Near-Optimal Discrete Optimization for Experimental Design *

Yining Wang[1], Zeyuan Allen-Zhu[2], Yuanzhi Li[3], and Aarti Singh[4]

[1]Machine Learning Department, Carnegie Mellon University
[2]Microsoft Research Redmond
[3]Department of Computer Science, Princeton University

July 11, 2018

Abstract

In this paper we consider computationally tractable methods for discrete optimization in experimental design, an important question in data mining and analysis when labeled data are expensive and difficult to obtain. We propose an efficient algorithm based on continuous relaxations and greedy swapping rounding techniques, which enjoy rigorous near-optimal approximation guarantees. We also test our algorithm on synthetic and real-world 3D lightweight structure design problems and demonstrate the practical effectiveness of our proposed method.

Keywords: experimental design, combinatorial optimization, continuous relaxation, greedy algorithm, spectral sparsification, approximation algorithm.

1 Introduction

Data are becoming increasingly important in scientific and computing disciplines. On the other hand, collecting labels on all possible data points is usually expensive or even infeasible in many applications. For example, in 3D lightweight structure design the evaluation of stress maps of external forces applied at certain surface locations requires computer simulation of finite-element analysis and could be very time consuming (Ulu et al., 2017). In another application of low-temperature microwave-assisted thin film crystallization (Reeja-Jayan et al., 2012; Nakamura et al., 2017), the quality of the crystalized film depends crucially on the experimental environments, and it could take days to actually perform the crystallization and measure the quality of the outcome for a particular experimental setup. In fMRI image analysis, experimental design could be applied to design visual stimuli that most efficiently invoke certain neural responses (Leeds et al., 2014; Leeds & Tarr, 2016).

As data labels are difficult to obtain, it is of vital importance in the above mentioned types of data analytical problems to design principled methods to *select*, from a large unlabeled data points pool, a small subset of design points on which labels are to be obtained. Such ***experimental design***

*Full version available at https://arxiv.org/abs/1711.05174.

problems are extensively studied in statistics, machine learning and operations research. We refer the readers to Sec. 1.2 for a brief summary of existing works on this classical problem.

Mathematically, let $x_1, \cdots, x_n$ be *known* p-dimensional unlabeled design points, and $k \ll n$ be a "budget parameter" indicating how many design points should be selected for labeling. Experimental design can then be formulated as a discrete (combinatorial) optimization problem:

$$\min_s F(s) = \min_s f\left(\sum_{i=1}^n s_i x_i x_i^\top\right) \quad s.t. \;\; s_i \in \{0,1\}, \sum_{i=1}^n s_i \leqslant k. \tag{1}$$

Here, $f : \mathbb{S}_+^d \to \mathbb{R}^+$ is an objective that maps d-dimensional positive definite matrices to positive real numbers. The objective f is usually referred to as *optimality criteria* in the experimental design literature (Pukelsheim, 2006; Fedorov, 1972) that reflects certain types of statistical efficiency of a design subset $\{x_i : s_i = 1\}$. We give a list of popular optimality criteria in Secs. 1.1, 1.2 and explain their motivations. For notational simplicity, we also use $X \in \mathbb{R}^{n\times p}$ to denote the $n \times p$ matrix X by stacking all unlabeled design points together.

In this paper we consider computationally efficient approaches to approximately solve the discrete optimization problem in Eq. (1) for experimental design. Our algorithmic framework can be roughly divided into two steps. In the first step, we consider a *continuous relaxation* of the discrete optimization problem, which can be solved efficiently using conventional convex continuous optimization methods. We then proceed to design a *greedy swapping* algorithm based on a novel spectral-related potential function motivated by online matrix games. Detailed descriptions of our proposed algorithmic framework and analysis of its properties are given in Secs. 2 and 3.

Our proposed algorithm enjoys three advantages. First, it is computationally efficient, as it is theoretically polynomial-time and in practice the algorithm also converges fast by using the entropic mirror descent method (Beck & Teboulle, 2003), as shown in Sec. 2.3 and 4. Second, our method enjoys rigorous theoretical approximation ratio guarantees, achieving $(1+\varepsilon)$-relative approximation provided that $k = \Omega(p/\varepsilon^2)$, which is proven not improvable for general objectives and continuous relaxation based methods (Nikolov et al., 2018). Finally, our algorithm was shown to work very effectively on a real-world 3D lightweight structural design problem (Ulu et al., 2017; Wang et al., 2018), improving over baseline sampling methods by a large margin.'

1.1 Popular choices of objectives

Below we list several popular choices of the objective functions f:

- A-optimality (**Average**): $f_A(\Sigma) = \frac{1}{p}\mathrm{tr}(\Sigma^{-1})$;
- D-optimality (**Determinant**): $f_D(\Sigma) = (\det|\Sigma|)^{-1/p}$;
- T-optimality (**Trace**): $f_T(\Sigma) = p/\mathrm{tr}(\Sigma)$;
- E-optimality (**Eigenvalue**): $f_E(\Sigma) = \|\Sigma^{-1}\|_{\mathrm{op}} = \lambda_{\max}(\Sigma^{-1})$;
- V-optimality (**Variance**): $f_V(\Sigma) = \frac{1}{n}\sum_{i=1}^n x_i^\top \Sigma^{-1} x_i$;
- G-optimality: $f_G(\Sigma) = \max_i x_i^\top \Sigma^{-1} x_i$.

The first four objectives (A, D, T, E) concerns the quality of estimating an unknown linear model, and the last two objectives (V, G) are more associated with prediction performance of the

estimated models. We refer the readers to (Pukelsheim, 2006) for a complete list and discussion of various optimality criteria used in the experimental design literature. Note that the optimality criteria in the above list are "normalized" (by multiplying or raising to the power of $1/p$) so that the objective values do not depend heavily on the problem dimension p.

1.2 Related works

Experimental design is an old topic in statistics (Pukelsheim, 2006; Fedorov, 1972; Chaloner & Verdinelli, 1995). Computationally efficient experimental design algorithms (with provable guarantee) are, however, a less studied field.

Perhaps the most well-studied optimality criterion is D-optimality $f_D(\Sigma) = \det(\Sigma)^{1/p}$ (see also Sec. 2.1), whose negative logarithm (i.e., $\log\det\Sigma$) is *submodular*, a property that sometimes gives rises to $1 - 1/e$ approximation ratio using pipage rounding (Ageev & Sviridenko, 2004). Unfortunately, $\log\det\Sigma$ can be negative and thus pipage rounding could fail to provide a constant relative approximation ratio with respect to $\det(\Sigma)$ or $\det(\Sigma)^{1/p}$. In (Bouhtou et al., 2010), Bouhtou et al. proposed to maximize a function $h(\Sigma) := \frac{1}{p}\mathrm{tr}(\Sigma^q)$ for $q \in (0, 1]$, and it satisfies $\lim_{q\to 0}(h(\Sigma))^{-1/q} = f_D(\Sigma)$. They showed that $h(\Sigma)$ is submodular and gave a $(1 - 1/e)$ approximation to $h(\Sigma)$ for every $q \in (0, 1]$ using pipage rounding. This does not translate to any bounded approximation ratio for $f_D(\Sigma)$ because $(1 - 1/e)^{-1/q}$ is unbounded when q approaches zero.

