The Geometry of Online Packing Linear Programs

Marco Molinaro  R. Ravi

Abstract

We consider packing linear programs with $m$ rows where all constraint coefficients are in the unit interval. In the online model, we know the total number $n$ of columns that arrive in random order. The goal is to set the decision variables corresponding to the arriving columns irrevocably so as to maximize the expected reward. Previous $(1 - \epsilon)$-competitive algorithms require that the right-hand sides of the constraints are of magnitude at least $\Omega(\frac{m}{\epsilon^2 \log \frac{n}{\epsilon}})$, a bound that worsens with the number of columns and rows. However, the dependence on the number of columns is not required in the single-row case of online secretary problems. Moreover, known lower bounds for the general case of $m$ rows only demonstrate that the right-hand sides must be as large as $\Omega(\frac{\log m}{\epsilon^2})$, with no dependence on $n$ to obtain $(1 - \epsilon)$-competitive algorithms.

Our goal is to understand whether the dependence on $n$ is required in the multi-row case, making it fundamentally harder than the single-row version. We show that this is not the case by exhibiting an algorithm which is $(1 - \epsilon)$-competitive as long as the right-hand sides are $\Omega(\frac{m^2}{\epsilon^2 \log \frac{n}{\epsilon}})$. Our techniques refine previous PAC-learning based approaches, which interpret the online decisions as linear classifications of the columns based on dual prices obtained from sampled columns. Our improved bounds are proved by constructing a small set of witnesses for misclassifications, which are then used to obtain improved generalization bounds for the learning algorithm. The key component of our improvement is recognizing why the single-row problem is seemingly easier: if the columns of the LP belong to few one-dimensional subspaces, there is high overlap among the misclassifications and hence the associated learning problem is intrinsically more robust. For general linear programs, the idea is to modify the input to make the columns lie in a few one-dimensional subspaces while not changing the feasible set by much.
1 Introduction

Traditional optimization models usually assume that the input is known a priori. However, in most applications, the data is either revealed over time or only coarse information about the input is known, often modeled in terms of a probability distribution. Consequently, much effort has been directed towards understanding the quality of solutions that can be obtained without full knowledge of the input, which led to the development of online and stochastic optimization [7, 6]. Emerging problems such as allocating advertisement slots to advertisers and yield management in the internet are of inherent online nature and have further accelerated this development [1].

Linear programming is arguably the most important and thus well-studied optimization problem. Therefore, understanding the limitations of solving linear programs when complete data is not available is a fundamental theoretical problem with a slew of applications, including the ad allocation and yield management problems above. Indeed, a simple linear program with one uniform knapsack, the Secretary Problem, was one of the first online problems to be considered and an optimal solution was already obtained by the early 60’s [13, 15]. Although the single knapsack case is currently well-understood under different models of how information is revealed [4], much less is known about problems with multiple knapsacks. Only recently, algorithms with guaranteed solution quality have been developed for these more general packing problems [14, 1, 10].

The Model. We consider the following online packing LP problem. Consider a fixed but unknown LP with $n$ columns $a^t \in [0,1]^m$ (whose associated variables are constrained to be in $[0,1]$) and $m$ packing constraints:

$$\begin{align*}
\text{OPT} &= \max \sum_{t=1}^{n} \pi_t x_t \\
\sum_{t=1}^{n} a^t x_t &\leq B \\
x_t &\in [0,1].
\end{align*}$$

Columns are presented in a random (uniform) order, and whenever a column is presented we are required to irrevocably choose the value of its corresponding variable. We assume that the number of columns $n$ is known.\footnote{Actually knowing $n$ up to $(1 \pm \epsilon)$ factor is enough. This assumption is required to allow algorithms with non-trivial competitive ratio [11].}

The goal is to obtain a feasible solution to the LP while maximizing its value. Note that we use OPT to denote the optimum value of the (offline) LP.

By scaling down rows as necessary, we assume without loss of generality that all entries of $B$ are the same, which we also denote with some overload of notation by $B$. Due to the packing nature of the problem, we also assume without loss of generality that all the $\pi_t$’s are non-negative and all the $a^t$’s are non-zero: we can simply ignore columns which do not satisfy the first property and always set to 1 the variables associated to the remaining columns which do not satisfy the second property. Finally, we assume that the columns $a^t$’s are in general position: for all $p \in \mathbb{R}^m$, there are at most $m$ different $t \in [n]$ such that $\pi_t = pa^t$. Notice that perturbing the input randomly by a tiny amount achieves this property with probability one, while the effect of the perturbation is absorbed in our approximation guarantees [11, 1].

The random permutation model where the input is presented in a random order has grown in popularity [16, 11, 4], since it avoids strong lower bounds of the pessimistic adversarial-order model [8], while still capturing the lack of total information a priori. Moreover, the random permutation model is weaker than the i.i.d. model that assumes that the parts constituting the input are sampled independently from a fixed distribution, which is either known or unknown.

Related work. Many different types of online problems have already been studied in the random permutation model. These include bin-packing [20], matchings [19, 16], the AdWords Problem [11] and different generalizations of the Secretary Problem [4, 2, 5, 25, 18]. Closest to our work are packing problems with a single knapsack model.
constraint. In [21], Kleinberg considered the $B$-Choice Secretary Problem, where the goal is to select at most $B$ items coming online in random order to maximize profit. The author presented an algorithm with competitive ratio $1 - O(1/\sqrt{B})$ and showed that $1 - \Omega(1/\sqrt{B})$ is best possible. Generalizing the $B$-Choice Secretary Problem, Babaioff et al. [3] considered the online knapsack problem and presented a $(1/10e)$-competitive algorithm. Notice that in both cases the competitive ratio does not depend on $n$.

Despite all these works, the first result for the more general online packing LP under study here was only recently obtained by Feldman et al. [14]. They gave an algorithm that obtains with high probability a solution of value at least $(1 - \epsilon)OPT$ whenever $B \geq \Omega(m \log n / \epsilon^2)$ and $OPT \geq \Omega\left(\frac{\pi_{\text{max}}^2 m \log n}{\epsilon^2}\right)$, where $\pi_{\text{max}}$ is the largest profit. The authors actually considered a more general allocation problem, where a set of columns representing various options arrive at each step, and the solution may choose at most one of the options. Their algorithm is training-based and generalizes the work of Devanur and Hayes [11] on the AdWords problem.

