ARTICLE IN PRESS

Discrete Applied Mathematics ▮ (▮▮▮▮) ▮▮▮–▮▮▮

Contents lists available at ScienceDirect

Discrete Applied Mathematics

journal homepage: www.elsevier.com/locate/dam

Note

Approximately counting paths and cycles in a graph

Masaki Yamamoto

Seikei University, Japan

ARTICLE INFO

Article history:
Received 10 January 2014
Received in revised form 6 July 2016
Accepted 4 September 2016
Available online xxxx

Keywords:
Approximate counting
#P-hard
FPRAS
Inapproximability

ABSTRACT

We consider the time complexity of problems of counting paths and cycles in a graph, respectively. We first show that these two problems are #P-hard. Then, we show an inapproximability result: there is no FPRAS for approximately counting paths and cycles in a graph, respectively unless $RP = NP$.

1. Introduction

The complexity of counting problems has been extensively studied since Valiant [14] initiated this research area. He defined the class #P of counting problems as well as the notion of #P-hard problems, and showed that the problem of evaluating the permanent of a 0/1 matrix, which is equivalent to the problem of counting perfect matchings in a bipartite graph, is #P-hard. Since then, many counting and evaluating problems have been shown to be #P-hard, e.g., counting bases of a matroid [15], satisfying assignments of a 2-CNF formula [15], Hamilton paths (cycles) in a graph [15], independent sets in a graph of degree up to three [5], and evaluating the partition function of the Ising model [8], to name but a few.

On the other hand, approximation algorithms have been proposed for #P-hard counting problems, the same way as many approximation algorithms have been proposed for NP-hard optimization problems, e.g., approximately counting matchings in a graph [9], perfect matchings in a bipartite graph [7], Hamilton cycles in a dense graph [1], independent sets in a sparse graph [12,5,16] (degree up to five [16]), and approximately evaluating the partition function of the Ising model with certain parameters [8], the permanent of a non-negative matrix [10], to name but a few.

While many approximability results have been shown for #P-hard counting problems, inapproximability results have also been shown (under usual complexity theory assumptions) for several problems such as, approximately counting satisfying assignments of a CNF formula [17], independent sets in a sparse graph [12,2,13] (of maximum degree six [13]), satisfying assignments of a CSP formula of a certain type [4].

In this paper, we investigate the possibility of approximately counting paths and cycles in a graph, which are fundamental graph structures. Before we do that, we establish the #P-hardness of the two problems. As far as the author knows, there is no formal result showing the #P-hardness of the problems.

Theorem 1.1. *The problem of counting paths (cycles) in a graph is #P-hard.*

As mentioned above, many efficient approximation algorithms have been proposed for #P-hard counting problems. Such an approximation algorithm is called a "fully polynomial-time randomized approximation scheme", "FPRAS" for short. (See the next section for the precise definition.)

E-mail address: yamamoto@st.seikei.ac.jp.

http://dx.doi.org/10.1016/j.dam.2016.09.002
0166-218X/

Please cite this article in press as: M. Yamamoto, Approximately counting paths and cycles in a graph, Discrete Applied Mathematics (2016), http://dx.doi.org/10.1016/j.dam.2016.09.002

Theorem 1.2. *There is no FPRAS for counting paths (cycles) in a graph unless RP = NP.*

Proof technique

Our proofs use polynomial-time *truth-table reductions*, which are Turing reductions where later queries do not depend on earlier answers. The idea of our Turing reductions to show the #P-hardness of exactly counting paths (cycles) is inspired by the reduction of Jerrum [6], that showed the #P-hardness of the problem of counting trees in a graph. Given a graph G, each edge of G is replaced with a gadget graph of size i to produce the graph G_i. Then, the numbers of (s, t)-paths in G of all lengths determine the number of (s, t)-paths in each G_i. The function is linear and invertible, so the numbers of (s, t)-paths in G_i for all sizes i determine the numbers of (s, t)-paths in G of all lengths, including the number of Hamilton (s, t)-paths.

