Packing Interdiction and Partial Covering Problems

Michael Dinitz[1] and Anupam Gupta[2]

[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. [1,2,3,4]). 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) [5,6], 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 $w : E \to \mathbb{R}^+$, a cost function $c : E \to \mathbb{R}^+$, 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 $\max\{w^\intercal x \mid Ax \leq b, x \geq 0\}$, 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*

Packing Interdiction problems turn out to be closely related to the well-studied problems called *partial covering problems*; indeed, there is an algorithm for one if and only if there is an algorithm for the other (see Theorem 3). In a partial covering problem we are given a *covering* LP (i.e., a problem of the form $\min\{w^\intercal x \mid Ax \geq b, x \geq 0\}$, where A, b, w are again all non-negative, together with costs for each row (rather than for each column as in Packing Interdiction), and a budget B. We seek an vector $x \geq 0$ that minimizes the linear objective function $w^\intercal x$, subject to x being feasible for all the constraints except those in a subset of total cost at most B. In other words, rather than our (fractional) solution x being forced to satisfy all constraints as in a typical linear program, we are allowed to choose constraints with total cost at most B and violate them arbitrarily. When the matrix A defining the covering problem has at most k nonzero entries in each row and at most q nonzero entries in each column, we refer to this as the (k, q)-*partial covering* problem, or (k, q)-PC for short. We prove the following theorem about partial covering:

Theorem 2. *There is a polynomial time $O(\log k \cdot \min\{k, \log q\})$-approximation algorithm for the (k, q)-partial covering problem.*

Using the correspondence between Packing Interdiction and partial covering alluded to above, Theorem 1 follows from Theorem 2.

While many specific partial covering problems have been studied, the general partial covering problem we define above appears to be new. The closest work is by Könemann, Parekh, and Segev [11], who define the *generalized partial cover problem* to be the version in which the variables are required to be integral (i.e., even after choosing which rows to remove, they still have to solve an integer programming problem, whereas we have only a linear program which we want to solve fractionally); moreover, they consider the case where A is a $\{0, 1\}$ matrix. Their main result is a general reduction of these integer partial covering problems to certain types of algorithms for the related "prize-collecting" covering problems (where covering constraints may be violated by paying some additive penalty in the objective function). They use this reduction to prove upper bounds for many special cases of integer partial covering, such as the partial vertex cover, and partial set cover problem. Our approach to partial covering will, to a large extent, follow their framework with suitable modifications.

2 Packing Interdiction and Partial Covering

A packing LP consists of a matrix $A \in \mathbb{R}^{n \times m}$, a vector $c \in \mathbb{R}^m$, and a vector $b \in \mathbb{R}^n$, all of which have only nonnegative entries. The packing LP is defined as:

$$\max\{c^\intercal x \mid Ax \leq b, x \in \mathbb{R}^m_{\geq 0}\}$$

A packing LP is called *q-column-sparse* if every column of A has at most q nonzero entries, and *k-row-sparse* if every row of A has at most k nonzero entries. Note that $q \leq m$ and $k \leq n$. We consider the problem of Packing Interdiction (or PI), in which we are given a packing LP and are asked to play the role of an adversary that is allowed to essentially "forbid" certain variables (which corresponds to setting their c_i multiplier to 0) in an attempt to force the optimum to be small. More formally, an instance of Packing Interdiction is a 5-tuple (c, A, b, r, B) where $c \in \mathbb{R}^m, A \in \mathbb{R}^{n \times m}, b \in \mathbb{R}^n, r \in \mathbb{R}^m$, and $B \in \mathbb{R}^+$ and all entries of c, A, B and r are nonnegative. Given such an instance and a vector $z \in \{0,1\}^m$, define

$$\Phi(z, c, A, b) := \begin{cases} \max & \sum_{i=1}^{m} c_i(1 - z_i)x_i \\ s.t. & Ax \leq b \\ & x_i \geq 0 \qquad \forall i \in [m] \end{cases}$$

to be the optimum value of the packing LP when we interdict the columns with $z_i = 1$. The Packing Interdiction problem on (c, A, b, r, B) is the following integer program:

$$\begin{aligned} \min \quad & \Phi(z, c, A, b) \\ s.t. \quad & \sum_{i=1}^{m} r_i z_i \leq B \\ & z_i \in \{0,1\} \qquad \forall i \in [m] \end{aligned} \tag{MIP_{PI}}$$