In an as yet unpublished manuscript, Agrawal et al. [1] presented an algorithm (DPA) which managed to further reduce the required dependence on the size of $B$ and $OPT$. Their algorithm returns a solution with expected value at least $(1 - \epsilon)OPT$ whenever $B \geq \Omega\left(\frac{m \log n}{\epsilon^2}\right)$ or $OPT \geq \Omega\left(\frac{\pi_{\text{max}}^2 m^2 \log n}{\epsilon^2}\right)$. Another way of stating this result is that the algorithm obtains a solution with competitive ratio $1 - O\left(\sqrt{\frac{m \log(n) \log B}{B}}\right)$; notice that the guarantee degrades as $n$ increases. Their algorithm also uses training-based ideas, but now re-training as the sample size doubles to obtain the improved guarantees. They also show that there are instances with $B \leq \frac{\log m}{\epsilon^2}$ for which no online algorithm can be $(1 - \epsilon)$-competitive in the random permutation model.

The above works on the problem under study draw a connection between solving the online LP and PAC-learning [9] a linear classification of its columns. Here we further explore this connection, and our improved bounds can be seen as a consequence of making the learning algorithm more robust by suitably changing the input LP. Robustness is a topic well-studied in learning theory [12, 22], although existing results do not seem to apply directly to our problem. We remark that a component of robustness more closely related to the standard PAC-learning literature is used in [11].

In recent work, Devanur et al. [10] consider the weaker i.i.d. model for the general allocation problem and substantially improve the lower bound on $B$ to $\Omega\left(\frac{\log m}{\epsilon^2}\right)$, while showing that the lower bound of $\frac{\log m}{\epsilon^2}$ is still required on $B$ to get $(1 - \epsilon)$-competitive algorithms.

**Our results.** Our focus is to understand how large $B$ is required to be in order to allow $(1 - \epsilon)$-competitive algorithms. In particular, the best known bounds for $B$ mentioned above degrade as the number of columns in the LP increases, while the minimum requirement on its magnitude does not. With the trend of handling LP’s with larger number of columns (e.g. these columns correspond to the keywords in the ad allocation problem, which in turn correspond to visits of a search engine’s webpage), this gap is very unsatisfactory from a practical point of view. Furthermore, given that guarantees for the single knapsack case do not depend on the number of columns, it is important to understand if the multi-knapsack case is fundamentally more difficult. In this work, we give a precise indication of why the latter problem was resistant to arguments used in the single knapsack case, and overcome this difficulty to exhibit an algorithm with dimension-independent guarantee.

We show that a modification of DPA [1] that we call Robust DPA obtains a $(1 - \epsilon)$-competitive solution for online packing LP’s with $m$ constraints in the random permutation model whenever $B \geq \Omega\left(\frac{m^2 \log m}{\epsilon^2}\right)$. Another way of stating this result is that the algorithm has competitive ratio $1 - O(\sqrt{m \log B} / \sqrt{B})$. Contrasting to previous results, our guarantee does not depend on $n$ and in the case $m = 1$ matches the bounds for the $B$-Choice Secretary Problem up to lower order terms. We finally remark that we can replace the requirement $B \geq \Omega\left(\frac{m^2 \log m}{\epsilon^2}\right)$ by $OPT \geq \Omega\left(\frac{\pi_{\text{max}}^2 m}{\epsilon^2}\right)$ exactly as done in Section 5.1 of [1].

**High-level outline.** As mentioned before, we use the connection between solving an online LP and PAC-learning a good linear classification of its columns; in order to obtain the improved guarantee, we focus on tightening the bounds for the generalization error of the learning problem. More precisely, solving the LP can be
seen as classifying the columns into 0/1, which corresponds to setting their associated variable to 0/1. Consider a family $\mathcal{X} \subseteq \{0, 1\}^n$ of linear classifications of the columns. Our algorithms essentially sample a set $S$ of columns and learn a classification $x^S \in \mathcal{X}$ which is “good” for the columns $S$ (i.e., obtains large proportional revenue while not filling up the proportionally scaled budget too much). The goal is to upper bound the probability that $x^S$ is not good for the whole LP. This is typically done by union bounding over the classifications in $\mathcal{X}$ [11, 1].

To obtain improved guarantees, we refine the union bound using an argument akin to covering: we consider witness classifications which can be used to bound the probability that any bad classification is learned. The problem is that, when the columns $(\pi_t, a^t)$’s do not lie in a two-dimensional subspace of $\mathbb{R}^m$, the set $\mathcal{X}$ may contain a large number of disjoint bad classifications; this is a roadblock for obtaining a small set of witnesses. In stark contrast, when these columns do lie in a two-dimensional subspace, the (support of the) classifications form a union of two chains with respect to inclusion; in the special case where the $a^t$’s belong to a one-dimensional subspace (e.g., case $m = 1$), they form a single chain. The fact that the latter learning problem is intrinsically more robust than the former seems to precisely capture the increased difficulty in obtained good bounds for the multi-knapsack case.

Motivated by this discussion, we first consider LP’s whose columns $a^t$’s lie in few one-dimensional subspaces (Section 2). For each of these subspaces, we are able to approximate the classifications induced in the columns lying in the subspace by considering a small subset of the induced classifications. Taking the product of these subsets gives us a witness set for $\mathcal{X}$. However, this strategy as stated does not make use of the fact that the subspaces are embedded in an $m$-dimensional space, and hence obtains large witness sets. By establishing a connection between the “useful” terms in the product with faces of a hyperplane arrangement in $\mathbb{R}^m$, we are able to make use of the dimension of the host space and exhibit witness sets of much smaller sizes, which leads to improved bounds.

For the general problem, the idea is to perturb the columns $a^t$’s to make them lie in few one-dimensional subspaces, while not altering the feasibility and optimality of the LP by more than a $(1 + \epsilon)$ factor (Section 3). Finally, we tighten the bound by using the idea of recomputing the classification as the number of columns doubles, following [1] (Section 4).

