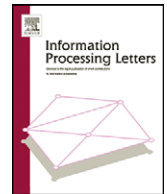




Contents lists available at ScienceDirect

Information Processing Letters

www.elsevier.com/locate/ipl

A $\Theta(\log n)$ -approximation for the set cover problem with set ownership

Mira Gonen*, Yuval Shavitt

School of Electrical Engineering, Tel Aviv University, Ramat Aviv 69778, Israel

ARTICLE INFO

Article history:

Received 21 April 2008

Received in revised form 14 September 2008

Available online 24 September 2008

Communicated by C. Scheideler

Keywords:

Internet

Measurement systems

Traceroute

Approximation algorithms

ABSTRACT

In highly distributed Internet measurement systems distributed agents periodically measure the Internet using a tool called `traceroute`, which discovers a path in the network graph. Each agent performs many traceroute measurements to a set of destinations in the network, and thus reveals a portion of the Internet graph as it is seen from the agent locations. In every period we need to check whether previously discovered edges still exist in this period, a process termed *validation*. To this end we maintain a database of all the different measurements performed by each agent. Our aim is to be able to *validate* the existence of all previously discovered edges in the minimum possible time.

In this work we formulate the validation problem as a generalization of the well know set cover problem. We reduce the set cover problem to the validation problem, thus proving that the validation problem is \mathcal{NP} -hard. We present a $O(\log n)$ -approximation algorithm to the validation problem, where n is the number of edges that need to be validated. We also show that unless $\mathcal{P} = \mathcal{NP}$ the approximation ratio of the validation problem is $\Omega(\log n)$.

© 2008 Elsevier B.V. All rights reserved.

1. Introduction

Our problem arises in the context of highly distributed Internet measurement systems [7,8]. In this type of systems, distributed agents periodically measure the Internet using a tool called `traceroute`, which discovers a path in the network graph.¹ Each agent performs many traceroute measurements to a set of destinations in the network, and thus reveals a portion of the Internet graph as it is seen from the agent locations. While some edges can be seen from many measurement locations, others can be seen only from a handful of locations [7,8,1], which is the major reason for distributing this process. We create a pe-

riodic map by unifying the measurements made by all the agents over this period.

There are many possible heuristics to direct agents to destinations in order to find as many graph edges as possible. However, one thing we have to do in every period is to check whether previously discovered edges still exist in this period, a process termed *validation*. To this end we maintain a database of all the different measurements performed by each agent.² Our aim is to be able to *validate* the existence of all previously discovered edges in the minimum possible time.

A solution to the validation problem is to model each traceroute measurement as a set of edges, and then look for the smallest group of traceroute measurements (the sets) that covers the known graph, e.g., using a set cover logarithmic approximation algorithms [4]. However, this solution may end up finding many groups which are measured by one agent while leaving other agents with little or no measurements to perform. Since all agents measure

* Corresponding author.

E-mail addresses: gonenmir@post.tau.ac.il (M. Gonen), shavitt@eng.tau.ac.il (Y. Shavitt).

¹ The path can be expressed at various levels of abstraction. The most common level in use is the autonomous system (AS) level, where each node in the graph (and thus in the path) represent an AS (or a network) in the Internet.

² The list is kept at the abstraction level we are interested in, e.g., at the AS level.

at roughly the same rate, the termination time of the validation task is determined by the time it will take the agent with the largest number of measurements to complete its task. Thus, our aim is not to minimize the number of measurements that cover the graph, but to minimize the maximal number of measurement which is assigned to the agent with the most measurements. Therefore reducing the validation problem to the set cover problem will not necessarily give us the best solution, so we describe the validation problem as a generalization of the set cover problem.

Our results. We define a new generalization of the set cover problem that is equivalent to the validation problem, and give an $O(\log n)$ -approximation algorithm, where n is the number of edges in the validation problem, and show that our approximation ratio is tight, namely that our generalization of set cover cannot be approximated in polynomial time to within a factor of $o(\log n)$.

Organization. In Section 2 we give notations and a formal definition of the problem. In Section 3 we present an $O(\log n)$ -approximation algorithm for the generalized set cover problem, and prove that this ratio cannot be asymptotically improved.

2. Preliminaries

For an algorithm **A**, denote the objective value of a solution it delivers on an input I by $\mathbf{A}(I)$. An optimal solution is denoted by OPT , and the optimal objective value is denoted by OPT as well. The (absolute) approximation ratio of **A** is defined as the infimum ρ such that for any input I , $\mathbf{A}(I) \leq \rho \cdot \text{OPT}(I)$.