Summa et al. (2015) gave a polynomial-time algorithm for a related maximum volume simplex (MVS) problem in computational geometry with an $O(\log p)$ approximation ratio, which was later improved to $O(1)$ by Nikolov (2015); Nikolov & Singh (2016). Their results imply an e approximation ratio in the special case of $k = p$. On the other hand, Summa et al. (2015) showed that there exists a constant $c > 1$ such that polynomial-time c-approximation of the D-optimality is impossible for the $p = k$ case, unless P = NP. Therefore, additional assumptions on k are necessary for the $(1 + \varepsilon)$-approximation regime we consider in this paper.

Another well-studied optimality criterion is the A/V-optimality (f_A and/or f_V, as we define in Sec. 2.1), which is not submodular and hence pipage rounding is no longer relevant. Chamon & Ribeiro (2017); Bian et al. (2017) considered an alternative "approximate supermodular" formulation and derived a greedy algorithm with an $O(1)$ approximation ratio for the A/V-optimality. Their results, however, only apply to Bayesian experimental design settings and require the number of samples (k) to be lower bounded by a quantity that depends on the condition number of the original design, which might be unbounded.

For the A-optimality criterion, Avron & Boutsidis (2013) proposed a greedy algorithm with an approximation ratio $O(n/k)$ with respect to $f(\sum_{i=1}^n x_i x_i^\top)$. This ratio is tight for their algorithm in the worst case,[1] Li et al. (2017) further computationally accelerated this greedy algorithm, and achieved similar approximation guarantees.

Perhaps closest to this work, for A/V-optimality, Wang et al. (2017) introduced the idea of continuous relaxation followed by a variant of this greedy algorithm of Avron & Boutsidis (2013), and proved an approximation ratio quadratic in design dimension p and independent of pool size n. This result can also be turned into an $1 + \varepsilon$ approximation but requiring $k \geqslant \Omega(p^2/\varepsilon)$. For

[1] In the worst case, even the exact minimum $\min_{|S|\leqslant k} f(\sum_{i\in S} x_i x_i^\top)$ can be indeed $O(n/k)$ times larger than $f(\sum_{i=1}^n x_i x_i^\top)$ (Avron & Boutsidis, 2013) different from the subset selection objective in Eq. (1). This worst-case scenario may not always happen, but to the best our knowledge, their proof is tight in this worst case.

the same A/V-optimality criteria, Wang et al. (2017) derived another algorithm based on effective-resistance sampling (Spielman & Srivastava, 2011), that attains a $(1+\varepsilon)$ *pseudo*-approximation ratio if $k = \Omega(p \log p/\varepsilon^2)$. Specifically, their output set S is of cardinality $O(k)$ instead of k, and allows "with replacement" selection by considering constraints $s_i \in \mathbb{N}$ instead of $s_i \in \{0,1\}$ in Eq. (1). Wang et al. (2017); Chaudhuri et al. (2015); Dhillon et al. (2013) achieved $(1+\varepsilon)$ approximations but require the subset size k to be much larger than the condition number of X.

Finally, while preparing the journal version of this paper, we were informed of independent works of (Singh & Xie, 2017; Nikolov et al., 2018), which achieved $(1+\varepsilon)$-relative approximation for the A/D-optimality criteria under the weaker condition that $k = \Omega(p/\varepsilon + \log(\varepsilon^{-1})/\varepsilon^2)$. Their techniques are based on volume sampling of symmetric elementary functions of matrix eigenvalues, and are less likely to be extendable to general optimality criteria objectives. Indeed, Nikolov et al. (2018) proved a negative result showing that no continuous-relaxation based method can possibly attain $(1+\varepsilon)$ approximation for the E-optimaliy unless $k = \Omega(p/\varepsilon^2)$, essentially showing our theoretical analysis is the best possible one can hope for.

2 Preliminaries

2.1 Regularity conditions on objectives

We start with the definition of *regular* optimality criteria:

Definition 1 (Regular criteria)**.** *An optimality criterion $f : \mathbb{S}_p^+ \to \mathbb{R}$ is* regular *if it satisfies the following properties:*

1. Convexity*:* [2] *$f(\lambda A + (1-\lambda)B) \leqslant \lambda f(A) + (1-\lambda) f(B)$ for all $\lambda \in [0,1]$ and $A, B \in \mathbb{S}_p^+$;*
2. Monotonicity*: If $A \preceq B$ then $f(A) \geqslant f(B)$;*
3. Reciprocal multiplicity*: $f(tA) = t^{-1} f(A)$ for all $t > 0$ and $A \in \mathbb{S}_p^+$.*

It can be verified that all objective functions listed in Sec. 1.1 are regular.

The first condition ensures that the objective f (or one of its surrogate functions) is *convex*, implying that a *continuous relaxation* of the original discrete optimization problem is efficiently solvable using classical optimization methods for continuous convex programs.

The second condition in Definition 1 is natural in experimental design, because the sample covariance A is the *Fisher's information matrix* linear regression models and a design with sample covariance A can never achieve superior statistical efficiency than a design with B if $A \preceq B$, according to the Cramer-Rao information-theoretic limit (Van der Vaart, 2000).

The last condition in Definition 1 captures the linearity in experiment design, where a design with sample covariance tA is exactly t-times as efficient as a design with covariance A.

Also, for D-optimality the proxy function $g_D(\Sigma) = -\log\det(\Sigma)$ is considered to satisfy the convexity property.

[2] This property could be relaxed to allow a proxy function $g : \mathbb{S}_p^+ \to \mathbb{R}$ being convex, where $g(A) \leqslant g(B) \Leftrightarrow f(A) \leqslant f(B)$.

2.2 Continuous relaxation

A continuous relaxation to the discrete optimization problem in Eq. (1) takes the following form:

$$\pi^* \in \arg\min_{\pi} F(\pi) = \arg\min_{\pi_1,\cdots,\pi_n} \quad f\left(\sum_{i=1}^{n} \pi_i x_i x_i^\top\right) \quad s.t.\ \ 0 \leqslant \pi_i \leqslant 1,\ \ \sum_{i=1}^{n} \pi_i \leqslant k. \tag{2}$$

It should be noted that any feasible solution to the original discrete optimization problem (1) is also feasible for the continuously relaxed program in Eq. (2), and hence $F(\pi^*) \leqslant F(s^*)$ always holds. In addition, thanks to the first property in Definition 1, any regular optimality criterion f leads to a continuous program in Eq. (2) with a *convex* objective and a *convex and compact* feasibility region. In the next section we present a classical entropic mirror descent method that efficiently solves the continuous convex program in Eq. (2).

2.3 Entropic mirror descent

We first note that Eq. (2) can be re-formulated as

$$\min_{\omega} \widetilde{F}(\omega) := \min_{\omega} f\left(\sum_{i=1}^{n} \omega_i x_i x_i^\top\right) \quad s.t.\ \ 0 \leqslant \omega_i \leqslant 1/k,\ \ \sum_{i=1}^{k} \omega_i = 1, \tag{3}$$

by the change of variables $\omega_i = \pi_i/k$ and noting that the $\sum_{i=1}^{n} \pi_i \leqslant k$ constraint in Eq. (2) must bind, meaning that the optimal solution π^* must satisfy $\sum_{i=1}^{n} \pi_i^* = k$.

The *entropic mirror descent* (Beck & Teboulle, 2003) is a classical algorithm that takes into account the geometry of high-dimensional probabilistic simplex to efficiently solve constrained convex optimization problems. At a high level, entropic mirror descent uses the Kullbeck-Leibler (KL) divergence $\sum_i x_i \log(x_i/y_i)$ as the Bregman divergence, whose proximal operator can be evaluated in closed form as multiplicative weight updates.

