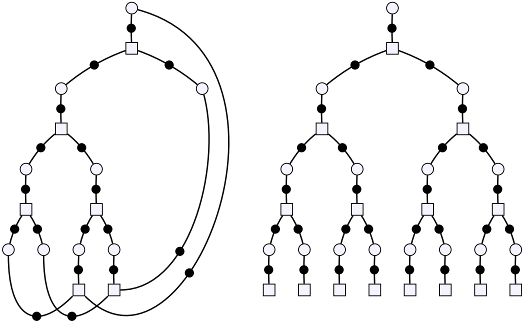
Local algorithms and max-min linear programs

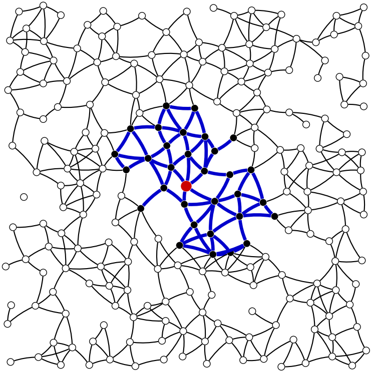
Patrik Floréen
Marja Hassinen
Joel Kaasinen
Petteri Kaski
Topi Musto
Jukka Suomela



Hecse Autumn School · Porvoo · 20 October 2008

Local algorithms

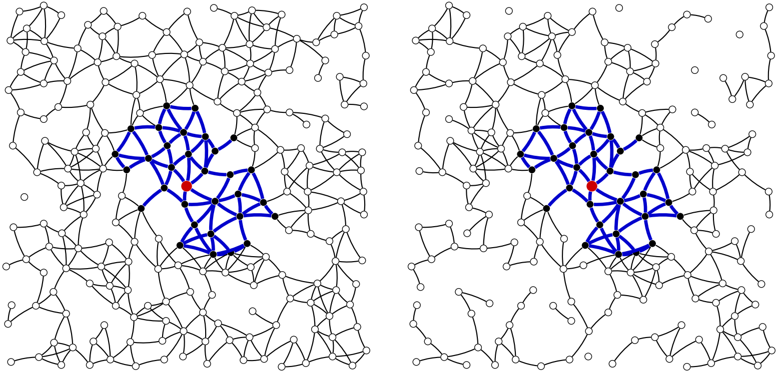
Local algorithm: output of a node is a function of input within its *constant-radius neighbourhood*



(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 for each node
- Distributed constant time – even in an infinite network

. . . and fault-tolerant:

- Recovers in constant time
- Topology change only affects
  a constant-size part of the network

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

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,
  dominating set, etc. (Kuhn 2005; Kuhn et al. 2004, 2006)

Local algorithms

Some previous positive results:

- Weak colouring (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)

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

Objective: maximise $\min_k \mathbf{c}_k \cdot \mathbf{x}$ subject to $A\mathbf{x} \leq \mathbf{1}$, $\mathbf{x} \geq \mathbf{0}$

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 $c_{kv} > 0$

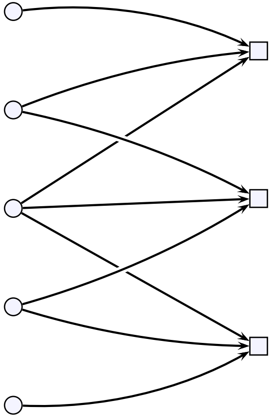
Key parameters:

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

Example

Task: Fair bandwidth allocation in a communication network

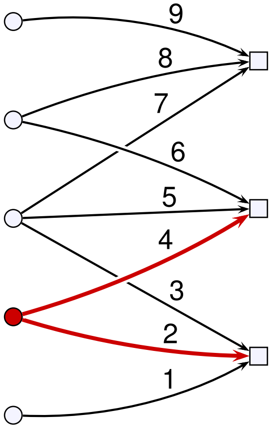
- circle = customer
- square = access point
- edge = network connection



Example

Task: Allocate a fair share of bandwidth for each customer

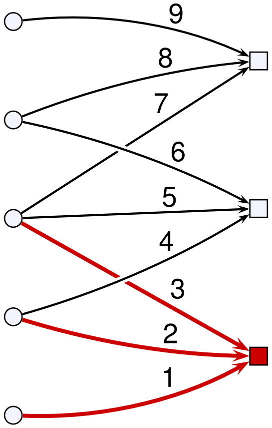
$$\text{maximise} \quad \min \Big\{ x_1,\ \underline{x_2 + x_4},\ x_3 + x_5 + x_7,\ x_6 + x_8,\ x_9 \Big\}$$



Example

Task: Allocate a fair share of bandwidth for each customer; each access point has a limited capacity