Observe that while we want the z variables to be $\{0,1\}$ (to denote which variables are zero to zero), the x variables are allowed to be fractional. When the matrix A is k-row-sparse and q-column-sparse we call this problem (k,q)-Packing Interdiction (or (k,q)-PI). We say that an algorithm is an α-approximation if it always returns a solution z to (MIP_{PI}) that is within α of optimal, i.e., the vector z is feasible for (MIP_{PI}) (satisfies the budget and integrality constraints), and $\Phi(z', c, A, b) \leq \alpha \cdot \Phi(z, c, A, b)$ for any other feasible vector z'.

Our main example, and the initial motivation for this work, is (integer) *matching interdiction*. In this problem we are given a graph $G = (V, E)$, where each edge e has a weight $w_e \geq 0$ and a cost $r_e \geq 0$, and budget B. We, as the interdictor, seek a set $E' \subseteq E$ with $\sum_{e \in E'} r_e \leq B$ that minimizes the maximum weight matching in $G \setminus E'$. We can relax this to the *fractional* matching interdiction problem, where instead of interdicting the maximum weight matching we interdict the maximum weight fractional matching, defined as the optimum solution to the following LP relaxation:

$$\begin{aligned} \max \quad & \sum_{e \in E} w_e x_e \\ s.t. \quad & \sum_{e \in \partial(v)} x_e \leq 1 \ \forall v \in V \\ & x_e \geq 0 \qquad \forall e \in E \end{aligned} \tag{2.1}$$

It is easy to see that fractional matching interdiction is a special case of $(n,2)$-PI.

Lemma 1. *A ρ-approximation for fractional matching interdiction gives a 2ρ-approximation for the integer matching interdiction problem.*

Proof. Consider the optimal solution E' to integer matching interdiction, and say the weight of the max-weight matching in $G \setminus E'$ is W. It is well-known that dropping the odd-cycle constraints in the LP for non-bipartite matching results in the vertices being half-integral (and hence an integrality gap of at most 2). This means the value of the LP after interdicting E' is at most $2W$, which gives an upper bound on the optimal fractional interdiction solution. Now a ρ-approximation finds a set E'' such that the fractional solution on $G \setminus E''$ is at most $2\rho W$. Since (2.1) is a relaxation, the weight of the max-weight matching in $G \setminus E''$ is also at most $2\rho W$, giving the claimed approximation.

2.1 Partial Covering Problems

A dual problem which will play a crucial role for us when designing an algorithm for (k, q)-PI, is the problem of *Partial Covering.* Given a matrix $A \in \mathbb{R}^{m\times n}$, vectors $c \in \mathbb{R}^m$ and $b \in \mathbb{R}^n$, all having nonnegative entries, the covering LP is:

$$\min\{b^{\intercal}x \mid Ax \geq c, x \in \mathbb{R}^n_{\geq 0}\}$$

As before, we say that a covering LP is *q-column-sparse* if every column of A has at most q nonzeros and is *k-row-sparse* if every row of A has at most k nonzeros. For $j \in [m]$ and $i \in [n]$, let a_{ji} denote the entry of A in row j and column i. An instance of Partial Covering is a 5-tuple (b, A, c, r, B) in which $b \in \mathbb{R}^n, A \in \mathbb{R}^{m\times n}, c \in \mathbb{R}^m, r \in R^m, B \in \mathbb{R}^+$, and all entries of b, A, c, r are nonnegative. Given such an instance and a vector $z \in \{0, 1\}^m$, we define

$$\Psi(z, b, A, c) := \begin{cases} \min & b^{\intercal}x \\ s.t. & \sum_{i=1}^{n} a_{ji}x_i \geq c_j(1 - z_j) \ \forall j \in [m] \\ & x_i \geq 0 \qquad\qquad\qquad \forall i \in [n] \end{cases}$$

to be the value of the covering LP we get when we make the constraints j with $x_j = 1$ trivial by setting their right side to 0. Then the Partial Covering problem is the problem of computing

$$\begin{aligned} \min \quad & \Psi(z, b, A, c) \\ s.t. \quad & \sum_{i=1}^{m} r_i z_i \leq B \\ & z_i \in \{0, 1\} \qquad \forall i \in [m] \end{aligned} \tag{MIP_{PC}}$$