We describe in Algorithm 1 how (projected) entropic mirror descent is applied to solve program 3. As our problem has an extra box constraint $\omega_i \leqslant 1/k$, we present in Appendix A a simple algorithm that computes such projection in $O(n \log n)$ time and the KL divergence. The projection algorithm is (in principle similar to but) much simpler than existing algorithms that compute projections onto simplex or L_1 balls (Duchi et al., 2008; Condat, 2015). We also list in Appendix B gradient calculations of popular optimality criteria f mentioned in Sec. 2.1.

3 Efficient rounding via greedy swapping

In the previous sections we discussed how to efficiently compute *fractional* solutions π^* to the continuously relaxed optimization program in Eq. (2). It remains "round" the fractional solution π^* to an integral one $\widehat{s} \in \{0,1\}^n$. Some intuitive methods such as thresholding or sampling do not yield good solutions, as discussed in (Wang et al., 2017).

In this section, we present our novel "greedy swapping" technique to round the fractional solution π^* efficiently with good theoretical approximation guarantees. Our procedure consists of a pre-processing whitening step, which reduces the problem to a least eigenvalue problem on a

Algorithm 1 The projected entropic mirror descent algorithm for solving Eq. (3).

Require: function $\min_\omega \widetilde{F}(\omega)$ defined in Eq (3); T number of iterations, step size rules $\{\eta_t\}$

$\omega^{(0)} = (1/n, \cdots, 1/n)$; ▷ initialization

for $t \leftarrow 0$ **to** $T-1$ **do**

  Compute subgradient $g^{(t)} \in \partial \widetilde{F}(\omega^{(t)})$; ▷ see Appendix B

  Update: $\omega_i^{(t+1/2)} \propto \omega_i^{(t)} \exp\{-\eta_t g_i^{(t)}\}$, normalized so that $\sum_{i=1}^n \omega_i^{(t+1/2)} = 1$;

  Projection: $\omega^{(t+1)} \leftarrow \textsc{BoxSimplexProject}(\omega^{(t+1/2)}, 1/k)$; ▷ see Appendix A

end for

return $\widehat{\omega} := \frac{1}{T}\sum_{t=0}^{T-1} \omega^{(t)}$.

whitened data set, and a greedy swapping algorithm with respect to a carefully designed potential function ϕ in Eq. (8) and suitable stopping conditions.

3.1 Pre-processing: whitening and minimum eigenvalue problems

Let $\pi^* = (\pi_1^*, \cdots, \pi_n^*)$ be the optimal *fractional* solution to the continuously relaxed program (2) and define $W := \sum_{i=1}^n \pi_i^* x_i x_i^\top$. The first (pre-processing) step of our algorithm is to *whiten* the design points $\{x_i\}_{i=1}^n$ by taking

$$\textbf{Pre-processing}: \qquad \widetilde{x}_i := W^{-1/2} x_i. \tag{4}$$

Note that after pre-processing, the transformed data points $\{\widetilde{x}_i\}_{i=1}^n$ satisfies $\sum_{i=1}^n \pi_i^* \widetilde{x}_i \widetilde{x}_i^\top = I_{p\times p}$, meaning that they are "whitened" to have identity sample covariance, and hence the name.

Our pre-processing step is motivated by the following observation, which shows that for regular criteria f, a lower bound on the least eigenvalue of $\sum_{i=1}^n \widehat{s}_i \widetilde{x}_i \widetilde{x}_i^\top$ implies a relative approximation guarantee on $F(\widehat{s})$.

$$\lambda_{\min}\left(\sum_{i=1}^n \widehat{s}_i \widetilde{x}_i \widetilde{x}_i^\top\right) \geqslant \tau \implies \left(\sum_{i=1}^n \widehat{s}_i x_i x_i^\top\right) \succeq \tau\left(\sum_{i=1}^n \pi_i^* x_i x_i^\top\right) \implies F(\widehat{s}) \geqslant \tau^{-1} F(\pi^*) \tag{5}$$

$$\implies F(\widehat{s}) \geqslant \tau^{-1} F(s^*) = \tau^{-1} \min_{s\in\{0,1\}^n, \|s\|_1 \leqslant k} F(s). \tag{6}$$

The above observation motivates the following minimum eigenvalue problem:

The minimum eigenvalue problem. Suppose $\pi^* \in [0,1]^n$, $\sum_{i=1}^n \pi_i^* = k$ and $\sum_{i=1}^n \pi_i^* \widetilde{x}_i \widetilde{x}_i^\top = I_{p\times p}$. Find $\widehat{s} \in \{0,1\}^n$, $\sum_{i=1}^n s_i \leqslant k$ such that

$$\lambda_{\min}\left(\sum_{i=1}^n \widehat{s}_i x_i x_i^\top\right) \geqslant (1-3\varepsilon)\cdot I. \tag{7}$$

As a corollary of Eq. (6), putting $\tau = 1-3\varepsilon$ for arbitrary $\varepsilon \in (0, 1/6]$, any $\widehat{s}$ satisfying Eq. (7) will also satisfy $F(\widehat{s}) \leqslant (1+6\varepsilon)F(s^*)$.

3.2 Potential functions

The main component of our proposed rounding algorithm is a carefully designed *potential* function $\phi(u, v; Z)$, which measures contributions to the least eigenvalue of a p-dimensional positive semi-definite matrix Z by *swapping* design points u and v.

Fix hyper-parameter $\alpha > 0$, whose values will be discussed in the next section. For any p-dimensional positive semi-definite matrix Z define $A_Z := (cI_{p\times p} + \alpha Z)^{-2}$ where $c \in \mathbb{R}$ is the unique real number such that $\mathrm{tr}(A_Z) = 1$. The potential function $\phi(u, v; Z)$ for any pairs of p-dimensional vectors $u, v \in \mathbb{R}^d$ are then defined as

$$\phi(u, v; Z) = \phi_+(u; Z) - \phi_-(v; Z) \tag{8}$$

where

$$\phi_+(u; Z) = \frac{u^\top A_Z u}{1 + 2\alpha u^\top A_Z^{1/2} u} \quad \text{and} \quad \phi_-(v; Z) = \frac{v^\top A_Z v}{1 - 2\alpha v^\top A_Z^{1/2} v}. \tag{9}$$

While the definitions of the potential function ϕ seems arbitrary, its form has deep roots in online matrix games. More specifically, the form of the intermediate matrix $A_Z = (cI + \alpha Z)^{-2}$, $\mathrm{tr}(A_Z) = 1$ corresponds to updates rules in a *Follow-The-Regularized-Leader (FTRL)* (McMahan, 2011) with the matrix $\ell_{1/2}$-regularizer $\psi(A) = -2\mathrm{tr}(A^{1/2})$, first considered by Allen-Zhu et al. (2015) for a related spectral sparsification problem. The potential function ϕ then falls naturally from a regret analysis of FTRL type policies in online matrix games, summarized in the following lemma:

Lemma 1. *For any p-dimensional vectors $\{u_t, v_t\}_{t=1}^T$ and fixed positive-semidefinite matrix Z_0, define $Z_t := Z_0 + \sum_{t'=1}^t u_{t'}u_{t'}^\top - v_{t'}v_{t'}^\top$. If further $v_t^\top A_{Z_{t-1}}^{1/2} v_t < 1/2\alpha$ holds for all t, then*

$$\lambda_{\min}(Z_T) \geqslant \sum_{t=1}^T \phi(u_t, v_t; Z_{t-1}) - \frac{2\sqrt{p}}{\alpha}. \tag{10}$$

Due to space constraints, we omit the proof of Lemma 1 and the readers are referred to Lemma 2.8 in the full-version of our paper (Allen-Zhu et al., 2017) and its associated proofs.

