Geometry of Online Packing Linear Programs*

Marco Molinaro and R. Ravi

Carnegie Mellon University

Abstract. We consider packing LP's with m rows where all constraint coefficients are normalized to be in the unit interval. The n columns arrive in random order and the goal is to set the corresponding decision variables irrevocably when they arrive to obtain a feasible solution maximizing the expected reward. Previous $(1-\epsilon)$-competitive algorithms require the right-hand side of the LP to be $\Omega(\frac{m}{\epsilon^2}\log\frac{n}{\epsilon})$, 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 and known lower bounds for the general case are also independent of n.
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 refute this by exhibiting an algorithm which is $(1-\epsilon)$-competitive as long as the right-hand sides are $\Omega(\frac{m^2}{\epsilon^2}\log\frac{m}{\epsilon})$. Our techniques refine previous PAC-learning based approaches which interpret the online decisions as linear classifications of the columns based on sampled dual prices. The key ingredient of our improvement comes from a non-standard covering argument together with the realization that only when the columns of the LP belong to few 1-d subspaces we can obtain small such covers; bounding the size of the cover constructed also relies on the geometry of linear classifiers. General packing LP's are handled by perturbing the input columns, which can be seen as making the learning problem more robust.

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 [6, 7]. 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 constraint, the Secretary Problem,

* Full version available at http://arxiv.org/abs/1204.5810. The first author is supported by NSF grant CMMI1024554 and the second author is supported in part by NSF award CCF-1143998.

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 and only recently algorithms with solution guarantees have been developed [1, 10, 14].

The Model. We study online packing LP's in the *random permutation model*. Consider a fixed but unknown LP with n columns $a^1, a^2, \ldots, a^n \in [0,1]^m$, whose associated variables are constrained to be in $[0,1]$, and m packing constraints:

$$\begin{aligned} \text{OPT} = \max \sum_{t=1}^{n} \pi_t x_t & \\ \sum_{t=1}^{n} a^t x_t \le B & \qquad \text{(LP)} \\ x_t \in [0,1]\,. & \end{aligned}$$

We know B in advance but columns and their associated π_t's are presented in uniformly random order, and when 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.[1] The goal is to obtain a feasible solution while maximizing its value. 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 π_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 [1, 11].

Related work. The random permutation model has grown in popularity [4, 11, 16] since it avoids strong lower bounds of the pessimistic adversarial-order model [8] while still capturing the lack of total information about the input. Different online problems have already been studied in this model, including bin-packing [19], matchings [16, 18], the AdWords Problem [11] and different generalizations of the Secretary Problem [2, 4, 5, 17, 23]. Closest to our work are packing problems with a single knapsack constraint. In [20], 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 Secretary 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, results for the more general online packing LP's considered here were only recently obtained by Feldman et al. [14] and Agrawal et al. [1]. The

[1] Knowing n up to $1 \pm \epsilon$ factor is enough; this is required for non-trivial competitive ratios [11].

first paper presents an algorithm that obtains with high probability a solution of value at least $(1-\epsilon)$OPT whenever $B \geq \Omega(\frac{m \log n}{\epsilon^3})$ and OPT $\geq \Omega(\frac{\pi_{\max} m \log n}{\epsilon})$, where $\pi_{\max}$ is the largest profit. In the second paper, the authors present an algorithm which obtains a solution of expected value at least $(1-\epsilon)$OPT under the weaker assumptions $B \geq \Omega(\frac{m}{\epsilon^2} \log \frac{n}{\epsilon})$ or OPT $\geq \Omega(\frac{\pi_{\max} m^2}{\epsilon^2} \log \frac{n}{\epsilon})$. One other way of stating this result is that the algorithm has competitive ratio $1 - O(\sqrt{m \log(n) \log B}/\sqrt{B})$; this guarantee degrades as n increases. The current lower bound on B to allow $(1-\epsilon)$-competitive algorithms is $B \geq \frac{\log m}{\epsilon^2}$, also presented in [1]. We remark that these algorithms actually work for more general allocation problems, where a set of columns representing various options arrive at each step and the solution may choose at most one of the options.

