

Competitive online algorithms for resource allocation over the positive semidefinite cone

Reza Eghbali¹ · James Saunderson² ·
Maryam Fazel³

Received: 2 February 2018 / Accepted: 21 May 2018 / Published online: 12 June 2018
© Springer-Verlag GmbH Germany, part of Springer Nature and Mathematical Optimization Society 2018

Abstract We consider a new and general online resource allocation problem, where the goal is to maximize a function of a positive semidefinite (PSD) matrix with a scalar budget constraint. The problem data arrives online, and the algorithm needs to make an irrevocable decision at each step. Of particular interest are classic experiment design problems in the online setting, with the algorithm deciding whether to allocate budget to each experiment as new experiments become available sequentially. We analyze two greedy primal-dual algorithms and provide bounds on their competitive ratios. Our analysis relies on a smooth surrogate of the objective function that needs to satisfy a new diminishing returns (PSD-DR) property (that its gradient is order-reversing with respect to the PSD cone). Using the representation for monotone maps on the PSD cone given by Löwner’s theorem, we obtain a convex parametrization of the family of functions satisfying PSD-DR. We then formulate a convex optimization problem to directly optimize our competitive ratio bound over this set. This design problem can be solved *offline* before the data start arriving. The online algorithm that uses the designed smoothing is tailored to the given cost function, and enjoys a competitive ratio at least as good as our optimized bound. We provide examples of computing the

✉ Reza Eghbali
reghbali@cisco.com; eghbali@uw.edu

James Saunderson
james.saunderson@monash.edu

Maryam Fazel
mfazel@uw.edu

¹ Tetration Analytics, Cisco Systems, Inc., Palo Alto, CA 94301, USA

² Department of Electrical and Computer Systems Engineering, Monash University, VIC 3800, Australia

³ Department of Electrical Engineering, University of Washington, Seattle, WA 98195, USA

smooth surrogate for D-optimal and A-optimal experiment design, and demonstrate the performance of the custom-designed algorithm.

Keywords Online algorithms · Competitive ratio · Positive semidefinite cone · Löwner's theorem

Mathematics Subject Classification 68W27 Online Algorithms · 90C25 Convex Programming · 62K05 Optimal Design

1 Introduction

Online resource allocation problems and algorithms have been traditionally studied in operations research (e.g., revenue management [2] and references therein), online network routing [6] and computer science (e.g., online packing and covering in linear or convex case [3,8], online welfare maximization [19]). In recent years, applications in online advertising such as the Adwords problem [22] have garnered renewed interest in this topic. In all these applications, the demands for resources arrive in an online, sequential fashion, and resource allocation also happens online. For example, in online revenue management, customers arrive sequentially, reveal their demand for the resources and offer a bid price. The inventory owner then needs to make an irrevocable decision about the latest customer's offer, without knowing future demands, while aiming to maximize his or her revenue. A feature in these problems is that the variables denoting allocation amounts, the bid prices, and the coefficients in the resource constraints are all *nonnegative*, thus a resource can only be used up as time goes on (resource usage is nondecreasing) while the total available resource is fixed. In the context of linear programming (LP), these are sometimes called “packing” problems, e.g., [8,24]. For an overview of online linear programming and more generally online convex problems with different models for online information arrival, we refer the reader to [1,15,17,20].

In this paper, we consider a new and general online resource allocation problem, where the objective is a function of a positive semidefinite (PSD) matrix (thus the problem is defined on the PSD cone S_+^n), with a scalar budget constraint. The problem can be described as follows: At round t , the algorithm receives a matrix $A_t \in S_+^n$ and scalar $c_t > 0$, and then chooses $x_t \in [0, 1]$. The goal of the algorithm is to maximize $H\left(\sum_{t=1}^m A_t x_t\right)$ subject to the budget constraint $\sum_{t=1}^m c_t x_t \leq b$. The offline problem can be written as

$$\text{maximize } H\left(\sum_{t=1}^m A_t x_t\right) \quad \text{subject to} \quad \begin{cases} \sum_{t=1}^m c_t x_t \leq b, \\ 0 \leq x_t \leq 1, \quad t = 1, \dots, m. \end{cases} \quad (1)$$

We assume that $H : S_+^n \rightarrow \mathbf{R}$ is a monotone, concave *trace function*, i.e.,

$$H(X) = \sum_{i=1}^n h(\lambda_i(X)),$$

where $h : \mathbf{R}_+ \rightarrow \mathbf{R}$ is a monotone concave function and $\lambda_i(X)$ denotes the i th eigenvalue of X . Note that analogous to packing LPs, $\sum_{s=1}^t A_s x_s$ is nondecreasing with respect to the PSD cone. In other words,

$$\sum_{s=1}^t A_s x_s \preceq \sum_{s=1}^{t+1} A_s x_s \quad \text{for all } t,$$

where $X \preceq Y$ means that $Y - X$ is positive semidefinite.

In the online algorithms literature, various scenarios have been considered for the information being revealed to the algorithm (the sequence of A_t and c_t): from the worst-case adversarial model to various stochastic models. In this paper, since we are interested in understanding the limits of performance and providing competitive ratio bounds, we assume an adversarial model, where little is known about the arriving data. One could use techniques similar to ours in other scenarios as well, which we leave for future work.

Our setup covers online versions of problems such as *sensor selection* or *experiment design*, and *graph formation* [23]. In these problems, $A_t = a_t a_t^T$ for some $a_t \in \mathbf{R}^n$. In online experiment design, the vector a_t is an experiment or measurement vector that provides linear noisy measurements of an unknown vector w , i.e., $\xi_t = \langle a_t, w \rangle + n_t$, where n_t is Gaussian noise and $c_t x_t$ is the experiment cost. In this paper, we consider two algorithms: one that assigns fractional values to x_t and one that assign integer values to x_t . The algorithm makes online decisions about the budget to allocate to an experiment, aiming to minimize various functions of the error covariance matrix of the maximum a priori estimate of w .

The online experiment design problem can be expressed as (1), with different choices of H giving rise to different criteria for optimal experiment design. Examples include $H(X) = \log \det(\epsilon I + X)$ for the D-optimal criterion, $H(X) = -\mathbf{tr}(\epsilon I + X)^{-1}$ for the A-optimal criterion, and $H(X) = -\mathbf{tr}(\epsilon I + X)^{-p}$ for the p th mean criterion. (The prior distribution on w is assumed to be $\mathcal{N}(0, \frac{1}{\epsilon} I)$.)

We use the A-optimal and D-optimal criteria, as well as the problem where $H(X) = \mathbf{tr}(X)$, as running examples throughout the paper. The latter simply recovers a linear program with one packing constraint, and provides insight into the connections and differences with online Linear Programming results, e.g., [8] (discussed in Sect. 3).

Our results exploit a crucial property of the objective function H or its surrogate H_S (discussed in Sect. 3.2) which we refer to as the *PSD diminishing returns* property. For a function with a scalar variable, this property simply means the larger the variable, the smaller the derivative. More generally, we define it as follows.

Definition 1 (*PSD diminishing returns*) A concave trace function F satisfies *PSD diminishing returns* (*PSD-DR*) if ∇F is order-reversing with respect to S_+^n , i.e.,

$$\nabla F(U) \preceq \nabla F(U'), \quad \text{whenever } U \succeq U'.$$

Our contributions. Our focus is on developing algorithms with (multiplicative) competitive ratio guarantees for problems of the form (1). In other words, our algorithms provide feasible solutions x_1, x_2, \dots, x_m such that $H\left(\sum_{t=1}^m A_t x_t\right) - H(0) \geq \text{cr}(P^* - H(0))$, where cr bounds the competitive ratio and P^* is the optimal value of (1)¹.

The algorithms we consider (Algorithms 1 and 2, in Sect. 2) are parameterized by a smoothed surrogate, H_S , of the objective function H , and a smoothed penalty function G_S in place of the budget constraint $\sum_{t=1}^m x_t \leq b$. Our main contribution is a systematic method, based on solving an offline convex optimization problem, to design H_S and G_S , and to compute a bound on the competitive ratio of the associated algorithms.

Our approach builds on the results in [13], which considers a general online conic optimization problem and shows that smoothing the objective function (to obtain a surrogate) before applying a primal-dual greedy algorithm improves the competitive ratio. However, prior work was not able to address the problem of finding the best surrogate (or even representing the space of suitable surrogates)—except for the special (and restrictive) case of separable objectives defined on the non-negative orthant, which reduces to a search over concave scalar functions. There is no hope of employing a similar strategy for functions defined on the PSD cone, unless one has access to representations that lead to a convex parametrization for PSD-DR functions. The present paper resolves this issue.

We require the smoothed objective H_S to satisfy the PSD-DR property. Note that this does not hold for all concave trace functions (indeed fails to hold for the objective function in A-optimal experiment design). If the objective function in problem (1) does not satisfy this property, our approach allows us to design an appropriate surrogate H_S that satisfies it, and use this H_S to construct algorithms with competitive ratio guarantees. And if the original H does satisfy the property, our approach still helps improve the competitive ratio; see Sect. 3.2. A key observation in this paper is that we can use Löwner’s theorem, characterizing operator monotone functions, to impose this constraint in a computationally effective way by requiring it to have a certain integral representation (see 17).

The rest of the paper is structured as follows. In Sect. 2, we describe the algorithms. In Sects. 3 and 4, we provide the competitive ratio analysis for the online algorithms described in Sect. 2 and determine how the competitive ratio depends on the smoothed surrogate of the objective function. We set up optimization problems for finding surrogates that maximize the competitive ratio bound derived in these sections. Numerical examples are presented in Sect. 5. Related works are discussed in Sect. 6.

2 Algorithms

In this section we describe the class of online algorithms we consider to solve (1), and give bounds on the duality gap achieved by these algorithms. The algorithms we consider (Algorithms 1 and 2, stated in Sect. 2.1) are (modified) greedy algorithms,

¹ To simplify the notation in the rest of the paper, assume $H(0) = 0$ by replacing $h(u)$ with $h(u) - h(0)$.

where a primal-dual greedy approach is applied to a smoothed surrogate for the objective. These algorithms are instances of the general algorithmic framework for conic online optimization introduced in [13], which can recover state-of-the-art algorithms developed for Adwords [9], Adwords with concave returns [10], and online LP [7].