3.3 Greedy swapping and its approximation ratio

The lower bound of least eigenvalues in Lemma 1 immediately suggests a greedy swapping algorithm, which starts with an arbitrary subset $S_0 \subseteq [n]$ of size K and repeatedly find $i \in S_0$, $j \notin S_0$ for "swapping" so as to maximize $\phi(\widetilde{x}_j, \widetilde{x}_i; Z)$, where $Z = \sum_{\ell\in S} \widetilde{x}_\ell \widetilde{x}_\ell^\top$.

Detailed pseudo-codes of this greedy swapping procedure is given in Algorithm 2. Note that the pair $i_t \in S_{t-1}$, $j_t \notin S_{t-1}$ that maximizes $\phi(\widetilde{x}_{j_t}, \widetilde{x}_{i_t}; Z_{t-1})$ can be found in $O(n + k)$ instead of $O(nk)$ time by separately maximizing and minimizing $\phi_+(\widetilde{x}_{j_t}; Z_{t-1})$ and $\phi_-(\widetilde{x}_{i_t}; Z_{t-1})$ as shown in Steps 6 and 7 in Algorithm 2, because the potential ϕ decomposes additively. In step 5 of Algorithm 2, the unique real number $c_t \in \mathbb{R}$ such that $\mathrm{tr}(A_{Z_{t-1}}) = 1$ can be found by a binary search, because $\mathrm{tr}[(c_t I + \alpha Z_{t-1})^{-2}]$ is a monotonically decreasing function in c_t.

To further understand the validity and effectiveness of Algorithm 2, in light of eigenvalue lower bound in Lemma 1, the following three questions need to be addressed:

Algorithm 2 A swapping algorithm for rounding

Require: design points $\{x_i\}_{i=1}^n$, optimal fractional solution π^*, budget desired accuracy $k, \varepsilon > 0$.

1: $\alpha \leftarrow \sqrt{p}/\varepsilon$; ▷ configuration of hyper-parameters
2: Compute $\widetilde{x}_i = W^{-1/2} x_i$ where $W = \sum_{j=1}^n \pi_j^* x_j x_j^\top$; ▷ the whitening step
3: $S_0 \subseteq [n]$ an arbitrary subset of size k and $t \leftarrow 1$; ▷ initialization
4: **while** $\lambda_{\min}(\sum_{i \in S_{t-1}} \widetilde{x}_i \widetilde{x}_i^\top) \leqslant 1 - 3\varepsilon$ **do**
5: Compute $A_{Z_{t-1}} = (c_t I + \alpha Z_{t-1})^{-2}$ such that $\mathrm{tr}(A_{Z_{t-1}}) = 1$, where $Z_{t-1} = \sum_{i \in S_{t-1}} \widetilde{x}_i \widetilde{x}_i^\top$;
6: Find $i_t \in S_{t-1}, \widetilde{x}_{i_t}^\top A_{Z_{t-1}}^{1/2} \widetilde{x}_{i_t} < 1/2\alpha$ that minimizes $\phi_-(\widetilde{x}_{i_t}; Z_{t-1})$;
7: Find $j_t \notin S_{t-1}$ that maximizes $\phi_+(\widetilde{x}_{j_t}; Z_{t-1})$;
8: Swapping update: $S_t = S_{t-1} \cup \{j_t\} \backslash \{i_t\}$, and $t \leftarrow t + 1$;
9: **end while**
10: **return** $\widehat{s} \in \{0,1\}^n$ where $\widehat{s}_i = 1$ iff $i \in S_T$.

1. Whether one can always find $i_t \in S_{t-1}$ such that $\widetilde{x}_{i_t}^\top A_{Z_{t-1}}^{1/2} \widetilde{x}_{i_t} < 1/2\alpha$, making step 6 valid;
2. Whether $\phi(\widetilde{x}_{j_t}, \widetilde{x}_{i_t}; Z_t)$ remains sufficiently large such that the lower bound in Lemma 1 eventually approaches 1; and
3. Whether the while loop in Algorithm 2 terminates in a reasonable (polynomial) number of iterations and produce a good solution $\widehat{s}$.

In the following important lemma, we give affirmative answers to all questions above before the stopping condition in the while loop of Algorithm 2 is met, provided that k is sufficiently large compared to data dimension p, and the hyper-parameter α is set appropriately.

Lemma 2 (main averaging lemma)**.** *For every $\varepsilon \in (0, 1/6]$ and subset $S \subseteq [n]$ of cardinality k, suppose $\alpha = \sqrt{p}/\varepsilon$, $k \geqslant 5p/\varepsilon^2$ and $\lambda_{\min}(Z) \leqslant 1 - 3\varepsilon$, where $Z = \sum_{i \in S} \widetilde{x}_i \widetilde{x}_i^\top$. Then there exists $i \in S$ such that $\widetilde{x}_i^\top A_Z^{1/2} \widetilde{x}_i < 1/2\alpha$. Furthermore,*

$$\max_{j \notin S} \min_{i \in S, \widetilde{x}_i^\top A_Z^{1/2} \widetilde{x}_i < 1/2\alpha} \phi(\widetilde{x}_j, \widetilde{x}_i; S) \geqslant \frac{\varepsilon}{k}. \tag{11}$$

At a higher level, Lemma 2 shows that before the stopping condition $\lambda_{\min}(Z) \geqslant 1 - 3\varepsilon$ is met, the greedy swapping procedure is always well-defined and a steady ε/k rise in the potential function (corresponding to the lower bound of $\lambda_{\min}(Z)$ in Lemma 1) after each swap. Additionally, as a direct corollary of Lemmas 1 and 2, the while loop in Algorithm 2 terminates within $O(k/\varepsilon)$ iterations, after which the stopping condition $\lambda_{\min}(Z) \geqslant 1 - 3\varepsilon$ will be met.

Lemma 2 corresponds to Lemma 2.8 in the full version of our paper (Allen-Zhu et al., 2017), which is proved by careful case analysis and used the particular forms and properties of potential ϕ. Due to space constraints we will not give the proof of Lemma 2 in this paper, and interested readers should refer to Sec. 2.5 in (Allen-Zhu et al., 2017) for a complete proof.

The following theorem, combining our previous analysis, gives an explicit approximation ratio of our algorithm when k is not too small compared to p.

Theorem 1. *Suppose $k \geqslant 5p/\varepsilon^2$ for some $\varepsilon \in (0, 1/6]$. Then for any regular f, $\widehat{s} \in \{0,1\}^n$ output by Algorithm 2 has size $\sum_{i=1}^n \widehat{s}_i \leqslant k$ and satisfies*

$$F(\widehat{s}) \leqslant (1+6\varepsilon)F(s^*) = (1+6\varepsilon) \max_{s\in\{0,1\}^n, \|s\|_1 \leqslant k} F(s).$$

Theorem 1 shows that, under the condition of $k = \Omega(p/\varepsilon^2)$, our computationally efficient algorithm achieves $(1+\varepsilon)$-relative approximation for any regular optimality criteria f. We remark that the $k = \Omega(p)$ dependency is absolutely necessary, as $k \geqslant p$ is needed in order for the sample covariance $\sum_{i\in S} x_i x_i^\top$ to be invertible and therefore positive definite. In addition, the $k = \Omega(1/\varepsilon^2)$ dependency is also unavoidable for continuous relaxation based methods and general objectives f, as shown by Nikolov et al. (2018). For specific objectives such as the A- or D-optimality, it is possible to obtain improved condition between k and p using volume sampling based methods (Singh & Xie, 2017; Nikolov et al., 2018). It is an interesting open question whether our algorithmic framework can also be adapted to give improved results for special optimality criteria f.