Analogously to Packing Interdiction, we say that an algorithm is an α-approximation to partial covering if on any instance it returns a vector z such that the value of (MIP_{PC}) is at most α times the optimal value. We let (k, q)-Partial Covering be partial covering restricted to covering LPs that are k-row-sparse and q-column-sparse.

2.2 Relating Packing Interdiction and Partial Covering

Theorem 3. *There is a polynomial time α-approximation algorithm for (q,k)-Packing Interdiction if and only if there is a polynomial time α-approximation algorithm for (k,q)-Partial Covering.*

Proof. Suppose that we have an α-approximation algorithm for (k,q)-partial covering. Let (c,A,b,r,B) be an instance of (q,k)-Packing Interdiction. Then $(b,A^{\intercal},c,r,B)$ is an instance of (k,q)-Partial Covering. Note that for a fixed $z \in \{0,1\}^m$, linear programming strong duality implies that $\Phi(z,c,A,b) = \Psi(z,b,A^{\intercal},c)$. Let $z^* \in \{0,1\}^m$ be the optimal solution to the Packing Interdiction problem, and let $\widehat{z} \in \{0,1\}^m$ be the solution to the partial covering problem computed by the algorithm. Then $\Phi(\widehat{z},c,A,b) = \Psi(\widehat{z},b,A^{\intercal},c) \le \alpha \cdot \Psi(z^*,b,A^{\intercal},c) = \alpha \cdot \Phi(z^*,c,A,b)$, where the first and last step are by strong duality and the middle inequality is by the definition of an α-approximation algorithm. Thus $\widehat{z}$ is an α-approximation to the Packing Interdiction instance. The proof of the other direction is entirely symmetric.

2.3 Partial Fractional Set Cover

A useful case of (k,q)-Partial Covering is (k,q)-*Partial Fractional Set Cover* (or (k,q)-PFSC), in which the matrix A has all entries from $\{0,1\}$, and moreover where the covering requirement $c_i = 1$ for all rows $i \in [m]$. As the name suggests, we can interpret (k,q)-PFSC in terms of set systems as follows: The universe of elements U is $[m]$, the set of rows in A. For each column $j \in [n]$ of A we have a set $S_j := \{i \in [m] \mid a_{ij} = 1\}$ corresponding to the rows which have a 1 in the j^{th} column. The k-row-sparsity of A means that every element is in at most k sets, and the q-column-sparsity means that every set contains at most q elements. Then (k,q)-PFSC corresponds to choosing a set of elements E with $\sum_{i \in E} r_i \le B$ to ignore the covering requirement for, and then constructing a minimum-cost *fractional* set cover for the remaining elements.

The version of this problem in which the x variables are forced to be integral is the *partial set cover* problem. For this problem, algorithms are known that achieve an approximation ratio of $O(\min\{k, H(q)\})$, where $H(q) = \sum_{i=1}^{q} 1/i = \Theta(\log q)$ is the harmonic number. (See, e.g., [11], which improves on [12,13,14,15].)

3 Algorithms for Partial Covering

Thanks to Theorem 3 we know that an algorithm for partial covering implies one for packing interdiction (i.e., Theorem 2 implies Theorem 1). We now prove Theorem 2 by designing an approximation algorithm for (k,q)-Partial Covering. The approach is to first reduce the general (k,q)-Partial Covering problem to the (k,q)-Partial Fractional Set Cover problem with a loss of $O(\log k)$ (i.e., with this logarithmic loss we can assume that A, c are not just non-negative, but have entries in $\{0,1\}$). We finally give an approximation for (k,q)-PFSC.

