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

(Linial 1992; Naor and Stockmeyer 1995)
Local algorithm: changes outside the local horizon of a node do not affect its output

(Linial 1992; Naor and Stockmeyer 1995)
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 $A \geq 0$, $c_k \geq 0$

**Objective:**

$$\text{maximise } \min_{k \in K} c_k \cdot x$$

subject to $A x \leq 1$, $x \geq 0$

**Generalisation of packing LP:**

$$\text{maximise } c \cdot x$$

subject to $A x \leq 1$, $x \geq 0$
Max-min linear program

Objective: maximise \( \min_k c_k \cdot x \) subject to \( A x \leq 1, \ x \geq 0 \)

Distributed setting:

- one node \( v \in V \) for each variable \( x_v \),
- one node \( i \in I \) for each constraint \( a_i \cdot x \leq 1 \),
- one node \( k \in K \) for each objective \( c_k \cdot x \)

- \( v \in V \) and \( i \in I \) adjacent if \( a_{iv} > 0 \),
- \( v \in V \) and \( k \in K \) adjacent if \( c_{kv} > 0 \)

Key parameters:

- \( \Delta_I = \text{max. degree of } i \in I \)
- \( \Delta_K = \text{max. degree of } k \in K \)
**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*

maximise \( \min \{ \)
\[
x_1, \ x_2 + x_4, \\
x_3 + x_5 + x_7, \\
x_6 + x_8, \ x_9
\]

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

maximise \( \min \{ \)
\[
\begin{align*}
x_1, & \quad x_2 + x_4, \\
x_3, & \quad x_5 + x_7, \\
x_6, & \quad x_8, x_9
\end{align*}
\]

\} 

subject to \[
\begin{align*}
x_1 + x_2 + x_3 & \leq 1, \\
x_4 + x_5 + x_6 & \leq 1, \\
x_7 + x_8 + x_9 & \leq 1, \\
x_1, x_2, \ldots, x_9 & \geq 0
\end{align*}
\]
Example

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

An optimal solution:

\[ x_1 = x_5 = x_9 = \frac{3}{5}, \]
\[ x_2 = x_8 = \frac{2}{5}, \]
\[ x_4 = x_6 = \frac{1}{5}, \]
\[ x_3 = x_7 = 0 \]
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 $\Delta_i$ 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 $\Delta_I$ approximation

**Theorem**

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

**Theorem**

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

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

Degree of an objective $k \in K$ is at most $\Delta_K$
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)
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
Max-min linear program: given $A, c_k \geq 0$,

\[
\text{maximise } \min_{k \in K} c_k \cdot x \\
\text{subject to } A x \leq 1, \quad x \geq 0
\]

Local algorithm: constant-time distributed algorithm

Main result: tight characterisation of local approximability

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