4 Simulation results

We provide simulation results on synthetic data and compare the performance of our algorithm with several popular competitors.

4.1 Methods and Our Implementation

The choice $\alpha = \sqrt{p}/\varepsilon$ and the stopping rule in Algorithm 2 are backed by our theoretical proofs, and may be too pessimistic for practical usage. Therefore, we make the following slight changes. For the choice of α, we consider a grid of values $\alpha = \nu\sqrt{p}$ for $\nu = 0.2, 0.4, 0.6, 0.8, 1.0, 1.2, 1.4, 1.6, 1.8, 2.0, 2.5, 3.0, 4.0, 5.0$, and select the output $\widehat{s}$ that leads to the largest $\lambda_{\min}(\sum_i \widehat{s}_i x_i x_i^\top)$. For the stopping rule, we stop the algorithm whenever no $i \in S_{t-1}$ satisfies $2\alpha\langle A_t^{1/2}, x_i x_i^\top\rangle < 1$, or a consecutive of p iterations fail to improve the minimum eigenvalue of the current solution with non-singular Z.Finally, a safeguard is added to record the history of all solutions $\widehat{s}$ appeared throughout the iterations. The algorithm also terminates once the same solution is visited twice.

To find the relaxed continuous solution π, we use the projected entropic mirror descent Algorithm 1. We use backtracking line search[3] for *differentiable* objectives (e.g., $f_A(\Sigma) = \mathrm{tr}(\Sigma^{-1})/p$ and $g_D(\Sigma) = -1/p \cdot \log\det\Sigma$) and step length $\eta_t = \gamma_0/\sqrt{t+1}$ for *non-differentiable* objectives (e.g., $f_E(\Sigma) = \|\Sigma^{-1}\|_{\mathrm{op}}$), where γ_0 is chosen so that the algorithm does not overshoot too much. In practice, we start with $\gamma_0 = 0$ and half it (i.e., $\gamma_0 \leftarrow \gamma_0/2$) whenever $f(\omega^{(t+1)}) \geqslant 2f(\omega^{(t)})$. We stop the algorithm after 100 iterations for differentiable objectives, and after 1000 iterations for non-differentiable objectives.

We compare our proposed algorithm with several previous works listed below.

[3]In backtracking line search, for every iteration t a preliminary step size of $\eta_t = 1$ is used and the step size is repeatedly halved until the Armijo-Goldstein condition $f(\omega^{(t+1)}) \leqslant f(\omega^{(t)}) + 0.5\langle g^{(t)}, \omega^{(t+1)} - \omega^{(t)}\rangle$ is satisfied, where $\omega^{(t+1)}$ is the (projected) next step under step size η_t.

– *Uniform sampling (*UNIFORM*)*: sample k coordinates from $[n]$ uniformly at random without replacement. The sampling is repeated for 10 times and the best objective in the 10 samples is reported.

– *Weighted sampling (*WEIGHTED*)*: sample k coordinates from $[n]$ without replacement according to the distribution π^*/k, where π^* is the optimal (continuous) solution to Eq. (2). The sampling is repeated for 10 times and the best objective in the 10 samples is reported.

– *Fedorov's exchange (*FEDOROV*)*: the Fedorov's exchange algorithm (Fedorov, 1972; Miller & Nguyen, 1994) is a popular *heuristic* widely used in statistical computing of optimal designs. The algorithm starts with a random subset of k points and at each iteration selects a pair of points for exchange such that the objective f is minimized over all such changes. In our experiments we limit the maximum number of changes to 1000, or terminate the algorithm whenever no such exchanges improve the objective.

– *Greedy removal (*GREEDY*)*: the greedy removal procedure starts with the full set $S = [n]$ and removes one coordinate at a time so that the objective is minimized over all such single removals; the algorithm is accurate in most practical applications at the cost of quadratic running time in terms of n, making it less practical for large design pools.

  Note that, the greedy method has a provably guarantee for f_A and f_E criteria, but with a large $\frac{n-p+1}{k-p+1}$ factor approximation rate (Avron & Boutsidis, 2013). Its theoretical guarantees for other optimality criteria are unknown.

In the above list, we limit our attention to general-purpose algorithms that can (at least in practice) handle arbitrary optimality criteria, and skip methods that are designed specifically for certain objectives (e.g., submodular optimization for f_D and dual volume sampling (Avron & Boutsidis, 2013; Li et al., 2017) for f_A and f_D). We also only consider algorithms that can handle "frequentist" objectives which are infinity when $\Sigma = \sum_i \widehat{s}_i x_i x_i^\top$ is singular, thus excluding algorithms like (Chamon & Ribeiro, 2017; Bian et al., 2017) that require the objective f to be well-defined and finite-valued for all positive semi-definite matrices.

4.2 Data and Objectives

We synthesize a $n \times p$ design pool X as follows:

$$X = \begin{bmatrix} X_A & 0_{(n/2)\times(p/2)} \\ 0_{(n/2)\times(p/2)} & X_B \end{bmatrix},$$

where X_A is an $(n/2) \times (p/2)$ random Gaussian matrix, re-scaled so that the eigenvalues of $X_A^\top X_A$ satisfy a quadratic decay: $\sigma_j(X_A^\top X_A) \propto j^{-2}$; X_B is an $(n/2) \times (p/2)$ random Gaussian matrix, re-scaled so that the eigenvalues of $X_B^\top X_B$ satisfy a linear decay: $\sigma_j(X_B^\top X_B) \propto j^{-1}$. Such synthetic setting is carefully selected: the decay of the eigenvalues in the Gaussian designs mean that there are important design points x_i, and hence uniform sampling may not work well; on the other hand, the split of the two "signal" matrices X_A and X_B demands a careful balance between points allocated in A and B, because algorithms that focus solely on one set would produce close to singular designs and thus suffer high objective loss.

