Maximizing Profit with Convex Costs in the Random-order Model*

Anupam Gupta[1]
Carnegie Mellon University, Pittsburgh, USA
anupamg@cs.cmu.edu

Ruta Mehta[2]
University of Illinois Urbana-Champaign, Champaign, USA
rutameht@illinois.edu

Marco Molinaro[3]
PUC-Rio, Rio de Janeiro, Brazil
mmolinaro@inf.puc-rio.br

Abstract

Suppose a set of requests arrives online: each request gives some value v_i if accepted, but requires using some amount of each of d resources. Our cost is a convex function of the vector of total utilization of these d resources. Which requests should be accept to maximize our profit, i.e., the sum of values of the accepted demands, minus the convex cost?

We consider this problem in the random-order a.k.a. secretary model, and show an $O(d)$-competitive algorithm for the case where the convex cost function is also *supermodular*. If the set of accepted demands must also be independent in a given matroid, we give an $O(d^3\alpha)$-competitive algorithm for the supermodular case, and an improved $O(d^2\alpha)$ if the convex cost function is also separable. Here α is the competitive ratio of the best algorithm for the submodular secretary problem. These extend and improve previous results known for this problem. Our techniques are simple but use powerful ideas from convex duality, which give clean interpretations of existing work, and allow us to give the extensions and improvements.

2012 ACM Subject Classification Theory of computation → Online algorithms

Keywords and phrases Online algorithms, secretary problem, random order, convex duality

Digital Object Identifier 10.4230/LIPIcs.ICALP.2018.71

Related Version A full version of the paper is available at `https://arxiv.org/abs/1804.08172`.

1 Introduction

The problem we consider is a basic convex optimization problem in the online setting: n items appear one-by-one. Each item/element e has a *d-dimensional size* $s(e) \in \mathbb{R}^d_+$ and a *value* $v(e) \in \mathbb{R}_+$, which are both revealed to us when the item arrives.We must either accept

* This work was done in part while the authors were visiting the Simons Institute for the Theory of Computing.

[1] Computer Science Department, Carnegie Mellon University, Pittsburgh, USA. Supported in part by NSF awards CCF-1536002, CCF-1540541, and CCF-1617790.
[2] Supported in part by NSF award CCF-1750436, and the Simons Institute for the Theory of Computing.
[3] Supported in part by CNPq grants Universal #431480/2016-8 and Bolsa de Produtividade em Pesquisa #310516/2017-0, and a Microsoft Research Fellowship at the Simons Institute for the Theory of Computing.

45th International Colloquium on Automata, Languages, and Programming (ICALP 2018).
Editors: Ioannis Chatzigiannakis, Christos Kaklamanis, Dániel Marx, and Donald Sannella;
Article No. 71; pp. 71:1–71:14

Leibniz International Proceedings in Informatics
Schloss Dagstuhl – Leibniz-Zentrum für Informatik, Dagstuhl Publishing, Germany

or reject an item when it arrives, before seeing the future items. If we accept a certain subset $A \subseteq [n]$ of the items, we get their total value $v(A) := \sum_{e \in A} v_e$, but incur a production cost $g(s(A)) := g(\sum_{e \in A} s(e))$, where $g : \mathbb{R}^d_+ \to \mathbb{R}_+$ is a non-decreasing *convex cost* function with $g(0) = 0$. Optionally, we may also be given a downwards-closed family of subsets $\mathscr{F} \subseteq 2^{[n]}$, and now the accepted set of elements A must lie in $\mathscr{F}$. More formally, we want to solve

$$\max_{A \in \mathscr{F}} \text{ profit } \pi(A) := \big[v(A) - g(s(A))\big]. \tag{1.1}$$

This question arises, e.g., when we are selling some service that depends on d commodities, where the value is the amount of money customer e is willing to pay for the service, and the size vector $s(e)$ is the amount of resources she will require. The cost function $g(\cdot)$ captures our operating expenses; its convexity models *diseconomies of scale* that arise when dealing with scarce commodities. In particular, it can capture d-dimensional knapsack constraints, by setting $g(z) = 0$ until the knapsack size, and ∞ afterwards. When the cost function is linear $g(z) = \langle a, z \rangle$, we want to pick a max-weight subset from $\mathscr{F}$ using item weights $v(e) - \langle a, s(e) \rangle$, which is tractable/approximable for $\mathscr{F}$ being a matroid, p-system, etc.

Blum et al. [6] defined this problem in the adversarial model, and gave posted-price algorithms for "low-degree" *separable* cost functions g, that is, of the form $g(z) = \sum_{i=1}^{d} g_i(z_i)$ for 1-dimensional functions g_i's. This result was tightened by Huang and Kim [16], still for separable functions with additonal growth control. More recently, Azar et al. [3] studied this problem for more general *supermodular* non-separable convex functions g (see also [9]). A differentiable function g is supermodular if for any vectors $x \leq x'$ we have $\nabla g(x) \leq \nabla g(x')$. Equivalently, if g is twice-differentiable, it is supermodular if $\frac{\partial^2 g}{\partial x_i \partial x_j} \geq 0$ for all $i \neq j$, i.e., increasing the consumption of a resource cannot decrease the marginal cost for another. However, to handle the worst-case ordering, Azar et al. also require the cost functions to have essentially low-degree.

Can we do better by going beyond the worst-case model? In this paper, we focus on the random-order or "secretary" setting, where the set of items is fixed by an adversary but they arrive in random order. In the single-dimensional case $d = 1$, it is easy to see that a solution that learns a "good" threshold λ and picks all further items with density $v(e)/s(e)$ at least λ essentially gives a constant approximation, much like in the secretary and knapsack secretary problems [13, 4]. The multi-dimensional case is much more challenging. This was studied by Barman et al. [5], again assuming a separable cost function $g(z) = \sum_{i=1}^{d} g_i(z_i)$. They give an $O(d)$-competitive algorithm for the unconstrained case, and an $O(d^5\alpha)$-competitive algorithm for the problem with a downward closed constraint set $\mathscr{F}$, where α is the competitive ratio for the $\mathscr{F}$-secretary problem. Their main idea is to perform a clever decomposition of the value of each item into "subvalues" $v_i(e)$ for each of the coordinate cost functions g_i's; this effectively decomposes the problem into d 1-dimension problems with values v_i's and costs g_i's. Unfortunately, since their solution explicitly relies on the decomposability of the cost function, it is unclear how to extend it to general supermodular functions. We note that when the cost function is supermodular, the profit function is a *submodular* set function (Section 2.1). However, the profit can take *negative values*, and then existing algorithms for submodular maximization break down.

Our work is then motivated by trying to better understand the multi-dimensional nature of this problem, and provide a more principled algorithmic approach.

1.1 Our Results

We use techniques from convex duality to re-interpret, simplify, and improve the existing results. First, we obtain the first approximation for non-separable supermodular cost functions. (We omit some mild regularity conditions for brevity; see Section 3 for full details.)

▶ **Theorem 1** (Unconstrained & Supermodular). *For the unconstrained problem with* supermodular *convex cost functions g, we give an $O(d)$-competitive randomized algorithm in the random-order model.*

This result generalizes the $O(d)$-approximation of Barman et al. [5] to the non-separable case. The factor d seems unavoidable, since our problem inherits the (offline) $\Omega(d^{1-\varepsilon})$ hardness of the d-dimensional knapsack, assuming $NP \neq ZPP$ [7].

