Online Primal-Dual Algorithms for Covering and Packing Problems

Niv Buchbinder and Joseph (Seffi) Naor ⋆

Computer Science Department, Technion, Haifa 32000, Israel.
{nivb,naor}@cs.technion.ac.il

Abstract. We study a wide range of online covering and packing optimization problems. In an online covering problem a linear cost function is known in advance, but the linear constraints that define the feasible solution space are given one by one in an online fashion. In an online packing problem the profit function as well as the exact packing constraints are not fully known in advance. In each round additional information about the profit function and the constraints is revealed. We provide general deterministic schemes for online *fractional* covering and packing problems. We also provide deterministic algorithms for a couple of *integral* covering and packing problems.

1 Introduction

We study a wide range of online covering and packing optimization problems. In an "offline" (fractional) covering problem the objective is to minimize the total cost given by a linear cost function $\sum_{i=1}^{n} c(i)x(i)$. The feasible solution space is defined by a set of m linear constraints of the form $\sum_{i=1}^{n} a(i,j)x(i) \geq b(j)$, where the entries $a(i,j)$ and $b(j)$ are non negative.

The *general online fractional covering problem* is an online version of the covering problem, described as a game between an algorithm and an adversary. In this setting the cost function is known in advance, but the linear constraints that define the feasible solution space are given to the algorithm one by one in an online fashion. In order to maintain a feasible solution to the current set of given constraints, the algorithm is allowed to augment the variables $x(i)$. It may not, however, decrease any previously augmented variable. We also extend our study to cases where the value of each variable has an upper bound $u(i)$, referred to as a *box constraint*. The box constraints are known to the online algorithm in advance. This captures online settings in which the amount of resources is limited. As usual, the performance of an online algorithm on a given input is defined to be the ratio between the total cost of its solution and the minimal (optimal) cost of any solution for the given instance. The maximum ratio, taken over all input sequences, is defined to be the *competitive ratio* of the algorithm. Our setting generalizes the fractional graph optimization problems discussed in [1, 2]. There, the constraints are of the of the form $\sum_{i=1}^{n} a(i,j)x(i) \geq 1$ with each

⋆ Research supported in part by US-Israel BSF Grant 2002276

coefficient $a(i, j) \in \{0, 1\}$ and no upper bounds on the values of the variables are given.

We show that the online covering problem is closely related to a dual online packing problem. In an offline packing problem we are given n packing constraints of the type $\sum_{j=1}^{m} a(j, i)y(j) \leq c(i)$. The goal is to find a feasible solution that maximizes a profit function $\sum_{i=1}^{m} y(j)$. The *general online fractional packing problem* is an online version of this problem. In the online setting the values $c(i)$ $(1 \leq i \leq n)$ are known in advance, but the profit function and the exact packing constraints are not known in advance. In the jth round a new variable $y(j)$ is introduced to the algorithm, along with its set of coefficients $a(i, j)$ $(1 \leq i \leq n)$. Note that other variables that were not yet introduced may also appear in the packing constraints. This means that each packing constraint is revealed to the algorithm gradually. The algorithm may increase the value of a variable $y(j)$ only in the round where it is given and may not decrease or increase the values of any previously given variables. The performance of the algorithm is measured with respect to the maximum possible profit of the packing problem instance.

Although this online setting seems a bit unnatural, it generalizes the well known (fractional) problems of throughput-competitive online routing of virtual circuits problem and online load balancing [4]. In the (integral) online virtual circuits routing problem the goal is to maximize the number of calls accepted. The corresponding packing constraints are given by the limited edge capacities in the network. Note that in this case all the coefficients $a(i, j)$ associated with some call j are either the bandwidth of the current call (for an edge on the route) or zero otherwise.

Online algorithms for fractional problems may in some cases yield good online algorithms for their corresponding online integral versions. Such an approach is applicable when the fractional solution can be rounded into an integral solution in an online fashion. This usually results in randomized algorithms that may sometimes be derandomized to achieve a deterministic online algorithm for the integral problem. The idea of rounding fractional solutions in an online fashion was demonstrated in [2], yielding randomized online algorithms for many interesting cases. The same idea was implicity used in [1] to achieve a deterministic algorithm for the online set-cover problem.