Table 1: Results on synthetic data with $n = 5000$ and $p = 50$. Numbers in brackets indicate the running time (in seconds) for each algorithm (omitted for UNIFORM, which does not take significant running time). Our proposed algorithm (Alg. 2) appears as SWAPPING. The running time for both WEIGHTED and SWAPPING takes into account the time for continuous optimization.

	f_A		f_D		f_E		f_V		f_G	
$k = 1.2p = 60$										
UNIFORM	32.1		3.89		40.0		53.2		300	
WEIGHTED	1.83	(*3.3*)	1.04	(*2.6*)	13.5	(*29*)	4.13	(*13*)	203	(*80*)
FEDOROV	9.68	(*0.0*)	6.86	(*0.0*)	Inf	(*0.0*)	94.4	(*0.0*)	26.1	(1.4×10^3)
GREEDY	1.27	(*9.5*)	0.88	(*9.6*)	4.46	(0.6×10^3)	3.24	(*10*)	17.9	(5.5×10^3)
SWAPPING	1.31	(*3.5*)	0.87	(*2.8*)	3.41	(*29*)	3.11	(*13*)	19.6	(*80*)
$k = 1.5p = 75$										
UNIFORM	8.05		3.91		109		17.9		155	
WEIGHTED	1.29	(*3.5*)	0.96	(*2.9*)	6.93	(*30*)	3.48	(*19*)	51.0	(*77*)
FEDOROV	25.1	(*0.0*)	Inf	(*0.0*)	41.7	(1.0×10^3)	3.2	(*0.0*)	25.0	(0.6×10^3)
GREEDY	1.20	(*9.7*)	0.87	(*9.8*)	4.15	(0.6×10^3)	2.97	(*10*)	16.8	(5.4×10^3)
SWAPPING	1.30	(*3.8*)	0.93	(*3.1*)	2.39	(*30*)	3.07	(*19*)	15.1	(*78*)
$k = 2p = 100$										
UNIFORM	6.34		4.17		18.0		15.4		127	
WEIGHTED	1.20	(*3.5*)	0.89	(*2.6*)	4.80	(*28*)	2.96	(*17*)	28.5	(*82*)
FEDOROV	1.33	(*1.9*)	0.92	(*2.6*)	16.7	(1.5×10^3)	3.06	(*2.4*)	17.1	(1.4×10^3)
GREEDY	1.16	(*9.3*)	0.86	(*9.6*)	3.32	(0.6×10^3)	2.85	(*10*)	13.0	(5.4×10^3)
SWAPPING	1.22	(*3.8*)	0.86	(*3.0*)	2.10	(*28*)	2.91	(*17*)	12.9	(*83*)
$k = 3p = 150$										
UNIFORM	5.03		3.41		22.3		12.9		105	
WEIGHTED	1.22	(*3.0*)	0.90	(*2.9*)	4.12	(*30*)	2.99	(*8.1*)	18.1	(*82*)
FEDOROV	1.20	(*6.2*)	0.97	(*7.0*)	12.9	(5.0×10^3)	3.25	(*4.8*)	18.7	(2.1×10^3)
GREEDY	1.15	(*9.4*)	0.88	(*9.6*)	3.08	(0.6×10^3)	2.87	(*10*)	12.4	(5.2×10^3)
SWAPPING	1.18	(*3.6*)	0.89	(*3.5*)	1.92	(*30*)	2.87	(*8.6*)	11.9	(*82*)
$k = 5p = 250$										
UNIFORM	4.76		3.30		11.6		10.9		67.7	
WEIGHTED	1.24	(*3.4*)	0.93	(*3.1*)	3.43	(*33*)	2.99	(*6.5*)	20.0	(*82*)
FEDOROV	5.99	(*0.0*)	3.11	(*0.0*)	11.5	(5.6×10^3)	3.52	(*8.8*)	18.6	(5.6×10^3)
GREEDY	1.21	(*9.7*)	0.92	(*9.9*)	2.93	(0.6×10^3)	2.97	(*10.5*)	12.6	(5.5×10^3)
SWAPPING	1.27	(*4.5*)	0.92	(*4.2*)	1.96	(*34*)	2.98	(*7.4*)	12.1	(*83*)

Five objectives are selected: the A-optimality $f_A(\Sigma) = \mathrm{tr}(\Sigma^{-1})/p$, the D-optimality $f_D(\Sigma) = \det \Sigma^{-1/p}$, the E-optimality $f_E(\Sigma) = \|\Sigma^{-1}\|_{\mathrm{op}}$, the V-optimality $f_V(\Sigma) = \mathrm{tr}(X\Sigma^{-1}X^\top)/n$ and the G-optimality $f_G(\Sigma) = \max \mathrm{diag}(X\Sigma^{-1}X^\top)$.

4.3 Results

In Table 1 we report performance of our continuous relaxation and the swapping algorithm, together with other competitors mentioned in Section 4.1. We also report the running time (in brackets) of each algorithm, except for the uniform sampling algorithm which finishes instantly on all data sets. Due to space constraints we only include the result for $n = 5000$ and $p = 50$, while additional simulation results can be found in the full version of our paper (Allen-Zhu et al., 2017).

The simulation results suggest that our algorithm (SWAPPING) consistently outperforms uniform sampling (UNIFORM), weighted sampling (WEIGHTED) and Fedorov's exchange algorithm (FEDOROV) for all experimental settings and objectives, especially in cases where k is close to p. Recall these are the cases when UNIFORM and WEIGHTED perform very badly due to statistical fluctuation of the sampling procedures.

Our swapping algorithm performs comparable or slightly worse than GREEDY. However, our algorithm is computationally efficient and can handle a wide range of objectives and input sizes. In contrast, the time complexity of the greedy algorithm scales quadratically or even cubically (e.g., the G-optimality) with the number of input points n and soon becomes intractable for intermediate-sized inputs (e.g., $n > 10^4$).

5 Application to lightweight structure design

We consider an application of our method to a 3D lightweight structure design problem. Most results in this section appeared in (Wang et al., 2018) with more details.

5.1 Background

3D lightweight structure design is the question of carefully distributing material mass in complicated 3D structures so that the resulting object has sufficient strength to withstand everyday use. An important task is then to quantify the structural performance of an object under the external forces it may experience during its use. Figure 1 from (Ulu et al., 2017) gives an intuitive illustration of the performance of structures under external forces applied at different locations, measured by stress distributed on the rest of the structure among which the *maximum* stress defines the performance of structures under given external forces.

Suppose external forces can be applied on n possible locations for a specific structure. For each location $i \in [n]$, the stress distribution as well as the maximum stress suffered by an unit amount of external force can be computed by an accurate yet time consuming *finite element analysis (FEA)* method. As each external force location $i \in [n]$ requires an independent FEA run, it is very desirable to select a few "representative" locations $S \subseteq [n]$, $|S| \leqslant k \ll n$ and *estimate* the maximum stress outcomes of the other force locations not selected in S.

This "location selection" problem fits well within the experimental design framework considered in this paper, and in the next two paragraphs we explain how to apply our developed algorithm

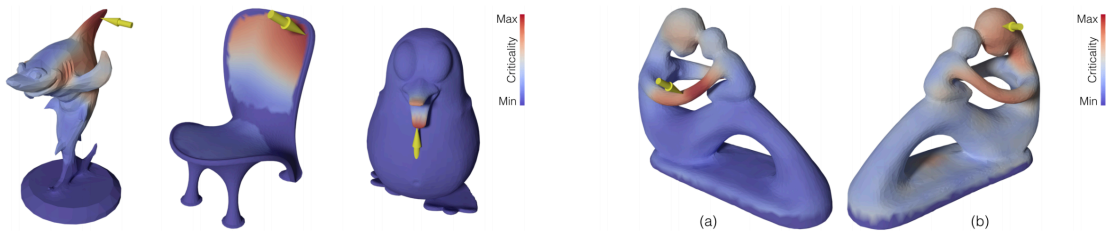


Figure 1: Stress distributions on designed structures under external forces.

as well as its experimental performances.

5.2 Method

Let G be a graph with n vertices, representing the spatial affinity of the n possible force locations on a structure surface. The readers are referred to (Ulu et al., 2017; Wang et al., 2018) for details of the construction of G. Let L be the graph Laplacian matrix of G, and $X \in \mathbb{R}^{n\times p}$ be the top-p eigenvectors of the graph Laplacian L. A linear regression model is used to model the maximum stress y_i induced by an unit external force applied at location $i \in [n]$ (corresponding to $x_i \in \mathbb{R}^d$ in the top eigenvectors matrix X), as

$$y_i = x_i^\top \beta_0 + \xi_i, \tag{12}$$

where β_0 is a p-dimensional unknown regression model and $\{\xi_i\}_{i=1}^n$ are noise variables.