Given a universe $U = \{u_1, \dots, u_n\}$ and a family of its subsets, $\mathcal{S} = \{S_1, \dots, S_k\} \subseteq P(U)$, $\bigcup_{S_j \in \mathcal{S}} S_j = U$, set cover is the problem of finding a minimal sub-family $\bar{\mathcal{S}}$ of \mathcal{S} that covers the whole universe, $\bigcup_{S_j \in \bar{\mathcal{S}}} S_j = U$. Set cover is a classic \mathcal{NP} -hard combinatorial optimization problem, and it is known that it can be approximated to within $\ln n - \ln \ln n + \Theta(1)$ [9,5,10]. By [6,2] it follows that unless $\mathcal{P} = \mathcal{NP}$, there exists a constant $0 < c < 1$ so that set cover cannot be efficiently approximated to within any number smaller than $c \log_2 n$. Feige [3] has shown hardness of approximating set cover in $(1 - o(1)) \ln n$.

We formalize the *validation problem* discussed in the introduction in the following manner: every edge in a traceroute is an element in a universe U . Each traceroute is modeled as a set of elements in U —its edges. Each agent is modeled as a family of sets, indicating the list of traceroutes it can perform. Moreover, each agent has a weight, indicating the number of traceroutes it can perform at a time period. Thus we get the following problem:

Problem 2.1 (*Validation Set Cover (VSC)*). Given a universe U of n elements, a collection of subsets of U , $\mathcal{S} = \{S_1, \dots, S_k\}$, a partition of \mathcal{S} $\pi = \{A_1, \dots, A_m\}$ where $A_i \subseteq \mathcal{S}$, and a weight function $\omega: \pi \rightarrow \mathbb{N}$, find a subcollection $\bar{\mathcal{S}}$ of \mathcal{S} that covers all elements of U such that $\max_{1 \leq i \leq m} \lceil |A_i \cap \bar{\mathcal{S}}| / \omega(A_i) \rceil$ is minimum.

Note. The Validation Set Cover problem is indeed a generalization of the set cover problem—if $m = 1$ then the Validation Set Cover problem is exactly the set cover problem. Thus the Validation Set Cover problem is also \mathcal{NP} -hard.

3. An $O(\log n)$ -approximation algorithm

In this section we give an approximation algorithm for the VSC problem with an approximation ratio of $O(\log n)$. We then show that this is the best ratio possible by showing a lower bound of $\Omega(\log n)$ on the approximation ratio.

The greedy strategy applies naturally to the VSC problem: iteratively for each $1 \leq i \leq m$ pick $\omega(A_i)$ sets in A_i that cover the maximum number of elements in U that are still uncovered. The algorithm stops when all the elements in U are covered, and outputs the number of steps performed.

Algorithm 1. Greedy VSC algorithm.

- (1) $\ell \leftarrow 0$
- (2) $C \leftarrow \emptyset$
- (3) while $C \neq U$
 - (a) $\ell \leftarrow \ell + 1$
 - (b) for $1 \leq i \leq m$
 - (i) repeat $\omega(A_i)$ times
 - (A) find a set S_j such that $S_j \in A_i$ and $S_j \cap (U \setminus C)$ is maximum
 - (B) pick S_j
 - (c) $C \leftarrow C \cup S_j$
- (4) output ℓ

Theorem 1. Algorithm 1 gives an approximation ratio of $O(\log n)$.

We next prove Theorem 1. We first define the ℓ -residual VSC problem. The input to this problem is the input to the VSC problems after ℓ steps of the algorithm, with the same objective function:

- Let n_ℓ be the number of elements in U that remain after ℓ steps of the algorithm. For $\ell = 0$, $n_\ell = n$.
- Let C_ℓ be the set of elements in U that are covered until step ℓ .
- For all $1 \leq j \leq k = |\mathcal{S}|$
 - let $S_j^\ell = S_j \setminus C_\ell$,
 - for all $1 \leq i \leq m$ let $A_i^\ell = A_i \setminus \{S_j \in A_i \mid S_j \text{ has been picked until step } \ell\}$,
 - let $\mathcal{S}^\ell = \{S_j^\ell \mid S_j^\ell \neq \emptyset\}$.
- For all $1 \leq i \leq m$ let $\omega(A_i^\ell) = \omega(A_i)$.
- Let OPT_ℓ be the optimal solution of the residual input after ℓ steps.³

Then

$$\text{OPT}_\ell = \min_{\bar{\mathcal{S}}^\ell} \max_{1 \leq i \leq m} \left\lceil \frac{|A_i^\ell \cap \bar{\mathcal{S}}^\ell|}{\omega(A_i^\ell)} \right\rceil,$$

³ Recall that OPT is the optimal solution.

where \bar{S}^ℓ is a subcollection of S^ℓ that covers all elements of $U \setminus C_\ell$.