Before describing the algorithms, it is helpful to slightly rewrite (1) and state its dual. Throughout, let $G : \mathbf{R}_+ \rightarrow \mathbf{R}$ be the indicator function of the interval $[0, b]$, i.e.,

$$G(u) = \begin{cases} 0 & \text{if } u \in [0, b] \\ -\infty & \text{otherwise.} \end{cases} \quad (2)$$

Then we can rewrite (1) as

$$\underset{x_1, \dots, x_m \in [0, 1]}{\text{maximize}} \quad H \left(\sum_{t=1}^m A_t x_t \right) + G \left(\sum_{t=1}^m c_t x_t \right). \quad (3)$$

To derive a dual program, we rewrite the primal as:

$$\begin{aligned} & \underset{x_1, \dots, x_m \in [0, 1]}{\text{maximize}} \quad H(U) + G(u) . \\ \text{subject to} \quad & U = \sum_{t=1}^m A_t x_t \\ & u = \sum_{t=1}^m c_t x_t \end{aligned}$$

Introducing a dual matrix variable Y and dual scalar variable z corresponding to the two equality constraints, we can write the Lagrangian as follows:

$$\mathcal{L}(u, U, x_1, \dots, x_m, Y, z) = H(U) - \langle Y, U \rangle + G(u) - \langle z, u \rangle + \sum_{t=1}^m (\langle A_t, Y \rangle + c_t z) x_t$$

where $\text{dom}(\mathcal{L}) = \mathbf{S}^n \times \mathbf{R} \times [0, 1]^m \times \mathbf{S}^n \times \mathbf{R}$. Maximizing \mathcal{L} over all the primal variables, we can derive the dual of (1) in terms of conjugate functions as

$$\underset{z, Y}{\text{minimize}} \quad \sum_{t=1}^m (\langle A_t, Y \rangle + c_t z)_+ - H^*(Y) - G^*(z), \quad (4)$$

where, for a function $\psi : \mathbf{R}^n \rightarrow \mathbf{R}$, ψ^* denotes the concave conjugate of ψ , defined as $\psi^*(y) = \inf_u \langle y, u \rangle - \psi(u)$. The concave conjugate of G is

$$G^*(z) = \begin{cases} bz & \text{if } z \leq 0 \\ 0 & \text{otherwise.} \end{cases}$$

For a trace function $H(U) = \sum_{i=1}^n h(\lambda_i(U))$ we have that $H^*(Y) = \sum_{i=1}^n h^*(\lambda_i(Y))$, a result established in [21].

2.1 Statement of algorithms

Recall that our objective function H is a monotone, concave, trace function, and that G is the concave indicator function of the set $[0, b]$. We consider two online algorithms for (1). Algorithm 1 makes sequential updates to the primal and dual variables, whereas Algorithm 2 makes these updates simultaneously. These algorithms require a choice of functions H_S and G_S that are smooth surrogates for H and G . Throughout, we assume these functions satisfy the following.

Assumption 1 (*Assumptions on H_S*) We assume that $h_S : O \rightarrow \mathbf{R}$ is concave, differentiable, and satisfies $h_S(0) = 0$ and $h'_S(0) = h'(0)$, where O is an open interval that contains \mathbf{R}_+ . We assume that $H_S : S_+^n \rightarrow \mathbf{R}$ is the trace function $H_S(U) = \sum_{i=1}^n h_S(\lambda_i(U))$.

Assumption 2 (*Assumptions on G_S*) We assume that $G_S : O \rightarrow \mathbf{R}$ is concave, differentiable, and satisfies $G_S(0) = 0$ and $G'_S(0) = 0$, where O is an open interval that contains \mathbf{R}_+ .

The problem of *designing* H_S and G_S , given H and G , is the main focus of the paper, and is the subject of Sect. 3. For now we merely point out that the results of this section hold for any choice of H_S and G_S satisfying the basic assumptions above.

Initialize $\hat{z}_0 = G'_S(0)$, $\hat{Y}_0 = \nabla H_S(0)$;

for $i \leftarrow 1$ **to** m **do**

Receive A_t, c_t ;
 $\hat{x}_t = \begin{cases} 1, & \text{if } c_t \hat{z}_{t-1} + \langle A_t, \hat{Y}_{t-1} \rangle > 0 \\ 0, & \text{if } c_t \hat{z}_{t-1} + \langle A_t, \hat{Y}_{t-1} \rangle \leq 0; \end{cases}$
 $\hat{Y}_t = \nabla H_S \left(\sum_{s=1}^t A_s \hat{x}_s \right)$;
 $\hat{z}_t = G'_S \left(\sum_{s=1}^t c_s \hat{x}_s \right)$;

end

Algorithm 1: Sequential Update

for $i \leftarrow 1$ **to** m **do**

Receive A_t, c_t ;
 $(\tilde{z}_t, \tilde{Y}_t, \tilde{x}_t) \in \arg \min_{z, Y} \max_{x \in [0, 1]} \langle Y, A_t x + \sum_{s=1}^{t-1} A_s \tilde{x}_s \rangle + z \left(c_t x + \sum_{s=1}^{t-1} c_s \tilde{x}_s \right) - H_S^*(Y) - G_S^*(z)$

end

Algorithm 2: Simultaneous Update

In Algorithm 2, at each step the primal variable \tilde{x}_t , and the dual variables \tilde{Y}_t and \tilde{z}_t , are updated together, by finding the saddle point of the Lagrangian (solution to the min-max problem). To argue for the existence of a saddle point, we use Corollary 11.41b in [27] which requires two conditions to be satisfied. The first condition is that the following problem has a nonempty, bounded set of maximizers,

$\tilde{x}_t \in \arg \max_{x \in [0, 1]} H_S \left(A_t x + \sum_{s=1}^{t-1} A_s \tilde{x}_s \right) + G_S \left(x + \sum_{s=1}^{t-1} \tilde{x}_s \right)$,
 which holds because $[0, 1]$ is a compact convex set. The second condition requires

$$\begin{aligned} \sum_{s=1}^{t-1} A_s \tilde{x}_s &\in \text{int} (\{-A_t x \mid x \in [0, 1]\} + \text{dom} H_S), \\ \sum_{s=1}^{t-1} c_s \tilde{x}_s &\in \text{int} (\{-c_t x \mid x \in [0, 1]\} + \text{dom} G_S), \end{aligned}$$

where the sum is the Minkowski sum of two sets, and int denotes the interior of a set. This condition is also satisfied since 0 is in the interior of the domains of h_S and G_S (from Assumptions 1 and 2).

Algorithm 1 can also be interpreted in similar terms, by observing that the primal \hat{x}_t update can be written as

$$\hat{x}_t \in \operatorname{argmax}_{x \in [0, 1]} x(c_t \hat{z}_{t-1} + \langle A_t, \hat{Y}_{t-1} \rangle),$$

and using the fact that for a differentiable concave function ψ , we have $\nabla \psi(u) = \arg \min_y \langle y, u \rangle - \psi^*(u)$, the dual updates can be written as

$$\begin{aligned} \hat{z}_t &= \operatorname{argmin}_z z \left(\sum_{s=1}^t c_s \hat{x}_s \right) - G_S^*(z) \quad \text{and} \\ \hat{Y}_t &= \operatorname{argmin}_Y \langle Y, \sum_{s=1}^t A_s \hat{x}_s \rangle - H_S^*(Y), \end{aligned}$$

As such, the sequential algorithm can be viewed as alternating over maximization and minimization of the Lagrangian over primal and dual variables.

Algorithm 1, assigns integer values to \hat{x}_t and the dual variable \hat{z}_t acts as a decision threshold in the assignment rule for \hat{x}_t . For the reader familiar with online learning, we point out a connection with the typical online learning setup. The \hat{Y}_t update step is the same as *Follow-the-Regularized-Leader (FTRL)* update with H_S as the regularizer, so this algorithm is running FTRL on the dual problem.

Algorithm 2 clearly requires more computation than the sequential algorithm, but it is also easier to bound its competitive ratio. We do not focus on implementation details of this algorithm in this paper, but we point out that one can decide how accurately to solve the saddle-point subproblem at each iteration, trading off computation with desired accuracy (for each subproblem). To analyze this more practical variation of Algorithm 2, our analysis should take into account how solving each saddle-point problem to ϵ accuracy affects the overall competitive ratio. We leave this additional analysis to future work.

2.2 Duality gap

The competitive analysis, and subsequent design of H_S and G_S (in Sect. 3), relies on bounding the duality gap. Define the dual value achieved by the sequential and simultaneous algorithms by

$$D_{\text{seq}} = \sum_{t=1}^m \left(\langle A_t, \hat{Y}_{t-1} \rangle + c_t \hat{z}_{t-1} \right)_+ - H^*(\hat{Y}_m) - G^*(\hat{z}_m)$$

$$D_{\text{sim}} = \sum_{t=1}^m \left(\langle A_t, \tilde{Y}_t \rangle + c_t \tilde{z}_t \right)_+ - H^*(\tilde{Y}_m) - G^*(\tilde{z}_m).$$

The following two lemmas are essentially taken from [14]. We include proofs in “Appendix A” to make the present paper self-contained. Lemma 1 bounds the duality gap, and Lemma 2 relates D_{seq} and D_{sim} to the dual optimal value D^* , given an additional assumption on the gradient of H_S .

Lemma 1 *Let \tilde{x}_t , \tilde{Y}_t , and \tilde{z}_t and \hat{x}_t , \hat{Y}_t , and \hat{z}_t denote the t th iterate of the simultaneous and sequential algorithms, respectively. Then the duality gaps for the two algorithms satisfy the lower bounds*

$$H_S \left(\sum_{t=1}^m A_t \tilde{x}_t \right) + G_S \left(\sum_{t=1}^m c_t \tilde{x}_t \right) - D_{\text{sim}} \geq H^*(\tilde{Y}_m) + G^*(\tilde{z}_m). \quad (5)$$

$$H_S \left(\sum_{t=1}^m A_t \hat{x}_t \right) + G_S \left(\sum_{t=1}^m c_t \hat{x}_t \right) - D_{\text{seq}} \geq H^*(\hat{Y}_m) + G^*(\hat{z}_m)$$

$$+ \sum_{t=1}^m \langle A_t \hat{x}_t, \hat{Y}_t - \hat{Y}_{t-1} \rangle + \sum_{t=1}^m \langle c_t \hat{x}_t, \hat{z}_t - \hat{z}_{t-1} \rangle. \quad (6)$$

Proof See “Appendix A”. □

This duality gap bound is reminiscent of regret bounds in online learning. In regret analysis one directly bounds the right hand side of (6) in terms of the horizon and problem parameters. However, bounding the competitive ratio requires more assumptions and further analysis. The PSD diminishing returns (PSD-DR) property is used to relate D_{seq} and D_{sim} to the dual optimal value D^* .