To select a subset $S \subseteq [n]$, $|S| \leqslant k$ of locations, we use the algorithm proposed in the previous sections to solve the discrete optimization problem in Eq. (2), restated below:

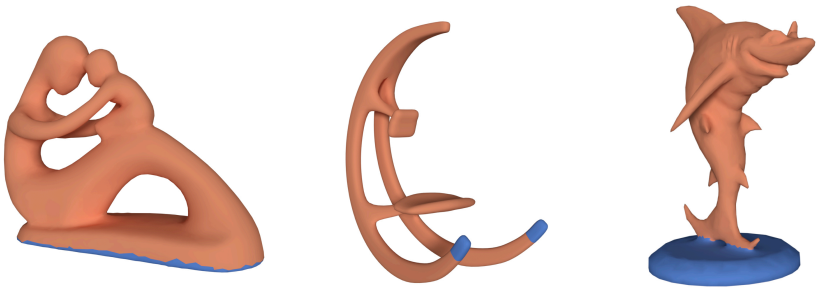
$$\min_s f\left(\sum_{i=1}^n s_i x_i x_i^\top\right) \quad s.t.\ \ s_i \in \{0,1\}, \sum_{i=1}^n s_i \leqslant k.$$

The selected subset S is then chosen as all locations with $s_i = 1$, and FEA analysis on these force locations is carried out to obtain their corresponding induced maximum stress y_i. The regression model β_0 is then estimated by ordinary least squares $\widehat{\beta} = (\sum_{i\in S} x_i x_i^\top)^{-1}(\sum_{i\in S} y_i x_i)$, and predictions on the other external force locations are produced by $\widehat{y}_i = x_i^\top \widehat{\beta}$ for $i \notin S$. The force locations $i \in [n]$ are then ranked in descending order according to $\{\widehat{y}_i\}_{i=1}^n$, and FEA analysis is computed again on the top ranked force locations to determine the final location $i^* \in [n]$ that yields the largest stress response y_{i^*}. More details of our algorithmic pipeline is given in (Wang et al., 2018).

5.3 Experimental settings

We evaluate the performance of our algorithm on three test structures (FERTILITY, ROCKINGCHAIR and SHARK) illustrated in Fig. 2. Descriptions and some basic statistics of the considered structures are given in (Wang et al., 2018).

In our experiments, we consider 5 methods to sample the force locations subset $S \subseteq [n]$, $|S| \leqslant k$. We compare our proposed algorithm (abbreviated as GREEDY) with baseline methods UNIFORM and LEVSCORE, as well as the previous work K-MEANS (Ulu et al., 2017) and SAMPLING (Wang et al., 2017) :



(a) Fertility (b) Rocking Chair (c) Shark

Figure 2: Example test structures with complex geometries. Fixed boundary conditions and contact regions are indicated in blue and red, respectively.

1. UNIFORM: The subset $S \subseteq [n]$ is obtained by sampling without replacement each force location $i \in [n]$ uniformly at random, until k samples are obtained;
2. LEVSCORE: The subset $S \subseteq [n]$ is obtained by sampling without replacement each force location $i \in [n]$ with probability proportional to its *leverage score*, defined as $x_i^\top (X^\top X)^{-1} x_i$, until k samples are obtained;
3. K-MEANS: The subset $S \subseteq [n]$ consists of k force locations such that the geodesic distance between the closest force locations in S is maximized. As the problem itself is NP-hard, the K-means (Lloyd's) algorithm is employed to get an approximate solution.
4. SAMPLING: The subset $S \subseteq [n]$ is obtained by sampling without replacement each force location $i \in [n]$ with probability π_i^* until k samples are obtained, where $\pi^* = (\pi_1^*, \cdots, \pi_n^*)$ is the optimal fractional solution of the continuously relaxed program (2).

The performance of an algorithm is evaluated by the smallest integer m required so that the top-m ranked lists according to $\{\widehat{y}_i\}_{i=1}^n$ include the external force location $i^* \in [n]$ that actually leads to the maximum stress a structure suffers.

5.4 Results and discussion

Table 2 reports the performance (m needed to cover $i^* \in [n]$ leading to maximum stress) of our algorithm and its competitors under variance k settings for all three different structures. It suggests that both the K-MEANS and the GREEDY algorithms outperform their competitors for most parameter settings. One important reason for the overall good performance of K-MEANS and GREEDY is their deterministic nature, which avoids poor designs due to statistical perturbations in the other randomized algorithms. Furthermore, the GREEDY algorithm remains accurate and robust even when k is very small (e.g., $k = 25$). For such a small k setting, the other methods require large m values to compensate for the prediction error. Therefore, the GREEDY algorithm can produce an accurate prediction of the overall maximum stress using much fewer number of total FEAs in both the first and the last stages of our algorithm pipeline, as shown by the rightmost columns in the tables. Our approach uses no more than 65 FEAs to successfully recover the maximum stress caused by worst-case external forces. In addition, when a 5% to 10% error tolerance is used, the number of FEAs can be further reduced to less than 40. This is close to a $100\times$ speed-up compared

Table 2: Results for the three test structures. Numbers in each cell are the smallest m such that the top ranked m force locations include the actual location $i^* \in [n]$ leading to maximal stress. Randomized algorithms (UNIFORM, LEVSCORE and SAMPLING) are run for 10 independent trials and the median performance is reported. Best performing settings are indicated in bold. Additional results available in (Wang et al., 2018).

	$k =$	25	50	100	150	200	250	300	Total FEAs $(k+m)$
Fertility	UNIFORM	316.5	149	78.5	37.5	98.5	42.5	39	178.5 ($k=100$)
	LEVSCORE	252.5	54.5	73.5	68.5	42.5	31	13.5	104.5 ($k=50$)
	K-MEANS	237	**25**	61	82	57	**17**	**16**	75 ($k=50$)
	SAMPLING	210.5	148.5	51	30	35.5	34	26.5	151 ($k=100$)
	GREEDY	**12**	26	**13**	**7**	**11**	25	**33**	**37** ($k=25$)
RockingChair	UNIFORM	716	857	385.5	42	135.5	269.5	**36**	192 ($n_{FL}=150$)
	LEVSCORE	764.5	208.5	**36**	**36**	**36**	**36**	**36**	136 ($k=100$)
	K-MEANS	4013	4400	4573	4301	4320	4620	4757	4038 ($k=25$)
	SAMPLING	672.5	282	38.5	38	38	**36**	**36**	138.5 ($k=100$)
	GREEDY	**36**	**35**	208	**35**	**36**	**36**	**36**	**61** ($k=25$)
Shark	UNIFORM	585	384	141.5	208.5	20	**9**	9.5	220 ($n_{FL}=200$)
	LEVSCORE	478.5	**9**	**9**	**9**	**9**	**9**	**9**	59 ($n_{FL}=50$)
	K-MEANS	133	102	**9**	**9**	**9**	**9**	**9**	109 ($n_{FL}=100$)
	SAMPLING	963.5	87	**9**	**9**	**9**	**9**	**9**	109 ($n_{FL}=100$)
	GREEDY	**9**	171	**9**	**9**	**9**	**9**	**9**	**34** ($n_{FL}=25$)

to the brute-force approach that performs FEA on the entire surface mesh. It also achieves a $5\times$ speed-up over the existing work Ulu et al. (2017) and is simpler to implement.