3.1 Reduction to Partial Fractional Set Cover

Let $\mathcal{I} = (b, A, c, r, B)$ be an instance of (k, q)-Partial Covering; the associated optimization problem is given by the following mixed-integer program obtained by expanding out (MIP_{PC}):

$$\begin{aligned} \min \quad & b^\intercal x \\ s.t. \quad & \sum_{i=1}^{n} a_{ji} x_i \geq c_j(1 - z_j) \; \forall j \in [m] \\ & \sum_{j=1}^{m} r_j z_j \leq B \\ & x_i \geq 0 && \forall i \in [n] \\ & z_j \in \{0,1\} && \forall j \in [m] \end{aligned} \tag{3.2}$$

We modify this mixed-integer program to be an instance of partial fractional set. If $a_{ji} \neq 0$, let $t(i,j) = \lceil \log_2(c_j/a_{ji}) \rceil$, and for each $j \in [m]$ let $S(j) = \{i \in [n] : a_{ji} \neq 0\}$. For every $i \in [n]$, we replace the variable x_i with a collection of variables $\{x_i^t\}_{t \in \mathbb{Z}}$, where the "intended" interpretation of $x_i^t = 1$ is that $x_i \geq 2^t$ as in the following mixed-integer program:

$$\begin{aligned} \min \; & \sum_{i=1}^{n} \sum_{t \in \mathbb{Z}} 2^t b_i x_i^t \\ s.t. \; & \sum_{i \in S(j)} x_i^{t(i,j)} \geq 1 - z_j \; \forall j \in [m] \\ & \sum_{j=1}^{m} r_j z_j \leq B \\ & x_i^t \geq 0 && \forall i \in [n], t \in \mathbb{Z} \\ & z_j \in \{0,1\} && \forall j \in [m] \end{aligned} \tag{3.3}$$

Although as stated there are an infinite number of variables, we only need the variables x_i^t where $t = t(i,j)$ for some j, and hence the number of x_i^t variables is at most nm. We will call this instance $\mathcal{I}' = (b', A', 1, r, B)$, where r and B are unchanged from the original instance.

Lemma 2. *$\Psi(z, b, A, c) \leq \Psi(z, b', A', 1) \leq O(\log k) \cdot \Psi(z, b, A, c)$ for any vector $z \in \{0,1\}^m$.*

Proof. To show that $\Psi(z, b, A, c) \leq \Psi(z, b', A', 1)$, take $\{x_i^t\}_{i \in [n], t \in \mathbb{Z}}$ to be a solution to (3.3) for integral vector z, and construct a solution $\{x_i\}_{i \in [n]}$ to (3.2) on the same vector z as follows: define

$$x_i := \max_{j : i \in S(j)} 2^{t(i,j)} \, x_i^{t(i,j)}.$$

Every constraint $j \in [m]$ with $z_j = 1$ is trivially satisfied, so consider some j with $z_j = 0$. For such a constraint j,

$$\sum_{i=1}^{n} a_{ji} x_i = \sum_{i \in S(j)} a_{ji} \cdot \max_{j' : i \in S(j')} 2^{t(i,j')} \, x_i^{t(i,j')} \geq \sum_{i \in S(j)} a_{ji} \, 2^{t(i,j)} \, x_i^{t(i,j)}$$

$$\geq \sum_{i \in S(j)} a_{ji}\,(c_j/a_{ji})\,x_i^{t(i,j)} = c_j \sum_{i \in S(j)} x_i^{t(i,j)} \geq c_j(1 - z_j),$$

where we use the definition of $2^{t(i,j)} \geq c_j/a_{ji}$, and that $x_i^{t(i,j)}$ is a feasible solution for (MIP_{PC}) on z. Thus $\{x_i\}_{i \in [n]}$ is a valid solution to (3.2) on z. Its cost is

$$\sum_{i=1}^{n} b_i x_i = \sum_{i=1}^{n} b_i \cdot \max_{j \in [m]} 2^{t(i,j)} x_i^{t(i,j)} \leq \sum_{i=1}^{n} b_i \sum_{t \in \mathbb{Z}} 2^t x_i^t,$$

which is exactly the value of $\{x_i^t\}_{i \in [n], t \in \mathcal{Z}}$ in (3.3) on z. Thus $\Psi(z, b, A, c) \leq \Psi(z, b', A', 1)$ for each binary vector z.