## 2 OTP for almost 1-dim columns

In this section we analyze the behavior of the algorithm OTP (One-Time Pricing) for LP’s whose columns are contained in few 1-dimensional subspaces of $\mathbb{R}^m$. The overall goal is to find an appropriate dual (perhaps infeasible) solution $p$ for (LP) and use it to classify the columns of the LP. More precisely, given $p \in \mathbb{R}^m$, we define $x(p)_t = 1$ if $\pi_t > pa^t$ and $x(p)_t = 0$ otherwise. Thus, $x(p)$ is the result of classifying the columns $(\pi_t, a^t)$’s with the homogeneous hyperplane in $\mathbb{R}^{m+1}$ with normal $(-1, p)$. The motivation behind this classification is that it selects the columns which have positive reduced cost with respect to the dual solution $p$, or alternatively, it solves to optimality the Lagrangian relaxation using $p$ as multipliers.

**Sampling LP’s.** In order to obtain a good dual solution $p$, we use the (random) LP consisting on the first $s$ columns of (LP) with appropriately scaled right-hand side.

\[
\begin{align*}
\max & \quad \sum_{t=1}^s \pi_{\sigma(t)} x_{\sigma(t)} & (s, \delta)-LP \\
\sum_{t=1}^s a^t_{\sigma(t)} x_{\sigma(t)} & \leq \frac{s}{n} \delta B \\
x_{\sigma(t)} & \in [0, 1] & t = 1, \ldots, s.
\end{align*}
\]

\[
\begin{align*}
\min & \quad \frac{s}{n} \delta B \sum_{i=1}^m p_i + \sum_{t=1}^s \alpha_{\sigma(t)} & (s, \delta)-Dual \\
p a^t_{\sigma(t)} + \alpha_{\sigma(t)} & \geq \pi_{\sigma(t)} & t = 1, \ldots, s \\
p & \geq 0 \\
\alpha & \geq 0.
\end{align*}
\]

Here $\sigma$ denotes the random permutation of the columns of the LP. We use OPT$(s, \delta)$ to denote the optimal value
of \((s, \delta)\)-LP, and \(\text{OPT}(s)\) to denote the optimal value of \((s, 1)\)-LP.

The static pricing algorithm OTP of [1] can then be described succinctly as follows.\(^2\)

1. Wait for the first \(\epsilon n\) columns of \((LP)\) and solve \((\epsilon n, 1 - \epsilon)\)-Dual, letting \((p, \alpha)\) be the obtained dual optimal solution.

2. Use the classification given by \(p\) as above by setting \(x_{\sigma(t)} = x(p)_{\sigma(t)}\) for \(t = \epsilon n + 1, \epsilon n + 2, \ldots\) for as long as the solution obtained remains valid. From this point on set all further variables to zero.

Note that by definition this algorithm outputs a feasible solution with probability one. Our goal is then to analyze the quality of the solution produced, ultimately leading to the following theorem.

**Theorem 2.1** Fix \(\epsilon \in (0, 1]\). Suppose that there are \(K \geq m\) 1-dim subspaces of \(\mathbb{R}^m\) containing the columns \(a^t\)'s and that \(B \geq \Omega \left( \frac{m}{\epsilon^2} \log \frac{K}{\epsilon} \right)\). Then algorithm OTP returns a feasible solution with expected value at least \((1 - 5\epsilon)\text{OPT}\).

Let \(S = \{\sigma(1), \ldots, \sigma(\epsilon n)\}\) be the (random) index set of the columns sampled by OTP. We use \(p^S\) to denote the optimal dual solution obtained by OTP; notice that \(p^S\) is completely determined by \(S\). To simplify the notation, we also use \(x^S\) to denote \(x(p^S)\).

Notice that, for all the scenarios where \(x^S\) is feasible, the solution returned by OTP is identical to \(x^S\) with its components \(x^S_{\sigma(1)}, \ldots, x^S_{\sigma(\epsilon n)}\) set to zero. Given this observation, we can actually focus on proving that \(x^S\) is a good solution.

**Lemma 2.2** Fix \(\epsilon \in (0, 1]\). Suppose that there are \(K \geq m\) 1-dim subspaces of \(\mathbb{R}^m\) containing the columns \(a^t\)'s and that \(B \geq \Omega \left( \frac{m}{\epsilon^2} \log \frac{K}{\epsilon} \right)\). Then with probability at least \((1 - \epsilon)\), \(x^S\) is a feasible solution for \((LP)\) with value at least \((1 - 3\epsilon)\text{OPT}\).

To see how Theorem 2.1 follows from this, first note that

\[
\mathbb{E} \left[ \sum_{t=1}^{\epsilon n} \pi_{\sigma(t)} x_{\sigma(t)} \right] = \mathbb{E} \left[ \sum_{t > \epsilon n} \pi_{\sigma(t)} x^S_{\sigma(t)} \right].
\]

Now let \(E\) denote the event that \(x^S\) is feasible for \((LP)\) with value at least \((1 - 3\epsilon)\text{OPT}\), which occurs with probability at least \((1 - \epsilon)\). By the non-negativity of the profits, we obtain

\[
\mathbb{E} \left[ \sum_{t=1}^{\epsilon n} \pi_{\sigma(t)} x^S_{\sigma(t)} \right] \geq \mathbb{E} \left[ \sum_{t=1}^{n} \pi_{\sigma(t)} x^S_{\sigma(t)} \mid E \right] \Pr(E) \geq (1 - 4\epsilon)\text{OPT}.
\]

Finally noticing that \(\mathbb{E} \left[ \sum_{t \leq \epsilon n} \pi_{\sigma(t)} x^S_{\sigma(t)} \right] \leq \epsilon\text{OPT}\) (see, e.g., Lemma 2.4 of [1]), we then get

\[
\mathbb{E} \left[ \sum_{t > \epsilon n} \pi_{\sigma(t)} x^S_{\sigma(t)} \right] \geq \mathbb{E} \left[ \sum_{t=1}^{n} \pi_{\sigma(t)} x^S_{\sigma(t)} \right] - \epsilon\text{OPT} \geq (1 - 5\epsilon)\text{OPT},
\]

and the result follows.

### 2.1 Connection to PAC learning

We assume from now on that \(B \geq \Omega \left( \frac{m}{\epsilon^2} \log \frac{K}{\epsilon} \right)\). Let \(X = \{x(p) : p \in \mathbb{R}^m_+\} \subseteq \{0, 1\}^n\) denote the set of all possible linear classifications of the LP columns which can be generated by OTP. With slight overload in the notation, we identify a vector \(x \in \{0, 1\}^n\) with the subset of \([n]\) corresponding to its support.