Both of the above algorithms use a connection between solving the online LP and PAC-learning [9] a linear classification of its columns, which was initiated by Devanur and Hayes [11] in the context of the AdWords problem. 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, 21], 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 was also used by Devanur and Hayes [11].

In recent work, Devanur et al. [10] consider the weaker *i.i.d. model* for the general allocation problem. While in the random permutation model one assumes that columns are sampled without replacement, in the i.i.d. model they are sampled with replacement. Making use of the independence between samples, Devanur et al. substantially improve requirement on B to $\Omega(\frac{\log(m/\epsilon)}{\epsilon^2})$ while showing that the lower bound $\Omega(\frac{\log m}{\epsilon^2})$ still holds in this model. We remark, however, that these models can present very different behaviors: as a simple example, consider an LP with n columns, $m = 1$ constraints and budget $B = 1$, where only one of the columns has $\pi_1 = a^1 = 1$ and all others have $\pi_t = a^t = 0$; in the random permutation model the expected value of the optimal solution is 1, while in the i.i.d. model this value is $1 - (1 - 1/n)^n \to 1 - 1/e$. The competitiveness of the algorithm of [10] under the random permutation model is still unknown and was left as an open problem by the authors.

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 requirements for B in the above algorithms degrade as the number of columns in the LP increases, while the the lower bound does not. With the trend of handling LP's with larger number of columns (e.g. 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 the DPA algorithm from [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 $B \geq \Omega(\frac{m^2}{\epsilon^2} \log \frac{m}{\epsilon})$. Another way of stating this result is that the algorithm has competitive ratio $1 - O(m\sqrt{\log B}/\sqrt{B})$. 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) and improves [3] for large B. We remark that we can replace the requirement $B \geq \Omega(\frac{m^2}{\epsilon^2} \log \frac{m}{\epsilon})$ by OPT $\geq \Omega(\frac{\pi_{\max} m^3}{\epsilon^2} \log \frac{m}{\epsilon})$ 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 sample a set S of columns and learn a classification $x^S \in \mathcal{X}$ which is 'good' for the columns in 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 via a union bound over the classifications in $\mathcal{X}$ [1, 11].

To obtain improved guarantees, we refine this bound using an argument akin to covering: we consider *witnesses* (Section 2.2), which are representatives of groups of 'similar' bad classifications that can be used to bound the probability that *any* classification in the group is learned; for that we need to use a non-standard measure of similarity between classifications which is based on the budget of the LP. The problem is that, when the columns (π_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 mutually dissimilar 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 (e.g., $m = 1$), these classifications have a much nicer structure which admits a small set of witnesses. This indicates that the latter learning problem is intrinsically more robust than the former, which seem to precisely capture the increased difficulty in obtained good bounds for the multi-row 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; patching together these partial classifications 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 leads to large witness sets. By establishing a connection between the 'useful' patching possibilities with faces of a hyperplane arrangement in $\mathbb{R}^m$ (Lemma 7), 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 a general packing LP, we perturb the columns a^t's to make them lie in few one-dimensional subspaces that form an 'ϵ-net' of the space, while not altering the feasibility and optimality of the LP by more than a $(1 \pm \epsilon)$ factor (Section 3). Finally, we tighten the bound by using the idea of periodically recomputing the classification, following [1] (Section 4). We remark that omitted proofs are presented in the full version [22].

2 OTP for almost 1-dim columns

In this section we describe and analyze the algorithm OTP (One-Time Pricing) over 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 (π_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 that uses 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.

