Theoretical Computer Science 547 (2014) 104–116

Contents lists available at ScienceDirect

Theoretical Computer Science

www.elsevier.com/locate/tcs

Approximating the minimum cycle mean ☆

Krishnendu Chatterjee [a], Monika Henzinger [b], Sebastian Krinninger [b], Veronika Loitzenbauer [b,*], Michael A. Raskin [c,d]

[a] Institute of Science and Technology, Am Campus 1, 3400 Klosterneuburg, Austria
[b] University of Vienna, Faculty of Computer Science, Währinger Straße 29, 1090 Vienna, Austria
[c] Independent University of Moscow, Bolshoy Vlasyevskiy 11, 115162 Moscow, Russia
[d] Moscow Institute of Physics and Technology, 9 Institutskiy per., 141700 Dolgoprudny, Russia

ARTICLE INFO

Article history:
Received 21 November 2013
Received in revised form 13 May 2014
Accepted 25 June 2014
Available online 30 June 2014
Communicated by C. Kaklamanis

Keywords:
Quantitative verification
Graph algorithm
Mean-payoff objective
Approximation algorithm

ABSTRACT

We consider directed graphs where each edge is labeled with an integer weight and study the fundamental algorithmic question of computing the value of a cycle with minimum mean weight. Our contributions are twofold: (1) First we show that the algorithmic question is reducible to the problem of a logarithmic number of *min-plus* matrix multiplications of $n \times n$-matrices, where n is the number of vertices of the graph. (2) Second, when the weights are nonnegative, we present the first $(1+\epsilon)$-approximation algorithm for the problem and the running time of our algorithm is $\widetilde{O}(n^{\omega}\log^3(nW/\epsilon)/\epsilon)$,[1] where $O(n^{\omega})$ is the time required for the *classic* $n \times n$-matrix multiplication and W is the maximum value of the weights. With an additional $O(\log(nW/\epsilon))$ factor in space a cycle with approximately optimal weight can be computed within the same time bound.

1. Introduction

Minimum cycle mean problem. We consider the fundamental algorithmic problem of computing the value of a minimum mean-weight cycle in a finite directed graph. The input to the problem is a directed graph $G = (V, E, w)$ with a finite set V of n vertices, a finite set E of m edges, and a weight function w that assigns an integer weight to every edge. Given a cycle C, the mean weight $\mu(C)$ of the cycle is the ratio of the sum of the weights of the cycle and the number of edges in the cycle. The algorithmic question asks to compute $\mu = \min\{\mu(C) \mid C \text{ is a cycle}\}$: the minimum cycle mean. The minimum cycle mean problem is an important problem in combinatorial optimization and has a long history of algorithmic study. An $O(nm)$-time algorithm for the problem was given by Karp [2]. The current best known algorithm for the problem by Orlin and Ahuja, which is over two decades old, requires $O(m\sqrt{n}\log(nW))$ time [3], where W is the maximum absolute value of the weights.

Applications. The minimum cycle mean problem is a basic combinatorial optimization problem that has numerous applications in network flows [4]. In the context of formal analysis of reactive systems, the performance of systems as well as the average resource consumption of systems is modeled as the minimum cycle mean problem. A reactive system is modeled

☆ A preliminary version appeared in [1].

* Corresponding author. Tel.: +43 1 4277 78320.
E-mail address: veronika.loitzenbauer@univie.ac.at (V. Loitzenbauer).

[1] The $\widetilde{O}$-notation hides a polylogarithmic factor.

http://dx.doi.org/10.1016/j.tcs.2014.06.031
0304-3975/

Table 1
Current fastest asymptotic running times for computing the minimum cycle mean.

Reference	Running time	Approximation	Range
Karp [2]	$O(mn)$	exact	$[-W, W]$
Orlin and Ahuja [3]	$O(m\sqrt{n}\log(nW))$	exact	$[-W, W]\cap\mathbb{Z}$
Sankowski [9] (implicit)	$\widetilde{O}(Wn^{\omega}\log(nW))$	exact	$[-W, W]\cap\mathbb{Z}$
Butkovic and Cuninghame-Green [10]	$O(n^2)$	exact	$\{0, 1\}$
This paper	$\widetilde{O}(n^{\omega}\log^3(nW/\epsilon)/\epsilon)$	$1+\epsilon$	$[0, W]\cap\mathbb{Z}$

as a directed graph, where vertices represent states of the system, edges represent transitions, and every edge is assigned a *nonnegative* integer representing the resource consumption (or delay) associated with the transition. The computation of a minimum average resource consumption behavior (or minimum average response time) corresponds to the computation of the minimum cycle mean. Several recent works model other quantitative aspects of system analysis (such as robustness) also as the mean-weight problem (also known as *mean-payoff* problem) [5,6].