Previous Work: Covering and packing optimization problems, as well as their online versions, have been studied extensively. Yet, in most cases only the integral version of the problem was considered. Examples of such online integral covering optimization problems are the online steiner problem that was considered in [10] and the generalized steiner problem that was considered in [3, 5]. Recently, Alon et al. [2] suggested a general two-phase approach for a wide class of online network optimization problems. First, a fractional solution to the online problem is generated. The solution is then rounded in an online fashion in the second phase. The two phases, although separated, are run simultaneously. In [2], the first phase is performed by a general $O(\log n)$-competitive method which is applicable to a wide class of online graph optimization problems. The rounding phase, on the other hand, is problem dependent. The approach was shown useful by the development of randomized algorithms for many interesting integral graph optimization problems. Examples are the online group steiner problem and the online non-metric facility location. The same approach was also implicity used in [1], where a deterministic online $O(\log n \log m)$-competitive algorithm for the integral set cover problem was provided. Our methods for generating fractional solutions for online covering problems generalize and improve upon the methods of [2]. An integral online packing problem was considered in [4]. They considered the problem of throughput-competitive online routing of virtual circuits.

There is a long line of work on generating a near-optimal fractional solution for *offline* covering and packing problems, e.g. [14, 16, 9, 7, 11]. Generating such a solution for the *offline* covering problem with upper bounds on the variables was considered in [6, 8]. All these methods take advantage of the offline nature of the problems. Offline randomized rounding of covering and packing problems, along with a derandomization method, was considered in [13, 12]. A slightly better rounding was provided in [15].

Results: We study the *general online fractional covering* problem with and without box constraints and the *general online fractional packing* problem. We show that the problems are closely related and provide general deterministic primal-dual schemes that compute a near-optimal fractional solution for each problem. In Section 3 we suggest a scheme for the *general online fractional packing*. Let n denote the number of packing constraints. The scheme gets the desired competitive ratio $B > 0$ and produces a solution that does not violate any of the packing constraints by more than a function of $1/B$ (see theorem 1 for the exact expression). In particular, when all coefficients $a(i, j) \in \{0, 1\}$, we show an $O(\log n)$-competitive algorithm that does not violate any of the constraints. We also prove tight lower bounds on any online algorithm for the problem, proving that our scheme is optimal for any $B > 0$.

The scheme for the covering problem is very similar. In fact, when each coefficient $a(i, j) \in \{0, 1\}$, the same scheme is applicable for both problems. Unfortunately, the above scheme is not suitable for the general online fractional covering problem where the coefficients $a(i, j) \geq 0$ are not limited to $\{0, 1\}$. For this problem we design a more complicated scheme in Section 4. For any $B > 0$, our scheme is $O(\frac{\log n}{B})$-competitive and covers each constraint up to a factor of $1/B$ (i.e. $\sum_{i=1}^{n} a(i, j)x(i) \geq \frac{1}{B}$). We also extend the scheme to handle box constraints, where we are given an upper bound on the value of each variable. A simple modification of the lower bounds in [2] proves our scheme to be tight.

The online fractional covering problem with $a(i, j) \in \{0, 1\}$ was studied previously in the context of online graph optimization problems [1, 2]. The scheme we propose for this case is simpler than the algorithms of [1, 2], since we do not need to guess the value of the optimum solution, and thus avoid the need for phases. More importantly, our scheme is more precise, and thus improves the competitive ratio of the algorithm. Specifically, we get a competitive ratio of $O(\log d)$ instead of $O(\log n)$, where d is the maximum number of variables in each given constraint ($d \leq n$). This improvement immediately reflects on the competitive ratio of any of the corresponding integral problems considered in

[1, 2]. The competitive ratio for the online set-cover improves to $O(\log d \log n)$, where d is the maximum number of sets that an element appears in. In the fractional connectivity problems considered in [2], we improve the $O(\log m)$ factor, where m is the number of edges, to $O(\log C)$, where C is the size of the maximum cut in the graph. For certain integral versions of the problems considered in [2] the competitive ratio can be further improved. For instance, in the online group steiner problem we improve one of the $\log m$ factors to $\log |g|$, where $|g|$ is the maximum size of a group. Similarly, in the fractional cuts problems considered in [2] we improve the $O(\log m)$ factor to $O(\log L)$, where L is the length of the longest simple path in the graph.

In Section 5 we show the applicability of our schemes for solving their corresponding integral versions via randomized rounding. To do so, we use the schemes for the fractional cases as a "black box". We focus on converting the randomized algorithm we obtain into a deterministic algorithm. To this end, we suggest a method for rounding the fractional solution deterministically by transforming an (offline) pessimistic estimator into an online potential function. We note that, in general, the existence of an offline derandomization method does not necessarily yield an online deterministic algorithm. In particular, while all randomized rounding methods in [2] can be derandomized offline, it is an open question whether online deterministic algorithms for these problems exist. Still, the perspective of deterministic rounding motivates us to consider derandomization methods that were previously suggested for various offline problems, and these enable us to obtain better deterministic online algorithms.