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

$$\begin{aligned} \min \frac{s}{n}\delta B \sum_{i=1}^{m} p_i + \sum_{t=1}^{s} \alpha_{\sigma(t)} & \\ & \qquad ((s,\delta)\text{-Dual}) \\ pa^{\sigma(t)} + \alpha_{\sigma(t)} \geq \pi_{\sigma(t)} & \quad t = 1, \ldots, s \\ p \geq 0 & \\ \alpha \geq 0. & \end{aligned}$$

Here σ denotes the random permutation of the columns of the LP. We use OPT(s, δ) to denote the optimal value of (s, δ)-LP and OPT(s) to denote the optimal value of $(s, 1)$-LP.

The static pricing algorithm OTP of [1] can then be described as follows.[2]

1. Wait for the first ϵn columns of the LP (indexed by $\sigma(1), \sigma(2), \ldots, \sigma(\epsilon n)$) and solve $(\epsilon n, 1-\epsilon)$-Dual. Let (p, α) 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 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* $B \geq \Omega\left(\frac{m}{\epsilon^3} \log \frac{K}{\epsilon}\right)$. *Then algorithm* OTP *returns a feasible solution with expected value at least* $(1 - 5\epsilon)$*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 and the fact that $\mathbb{E}[\sum_{t \leq \epsilon n} \pi_{\sigma(t)} x^S_{\sigma(t)}] \leq \epsilon$OPT, one can prove that the following proposition implies Theorem 1.

Proposition 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* $B \geq \Omega\left(\frac{m}{\epsilon^3} \log \frac{K}{\epsilon}\right)$. *Then with probability at least* $(1-\epsilon)$, x^S *is a feasible solution for* (LP) *with value at least* $(1 - 3\epsilon)$*OPT.*

[2] To simplify the exposition, we assume that ϵn is an integer.

2.1 Connection to PAC learning

We assume from now on that $B \geq \Omega(\frac{m}{\epsilon^3} \log \frac{K}{\epsilon})$. Let $\mathcal{X} = \{x(p) : p \in \mathbb{R}^m_+\} \subseteq \{0,1\}^n$ 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.

Definition 1 (Bad solution). *Given a scenario, we say that x^S is* bad *if it does not satisfy the properties of Proposition 1, namely x^S is either infeasible or has value less than $(1-3\epsilon)$OPT. We say that x^S is* good *otherwise.*

As noted in previous work, since our decisions are made based on reduced costs it suffices to analyze the *budget occupation* (or complementary slackness) of the solution in order to understand its *value*. To make this precise, given $x \in \{0,1\}^n$ let $a_i(x) = \sum_{t \in x} a_i^t$ be its occupation of the ith budget and let $a_i^S(x) = \frac{1}{\epsilon} \sum_{t \in x \cap S} a_i^t$ be its appropriately scaled occupation of ith budget in the sampled LP (recall $|S| = \epsilon n$).

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

Moreover, since we are making decisions based on the *optimal* reduced cost for the sampled LP, our solution satisfies the above properties for the sampled LP.

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

Given that $a_i(x) = \mathbb{E}[a_i^S(x)]$ for all x, the idea is to use concentration inequalities to argue that the conditions in Lemma 1 hold with good probability. Although concentration of $a_i^S(x)$ for *fixed* x can be achieved via Chernoff-type bounds, the quantity $a_i^S(x^S)$ has undesired correlations; obtaining an effective bound is the main technical contribution of this paper.

Definition 2 (Badly learnable). *For a given scenario, we say that $x \in \mathcal{X}$ can be* badly learned for budget i *if either (i) $a_i^S(x) \leq (1-\epsilon)B$ and $a_i(x) > B$ or (ii) $a_i^S(x) \geq (1-2\epsilon)B$ 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). Putting Lemmas 1 and 2 together and unraveling the definitions gives that

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

Notice that the right-hand side of this inequality does not depend on x^S, it is only a function of how skewed $a_i^S(x)$ is as compared to its expectation $a_i(x)$ (over all $x \in \mathcal{X}$).

