Packing Interdiction and Partial Covering Problems

Michael Dinitz¹ and Anupam Gupta²

¹ Weizmann Institute of Science
² Carnegie Mellon University
{mdinitz,anupamg}@cs.cmu.edu

Abstract. In the Packing Interdiction problem we are given a packing LP together with a separate interdiction cost for each LP variable and a global interdiction budget. Our goal is to harm the LP: which variables should we forbid the LP from using (subject to forbidding variables of total interdiction cost at most the budget) in order to minimize the value of the resulting LP? Interdiction problems on graphs (interdicting the maximum flow, the shortest path, the minimum spanning tree, etc.) have been considered before; here we initiate a study of interdicting packing linear programs. Zenklusen showed that matching interdiction, a special case, is NP-hard and gave a 4-approximation for unit edge weights. We obtain an constant-factor approximation to the matching interdiction problem without the unit weight assumption. This is a corollary of our main result, an $O(\log q \cdot \min\{q, \log k\})$-approximation to Packing Interdiction where $q$ is the row-sparsity of the packing LP and $k$ is the column-sparsity.

1 Introduction

In an interdiction problem we are asked to play the role of an adversary: e.g., if a player is trying to maximize some function, how can we best restrict the player in order to minimize the value attained? One of the classic examples of this is the Network Interdiction Problem (also called network inhibition), in which the player is attempting to maximize the $s$-$t$ flow in some graph $G$, and we (as the adversary) are trying to destroy part of the graph in order to minimize this maximum $s$-$t$ flow. Our ability to destroy the graph is limited by a budget constraint: each edge, along with its capacity, has a cost for destroying it, and we are only allowed to destroy edges with a total cost of at most some value $B \geq 0$ (called the budget). This interdiction problem has been widely studied due to the many applications (see e.g. [1234]). Obviously, if the cost of the minimum $s$-$t$ cut (with respect to the destruction costs) is at most $B$, then we can simply disconnect $s$ from $t$, but if this is not the case then the problem becomes NP-hard. Moreover, good approximation algorithms for this problem have been elusive. Similarly, a significant amount of work has been done on interdicting shortest paths (removing edges in order to maximize the shortest path) [56], interdicting minimum spanning trees [7], and interdicting maximum matchings [8].
Our motivation is from the problem of interdicting the maximum matching. Zenklusen [8] defined both edge and vertex versions of this problem, but we will be concerned with the edge version. In this problem, the input is a graph $G = (V, E)$, a weight function $w : E \rightarrow \mathbb{R}^+$, a cost function $c : E \rightarrow \mathbb{R}^+$, and a budget $B \in \mathbb{R}^+$. The goal is to find a set $R \subseteq E$ with cost $c(R) := \sum_{e \in R} c(e)$ at most $B$ that minimizes the weight of the maximum matching in $G \setminus R$. Zenklusen et al. [9] proved that this problem is NP-complete even when restricted to bipartite graphs with unit edge weights and unit interdiction costs. Subsequently, Zenklusen [8] gave a 4-approximation for the special case when all edge weights are unit (which is also a 2-approximation for the unit-weight bipartite graph case) and also an FPTAS for bounded treewidth graphs. These papers left open the question of giving a constant-factor approximation without the unit-weight assumption. This is a special case of the general problem we study, and indeed our algorithm resolves this question.

Maximum matching is a classic example of a packing problem. If we forget about the underlying graph and phrase the matching interdiction problem as an LP, we get the following problem: given a packing LP (i.e., an LP of the form $\max \{w^\top x \mid Ax \leq b, x \geq 0\}$, where $A, b, w$ are all non-negative), in which every column $j$ has an interdiction cost (separate from the weight $w_j$ given to the column by the objective function), find a set of columns of total cost at most $B$ that when removed minimizes the value of the resulting LP. This is the problem of Packing Interdiction, and is the focus of this paper. Interestingly, it appears to be one of the first versions of interdiction that is not directly about graphs: to the best of our knowledge, the only other is the matrix interdiction problem of Kasiviswanathan and Pan [10], in which we are given a matrix and are asked to remove columns in order to minimize the sum over rows of the largest element remaining in the row.

The Packing Interdiction problem is NP-hard, by the fact that bipartite matching interdiction is a special case due to the integrality of its standard LP relaxation, and the results of [9], and hence we consider approximation algorithms for it. Let $(k, q)$-packing interdiction, or $(k, q)$-PI for short, denote the Packing Interdiction problem in which the given non-negative matrix $A \in \mathbb{R}^{m \times n}$ has at most $k$ nonzero entries in each row and at most $q$ nonzero entries in each column. So, for example, bipartite matching interdiction is a special case of $(|V|, 2)$-PI, where $|V|$ is the number of nodes in the bipartite graph. Note that $k \leq n$ (where $n$ is the number of variables in the LP) and $q \leq m$ (where $m$ is the number of constraints). Our main result is the following.

**Theorem 1.** There is a polynomial time $O(\log q \cdot \min\{q, \log k\})$-approximation algorithm for the $(k, q)$-Packing Interdiction problem.

As a corollary, we get an $O(1)$-approximation for matching interdiction without assuming unit weights, since the natural LP relaxation has $q = 2$ and an integrality gap of 2. (See Lemma 1 for a formal proof.)

**Corollary 1.** There is a polynomial-time $O(1)$-approximation for matching interdiction.