We demonstrate our derandomization method on two online problems, one covering problem and one packing problem. We first consider the online unweighted set-cover problem [1] in Section 5.1. For this problem we provide a better deterministic algorithm in terms of competitive ratio. We draw ideas from the improved offline randomized rounding and derandomization methods of [15] to come up with an improved potential function for the online problem. This yields a competitive ratio of $O(\log d \log \frac{n}{OPT})$ instead of $O(\log d \log n)$, where OPT is the optimal number of sets needed to cover the requested elements. The improvement in the competitive ratio is significant when the number of sets needed for the cover is large, and in particular when the sets are small. Due to space limitations we omit all proofs.

Finally, in Section 5.2 we consider the problem of throughput-competitive online routing of virtual circuits [4]. We show that a deterministic algorithm equivalent to the one from [4] can be derived easily by combining our schemes with the method of conditional expectations of the randomized rounding algorithm presented in [13]. This is an interesting way of viewing the algorithm of [4], which also provides a systematic method for deriving it. In particular, we note that our scheme produces in an online fashion a near-optimal *fractional solution* to the problem that does not violate the capacity constraints. Producing such a fractional solution, unlike an integral solution, is applicable to any values of edge capacities. We also note that our scheme may be used to deal with a more general setting of the problem. See section 5.2 for more details.

Primal (Covering)		Dual (Packing)	
Minimize:	$\sum_{i=1}^{n} c(i)x(i)$	Maximize:	$\sum_{j=1}^{m} y(j)$
Subject to:		Subject to:	
For each $1 \leq j \leq m$:	$\sum_{i=1}^{n} a(i,j)x(i) \geq 1$	For each $1 \leq i \leq n$:	$\sum_{j=1}^{m} a(i,j)y(j) \leq c(i)$
For each $1 \leq i \leq n$:	$x(i) \geq 0$	For each $1 \leq j \leq m$:	$y(j) \geq 0$

Fig. 1. Primal (covering) and dual (packing) problems

2 Preliminaries

In this section we formally define our problems and discuss their dual nature. In an "offline" (fractional) covering problem the objective is to minimize the total cost given by a linear cost function $\sum_{i=1}^{n} c(i)x(i)$. The feasible solution space is defined by a set of m linear constraints of the form $\sum_{i=1}^{n} a(i,j)x(i) \geq b(j)$, where the entries $a(i,j)$ and $b(j)$ are non negative. Given an instance of a covering problem we may first normalize each constraint to the form: $\sum_{i=1}^{n} a(i,j)x(i) \geq 1$. By the duality theorem, any primal covering instance has a corresponding dual packing problem that provides a lower bound on any feasible solution to the instance. A general form of a (normalized) primal covering problem along with its dual packing problem is given in Figure 1.

The *general online fractional covering problem* is an online version of the covering problem. In this setting the cost function is known in advance, but the linear constraints that define the feasible solution space are given to the algorithm one-by-one. In order to maintain a feasible solution to the current set of given constraints, the algorithm is allowed to augment the variables $x(i)$. It may not, however, decrease any previously augmented variable. We also define an online version of the packing problem. In the *general online fractional packing problem* the values $c(i)$ $(1 \leq i \leq n)$ are known in advance. However, the profit function and the exact packing constraints are not known in advance. In the jth round a new variable $y(j)$ is introduced to the algorithm, along with its set of coefficients $a(i,j)$ $(1 \leq i \leq n)$. Note that other variables that were not yet introduced may also appear in the packing constraints. This means that each packing constraint is revealed to the algorithm gradually. The algorithm may increase the value of a variable $y(j)$ only in the round where it is given, and may not decrease or increase the values of any previously given variables.

We observe that these two online settings form a primal-dual pair in the following sense: At any time an algorithm for the general online fractional covering problem maintains a subset of the final linear constraints. This subset defines a *sub-instance* of a final covering instance. The dual packing problem of this sub-instance is a *sub-instance* of the final dual packing problem. In the dual packing sub-instance only part of the dual variables are known along with their corresponding coefficients. The two sub-instances form a primal-dual pair. In each round of the general online fractional covering problem a new constraint on the feasible solution space is given. The primal covering sub-instance is updated by adding the new constraint. To update the dual sub-instance we add a new dual variable to the profit function along with its coefficients that are defined by the new primal constraint. Note that the dual update is the same as in the setting of the online fractional packing problem.

