A Simple Local-Control Approximation Algorithm for Multicommodity Flow

BARUCH AWERBUCH & TOM LEIGHTON

1993

PRESENTER: ERIC KANGAS

About the Authors

Some factoids about Baruch Awerbuch

- Professor of Computer Science at Johns Hopkins
- Has written or contributed to 180+ papers.
- Associate Editor of the Journal of Algorithms
- Winner of Edsger W. Dijkstra Prize in Distributed Computing 2008 along with David Peleg for a paper on sparse partitions.
- Has an Erdös number of 2.

About the Authors

Some factoids about Tom Leighton

- Real name is Frank Thomson Leighton.
- Professor of Applied Mathematics.
- Co-founded Akamai Technologies and currently is Chief Scientist.
- Head of Algorithms group at MIT's Computer Science and Artificial Intelligence Lab since 1996.
- Served on the President's Information Technology Advisory Committee (PITAC) around 2005.
- Also has Erdös number of 2.

About the Publication

- Appeared in the 34th IEEE Conference on Foundations of Computer Science in 1993.
- Supported by Air Force, DARPA, and IBM funding.
- Has been cited 103 times according to Google Scholar.

Multicommodity Flow

$G = (N, M)$, *where edge* $(u, v) \in E$ *has capacity* $c(e)$.

N *nodes*, M *edges*, *and* K *commodities*.

Commodity i *has a demand* d_i.

Each commodity has a single source and a single sink.

Facts about Multicommodity Flow

- Has been well studied for several decades.
- Many useful applications, such as:
  - product distribution
  - traffic planning
  - scheduling problems
  - fault prone distributed networks.

- In addition, a lot of NP hard problems can be approximately solved through multicommodity flow problems, such as:
  - Graph partitioning
  - Minimum feedback arc set
  - Minimum cut linear arrangement
  - Minimum 2D area layout
  - Via minimization
  - Optimal matrix arrangement for nested dissection

Motivations for discovery of a local-control online algorithm

- Solutions to other problems can be effectively approximated using local-control online multicommodity flow such as:
  - ○ Packet routing
  - ○ Communication problems.
- Previously, flow techniques were not commonly used for these problems.

Additional motivations for the research

- Previously, most of the research has focused on 1-commodity flow (Max flow).
- Most rely on finding augmented paths from source to sink.
- Best run in approximately $O(NM)$ steps.
- "All exact algorithms for multicommodity flow are based on linear programming, all have horrendous running times (even though polynomial), and none are used for large networks in practice."

Situation for Multicommodity flow

- Previous to this publication, P.M. Vaidya, in his 1989 paper titled *Speeding up linear programming using fast matrix multiplication*, developed a Polynomial Time Approximation Scheme (PTAS) for min-cost multicommodity flow.
- $O(K^{\frac{7}{2}}NM^{\frac{5}{2}}\log(DU\varepsilon^{-1}))$ steps approximately, where

$$D = largest\ demand$$
$$U = largest\ capacity$$

Situation for Multicommodity flow Part 2

- In 1991 Tom Leighton discovered a combinatorial PTAS algorithm for the 1 commodity min-cost flow problem.
- Runs in $O(K^2 NM\varepsilon^{-2}\log(\mathrm{K})\log^3(N))$ steps.
- This can be improved by a factor of K by randomization.

An abstract view of the solution

- **Model the problem as a continuous flow problem.** Each turn we add new flow to each source.
- **Utilize an edge balancing technique**, similar to natural fluid flow, which (simplifying the physics) distributes by the fluids eventually moving to areas of less pressure.
- **Calculate the disparity of the queue size.** Here, we use *queues* for both sides of each edge for each commodity,,
- **Push flow across each edge to balance the queues.**

$$\Delta_i(e) = q_i\big(tail(e)\big) - q_i(head(e))$$

- **Remove commodities when they reach their correct sinks.**

Runtime

- The algorithm takes at most

  $$O(M^3K^{5/2}L\varepsilon^{-3}logK)$$

  steps to find feasible flow, where

  $$L = length\ of\ longest\ flow\ path$$

Justifications

1. They suspect that the run time is much better, but have not proved it.
2. The algorithm is simpler than previous Max flow algorithms.
3. It works well, and there are several variations which are comparable or superior to the best known algorithms in run time.
4. Can be used where not all information is known and global control is not possible.

Algorithm in more detail

- Reduce a static problem to a continuous problem:
  - $(1+\varepsilon)d_i$ units of commodity i are pumped into each source at the start of each round.
  - We may move commodities through one edge only per round.
  - We may not exceed edge capacity constraints per round.
  - Flow that reaches the commodity's sink is removed at the end of the round.

Finding a feasible solution to the continuous problem

- Run the continuous algorithm until the amount of each commodity in the queues is at most

  $$\frac{\varepsilon}{(1+\varepsilon)}$$

  of the total amount that has been pumped into the network.
- After *R* rounds, an average of $d_i \ units \ of \ commodity \ i$ have been shipped through the network, and
- At most $(1+\varepsilon)d_iR$ units of i will remain.
- Solution = average of the solution for each round.

Phase 1

- For each commodity i , evenly distribute $(1 + \varepsilon)d_i$ units of i to each queue corresponding to the edge incident to $source_i$.

Phase 2

- $\Delta_i(e) = q_i(tail(e)) - q_i(head(e))$ is the imbalance for commodity i along edge e.
- $q_i(tail(e))$ is the height of q_i at the tail end of e, where $q_i(head(e))$ is the height for the head.
- Find the commodity where $\frac{\Delta_i(e)}{d_i^2}$ is maximized.