Lemma 2 *If H_S satisfies PSD-DR, then $D_{\text{seq}} \geq D^*$ and $D_{\text{sim}} \geq D^*$.*

For completeness the proof is given in “Appendix A”. We remark that the PSD-DR property is a special case of the property used in the abstract framework of [13] that was applied to general concave functions on proper cones. In order to develop computational methods to *design* the function H_S (in Sect. 3.2), we will need to impose the PSD-DR property on H_S in a computationally tractable way. A central observation of this paper is that this is possible for the PSD-DR property by exploiting a celebrated theorem of Löwner. We discuss this in detail in Sect. 3.2.

3 Competitive ratio analysis

In this section we show how to design H_S and G_S in the sequential algorithm (Algorithm 2) to maximize a lower bound on the competitive ratio achieved. We discuss the appropriate modifications for the sequential algorithm in Sect. 4. In what follows, $e = 2.718\dots$ is Euler's number.

We first state a technical lemma (Lemma 3) that relates the budget consumed and competitive ratio achieved by Algorithm 2 to the functions G_S and H_S . The only dependence on the data (the values of A_t and c_t) is via two parameters, θ and Θ , which are lower and upper bounds on $c_t^{-1}\mathbf{tr}(A_t)$ for all t . We use a parameter $\gamma \geq 1$ as a trade-off parameter between the budget consumption and competitive ratio. The result says that if H_S, G_S satisfy certain inequalities that involve γ and β , then Algorithm 2 achieves a competitive ratio of at least $\frac{1}{\gamma/(e-1)+\beta}$ using a budget of $G_S'^{-1}(-h'(0)\Theta)$ (the inverse function of the derivative of G_S evaluated at a point). Concrete values for these bounds are worked out at the end of this section. This is a similar style of result to [7], which applies to a special class of linear programs.

Lemma 3 *Let G be as defined in (2), and let h be concave and monotonically increasing, with corresponding trace function H . Let G_S satisfy Assumption 2, h_S satisfy Assumption 1, and suppose that the corresponding trace function H_S satisfies the PSD-DR property. Suppose that $\theta \leq c_t^{-1}\mathbf{tr}(A_t) \leq \Theta$ for all t and $u_{\max} \geq \lambda_{\max}(\sum_{t=1}^m A_t \tilde{x}_t)$.² Then*

1. *If for a given $\gamma \geq 1$, G_S satisfies*

$$\gamma G_S(u) \leq G^*(G_S'(u)) + \frac{\gamma}{e-1}h(\theta u) \quad \text{for all } u \in [0, \infty), \quad (7)$$

and there exists $\beta > 0$ such that h_S satisfies

$$\gamma h_S(u) \leq h^*(h_S'(u)) + \beta h(u) \quad \text{for all } u \in [0, u_{\max}], \quad (8)$$

then the iterates $\tilde{x}_1, \dots, \tilde{x}_m$ of Algorithm 2 satisfy

$$H\left(\sum_{t=1}^m A_t \tilde{x}_t\right) \geq \frac{1}{\gamma/(e-1)+\beta} D^*. \quad (9)$$

2. *The iterates $\tilde{x}_1, \dots, \tilde{x}_m$ of Algorithm 2 also satisfy*

$$\sum_{t=1}^m c_t \tilde{x}_t \leq b' := \inf\{u : G_S'(u) \leq -h'(0)\Theta\}. \quad (10)$$

Before providing a proof of Lemma 3, we explain how we use it in what follows. As mentioned before, we consider γ a design parameter that trades off between (possible)

² Note that we could choose, for instance, $u_{\max} = b' \max_t c_t^{-1} \lambda_{\max}(A_t)$, but for certain classes of problems better bounds may be available.

budget violation and competitive ratio achieved. The smaller the parameter γ , the larger the competitive ratio (evident from 8 and 9) and the larger the budget violation. The role of γ in the amount of budget consumed and the reason for requiring $\gamma \geq 1$ is discussed in Sect. 3.1. In Sect. 3.1 we use (7) and (10) in Lemma 3 to design G_S to minimize the budget b' consumed for a given γ . This design problem requires the parameter θ only. The parameter γ allows us to decouple the design of G_S and h_S .

In Sect. 3.2, we use (8) and (9) in Lemma 3 to design H_S via solving a convex optimization problem that maximizes the bound on competitive ratio (by minimizing β). This design problem requires the parameter u_{\max} only.

Proof (of Lemma 3) First we show that $\sum_{t=1}^m c_t \tilde{x}_t \leq b'$. To do this we use the optimality conditions of the saddle point problem in Algorithm 2. Indeed, for any t , we have that

$$\tilde{Y}_t = \nabla H_S \left(\sum_{s=1}^t A_s \tilde{x}_s \right) \quad \text{and} \quad \tilde{z}_t = G'_S \left(\sum_{s=1}^t c_s \tilde{x}_s \right),$$

and that $\langle \tilde{Y}_t, A_t \rangle + c_t \tilde{z}_t < 0$ implies $\tilde{x}_t = 0$.

Arguing by contradiction, let T be the smallest index such that $\sum_{t=1}^T c_t \tilde{x}_t > b'$. Then, by the definition of b' we have that $\tilde{z}_T = G'_S \left(\sum_{t=1}^T c_t \tilde{x}_t \right) < -h'(0)\Theta$. Observe that $\sum_{t=1}^T A_t \tilde{x}_t \succeq 0$, from which it follows, from the fact that H_S is PSD-DR, that $\nabla H_S \left(\sum_{t=1}^T A_t \tilde{x}_t \right) \preceq \nabla H_S(0) = h'_S(0)I$. Then

$$\langle \tilde{Y}_T, A_T \rangle + c_T \tilde{z}_T = \langle \nabla H_S \left(\sum_{t=1}^T A_t \tilde{x}_t \right), A_T \rangle + c_T \tilde{z}_T \leq h'_S(0) \mathbf{tr} A_T + c_T \tilde{z}_T < 0$$

where, for the last inequality, we use the fact that $h'_S(0) = h'(0)$ and that $c_T^{-1} \mathbf{tr} A_T \leq \Theta$. It follows from the optimality conditions that $\tilde{x}_T = 0$. But then $\sum_{t=1}^{T-1} c_t \tilde{x}_t > b'$, contradicting our choice of T . It follows that $\sum_{t=1}^m c_t \tilde{x}_t \leq b'$.

We now turn our attention to the bound on the competitive ratio. Let $U = \sum_{t=1}^m A_t \tilde{x}_t$ and $u = \sum_{t=1}^m c_t \tilde{x}_t$, so that $\tilde{Y}_m = \nabla H_S(U)$, and $\tilde{z}_m = G'_S(u)$. By the duality gap bound (Lemma 1) we have

$$H_S(U) + G_S(u) - D_{\text{sim}} \geq H^*(\tilde{Y}_m) + G^*(\tilde{z}_m). \quad (11)$$

By the primal allocation rule in Algorithm 2, we have $\tilde{x}_t \left(c_t \tilde{z}_t + \langle A_t, \tilde{Y}_t \rangle \right) \geq 0$. Combining this observation with the concavity of H_S and G_S , we get

$$\begin{aligned} H_S \left(\sum_{s=1}^t A_s \tilde{x}_s \right) + G_S \left(\sum_{s=1}^t c_s \tilde{x}_s \right) - H_S \left(\sum_{s=1}^{t-1} A_s \tilde{x}_s \right) - G_S \left(\sum_{s=1}^{t-1} c_s \tilde{x}_s \right) \\ \geq \tilde{x}_t \left(c_t \tilde{z}_t + \langle A_t, \tilde{Y}_t \rangle \right) \geq 0. \end{aligned}$$

By taking the sum over t and telescoping the sum we get

$$H_S \left(\sum_{s=1}^m A_s \tilde{x}_s \right) + G_S \left(\sum_{s=1}^m c_s \tilde{x}_s \right) \geq 0. \quad (12)$$

The last preparatory observation we need is the inequality $H(U) \geq h(\theta u)$. To see why this holds, note that because h is monotonically increasing, and $\theta \leq c_t^{-1} \mathbf{tr}(A_t)$ for all t ,

$$\begin{aligned} h(\theta u) &= h \left(\theta \sum_{t=1}^m c_t \tilde{x}_t \right) \leq h \left(\mathbf{tr} \left(\sum_{t=1}^m A_t \tilde{x}_t \right) \right) \\ &= h(\mathbf{tr}(U)) \leq \sum_{i=1}^n h(\lambda_i(U)) = H(U) \end{aligned} \quad (13)$$

where the last inequality holds because h is concave and $h(0) \geq 0$ so h is subadditive on $[0, \infty)$. Now we can write

$$\begin{aligned} H(U) - D_{\text{sim}} &\geq -H_S(U) - G_S(u) + H^*(Y) + G^*(z) + H(U) && \text{By (11)} \\ &\geq -H_S(U) + (\gamma - 1) G_S(u) + \left(1 - \frac{\gamma}{e-1}\right) H(U) + H^*(Y) && \text{By (7) and (13)} \\ &\geq -H_S(U) + (1 - \gamma) H_S(U) + \left(1 - \frac{\gamma}{e-1}\right) H(U) + H^*(Y) && \text{By (12) and } \gamma \geq 1 \\ &= -\gamma H_S(U) + \left(1 - \frac{\gamma}{e-1}\right) H(U) + H^*(Y) \\ &\geq \left(1 - \frac{\gamma}{e-1} - \beta\right) H(U) && \text{By (8).} \end{aligned}$$

Then the result follows from Lemma 2. \square

These results allow us to search for functions h_S and G_S that satisfy the assumptions of Lemma 3. We use γ as a design parameter, controlling the trade-off between competitive ratio and the (possible) budget violation.

3.1 Smoothing the budget penalty function G , taking h into account.

For a fixed choice of γ , our aim is to design G_S so as to minimize b' . We do so by explicitly constructing $G'_S(u)$ such that (7) is satisfied with equality for all $u \geq 0$, and showing that such a G'_S is optimal, in the sense that for a given $\gamma \geq 1$, defining G_S via (14) minimizes the bound b' on the budget consumed by the simultaneous algorithm.