The schemes we propose maintain at each step primal and dual solutions for these primal-dual sub-instances. When a new constraint is given to the scheme for the online fractional covering problem, it also considers the new *corresponding* dual variable and its coefficients. When the scheme for the online fractional packing problem receives a new variable along with its coefficients, it also considers the *corresponding* new constraint in the primal sub-instance. In the rest of the paper we use the notion of primal and dual sub-instances and the notion of corresponding dual variable and primal constraint.

3 The General Online Fractional Packing Problem

In this section we describe an online scheme for computing a near-optimal fractional solution for the general online fractional packing problem. The scheme gets the desired competitive ratio $B > 0$ and returns a solution within B of the optimal that does not violate the packing constraints by too much. The scheme simultaneously maintains primal (covering) and dual (packing) solutions for the primal and dual sub-instances.

Initially, each variable $x(i)$ is initialized to zero. In each round a new variable $y(j)$ is introduced along with its coefficients $a(i,j)$ $(1 \leq i \leq n)$. In the corresponding primal sub-instance a new constraint is introduced of the form $\sum_{i=1}^{n} a(i,j)x(i) \geq 1$. This constraint is non empty, since otherwise, it means that there is no bound on the value of $y(j)$ and the profit function is unbounded. The algorithm increases the value of the new variable $y(j)$ and the values of the primal variables $x(i)$ until the new primal constraint is satisfied. The augmentation method is described here in a continuous fashion, but it is not hard to implement the augmentation in a discrete way in any desired accuracy. In our continuous description the variables $x(i)$ behave according to a monotonically increasing function of $y(j)$. To implement the scheme in a discrete fashion, one should find the minimal $y(j)$ such that the new primal constraint is satisfied. Note that any variable $y(j)$ is being increased only in the jth round and the values of the primal variables never decrease. A single round is described by the following scheme. The performance of the scheme is analyzed in theorem 1.

1. $y(j) \leftarrow 0$; For each $x(i)$: $a_i(\max) \leftarrow \max_{k=1}^{j}\{a(i,k)\}$.
2. While $\sum_{i=1}^{n} a(i,j)x(i) < 1$:
   (a) Increase $y(j)$ continuously.
   (b) Increase each variable $x(i)$ by the following increment function:
$$x(i) \leftarrow \max\left\{x(i), \frac{1}{na_i(\max)}\left[\exp\left(\frac{B}{2c(i)}\sum_{k=1}^{j} a(i,k)y(k)\right) - 1\right]\right\}$$

Theorem 1. *For any $B > 0$, the scheme is $B-$competitive algorithm for the general online fractional packing problem and for any constraint:*
$$\sum_{k=1}^{m} a(i,k)y(k) = c(i) \cdot O\left(\frac{\log n + \log \frac{a_i(\max)}{a_i(\min)}}{B}\right)$$

Where, $a_i(\max) = \max_{k=1}^{m}\{a(i,k)\}$ *and* $a_i(\min) = \min_{k=1}^{m}\{a(i,k)|a(i,j) \neq 0\}$.

Corollary 1. *When* $a(i,j) \in \{0,1\}$ *there exists an* $O(\log n)$*-competitive algorithm for the fractional packing problem that does not violate the constraints.*

Remark 1. When all entries $a(i,j) \in \{0,1\}$, we do not need to update the value $a_i(\max)$ in line 2(b) and the scheme is simplified. In this case, if we know in advance that each primal constraint consists of at most ℓ non-zero coefficients, it is possible to improve the competitive factor from $\log n$ to $\log \ell$. This is done by replacing the value n in line 2(b) by ℓ. Moreover, in this case, the same scheme is applicable to the online fractional covering problem (with $a(i,j) \in \{0,1\}$). The improvement of the competitive ratio to $\log \ell$ immediately reflects on the competitiveness of several corresponding integral covering problems. See Results Section for more details.

Remark 2. The algorithm can handle an unbounded number of dual variables in each iteration, if there exists an oracle that finds at each iteration a variable whose primal constraint is not satisfied, or states that there is no such variable.

3.1 Lower Bounds

In this section we state two lower bounds that show that our scheme is optimal up to constants.