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

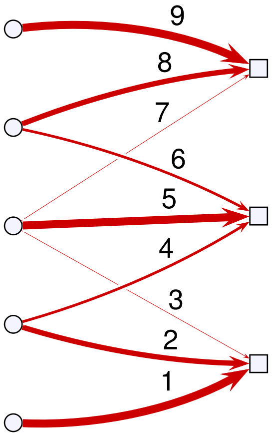


Example

Task: Allocate a fair share of bandwidth for each customer; each access point has a limited capacity

An optimal solution:

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



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

The safe algorithm is factor Δ_I approximation

Theorem

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

Theorem

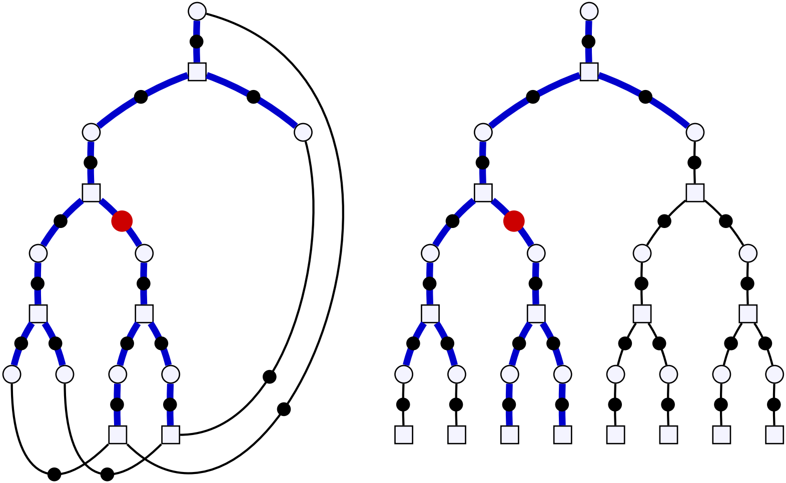
For any $\epsilon > 0$*, there is a local algorithm for max-min LPs with approximation ratio* $\Delta_I(1 - 1/\Delta_K) + \epsilon$

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

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

Inapproximability

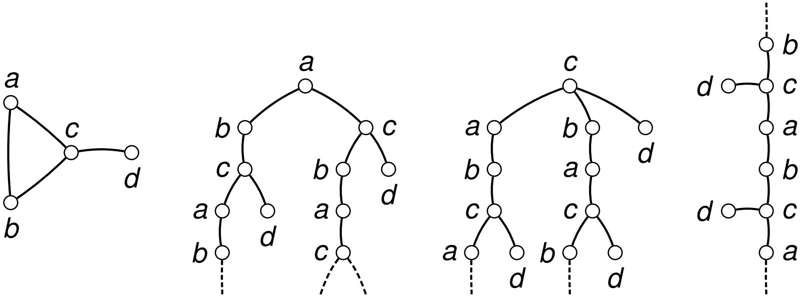
Regular high-girth graph or regular tree?



Approximability

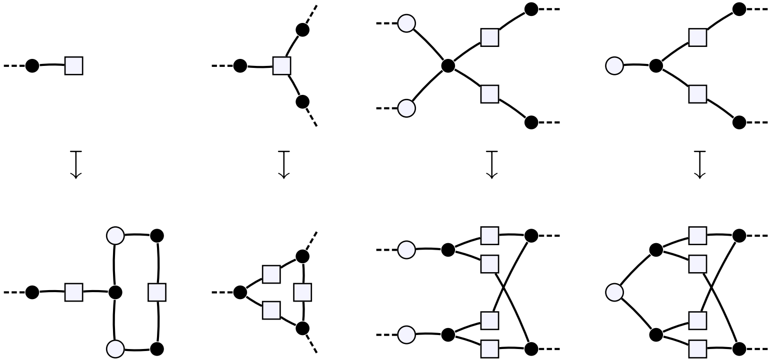
Preliminary step 1:

Unfold the graph into an infinite tree

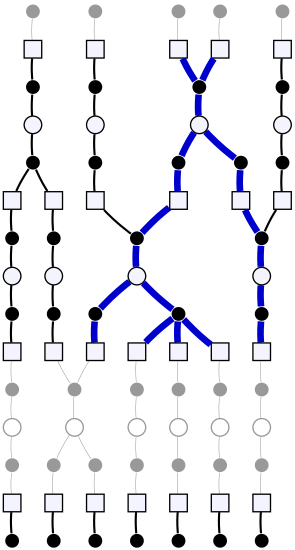


Approximability

Preliminary step 2:
Apply a sequence of local transformations (and unfold again)



Approximability



Alternating layers of “up” agents and “down” agents

- “up” nodes choose as small values as possible
- “down” nodes choose as large values as possible

But there is no local algorithm that chooses the roles in a globally consistent manner

Key idea: consider both roles, take averages

Summary

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

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

Local algorithm: constant-time distributed algorithm

Main result: tight characterisation of local approximability

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

References (1)

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

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

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

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

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

References (2)

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]

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