Next, we consider the constrained case. For simplicity, we focus on the most interesting case where $\mathscr{F}$ is a matroid constraint; more general results can be obtained from the results and techniques in Section 5.

▶ **Theorem 2** (Constrained & Separable). *For the constrained problem with $\mathscr{F}$ being a matroid constraint, and the cost function g being* separable*, we get an $O(d^2 \log\log \mathit{rank})$-competitive randomized algorithm in the random-order model.*

This improves by a factor of d^3 the $O(d^5 \log\log \mathrm{rank})$-approximation given by [5]. Finally, we give a general reduction that takes an algorithm for *separable* functions and produces an algorithm for *supermodular* functions, both with respect to a matroid constraint, implying:

▶ **Theorem 3** (Constrained & Supermodular). *For the constrained problem with $\mathscr{F}$ being a matroid constraint, and the cost function g being* supermodular*, we get an $O(d^3 \log\log \mathit{rank})$-competitive randomized algorithm in the random-order model.*

On conceptual contributions are in bringing techniques from convex duality to obtain, in a principled way, *threshold-based* algorithms for non-linear secretary problems. Since this is a classical and heavily used algorithmic strategy for secretary problems [13, 4, 18, 2, 20] we hope that the perspectives used here will find use in other contexts.

1.2 Other Related Work

There is a vast literature on secretary problems [13]. Closest to our setting, Agrawal and Devanur study an online convex optimization problem in the random order model, and give a powerful result showing strong regret bounds in this setting [1]. They extend this result to give algorithms for online packing LPs with "large" right-hand sides. However, it is unclear how to use their algorithm to obtain results in our setting. Other algorithms solving packing LPs with large right-hand sides appear in [2, 8, 20, 17, 14, 10].

Feldman and Zenklusen [12] show how to transform any algorithm for (linear) matroid secretary into one for *submodular* matroid secretary. They give an $O(\log\log \mathrm{rank})$-algorithm for the latter, based on results of [19, 11]. All these algorithms critically assume the submodular function is non-negative everywhere, which is not the case for us, since picking too large a set may cause the profit function to go negative. Indeed, one technical contribution is a procedure for making the profit function non-negative while preserving submodularity (Section 4.1), which allows us to use these results as part of our solution.

1.3 Structure of the paper

Section 3 develops the convex duality perspective used in the paper for the offline version of the unconstrained case, hopefully in an manner accessible to non-experts. Section 4 gives the small changes required to extend this to the constrained case. Section 5 shows how transform these into online algorithms. Section 6 shows how to convert an algorithm for separable functions into one for supermodular functions, both subject to matroid constraints. To improve the presentation, we make throughout mild assumptions, which are discharged in the full version of the paper.

2 Preliminaries

Elements from a universe U of size n are presented in random order. Each element e has value $v(e) \in \mathbb{R}_+$ and size $s(e) \in \mathbb{R}^d_+$. We are given a convex cost function $g : \mathbb{R}^d_+ \to \mathbb{R}_+$. On seeing each element we must either accept or discard it. A downwards-closed collection $\mathscr{F} \subseteq 2^U$ of feasible sets is also given. When $\mathscr{F} = 2^U$, we call it the *unconstrained* problem. The goal is to pick a subset $A \in \mathscr{F}$ to maximize the *profit*

$$\pi(A) := \sum_{e \in A} v(e) - g\big(\sum_{e \in A} s(e)\big). \tag{2.2}$$

We often use vectors in $\{0,1\}^n$ to denote subsets of U; χ_A denotes the indicator vector for set A. Hence, $\mathscr{F} \subseteq \{0,1\}^n$ is a down-ideal on the Boolean lattice, and we can succinctly write our problem as

$$\max_{x \in \mathscr{F}} \quad \pi(x) := \langle v, x \rangle - g(Sx), \tag{2.3}$$

where columns of $S \in \mathbb{R}^{d \times n}$ are the item sizes. Let opt denote the optimal value. For a subset $A \subseteq U$, $v(A)$ and $s(A)$ denote $\sum_{e \in A} v(e) = \langle v, \chi_A \rangle$ and $\sum_{e \in A} s(e) = S\chi_A$ respectively.

▶ **Definition 4** (Exceptional). Item $e \in U$ is *exceptional* if $\arg\max_{\theta \in [0,1]} \{\theta\, v(e) - g(\theta\, s(e))\} \in (0,1)$.

▶ **Definition 5** (Marginal Function). Given $g : \mathbb{R}^d \to \mathbb{R}$, define the i^{th} *marginal function* $g_i : \mathbb{R} \to \mathbb{R}$ as $g_i(x) := g(x\mathbf{e}_i)$, where $\mathbf{e}_i$ is the i^{th} standard unit vector.

▶ **Definition 6** (Convex Dual). For any function $g : \mathbb{R}^d \to \mathbb{R}$, its *convex dual* is the function $g^\star : \mathbb{R}^d \to \mathbb{R}$ given by $g^\star(y) := \sup_x \big[\langle y, x \rangle - g(x)\big]$.

2.1 Supermodular Functions

While supermodular functions defined over the Boolean lattice are widely considered, one can define supermodularity for all real-valued functions.

▶ **Definition 7** (Supermodular). Let $X \subseteq \mathbb{R}^d$ be a lattice. A function $f : X \to \mathbb{R}$ is *supermodular* if for all $x, y \in X$, $f(x) + f(y) \le f(x \wedge y) + f(x \vee y)$, where $x \wedge y$ and $x \vee y$ are the component-wise minimum and maximum operations.

This corresponds to the usual definition of (discrete) supermodularity when $X = \{0,1\}^d$. For proof of the lemma below and other equivalent definitions, see, e.g., [21].

▶ **Lemma 8** (Supermodularity and Gradients). *A convex function $f : \mathbb{R}^d_+ \to \mathbb{R}$ is supermodular if and only if any of the following are true.*

- *∇f is increasing in each coordinate, if f is differentiable.*
- *$\frac{\partial^2 f(x)}{\partial x_i \partial x_j} \ge 0$ for all i, j, if f is twice-differentiable.*

▶ **Lemma 9** (Superadditivity). *If $f : \mathbb{R}^d_+ \to \mathbb{R}$ is differentiable, convex, and supermodular, then for $x, x', y \in \mathbb{R}^d_+$ such that $x' \le x$, $f(x' + y) - f(x') \le f(x + y) - f(x)$. In particular, if $f(0) = 0$, setting $x' = 0$ gives $f(x) + f(y) \le f(x + y)$.*

▶ **Corollary 10** (Subadditivity of profit). *The profit function π is subadditive.*

The next fact shows that the cost g is also supermodular when seen in a discrete way.

▶ **Fact 11** (Continuous vs. Discrete Supermodularity). *Given a convex supermodular function $g : \mathbb{R}^d \to \mathbb{R}$ and n items with sizes $s_1, \ldots, s_n \in \mathbb{R}^d_+$, define the function $h : \{0,1\}^n \to \mathbb{R}$ as $h(v) = g(\sum_i s_i v_i) = g(Sv)$. Then $h(\cdot)$ is a (discrete) supermodular function.*

3 The Offline Unconstrained Problem

We first present an offline algorithm for supermodular functions in the **unconstrained** case (where $\mathscr{F} = \{0,1\}^n$). We focus on the main techniques and defer some technicalities and all computational aspects for now. Just for this section, we assume item sizes are "infinitesimal". We make the following assumptions on the cost function g and the elements.