Lemma 1. *There is an instance of the general fractional packing problem with a single constraint such that* $\sum_{j=1}^{m} a(i,j)y(j) \geq c\frac{H(a(\max)/a(\min))}{B}$ *for any online* $B-$*competitive algorithm.* $H(n)$ *is the nth harmonic number, and* $a(\max)/a(\min)$ *is the ratio between the maximal and the minimal value in the constraint.*

Lemma 2. *There is an instance of the general online fractional packing problem with* n *constraints and* $a(i,j) \in \{0,1\}$*, such that for any* $B-$*competitive online algorithm there exists a constraint such that* $\sum_{j=1}^{m} a(i,j)y(j) \geq c(i)\frac{\log n}{2B}$.

4 The General Online Fractional Covering Problem

In this section we describe our online scheme for computing a near-optimal fractional solution for the online fractional covering problem. As stated in Remark 1, in case the coefficients $a(i,j) \in \{0,1\}$ the scheme in section 3 is also applicable for the online fractional covering problem. Unfortunately, the scheme is not applicable to the general online fractional covering problem, where the coefficients $a(i,j) \geq 0$ are not limited to $\{0,1\}$. This happens since the scheme described in Section 3 does not always produce a feasible dual solution that can bound the primal solution efficiently. In this section we design a more complicated scheme for the general online fractional covering problem.

Our scheme for the general online fractional covering problem gets a parameter $B > 0$. With $B > 0$ the competitive ratio of the scheme is $O(\frac{\log n}{B})$ and for each constraint $\sum_{i=1}^{n} a(i,j)x(i) \geq \frac{1}{B}$. The scheme works in phases:

Primal (Covering)		Dual (Packing)	
Minimize:	$\sum_{i=1}^{n} c(i)x(i)$	Maximize:	$\sum_{j=1}^{m} y(j) - \sum_{i=1}^{n} u(i)z(i)$
Subject to:		Subject to:	
$\forall j : 1 \leq j \leq m$:	$\sum_{i=1}^{n} a(i,j)x(i) \geq 1$	$\forall i : 1 \leq i \leq n$:	$\sum_{j=1}^{m} a(i,j)y(j) - z(i) \leq c(i)$
$\forall i : 1 \leq i \leq n$:	$0 \leq x(i) \leq u(i)$	$\forall i, j$:	$y(j), z(i) \geq 0$

Fig. 2. Primal (covering) with box constraints and its dual (packing) problem.

When the first constraint is introduced the scheme generates a first lower bound $\alpha(1) \leftarrow \frac{1}{B} \min_{i=1}^{n}\{c(i)/a(i,1)\} \leq \frac{OPT}{B}$. It runs with the lower bound $\alpha(r)$ on the optimum as long as the total primal cost does not exceed the value $\alpha(r)$. When the cost exceeds this value the scheme "forgets" about all the values given to the primal and dual variables so far, updates the value of α by doubling it, and starts a new phase with $\alpha(r+1) \leftarrow 2\alpha(r)$. Nevertheless, the values of the "forgotten" variables are accounted in the total cost of the solution. In the following we describe one round of our scheme in the rth phase. Let $\sum_{i=1}^{n} a(i,j)x(i) \geq 1$ be the new primal constraint that was introduced. Let $y(j)$ be the new corresponding dual variable. The scheme increases the values of the primal and dual variables as described by the following scheme. Note that during each phase $x(i)$ only increases. The performance of the scheme is analyzed in theorem 2.

1. $y(j) \leftarrow 0$
2. While $\sum_{i=1}^{n} a(i,j)x(i) < \frac{1}{B}$:
   (a) increase $y(j)$ continuously.
   (b) Increase each variable $x(i)$ by the following increment function:
$$x(i) \leftarrow \frac{\alpha(r)}{2nc(i)} \exp\left(\frac{\log 2n}{c(i)} \sum_{k=1}^{j} a(i,k)y(k)\right)$$

Theorem 2. *For any $B > 0$, the scheme for the general online fractional covering problem achieves a competitive ratio of $O(\frac{\log n}{B})$, such that for each constraint $\sum_{i=1}^{n} a(i,j)x(i) \geq \frac{1}{B}$.*

Adding Box Constraints: We now extend our methods to handle upper bounds on the variables $x(i)$. Let $u(i)$ be the upper bound on the variable $x(i)$. Such an upper bounds may result in an instance of the covering problem with no feasible solution. We next sketch the main ideas and changes that are needed in order to deal with the upper bounds. Adding box constraints to a covering problem results in new negative variables $z(i)$ in the dual program. The primal covering with box constraints and the new corresponding dual program are described in Figure 2. The performance of the scheme is analyzed in theorem 3.

