

Brief Announcement: Minimizing the Total Cost of Network Measurements in a Distributed Manner: A Primal-Dual Approach

Baruch Awerbuch^{*}
Johns Hopkins University.
baruch@cs.jhu.edu

Rohit Khandekar
IBM T.J. Watson Research Center.
rkhandekar@gmail.com.

ABSTRACT

We consider the Active Min-Cost Measurement problem to minimize the cost incurred by measuring network link delays. Although the problem has a polynomial representation, its covering LP formulation, for which most of the previous distributed algorithms apply, has an exponential number of variables, one for each path.

We present *first known* distributed $(1 + \epsilon)$ -approximation algorithm for this problem that converges in time that is linear in the maximal path length and poly-logarithmic in the size of the entire network and has polynomial computational overhead. Previous distributed solutions achieving similar approximations required either convergence time that is polynomial or computational overhead that is exponential in the size of the entire network.

Categories and Subject Descriptors

F.2.2 [Analysis of Algorithms and Problem Complexity]: [Non-numerical Algorithms and Problems]

General Terms

algorithms, theory

Keywords

distributed optimization, packing and covering linear programs, network measurements

1. ACTIVE NETWORK MEASUREMENT

An active min-cost link measurement problem uses packets generated by a measurement device to probe the Internet and measure its characteristics. Examples of this approach include the ping utility, the traceroute utility, and the pathchar tool used to determine Internet routing paths and their latencies. Given are a directed graph $G = (V, E)$ with k commodities, each specified by a source s_i , a sink t_i , a measurement cost $b_i > 0$, and an integer $L > 0$. Let \mathcal{P}_i be the set of (simple and non-simple) paths between s_i and t_i of hop-length at most L . Each commodity i needs to decide a frequency f_p^i of “measurements” for $p \in \mathcal{P}_i$. At this

^{*}Partially supported by NSF grants CCF 0515080, ANIR-0240551, CCR-0311795, and CNS-0617883.

frequency, f_p^i measurement probes are done along p per unit time. The objective is to collectively measure each link at a unit frequency. Since the measurements done along each path are independent of other paths, the total frequency of measurement on a link is the sum of individual frequencies of the paths going through that link. Let \mathcal{A}_e^p be the multiplicity of an edge e on a path p . Thus a path p with frequency f_p^i covers an edge e to an extent $\mathcal{A}_e^p f_p^i$. In practice, $L \ll n$ is much smaller as compared to the size of the entire graph; e.g., $L = 1$ in a bipartite case. This problem can be formulated as the following linear program.

$$\begin{aligned} \min \quad & \sum_i \sum_{p \in \mathcal{P}_i} b_i \cdot f_p^i \\ \text{s.t.} \quad & \sum_i \sum_{e \in p} \mathcal{A}_e^p f_p^i \geq 1 \quad \forall \text{ links } e \\ & f_p^i \geq 0 \quad \forall \text{ paths } p \end{aligned} \quad (1)$$

This problem is an instance of the **set cover problem**, with the links being elements and the paths being sets. Aggregating all the sets for a single commodity into a “super-set” allows a possibility of a polynomial overhead solution, but does not let us use the set cover framework in a direct way.

Distributed BILLBOARDS Model. We assume that there are non-strategic agents associated with each commodity and each link. They have access to a global clock. The only allowable communication between agents proceeds as follows: each link-agent (resp. commodity-agent) may send private messages to commodity-agents (resp. link-agents) once per clock cycle, e.g., by posting encrypted messages onto a shared “billboard”, or flooding these messages through the network. For a given error parameter ϵ , the convergence complexity of an algorithm is the number of global clock cycles and the computational complexity is the total amount of computational overhead that each agent incurs to converge to a $(1 + \epsilon)$ -approximate solution.

Denote by $P < n^L$ an upper bound on the total number of paths. Let k and m be the number of commodities and edges resp., and $B = \max_i b_i / \min_i b_i$.

THEOREM 1.1. *There exists a $(1 + \epsilon)$ -approximation algorithm for the active min-cost measurement (1) in distributed BILLBOARDS model converging in $O(L \cdot \frac{\log^2(mB)}{\epsilon^4}) \cdot \log \frac{kB}{\epsilon}$ steps. The computational overhead per agent is bounded by $\tilde{O}(m^2 \cdot L)$.*

The essence of our improvement over prior work [5, 6, 1, 4, 7] is that, without compromising on the distributed convergence time, we accomplish a solution with *polynomial*

computation overhead of $\tilde{O}(\log P) = \tilde{O}(L)$ assuming upper bound of L on path length.