Thus we get the following claim:

Claim 3.1. *At step $\ell \geq 1$ of Algorithm 1 at least $n_{\ell-1}/\text{OPT}_{\ell-1}$ elements in U are covered.*

Proof. The main observation is that any optimal algorithm covers all the elements in OPT stages. Obviously, there exists a stage in which at least n/OPT elements are covered. Since the order of the stages does not change OPT , assume without loss of generality that at stage 1 any optimal algorithm covers at least n/OPT elements. Thus, if $\ell = 1$ then, since Algorithm 1 picks a set that covers the maximum number of elements, it holds that at least $n/\text{OPT} = n_{\ell-1}/\text{OPT}_{\ell-1}$ elements are covered at step ℓ . If $\ell > 1$ then an optimal algorithm covers all the $n_{\ell-1}$ remaining elements of $U \setminus C_{\ell-1}$ in $\text{OPT}_{\ell-1}$ steps. Since Algorithm 1 picks a set that covers the maximum number of remaining elements, it holds that at least $n_{\ell-1}/\text{OPT}_{\ell-1}$ elements are covered at step ℓ . \square

Using the above claim and the observation that for all ℓ $\text{OPT}_\ell \leq \text{OPT}$, we get the following lemma.

Lemma 3.2. $n_\ell \leq n(1 - 1/\text{OPT})^{\ell-1}$.

Proof. By induction on ℓ :

$$\begin{aligned} n_1 &\leq n - \frac{n}{\text{OPT}} = n\left(1 - \frac{1}{\text{OPT}}\right), \\ n_2 &\leq n_1 - \frac{n_1}{\text{OPT}_1} = n_1\left(1 - \frac{1}{\text{OPT}_1}\right) \\ &\leq n_1\left(1 - \frac{1}{\text{OPT}}\right) \leq n\left(1 - \frac{1}{\text{OPT}}\right)^2. \end{aligned}$$

Assume that for all $i < \ell$ it holds that $n_i \leq n(1 - 1/\text{OPT})^i$. Then

$$\begin{aligned} n_\ell &\leq n_{\ell-1} - \frac{n_{\ell-1}}{\text{OPT}_{\ell-1}} \\ &\leq n\left(1 - \frac{1}{\text{OPT}}\right)^{\ell-1} - \frac{n_{\ell-1}}{\text{OPT}_{\ell-1}} \\ &\leq n\left(1 - \frac{1}{\text{OPT}}\right)^{\ell-1} - \frac{n\left(1 - \frac{1}{\text{OPT}}\right)^{\ell-1}}{\text{OPT}} \\ &= n\left(1 - \frac{1}{\text{OPT}}\right)^\ell. \quad \square \end{aligned} \tag{1}$$

Proof of Theorem 1. In the worst case the algorithm stops after $\ell + 1$ steps for the minimal ℓ such that $n_\ell \leq 1$. Since by the above lemma $n_\ell \leq n(1 - 1/\text{OPT})^\ell$, for that ℓ with $n(1 - 1/\text{OPT})^\ell \leq 1$ it holds that $n_\ell \leq 1$.

$$\begin{aligned} n\left(1 - \frac{1}{\text{OPT}}\right)^\ell &\leq 1 \\ \Leftrightarrow \left(1 - \frac{1}{\text{OPT}}\right)^\ell &\leq \frac{1}{n} \end{aligned}$$

$$\begin{aligned} \Leftrightarrow \ell &\leq \frac{\log(1/n)}{\log\left(1 - \frac{1}{\text{OPT}}\right)} = \frac{\log n}{\log\left(\frac{\text{OPT}}{\text{OPT}-1}\right)} \\ \Leftrightarrow \ell &\leq \frac{\log n}{\log\left(1 + \frac{1}{\text{OPT}-1}\right)}. \end{aligned} \tag{2}$$

We now prove that $\frac{\log n}{\log(1+1/(\text{OPT}-1))} \leq \log n \cdot \text{OPT}$. It holds that

$$\begin{aligned} \frac{\log n}{\log\left(1 + \frac{1}{\text{OPT}-1}\right)} &\leq \log n \cdot \text{OPT} \\ \Leftrightarrow 1 + \frac{1}{\text{OPT}-1} &\geq e^{1/\text{OPT}}. \end{aligned}$$