Proposition 1 Let \bar{G}_S be any function that satisfies (7). Let ³

$$G'_S(u) = -\frac{\gamma}{b(e-1)} \int_0^u \exp\left(\frac{\gamma}{b}(u-v)\right) \theta h'(\theta v) \, dv \quad \text{for } u \geq 0 \quad (14)$$

Then $\bar{G}'_S(u) \geq G'_S(u)$ for all $u \geq 0$ and so

$$\inf\{u : G'_S(u) \leq -h'(0)\Theta\} \leq \inf\{u : \bar{G}'_S(u) \leq -h'(0)\Theta\}.$$

Proof Clearly $G'_S(u) \leq 0$ for all $u \geq 0$, and so for any $u \geq 0$, we have that $G^*(G'_S(u)) = bG'_S(u)$. We now explain why G'_S satisfies (7) with equality. (This can also be verified by direct substitution.) Because $G_S(0) = 0$ (by assumption), it is enough to show that G'_S satisfies the linear, constant coefficient differential equation obtained by equating, and then differentiating, both sides of (7). After rearranging terms, this differential equation is

$$G''_S(u) = \frac{\gamma}{b} G'_S(u) - \frac{\gamma}{b(e-1)} \theta h'(\theta u).$$

The function $G'_S(u)$ that satisfies this equation for $u \in [0, \infty)$ together with $G'(0) = 0$, is given by the convolution of $-\frac{\gamma}{b(e-1)}\theta h'(\theta u)$ with $e^{\gamma u/b}$, as in (14).

Since G'_S satisfies (7) with equality, it follows from Gronwall's inequality (see [11] Corollary 2) that for any other function \bar{G}_S that satisfies (7), we have $\bar{G}'_S(u) \geq G'_S(u)$ for all $u \geq 0$. \square

Remark 1 Note that the function G_S is a convolution of $h'(\theta u)$ with an exponential function. It can also be viewed as a function derived by applying Nesterov's smoothing technique to G . To show this, we have to argue that $G_S^* - G^*$ is a concave function. Note that G^* is linear over \mathbf{R}_- ; therefore, $G_S^* - G^*$ is concave on the positive reals with $G_S^*(0) - G^*(0) = 0$, and for $y > 0$, $G_S^*(y) - G^*(y) = 0$. This establishes the concavity of $G_S^* - G^*$. As we discuss in Example 1 below, $G_S^* - G^*$ has a closed form when h is a linear function. However, for more general examples of h we do not have a closed form for $G_S^* - G^*$.

Now, to justify the fact that we require $\gamma \geq 1$, we find a lower bound G'_S given in (14) in terms of γ :

$$\begin{aligned} G'_S(u) &= -\frac{\gamma}{b(e-1)} \exp\left(\frac{\gamma}{b}u\right) \int_0^u \exp\left(-\frac{\gamma}{b}v\right) \theta h'(\theta v) \, dv \\ &\geq -\frac{\gamma \theta h'(0)}{b(e-1)} \exp\left(\frac{\gamma}{b}u\right) \int_0^u \exp\left(-\frac{\gamma}{b}v\right) \, dv = -\frac{\theta h'(0) (\exp(\frac{\gamma}{b}u) - 1)}{e-1} \end{aligned}$$

³ We can extend the domain of G_S to negative reals by letting $G_S = 0$ on \mathbf{R}_- to satisfy the technical assumption on the domain of G_S in Assumption (2).

The inequality above is exact if h is linear, e.g., $H(U) = \mathbf{tr}(U)$. Now we can bound b' as:

$$b' = \inf\{u : G'_S(u) \leq -h'(0)\Theta\} \geq \frac{b}{\gamma} \log\left(\frac{\Theta}{\theta}(e-1) + 1\right)$$

This shows that if $\gamma < 1$, then even in the trivial case where $\theta = \Theta$ and h is linear the algorithm can go over budget.

We now compute G'_S , and bound the budget b' consumed by the simultaneous algorithm using G'_S , for three examples.

Example 1 (Linear objective function) Consider the linear function $h(u) = u$ which translates to $H(U) = \mathbf{tr}(U)$. This case allows us to show that our approach specializes to recover known results. In this case, the problem reduces to a linear program with one budget constraint,

$$\text{maximize} \quad \sum_{t=1}^m \mathbf{tr}(A_t)x_t \quad \text{subject to} \quad \begin{cases} \sum_{t=1}^m c_t x_t \leq b \\ 0 \leq x_t \leq 1 \quad \text{for } t = 1, 2, \dots, m. \end{cases}$$

Computing the integral (14) gives $G'_S(u) = \theta(1 - \exp(-\frac{\gamma}{b}u))/(e-1)$. This choice of G'_S corresponds to the exponential update algorithm for online LP [7], and, in this particular case, G_S can also be derived as a smooth surrogate for G using Nesterov smoothing with a shifted entropy as the proximity function [13]. In this case the bound b' on the budget consumed is given by

$$b' = \inf\{u \mid G'_S(u) < -h'(0)\Theta\} = \frac{b}{\gamma} \log\left((e-1)\frac{\Theta}{\theta} + 1\right).$$

Choosing $\gamma \geq \log((e-1)\frac{\Theta}{\theta} + 1)$ ensures that the budget is not violated.

Example 2 (D-optimal experiment design) Suppose that $h(u) = \log(u+1)$, the objective function of interest in D-optimal experiment design. In this case $h'(u) = (1+u)^{-1}$, and we note that it is possible to express the optimal G'_S from (14) in terms of special functions called *exponential integrals*. To obtain an upper bound on b' , and hence on the budget consumed, we use the fact that $h'(\theta v) = (1+\theta v)^{-1} \geq e^{-\theta v}$ whenever $\theta v \geq 0$. Using this inequality in (14), and computing the resulting (elementary) integral, we obtain the bound

$$G'_S(u) \leq \frac{1}{(e-1)} \frac{e^{-\theta u} - e^{\frac{\gamma}{b}u}}{\theta^{-1} + (b/\gamma)} \leq \frac{1}{(e-1)} \frac{1 - e^{\frac{\gamma}{b}u}}{\theta^{-1} + (b/\gamma)}.$$

Rearranging and using the definition of b' we see that

$$b' \leq \frac{b}{\gamma} \log\left[(e-1)\Theta\left(\theta^{-1} + b/\gamma\right) + 1\right]. \quad (15)$$

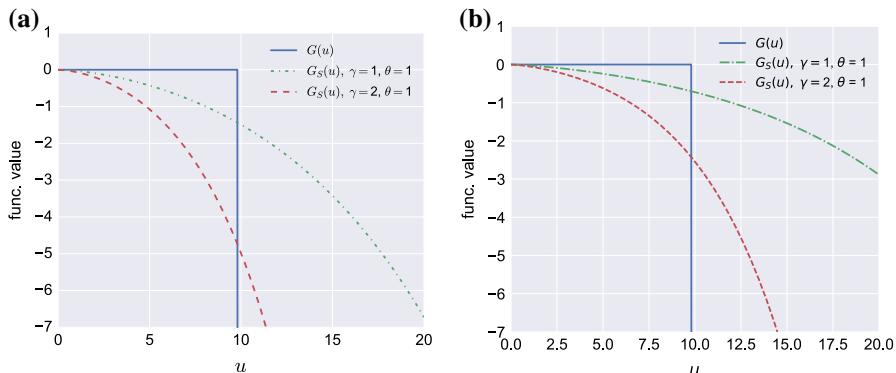


Fig. 1 Examples of G_S given in (14) for different values of γ and θ . **a** Examples of G_S when $h(u) = \log(u+1)$. **b** Example of G_S when $h(u) = 1 - \frac{1}{u+1}$

Since $\gamma \geq 1$, we can ensure that there is no budget violation, i.e., $b' \leq b$, by choosing $\gamma \geq \log[(e-1)\Theta(\theta^{-1} + b) + 1]$. Figure 1a shows examples of G_S for two values of γ when $h(u) = \log(1+u)$.

Example 3 (A-optimal experiment design) When $h(u) = 1 - \frac{1}{1+u}$, and so $h'(u) = (1+u)^{-2}$, it is again possible to express the optimal G'_S in terms of exponential integrals. Using the same strategy as example 2, but with the bound $h'(\theta v) = (1+\theta v)^{-2} \geq e^{-2\theta v}$ (for $v \geq 0$), gives

$$G'_S(u) \leq \frac{1}{(e-1)} \frac{1 - e^{\frac{\gamma}{b}u}}{\theta^{-1} + (2b/\gamma)} \text{ and } b' \leq \frac{b}{\gamma} \log \left[(e-1)\Theta(\theta^{-1} + 2b/\gamma) + 1 \right].$$

Figure 1b shows examples of G_S for two values of γ when $h(u) = 1 - \frac{1}{1+u}$.

3.2 Smoothing h

We are now in a position to design h_S , a smoothed surrogate for h . Recall that, for a fixed choice of $\gamma \geq 1$, we can use (14) to design an optimal G'_S . This gives a bound b' on the budget consumed by Algorithm 2. We design h_S by solving the following optimization problem over β and the function h_S :

$$\text{minimize } \beta \text{ subject to } \begin{cases} \gamma h_S(u) \leq h^*(h'_S(u)) + \beta h(u) & \forall u \in [0, u_{\max}] \\ h_S \text{ satisfies PSD-DR.} \end{cases} \quad (16)$$

Note that this problem comes directly from Lemma 3 and the fact that for fixed γ , we can maximize the bound on the competitive ratio, $(\gamma/(e-1) + \beta)^{-1}$, by minimizing β .

The key challenge in solving this optimization problem is imposing the PSD-DR property. A crucial observation in this paper is that, for a differentiable trace function,

satisfying PSD-DR can be expressed in an equivalent, more explicit form. This is an easy consequence of *Löwner's theorem* for matrix monotone functions [18], an important result in matrix analysis.