Phase 2 continued:

- Push f_i units of i along e for each commodity, where $f_1, f_2, \ldots, f_K$ are chosen to maximize

$$\sum_{1 \le i \le K} f_i(\Delta_i(e) - f_i) d_i^{-2}$$

s.t.

1) $f_i \ge 0 \; for \; all \; 1 \le i \le K,$

2) $\displaystyle\sum_{1 \le i \le K} f_i \le c(e),$

3) $f_i = 0 \; if \; c(e) \le \dfrac{\varepsilon d_i}{M}$

Phase 3

- Remove all flow for commodity i which has reached $sink_i$.

Phase 4

- “Rebalance the nodes:
  Reallocate each commodity within each node so that the queues for commodity i are all equal within each node $(1 \leq i \leq K)$.”

Cost

- Each round is implemented in $O(MKlogK)$ steps.
- Phases 1 & 3 take $K\delta$ steps,

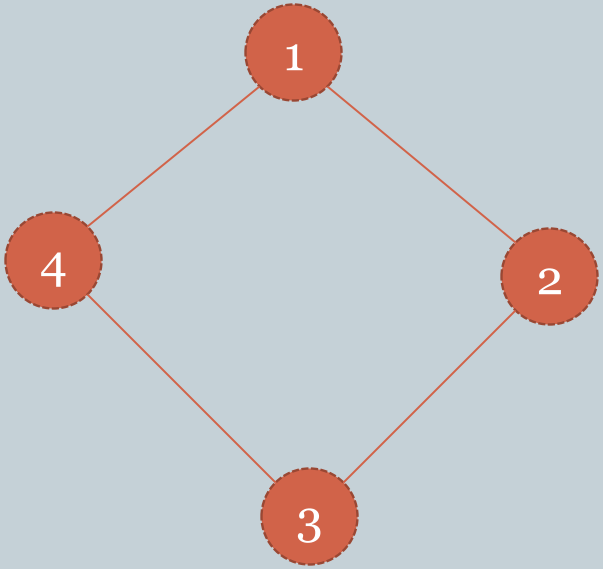
  $\delta = maximum\ node\ degree$
- Phase 2 takes $O(MKlogK)$ steps.
- Phase 4 takes *2MK* steps.

Results of Analysis

- Max of $O(M^2K^{3/2}L\varepsilon^{-2}d_i)$ units of i in network at any time.
- At most $O(M^3K^{5/2}L\varepsilon^{-3})$ rounds required to compute feasible solution, so
- $O(M^3K^{5/2}L\varepsilon^{-3}logK)$ steps to solve the static problem with demand d_i.
- Also works for directed networks, but it's slower by $O(\sqrt{M})$.

“Nasty example”

- 4 nodes, 4 edges, 4 commodities.
- The source for i is node i and the sink is node i+ 2.
- Demand for each commodity is 4 and capacity for each edge in each direction is 1.
- The feasible solution is to ship one of each commodity per round.



An Extension to this work

- In ‘94, Awerbuch and Leighton published a following work, “Improved approximation algorithms for the multi-commodity flow problem and local competitive routing in dynamic networks,” which had 2 added benefits.
  - The new algorithm has a better run time
  - It works “even in networks where edge capacities can vary in an unpredictable and unknown fashion.”
  - The runtime for this algorithm is

    $$O\left(\frac{KL^2M}{\varepsilon^3}\ln^3 M/\varepsilon\right)$$

    steps, which “is competitive with (and in some case superior to) the best previous bound for deterministic algorithms.

Sources

- Baruch Awerbuch and Tom Leighton, “A Simple Local-Control Approximation Algorithm for Multicommodity Flow,” *Foundations of Computer Science*, no. 34 (November 3, 1993): 459-468.
- P.M. Vaidya, “Speeding-up linear programming using fast matrix multiplication,” *Annual Symposium on Foundations of Computer Science* 30 (1989): 332-337.
- “Executive Insights - Tom Leighton,” n.d., http://www.akamai.com/html/perspectives/insight_tl.html.
- “a simple local-control approximation algorithm for multicommodity flow - Google Scholar,” n.d., http://scholar.google.com/scholar?q=a+simple+local-control+approximation+algorithm+for+multicommodity+flow&hl=en&btnG=Search&as_sdt=80000000000001&as_sdtp=on.
- “Multi-commodity flow problem - Wikipedia, the free encyclopedia,” n.d., http://en.wikipedia.org/wiki/Multi-commodity_flow_problem.

A few publications extending the cited work.

- Sankar, A.; Zhen Liu; , "Maximum lifetime routing in wireless ad-hoc networks," *INFOCOM 2004. Twenty-third AnnualJoint Conference of the IEEE Computer and Communications Societies* , vol.2, no., pp. 1089- 1097 vol.2, 7-11 March 2004
doi: 10.1109/INFCOM.2004.1356995
URL: http://ieeexplore.ieee.org/stamp/stamp.jsp?tp=&arnumber=1356995&isnumber=29790
- Baruch Awerbuch and Tom Leighton, "Improved approximation algorithms for the multi-commodity flow problem and local competitive routing in dynamic networks," in *Proceedings of the twenty-sixth annual ACM symposium on Theory of computing - STOC '94* (presented at the the twenty-sixth annual ACM symposium, Montreal, Quebec, Canada, 1994), 487-496, http://portal.acm.org/citation.cfm?doid=195058.195238.