To prove that the second inequality of the lemma, consider $\{x_i\}_{i \in [n]}$, a solution to (3.2) on z, and construct a solution for (3.3) on z of cost at most $O(\log k) \cdot \sum_{i=1}^{n} b_i x_i$ as follows. For each $i \in [n]$, set $x_i^t = 1$ for all integers $t \leq \log_2 x_i$. For integers t that satisfy $\log_2 x_i < t \leq \log_2(4kx_i)$, set $x_i^t = x_i/2^t$, and for larger integers t, set $x_i^t = 0$.

Define $I_j = \{i \in [n] : a_{ji}x_i \geq c_j(1 - z_j)/2k\} \subseteq S(j)$. Since there are at most k values of i for which $a_{ji} \neq 0$, the value $\sum_{i \notin I_j} a_{ji}x_i < k \cdot c_j(1 - z_j)/2k = c_j(1 - z_j)/2$; hence $\sum_{i \in I_j} a_{ji}x_i \geq \frac{1}{2}c_j(1 - z_j)$.

If $z_j = 0$, we claim that for each $i \in I_j$, either $x_i^{t(i,j)} \in \{1, x_i/2^{t(i,j)}\}$. In other words, we need to show that $t(i,j) \leq \log_2(4kx_i)$ and hence is not set to zero. Indeed, by definition, $t(i,j) \leq \log_2(c_j/a_{ji}) + 1 = \log_2(2c_j/a_{ji})$. Moreover, by the definition of I_j, if $i \in I_j$ and $z_j = 0$ we know that $c_j/a_{ji} \leq 2k\,x_i$. Combining the two inequalities, $t(i,j) \leq \log_2(4k\,x_i)$ as claimed.

Consider some constraint $j \in [m]$ for (3.3); we will show that the solution $4x_i^t$ satisfies this constraint. If $z_j = 1$ then the constraint is trivially satisfied. Else $z_j = 0$; then for each $i \in I_j$, we have $x_i^{t(i,j)} \in \{1, x_i/2^{t(i,j)}\}$. If any of these $x_i^{t(i,j)}$ values are set to 1, the constraint j in (3.3) is satisfied by that variable alone. If not, $x_i^{t(i,j)} = x_i/2^{t(i,j)}$ for all $i \in I_j$, and

$$\begin{aligned}\sum_{i \in S(j)} x_i^{t(i,j)} &\geq \sum_{i \in I_j} x_i^{t(i,j)} = \sum_{i \in I_j} x_i/2^{t(i,j)} \geq \sum_{i \in I_j} (a_{ji}/2c_j)x_i \geq \frac{1}{2c_j}\frac{c_j}{2}(1 - z_j) \\ &= \frac{1}{4}(1 - z_j).\end{aligned}$$

Now by multiplying all of the x_i^t variables by 4 we have a valid solution to (3.3) on z. The cost of this solution is at most

$$4\sum_{i=1}^{n} b_i \sum_{t \in \mathbb{Z}} 2^t x_i^t \leq 4\sum_{i=1}^{n} b_i \left(\sum_{t \trianglelefteq \log x_i} 2^t \cdot 1 + \sum_{t = \log x_i}^{\log x_i + \log(4k)} 2^t \cdot \frac{x_i}{2^t} \right) \leq O(\log k) \sum_{i=1}^{n} b_i x_i,$$

as desired.

Lemma 3. *An α-approximation algorithm for (k, q)-Partial Fractional Set Cover gives an $O(\alpha \log k)$-approximation for (k, q)-Partial Covering.*

Proof. The above reduction to Partial Fractional Set Cover loses a factor of $O(\log k)$ in the approximation due to Lemma 6, so it remains to show that the instance $(b', A', 1, r, B)$ is in fact a (k, q)-PFSC instance, i.e., the row and column sparsities of A' are the same as in A.

