Discrete Applied Mathematics 158 (2010) 1676–1690

Contents lists available at ScienceDirect

Discrete Applied Mathematics

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

Matching interdiction

Rico Zenklusen *

Institute for Operations Research, ETH Zurich, Switzerland

ARTICLE INFO

Article history:
Received 30 September 2009
Received in revised form 20 May 2010
Accepted 7 June 2010
Available online 5 July 2010

Keywords:
Maximum matching
Interdiction
Complexity
Approximation algorithm
Bounded treewidth

ABSTRACT

We introduce two interdiction problems involving matchings, one dealing with edge removals and the other dealing with vertex removals. Given is an undirected graph G with positive weights on its edges. In the edge interdiction problem, every edge of G has a positive cost and the task is to remove a subset of the edges constrained to a given budget, such that the weight of a maximum matching in the resulting graph is minimized. The vertex interdiction problem is analogous to the edge interdiction problem, with the difference that vertices instead of edges are removed. Hardness results are presented for both problems under various restrictions on the weights, interdiction costs and graph classes. Furthermore, we study the approximability of the edge and vertex interdiction problem on different graph classes. Several approximation-hardness results are presented as well as two constant-factor approximations, one of them based on iterative rounding. A pseudo-polynomial algorithm for solving the edge interdiction problem on graphs with bounded treewidth is proposed which can easily be adapted to the vertex interdiction problem. The algorithm presents a general framework to apply dynamic programming for solving a large class of min–max problems in graphs with bounded treewidth. Additionally, we present a method to transform pseudo-polynomial algorithms for the edge interdiction problem into fully polynomial approximation schemes, using a scaling and rounding technique.

1. Introduction

In this paper, we introduce and study two interdiction problems for analyzing the sensitivity of the value of maximum matchings in a graph with respect to removals of edges and vertices. More precisely, given is an undirected graph $G = (V, E)$ with edge weights $w : E \to \mathbb{N}$ and a budget $B \in \mathbb{Z}_+$. We denote by $\nu(G)$ the weight of a maximum matching in G. In the first problem, which we call the *edge interdiction problem*, a cost function $c^E : E \to \mathbb{N}$ is given which associates to every edge an *interdiction cost* and the task is to find a set $R \subseteq E$ with $c^E(R) = \sum_{e\in R} c^E(e) \leq B$, such that the value of a maximum weight matching in the graph G without the edges R is minimized. Thus, an optimal removal set R achieves the minimum in

$$\min\{\nu(G - R) \mid R \subseteq E, c^E(R) \leq B\},$$

where $G - R$ is the graph obtained from G by removing the edges in R. We denote by $\nu_B^E(G)$ the value of the above minimum. In the second problem we consider, which we call the *vertex interdiction problem*, vertices instead of edges can be removed from G. More precisely, a cost function $c^V : V \to \mathbb{N}$ on the vertices is given and the task is to find a set of vertices $R \subseteq V$ with $c^V(R) = \sum_{v\in R} c^V(v) \leq B$, such that the value of a maximum weight matching in the subgraph of G that is induced by the vertices $V \setminus R$ is minimized. Thus, in the vertex interdiction problem, an optimal removal set R achieves the minimum in

$$\min\{\nu(G[V \setminus R]) \mid R \subseteq V, c^V(R) \leq B\},$$

* Fax: +41 44 632 1025.
E-mail address: rico.zenklusen@ifor.math.ethz.ch.

0166-218X/$ – see front matter © 2010 Elsevier B.V. All rights reserved.
doi:10.1016/j.dam.2010.06.006

where $G[V \setminus R]$ is the subgraph of G that is induced by the vertices $V \setminus R$. We denote by $\nu_B^V(G)$ the value of the above minimum. Throughout this paper, we use the term *maximum matching* for a matching of maximum weight.

One important motivation for studying interdiction problems is to get a measure of sensitivity for the value of an optimal solution with respect to discrete changes in the underlying graph. This is in contrast to classical sensitivity analysis, where bounded continuous changes of the input are considered. In particular, assignment problems, where some set of tasks have to be assigned in a cost-optimal way to a set of machines, are an interesting subclass of matching problems, for which a discrete sensitivity analysis is often of interest. For example, the problem of determining the worst-case influence on an assignment problem when some fixed number of machines fail, is a vertex interdiction problem. Interdiction problems were considered for a variety of other underlying combinatorial optimization problems, as for example shortest paths [1,9,11,12], network flows [20,24,27] or minimum spanning trees [8,18].

