Approximating max-min linear programs with local algorithms

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

HIIT seminar 29 February 2008



Example: Fair bandwidth allocation in a communication network

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



Example: Allocate a fair share of bandwidth for each customer

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



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

maximise min {

$$x_1, x_2 + x_4,$$

 $x_3 + x_5 + x_7,$
 $x_6 + x_8, x_9$
}
subject to $x_1 + x_2 + x_3 \le 1,$
 $x_4 + x_5 + x_6 \le 1,$
 $x_7 + x_8 + x_9 \le 1,$
 $x_1, x_2, \dots, x_9 \ge 0$

0

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

An optimal solution:

$$x_1 = x_5 = x_9 = 3/5,$$

$$x_2 = x_8 = 2/5,$$

$$x_4 = x_6 = 1/5,$$

$$x_3 = x_7 = 0$$



Max-min linear programs: Definition

Objective:

$$\begin{array}{lll} \text{maximise} & \min_{k \in \mathcal{K}} \sum_{v \in V} c_{kv} x_v \\ \text{subject to} & \sum_{v \in V} a_{iv} x_v \leq 1 & \forall i \in I, \\ & x_v \geq 0 & \forall v \in V \end{array}$$

Idea:

- One unit of activity by agent v ∈ V benefits party k ∈ K by c_{kv} ≥ 0 units and consumes a_{iv} ≥ 0 units of resource i ∈ I
- Objective: set the activities to provide a fair share of benefit for each party

Max-min linear programs: Definition

Let $A, c, c_k \geq 0$

In matrix notation:

 $\begin{array}{ll} \mbox{maximise} & \min_{k \in \mathcal{K}} c_k x \\ \mbox{subject to} & Ax \ \leq \ \mathbf{1}, \\ & x \ \geq \ \mathbf{0} \end{array}$

Generalisation of packing LP:

 $\begin{array}{ll} \text{maximise} & cx\\ \text{subject to} & Ax \leq \mathbf{1},\\ & x > \mathbf{0} \end{array}$

7/24

Max-min linear programs: Challenges

What about large networks? What if there are frequent changes in network topology?



Max-min linear programs: Challenges

Could we perhaps use solely *local* information to find a *provably near-optimal* solution to the global problem?



Local algorithms

Definition:

(e.g., Naor and Stockmeyer 1995)

- Distributed algorithm
- Output of a node is a function of input within its constant-radius neighbourhood

Our focus:

Problems where the size of input and output per node is bounded by a constant

Here *constant* = does not depend on input, in particular, does not depend on the number of nodes (but may depend on desired approximation ratio, etc.)

Local algorithms

Advantages of local algorithms:

- Space and time complexity is constant per node
- Distributed constant time (even in an infinite network)
- Topology change affects a constant-size part only
- Bounded-fan-in, constant-depth Boolean circuits: in NC⁰
- Simple linear-time centralised algorithm; in some cases randomised, approximate sublinear-time algorithms (Parnas and Ron 2007)
- Insight into algorithmic value of information (cf. Papadimitriou and Yannakakis 1991)

Local algorithms: Prior work

Some previous negative results:

- ► 3-colouring of *n*-cycle not possible (Linial 1992)
- ► No constant-factor approximation of vertex cover, etc.

(Kuhn et al. 2004)

Some previous positive results:

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

Max-min linear programs: given $A, c_k \ge 0$,

maximise $\min_{k \in K} c_k x$ subject to $Ax \leq 1, x \geq 0$

Local algorithms: output is a function of input in a constant-radius neighbourhood

Missing link: exactly what does a constant-radius neighbourhood mean in a max-min LP?

Max-min linear programs: Local setting

Communication hypergraph \mathcal{H} :

- agents are vertices
- ► {v ∈ V : a_{iv} > 0} and {v ∈ V : c_{kv} > 0} are edges for all i, k



Max-min linear programs: Local setting

Each agent knows:

- with whom it is competing for resources
- with whom it is working together



Max-min linear programs: Local setting

Each agent knows:

- with whom it is competing for resources
- with whom it is working together

For example, in this bandwidth allocation problem:



radius 3 local neighbourhood in hypergraph $\ensuremath{\mathcal{H}}$ is:



Challenges of locality

Two instances of the bandwidth allocation problem:

Different optimal solutions:

... but identical local neighbourhoods:

 Two instances of the bandwidth allocation problem:

Near-optimal solutions:

- Here we can make the same decisions in parts where local neighbourhoods are identical
- Can we generalise this idea to arbitrary instances?