\(^2\)To simplify the exposition, we assume that \(\epsilon n\) is an integer.
Definition 2.3 (Bad solution) Given a scenario, we say that $x^S$ is bad if it does not satisfy the properties of Lemma 2.2, namely $x^S$ is either infeasible or has value less than $(1-3\epsilon)\text{OPT}$. We say that $x^S$ is good otherwise.

As noted in previous work, the main observation used to control the guarantee of the solution output by the algorithm is that it suffices to analyze its budget occupation. To make this precise, given $x \in \{0, 1\}^m$ let $a_i(x) = \sum_{t \in x} a^i_t$ be its occupation of the $i$th budget and let $a^S_i(x) = \frac{1}{\epsilon} \sum_{t \in x \cap S} a^i_t$ be its appropriately scaled occupation of $i$th budget in the sampled LP (recall that $|S| = \epsilon n$).

Recall that the solution $x^S$ is obtained by selecting the columns with positive reduced cost with respect to the optimal dual solution $p^S$. Therefore, it is intuitively clear that $x^S$ resembles an optimal solution for $(\epsilon n, 1-\epsilon)$-LP and thus should (approximately) be feasible and satisfy complementary slackness conditions. Using the assumption that the input is in general position, this is made formal in the following lemma.

Lemma 2.4 In every scenario, $x^S$ satisfies the following: (i) for all $i \in [m]$, $a^S_i(x^S) \leq (1-\epsilon)B$ and (ii) for every $i \in [m]$ with $p^S_i > 0$, $a^S_i(x^S) \geq (1-2\epsilon)B$.

Conversely, the next lemma states that this approximate complementary slackness (now with respect to (LP)) is enough to guarantee near-optimality, making formal a previous observation that budget occupation determines the quality of the solution.\footnote{This lemma can be seen as an approximate version of an observation on Lagrangian relaxation made by Everett in the early 60’s [17] and is also related to the approximate complementary slackness conditions in [26].}

Lemma 2.5 Consider a scenario where $x^S$ satisfies the following: (i) for all $i \in [m]$, $a_i(x^S) \leq B$ and (ii) for all $i \in [m]$ with $p^S_i > 0$, $a_i(x^S) \geq (1-3\epsilon)B$. Then $x^S$ is good.

Given the properties of $x^S$ guaranteed by Lemma 2.4, together with the observation that $a_i(x) = \mathbb{E}[a^S_i(x)]$ for all $x$, the idea is to use concentration inequalities to argue that the conditions in Lemma 2.5 holds with good probability. One difficulty is the presence of correlation between the sample set $S$ and the solution $x^S$. We deal with this difficulty in standard PAC-learning way.

Definition 2.6 (Badly learnable) For a given scenario, we say that $x \in \mathcal{X}$ can be badly learned for budget $i$ if either (i) $a^S_i(x) \leq (1-\epsilon)B$ and $a_i(x) > B$ or (ii) $a^S_i(x) \geq (1-2\epsilon)B$ and $a_i(x) < (1-3\epsilon)B$.

Essentially these are the classifications which look good for the sampled $(\epsilon n, 1-\epsilon)$-LP but are actually bad for (LP). More precisely, Lemmas 2.4 and 2.5 give the following.

Observation 2.7 Consider a scenario for which $x^S$ is bad. Then $x^S = x$ for some $x$ that can be badly learned in this scenario for some budget $i \in [m]$.

This observation directly implies that

$$
\Pr \left( x^S \text{ is bad} \right) \leq \Pr \left( \bigvee_{i \in [m], x \in \mathcal{X}} x \text{ can be badly learned for budget } i \right).
$$

(2.1)

Notice that indeed the right-hand side of this inequality does not depend on $x^S$, it is only a function of how skewed $a^S_i(x)$ is as compared to its expectation $a_i(x)$.

From this point on, usually the right-hand side in the previous equation is upper bounded by taking a union bound over all its terms [1]. However, this strategy can be too wasteful, because if $x$ and $x'$ are “similar” there is a large overlap between the scenarios where $a^S_i(x)$ is skewed and those where $a^S_i(x')$ is skewed. In order to obtain improved guarantees we use something akin to a covering argument, although we need to use a suitable (and non-standard) measure to capture the similarity between classifications.
2.2 Similarity via witnesses

First, we partition the classifications which can be badly learned for budget $i$ into two sets, depending on why they are bad: for $i \in [m]$, let $\mathcal{X}_i^+ = \{ x \in \mathcal{X} : a_i(x) > B \}$ and $\mathcal{X}_i^- = \{ x \in \mathcal{X} : a_i(x) < (1-3\epsilon)B \}$. In order to simplify the notation, given a set $x$ we define $\text{skew}_m(x)$ to be the event that $a^S_m(x) \leq (1-\epsilon)B$ and $\text{skew}_p(x)$ to be the event that $a^S_p(x) \geq (1-2\epsilon)B$. Notice that if $x \in \mathcal{X}_i^+$, then $\text{skew}_m(x)$ is the event that $a^S_m(x)$ is significantly smaller than its expectation (skewed in the minus direction), while for $x \in \mathcal{X}_i^-$ $\text{skew}_p(x)$ is the event that $a^S_p(x)$ is significantly larger than its expectation (skewed in the plus direction). These definitions directly give the equivalence

$$\Pr \left( \bigvee_{i,x \in \mathcal{X}} x \text{ can be badly learned for budget } i \right) = \Pr \left( \bigvee_{i,x \in \mathcal{X}_i^+} \text{skew}_m(x) \vee \bigvee_{i,x \in \mathcal{X}_i^-} \text{skew}_p(x) \right). \quad (2.2)$$

In order to introduce the concept of witnesses, consider two sets $x, x'$, say, in $\mathcal{X}_i^+$. Take a subset $w \subseteq x \cap x'$; the main observation is that, since $a^t_i \geq 0$ for all $t$, for all scenarios we have $a^S_i(w) \leq a^S_i(x)$ and $a^S_i(w) \leq a^S_i(x')$. In particular, the event $\text{skew}_m(x, w) \vee \text{skew}_m(x', w)$ is contained in $\text{skew}_m(x, w)$. The set $w$ serves as a witness for scenarios which are skewed for either $x$ or $x'$; if additionally $a_i(w)$ reasonably larger than $(1-\epsilon)B$, we can then use concentration inequalities over $\text{skew}_m(x, w)$ in order to bound probability of $\text{skew}_m(x, w) \vee \text{skew}_m(x', w)$. This ability of bounding multiple terms of the right-hand side of (2.2) simultaneously is what gives an improvement over the naive union bound.