Proposition 2 Suppose $H_S(U) = \sum_{i=1}^n h_S(\lambda_i(U))$ where h_S is a monotone increasing, concave function, and $h_S(0) = 0$. Then H_S is PSD-DR for all n , if and only if there exists a positive measure μ supported on $[0, 1]$ such that

$$h_S(u) = \int_0^u y(u') du' \quad \text{where} \quad y(u) = \int_0^1 \frac{1}{u\lambda + (1-\lambda)} d\mu(\lambda). \quad (17)$$

Proof This follows from [18, Theorem 4.9]. \square

This alternative description allows us to impose the constraint that h_S satisfies PSD-DR in the optimization problem for designing h_S . We now rewrite this optimization problem in a more computationally useful form. We use the change of variable $h_S(u) = \int_0^u y(s) ds$, introduce μ from (17) as a decision variable, and express y in terms of μ . Doing so we obtain the following optimization problem.

$$\begin{aligned} & \text{minimize}_{\beta, y, \mu} \quad \beta \\ & \text{subject to} \quad \gamma \int_0^u y(s) ds - h^*(y(u)) \leq \beta h(u) \quad \forall u \in [0, u_{\max}] \\ & \quad y(t) = \int_0^1 \frac{1}{t\lambda + (1-\lambda)} d\mu(\lambda) \\ & \quad \mu \text{ a positive measure supported on } [0, 1]. \end{aligned} \quad (18)$$

We denote by $\beta(\gamma)$, the optimal value of (18) for a given γ . We extend h_S linearly on \mathbf{R}_- to satisfy the technical assumption on the domain of h_S in Assumption (1).

Theorem 1 Suppose that G_S is defined as in (14), and H_S is the trace function corresponding to $\int_0^u y(s) ds$, where y is optimal for (18). Then the iterates $\tilde{x}_1, \dots, \tilde{x}_m$ of Algorithm 2 satisfy

$$H\left(\sum_{t=1}^m A_t \tilde{x}_t\right) \geq \frac{1}{\gamma/(e-1) + \beta(\gamma)} D^* \quad \text{and} \quad \sum_{t=1}^m c_t \tilde{x}_t \leq G_S'^{-1}(-h'(0)\Theta)$$

where $\Theta \geq c_t^{-1} \mathbf{tr}(A_t)$ for all t and $\beta(\gamma)$ is the optimal value of β in (18). Moreover, if H satisfies the PSD-DR property, then $\beta(\gamma) \leq \gamma + 1$.

Proof The proofs of the first two inequalities follow directly from Lemma 3. To see that $\beta(\gamma) \leq \gamma + 1$ when H satisfies the PSD-DR property, one can note that $y(u) = h'(u)$ and $\beta = \gamma + 1$ is a feasible solution for (18) in that case. \square

Note that if H_S satisfies PSD-DR, then H is a feasible solution to problem (18); however, it is not necessarily the optimal solution. As we will see in the next section, the objective function in D-optimal experiment design, $H(U) = \log \det(I + U)$, provides such an example, where the competitive ratio improves as the result of smoothing even though H is PSD-DR. In the next section, we also consider the A-optimal experiment design where the objective function does not satisfy PSD-DR.

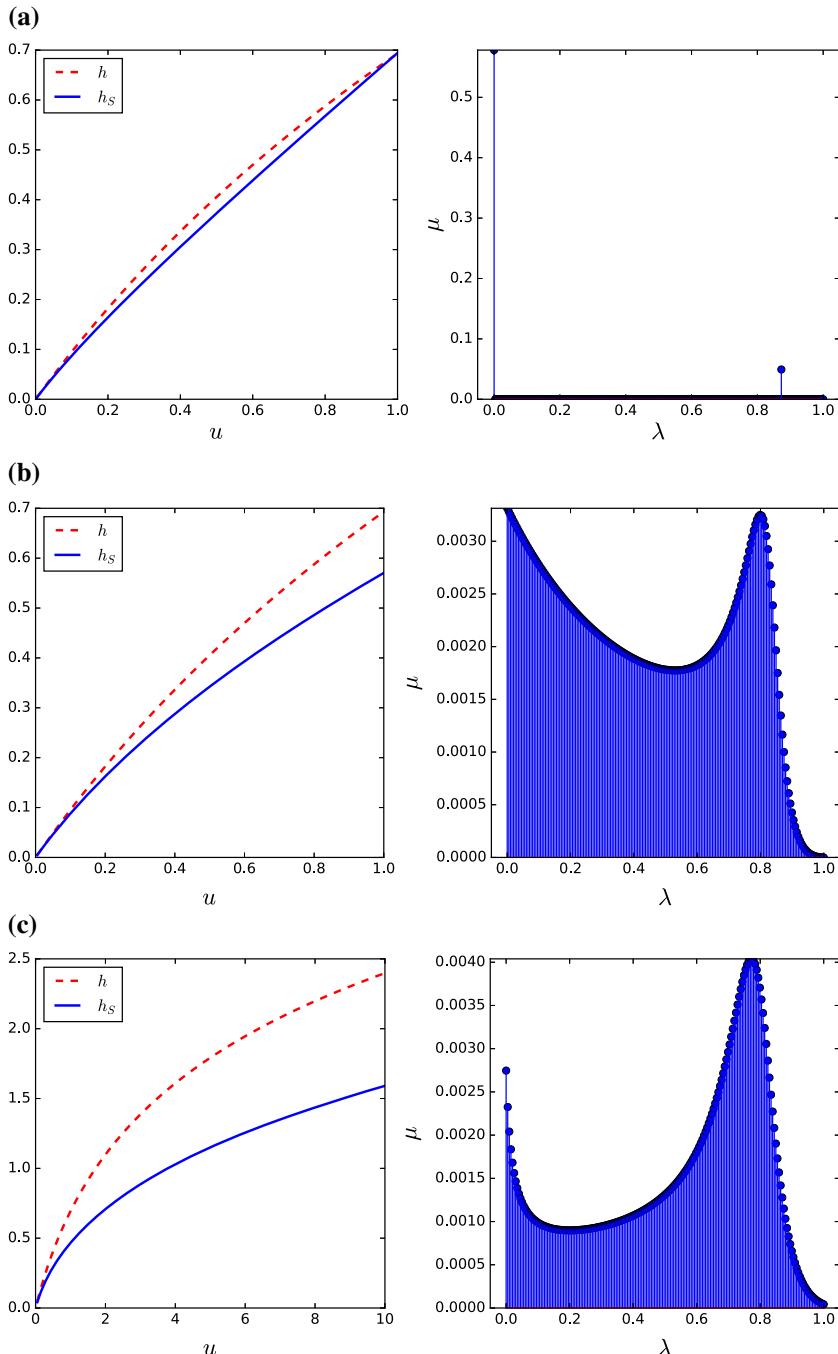


Fig. 2 The smoothed function h_S and the corresponding measure μ , when $h(u) = \log(u+1)$. The smooth h_S is found by solving the convex problem (18) via discretization of measure μ . In **a**, $\gamma = 1$ and $u_{\max} = 1$. In **b**, $\gamma = 4$ and $u_{\max} = 1$. In **c**, $\gamma = 4$ and $u_{\max} = 10$

3.2.1 Examples

Example 4 (Linear objective) When $h(u) = u$ then

$$h^*(y) = \begin{cases} 0 & \text{if } y = 1 \\ -\infty & \text{otherwise.} \end{cases}$$

Therefore, the only feasible solution to (16) is $h_S(u) = u$. This is a PSD-DR function. The corresponding value of β is $\beta = \gamma$. As such, the competitive ratio achieved by Algorithm 2 is at least $\frac{1}{\gamma} (1 - \frac{1}{e})$.

Example 5 (D-optimal experiment design) In this case $h(u) = \log(u + 1)$, i.e., $H(U) = \log \det(I + U)$. Note that H satisfies the PSD-DR property, because we can write $h'(u) = 1/(1+u)$ in the form of (17) by choosing the measure $\mu(\lambda)$ to have mass $1/2$ at $\lambda = 1/2$. By putting $h_S(u) = h(u)$, we can conclude that $\beta(\gamma) \leq 1 + \gamma$. This means that the competitive ratio achieved by Algorithm 2 is at least $\frac{1}{\gamma(1-1/e)^{-1}+1}$.

By solving (18) computationally, we can design an h_S that achieves a better competitive ratio. Figure 2a shows the solution of (18) for $\gamma = 1$ over the finite horizon $[0, u_{\max}] = [0, 1]$. Figure 2b, c show the solution of (18) for $\gamma = 4$ and $u_{\max} = 1$ and $u_{\max} = 10$ respectively. We note that the optimal measures are quite complicated—in the case $\gamma = 1$ the optimal measure seems to be atomic, whereas in the case $\gamma = 4$, it has a qualitatively different structure. In Fig. 3a, we plot the competitive ratio bound of Theorem 1 versus γ , and compare it with the bound obtained without smoothing h .

We can see that γ captures the trade off between the budget violation and competitive ratio. A smaller γ gives a better competitive ratio at the expense of larger budget violation, quantified by the bound given in Example 2.

Example 6 (A-optimal experiment design) In this case H is not a PSD-DR function, so to obtain competitive ratio bounds by our method we must construct a PSD-DR surrogate H_S numerically. Figure 3b, shows the competitive ratio bound given by solving (18) for $h(u) = 1 - 1/(u + 1)$ for different values of the parameter γ . In Fig. 4, we have provided examples of h_S for different values of γ and u_{\max} .

4 Analysis of the sequential algorithm

To design the functions H_S and G_S for the sequential algorithm (Algorithm 1), we need additional information about the problem data. Since Algorithm 1 uses the dual variable from the *previous* time step to assign the primal variable, the maximum length of each primal step, which is captured by the following two parameters, plays a role in the competitive ratio. Let

$$\rho_1 \geq \max_t c_t, \quad \text{and} \quad \rho_2 \geq \max_t \lambda_{\max}(A_t). \quad (19)$$

The next lemma is an analogue of Lemma 3 but for the sequential algorithm.