For each constraint $j \in [m]$, the number of non-zeroes in the partial covering constraint of (3.3) for j equals the number of nonzeros in the partial covering constraint of (3.2) for j: both have one nonzero for each $i \in S(j)$. Thus the row sparsity of our PFSC instance is at most k. Similarly, for $i \in [n]$ and value $t \in \mathbb{Z}$, the variable x_i^t has a nonzero coefficient in (3.3) only for constraints j in which $i \in S(j)$ and $t = t(i, j)$, which is at most the number of constraints j in which $i \in S(j)$. This is the number of constraints j for which $a_{ji} \neq 0$, which is at most q, and the column sparsity of our PFSC instance is at most q, as desired.

3.2 Approximating Partial Fractional Set Cover

We now give approximation algorithms for (k, q)-Partial Fractional Set Cover. Könemann, Parekh, and Segev [11] give good algorithms for the partial set cover problem, i.e., the variant in which the x variables are also required to be integral. We adapt their framework to our setting of partial *fractional* set cover, giving the desired approximation for (k, q)-PFSC, and thus for (k, q)-Partial Cover and (q, k)-Packing Interdiction.

Prize-Collecting Covering Problems. *Prize-collecting fractional set cover* can be interpreted as the Lagrangian relaxation of partial fractional set cover, and is defined thus: given a collection of sets $\mathcal{S}$ over a universe of elements U, cost function $c : \mathcal{S} \to \mathbb{R}$, and for each element $e \in U$ there is a penalty $p(e)$, every element needs to either be covered by a set or else we pay the penalty for that element, and the goal is to minimize the total cost. We are allowed to cover an element fractionally, i.e., by fractionally buying sets which in total cover the element; however, the decision of whether to cover the set or pay the penalty for it is an integral decision. This is formalized as the following mixed integer program.

$$\begin{array}{lll}
\min \displaystyle\sum_{S \in \mathcal{S}} c(S) x_S + \sum_{e \in U} p(e) z_e & & \\
s.t. \displaystyle\sum_{S \ni e} x_S + z_e \geq 1 & \forall e \in U & \\
\quad x_S \geq 0 & \forall S \in \mathcal{S} & \\
\quad z_e \in \{0, 1\} & \forall e \in U &
\end{array} \tag{3.4}$$

In prize-collecting set cover, we change the requirements that $x_S \geq 0$ to $x_S \in \{0, 1\}$. A *ρ-Lagrangian multiplier preserving (ρ-LMP) algorithm* for prize-collecting (integral) set cover, as defined by [11], is one which on any instance I of prize-collecting (integral) set cover returns a solution with $C + \rho \cdot \Pi \leq \rho \cdot OPT(I)$, where C is the cost of the sets chosen and Π is the sum of penalties of all uncovered elements. The modification for our context is natural: an algorithm is

ρ-LMP for prize-collecting *fractional* set cover if it always returns a solution to (3.4) with $C + \rho \cdot \Pi \leq \rho \cdot OPT_{MIP}$, where as before Π is the sum of the penalties of uncovered elements (elements where $z_e = 1$), C is the total cost of the fractional covering (i.e. $\sum_{S \in \mathcal{S}} c(S)x_S$), and OPT_{MIP} is value of the optimal solution to (3.4).

Könemann et al. [11, Theorem 1] show that a ρ-LMP algorithm for prize-collecting (integral) set cover gives a $(\frac{4}{3}+\epsilon)\rho$-approximation for partial set cover, for any constant $\epsilon > 0$. Theorem 4 below generalizes this to the fractional version in a natural way, but we need an additional property: even for the fractional prize-collecting problem, the algorithm returns a solution where both x, z variables are integral. (We defer the simple proof to the full version of the paper; the crucial idea is that for any PFSC instance I, if we ignore all sets of cost more than $2OPT_{LP}$, the optimal value of the resulting PFSC instance remains at most $2OPT_{LP}$. And once every set has small cost, one can follow the earlier analysis.)