Results. This work contains the following results.

1. *Reduction to min-plus matrix multiplication.* We show that the minimum cycle mean problem is reducible to the problem of a logarithmic number of min-plus matrix multiplications of $n \times n$-matrices, where n is the number of vertices of the graph. Our result implies that algorithmic improvements for min-plus matrix multiplication will carry over to the minimum cycle mean problem with a logarithmic multiplicative factor in the running time.
2. *Faster approximation algorithm.* When the weights are nonnegative, we present the first $(1+\epsilon)$-approximation algorithm for the problem that outputs $\hat{\mu}$ such that $\mu \leq \hat{\mu} \leq (1+\epsilon)\mu$ and the running time of our algorithm is $\widetilde{O}(n^{\omega}\log^3(nW/\epsilon)/\epsilon)$. As usual, the $\widetilde{O}$-notation is used to "hide" a polylogarithmic factor, i.e., $\widetilde{O}(T(n,m,W)) = O(T(n,m,W)\cdot \text{polylog}(n))$, and $O(n^{\omega})$ is the time required for the *classic* $n \times n$-matrix multiplication. The current best known bound for ω is $\omega < 2.3727$ [7,8].
   For the computation of $\hat{\mu}$, $O(n^2)$ space is needed. If $O(n^2\log(nW/\epsilon))$ space is used instead, i.e., the intermediate results of the approximation algorithm are saved, we can additionally output a cycle with mean weight at most $\hat{\mu}$.
   The worst case complexity of the current best known algorithm for the minimum cycle mean problem is $O(m\sqrt{n}\log(nW))$ [3], which could be as bad as $O(n^{2.5}\log(nW))$. Thus for $(1+\epsilon)$-approximation our algorithm provides better dependence on n.
   Note that in applications related to systems analysis the weights are always nonnegative (they represent resource consumption, delays, etc.); and the weights are typically small, whereas the state space of the system is large. Moreover, due to imprecision in modeling, approximations in weights are already introduced during the modeling phase. Hence $(1+\epsilon)$-approximation of the minimum cycle mean problem with small weights and large graphs is a relevant algorithmic problem for reactive system analysis, and we improve the long-standing complexity of the problem.
   The key technique that we use to obtain the approximation algorithm is a combination of the value iteration algorithm for the minimum cycle mean problem, and a technique used for an approximation algorithm for all-pair shortest path problem for directed graphs. Table 1 compares our algorithm with the asymptotically fastest existing algorithms.

Outline. In the rest of this section we discuss related work and motivate the minimum cycle mean problem by its relation to negative cycle detection. We summarize all needed definitions in Section 2. In Section 3 we describe how min-plus matrix multiplication can be used to compute the minimum cycle mean exactly. In Section 4 we present our approximation algorithm and prove its correctness and running time. In Section 5 we show how at the cost of storing the intermediate results an approximately optimal cycle can be output.

1.1. Related work

The minimum cycle mean problem is basically equivalent to solving a deterministic Markov decision process (MDP) [11]. The latter can also be seen as a single-player mean-payoff game [12,13,11]. We distinguish two types of algorithms: algorithms that are independent of the weights of the graph and algorithms that depend on the weights in some way. By W we denote the maximum absolute edge weight of the graph.

Algorithms independent of weights. The classic algorithm of Karp [2] uses a dynamic programming approach to find the minimum cycle mean and runs in time $O(mn)$. A corresponding cycle can easily be computed given the outcome of the algorithm. The main drawback of Karp's algorithm is that its best-case and worst-case running times are the same. The algorithms of Hartmann and Orlin [14] and of Dasdan and Gupta [15] address this issue, but also have a worst-case complexity of $O(mn)$. By solving the more general parametric shortest path problem, Karp and Orlin [16] can compute the minimum cycle mean in time $O(mn\log n)$. Young, Tarjan, and Orlin [17] improve this running time to $O(mn + n^2\log n)$.

A well known algorithm for solving MDPs is the value iteration algorithm. In each iteration this algorithm spends time $O(m)$ and in total it performs $O(nW)$ iterations. Madani [18] showed that, for *deterministic* MDPs (i.e., weighted graphs