▶ **Assumption 12.** *We assume that cost function g is non-negative, strictly convex, closed, and differentiable. We assume $g(0) = 0$, g is supermodular, and that gradients of g go to ∞ along every positive direction. We assume elements are in general positionand that there are no exceptional items. We also assume that every individual item has profit at most $M := \mathsf{opt}/\eta d$ for $\eta \geq 10^4$ (see the full version to remove these assumptions).*

Classifiers. The offline algorithm will be based on *linear classifiers*, where a set of weights is used to aggregates the multidimensional size of an item into a scalar, and the algorithm picks all items that have high-enough value/aggregated-size ratio.

▶ **Definition 13** (Classifiers and Occupancy)**.** Given a vector $\lambda \in \mathbb{R}^d_+$ (a "classifier"), define the set of items picked by λ as $U_\lambda := \{e \in U \mid v(e) \geq \langle \lambda, s(e)\rangle\}$. Let $\mathsf{occ}_\lambda := \sum_{e:v(e)\geq\langle\lambda,s(e)\rangle} s(e)$ denote the multidimensional occupancy induced by choosing items in U_λ.

To understand the importance of classifier-based solutions it is instructive to consider the problem with single-dimensional size. A little thought shows that an optimal solution is to pick items in decreasing order of value density $v(e)/s(e)$. Adding these items causes the total occupancy – and hence the incurred cost – to increase, so we stop when the value density of the current item becomes smaller than the derivative of the cost function at the current utilization. That is, we find a density threshold λ such that $g'(\textit{total size of items having } v(e) \geq \lambda\, s(e)) \approx \lambda$, and take all these high-density items. Thus, the optimal solution is one based on the classifier λ.

To see that this holds in the multi-dimensional case, express g in terms of linearizations

$$g(z) = \max_{\lambda \in \mathbb{R}^d_+} (\langle \lambda, z\rangle - g^\star(\lambda)), \tag{3.4}$$

where $g^\star$ is its Fenchel dual. (Note we are maximizing over positive, but this is WLOG.) Then our unconstrained problem (2.2) becomes a minimax problem:

$$\max_{x\in\{0,1\}^n} \min_{\lambda\in\mathbb{R}^d_+} \left[\langle v, x\rangle - \Big(\langle \lambda, Sx\rangle - g^\star(\lambda)\Big)\right].$$

Consider an optimal pair (x^*, λ^*); i.e., a pair that is a saddle-point solution, so neither x^* nor λ^* can be improved keeping the other one fixed. This saddle-point optimality implies:

(a) Since $\lambda^* = \operatorname{argmax}_{\lambda\in\mathbb{R}^d_+} (\langle \lambda, Sx^*\rangle - g^\star(\lambda))$, it is the right linearization of g at Sx^* and thus $\lambda^* = \nabla g(Sx^*)$ (see [15, Theorem E.1.4.1] and [15, Corollary E.1.3.6]).

(b) x^* is such that $x^*_i = 1$ if $v_i > \langle \lambda^*, S^i\rangle$ and $x^*_i = 0$ if $v_i < \langle \lambda^*, S^i\rangle$, with S^i being the i^{th} column of S and the size of the i^{th} item.

From part (b) we see the optimal solution x^* is essentially the one picked by the classifier λ^* (ignoring coordinates with "0 marginal value" $v_i = \langle \lambda^*, S^i\rangle$). The converse also holds.

▶ **Claim 14.** *For a classifier $\lambda \in \mathbb{R}^d_+$, let x be the items picked by it. If we have $\lambda = \nabla g(Sx) \stackrel{def}{=} \nabla g(\mathsf{occ}_\lambda)$, then x is an optimal solution.*

Proof. For any solution x',

$$\pi(x') = \langle v, x' \rangle - g(Sx') \le \langle v, x' \rangle - \langle \lambda, Sx' \rangle + g^\star(\lambda)$$
$$\le \langle v, x \rangle - \langle \lambda, Sx \rangle + g^\star(\lambda) \overset{(\lambda = \nabla g(Sx))}{=} \langle v, x \rangle - g(Sx) = \pi(x),$$

where the second inequality holds since, by definition, x maximizes $\langle v, x \rangle - \langle \lambda, Sx \rangle$. ◀

Restricting the Set of Classifiers. The existence of such good classifiers is not enough, since we need to *find* them online. This is difficult not only because of d degrees of freedom and no control over the magnitude of the values/sizes (to be exploited in concentration inequalities), but also because picking too few or too many items could lead to low profits.

So we restrict the set of candidate classifiers to be a *monotone*[4] *1-dimensional* curve $\mathcal{C} \subseteq \mathbb{R}^d_+$, satisfying additional properties given below. The main motivation is that it imposes a total ordering on the set of items picked by the classifiers: given $\lambda \le \mu$ on such a curve $\mathcal{C}$, the sets of items picked satisfy the inclusion $U_\lambda \supseteq U_\mu$. This allows us to select a "minimally good" classifier in $\mathcal{C}$ in a robust way, avoiding classifiers that select too many items.

To design the curve $\mathcal{C}$ so it contains a classifier with profit $\approx \frac{\mathsf{opt}}{d}$, we relax the condition $\nabla g(\mathsf{occ}_\lambda) = \lambda$ from Claim 14 (too much to ask) and require the existence of $\lambda \in \mathcal{C}$ satisfying:

(P1) (don't pick too many items) $\nabla g(\mathsf{occ}_\lambda) \le \lambda$.
(P2) (partial gradient equality) There is a coordinate i^* where $(\nabla g(\mathsf{occ}_\lambda))_{i^*} = \lambda_{i^*}$.
(P3) (balanced curve) $g_i^\star(\lambda_i) = g_j^\star(\lambda_j) \quad \forall i, j \in [d]$.

Property (P1) enforces half of the equality in Claim 14, and (P2) guarantees that equality holds for *some* coordinate. Now for property (P3). Since $\lambda \ne \nabla g(\mathsf{occ}_\lambda)$ the optimality proof of Claim 14 does not go though, since $g(\mathsf{occ}_\lambda) \ne \langle \lambda, \mathsf{occ}_\lambda \rangle - g^\star(\lambda)$. As we prove later, the difference between these terms can be at most $g^\star(\lambda) \le \sum_i g_i^\star(\lambda_i)$. Property (P3) is used to control this sum, by charging it to the coordinate i^* where we know we have "the right linearization" (by property (P2)). Reinterpreting the construction of [5] in our setting, we then define $\mathcal{C}$ as any monotone curve where every $\lambda \in \mathcal{C}$ satisfies (P3).