**Definition 2.8 (Witness)** We say that $\mathcal{W}_i^+$ is a witness set for $X_i^+$ if: (i) for all $w \in \mathcal{W}_i^+$, $a_i(w) \geq (1-\epsilon/2)B$ and (ii) for all $x \in X_i^+$ there is $w \in \mathcal{W}_i^+$ contained in $x$. Similarly, we say that $\mathcal{W}_i^-$ is a witness set for $X_i^-$ if: (i) for all $w \in \mathcal{W}_i^-$, $a_i(w) \leq (1-3\epsilon/2)B$ and (ii) for all $x \in X_i^-$ there is $w \in \mathcal{W}_i^-$ containing $x$.

As indicated by the previous discussion, given witness sets $\mathcal{W}_i^+$ and $\mathcal{W}_i^-$ for $X_i^+$ and $X_i^-$, we directly get the bound

$$\Pr \left( \bigvee_{i,x \in \mathcal{X}_i^+} \text{skew}_m(x) \vee \bigvee_{i,x \in \mathcal{X}_i^-} \text{skew}_p(x) \right) \leq \Pr \left( \bigvee_{i,w \in \mathcal{W}_i^+} \text{skew}_m(w) \vee \bigvee_{i,w \in \mathcal{W}_i^-} \text{skew}_p(w) \right). \quad (2.3)$$

Using this inequality, together with (2.1) and (2.2), we can bound the probability that $x^S$ is bad in term of the size of witnesses sets.

**Lemma 2.9** Suppose that, for all $i \in [m]$, there are witness sets for $\mathcal{X}_i^+$ and $\mathcal{X}_i^-$ of size at most $M$. Then $\Pr(x^S \text{ is bad }) \leq 8mM \exp \left( -\frac{\epsilon B}{33} \right)$.

The usefulness of defining witnesses as such is of course contingent upon the ability of finding witness sets which are much smaller than $\mathcal{X}_i^+$ and $\mathcal{X}_i^-$. One reasonable choice of a witness set for, say, $\mathcal{X}_i^+$ is the collection of all of its minimal sets; unfortunately, this may not give a witness set of small enough size. However, notice that a witness set need not be a subset of $\mathcal{X}_i^+$ (or even $\mathcal{X}$). Allowing elements outside $\mathcal{X}_i^+$ gives the flexibility of obtaining witnesses which are associated to multiple “similar” minimal elements of $\mathcal{X}_i^+$, which is effective in reducing the size of witness sets.

2.3 Good witnesses for almost 1-dim columns

Given the previous lemma, our task is to find small witness sets. Unfortunately, when the $(\pi_t, a^t)$’s lie in a space of dimension at least 3, $\mathcal{X}_i^+$ and $\mathcal{X}_i^-$ may contain many ($\Omega(n)$) disjoint sets (see Figure 2.1), which shows that in general we cannot find small witness sets directly. This sharply contrasts with the case where the $(\pi_t, a^t)$’s lie
in a 2-dimensional subspace of \( \mathbb{R}^{m+1} \). In this case, it is not difficult to show that \( \mathcal{X} \) is a union of 2 chains with respect to inclusion. In the special case where the \( a^t \)'s lie in a 1-dimensional subspace of \( \mathbb{R}^m \), we show that \( \mathcal{X} \) is actually a single chain (Lemma 2.11), and therefore we can take \( W_i^+ \) as the minimal set of \( \mathcal{X}_i^+ \) and \( W_i^- \) as the maximal set of \( \mathcal{X}_i^- \).

Due to the above observations, we focus on LP’s whose \( a^t \)'s lie in few 1-dimensional subspaces. In this case, \( \mathcal{X}_i^+ \) and \( \mathcal{X}_i^- \) are sufficiently well-behaved so that we can find small (independent of \( n \)) witness sets.

**Lemma 2.10** Suppose that there are \( K \geq m \) 1-dimensional subspaces of \( \mathbb{R}^m \) which contain the \( a^t \)'s. Then there are witness sets for \( \mathcal{X}_i^+ \) and \( \mathcal{X}_i^- \) of size at most \( O(\frac{K}{\varepsilon} \log \frac{K}{\varepsilon})^m \).

Assuming the hypothesis of the lemma, partition the index set \([n]\) into \( C_1, C_2, \ldots, C_K \) such that for all \( j \in [K] \) the columns \( \{a^t\}_{t \in C_j} \) belong to the same 1-dimensional subspace. Equivalently, for each \( j \in [K] \) there is a vector \( e^j \) of \( \ell_\infty \)-norm 1 such that for all \( t \in C_j \) we have \( a^t = \|a^t\|_\infty e^j \). An important observation is that now we can order the columns (locally) by the ratio of profit over budget occupation: without loss of generality assume that for all \( j \in [K] \) and \( t, t' \in C_j \) with \( t < t' \), we have \( \frac{\pi_t}{\|a^t\|_\infty} \geq \frac{\pi_{t'}}{\|a^{t'}\|_\infty} \).

Given a classification \( x \), we use \( x|_{C_j} \) to denote its projection onto the coordinates in \( C_j \); so \( x|_{C_j} \) is the induced classification on columns with indices in \( C_j \). Identifying singleton sets with their only element, we use the product notation for \( x = \prod_{j \in [K]} x|_{C_j} \). Similarly, we define \( \mathcal{X}|_{C_j} = \{ x|_{C_j} : x \in \mathcal{X} \} \) as the set of all classifications induced in the columns in \( C_j \).

Strengthening a previous observation, the main property that we get from working with 1-dim subspaces is the following.