Usually the right-hand side in the previous equation is upper bounded by taking a union bound over all its terms [1]. Unfortunately this is too wasteful: when x and x' are 'similar' there is a large overlap between the scenarios where $a_i^S(x)$ is skewed and those where $a_i^S(x')$ is skewed. In order to obtain improved guarantees, we introduce in the next section a new way of bounding the right-hand side of the above expression.

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 $\mathrm{skewm}_i(\epsilon, x)$ to be the event that $a_i^S(x) \le (1-\epsilon)B$ and $\mathrm{skewp}_i(\epsilon, x)$ to be the event that $a_i^S(x) \ge (1-2\epsilon)B$. Notice that if $x \in \mathcal{X}_i^+$, then $\mathrm{skewm}_i(\epsilon, x)$ is the event that $a_i^S(x)$ is significantly smaller than its expectation (skewed in the minus direction), while for $x \in \mathcal{X}_i^-$ $\mathrm{skewp}_i(\epsilon, x)$ is the event that $a_i^S(x)$ is significantly larger than its expectation (skewed in the plus direction). These definitions directly give the equivalence

$$\Pr\left(\bigvee_{i, x \in \mathcal{X}} x \text{ can be badly learned for budget } i\right) = \Pr\left(\bigvee_{i, x \in \mathcal{X}_i^+} \mathrm{skewm}_i(\epsilon, x) \vee \bigvee_{i, x \in \mathcal{X}_i^-} \mathrm{skewp}_i(\epsilon, x)\right).$$

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 $a^t \ge 0$ for all t, for all scenarios we have $a_i^S(w) \le a_i^S(x)$ and $a_i^S(w) \le a_i^S(x')$. In particular, the event $\mathrm{skewm}_i(\epsilon, x) \vee \mathrm{skewm}_i(\epsilon, x')$ is contained in $\mathrm{skewm}_i(\epsilon, w)$. 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 $\mathrm{skewm}_i(\epsilon, w)$ in order to bound probability of $\mathrm{skewm}(\epsilon, x) \vee \mathrm{skewm}(\epsilon, x')$. 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 3 (Witness). *We say that $\mathcal{W}_i^+$ is a* witness set *for $\mathcal{X}_i^+$ if: (i) for all $w \in \mathcal{W}_i^+$, $a_i(w) \ge (1-\epsilon/2)B$ and (ii) for all $x \in \mathcal{X}_i^+$ there is $w \in \mathcal{W}_i^+$ contained in x. Similarly, we say that $\mathcal{W}_i^-$ is a* witness set *for $\mathcal{X}_i^-$ if: (i) for all $w \in \mathcal{W}_i^-$, $a_i(w) \le (1-3\epsilon/2)B$ and (ii) for all $x \in \mathcal{X}_i^-$ 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 $\mathcal{X}_i^+$ and $\mathcal{X}_i^-$, we directly get the bound

$$\Pr\left(\bigvee_{i, x \in \mathcal{X}_i^+} \mathrm{skewm}(\epsilon, x) \vee \bigvee_{i, x \in \mathcal{X}_i^-} \mathrm{skewp}(\epsilon, x)\right) \le \Pr\left(\bigvee_{i, w \in \mathcal{W}_i^+} \mathrm{skewm}(\epsilon, w) \vee \bigvee_{i, w \in \mathcal{W}_i^-} \mathrm{skewp}(\epsilon, w)\right). \tag{2.1}$$

Putting together the last three displayed equations and using Chernoff-type bounds, we can get an upper estimate on the probability that x^S is bad in terms of the size of witnesses sets.