▶ **Lemma 15.** *The curve $\mathcal{C}$ exists and contains a λ satisfying properties (P1)-(P3).*

Proof. We first show existence, that is, the set $\{\lambda \in \mathbb{R}^d_+ \mid g_i^\star(\lambda_i) = g_j^\star(\lambda_j) \ \ \forall i, j\}$ contains a monotone curve. Notice that this set is the union of the box $\{\lambda \in \mathbb{R}^d_+ \mid g_i^\star(\lambda_i) = 0 \ \ \forall i\} = \prod_i [0, g_i'(0)]$ (range of slopes where we can swivel around $g_i(0) = 0$) and a monotone curve $\{\lambda(\tau) \mid \tau > 0\}$, where $\lambda(\tau)$ is the unique vector satisfying $g_i^\star(\lambda_i(\tau)) = \tau$; uniqueness follows from the fact $g_i^\star$ stays at value zero in the interval $[0, g_i'(0)]$, but after that is strictly increasing due to its convexity, and monotonicity of this curve also follows from monotonicity of the $g_i^\star$'s. Thus, $\mathcal{C}$ is this curve plus any monotone curve extending it to the origin.

To see that $\mathcal{C}$ satisfies properties (P1) and (P2), we note that since the $g_i^\star$'s are increasing and not identically 0, $\mathcal{C}$ is unbounded in all coordinates. Thus, a sufficiently large $\lambda \in \mathcal{C}$ satisfies (P1), and we can start with such λ and move down the curve (decreasing in each coordinate) until we obtain $\lambda' \in \mathcal{C}$ with $\lambda' = \nabla g(\mathsf{occ}_{\lambda'})$, since the g has increasing gradients. (The equality in this final step uses the assumption that item sizes are infinitesimal, which we made for simplicity in this section). ◀

Making the above discussion formal, we show that $\mathcal{C}$ has a high-value classifier. Recall that U_λ is the set of items picked by λ (Definition 13).

[4] A curve $\mathcal{C}$ is *monotone* if for every pair $\lambda, \lambda' \in \mathcal{C}$, one is coordinate-wise smaller than the other.

▶ **Theorem 16.** *Given Assumption 12, let λ^* be a classifier in $\mathcal{C}$ satisfying properties (P1)-(P3). Then for all $x' \in [0,1]^n$ we have $\pi(U_{\lambda^*}) \geq \frac{1}{d+1} \cdot \pi(x')$.*

Proof. Let $x^* = \chi_{U^*_\lambda}$ be the solution picked by the classifier λ^*, and note that $\mathsf{occ}_{\lambda^*} = Sx^*$. Let $L(y,\mu) := \langle v, y\rangle - [\langle \mu, Sy\rangle - g^\star(\mu)]$ be the linearization of $\pi(y)$ at some slope μ. From (3.4) we know $g(y) \geq L(y,\mu)$ for all $\mu \geq 0$. Since x^* is optimal for the linearization $L(y,\lambda^*)$ (because $x^*_i = 1$ iff $v_i - \langle \lambda^*, S^i\rangle \geq 0$), we have

$$L(x^*,\lambda^*) \geq L(x',\lambda^*) \geq \pi(x') \qquad \text{for all } x' \in [0,1]^n. \tag{3.5}$$

Now we relate the true profit $\pi(x^*)$ to this linearized value. Observe that

$$\begin{aligned}\pi(x^*) = \langle v, x^*\rangle - g(Sx^*) &= \langle v, x^*\rangle - [\langle \nabla g(Sx^*), Sx^*\rangle - g^\star(\nabla g(Sx^*))] \\ &\geq \underbrace{\langle v, x^*\rangle - \langle \lambda^*, Sx^*\rangle}_{\geq 0} + \underbrace{g^\star(\nabla g(Sx^*))}_{\geq 0},\end{aligned} \tag{3.6}$$

where the inequality uses that $\lambda^* \geq \nabla g(Sx^*)$ by property (P1) and $Sx^* \geq 0$. The first term is non-negative because we only pick items for which $v_i - \langle \lambda, S^i\rangle \geq 0$. The second term is non-negative because $g(0) = 0$. We can now prove three lemmas that imply the theorem.

▶ **Lemma 17.** *For any $x' \in [0,1]^n$, $\pi(x^*) \geq L(x^*,\lambda^*) - g^\star(\lambda^*) \geq \pi(x') - g^\star(\lambda^*)$.*

Proof. Drop the second term from (3.6), then use the definition of $L(\cdot,\cdot)$ and (3.5). ◀

▶ **Lemma 18.** $g^\star(\lambda^*) \leq d \cdot g^\star_{i^*}(\lambda^*_{i^*})$.

Proof. Using the superadditivity of g, one can show $g^\star(\lambda^*) \leq \sum_i g^\star_i(\lambda^*_i)$. Now from property (P3) of the classifier λ^*, all the terms in the sum are equal. ◀

▶ **Lemma 19.** $\pi(x^*) \geq g^\star_{i^*}(\lambda^*_{i^*})$.

Proof. We claim that $g^\star(\nabla g(Sx^*)) \geq g^\star_{i^*}(\lambda^*_{i^*})$; plugging this into (3.6) proves the lemma. For the claim, define $\lambda' = \nabla g(Sx^*)$. By Property (P2), $\lambda'_{i^*} = \lambda^*_{i^*}$, so we want to show $g^\star(\lambda') \geq g^\star_{i^*}(\lambda'_{i^*}) = g^\star(\lambda'_{i^*}\mathbf{e}_{i^*})$. This follows because $g^\star$ is monotone. ◀

This completes the proof of Theorem 16. ◀

4 The Offline Constrained Case

Having built up tools and intuition in the unconstrained case, we turn to the case where there is a downwards-closed constraint $\mathscr{F} \subseteq \{0,1\}^n$, and the goal is to maximize the profit subject to $x \in \mathscr{F}$. We again work with Assumption 12, but do not assume anything about items sizes. We discuss computational aspects at the end of this section.

The general idea is again to use classifiers $\lambda \in \mathbb{R}^d_+$, and only consider items in U_λ, namely those with "high-enough" value $v_i \geq \langle \lambda, S^i\rangle$. However, because of the constraints $\mathscr{F}$ we may no longer be able to pick all these items. Thus, we need to consider the most profitable solution from $\mathscr{F}$ in this filtered feasible set U_λ (whose quality is less clear how to analyze).

Again we restrict to the 1-dimensional curve $\mathcal{C}$ defined in the previous section; however, it only satisfies slightly modified versions of properties (P1)-(P2), since we do not assume the item sizes to be infinitesimal anymore. To make this precise, define the "open" set $U^\circ_\lambda := \{e \in U \mid v(e) > \langle \lambda, s(e)\rangle\}$; note the strict inequality. Under the assumption of items being in general position, there is at most one "threshold" item with $v_i = \langle \lambda, S^i\rangle$, i.e., $|U_\lambda \setminus U^\circ_\lambda| \leq 1$. Now a "good" classifier is one that satisfies the following:

(P1') For all binary x with $\text{support}(x) \subseteq U^\circ_\lambda$ and $x \in \mathscr{F}$, $\nabla g(Sx) \le \lambda$.

(P2') There exists a binary x^{occ} with $\text{support}(x^{occ}) \subseteq U_\lambda$ and $x^{occ} \in \mathscr{F}$, and index i^* such that $(\nabla g(Sx^{occ}))_{i^*} \ge \lambda_{i^*}$. (Note that if $\text{support}(x^{occ}) \subseteq U^\circ_\lambda$, then by property (P1') the above inequality holds at equality; else x^{occ} contains the unique element in $U_\lambda \setminus U^\circ_\lambda$.)