**Lemma 2.11** For each \( i \in [K] \), the sets in \( \mathcal{X}|_{C_j} \) are prefixes of \( C_j \).

**Proof.** Fix \( j \in [K] \). Consider a set \( x \in \mathcal{X} \) and let \( p \) be a dual vector such that \( x(p) = x \). Let \( t' \) be the last index of \( C_j \) which belongs to \( x|_{C_j} \); this implies that \( \pi_{t'} > p a^{t'} = p \|a^{t'}\|_\infty \), or alternatively \( \frac{\pi_{t'}}{\|a^{t'}\|_\infty} > p \). By the ordering of the columns, for all \( t \in C_j \) smaller than \( t' \) we have \( \frac{\pi_t}{\|a^t\|_\infty} \geq \frac{\pi_{t'}}{\|a^{t'}\|_\infty} > p \) and hence \( x \in x|_{C_j} \). By definition of \( t' \) it follows that \( x|_{C_j} = \{ t \in C_j : t \leq t' \} \), a prefix of \( C_j \); this concludes the proof.

To simplify the notation, fix \( i \in [m] \) for the rest of this section, so we aim at providing witness sets for \( \mathcal{X}_i^+ \) and \( \mathcal{X}_i^- \). It is instructive to map a classification \( x = \prod_{j=1}^K x|_{C_j} \) to a box with sides of length \( a_i(x|_{C_j}) \). The idea for producing a witness set for \( \mathcal{X}_i^+ \) is simple: for \( x \in \mathcal{X}_i^+ \), we include in the witness set a classification \( w \) whose sides are prefixes of the \( C_j \)'s obtained by shortening the sides of \( x \) (more specifically, rounding their lengths down to a power of \( (1 + \varepsilon) \)). The point is that all boxes in \( \mathcal{X}_i^+ \) which have sides in the same powers of \( (1 + \varepsilon) \) will give rise to the same witness. Using the fact that “reasonable” boxes in \( \mathcal{X}_i^+ \) have side lengths upper bounded by \( O(B) \), this gives a witness set of size only dependent on \( \varepsilon, B \) and \( m \).

---

4 Notice that this ratio is well-defined since by assumption \( a^t \neq 0 \) for all \( t \in [n] \).
To make this formal, we first classify boxes according to their size lengths. Start by covering the interval \([0, B + m]\) with intervals \(\{I_\ell\}_{\ell \in L}\), where \(I_0 = [0, B], I_\ell = [\frac{\ell B}{4K} (1 + \frac{\epsilon}{4})^{\ell - 1}, \frac{\ell B}{4K} (1 + \frac{\epsilon}{4})^\ell]\) for \(\ell > 0\) and \(L = \{0, \ldots, \lceil \log_{1+\epsilon/4} \frac{8K}{\epsilon} \rceil\}\) (note that since \(B \geq m\), we have \(B + m \leq 2B\)). Define \(B^\ell_{i,j}\) as the set of classifications \(x \in \mathcal{X}|_{C_j}\) whose budget occupation \(a_i(x)\) lies in the interval \(I_\ell\). For \(v \in L^K\), define the family of classifications \(B_i^v = \prod_j B^v_{i,j}\) Notice that every box in \(B_i^v\) has similar (within \((1 + \epsilon/4)\)) side lengths. Also note that the \(B_i^v\)'s may include classifications not in \(\mathcal{X}\) and may not include classifications in \(\mathcal{X}\) which have occupation \(a_i(\cdot)\) greater than \(B + m\).

Now consider a non-empty \(B_i^v\). Let \(w_i^v\) be the inclusion-wise smallest element in \(B_i^v\). Notice that such unique smallest element exists: since \(\mathcal{X}|_{C_j}\) is a chain, so is \(B^\ell_{i,j}\), and hence \(w_i^v\) is the product (over \(j\)) of the smallest element in \(B^\ell_{i,j}\). Similarly, let \(w_i^v\) denote the largest element in \(B_i^v\). Intuitively, \(w_i^v\) and \(w_i^v\) will serve as witnesses for all the sets in \(B_i^v\).

Finally, define the witness sets by adding the \(w_i^v\) and \(w_i^v\)'s of appropriate size corresponding to meaningful \(B_i^v\)'s: set \(W_i^+ = \{w_i^v : B_i^v \cap \mathcal{X} \neq \emptyset, a_i(w_i^v) \geq (1 - \epsilon/2)B\}\) and \(W_i^- = \{w_i^v : B_i^v \cap \mathcal{X} \neq \emptyset, a_i(w_i^v) \leq (1 - 3\epsilon/2)B\}\).

It is not too difficult to see that indeed, \(W_i^+\) is a witness set for \(\mathcal{X}^+\): If \(x \in \mathcal{X}^+\) belongs to some \(B_i^v\), then \(w_i^v\) belongs to \(W_i^+\) and is easily shown to be a witness for \(x\). However, if \(x\) does not belong to any \(B_i^v\), by having too large sides, the idea is to find a smaller set \(x' \subseteq x\) which belongs to some \(B_i^v\) and to \(\mathcal{X}\), and then use \(w_i^v\) as a witness for \(x\). We note that considering \(B_i^v\)'s for side lengths at most \(B + m\) and only adding witnesses for \(B_i^v\)'s which intersect \(\mathcal{X}\) are crucially used when bounding the size of \(W_i^+\) and \(W_i^-\).

**Lemma 2.12** The sets \(W_i^+\) and \(W_i^-\) are witness sets for \(\mathcal{X}^+\) and \(\mathcal{X}^-\).

Clearly these witness sets have size at most \((\log_{1+\epsilon/4} \frac{8K}{\epsilon} + 1)^K\). Although this size is independent of \(n\), it is still unnecessarily large since it only uses locally for each \(C_j\) the fact that \(\mathcal{X}\) consists of linear classifications; in particular, it does not use the dimension of the ambient space \(\mathbb{R}^m\). Suppose that \(J \subseteq K\), of cardinality \(m\), is such that the directions \(\{e_j\}_{j \in J}\) form a basis of \(\mathbb{R}^m\). Knowing the partial classification \(x(p)|_{C_j}\), or more precisely the value of \(p e_j\), for all \(j \in J\) completely determines the whole classification \(x(p)\). Similarly, knowing that \(x(p)|_{C_j} \in B^\ell_{i,j}\) for all \(j \in J\) should give some information about which \(B^\ell_{i,j}\)'s \(x(p)|_{C_j}\) can belong to for \(j \notin J\); this indicates that there are not enough degrees of freedom to allow a linear classification in \(B_i^v\) for each \(v \in L^K\). The difficulty in making this argument formal is that the latter information does not completely determine which \(B_i^v\) the classification \(x(p)\) belongs to. The idea is not to use a fixed set \(J\) of indices, but look at the whole \(K\) simultaneously.