Lemma 3. *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 $\Pr(x^S$ is bad $) \le 8mM \exp\left(-\frac{\epsilon^3 B}{33}\right)$.*

One natural 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. But 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 Small witness sets for almost 1-dim columns

Given the previous lemma, our task is to find small witness sets. Unfortunately, when the (π_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 [22], which shows that in general we cannot find small witness sets directly. This sharply contrasts with the case where the (π_t, a^t)'s lie in a 2-dimensional subspace of $\mathbb{R}^{m+1}$, where one can 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 5) and therefore we can take $\mathcal{W}_i^+$ as *the* minimal set of $\mathcal{X}_i^+$ and $\mathcal{W}_i^-$ 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 4. *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 $(O(\frac{K}{\epsilon}\log\frac{K}{\epsilon}))^m$.*

To prove this lemma, assume its hypothesis and 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 c^j of ℓ_∞-norm 1 such that for all $t \in C_j$ we have $a^t = \|a^t\|_\infty c^j$. 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 $\frac{\pi_t}{\|a^t\|_\infty} \geq \frac{\pi_{t'}}{\|a^{t'}\|_\infty}$.[3]

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. 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. The most important structure that we get from working with 1-d subspaces, which is implied by the local order of the columns, is the following.

Lemma 5. *For each $j \in [K]$, the sets in $\mathcal{X}|_{C_j}$ are prefixes of C_j.*

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 $\mathcal{X}_i^-$. The idea is to group the classifications according to their budget occupation caused by the different column classes C_j's. To make this formal, start by covering the interval $[0, B+m]$ with intervals $\{I_\ell\}_{\ell \in L}$, where $I_0 = [0, \frac{\epsilon B}{4K})$ and $I_\ell = [\frac{\epsilon B}{4K}(1+\frac{\epsilon}{4})^{\ell-1}, \frac{\epsilon B}{4K}(1+\frac{\epsilon}{4})^{\ell})$ 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 $\mathcal{B}_{i,j}^\ell$ as the set of partial classifications $y \in \mathcal{X}|_{C_j}$ whose budget occupation $a_i(y)$ lie in the interval I_ℓ. For $v \in L^K$ define the family of classifications $\mathcal{B}_i^v = \{(y^1, y^2, \ldots, y^K) : y^j \in \mathcal{B}_{i,j}^{v_j}\}$. The $\mathcal{B}_i^v$'s then provide the desired grouping of the classifications. Note that the $\mathcal{B}_i^v$'s may include classifications not in $\mathcal{X}$ and may not include classifications in $\mathcal{X}$ which have occupation $a_i(.)$ greater than $B + m$.

Now consider a non-empty $\mathcal{B}_i^v$. Let $\underline{w}_i^v$ be the inclusion-wise smallest element in $\mathcal{B}_i^v$. Notice that such unique smallest element exists: since $\mathcal{X}|_{C_j}$ is a chain, so is $\mathcal{B}_{i,j}^{v_j}$, and hence $\underline{w}_i^v$ is the product (over j) of the smallest elements in the sets $\{\mathcal{B}_{i,j}^{v_j}\}_j$. Similarly,

[3] Notice that this ratio is well-defined since by assumption $a^t \neq 0$ for all $t \in [n]$.

let $\overline{w}_i^v$ denote the largest element in $\mathcal{B}_i^v$. Intuitively, $\underline{w}_i^v$ and $\overline{w}_i^v$ will serve as witnesses for all the sets in $\mathcal{B}_i^v$.

Finally, define the witness sets by adding the $\underline{w}_i^v$ and $\overline{w}_i^v$'s of appropriate size corresponding to meaningful $\mathcal{B}_i^v$'s: set $\mathcal{W}_i^+ = \{\underline{w}_i^v : v \in L^K, \mathcal{B}_i^v \cap \mathcal{X} \neq \emptyset, a_i(\underline{w}_i^v) \geq (1-\epsilon/2)B\}$ and $\mathcal{W}_i^- = \{\overline{w}_i^v : v \in L^K, \mathcal{B}_i^v \cap \mathcal{X} \neq \emptyset, a_i(\overline{w}_i^v) \leq (1-3\epsilon/2)B\}$.