(P3') This is the same as before: $g^\star_i(\lambda_i) = g^\star_j(\lambda_j) \quad \forall i, j \in [d]$.

The arguments of Lemma 15 show the following.

▶ **Lemma 20.** *Given Assumption 12, the curve $\mathcal{C}$ defined in the previous section contains a λ satisfying properties (P1')-(P3').*

Next, we show that for a good classifier $\lambda \in \mathcal{C}$, the maximum profit solution from $\mathscr{F}$ contained within U°_λ essentially gives an $O(1/d)$-approximation.

▶ **Theorem 21** (Offline Approach)**.** *Suppose Assumption 12 holds. Let λ^* be a classifier in $\mathcal{C}$ satisfying properties (P1')–(P3'). Then the better of the two solutions: (a) the maximum profit solution in $\mathscr{F}$ containing elements only from $U^\circ_{\lambda^*}$, and (b) the optimal single element in U_{λ^*}, has profit at least $\pi(x')/(2d+1)$ for any vector $x' \in \text{Conv}(\mathscr{F}) \subseteq [0,1]^n$.*

Proof. The idea is to follow the development in Theorem 16. There same solution x^* satisfied the value lower bounds of Lemmas 17 and 19; to satisfy the first lemma, we needed the solution to be optimal for the linearization of π using "slope" λ^*; to satisfy the second, we needed to satisfy (P2). Here, we construct two solutions in $\mathscr{F}$ intersect U_{λ^*} to satisfy these lemmas separately:

$x^{lin} := \text{argmax}\{\langle v, y\rangle - \langle \lambda^*, Sy\rangle \mid y \subseteq U^\circ_{\lambda^*}, y \in \mathscr{F}\}$

$x^{occ} :=$ the solution promised by property (P2').

Since property (P1') and (P3') holds for x^{lin}, Lemmas 17 and 18 hold essentially unchanged, and thus for any vector $x' \in \text{Conv}(\mathscr{F})$ we have

$$\pi(x^{lin}) \ge \pi(x') - d \cdot g^\star_{i^*}(\lambda^*_{i^*}). \tag{4.7}$$

The solution x^{occ} may not belong to the set $U^\circ_{\lambda^*}$, since it may contain the threshold item $e^\circ = \langle \lambda^*, s(e^\circ)\rangle$, if it exists (let $x^\circ = \chi_{\{e^\circ\}}$ be its characteristic vector, all 0's vector if does not exists). Let $x^{rest} = x^{occ} - x^\circ$.

▶ **Lemma 22.** *These solutions satisfy $\pi(x^{rest}) + \pi(x^\circ) \ge g^\star_{i^*}(\lambda^*_{i^*})$.*

Proof. Property (P1') gives $\nabla g(Sx^{rest}) \le \lambda^*$, and Property (P2') implies $\nabla g(S(x^{rest}+x^\circ)) = \nabla g(Sx^{occ})$ is at least λ^* at some coordinate i^*. Since g is convex and differentiable, the gradients are continuous [15, Remark D.6.2.6], so there is $\delta \in [0,1]$ where the vector $\widehat{x} := x^{rest} + \delta x^\circ$ satisfies $\nabla g(S\widehat{x}) \le \lambda^*$ and $\nabla g(S\widehat{x})_{i^*} = \lambda^*_{i^*}$ for some coordinate i^*. Due to these properties, the proof of Lemma 19 holds for $\widehat{x}$ and shows $\pi(\widehat{x}) \ge g^\star_{i^*}(\lambda^*_{i^*})$.

The assumption of no exceptional items gives $\pi(\delta x^\circ) \le \pi(x^\circ)$. From subadditivity of profit π, $g^\star_{i^*}(\lambda^*_{i^*}) \le \pi(\widehat{x}) \le \pi(x^{rest}) + \pi(\delta x^\circ) \le \pi(x^{rest}) + \pi(x^\circ)$. This concludes the proof. ◀

Combining Lemma 22 with inequality (4.7) we have $\pi(x') \le \pi(x^{lin}) + d\,\pi(x^{rest}) + d\,\pi(x^\circ)$ for any $x' \in \mathscr{F}$. Since x^{lin}, x^{rest} are feasible for (a) in the theorem statement, and x° is feasible for (b), the best of them gives a $(2d+1)$-approximation. This proves Theorem 21. ◀

Picking the most profitable singleton is trivial offline, and well-approximable online by the secretary algorithm [13]. Moreover, we need to approximately optimize the *submodular* function π (Fact 11) over $\mathscr{F}|_{U^\circ_{\lambda^*}}$ (i.e., the sets in $\mathscr{F}$ with only elements of $U^\circ_{\lambda^*}$). For several constraint structures (e.g., matroids, p-systems), there are known algorithms for approximately optimizing *non-negative* (and sometimes also monotone) submodular functions. Unfortunately, our profit function π may take negative values, so we cannot directly use these algorithms. Simply considering the truncated function $\max\{\pi(z), 0\}$ does not work because it may be non-submodular. In the next section, when g is *separable*, we introduce a way of making our profit function non-negative everywhere, while maintaining submodularity and preserving the values at the region of interest $\mathscr{F}|_{U^\circ_{\lambda^*}}$.

4.1 Making the Profit Function π Non-negative

We first show that π already satisfies the desired properties over the sets in $\mathscr{F}|_{U^\circ_{\lambda^*}}$.

▶ **Lemma 23.** *The profit function π is non-negative monotone over $\mathscr{F}|_{U^\circ_{\lambda^*}}$.*

Proof. Since $\pi(\emptyset) = 0$ it suffices to show monotonicity. Consider $x \in \mathscr{F}|_{U^\circ_{\lambda^*}}$ and let χ_e be the indicator os an item in x. Comparing the costs with and without e we have

$$g(Sx) \overset{\text{(convexity)}}{\leq} g(S(x-\chi_e)) + \langle \nabla g(Sx), S\chi_e \rangle \overset{\text{(Property (P1'))}}{\leq} g(S(x-\chi_e)) + \langle \lambda^*, s(e) \rangle.$$

Since $x \in U^\circ_{\lambda^*}$, we have $v(e) > \langle \lambda^*, s(e) \rangle$ and thus $\pi(x) > \pi(x - \chi_e)$, i.e., monotonicity. ◀

However, to run algorithms that approximately optimize π over $\mathscr{F}|_{U^\circ_{\lambda^*}}$ in a black-box fashion, non-negativity over the feasible sets $\mathscr{F}|_{U^\circ_{\lambda^*}}$ is not enough, even if the algorithm only probes π over these sets, since their *proof of correctness* may require this property outside of feasible sets. Thus, we need to modify π to ensure non-negativity outside of $\mathscr{F}|_{U^\circ_{\lambda^*}}$.

For that, the idea is to truncate the gradient of the cost g so $\nabla g(Sx)$ becomes at most λ^* for all subsets $x \subseteq U^\circ_{\lambda^*}$ (i.e., so Property (P1') holds for all subsets); this was the crucial element for the monotonicity (and hence non-negativity) proof above. Notice that since Property (P1') guarantees already $\nabla g(Sx) \leq \lambda^*$ for all $x \in \mathscr{F}|_{U^\circ_{\lambda^*}}$, this does not change the value of π over these points. The proof of the lemma is deferred to the full version.