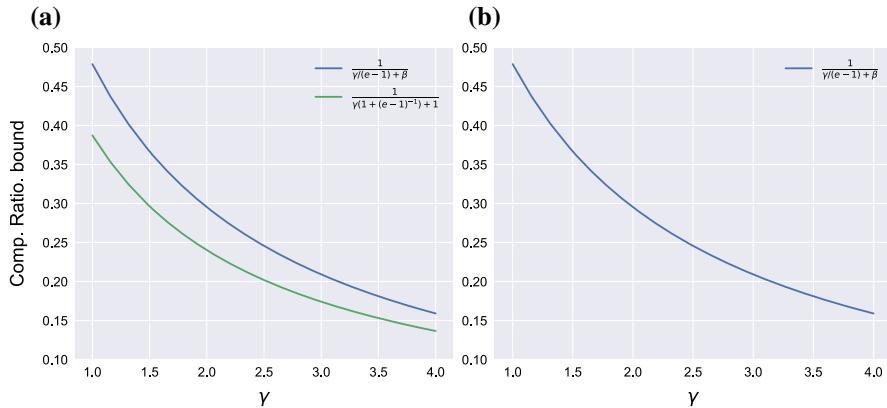


Fig. 3 **a** Two competitive ratio bounds vs γ for $h(u) = \log(u + 1)$. The first bound is achieved by numerically finding the optimum $\beta(\lambda)$ in (16), while the second bound uses the fact that $\beta(\lambda) \leq \lambda + 1$ when $h(u) = \log(u + 1)$. **b** Competitive ratio bound achieved by numerically finding the optimum $\beta(\lambda)$ for $h(u) = 1 - 1/(u + 1)$

Lemma 4 Let G_S satisfy Assumption 2, h_S satisfy Assumption 1, and suppose that the corresponding trace function H_S satisfies the PSD-DR assumption. Suppose that $\theta \leq c_t^{-1} \mathbf{tr}(A_t) \leq \Theta$ for all t , that ρ_1 and ρ_2 are defined as in (19), and $u_{\max} \geq \lambda_{\max} \left(\sum_{t=1}^m A_t \hat{x}_t \right)$.⁴ Then

1. If for a given $\gamma \geq 1$, G_S satisfies:

$$\gamma [G_S(u) - \rho_1 G'_S(u)] \leq G^*(G'_S(u)) + \frac{\gamma}{e-1} h(\theta u) \quad \forall u \in [0, \infty) \quad (20)$$

and there exists $\beta > 0$ such that h_S satisfy the following inequality:

$$\gamma [h_S(u) + \rho_2 (h'_S(0) - h'_S(u))] \leq h^*(h'_S(u)) + \beta h(u) \quad \forall u \in [0, u_{\max}] \quad (21)$$

then the iterates $\hat{x}_1, \dots, \hat{x}_m$ of Algorithm 1 satisfy

$$H \left(\sum_{t=1}^m A_t \hat{x}_t \right) \geq \frac{1}{\gamma/(e-1) + \beta} D^*. \quad (22)$$

2. The iterates $\hat{x}_1, \dots, \hat{x}_m$ of Algorithm 1 satisfy

$$\sum_{t=1}^m c_t \hat{x}_t \leq b' \quad \text{where} \quad b' := \rho_1 + \inf\{u : G'_S(u) \leq -h'(0)\Theta\}. \quad (23)$$

⁴ Note that we could choose, for instance, $u_{\max} = b' \max_t c_t^{-1} \lambda_{\max}(A_t)$, but for certain classes of problems better bounds may be available.

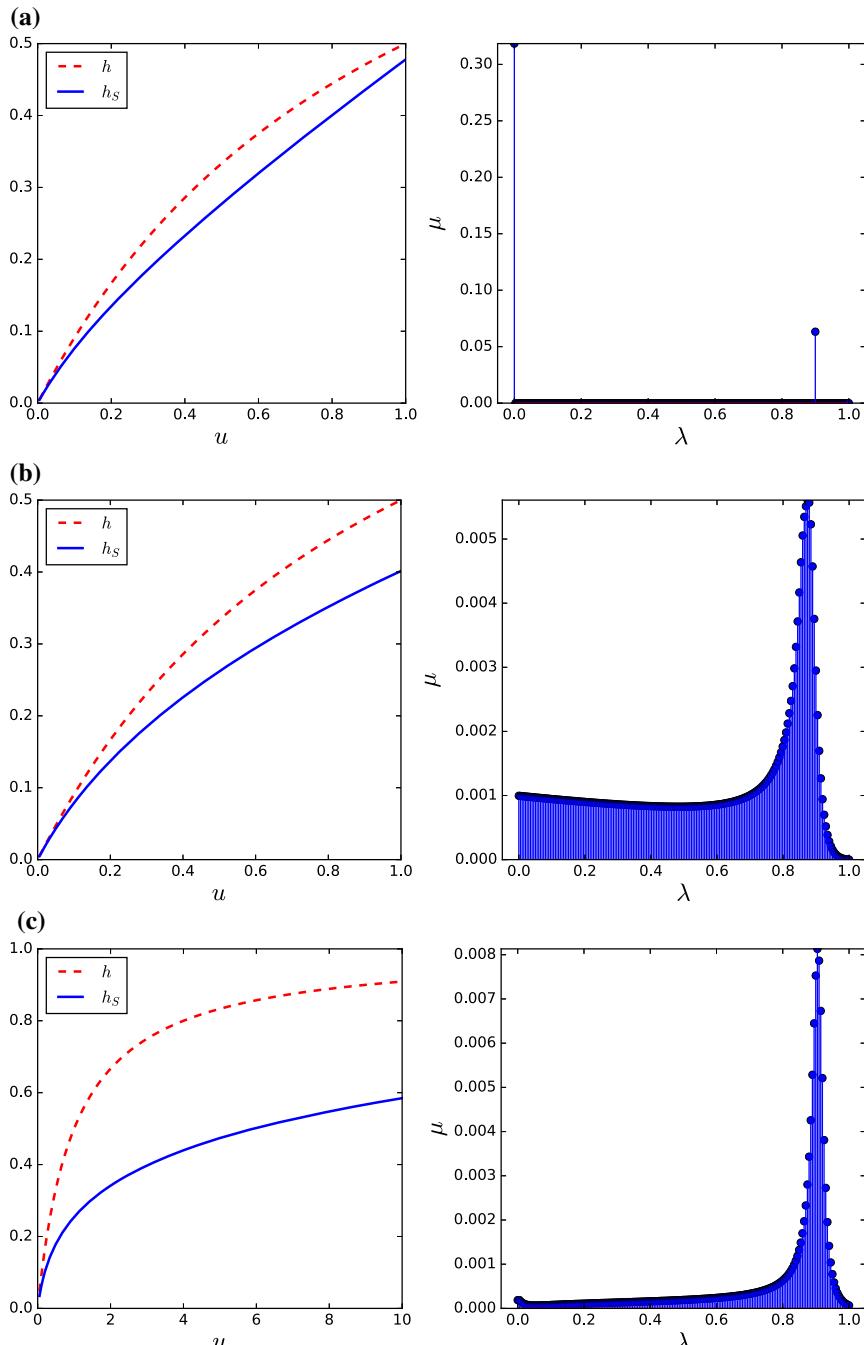


Fig. 4 The smoothed function h_S and the corresponding measure μ , when $h(u) = 1 - \frac{1}{u+1}$. The smooth h_S is found by solving the convex problem (18) via discretization of measure μ . In **a**, $\gamma = 1$ and $u_{\max} = 1$. In **b**, $\gamma = 4$ and $u_{\max} = 1$. In **c**, $\gamma = 4$ and $u_{\max} = 10$

Proof First we show that $\sum_{t=1}^m c_t \hat{x}_t \leq b'$. Arguing by contradiction, let T be the smallest index such that $\sum_{t=1}^{T+1} c_t \hat{x}_t > b'$. Then, $\sum_{t=1}^T c_t \hat{x}_t > b' - \rho_1$ (since $c_{T+1} \hat{x}_{T+1} \leq \rho_1$). We have that $\hat{z}_T = G'_S \left(\sum_{t=1}^T c_t \hat{x}_t \right) < -h'(0)\Theta$, where the inequality holds by the definition of b' . By the PSD-DR assumption, $\nabla H_S \left(\sum_{t=1}^T A_t \hat{x}_t \right) \preceq \nabla H_S(0) = h'_S(0)I$. Then

$$\langle \hat{Y}_T, A_{T+1} \rangle + c_{T+1} \hat{z}_T \leq c_{T+1} \left(h'_S(0) c_{T+1}^{-1} \mathbf{tr}(A_{T+1}) + \hat{z}_T \right) < 0.$$

It follows that $\hat{x}_{T+1} = 0$, contradicting our choice of T . Hence $\sum_{t=1}^m c_t \hat{x}_t \leq b'$.

We now bound on the competitive ratio. Let $U = \sum_{t=1}^m A_t \hat{x}_t$, $u = \sum_{t=1}^m c_t \hat{x}_t$, $\hat{Y}_m = \nabla H_S(U)$, and $\hat{z}_m = G'_S(u)$. First we note that

$$\begin{aligned} \sum_{t=1}^m \left[\langle A_t \hat{x}_t, \hat{Y}_{t-1} - \hat{Y}_t \rangle + c_t \hat{x}_t (\hat{z}_{t-1} - \hat{z}_t) \right] &\leq \sum_{t=1}^m \left[\rho_2 \mathbf{tr}(\hat{Y}_{t-1} - \hat{Y}_t) + \rho_1 (\hat{z}_{t-1} - \hat{z}_t) \right] \\ &= \rho_2 (\mathbf{tr}(\hat{Y}_0 - \hat{Y}_m)) - \rho_1 (\hat{z}_m) \end{aligned} \quad (24)$$

where the inequality holds because $\hat{Y}_t \preceq \hat{Y}_{t-1}$ (by the PSD-DR assumption), and, similarly, $\hat{z}_m \leq \dots \leq \hat{z}_0 = 0$ (since $G'_S(0) = 0$). By the primal allocation rule in Algorithm 1, we have $\hat{x}_t \left(c_t \hat{z}_{t-1} + \langle A_t, \hat{Y}_{t-1} \rangle \right) \geq 0$. Combining this with the concavity of H_S and G_S , we get

$$\begin{aligned} H_S \left(\sum_{s=1}^t A_s \hat{x}_s \right) + G_S \left(\sum_{s=1}^t c_s \hat{x}_s \right) - H_S \left(\sum_{s=1}^{t-1} A_s \hat{x}_s \right) - G_S \left(\sum_{s=1}^{t-1} c_s \hat{x}_s \right) \\ + \hat{x}_t \left(c_t (\hat{z}_{t-1} - \hat{z}_t) + \langle A_t, \hat{Y}_{t-1} - \hat{Y}_t \rangle \right) \geq 0. \end{aligned}$$

Taking the sum over t , telescoping, and using $h_S(0) = G_S(0) = 0$, gives

$$H_S(U) + G_S(u) + \sum_{t=1}^m \left[\langle A_t \hat{x}_t, \hat{Y}_{t-1} - \hat{Y}_t \rangle + c_t \hat{x}_t (\hat{z}_{t-1} - \hat{z}_t) \right] \geq 0. \quad (25)$$

Now, the proof follows the same step as the proof of Lemma 3 and uses (6) and the above inequalities.

