References (3)

F. Kuhn and R. Wattenhofer (2005). Constant-time distributed dominating set approximation. *Distributed Computing*, 17(4):303–310. [DOI]

F. Kuhn, T. Moscibroda, and R. Wattenhofer (2004). What cannot be computed locally! *PODC 2004*. [DOI]

F. Kuhn, T. Moscibroda, and R. Wattenhofer (2006). The price of being near-sighted. *SODA 2006*. [DOI]

C. Lenzen, Y. A. Oswald, and R. Wattenhofer (2008). What can be approximated locally? *SPAA 2008*.

N. Linial (1992). Locality in distributed graph algorithms. *SIAM Journal on Computing*, 21(1):193–201. [DOI]

M. Naor and L. Stockmeyer (1995). What can be computed locally? *SIAM Journal on Computing*, 24(6):1259–1277. [DOI]

References (4)

C. H. Papadimitriou and M. Yannakakis (1991). On the value of information in distributed decision-making. *PODC 1991*. [DOI]

C. H. Papadimitriou and M. Yannakakis (1993). Linear programming without the matrix. *STOC 1993*. [DOI]

M. Parnas and D. Ron (2007). Approximating the minimum vertex cover in sublinear time and a connection to distributed algorithms. *Theoretical Computer Science*, 381(1–3):183–196. [DOI]

J. Urrutia (2007). Local solutions for global problems in wireless networks. *Journal of Discrete Algorithms*, 5(3):395–407. [DOI]

Y. Wang and X.-Y. Li (2006). Localized construction of bounded degree and planar spanner for wireless ad hoc networks. *Mobile Networks and Applications*, 11(2):161–175. [DOI]

A. Wiese and E. Kranakis (2008). Local PTAS for independent set and vertex cover in location aware unit disk graphs. *DCOSS 2008*. [DOI]