▶ **Lemma 24.** *If g is separable, there is a submodular function π^+ satisfying the following:*

(i) *π^+ is non-negative and monotone over all subsets of $U^\circ_{\lambda^*}$, and*

(ii) *$\pi^+(x) = \pi(x)$ for every $x \in \mathscr{F}|_{U^\circ_{\lambda^*}}$.*

4.2 The Offline Algorithm: Wrap-up

Using this non-negativization procedure, we get an $O(d)$-approximation *offline* algorithm for constrained profit maximization for *separable* cost functions g; this is an offline analog of Theorem 2. For the unconstrained case, Lemma 23 implies that the profit function π it itself monotone, so we get an $O(d)$-approximation offline algorithm for the *supermodular* case. In the next section we show how to convert these algorithms into online algorithms.

One issue we have not discussed is the computational cost of finding λ^* satisfying (P1')–(P3'). In the full version of the paper, we show that for any $\varepsilon > 0$ we can efficiently find a λ^* satisfying (P1'), (P2'), and a slightly weaker condition: $|g^\star_i(\lambda^*_i) - g^\star_j(\lambda^*_j)| \leq 2\varepsilon$ for all $i, j \in [d]$. Using this condition in Theorem 21 means we get a profit of at least $\frac{\mathsf{opt}-2d\varepsilon}{2d+1} \geq [\mathsf{opt}/(2d+1)] - \varepsilon$; the running time depends on $\log \varepsilon^{-1}$ so we can make this loss negligible.

5 The Online Algorithm

In the previous sections we were working offline: in particular, in computing the "good" classifier $\lambda \in \mathcal{C}$, we assumed knowledge of the entire element set. We now present the online framework for the setting where elements come in random order. Recall the definition of the curve $\mathcal{C}$ from §3, and the fact that there is a total order among all $\lambda \in \mathcal{C}$. Recall that for simplicity we restrict the constraints $\mathscr{F}$ to be matroid constraints.

For a subset of elements $A \subseteq U$, let $\mathsf{opt}(A)$ and $\mathsf{fopt}(A)$ denote the integer and fractional optimal profit for $\mathscr{F}|_A$, the feasible solutions restricted to elements in A. Note that in the fractional case this means the best solution in the convex hull $\mathrm{Conv}(\mathscr{F}|_A)$. Clearly, $\mathsf{fopt}(A) \geq \mathsf{opt}(A)$. We use opt and fopt to denote $\mathsf{opt}(U)$ and $\mathsf{fopt}(U)$ for the entire instance U.

Again we work under Assumption 12. We will also make use of any algorithm for maximizing submodular functions over $\mathscr{F}$ in the random-order model satisfying the following.

▶ **Assumption 25.** *Algorithm SubmodMS takes a nonnegative monotone submodular function f with $f(\emptyset) = 0$, and a number N. When run on a sequence X of N elements presented in random order, it returns a (random) subset $X_{alg} \in \mathscr{F}$ with expected value $\mathbb{E}[f(X_{alg})] \geq \frac{1}{\alpha} \max_{X' \in \mathscr{F}} f(X)$. Moreover, the it only evaluates the function f on **feasible** sets.*

Our algorithm is very simple:

Algorithm 5.1 Online Algorithm for Profit Maximization

1: $L \leftarrow$ first Binomial$(n, 1/2)$ items.
2: $\mu \leftarrow$ largest vector on curve $\mathcal{C}$ s.t. $\mathsf{fopt}(L_\mu) \geq \frac{1}{12d}\,\mathsf{fopt}(L)$.
3: $R \leftarrow$ remaining instance, namely the last $n - |L|$ items.
4: $R^\circ_\mu \leftarrow \{e \in R \mid v(e) > \langle \mu, s(e) \rangle\}$ be the (strictly) "filtered" remaining instance.
5: **Un-constrained:** Select items in R°_μ not decreasing the current value of the solution.
Constrained: Run algorithm SubmodMS on R°_μ using profit function π, selecting items according to this algorithm, but do not add items that decrease the current value of the solution.

Note that L_μ denotes the set of items in the sample L picked by μ (Definition 13). In Step 2, we can use the Ellipsoid method to find fopt within negligible error. Moreover, we must do this for several sets L_μ and pick the largest one on $\mathcal{C}$ using a binary-search procedure. We defer the technical details to the full version of the paper.

5.1 Analysis

To analyze the algorithm, we need to show that the classifier μ learned in Step 2 is large enough that we do not waste space with useless items, but low enough that we admit enough useful items. For that we need the following concentration bound.

▶ **Lemma 26.** *Consider a submodular function $f : 2^{\mathcal{U}} \to \mathbb{R}$. Consider a set $Y \subseteq \mathcal{U}$ such that f is non-negative over all of its subsets, and that for some M:*

$$\text{For all } Y' \subseteq Y \text{ and element } e \in Y', \quad |f(Y') - f(Y' - e)| \leq M. \tag{5.8}$$

Let $\mathbf{Y}$ be the random subset obtained by picking each element from Y independently with some probability (possibly different for each item). Then $\Pr(|f(\mathbf{Y}) - \mathbb{E}[f(\mathbf{Y})]| \geq t) \leq \frac{2M\,\mathbb{E}[f(\mathbf{Y})]}{t^2}$.

We then also need π to satisfy (5.8) on the optimal solutions of any given sub-instance. For a vector $y \in \mathbb{R}^n$ and subset $A \subseteq U$, let y_A be the same as y on A, and zero outside A.

▶ **Claim 27.** *Consider any $U' \subseteq U$, and let y be an optimal fractional solution on $\mathscr{F}|_{U'}$ (so $\pi(y) = \mathsf{fopt}(U')$). Then for any $B \subseteq A \subseteq U'$ with $|A \setminus B| = 1$, we have $|\pi(y_A) - \pi(y_B)| \leq M$, where M is an upper bound on the profit from any single item.*

From Section 4, recall $\lambda^* \in \mathbb{R}^d_+$ is a classifier that satisfies properties (P1')–(P3').

▶ **Lemma 28.** *Given Assumption 12, the classifier μ of Line 2 of Algorithm 5.1 satisfies:*

(a) *(Not too small) $\mu \geq \lambda^*$, with probability at least $19/20$.*

(b) *(Not too big) $\mathsf{fopt}(U_\mu) \geq \frac{\mathsf{fopt}}{48d}$ with probability at least $1 - 1/20d \geq 19/20$.*

Proof sketch. For the first part, we show that the classifier λ^* satisfies the properties needed in Line 2 with probability $1 - 1/20$; since μ is the largest such vector, we get $\mu \geq \lambda^*$. Using Theorem 21 and the assumption that no item has large profit, we have $\mathsf{fopt}(U_{\lambda^*}) \geq \frac{\mathsf{fopt}}{3d}$. Moreover, the sample obtains at least half of this profit in expectation, i.e., $\mathbb{E}\,\mathsf{fopt}(L_{\lambda^*}) \geq \frac{\mathsf{fopt}}{3d}$. Then using Lemma 26 with the Lipschitz property of Claim 27 and the no-high-profit-item assumption, we have $\mathsf{fopt}(L_{\lambda^*}) \geq \frac{\mathsf{fopt}}{12d} \geq \frac{\mathsf{fopt}(L)}{12d}$ with probability at least $19/20$. Thus, with this probability λ^* satisfies the properties needed in Line 2 of the algorithm, as desired.