$$\begin{aligned} H(U) - D_{\text{seq}} &\geq H(U) + H^*(\hat{Y}_m) - H_S(U) - G_S(u) + G^*(\hat{z}_m) \\ &\quad + \sum_{t=1}^m \hat{x}_t \left[\langle A_t, \hat{Y}_t - \hat{Y}_{t-1} \rangle + c_t (\hat{z}_t - \hat{z}_{t-1}) \right] \quad \text{By (6)} \\ &\geq \left[1 - \frac{\gamma}{e-1} \right] H(U) + H^*(\hat{Y}_m) - H_S(U) - (1-\gamma) G_S(u) - \gamma \rho_1 G'_S(u) \end{aligned}$$

$$\begin{aligned}
& + (\gamma + (1 - \gamma)) \sum_{t=1}^m \hat{x}_t \left[\langle A_t, \hat{Y}_t - \hat{Y}_{t-1} \rangle + c_t (\hat{z}_t - \hat{z}_{t-1}) \right] \quad \text{By (20) and (13)} \\
& \geq \left[1 - \frac{\gamma}{e-1} \right] H(U) + H^*(\hat{Y}_m) - H_S(U) + (1 - \gamma) H_S(U) - \gamma \rho_1 \hat{z}_m \\
& \quad + \gamma \sum_{t=1}^m \hat{x}_t \left[\langle A_t, \hat{Y}_t - \hat{Y}_{t-1} \rangle + c_t (\hat{z}_t - \hat{z}_{t-1}) \right] \quad \text{By (25)} \\
& \geq \left[1 - \frac{\gamma}{e-1} \right] H(U) + H^*(\hat{Y}_m) - \gamma \left[H_S(U) + \rho_2 \mathbf{tr}(\hat{Y}_0 - \hat{Y}_m) \right] \quad \text{By (24)} \\
& \geq \left[1 - \frac{\gamma}{e-1} - \beta \right] H(U) \quad \text{By (21).}
\end{aligned}$$

Applying Lemma 2 completes the proof. \square

For the sequential algorithm to minimize the bound b' on the budget consumption we choose G_S such that

$$G'_S(u) = -\frac{\theta\gamma}{(b + \rho_1\gamma)(e-1)} \int_0^u \exp\left(\frac{\gamma}{b + \rho_1\gamma}(u - v)\right) h'(v) dv. \quad (26)$$

To find h_S for the sequential algorithm, the problem (18) is modified to:

$$\begin{aligned}
& \text{minimize}_{\beta, y, \mu} \quad \beta \\
& \text{subject to } \gamma \int_0^u y(s) ds + \gamma \rho_2 (y(0) - y(u)) - h^*(y(u)) \leq \beta h(u) \quad \forall u \in [0, u_{\max}] \\
& y(t) = \int_0^1 \frac{1}{t\lambda + (1 - \lambda)} d\mu(\lambda) \\
& \mu \text{ a positive measure supported on } [0, 1].
\end{aligned} \quad (27)$$

Let $\beta(\gamma)$ be the optimal value of β in (27). Combining our arguments gives the following analogue of Theorem 1 for the sequential algorithm. It describes the tradeoff between budget consumption and competitive ratio achieved for the sequential algorithm when G_S and H_S are designed optimally.

Theorem 2 *If G_S is defined by (26), and H_S is the trace function corresponding to $\int_0^u y(s) ds$, where y is optimal for (27), then the iterates, $\hat{x}_1, \dots, \hat{x}_m$ of Algorithm 1 satisfy*

$$\sum_{t=1}^m c_t \hat{x}_t \leq G_S'^{-1}(-h'(0)\Theta) + \rho_1 \quad \text{and} \quad H\left(\sum_{t=1}^m A_t \hat{x}_t\right) \geq \frac{1}{\gamma/(e-1) + \beta(\gamma)} D^*$$

where $c_t^{-1} \mathbf{tr}(A_t) \leq \Theta$ for all t .

Proof The proof is the natural analogue of Theorem 1. \square

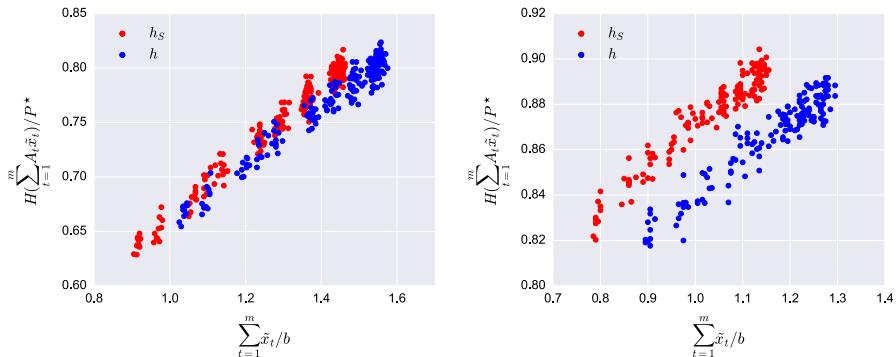


Fig. 5 Performance of Algorithm 2 (sequential updates) for online D-optimal and A-optimal experiment design with two choices of H_S . (Left) D-optimal experiment design. (Right) A-optimal experiment design

5 Numerical experiments

In this section, we provide the results of our numerical experiments and discuss the numerical implementation of smoothing design.

In our numerical simulations, we consider the D-optimal ($H(U) = \log \det(I + U)$) and the A-optimal ($H(U) = n - \text{tr}((I + U)^{-1})$) experiment design problems. The t^{th} matrix $A_t = a_t a_t^T$ is generated as follows: we sample a vector η uniformly at random from $\{-1, 1\}^n$ and then set $a_t = \frac{\sqrt{m-t+1}}{\sqrt{n}} \eta$. This is an *adversarial* weighting, inspired by worst-case examples for online LP, e.g., [8].

We compare the performance of Algorithm 2 (sequential updates) with two choices of H_S . In one case, we use the smoothed H_S given by solving (18). In the second case, we use H without smoothing ($H_S = H$). For both cases, we use the smooth G_S given in (14). (Note that a pure greedy algorithm without smoothing G simply picks the first b experiments and is a trivial algorithm, which we did not try.) We varied the parameter γ and plotted the competitive ratio versus the budget used by the algorithm in Fig. 5. For each value of γ , we have 10 random repeats.

Next, we briefly discuss the numerical implementation of the smoothing design problems for the simultaneous algorithm, introduced in Sect. 3. Similar ideas apply for the smoothing design problems for the sequential algorithm. We note that the algorithm does not require G_S itself, but only $G'_S(u)$. The formula for G'_S can be computed up to desired accuracy using Gauss-Leguerre quadrature [26].

To solve problem (18), we restrict μ to be an atomic measure supported on the $q + 1$ points $\lambda_j = j/q \in [0, 1]$ for $j = 0, 1, \dots, q$. The decision variables are then β and $\mu_j := \mu(\lambda_j)$ for $j = 0, 1, \dots, q$. Rather than imposing the constraint for all $u \in [0, u_{\max}]$ we impose it on a non-uniformly sampled subset. In particular, we sample u more densely where h has a larger local Lipschitz constant by choosing the discretization points to be $u_i = h^{-1}(i u_{\max}/d)$ for $i = 0, 1, \dots, d$. The optimal $y = h'_S$, which is all that is needed for the algorithm, is $y(u) = \sum_{j=0}^q \frac{\mu_j}{u\lambda_j + (1 - \lambda_j)}$. Note that since the integral of every individual function in the summand representation of h'_S satisfies PSD-DR, h_S satisfies PSD-DR.

6 Related work and discussion

6.1 Submodularity and experiment design

Algorithms for the *offline* optimal experiment design (both with and without integer constraints) have been extensively studied [25]. Let X_S denote a principal submatrix of X , then it is well known that the set function $S \mapsto \log \det(X_S)$ is submodular. Based on this, greedy subset selection is shown to have a $1 - 1/e$ approximation ratio [5, 28, 30] for the (integer) experiment selection problem. The paper [4] also gives an approximation ratio for the weakly submodular A-optimal design problem; also see [31] and references therein for statistical bounds for solving the convex relaxation followed by a greedy post-processing. In the existing literature, however, algorithms are assumed to have access to all possible experiments at the start, and therefore the result do not apply to the worst-case *online* setting that we consider. There is a connection between analysis of online algorithm under stochastic i.i.d setting and greedy algorithm for submodular maximization. We refer the reader to [12] for the details of this connection.

6.2 Online SDP problem of [16]

To the best of our knowledge, [16] is the only existing work that studies an online semidefinite program. While the problem considered is different from ours and its results do not apply to our setup, we briefly discuss the idea. [16] considers a generalization of the online *covering* linear program to the semidefinite cone as follows,

$$\text{minimize } b^T y \text{ subject to } \begin{cases} \mathcal{A}(y) \succeq C \\ y \geq 0, \end{cases}$$

where \mathcal{A} denotes a linear map from vectors to matrices. The algorithm receives a sequence of PSD matrices $C^{(0)} \preceq C^{(1)} \preceq \dots \preceq C^{(m)} = C$ over time, and needs to increase the variable y so as to satisfy the new matrix covering constraint. The dual of this problem is a packing problem, however the model for the online information the algorithm receives is still the sequence of matrices $C^{(i)}$ which is different from receiving a new experiment and bid price to decide on, as in our model. Indeed, extending the *packing* linear program to the PSD cone yields a different problem, i.e.,

$$\text{maximize } c^T x \text{ subject to } \begin{cases} \mathcal{A}(x) \preceq B \\ x \geq 0, \end{cases}$$

which is in line with the setting we consider in this paper, after a reformulation to bring the matrix constraint to the objective in a penalized form (which corresponds to H). The possibility of using Löwner's representation to design a matrix penalty function for this matrix-valued budget is an interesting direction for future work.

6.3 Online learning, regret, and FTRL

As mentioned earlier, the dual update in Algorithm 1 is the same as in Follow-the-Regularized-Leader (FTRL) algorithm with $-H_S^*$ as the regularizer. This primal-dual perspective has been used in [29] for design and analysis of online learning algorithms. In the online learning literature, the goal is to derive a bound on the *regret* that optimally depends on the horizon, m ; whereas in this work we study the competitive ratio for the algorithm that depends on the functions H and G . In order to optimize the competitive ratio, the regularization functions should be crafted based on H , and a general choice of regularization which yields an optimal regret bound in terms of m is not enough for a competitive ratio argument, so existing results in online learning do not address our aim. There are however some shared proof steps in the analysis that are worth exploring further.