It is not too difficult to see that, say, $\mathcal{W}_i^+$ is a witness set for $\mathcal{X}_i^+$: If $x \in \mathcal{X}_i^+$ belongs to some $\mathcal{B}_i^v$, then $\underline{w}_i^v$ belongs to $\mathcal{W}_i^+$ and is easily shown to be a witness for x. However, if x does not belong to any $\mathcal{B}_i^v$, by having too large $a_i(x)$, the idea is to find $x' \subseteq x$ which belongs to some $\mathcal{B}_i^v$ *and* to $\mathcal{X}$, and then use $\underline{w}_i^v$ as witness for x. We note that ignoring induced classifications with occupation larger than $B + m$ and ignoring B_i^v's which do not intersect $\mathcal{X}$ is very important for guaranteeing that $\mathcal{W}_i^+$ and $\mathcal{W}_i^-$ are small.

Lemma 6. *The sets $\mathcal{W}_i^+$ and $\mathcal{W}_i^-$ are witness sets for $\mathcal{X}_i^+$ and $\mathcal{X}_i^-$.*

Bounding the size of witness sets. Clearly the witness sets $\mathcal{W}_i^+$ and $\mathcal{W}_i^-$ have size at most $|L|^K$. 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 $\mathbb{R}^m$. Now we sketch the argument for an improved bound, and details are provided in the full version.

First notice that the partial classification $x(p)|_{C_j}$ is completely defined by the value pc^j. Thus, if $J \subseteq [K]$ is such that the directions $\{c^j\}_{j \in J}$ form a basis of $\mathbb{R}^m$ then knowing pc^j for all $j \in J$ completely determines the whole classification $x(p)$. Similarly, if we know that $x(p)|_{C_j} \in \mathcal{B}_i^{v_j}$ for all $j \in J$, then for each $j \notin J$ we should have fewer possible $\mathcal{B}_i^{v_j}$'s where the partial classification $x(p)|_{C_j}$ can belong to; this indicates that some of the sets $\{\mathcal{B}_i^v\}_{v \in L^K}$ do not contain any element from $\mathcal{X}$, which implies a reduced size for the witness sets.

In order to capture this idea, we focus on the space of dual vectors p and define the sets $P_j^\ell = \{p \in \mathbb{R}_+^m : x(p)|_{C_j} \in \mathcal{B}_{i,j}^\ell\}$ and $P^v = \{p \in \mathbb{R}_+^m : x(p) \in \mathcal{B}_i^v\}$. Notice that $P^v = \bigcap_j P_j^{v_j}$ and that $\mathcal{B}_i^v$ is empty iff P^v is. The main step is to show that each P_j^ℓ is a polyhedron with 'few' facets, which uses the definition of $x(p)$ and Lemma 5. We then consider the arrangement of the hyperplanes which are facet-defining for the P_j^ℓ's and conclude that the P^v's are given by unions of the cells in this arrangement; classical bounds on the number of cells in a hyperplane arrangement in $\mathbb{R}^m$ then allow us to upper bound the number of nonempty P^v's. This gives the following.

Lemma 7. *At most $(O(\frac{K}{\epsilon}\log\frac{K}{\epsilon}))^m$ of the $\mathcal{B}_i^v$'s contain an element from $\mathcal{X}$.*

This implies that both $\mathcal{W}_i^+$ and $\mathcal{W}_i^-$ have size at most $(O(\frac{K}{\epsilon}\log\frac{K}{\epsilon}))^m$, which then proves Lemma 4. Finally, applying Lemma 3 we conclude the proof of Proposition 1.

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 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 for this perturbed LP is also an approximate solution for the original one. More precisely, we obtain a set of vectors $Q \subseteq \mathbb{R}^m$ and transform each column a^t into a column $\tilde{a}^t$ which is a scaling of a vector in Q, and we let the rewards π_t remain unchanged.

The crucial observation is that the solutions of an LP are robust to slight changes in the the constraint matrix.