For the part (b) of the lemma, notice that for each scenario $\mathsf{fopt}(U_\mu) \geq \mathsf{fopt}(L_\mu)$, since feasible solutions for the sample are feasible for the whole instance. Next, by definition of μ, $\mathsf{fopt}(L_\mu) \geq \frac{\mathsf{fopt}(L)}{12d}$. Finally, if x is the fractional optimal solution on U with $\pi(x) = \mathsf{fopt}$, then $\mathbb{E}[\pi(x_L)] \geq \mathsf{fopt}/2$, since g is superadditive. Again using Lemma 26, the profit $\pi(x_L)$ is at least $\frac{\mathsf{fopt}}{4}$ with probability at least $(1 - 1/20d)$. Of course, $\mathsf{fopt}(L) \geq \pi(x_L)$. Chaining these inequalities, $\mathsf{fopt}(U_\mu) \geq \frac{\mathsf{fopt}}{48d}$ with this probability. ◀

In view of Theorem 21, we show the filtered out-of-sample instance R°_μ behaves like $U^\circ_{\lambda^*}$.

▶ **Lemma 29.** *The filtered out-of-sample instance R°_μ satisfies the following w.p. $19/20$:*

(a) *For all $e \in R^\circ_\mu$, $v(e) \geq \langle \lambda^*, s(e) \rangle$.*

(b) *For all x with support$(x) \subseteq R^\circ_\mu$ such that $x \in \mathscr{F}$, $\nabla g(Sx) \leq \lambda^*$.*

(c) $\mathsf{fopt}(R^\circ_\mu) \geq \frac{\mathsf{fopt}}{200d}$.

Proof. By Lemma 28(a), threshold $\mu \geq \lambda^*$ with probability $19/20$. When that happens, $U^\circ_\mu \subseteq U^\circ_{\lambda^*}$. Since the first two properties hold for $U^\circ_{\lambda^*}$, they also hold for U°_μ, and by downward-closedness, also for R°_μ.

For the third part, let λ^+ be the largest threshold in $\mathcal{C}$ such that $\mathsf{fopt}(U_{\lambda^+}) \geq \frac{\mathsf{fopt}}{48d}$. From Lemma 28(b), with good probability we have $\mu \leq \lambda^+$. Since μ is a smaller threshold, the instance U_{λ^+} is contained in the instance U_μ, which implies that for every scenario $\mathsf{fopt}(R_\mu) \geq \mathsf{fopt}(R_{\lambda^+})$. Next we show that that with good probability $\mathsf{fopt}(R_{\lambda^+}) \geq \frac{\mathsf{fopt}}{200d}$, and hence get the same lower bound for $\mathsf{fopt}(R_\mu)$. If y is the optimal fractional solution for U_{λ^+}, then y_R is feasible for R_{λ^+} with $\mathbb{E}[\pi(y_R)] = \frac{1}{2}\mathsf{fopt}(U_{\lambda^+}) \geq \frac{\mathsf{fopt}}{96d}$. Moreover, using the concentration bound again, we get that $\pi(y_R) \geq \frac{\mathsf{fopt}}{192d}$ with probability at least $19/20$. Finally, by the assumption of general position, there is at most one item in $R_\mu \setminus R^\circ_\mu$. Dropping this item from the solution y to get y° reduces the value by at most $M = \frac{\mathsf{fopt}}{10^4 d}$; here we use subadditivty of the profit, and that there are no exceptional items. Hence, with probability at least $19/20$: $\mathsf{fopt}(R^\circ_\mu) \geq \mathsf{fopt}(R^\circ_{\lambda^+}) \geq \pi(y^\circ_R) \geq \frac{\mathsf{fopt}}{196d} - M \geq \frac{\mathsf{fopt}}{200d}$. ◀

Finally, we are ready to prove the main theorems in the online setting.

▶ **Theorem 30** (Unconstr. Case: Supermodular Cost)**.** *Algorithm 5.1 gives an $O(d)$-approximation in expectation for the unconstrained case, if the cost function is supermodular.*

Proof. Define the event $\mathcal{E}$ that Lemmas 28 and 29 hold; $\Pr(\mathcal{E}) \geq 17/20$. Now, by Lemma 29(c), the optimal fractional solution for R°_μ has profit at least $\mathsf{fopt}/200d$. Moreover, since there are no constraints, the profit function is monotone submodular over all of $U^\circ_{\lambda^*}$ by Lemma 23. Conditioning on the good event $\mathcal{E}$, Lemma 28(a) gives that $R^\circ_\mu \subseteq U^\circ_{\lambda^*}$, so the algorithm to maximize the monotone submodular function (both integrally and fractionally) is to pick all elements. Hence, conditioned on $\mathcal{E}$, the profit we get is at least $\mathsf{fopt}/200d$. In the other case, we never pick an item that gives negative marginal value, so our solution is always non-negative. Hence our expected profit is at least $\Pr[\mathcal{E}] \cdot \mathsf{opt}(R_\mu) = \Omega(\mathsf{fopt}/d) \geq \Omega(\mathsf{opt}/d)$. ◀

The analysis of the algorithm for the constrained separable-cost case is similar, only using the constrained offline guarantees of Theorem 21, and the non-negativization Lemma 23 to argue that SubmodMS maintains its guarantees.

▶ **Theorem 31** (Constr. Case: Separable Cost)**.** *Suppose algorithm SubmodMS satisfies Assumption 25 and is α-competitive in expectation. Then Algorithm 5.1 gives a $O(\alpha d^2)$-approximation in expectation.*

6 Separability versus Supermodularity

In this section, we show that an β-approximation algorithm for the separable-cost case gives a $O(d\beta)$-approximation for a slight generalization of the supermodular-cost case. Consider the problem of picking a set A to solve

$$\pi(A) := \max_{A \in \mathscr{F}} \left(v(A) - g\Big(\sum_{e \in A} s(e)\Big) \right),$$

where $v(A)$ is a (discrete) submodular function over $\{0,1\}^n$ with $v(\emptyset) = 0$, g is a convex, (continuous) supermodular function over $\mathbb{R}^d$, and $\mathscr{F}$ is some downward-closed constraint set. We show that for the case of matroid constraints, this problem can be reduced to the setting where the cost function is separable over its d coordinates, suffering a loss of $O(d)$.

▶ **Theorem 32** (Reduction)**.** *Given an β-approximation algorithm for profit-maximization for* separable *convex cost functions under matroid constraints, we can get an $d(\beta + 2\mathrm{e}d)$-approximation algorithm for the profit-maximization problem with* supermodular costs g*,* submodular values v*, and $\mathscr{F}$ being a* matroid *constraint.*

The reduction is the following:

1. Define separable costs $\overline{g}(y) := 1/d \sum_{i=1}^d g_i(dy_i)$, where g_i are marginal functions for g.
2. W.p. $p = \frac{\beta}{\beta + 2\mathrm{e}d}$, run single-secretary algorithm to return element with maximum profit.
3. W.p. $1 - p = \frac{2\mathrm{e}d}{\beta + \mathrm{e}d}$, run algorithm for value function $v(\cdot)$ and separable cost fn. $\overline{g}(\cdot)$.