In [21,29], a problem related to matching interdiction was studied with focus on graph-theoretical aspects, namely the problem of finding a minimum d-blocker in a given graph. The task is to determine a subset of the edges of minimum cardinality such that their removal from the graph decreases the cardinality of a maximum matching by at least d units. This problem corresponds to an edge interdiction problem with unit weights and unit interdiction costs. In [4] a polynomial delay algorithm for finding all minimum 1-blockers of a bipartite graph, that contains a perfect matching, is presented. The edge interdiction problem is related to some edge deletion and edge modification problems which have been studied in [6,19,25]. Similarly, the vertex interdiction problem is related to some vertex deletion and modification problems studied in [7,17,26].

The main goal of this paper is to get a better understanding of the computational tractability of the vertex interdiction and edge interdiction problem. In particular, we want to establish hardness results and define problem classes on which it is possible to efficiently solve, respectively to efficiently approximate the problems.

We show that both problems are (weakly) *NP*-hard on graphs consisting only of isolated edges. Furthermore, we show that both problems are strongly *NP*-hard when all edge weights and interdiction costs are equal to one and present further hardness results for special cases. Furthermore, we consider two *NP*-hard versions of the edge interdiction and vertex interdiction problem and present efficient constant-factor approximations for these problem classes. A 2-approximation is presented for approximating $\nu(G) - \nu_B^V(G)$ in a graph with unit interdiction costs. Furthermore, we suggest an efficient 4-approximation based on iterative rounding for the edge interdiction problem when all edge weights are equal to one. If the input graph is bipartite, we show that the same algorithm is a 2-approximation. A pseudo-polynomial algorithm for the edge interdiction problem on graphs with bounded treewidth is presented, which can easily be adapted to the vertex interdiction problem. The proposed algorithm extends the approach that is typically used for the creation of efficient algorithms on graphs with bounded treewidth to interdiction problems. Our algorithm uses dynamic programming, which is a standard technique to solve problems on graph with bounded treewidth. However, contrary to most classical optimization problems that were studied on graphs with bounded treewidth, the interdiction problems we consider are min–max problems. The algorithm we present suggests a general framework to apply dynamic programming to efficiently solve a large class of min–max problems on graphs with bounded treewidth. Finally, we present a method to transform pseudo-polynomial algorithms for the edge interdiction problem into FPTASs using a scaling and rounding technique. In particular, this procedure can be used to transform the pseudo-polynomial algorithm we present for the edge interdiction problem on graphs with bounded treewidth into an FPTAS.

The paper is organized as follows. In Section 2, complexity results are discussed for the edge and vertex interdiction problems in various settings. In Section 3, the approximability of the edge and vertex interdiction problems is considered and two constant-factor approximation algorithms are presented. In Section 4, a pseudo-polynomial algorithm for graphs with bounded treewidth is presented and we show how to transform a pseudo-polynomial algorithm for the edge interdiction problem into an FPTAS.

2. Complexity

In this section, we present various hardness results for the edge interdiction problem and the vertex interdiction problem on different input classes. When talking about the complexity of these problems, we always consider the following decision versions. Let $G = (V, E)$ be an undirected graph with edge weight $w : E \to \mathbb{N}$, let $B \in \mathbb{Z}_+$ be a fixed budget and let $K \in \mathbb{Z}_+$. For the edge interdiction problem, we additionally have interdiction costs $c^E : E \to \mathbb{N}$ on the edges and the task is to decide whether $\nu_B^E(G) \leq K$. For the vertex interdiction problem, interdiction costs $c^V : V \to \mathbb{N}$ on the vertices are given and the task is decide whether $\nu_B^V(G) \leq K$.

Theorem 1. *The edge interdiction problem and the vertex interdiction problem are NP-complete on graphs consisting only of isolated edges.*

Proof. Both problems clearly lie in *NP*. The result is proven by a transformation from the knapsack problem, which is well known to be *NP*-complete [10]. The result is first proven for the edge interdiction problem. An easy adaption then leads to the result for the vertex interdiction problem. Consider an instance of a knapsack problem where we are given a finite set I of knapsack items, two non-negative integers B, Z and for each $i \in I$ a size $s(i) \in \mathbb{N}$ and a value $v(i) \in \mathbb{N}$. The task is to decide whether there exists some set $I' \subseteq I$ with $\sum_{i \in I'} s(i) \leq B$ and $\sum_{i \in I'} v(i) \geq Z$. Let $G = (V, E)$ be an undirected graph consisting of $|I|$ isolated edges $E = \{e_i \mid i \in I\}$. For an element $e_i \in E$, we say that the knapsack element i *corresponds* to