Lemma 8. *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 ϵ-approximate solution for* (LP) *with columns $(\pi_t, \tilde{a}^t)$ and right-hand side $(1-\epsilon)B$, then x is a $(1-2\epsilon)$-approximate solution for* (LP).

Perturbing the columns. To simplify the notation, set $\delta = \frac{\epsilon}{m+1}$; 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 $\mathbb{R}^m_+$. Using ℓ_∞ as normalization, let Q be a δ-net of the unit ℓ_∞ sphere, namely let Q be the vectors in $\{0, \delta, 2\delta, 3\delta, \ldots, 1\}^m$ which have ℓ_∞ norm 1. Note that $|Q| = (O(\frac{m}{\epsilon}))^m$.

Given a vector $a^t \in \mathbb{R}^m$ we let $\tilde{a}^t = \|a^t\|_\infty q^t$, where q^t is the vector in Q closest (in ℓ_∞) to $\frac{a^t}{\|a^t\|_\infty}$. By definition of Q, for every vector $v \in \mathbb{R}^m$ with $\|v\|_\infty = 1$ there is a vector $q \in Q$ with $\|v - q\|_\infty \leq \delta$. It then follows from positive homogeneity of norms that the $\tilde{a}^t$'s satisfy the property required in Lemma 8: $\|a^t - \tilde{a}^t\|_\infty \leq \delta\|a^t\|_\infty$.

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 1 with $K = |Q| = (O(\frac{m}{\epsilon}))^m$, we obtain the following theorem.

Theorem 2. *Fix $\epsilon \in (0,1]$ and suppose $B \geq \Omega\left(\frac{m^2}{\epsilon^3} \log \frac{m}{\epsilon}\right)$. Then 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: we use the first $2^i\epsilon n$ columns to classify columns $2^i\epsilon n + 1, \ldots, 2^{i+1}\epsilon n$. 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, δ)-OTP.

Algorithm (s, δ)-OTP. This algorithm aims at solving the program $(2s, 1)$-LP and can be described as follows: it finds an optimal dual solution (p, α) for $(s, (1-\delta))$-LP and sets $x_{\sigma(t)} = x(p)_{\sigma(t)}$ for $t = s+1, s+2, \ldots, t' \leq 2s$ such that t' is the maximum one guaranteeing $\sum_{t=s+1}^{2s} a^{\sigma(t)} x_{\sigma(t)} \leq \frac{s}{n}B$ (for all other t's it sets $x_{\sigma(t)} = 0$).

The analysis of (s, δ)-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, δ)-OTP. Again let $S = \{\sigma(1), \sigma(2), \dots, \sigma(s)\}$ be the random index set of the first s columns of the LP, let $T = \{\sigma(s+1), \sigma(s+2), \dots, \sigma(2s)\}$ and $U = S \cup T$.

Proposition 2. *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 $\frac{\delta^2 sB}{n} \geq \Omega(m \ln \frac{K}{\delta})$. Then algorithm (s, δ)-*OTP *returns a solution x satisfying $a_i^T(x) \leq B$ for all $i \in [m]$ with probability 1 and with expected value $\mathbb{E}[\sum_{\tau \in U} \pi_\tau x_\tau] \geq (1 - 3\delta)\mathbb{E}[OPT(2s)] - \mathbb{E}[OPT(s)] - \delta^2 OPT$.*

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 can be thought as acting 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 $i = 0, \dots, \log(1/\epsilon) - 1$, it runs $(\epsilon 2^i n, \sqrt{\epsilon/2^i})$-OTP over (LP) with columns $(\pi_t, \tilde{a}^t)$ 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 $\epsilon 2^i n$ columns to classify the columns $\epsilon 2^i n + 1, \dots, \epsilon 2^{i+1} n$. This relative increase in the size of the training data for each learning problem allow us to reduce the dependence of B on ϵ 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, δ)-OTP. That is, using the definition of the parameters of (s, δ)-OTP used in Robust DPA and Proposition 2, 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 3. *Fix $\epsilon \in (0, 1/100)$ and suppose that $B \geq \Omega(\frac{m^2}{\epsilon^2} \ln \frac{m}{\epsilon})$. 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 these problems 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. One possibility is to investigate how much the techniques presented here can be pushed and what are their limitations. Another possibility is to analyze the performance of the algorithm from [10] under the random permutation model.