This reduction relies on the following simple but perhaps surprising observation that relates separability with supermodularity, which may find other applications.

▶ **Lemma 33.** *Given a monotone convex superadditive function g with $g(0) = 0$, let g_i be the marginal functions. Then for all $y \in \mathbb{R}^d_+$:*

1. $g(y) \geq \sum_i g_i(y_i)$
2. $g(y) \leq \frac{1}{d} \sum_i g_i(dy_i) = \overline{g}(y)$.

Proof. The first property follows from the superadditivity of g, and the second follows from Jensen's inequality. ◀

While the full proof of Theorem 32 is deferred to the full version of the paper, the main idea is clean. Given an optimal integer solution x^* for the original problem (with the original cost function), we use Lemma 33 and the Lovász (convex) extension of submodular functions to show that x^*/d is a good fractional solution for the separable cost function. Now using polyhedral properties of d-dimensional faces of the matroid polytope, and other properties of the Lovász extension, we show the existence of a good integer solution to the separable problem. Combining this reduction with Theorem 2 proves Theorem 3.

References

1. Shipra Agrawal and Nikhil R. Devanur. Fast algorithms for online stochastic convex programming. In *Proceedings of the Twenty-Sixth Annual ACM-SIAM Symposium on Discrete Algorithms*, pages 1405–1424. SIAM, Philadelphia, PA, 2015. doi:10.1137/1.9781611973730.93.
2. Shipra Agrawal, Zizhuo Wang, and Yinyu Ye. A dynamic near-optimal algorithm for online linear programming. *Oper. Res.*, 62(4):876–890, 2014. doi:10.1287/opre.2014.1289.
3. Yossi Azar, Niv Buchbinder, T-H. Hubert Chan, Shahar Chen, Ilan R. Cohen, Anupam Gupta, Zhiyi Huang, Ning Kang, Viswanath Nagarajan, Joseph (Seffi) Naor, and Debmalya Panigrahi. Online algorithms for covering and packing problems with convex objectives. In *57th Annual IEEE Symposium on Foundations of Computer Science, FOCS 2016, New Brunswick, NJ, USA, October 9-11, 2016*, 2016.
4. Moshe Babaioff, Nicole Immorlica, David Kempe, and Robert Kleinberg. A knapsack secretary problem with applications. In *APPROX-RANDOM*, pages 16–28, 2007. doi:10.1007/978-3-540-74208-1_2.
5. Siddharth Barman, Seeun Umboh, Shuchi Chawla, and David Malec. Secretary problems with convex costs. In *Automata, languages, and programming. Part I*, volume 7391 of *Lecture Notes in Comput. Sci.*, pages 75–87. Springer, Heidelberg, 2012. doi:10.1007/978-3-642-31594-7_7.
6. Avrim Blum, Anupam Gupta, Yishay Mansour, and Ankit Sharma. Welfare and profit maximization with production costs. In *FOCS*, pages 77–86, Nov 2011. doi:10.1109/FOCS.2011.68.
7. Brian C. Dean, Michel X. Goemans, and Jan Vondrák. Adaptivity and approximation for stochastic packing problems. In *Proceedings of the Sixteenth Annual ACM-SIAM Symposium on Discrete Algorithms, SODA 2005, Vancouver, British Columbia, Canada, January 23-25, 2005*, pages 395–404, 2005. URL: http://dl.acm.org/citation.cfm?id=1070432.1070487.
8. Nikhil R. Devanur, Kamal Jain, Balasubramanian Sivan, and Christopher A. Wilkens. Near optimal online algorithms and fast approximation algorithms for resource allocation problems. In Yoav Shoham, Yan Chen, and Tim Roughgarden, editors, *ACM Conference on Electronic Commerce*, pages 29–38. ACM, 2011. doi:10.1145/1993574.1993581.
9. Reza Eghbali and Maryam Fazel. Designing smoothing functions for improved worst-case competitive ratio in online optimization. In *Advances in Neural Information Processing Systems 29: Annual Conference on Neural Information Processing Systems 2016, December 5-10, 2016, Barcelona, Spain*, pages 3279–3287, 2016. URL: http://papers.nips.cc/paper/6073-designing-smoothing-functions-for-improved-worst-case-competitive-ratio-in-online-optimization.
10. Reza Eghbali, Jon Swenson, and Maryam Fazel. Exponentiated subgradient algorithm for online optimization under the random permutation model. *CoRR*, abs/1410.7171, 2014. URL: http://arxiv.org/abs/1410.7171, arXiv:1410.7171.

11 Moran Feldman, Ola Svensson, and Rico Zenklusen. A simple $O(\log \log(\text{rank}))$-competitive algorithm for the matroid secretary problem. In *Proceedings of the Twenty-Sixth Annual ACM-SIAM Symposium on Discrete Algorithms*, pages 1189–1201. SIAM, Philadelphia, PA, 2015. doi:10.1137/1.9781611973730.79.

12 Moran Feldman and Rico Zenklusen. The submodular secretary problem goes linear. In *2015 IEEE 56th Annual Symposium on Foundations of Computer Science—FOCS 2015*, pages 486–505. IEEE Computer Soc., Los Alamitos, CA, 2015.

13 P. R. Freeman. The secretary problem and its extensions: a review. *Internat. Statist. Rev.*, 51(2):189–206, 1983. doi:10.2307/1402748.

14 Anupam Gupta and Marco Molinaro. How the experts algorithm can help solve lps online. *Math. Oper. Res.*, 41(4):1404–1431, 2016. doi:10.1287/moor.2016.0782.

15 Jean-Baptiste Hiriart-Urruty and Claude Lemaréchal. *Fundamentals of convex analysis*. Grundlehren Text Editions. Springer-Verlag, Berlin, 2001. doi:10.1007/978-3-642-56468-0.

16 Zhiyi Huang and Anthony Kim. Welfare maximization with production costs: a primal dual approach. In *26th SODA*, pages 59–72. SIAM, Philadelphia, PA, 2015. doi:10.1137/1.9781611973730.6.

17 Thomas Kesselheim, Klaus Radke, Andreas Tönnis, and Berthold Vöcking. Primal beats dual on online packing LPs in the random-order model. In *STOC'14—Proceedings of the 2014 ACM Symposium on Theory of Computing*, pages 303–312. ACM, New York, 2014.

18 Robert Kleinberg. A multiple-choice secretary algorithm with applications to online auctions. In *Proceedings of the sixteenth annual ACM-SIAM symposium on Discrete algorithms*, SODA '05, pages 630–631, Philadelphia, PA, USA, 2005. Society for Industrial and Applied Mathematics. URL: http://portal.acm.org/citation.cfm?id=1070432.1070519.

19 Oded Lachish. O(log log rank) competitive ratio for the matroid secretary problem. In *55th IEEE Annual Symposium on Foundations of Computer Science, FOCS 2014, Philadelphia, PA, USA, October 18-21, 2014*, pages 326–335, 2014. doi:10.1109/FOCS.2014.42.

20 Marco Molinaro and R. Ravi. The geometry of online packing linear programs. *Math. Oper. Res.*, 39(1):46–59, 2014. doi:10.1287/moor.2013.0612.

21 Donald M. Topkis. *Supermodularity and complementarity*. Frontiers of Economic Research. Princeton University Press, Princeton, NJ, 1998.