Theorem 4. *If there is a ρ-LMP algorithm for the k-row-sparse, q-column-sparse prize-collecting fractional set cover problem which returns an integral solution, then there is an $O(\rho)$-approximation algorithm for (k, q)-PFSC.*

Using this theorem, it suffices to give algorithms for the prize-collecting fractional set cover problem. Könemann et al. [11, Section 4.1] show that a natural variant of the greedy algorithm is $H(q)$-LMP for prize-collecting (integer) set cover, where $H(q)$ is the q-th harmonic number. We show that their algorithm is, in fact, $H(q)$-LMP for prize-collecting *fractional* set cover (despite returning an integral solution) by analyzing their algorithm relative to an LP rather than relative to the optimal integer solution. The algorithm works as follows: given an instance of prize-collecting partial fractional set cover $(U, \mathcal{S}, c, p)$, we create a new collection of sets $\mathcal{S}'$ where, in addition to the sets in $\mathcal{S}$, we have a singleton set $\{e\}$ for every element $e \in U$. For every set $S \in \mathcal{S}$ we set $c'(S) = c(S)$, and for each element $e \in U$ we set $c'(\{e\}) = H(q) \cdot p(e)$. We now run the greedy algorithm on this collection of sets, where we iteratively buy the set in $\mathcal{S}'$ that maximizes the number of currently uncovered elements divided by the cost c' of the set.

Lemma 4. *This greedy algorithm is $H(q)$-LMP for prize-collecting fractional set cover.*

Proof. We prove this by a standard dual-fitting argument. Relaxing the integrality constraints on the z variables of (3.4) and taking the dual, we get:

$$\begin{aligned} \max & \sum_{e \in U} y_e \\ s.t. \quad & y_e \leq p(e) \qquad \forall e \in U \\ & \sum_{e \in S} y_e \leq c(S) \; \forall S \in \mathcal{S} \\ & y_e \geq 0 \qquad \forall e \in U \end{aligned} \tag{3.5}$$

Let OPT denote the value of an optimal solution for (3.4). Let $\mathcal{S}_{gr}$ denote the sets in $\mathcal{S}$ bought by the greedy algorithm, and let $\mathcal{P}_{gr}$ denote the singleton sets it bought. Suppose at some point the greedy algorithm has covered elements in $Z \subset U$, and then picks a set A containing an element e. Then we know that $\frac{c'(A)}{|A \backslash Z|} \leq \frac{c'(B)}{|B \backslash Z|}$ for every set $B \in \mathcal{S}'$. Defining price(e) to be $\frac{c'(A)}{|A \backslash Z|}$, and recalling the definition of $c'(\cdot)$, we get

$$\sum_{e \in U} \text{price}(e) = \sum_{S \in \mathcal{S}_{gr}} c(S) + \sum_{\{e\} \in \mathcal{P}_{gr}} H(q)p(e) = \sum_{S \in \mathcal{S}_{gr}} c(S) + H(q) \sum_{\{e\} \in \mathcal{P}_{gr}} p(e).$$

To prove the greedy algorithm is $H(q)$-LMP, it suffices to show $\sum_{e \in U} \text{price}(e) \leq H(q) \cdot OPT$. Let LP denote the optimal fractional solution to (3.4) where the z variables are no longer constrained to be integral. Then $LP \leq OPT$, and by duality any solution to (3.5) is a lower bound on LP. We claim that $y_e = \text{price}(e)/H(q)$ is a valid dual solution; hence $\sum_{e \in U} \text{price}(e) = H(q) \sum_{e \in U} y_e \leq H(q) \times LP \leq H(q) \times OPT$, as required.

Finally, we show y_e is a valid solution to (3.5). Since at any point we could choose the singleton set $\{e\}$ to cover element e, we get price(e) $\leq c'(\{e\})/1 = H(q)p(e)$, and thus $y_e \leq p(e)$ for every element $e \in U$. Now let S be an arbitrary element of $\mathcal{S}$, and order the elements of $S = \{x_1, x_2, \cdots, x_{|S|}\}$ by the time that they are covered. Then since S could have been picked to cover x_i, we know that price(x_i) $\leq \frac{c'(S)}{|S|-i+1} = \frac{c(S)}{|S|-i+1}$. Thus $\sum_{e \in S} y_e = (1/H(q)) \sum_{i=1}^{n} \text{price}(x_i) \leq (1/H(q)) \cdot c(S) \cdot H(|S|) \leq c(S)$, and thus our choice of y variables form a valid dual solution.

