Contents lists available at ScienceDirect

Discrete Applied Mathematics

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

Randomized algorithms for finding the shortest negative cost cycle in networks

James B. Orlin[a], K. Subramani[b,*], Piotr Wojciechowki[b]

[a] *Sloan School of Management, MIT, Cambridge, MA, United States*

[b] *LCSEE, West Virginia University, Morgantown, WV, United States*

ARTICLE INFO

Article history:
Received 24 December 2016
Received in revised form 30 May 2017
Accepted 12 October 2017
Available online xxxx

ABSTRACT

In this paper, we design and analyze a fast, randomized algorithm for the problem of finding a negative cost cycle having the smallest number of edges in a directed, weighted graph. This problem will henceforth be referred to as the Shortest Negative Cost Cycle problem (SNCC). SNCC is closely related to the problem of checking whether a directed, weighted graph contains a negative cost cycle (NCCD). NCCD is an extremely well-studied problem within the domains of operations research and theoretical computer science. SNCC is important in its own right and finds several applications in program verification (Satisfiability modulo theories), abstract interpretation and real-time scheduling. It is also an important subroutine in solving the generalized submodular flow problem, which has applications in trading networks. The randomized algorithm presented in this paper for SNCC determines a shortest negative cost cycle with probability at least $(1 - e^{-1})$ in $O(m \cdot n \cdot \log n)$ time, on a network with n vertices and m edges. This is, in general, a significant improvement over the best deterministic bound of $O(m \cdot n \cdot |C^*|)$ over the same parameters, where C^* is a shortest negative cost cycle. This algorithm requires $\Omega(n \cdot \log n)$ random bits. We then propose a second randomized algorithm that runs in $O(m \cdot n \cdot \log n)$ *expected time* and requires $O(n)$ random bits (expected).

1. Introduction

The focus of this paper is on the design and analysis of a randomized algorithm for finding a negative cost cycle having the smallest number of edges in a directed, weighted graph. Let $\mathbf{G} = \langle \mathbf{V}, \mathbf{E}, \mathbf{c} \rangle$ denote a directed graph (network) with vertex set $\mathbf{V}$, edge set $\mathbf{E}$ and cost (weight) function $\mathbf{c} : \mathbf{E} \to \mathbb{Z}$. We assume that $\mathbf{G}$ has n vertices and m edges, i.e., $|\mathbf{V}| = n$ and $|\mathbf{E}| = m$. The terms cost and weight are used interchangeably. One of the fundamental problems in operations research and theoretical computer science is the problem of checking if $\mathbf{G}$ has a negative cost cycle (NCCD). NCCD is an extremely well-studied problem, inasmuch as it arises in a number of domains such as real-time scheduling [14], program verification [9,8] and image segmentation [11]. The literature documents several algorithms for NCCD [1]. To date, the fastest strongly polynomial time algorithm for NCCD is the Bellman-Ford procedure which runs in $O(m \cdot n)$ time. [15] describes a scaling approach that runs in $O(\sqrt{n} \cdot m \cdot \log N)$ time, where N is the largest absolute value of a weight of any edge. The scaling approach is superior to Bellman-Ford approaches, when the edge weights are small integers.

In some practical applications, the mere isolation of a negative cost cycle is inadequate. Indeed, these applications mandate additional constraints on the negative cycle of interest. For instance, in the performance analysis of discrete event

* Corresponding author.
E-mail addresses: jorlin@mit.edu (J.B. Orlin), ksmani@csee.wvu.edu (K. Subramani), pwojciec@mix.wvu.edu (P. Wojciechowki).

https://doi.org/10.1016/j.dam.2017.10.011
0166-218X/© 2017 Elsevier B.V. All rights reserved.

systems [2], one needs a cycle whose mean cost (the cost divided by the number of edges) is minimum. This problem is known as the minimum mean cycle problem (MMC). MMC finds several applications in graph theory [16], VLSI Design [28,4] and real-time embedded systems [18]. In [17], an $O(m \cdot n)$ time algorithm is described for this problem. Likewise, in Satisfiability Modulo Theory (SMT) solvers [3,12], the goal is to identify "small" certificates of infeasibility. When the constraint system in question is a difference constraint system (DCS), then the problem of identifying small certificates leads naturally to the formulation of the shortest negative cost cycle problem (SNCC), that is finding a negative cost cycle with the smallest number of edges. We will refer to a shortest negative cost cycle in a graph as a shortest NCC to avoid using SNCC to refer to both the problem and the cycle.