Theorem 3. *For any $B > 0$, the scheme for the general online fractional covering problem with box constraints achieves a competitive ratio of $O(\frac{\log n}{B})$, such that for each constraint $\sum_{i=1}^{n} a(i,j)x(i) \geq \frac{1}{B}$ and for each variable $x(i) \leq \frac{u(i)}{B}$*

Our proposed scheme works in phases, where each phase has an upper bound on the total cost. When some variable reaches a value of $\frac{u(i)}{B}$, we start augmenting its corresponding dual variable $z(i)$. The augmentation of $z(i)$ causes the increment of the primal variable $x(i)$ to stop. Let X be the set of *tight variables* with value $\frac{u(i)}{B}$. If all variables in some unsatisfied constraint are tight, the scheme returns - "No feasible solution". A single round during the rth phase is described in the following:

1. $y(j) \leftarrow 0$
2. While $\sum_{i=1}^{n} a(i,j)x(i) < \frac{1}{B}$:
   (a) Increase $y(j)$ continuously.
   (b) For each variable $x(i) \in X$ increase $z(i)$ with rate $a(i,j)y(j)$.
   (c) Augment each variable $x(i)$ by the following increment function:

$$x(i) \leftarrow \min\{\frac{u(i)}{B}, \frac{\alpha(r)}{2nc(i)}\} \exp\left(\frac{\log 2n}{c(i)} \sum_{k=1}^{j} a(i,k)y(k) - z(i)\right)$$

5 Integral Online Problems: Derandomization

In this section we show the applicability of our fractional schemes for solving online integral problems. To do so, we use the schemes as "black boxes" and deterministically round the fractional solutions obtained. The deterministic rounding is obtained by transforming an (offline) pessimistic estimator into an online potential function. This was done implicity in [1]. The existence of an offline derandomization method does not necessarily yield an online deterministic algorithm. For example, finding deterministic algorithms for the problems considered in [2] is still an open problem. (They all have offline derandomizations.) Thus, we note that the ability to transform a (offline) pessimistic estimator into an online potential function is quite surprising.

We demonstrate our derandomization method on two online problems. We first consider the online unweighted set-cover problem [1]. For this problem we provide a better deterministic algorithm in terms of competitive ratio. To do so, we draw on ideas from the improved offline rounding and derandomization methods of [15] to generate an improved potential function. We then consider the online problem of throughput-competitive routing of virtual circuits. We show that the algorithm presented in [4] can be derived by a deterministic rounding of the fractional solution produced by our packing scheme.

This approach provides us with the following insight to the competitive factors obtained by [4]. The $O(\log m)$ competitive factor follows from an online generation of a fractional solution. The minimum edge capacity determines the scaling of the fractional solution that is needed in order to guarantee high probability of success in the rounding phase. We note that producing a near-optimal fractional solution that does not violate the capacity constraints is applicable with any values of edge capacities.

5.1 Improved Competitive Ratio for Online Unweighted Set Cover

We define the online unweighted set cover problem as follows. Let $X = \{e_1, \ldots, e_n\}$ be a ground set of n elements, and let $\mathcal{S}$ be a family of subsets of X, $|\mathcal{S}| = m$. A *cover* is a collection of sets such that their union is X. In an online setting the elements are given one-by-one. Once a new element is given, the algorithm has to cover it by some set of $\mathcal{S}$ containing it. Denote by $X' \subseteq X$ the set of elements given by the adversary. Our assumption is that the set cover instance, i.e., X and $\mathcal{S}$, is known in advance. The objective is to minimize the total number of sets chosen by the algorithm. Let $\mathcal{C}$ denote the family of sets in $\mathcal{S}$ that the algorithm chooses, and let C denote the set of elements covered by sets belonging to $\mathcal{C}$.

Note, first, that we may assume, by doubling, that the cardinality of OPT is known up to a factor of 2. Indeed, we can start guessing OPT =1, and run the algorithm with this value of the optimal solution. If it turns out that the cardinality of the optimal solution is already at least twice our current guess for it, (that is, the cost of $\mathcal{C}$ exceeds $\Theta(OPT \log d \log(n/OPT)))$, then we can "forget" all sets chosen so far to $\mathcal{C}$, update the value of OPT by doubling it, and keep going. It can be easily verified that this only multiply the competitive ratio by constants. Next, we transform the pessimistic estimator that appear in [15] into the following online potential function used by our algorithm:

For each element e_i $(1 \leq i \leq n)$: $f(e_i) = \min\left\{1, \exp\left(-\alpha + \alpha \sum_{s|e_i \in s} w(s)\right)\right\}$

$$\Phi = \left[1 - \prod_{e_i|e_i \notin C} (1 - f(e_i))\right] + \exp\left(\sum_{s \in \mathcal{S}} (\ln 2\chi_{\mathcal{C}}(s) - \alpha w(s)) - OPT\right)$$

The function $\chi_{\mathcal{C}}(s) = 1$ if $s \in \mathcal{C}$, and $\chi_{\mathcal{C}}(s) = 0$ otherwise. The parameter $\alpha = O(\log(n/OPT)$ will determine our competitive ratio. The first term of the potential function ensures that each element that was given to the algorithm is covered. The second term is used to bound the cardinality of the solution.

Using the above potential function the online algorithm is simple: Run the algorithm presented in Section 3 to produce a fractional solution. Note that since all coefficients $a(i,j) \in \{0,1\}$, the simpler scheme in section 3 is applicable. When the weight of some set is augmented, add the set to the cover $\mathcal{C}$, only if by adding it the potential function decreases. We prove claim 5.1 on the properties of the potential function and then prove our main lemma (lemma 3).

Claim. The potential function Φ satisfies the following properties:

1. At start $\Phi < 1$, and at any time during the run of the algorithm $\Phi > 0$.
2. When a set is augmented either taking it to $\mathcal{C}$ or excluding it does not increase Φ.

Lemma 3. *The online algorithm satisfies the following:*

1. *Each element that was given to the algorithm is covered.*
2. *The cardinality of* $\mathcal{C}$ *is at most* $OPT \cdot O(\log d \log(n/OPT))$.

5.2 Throughput-Competitive Online Routing of Virtual Circuits

The online problem of maximizing the throughput of scheduled virtual circuits was considered in [4]. In its simplest version we are given in advance a graph with capacities $u(e)$ on the edges. A set of clients $c_i = (s_i, t_i)$ $(1 \leq i \leq n)$ arrive in an

Minimize: $\sum_{e \in E} u(e)Y(e) + \sum_{c_i} U(c_i)$	Maximize: $\sum_{c_i} \sum_{P \in P(c_i)} f(c_i, P)$
Subject to:	Subject to:
$\forall P \in P(c_i)$: $\sum_{e \in P} Y(e) + U(c_i) \geq 1$	$\forall$ client c_i: $\sum_{P \in P(c_i)} f(c_i, P) \leq 1$
	$\forall$ edge e: $\sum_{P \mid e \in P} \sum_{c_i} f(c_i, P) \leq u(e)$

Fig. 3. The routing problem (Maximize) and its corresponding primal problem.

online fashion. To serve a client, the algorithm chooses a path between s_i and t_i and allocates a bandwidth of 1 on this path. The total bandwidth allocated on any edge may never exceed its capacity. The total profit of the algorithm is the number of clients served and its performance is measured with respect to the maximum number of clients that could have received service.

In a fractional version of the problem the allocation is not restricted to integral bandwidth of either zero or one, instead we can allocate to each client a fractional bandwidth in the range $[0, 1]$. In addition, the bandwidth allocated to a client can be divided between several paths. This is an online version of maximum multicommodity flow. We describe the problem as a packing problem in Figure 3. For $c_i = (s_i, t_i)$ let $P(c_i)$ be the set of simple paths between s_i and t_i. The first set of constraints ensures that each client gets at most a fractional flow (bandwidth) of 1. The second set of constraints are the capacity constraints of the edges. In the primal problem we assign a variable $U(c_i)$ to each client c_i and a variable $Y(e)$ to each edge in the graph. The problem is a special case of the general online fractional packing problem with $a(i, j) \in \{0, 1\}$. We note that our methods can generate a fractional solution to an extension of the problem to any non-negative coefficients $a(i, j)$. The extension can be viewed as giving each edge a different capacity with respect to the clients that are using it. This models a more complex relationship between the clients and the network capacities.