A different approximation ratio for prize-collecting fractional set cover is in terms of k, i.e., the maximum number of sets that any element is contained in (a.k.a. its *frequency*). Könemann et al. [11, Lemma 15] showed that the primal-dual algorithm of Bar-Yehuda and Even [16] can be modified to give the following result.

Lemma 5. *There is a k-LMP algorithm for the prize-collecting fractional set cover problem.*

We can now combine these ingredients into an algorithm for (k, q)-PFSC.

Lemma 6. *There is an $O(\min\{k, H(q)\})$-approximation algorithm for (k, q)-PFSC.*

Proof. Lemmas 4 and 5 give algorithms that always return integral solutions. Thus combined with Theorem 4 they give the lemma.

3.3 Putting It Together

Having assembled all the necessary components, we can now state our main results. (Observe that all the above reductions run in polynomial time.) Combining Lemmas 3 and 6, we get.

Theorem 5. *There is a polynomial time $O(\log k \cdot \min\{k, \log q\})$-approximation algorithm for (k, q)-Partial Covering.*

Now combining Theorem 5 with Theorem 3, we get

Theorem 6. *There is a polynomial time $O(\log q \cdot \min\{q, \log k\})$-approximation algorithm for (k, q)-Packing Interdiction.*

Corollary 2. *There is a polynomial time $O(1)$-approximation algorithm for the Matching Interdiction problem.*

Proof. As mentioned in Section 2, matching interdiction is a special case of $(n, 2)$-Packing Interdiction, and using $q = 2$ in Theorem 6 gives us an $O(1)$-approximation for fractional matching interdiction. By Lemma 1, we lose another factor of 2 in going to integer matching interdiction.

References

1. Phillips, C.A.: The network inhibition problem. In: Proceedings of the Twenty-Fifth Annual ACM Symposium on Theory of Computing, STOC 1993, pp. 776–785 (1993)
2. Burch, C., Carr, R., Krumke, S., Marathe, M., Phillips, C., Sundberg, E.: A decomposition-based pseudoapproximation algorithm for network flow inhibition. In: Network Interdiction and Stochastic Integer Programming, pp. 51–68 (2003)
3. Wood, R.: Deterministic network interdiction. Mathematical and Computer Modelling 17, 1–18 (1993)
4. Zenklusen, R.: Network flow interdiction on planar graphs. Discrete Appl. Math. 158, 1441–1455 (2010)
5. Fulkerson, D.R., Harding, G.C.: Maximizing Minimum Source-Sink Path Subject To A Budget Constraint. Mathematical Programming 13, 116–118 (1977)
6. Israeli, E., Wood, R.K.: Shortest-path network interdiction. Networks 40, 2002 (2002)
7. Frederickson, G.N., Solis-Oba, R.: Increasing the weight of minimum spanning trees. In: SODA 1996, pp. 539–546 (1996)
8. Zenklusen, R.: Matching interdiction. Discrete Appl. Math. 158, 1676–1690 (2010)
9. Zenklusen, R., Ries, B., Picouleau, C., de Werra, D., Costa, M.C., Bentz, C.: Blockers and transversals. Discrete Mathematics 309, 4306–4314 (2009)
10. Kasiviswanathan, S.P., Pan, F.: Matrix Interdiction Problem. In: Lodi, A., Milano, M., Toth, P. (eds.) CPAIOR 2010. LNCS, vol. 6140, pp. 219–231. Springer, Heidelberg (2010)
11. Könemann, J., Parekh, O., Segev, D.: A unified approach to approximating partial covering problems. Algorithmica 59, 489–509 (2011)
12. Kearns, M.J.: The computational complexity of machine learning. PhD thesis, Harvard University, Cambridge, MA, USA (1990)
13. Slavík, P.: Improved performance of the greedy algorithm for partial cover. Inf. Process. Lett. 64, 251–254 (1997)
14. Bar-Yehuda, R.: Using homogeneous weights for approximating the partial cover problem. J. Algorithms 39, 137–144 (2001)
15. Fujito, T.: On approximation of the submodular set cover problem. Oper. Res. Lett. 25, 169–174 (1999)
16. Bar-Yehuda, R., Even, S.: A linear-time approximation algorithm for the weighted vertex cover problem. Journal of Algorithms 2, 198–203 (1981)