On the other hand, the idea of our Turing reductions to show the inapproximability results is inspired by the reduction of Dyer, Goldberg, Greenhill, and Jerrum [3], that showed the inapproximability result for counting independent sets in a graph. Given a graph G, each edge of G is replaced with a gadget graph so that the number of paths is approximately equal to a known multiple of the number of Hamilton paths in G when the gadget graph is sufficiently large.

Organization

In the next section, we present notions and notations used in this paper. In Section 3, we first show the #P-hardness of the two counting problems. Then, we show the inapproximability results for the counting problems. In the concluding section, we mention future work in this vein.

2. Preliminaries

In this section, we present notions and notations used in this paper. We deal with simple graphs, that is, graphs with no multi-edges and no self-loops. We mean by a *path* (resp. *cycle*) a simple path (resp. cycle). The *length* of a path P (resp. cycle C), denoted by $|P|$ (resp. $|C|$), is the number of edges of P (resp. C). The problem of counting paths (resp. cycles), denoted by #Paths (resp. #Cycles), is, given a graph G, to compute the number of all the paths (resp. cycles) which are sub-graphs of G.

A counting problem $f : \Sigma^* \to \mathbb{N}$ is in *#P* if there is a non-deterministic polynomial-time Turing machine N such that for any string $x \in \Sigma^*$, the number of accepting paths of $N(x)$ is $f(x)$. A counting problem is *#P-hard* if every counting problem in #P is reduced to the problem by a polynomial-time Turing reduction. A counting problem is *#P-complete* if the problem is in #P and is #P-hard.

For showing the #P-hardness of #Paths and #Cycles, we reduce from basic #P-hard problems. A *Hamilton path* (resp. *Hamilton cycle*) in G is a path (resp. cycle) consisting of all the vertices of G. The problem of counting Hamilton paths (resp. Hamilton cycles), denoted by #HamPaths (resp. #HamCycles), is, given a graph G, to compute the number of Hamilton paths (resp. Hamilton cycles) in G. In particular, the problem of counting Hamilton paths where two endpoints are specified is denoted by #stHamPaths.

Fact 1 (*Valiant [15]*)**.** #HamPaths *is #P-complete. Moreover,* #stHamPaths *is #P-complete.*

Fact 2 (*Valiant [15]*)**.** #HamCycles *is #P-complete.*

For showing the inapproximability results, we need to present the notion of approximability of counting problems. The following definition of "FPRAS" for counting problems was first given by Karp and Luby [11].

Definition 2.1. A *randomized approximation scheme* (*RAS* for short) for a counting problem is a randomized algorithm which, given a string $x \in \Sigma^*$ and a real number $\epsilon > 0$, outputs $A(x)$ such that

$$\Pr\{|A(x) - \mathrm{OPT}(x)| \leq \epsilon \cdot \mathrm{OPT}(x)\} \geq 3/4.$$

A *fully polynomial-time randomized approximation scheme* (*FPRAS* for short) for a counting problem is a RAS which runs in time polynomial in the size of x and ϵ^{-1}.

As is mentioned in the Introduction, many FPRASs have been found for #P-hard counting problems. On the other hand, Zuckerman [17] showed an inapproximability result for #SAT in the following sense.

Theorem 2.1 (*Zuckerman [17]*)**.** #SAT *does not have an FPRAS unless RP = NP.*

For classifying counting problems with respect to approximability, Dyer, Goldberg, Greenhill, and Jerrum [3] introduced a notion of reducibility for counting problems.

Definition 2.2. Let $f, g : \Sigma^* \to \mathbb{N}$ be arbitrary counting problems. An *approximation-preserving reduction* (*AP reduction* for short) from f to g is a probabilistic oracle Turing machine M that takes as input a pair $(x, \epsilon) \in \Sigma^* \times (0, 1)$, and satisfies the following three conditions:

Please cite this article in press as: M. Yamamoto, Approximately counting paths and cycles in a graph, Discrete Applied Mathematics (2016), http://dx.doi.org/10.1016/j.dam.2016.09.002