Acknowledgements The authors thank Omid Sadeghi-Meibodi for helpful comments. The work of MF and RE was supported in part by Grants ONR N000141612789, NSF CCF 1409836, NSF Tripods 1740551, and ONR MURI N000141612710. Part of this work was done while RE and MF were visiting the Simons Institute for the Theory of Computing, partially supported by the DIMACS/Simons Collaboration on Bridging Continuous and Discrete Optimization through NSF Grant CCF-1740425.

A Additional proofs

Here, we provide additional proofs not given in detail in the body of the paper.

Proof (of Lemma 1) By the definition of D_{sim} , the definition of \tilde{x}_t , and the concavity of H_S and G_S , we have that

$$\begin{aligned} D_{\text{sim}} &= \sum_{t=1}^m \left[\langle A_t \tilde{x}_t, \tilde{Y}_t \rangle + c_t \tilde{x}_t \tilde{z}_t \right] - H^*(\tilde{Y}_m) - G^*(\tilde{z}_m) \\ &\leq \sum_{t=1}^m \left[H_S \left(\sum_{s=1}^t A_s \tilde{x}_s \right) - H_S \left(\sum_{s=1}^{t-1} A_s \tilde{x}_s \right) \right. \\ &\quad \left. + G_S \left(\sum_{s=1}^t c_s \tilde{x}_s \right) - G_S \left(\sum_{s=1}^{t-1} c_s \tilde{x}_s \right) \right] - H^*(\tilde{Y}_m) - G^*(\tilde{z}_m) \\ &= H_S \left(\sum_{s=1}^m A_s \tilde{x}_s \right) + G_S \left(\sum_{s=1}^m c_s \tilde{x}_s \right) - H^*(\tilde{Y}_m) - G^*(\tilde{z}_m). \end{aligned}$$

The inequality follows from concavity of G_S and H_S . The final equality holds by telescoping the sum and using the fact that $H_S(0) = 0 = G_S(0)$. For the sequential algorithm we can write:

$$D_{\text{seq}} = \sum_{t=1}^m \left[\langle A_t \hat{x}_t, \hat{Y}_{t-1} \rangle + c_t \hat{x}_t \hat{z}_{t-1} \right] - H^*(\hat{Y}_m) - G^*(\hat{z}_m)$$

$$\begin{aligned}
&= \sum_{t=1}^m \left[\langle A_t \hat{x}_t, \hat{Y}_t \rangle + c_t \hat{x}_t \hat{z}_t \right] - H^*(\hat{Y}_m) - G^*(\hat{z}_m) \\
&\quad + \sum_{t=1}^m \left[\langle A_t \hat{x}_t, \hat{Y}_{t-1} - \hat{Y}_t \rangle + c_t \hat{x}_t (\hat{z}_{t-1} - \hat{z}_t) \right]
\end{aligned}$$

Now, the rest follows similar to steps as the simultaneous case. \square

Proof (of Lemma 2) We write out the argument for the inequality $D_{\text{sim}} \geq D^*$. The argument showing that $D_{\text{seq}} \geq D^*$ is identical. We first show that the PSD-DR assumption on H_S implies

$$\sum_{t=1}^m \left(\langle A_t, \tilde{Y}_t \rangle + c_t \tilde{z}_t \right)_+ \geq \sum_{s=1}^m \left(\langle A_s, \tilde{Y}_m \rangle + c_s \tilde{z}_m \right)_+. \quad (28)$$

Since $A_s \in S_+^n$ and $\tilde{x}_s \geq 0$ for all $s \in [m]$, it follows that $\sum_{s=1}^t A_s \tilde{x}_s \preceq \sum_{s=1}^m A_s \tilde{x}_s$ for all $t \in [m]$. Since $\tilde{Y}_t = \nabla H_S(\sum_{s=1}^t A_s \tilde{x}_s)$, if H_S satisfies the PSD-DR assumption then $\tilde{Y}_t \succeq \tilde{Y}_m$ for all $t \in [m]$. By a similar argument, since G_S is concave, $\tilde{z}_t \geq \tilde{z}_m$ for all $t \in [m]$. Since $A_t \in S_+^n$ and $c_t \geq 0$ for all $t \in [m]$,

$$\langle A_t, \tilde{Y}_t \rangle + c_t \tilde{z}_t \geq \langle A_t, \tilde{Y}_m \rangle + c_t \tilde{z}_m \geq \langle A_t, \tilde{Y}_m \rangle + c_t \tilde{z}_m$$

for all $t \in [m]$. Taking the positive part and then summing establishes (28). To conclude that $D_{\text{sim}} \geq D^*$, we need only observe that D^* is a lower bound on the dual objective (4) evaluated at $(\tilde{Y}_m, \tilde{z}_m)$. \square

References

1. Agrawal, S., Devanur, N.R.: Fast algorithms for online stochastic convex programming. In: Proceedings of the Twenty-Sixth Annual Symposium on Discrete Algorithms, pp. 1405–1424. SIAM (2015)
2. Agrawal, S., Wang, Z., Ye, Y.: A dynamic near-optimal algorithm for online linear programming. Oper. Res. **62**(4), 876–890 (2014)
3. Azar, Y., Buchbinder, N., Chan, T.H., Chen, S., Cohen, I.R., Gupta, A., Huang, Z., Kang, N., Nagarajan, V., Naor, J., et al: Online algorithms for covering and packing problems with convex objectives. In: 2016 IEEE 57th Annual Symposium on Foundations of Computer Science (FOCS), pp. 148–157. IEEE (2016)
4. Bian, A.A., Buhmann, J.M., Krause, A., Tschiatschek, S.: Guarantees for greedy maximization of non-submodular functions with applications (2017). [arXiv:1703.02100](https://arxiv.org/abs/1703.02100)
5. Bouhouch, M., Gaubert, S., Sagnol, G.: Submodularity and randomized rounding techniques for optimal experimental design. Electron. Discret. Math. **36**, 679–686 (2010)
6. Buchbinder, N., Naor, J.: Improved bounds for online routing and packing via a primal-dual approach. In: 47th Annual IEEE Symposium on Foundations of Computer Science (FOCS’06), pp. 293–304. IEEE (2006)
7. Buchbinder, N., Naor, J.: Online primal-dual algorithms for covering and packing. Math. Oper. Res. **34**(2), 270–286 (2009)
8. Buchbinder, N., Naor, J.S.: The design of competitive online algorithms via a primal-dual approach Foundations and Trends®. Theor. Comput. Sci. **3**(2–3), 93–263 (2009)

9. Buchbinder, N., Jain, K., Naor, J.S.: Online primal-dual algorithms for maximizing ad-auctions revenue. In: Algorithms—ESA 2007, pp. 253–264. Springer, Berlin (2007)
10. Devanur, N.R., Jain, K.: Online matching with concave returns. In: Proceedings of the forty-fourth annual ACM symposium on theory of computing (STOC), pp. 137–144. ACM (2012)
11. Dragomir, S.S.: Some Gronwall Type Inequalities and Applications. Nova Science, New York (2003)
12. Eghbali, R.: Online algorithm design via smoothing with application to online experiment selection. Ph.D. thesis, The University of Washington (2017)
13. Eghbali, R., Fazel, M.: Designing smoothing functions for improved worst-case competitive ratio in online optimization. In: Advances in Neural Information Processing Systems, pp. 3279–3287 (2016)
14. Eghbali, R., Fazel, M.: Worst case competitive analysis of online algorithms for conic optimization (2016). [arXiv:1611.00507](https://arxiv.org/abs/1611.00507)
15. Eghbali, R., Swenson, J., Fazel, M.: Exponentiated subgradient algorithm for online optimization under the random permutation model (2014). [arXiv:1410.7171](https://arxiv.org/abs/1410.7171)
16. Elad, N., Kale, S., Naor, J.S.: Online semidefinite programming. In: LIPIcs-Leibniz International Proceedings in Informatics, Schloss Dagstuhl-Leibniz-Zentrum fuer Informatik, vol. 55 (2016)
17. Gupta, A., Molinaro, M.: How the experts algorithm can help solve LPs online. *Math. Oper. Res.* **41**(4), 1404–1431 (2016)
18. Hansen, F.: The fast track to Löwner's theorem. *Linear Algebra Appl.* **438**(11), 4557–4571 (2013)
19. Kapralov, M., Post, I., Vondrák, J.: Online submodular welfare maximization: greedy is optimal. In: Proceedings of the Twenty-Fourth Annual Symposium on Discrete Algorithms, pp. 1216–1225. SIAM (2013)
20. Legrain, A., Jaillet, P.: A stochastic algorithm for online bipartite resource allocation problems. *Comput. Oper. Res.* **75**, 28–37 (2016)
21. Lewis, A.S.: The convex analysis of unitarily invariant matrix functions. *J. Convex Anal.* **2**(1), 173–183 (1995)
22. Mehta, A., Saberi, A., Vazirani, U., Vazirani, V.: Adwords and generalized online matching. *J. ACM* **54**(5), 22 (2007)
23. Meng, D., Fazel, M., Mesbahi, M.: Online algorithms for network formation. In: IEEE Conference on Decision and Control (CDC), pp. 135–140 (2016)
24. Molinaro, M., Ravi, R.: The geometry of online packing linear programs. *Math. Oper. Res.* **39**(1), 46–59 (2013)
25. Pukelsheim, F.: Optimal Design of Experiments, vol. 50. SIAM, Philadelphia (1993)
26. Reiz, A.: On the numerical solution of certain types of integral equations. *Medd. Fran Lunds Astrono. Obs. Ser. I* **161**, 1–21 (1943)
27. Rockafellar, R.T., Wets, R.J.B., Wets, M.: Variational Analysis, vol. 317. Springer, Berlin (1998)
28. Seeger, M.: On the submodularity of linear experimental design. Tech. rep., Saarland University (2009)
29. Shalev-Shwartz, S., Singer, Y.: A primal-dual perspective of online learning algorithms. *Mach. Learn.* **69**(2–3), 115–142 (2007)
30. Shamaiah, M., Banerjee, S., Vikalo, H.: Greedy sensor selection: Leveraging submodularity. In: 49th IEEE Conference on Decision and Control (CDC), pp. 2572–2577. IEEE (2010)
31. Wang, Y., Yu, A.W., Singh, A.: On computationally tractable selection of experiments in regression models (2016). [arXiv:1601.02068](https://arxiv.org/abs/1601.02068)