**Lemma 2.13** At most \((O(\frac{K}{\epsilon} \log \frac{K}{\epsilon}))^m\) of the \(B_i^v\)'s contain an element from \(\mathcal{X}\).

**Proof.** In order to capture the fact that our classification is obtained via dual vectors in \(\mathbb{R}^m\), we move from analyzing classifications to analyzing dual vectors. For \(v \in L^K\) define \(P^v\) as the set of non-negative dual vectors \(p\) such that \(x(p)\) belongs to \(B_i^v\). It suffices to prove that at most \((O(\frac{K}{\epsilon} \log \frac{K}{\epsilon}))^m\) of the families \(P^v\)'s are non-empty. The main idea is to use that fact that the \(P^v\)'s come from a hyperplane arrangement [23] in \(\mathbb{R}^m\).

To start, for \(j \in [K]\) and \(\ell \in L\) define \(P^\ell_j = \{p \in \mathbb{R}^m : x(p)|_{C_j} \in B^\ell_{i,j}\}\). Since \(x(p) \in B_i^v\) if and only if for all \(j \in [K]\) we have \(x(p)|_{C_j} \in B^\ell_{i,j}\), it follows that \(P^v = \bigcap_j P^\ell_j\). Let \(\tau^\ell_j\) denote the first index in \(C_j\) such that the prefix \(\{t \in C_j : t \leq \tau^\ell_j\}\) occupies the budget \(i\) to the extent in \(I_\ell\). Using Lemma 2.11 and the fact that the \(a_j\)'s are non-negative, we get that \(B^\ell_{i,j}\) is the set of all prefixes of \(C_j\) which contain \(\tau^\ell_j\) but do not contain \(\tau_{j+1}^\ell\). Moreover, notice that the set \(x(p)|_{C_j}\) contains \(\tau^\ell_j\) if and only if \(\pi_{\tau^\ell_j} > p a_j^\ell\). It then follows from these observations we can express the set \(P^\ell_j\) using linear inequalities: \(P^\ell_j = \{p \in \mathbb{R}^m : \pi_{\tau^\ell_j} > p a_j^\ell, \pi_{\tau_{j+1}^\ell} \leq p a_{j+1}^\ell\}\). Since \(P^v = \bigcap_j P^\ell_j\), we have that \(P^v\) is given by the intersection of halfspaces defined by hyperplanes of the form \(\pi_{\tau^\ell_j} = p a_j^\ell\) and \(p_i = 0\).
So consider the arrangement given by all hyperplanes \( \{ \pi_{ij} = pa_{ij}^t\}_{j \in [K], \ell \in L} \) and \( \{ p_i = 0 \}_{i=1}^m \). Given a face \( F \) in this arrangement and a set \( P^\nu \), either \( F \) is contained in \( P^\nu \) or these sets are disjoint. Since the faces of the arrangement cover \( \mathbb{R}^m \), it follows that each non-empty \( P^\nu \) contains at least one these faces.

Notice that the arrangement is defined by \( |K| \cdot |L|^K \cdot m \leq O(\frac{Km}{\epsilon} \log \frac{K}{\epsilon}) \) hyperplanes, where the last inequality uses the fact that \( \log(1 + \frac{1}{\epsilon}) \geq \epsilon \log(1 + \frac{1}{\epsilon}) \) holds (by concavity) for \( \epsilon \in [0, 1] \). It is known that an arrangement with \( h \geq m \) hyperplanes in \( \mathbb{R}^m \) has at most \( (\frac{h}{m})^m \) faces (see Section 6.1 of [23] and p. 82 of [24]). Using the conclusion of the previous paragraph, we get that there are at most \( (O(\frac{K}{\epsilon} \log \frac{K}{\epsilon}))^m \) non-empty \( P^\nu \)'s and the result follows.

This lemma implies that \( \mathcal{W}_i^+ \) and \( \mathcal{W}_i^- \) each has size at most \( (O(\frac{K}{\epsilon} \log \frac{K}{\epsilon}))^m \), which then proves Lemma 2.10. Finally, applying Lemma 2.9 we conclude the proof of Lemma 2.2.

### 3 Robust OTP

In this section we consider (LP) with columns that may not belong to few 1-dimensional subspaces. Given the results of the previous section, the idea is clear: we would like to perturb the columns of this LP so that it belongs to few 1-dim subspaces and such that an approximate solution this perturbed LP is also an approximate solution for the original one. More precisely, we will obtain a set of vectors \( Q \subseteq \mathbb{R}^m \) and transform each the vector \( a^t \) into \( \tilde{a}^t \) which is a scaling of a vector in \( Q \), and we let the rewards \( \pi_t \) remain unchanged.

A basic but crucial observation is that solutions to an LP are robust to slight changes in the constraint matrix. The following lemma makes this precise and will guide us to obtaining the desired set \( Q \).

**Lemma 3.1** Consider real numbers \( \pi_1, \ldots, \pi_n \) and vectors \( a^1, \ldots, a^n \) and \( \tilde{a}^1, \ldots, \tilde{a}^n \) in \( \mathbb{R}^m_+ \) such that \( \|\tilde{a}^t - a^t\|_\infty \leq \frac{\epsilon}{m+1} \|a^t\|_\infty \). If \( x \) is an \( \epsilon \)-approximate solution for (LP) with columns \( (\pi_t, \tilde{a}^t) \) and right-hand side \((1 - \epsilon)B\), then \( x \) is a \( 2\epsilon \)-approximate solution for the LP (LP).

**Perturbing the columns.** To simplify the notation, set \( \delta = \frac{\epsilon}{m+1} \); also, for simplicity of exposition we assume that \( 1/\delta \) is integral.

When constructing \( Q \), we want the rays spanned by the each of its vectors to be “uniform” over \( \mathbb{R}^m_+ \). Naturally, we focus on the intersection of these rays and the unit \( \ell_\infty \) sphere: we set \( Q \) to be a \( \delta \)-net of the latter. More explicitly, we take \( Q \) to be the vectors in \( \{(0, \delta, 2\delta, 3\delta, \ldots, 1)^m\} \) which have \( \ell_\infty \) norm 1. Note that \( |Q| = (O(\frac{m}{\epsilon}))^m \).