According to the Taylor series we have that

$$f(x) = \sum_{i=0}^n f^{(i)}(0) \frac{x^i}{i!} + R_n(x),$$

where

$$R_n(x) = \frac{f^{(n+1)}(c)}{(n+1)!} x^{n+1},$$

for some $0 \leq c \leq x$. For $f(x) = e^x$ we get that

$$e^x = \sum_{i=0}^n \frac{x^i}{i!} + e^c \frac{x^{n+1}}{(n+1)!},$$

for some $0 \leq c \leq x$. For $x = 1/\text{OPT}$ and $n = 2$ we get that

$$e^{1/\text{OPT}} = 1 + \frac{1}{\text{OPT}} + \frac{1}{2\text{OPT}^2} + \frac{e^c}{6\text{OPT}^3},$$

for some $0 \leq c \leq 1/\text{OPT}$. Now,

$$\begin{aligned} 1 + \frac{1}{\text{OPT}-1} &\geq 1 + \frac{1}{\text{OPT}} + \frac{1}{2\text{OPT}^2} + \frac{e^c}{6\text{OPT}^3} \\ \Leftrightarrow \frac{1}{\text{OPT}-1} - \frac{1}{\text{OPT}} &\geq \frac{1}{2\text{OPT}^2} + \frac{e^c}{6\text{OPT}^3} \\ \Leftrightarrow \frac{1}{(\text{OPT}-1)\text{OPT}} &\geq \frac{1}{2\text{OPT}^2} + \frac{e^c}{6\text{OPT}^3} \\ \Leftrightarrow \frac{1}{\text{OPT}-1} &\geq \frac{1}{2\text{OPT}} + \frac{e^c}{6\text{OPT}^2} \\ \Leftrightarrow 6\text{OPT}^2 &\geq (\text{OPT}-1)(3\text{OPT} + e^c). \end{aligned} \tag{3}$$

The last inequality is valid since $e^c < 3$ (as $c \leq 1/\text{OPT}$). Thus $1 + 1/(\text{OPT}-1) \geq e^{1/\text{OPT}}$, so $\frac{\log n}{\log(1+1/(\text{OPT}-1))} \leq \log n \cdot \text{OPT}$. Therefore the number of steps used by Algorithm 1 is at most $1 + \log n \cdot \text{OPT}$, and the theorem follows. \square

By [6,2] it follows that unless $\mathcal{P} = \mathcal{NP}$ the approximation ratio of the set cover problem is $\Omega(\log n)$. Since for $m = 1$ and for $1 \leq i \leq m$, $\omega(A_i) = 1$ the VSC problem is exactly the set cover problem, we get that unless $\mathcal{P} = \mathcal{NP}$ the approximation ratio of the VSC problem is $\Omega(\log n)$.

Acknowledgements

This research was supported in part under ISF grant number 8008, and by the Yeshaya Horowitz Association through the Center for Complexity Science.

References

- [1] D. Achlioptas, A. Cluset, D. Kempe, C. Moore, On the bias of traceroute sampling or, power-law degree distributions in regular graphs, in: Proc. 37th Symposium on the Theory of Computing (STOC), Baltimore, MD, USA, May 2005, pp. 694–703.
- [2] N. Alon, D. Moshkovitz, M. Safra, Algorithmic construction of sets for k -restrictions, in: ACM Transactions on Algorithms (TALG), 2006, pp. 153–177.
- [3] U. Feige, A threshold of $\ln n$ for approximating set cover, J. ACM 45 (4) (1998) 634–652.
- [4] D.S. Johnson, Approximation algorithms for combinatorial problems, J. Comput. System Sci. 9 (1974) 256–278.
- [5] L. Lovász, On the ratio of optimal integral and fractional covers, SIAM J. Discrete Math. 13 (1975) 383–390.
- [6] R. Raz, S. Safra, A sub-constant error-probability PCP characterization of NP, in: Proc. 29th Symposium on the Theory of Computing (STOC), 1997, pp. 475–484.
- [7] Y. Shavitt, E. Shir, DIMES: Let the internet measure itself, ACM SIGCOMM Computer Communication Review 35 (2005) 71–74.
- [8] Y. Shavitt, E. Shir, DIMES: Let the Internet Measure Itself, in: M.H.W. Weber (Ed.), Distributed and Grid Computing—Science Made Transparent for Everyone. Principles, Applications and Supporting Communities, Tectum Verlag, 2008.
- [9] P. Slavik, Improved approximations of packing and covering problems, in: Proc. 27th Symposium on the Theory of Computing (STOC), Baltimore, MD, USA, May 1995, pp. 268–276.
- [10] A. Srinivasan, Improved approximations guarantees for packing and covering integer programs, SIAM J. Comput. 29 (2) (1999) 648–670.