References

1. S. Agrawal, Z. Wang, and Y. Ye. A dynamic near-optimal algorithm for online linear programming. http://arxiv.org/abs/0911.2974.
2. M. Babaioff, M. Dinitz, A. Gupta, N. Immorlica, and K. Talwar. Secretary problems: weights and discounts. In *SODA*, 2009.
3. M. Babaioff, N. Immorlica, D. Kempe, and R. Kleinberg. A knapsack secretary problem with applications. In *APPROX-RANDOM*, 2007.
4. M. Babaioff, N. Immorlica, D. Kempe, and R. Kleinberg. Online auctions and generalized secretary problems. *SIGecom Exchanges*, 7(2), 2008.
5. M. Bateni, M. Hajiaghayi, and M. Zadimoghaddam. Submodular secretary problem and extensions. In *APPROX-RANDOM*, 2010.
6. J. R. Birge and F. Louveaux. *Introduction to Stochastic Programming*. Springer Series in Operations Research and Financial Engineering. Springer, 1997.
7. A. Borodin and R. El-Yaniv. *Online computation and competitive analysis*. Cambridge University Press, 1998.
8. N. Buchbinder and J. S. Naor. Online primal-dual algorithms for covering and packing. *Mathematics of Operations Research*, 34:270–286, May 2009.
9. F. Cucker and D. X. Zhou. *Learning Theory: An Approximation Theory Viewpoint*. Cambridge University Press, 2007.
10. N. R. Devanur, K. Jain, B. Sivan, and C. A. Wilkens. Near optimal online algorithms and fast approximation algorithms for resource allocation problems. In *EC*, 2011.
11. N. R. Devenur and T. P. Hayes. The adwords problem: online keyword matching with budgeted bidders under random permutations. In *EC*, 2009.
12. L. Devroye and T. Wagner. Distribution-free performance bounds for potential function rules. *IEEE Transactions on Information Theory*, 25:601–604, 1979.
13. E. B. Dynkin. The optimum choice of the instant for stopping a Markov process. *Soviet Mathematics Doklady*, 4, 1963.
14. J. Feldman, M. Henzinger, N. Korula, V. S. Mirrokni, and C. Stein. Online stochastic packing applied to display ad allocation. In *ESA*, 2010.
15. J. P. Gilbert and F. Mosteller. Recognizing the Maximum of a Sequence. *Journal of the American Statistical Association*, 61(313):35–73, 1966.
16. G. Goel and A. Mehta. Online budgeted matching in random input models with applications to adwords. In *SODA*, 2008.
17. S. Im and Y. Wang. Secretary problems: Laminar matroid and interval scheduling. In *SODA*, 2011.
18. R. M. Karp, U. V. Vazirani, and V. V. Vazirani. An optimal algorithm for on-line bipartite matching. In *STOC*, 1990.
19. C. Kenyon. Best-fit bin-packing with random order. In *SODA*, 1996.
20. R. Kleinberg. A multiple-choice secretary algorithm with applications to online auctions. In *SODA*, 2005.
21. S. Kutin and P. Niyogi. Almost-everywhere algorithmic stability and generalization error. In *Uncertainty in Artificial Intelligence*, pages 275–282, 2002.
22. M. Molinaro and R. Ravi. Geometry of online packing linear programs. http://arxiv.org/abs/1204.5810.
23. J. A. Soto. Matroid secretary problem in the random assignment model. In *SODA*, 2011.