Given a vector \( a^t \) with \( \ell_\infty \)-norm 1, we set the transformed vector \( \tilde{a}^t \) to be the vector in \( Q \) closest (in \( \ell_\infty \)) to \( a^t \). More generally, we let \( \tilde{a}^t = \|a^t\|_\infty q^t \), where \( q^t \) is the vector in \( Q \) closest to \( \frac{a^t}{\|a^t\|_\infty} \).

By definition of \( Q \), for every vector \( v \in \mathbb{R}^m \) of unit \( \ell_\infty \)-norm, there is a vector \( q \in Q \) with \( \|v - q\|_\infty \leq \delta \). Using this observation, it follows that the vectors \( \tilde{a}^t \) satisfy the property required in Lemma 3.1:

\[
\|a^t - \tilde{a}^t\|_\infty = \|a^t\|_\infty \left\| \frac{a^t}{\|a^t\|_\infty} - q^t \right\|_\infty \leq \delta \|a^t\|_\infty.
\]

**Algorithm Robust OTP.** One way to think of the algorithm Robust OTP is that it works in two phases. First, it transforms the vectors \( a^t \) into \( \tilde{a}^t \) as described above. Then it returns the solution obtained by running the algorithm OTP over the LP with columns \( (\pi_t, \tilde{a}^t) \) and right-hand side \((1 - \epsilon)B\). Notice that this algorithm can indeed be implemented to run in an online fashion.

Putting together the discussion in the previous paragraphs and the guarantee of OTP for almost 1-dim columns given by Theorem 2.1 with \( K = |Q| = (O(\frac{m}{\epsilon}))^m \), we obtain the following theorem.

**Theorem 3.2** Fix \( \epsilon \in (0, 1] \) and suppose that \( B \geq \Omega \left( \frac{m^2}{\epsilon^2} \log \frac{m}{\epsilon} \right) \). Then the algorithm Robust OTP returns a solution to the online (LP) with expected value at least \((1 - 10\epsilon)OPT\).
4 Robust DPA

In this section we describe our final algorithm, which has an improved dependence on $1/\epsilon$. Following [1], the idea is to update the dual vector used in the classification as new columns arrive. More precisely, we use the first $2^e n$ columns to classify columns $2^e n + 1, \ldots, 2^{e+1} n$. This leads to improved generalization bounds, which in turn give the reduced dependence on $1/\epsilon$.

The algorithm Robust DPA (as the algorithm DPA) can be seen as a combination of solutions to multiple sampled LP’s, obtained via a modification of OTP denoted by $(s, \delta)$-OTP.

Algorithm $(s, \delta)$-OTP. This algorithm aims at solving the LP $(2s, 1)$-LP and can be described as follows: it finds an optimal dual solution $(p, \alpha)$ for $(s, (1 - \delta))$-LP and sets $x_t = x(p)_t$ (for $s < t \leq 2s$, 0 otherwise), but stops picking columns to guarantee that $\sum_{t=s}^{2s} a^\sigma(t)x_{\sigma(t)} \leq \frac{\delta}{n} B$.

The analysis of $(s, \delta)$-OTP is similar to the one employed for OTP. The main difference is that this algorithm tries to approximate the value of the random LP $(2s, 1)$-LP. This requires a partition of the bad classifications which is more refined than simply splitting into $\mathcal{X}_i^+$ and $\mathcal{X}_i^-$, and witness sets need to be redefined appropriately. Nonetheless, using these ideas we can prove the following guarantee for $(s, \delta)$-OTP.

Lemma 4.1 Suppose that there are $K \geq m$ 1-dim subspaces of $\mathbb{R}^m$ containing the columns $a^t$'s. Fix an integer $s$ and a real number $\delta \in (0, 1/10)$ such that $\frac{\epsilon^2 n}{s} \geq \Omega(m \ln \frac{K}{\delta})$. Then algorithm $(s, \delta)$-OTP returns a feasible solution for $(2s, 1)$-LP with expected value at least $(1 - 3\delta)E[OPT(2s)] - E[OPT(s)] - \delta^2 OPT$.

Algorithm Robust DPA. In order to simplify the description of the algorithm, we assume in this section that $\log(1/\epsilon)$ is an integer.

Again the algorithm Robust DPA be thought of in two phases. In the first phase it converts the vectors $a^t$ into $\tilde{a}^t$, just as in the first phase of Robust OTP. In the second phase, for $i = 0, \ldots, \log(1/\epsilon) - 1$, it runs $(\epsilon^2 n, \sqrt{\epsilon/2})$-OTP over (LP) with columns $(\pi_i, \tilde{a}^i)$ and right-hand side $(1 - \epsilon)B$ to obtain the solution $x^i$. The algorithm finally returns the solution $x$ consisting of the “union” of $x^i$'s: $x = \sum_i x^i$.

Note that the second phase corresponds exactly to using the first $\epsilon^2 n$ columns to classify the columns $\epsilon^2 n + 1, \ldots, \epsilon 2^{i+1} n$. This relative increase in the size of the training data for each learning problem allow us to reduce the dependence of $B$ on $\epsilon$ in each of the iterations, while the error from all the iterations telescope and are still bounded as before. Furthermore, notice that Robust DPA can be implemented to run online.

The analysis of Robust DPA reduces to that of $(s, \delta)$-OTP. That is, using the definition of the parameters of $(s, \delta)$-OTP used in Robust DPA and Lemma 4.1, it is routine to check that the algorithm produces a feasible solution which has expected value $(1 - \epsilon)OPT$. This is formally stated in the following theorem.

Theorem 4.2 Fix $\epsilon \in (0, 1/100)$ and suppose that $B \geq \Omega\left(\frac{m^2}{\epsilon^2} \ln \frac{m}{\epsilon}\right)$. Then the algorithm Robust DPA returns a solution to the online LP (LP) with expected value at least $(1 - 50\epsilon)OPT$.

5 Open problems

A very interesting open question is whether the techniques introduced in this work can be used to obtain improved algorithms for generalized allocation problems [14]. The difficulty in this problem is that the classifications of the columns are not linear anymore; they essentially come from a conjunction of linear classifiers. Given this additional flexibility, having the columns in few 1-dimensional subspaces does not seem to impose strong enough properties in the classifications. It would be interesting to find the appropriate geometric structure of the columns in this case.

Of course a direct open question is to improve the lower or upper bound on the dependence on the right-hand side $B$ to obtain $(1 - \epsilon)$-competitive algorithms.
References