An algorithm for the problem was proposed in [4]. We provide here an alternative equivalent algorithm. We build our algorithm systematically by using our two phase approach. First, we use our scheme to generate a *feasible* fractional solution which is $O(\log P(\max))$ far from the optimum, where $P(\max)$ is the length of the longest path in the graph. Next, we convert the standard pessimistic estimator designed for the offline version of the problem into an online potential function. A randomized method to achieve a feasible integral solution is to scale down all the flows by some factor $B \geq 1$. We then choose to route the flow on a path P, carrying fractional flow of $f(P)$, with probability $f(P)/B$. When B is large enough, there is a positive probability that no edge capacity is violated, and the total integral flow is large enough [13]. This algorithm was derandomized using the method of conditional expectations via a pessimistic estimator [12]. We transform this pessimistic estimator into an online potential function. When the flow on a path to a client is increased, the algorithm serves the client on that path if by doing so the potential function does not increase. Our online potential function Φ is the sum of the following functions:

$$\Phi_1 = \frac{1}{2} \exp \left\{ \ln 2 \left[\sum_{c_i} \sum_{P \in P(c_i)} \frac{f(c_i, P)}{B} - 2 \sum_{c_i} \chi(c_i) \right] \right\}$$

$$\Phi_2 = \frac{1}{2n}\sum_{e\in E}\exp\left\{\left(1+\frac{1+\ln n}{u(e)}\right)\chi(e) - \sum_{P|e\in P}\sum_{c_i} f(c_i,P)\right\}$$

Where, $\chi(c_i)$ is the characteristic function of c_i (i.e. $\chi(c_i) = 1$ iff it is serviced), and $\chi(e)$ is the total number of paths that use the edge. Choosing $B = \exp\{(1+\frac{1+\ln n}{u(\min)})\} - 1$, where $u(\min)$ is the minimal capacity of an edge, suffices to prove claim 5.2 that is used to prove lemma 4.

Claim. Initially, $\Phi < 1$ and throughout the algorithm, $\Phi > 0$. In addition, each time a flow on some path is increased, then either serving the client on this path or doing nothing reduces the potential function.

Lemma 4. *The algorithm does not violate the capacity constraints and accepts a least* $O(\frac{1}{B\log P(\max)})OPT$ *clients, where* $B = \exp\{(1+\frac{1+\ln n}{u(\min)})\} - 1$

Acknowledgements: We thank Yossi Azar for many helpful discussions.

References

1. N. Alon, B. Awerbuch, Y. Azar, N. Buchbinder, and J. Naor. The online set cover problem. In *the 35th annual ACM Symp. on the Theory of Computation*, 2003.
2. N. Alon, B. Awerbuch, Y. Azar, N. Buchbinder, and J. Naor. A general approach to online network optimization problems. In *SODA*, 2004.
3. B. Awerbuch, Y. Azar, and Y. Bartal. On-line generalized steiner problem. In *Proc. of the 7th Ann. ACM-SIAM Symp. on Discrete Algorithms*, 1996.
4. B. Awerbuch, Y. Azar, and S. Plotkin. Throughput-competitive online routing. In *Proc. of 34th FOCS*, pages 32–40, 1993.
5. P. Berman and C. Coulston. On-line algorithms for steiner tree problems. In *Proc. of the 29th annual ACM Symp. on the Theory of Computation*, 1997.
6. L. Fleischer. A fast approximation scheme for fractional covering problems with variable upper bounds. In *SODA 2004*, pages 1001–1010.
7. L. K. Fleischer. Approximating fractional multicommodity flow independent of the number of commodities. *SIAM Journal on Discrete Mathematics*, 2000.
8. N. Garg and R. Khandekar. Fractional covering with upper bounds on the variables: Solving lps with negative entries. In *ESA 2004*, pages 371–382.
9. N. Garg and J. Konemann. Faster and simpler algorithms for multicommodity flow and other fractional packing problems. In FOCS 1998, pages 300–309.
10. M. Imase and B. Waxman. Dynamic steiner tree problem. *SIAM Journal Discrete Math.*, 4:369–384, 1991.
11. S. G. Kolliopoulos and N. E. Young. Tight approximation results for general covering integer programs. In *FOCS 2001*, pages 522–528.
12. P. Raghavan. Probabilistic construction of deterministic algorithms: approximating packing integer programs. *J. Comput. Syst. Sci.*, 37(2):130–143, 1988.
13. P. Raghavan and C. Thompson. Randomized rounding: A technique for provably good algorithms and algorithmic proofs. *Combinatorica*, 7:365–374, 1987.
14. E. T. S. Plotkin and D. Shmoys. Fast approximation algorithms for fractional packing and covering problems. *Math. of Operations Research*, 20:257–301, 1995.
15. A. Srinivasan. Improved approximation guarantees for packing and covering integer programs. *SIAM Journal on Computing*, 29(2):648–670, 1999.
16. N. E. Young. Randomized rounding without solving the linear program. In *Proc. of the 6th annual ACM-SIAM symp. on Discrete algorithms*, pages 170–178, 1995.