SNCC also finds applications within the field of abstract interpretation [10]. In abstract interpretation, the semantics of a program are interpreted over a suitably chosen domain. One such domain is polyhedra defined by difference constraints [25]. It is well-known that a conjunction of difference constraints can be represented by a suitably chosen constraint network [7], such that the constraint network contains a negative cost cycle if and only if the difference constraint system is infeasible. The shortest negative cost cycle corresponds to a minimum infeasible subset.

It is important to note that SNCC arises as a fundamental subproblem in optimizing the generalized submodular flow problem (GSFP). [21] developed the GSFP as a common generalization of the submodular flow problem [13] as well as the valuated matroid intersection problem [19] and [20]. The optimality conditions for GSFP require that there is no negative cost cycle in a related auxiliary graph. If there is a negative cost cycle, then an improved solution can be obtained by augmenting around a shortest negative cost cycle in the auxiliary graph.

GSFP finds applications in economic allocation problems. [21] and [22] showed that the efficient allocation problem for a two-sided economy with multiple buyers and sellers can be formulated as a generalized submodular flow problem on a bipartite network. [5] extended the results on 2-sided economies to trading networks that are not bipartite. They modeled problems on trading networks as GSFPs. Their paper offered a unifying framework for trading networks involving bilateral trades under what is referred to as the full substitutability assumption.

In each of these applications of GSFP, a competitive equilibrium can be obtained by sending flow around shortest negative cost cycles in an auxiliary network.

The first polynomial time algorithm for SNCC is detailed in [26]. The algorithm therein is based on matrix multiplication and runs in $O(n^3 \cdot \log n)$ time. [26] also discusses several applications of SNCC and establishes the connection between SNCC and the minimum unsatisfiable core of a difference constraint system. The problem of finding minimum unsatisfiable cores lies at the heart of several SMT solvers [6]. [27] developed two alternative algorithms for SNCC; the *edge-progression* approach which runs in $O(n^2 + m \cdot n \cdot |C^*|)$ time and the *edge-relaxation* approach which runs in $O(m \cdot n \cdot |C^*|)$ time. A detailed implementation profile of the above three algorithms is also presented. In [29], the problem of finding the shortest NCC in planar networks is discussed. The authors exploit the fact that a planar network can be recursively decomposed and design a divide and conquer algorithm that runs in $O(n^{1.5} \cdot |C^*|)$ time, where C^* is a negative cycle having the smallest number of edges.

This paper focuses on the design and analysis of randomized algorithms for SNCC.

The main contributions of this paper are as follows:

1. A randomized algorithm for SNCC, which runs in deterministic time $O(m \cdot n \cdot \log n)$ and has error probability $\frac{1}{e}$. This probability can be reduced by repeatedly running the algorithm.
2. A randomized algorithm for SNCC, which runs in expected time $O(m \cdot n \cdot \log n)$, and in which the expected number of random bits is $O(n)$.

The remainder of this paper is organized as follows: Section 2 describes an algorithm for extracting negative cost cycles in directed graphs using walks. In Section 3, we describe the first of our two randomized algorithms. In Section 4, we propose another randomized algorithm for SNCC. We conclude in Section 5 by summarizing our contributions and identifying avenues for future research.

2. Extracting a negative cycle in a directed graph from closed walks

In this section, we describe an algorithm for extracting a negative cost cycle in a directed graph from closed walks.

A *walk* from a vertex x_i to a vertex x_j is a directed path commencing at x_i and ending at x_j. Note that the path need not be simple, i.e., a walk is permitted to repeat nodes and edges. If a walk commences and ends on the same vertex, it is said to be *closed*. A walk of k or fewer edges is called a k-walk.

A Bellman-Ford variant can be used to find a minimum cost closed k-walk around x_j in $O(m \cdot k)$ time. If a closed walk W has negative cost, then W contains a negative cost cycle. That is, W can be expressed as the union of cycles, such that at least one of the cycles has a negative cost.

Some observations are in order:

1. Let $d_i^{(k)}(j)$ denote the cost of the minimum cost k-walk from vertex x_i to vertex x_j. (We permit the case, where $i = j$.)
2. Using the principle of optimality, it is easy to see that

$$d_i^{(k+1)}(j) = \min \begin{cases} d_i^{(k)}(j) \\ d_i^{(k)}(r) + c_{r,j} : (x_r, x_j) \in \mathbf{E} \end{cases} \quad (1)$$

Given $d_i^{(k)}(j)$ for all $j = 1, 2, \ldots, n$, we can compute $d_i^{(k+1)}$ in $O(m)$ time.