Old results: approximability

Yes, there are local approximation algorithms for max-min linear programs

"Safe algorithm": node v chooses

$$\mathbf{x}_{v} = \min_{i:a_{iv}>0} \frac{1}{a_{iv} |\{u:a_{iu}>0\}|}$$

(Papadimitriou and Yannakakis 1993)

This is a factor Δ_l^V approximation where $\Delta_l^V =$ maximum number of variables in a constraint

Uses information only in radius 1 neighbourhood of v — a better approximation ratio with a larger radius?

New results: inapproximability

The safe algorithm is factor Δ_I^V approximation

In general, we cannot have a much better approximation ratio:

Theorem

There is no local algorithm for max-min LP with approximation ratio less than

$$\frac{\Delta_I^V+1}{2}-\frac{1}{2\Delta_K^V-2}$$

• Δ_I^V = maximum number of variables in a constraint

• Δ_{K}^{V} = maximum number of variables that benefit a party

Proof idea: inapproximability

- Construct instance S with no short cycles
- \blacktriangleright Apply the supposed approximation algorithm ${\cal A}$ to S
- ▶ Study the solution; choose a "bad" tree-like area $S' \subset S$
- \mathcal{A} has to make the same local decisions in S', suboptimal



New results: approximability

Define relative growth

$$\gamma(r) = \max_{v \in V} \frac{|B_{\mathcal{H}}(v, r+1)|}{|B_{\mathcal{H}}(v, r)|}$$

where $B_{\mathcal{H}}(v, r)$ = radius *r* neighbourhood of *v* in \mathcal{H}

If \mathcal{H} has bounded relative growth, then better approximation ratios can be achieved:

Theorem

For any R, there is a local algorithm for max-min LP with approximation ratio $\gamma(R-1)\gamma(R)$ and local horizon $\Theta(R)$

Algorithm idea: approximability

Choose local constant-size subproblems:



Solve them optimally:



Take averages of local solutions, add some slack:

Max-min linear programs: given $A, c_k \ge 0$,

maximise $\min_{k \in K} c_k x$ subject to $Ax \leq 1, x \geq 0$

Local algorithms: output is a function of input in a constant-radius neighbourhood

Results:

- Inapproximability results for general graphs
- Approximation algorithm for bounded-growth graphs

To appear in *IPDPS 2008*

References (1)

- P. Floréen, P. Kaski, T. Musto, and J. Suomela. Approximating max-min linear programs with local algorithms. In *Proc. 22nd IEEE International Parallel and Distributed Processing Symposium (IPDPS, Miami, FL, USA, April 2008)*, 2008. To appear.
- F. Kuhn and R. Wattenhofer. Constant-time distributed dominating set approximation. *Distributed Computing*, 17(4):303–310, 2005. [DOI]
- F. Kuhn, T. Moscibroda, and R. Wattenhofer. What cannot be computed locally! In Proc. 23rd Annual ACM Symposium on Principles of Distributed Computing (PODC, St. John's, Newfoundland, Canada, July 2004), pages 300–309, New York, NY, USA, 2004. ACM Press. [DOI]

References (2)

- F. Kuhn, T. Moscibroda, and R. Wattenhofer. The price of being near-sighted. In *Proc. 17th Annual ACM-SIAM Symposium on Discrete Algorithms (SODA, Miami, FL, USA, January 2006)*, pages 980–989, New York, NY, USA, 2006. ACM Press. [DOI]
- N. Linial. Locality in distributed graph algorithms. *SIAM Journal on Computing*, 21(1):193–201, 1992. [DOI]
- M. Naor and L. Stockmeyer. What can be computed locally? *SIAM Journal on Computing*, 24(6):1259–1277, 1995. [DOI]
- C. H. Papadimitriou and M. Yannakakis. On the value of information in distributed decision-making. In *Proc. 10th Annual ACM Symposium on Principles of Distributed Computing (PODC, Montreal, Quebec, Canada, August 1991)*, pages 61–64, New York, NY, USA, 1991. ACM Press. [DOI]

References (3)

- C. H. Papadimitriou and M. Yannakakis. Linear programming without the matrix. In *Proc. 25th Annual ACM Symposium on Theory of Computing (STOC, San Diego, CA, USA, May 1993)*, pages 121–129, New York, NY, USA, 1993. ACM Press. [DOI]
- M. Parnas and D. Ron. Approximating the minimum vertex cover in sublinear time and a connection to distributed algorithms. *Theoretical Computer Science*, 381(1–3):183–196, 2007. [DOI]