2. THE ALGORITHM

By scaling, we assume that $\min_i b_i = 1$ and $\max_i b_i = B$. For $p \in \mathcal{P}_i$, let $b_p = b_i$. In the algorithm (Figures 1-2), we use $f_e^i = \sum_{p \in \mathcal{P}_i: e \in p} \mathcal{A}_e^p f_p$ to denote the total flow of commodity i through e counting the multiplicities. The algorithm starts by routing zero flow f_e^i , however, allows an additive pre-flow $f_e^i \leftarrow \delta = \epsilon/kB$ over each edge. The total cost of this initialization is only an ϵ fraction of the optimum; hence it can be ignored. We maintain the “residual requirements” r_e with each link e :

$$r_e = \left((mB)^{1/\epsilon} \right)^{-\sum_i f_e^i}.$$

We maintain a variable α such that

$$\alpha \leq \min_p \frac{b_p}{\sum_{e \in p} r_e} \leq \alpha(1 + \epsilon).$$

The “reward” γ_e for each link e is defined as

$$\gamma_e = (1 + \epsilon)\alpha r_e.$$

Since there are exponentially many paths, we cannot afford to maintain their flows explicitly in a polynomial-time implementation. We therefore do not maintain a path decomposition of the flow and instead augment the flow along the *maximal* collection of “most beneficial” paths (similar to a **blocking flow**) subject to the “step-size constraint” that the flow of any commodity along an edge does not increase by a factor of more than $(1 + \beta)$ where $\beta = \epsilon^2 / \log(mB)$. A similar idea was used by Awerbuch et al. [3] for the multi-commodity flow problems.

We can assume that the link-agent e offers a “reward” $\gamma_e = \alpha(1 + \epsilon)r_e$ to each commodity-agent i . For each commodity i such that there is a “profitable” path p with $b_p < \sum_{e \in p} \mathcal{A}_e^p \gamma_e$, we iteratively augment the flow by maximal amount along such paths under the residual capacity constraints. In other words, we send a maximal flow along approximately most beneficial paths such that there is no residual capacity left along any approximately most beneficial path. To compute a (non-simple) most beneficial path of length at most L , we first construct an acyclic directed graph by taking L copies of the vertex-set in the original graph and adding directed edges in consecutive layers. We then compute the longest path (under the reward metric) in this DAG from s_i in the first layer to t_i in the last layer.

A phase has $T_{phase} = O\left(L \cdot \frac{\log(mB)}{\epsilon^2} \cdot \log \frac{1}{\delta}\right)$ steps. In any step, since we are routing a blocking flow along the $(1 + \epsilon)$ approximate most beneficial paths, at least one edge on each $(1 + \epsilon)$ approximate most beneficial path gets saturated and decreases its reward by a factor of $(1 - \epsilon)$. Since the initial flow on any link is δ , after $O\left(\frac{\log(mB)}{\epsilon^2} \cdot \log \frac{1}{\delta}\right)$ saturations, the total flow on that link becomes one. Since any most beneficial path has at most L links, after T_{phase} steps, this path stops to be most beneficial. The details of the proof of correctness are omitted from this extended abstract.

Conclusions. The minimum-cost link measurement problem can be formulated as a set cover problem with exponentially many sets. The polynomial computational overhead was achieved by representing these sets implicitly and us-

1. $\alpha \leftarrow \min_p \frac{b_p}{|p|}$, $\beta \leftarrow \frac{\epsilon^2}{\log(mB)}$, and $\delta \leftarrow \frac{\epsilon}{kB}$.
2. **repeat** for $T = O\left(\frac{\log(mB)}{\epsilon^2}\right)$ phases
 - (a) **repeat** for $T_{phase} = O\left(L \cdot \frac{\log(mB)}{\epsilon^2} \cdot \log \frac{1}{\delta}\right)$ steps:
 - i. update:

$$r_e \leftarrow \left((mB)^{1/\epsilon} \right)^{-\sum_i f_e^i} \text{ and } \gamma_e = (1 + \epsilon)\alpha r_e.$$
 - ii. update residual capacities $u_e^i = \max\{\beta f_e^i, \delta\}$ for each commodity i .
 - (b) Set $\alpha \leftarrow \alpha(1 + \epsilon)$.

Figure 1: A distributed algorithm for link e

- **repeat** for $T \cdot T_{phase}$ steps:
 - while** there is a “profitable” path p with positive residual capacity, i.e., $\min_{e \in p} u_e^i > 0$ and

$$b_p < \sum_{e \in p} \mathcal{A}_e^p \gamma_e$$
 - do**
 1. route $f = \min_{e \in p} u_e^i / \mathcal{A}_e^p$ flow along p : update $f_e^i \leftarrow f_e^i + f \mathcal{A}_e^p$ for all $e \in p$.
 2. update the residual capacities: $u_e^i \leftarrow u_e^i - f \mathcal{A}_e^p$ for all $e \in p$.

Figure 2: A distributed algorithm for commodity i

ing an approach similar to “blocking-flows” to increase all most-beneficial sets simultaneously.

3. REFERENCES

- [1] Baruch Awerbuch and Yossi Azar. Local optimization of global objectives: Competitive distributed deadlock resolution and resource allocation. In *FOCS*, 1994.
- [2] Baruch Awerbuch and Rohit Khandekar. Distributed network monitoring and multicommodity flows: a primal-dual approach. In *PODC*, 2007.
- [3] Baruch Awerbuch, Rohit Khandekar, and Satish Rao. Distributed algorithms for multicommodity flow problems via approximate steepest descent framework. In *SODA*, 2007.
- [4] Yair Bartal, John W. Byers, and Danny Raz. Global optimization using local information with applications to flow control. In *FOCS*, 1997.
- [5] N. Garg and J. Könemann. Faster and simpler algorithms for multicommodity flow and other fractional packing problems. In *FOCS*, 1998.
- [6] Michael Luby and Noam Nissan. A parallel approximation algorithm for positive linear programming. In *STOC*, 1993.
- [7] Neal E. Young. Sequential and parallel algorithms for mixed packing and covering. In *FOCS*, 2001.