In Fig. 3, we plot the sub-sampled force locations (i.e., S) of our proposed algorithm for $k = 200$ point. We provide the samples obtained by the K-MEANS algorithm in Ulu et al. (2017) for comparison. The difference in the sampling patterns between GREEDY and K-MEANS are quite obvious from the figures. We explain the differences for the three structures separately:

- **Fertility**: The K-MEANS algorithm emphasizes the pairwise geodesic distance between sample points and thus places samples in a uniform fashion on the bodies, necks and heads of the structure. On the other hand, the GREEDY algorithm places more samples on the arms connecting the mother and the child, which are the most fragile parts of the structure. By placing more samples on these parts the learned linear model is more accurate in predicting the maximum stress, and therefore fewer FEAs are required.
- **RockingChair**: The GREEDY algorithm produces more samples on the top end of the body, the surface area of the smaller back support and the edges of the larger seat compared to the equally distanced K-MEANS design. External forces applied onto these parts of this structure are likely to cause increased stress, and therefore more samples placed around this region can greatly improve the regression model built for the maximum stress.
- **Shark**: As reported in Table 2, most of the sampling methods can predict the maximum stress using very small number of FEAs. However, if we focus on the sample points on the fins of the shark there are some noticeable differences between K-MEANS and GREEDY. While the GREEDY algorithm places more points around the tips, samples produced by the K-MEANS algorithm are relatively uniformly distributed on the surfaces. This subtle difference makes GREEDY algorithm more robust in prediction accuracy for small n_{FL} values.

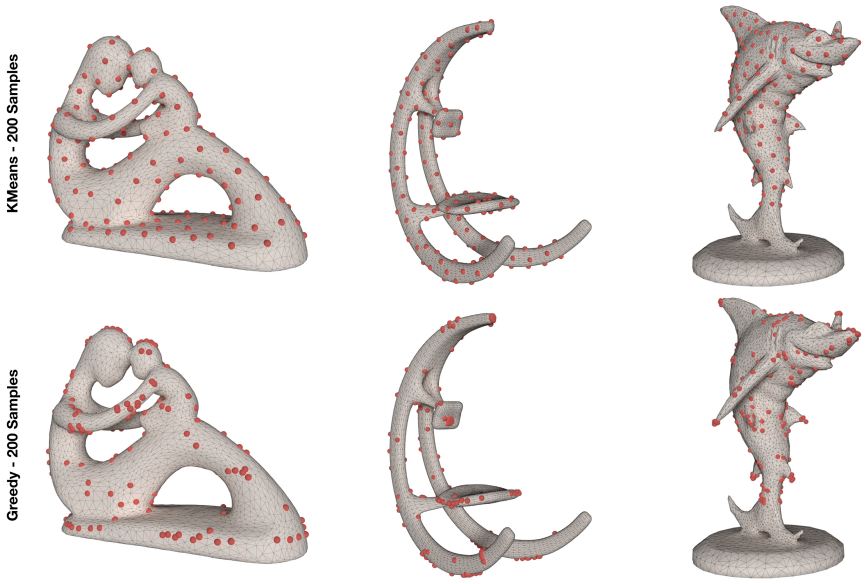


Figure 3: Sampled force locations (S) using the K-MEANS algorithm (top row) versus our proposed algorithm (bottom row), for $k = 200$. Results for smaller k values available in (Wang et al., 2018).

Despite the significant reduction in FEA time, one important limitation of the proposed algorithm is the lack of stopping criteria. In particular, the performance parameter k can only be evaluated once the ground-truth maximum stress is known. On the other hand, performance control in problems involving structural mechanics is of vital importance because designs with insufficient stress tolerance may actually fail in reality, leading to devastating consequences. In our examples, we empirically determined that $k = 40$ is sufficient for most of the k settings.

6 Extensions

We mention several extensions of our algorithmic framework and its analysis. A complete description of these extensions is available in the long versions of our papers (Wang et al., 2017; Allen-Zhu et al., 2017).

6.1 Generalized linear models

In a generalized linear model $\mu(x) = \mathbb{E}[Y|x]$ satisfies $g(\mu(x)) = \eta = x^\top \beta_0$ for some known link function $g : \mathbb{R} \to \mathbb{R}$. Under regularity conditions (Van der Vaart, 2000), the maximum-likelihood estimator $\widehat{\beta}_n \in \operatorname{argmax}_\beta \{\sum_{i=1}^n \log p(y_i|x_i;\beta)\}$ satisfies $\mathbb{E}\|\widehat{\beta}_n - \beta_0\|_2^2 = (1 + o(1))\mathrm{tr}(I(X,\beta_0)^{-1})$, where $I(X,\beta_0)$ is the Fisher's information matrix:

$$I(X,\beta_0) = -\sum_{i=1}^n \mathbb{E}\frac{\partial^2 \log p(y_i|x_i;\beta_0)}{\partial\beta\partial\beta^\top} = -\sum_{i=1}^n \left(\mathbb{E}\frac{\partial^2 \log p(y_i;\eta_i)}{\partial\eta_i^2}\right) x_i x_i^\top. \tag{13}$$

Here both expectations are taken over y conditioned on X and the last equality is due to the sufficiency of $\eta_i = x_i^\top \beta_0$. The experiment selection problem is then formulated to select a subset

$S \subseteq [n]$ of size k, either with or without duplicates, that minimizes $\mathrm{tr}(I(X_S, \beta_0)^{-1})$.

It is clear that the objective in Eq. (13) depends on the unknown parameter β_0, which itself is to be estimated. This issue is known as the *design dependence* problem for generalized linear models (Khuri et al., 2006). One approach is to consider *locally optimal designs* (Khuri et al., 2006; Chernoff, 1953), where a consistent estimate $\breve{\beta}$ of β_0 is first obtained on an initial design subset and then $\breve{\eta}_i = x_i^\top \breve{\beta}$ is supplied to compute a more refined design subset to get the final estimate $\widehat{\beta}$. With the initial estimate $\breve{\beta}$ available, one may apply transform $x_i \mapsto \widetilde{x}_i$ defined as

$$\widetilde{x}_i = \sqrt{-\mathbb{E}\partial^2/\partial\eta^2 \log p(y_i; \breve{\eta}_i)} \cdot x_i.$$

6.2 Bayesian designs

In some applications of experimental design, a *prior* is imposed on the regression model (Chaloner & Verdinelli, 1995). In cases where a Gaussian prior $\mathcal{N}(0, \lambda)$ is imposed and the homogeneous noise ξ_i is sub-Gaussian with parameter σ^2, the *Bayesian experimental design* problem can be formulated with objective $f_{\lambda,\sigma}(\Sigma) = f(\lambda/\sigma^2 \cdot I_{p\times p} + \Sigma)$. Here, $f(\cdot)$ is any optimality criterion we have introduced in the classical experimental design setting.

Such objectives $f_{\lambda,\sigma}$ are also known as *Bayesian alphabetical optimality* (Chaloner & Verdinelli, 1995) and are useful when the design budget k is close to the number of parameters p, or when classical experiment design yields ill-conditioned solutions. Our methods remain valid for Bayesian design problems, with the "reciprocal multiplicity" condition in Definition 1 replaced by a weaker "reciprocal sub-multiplicity" condition $f(tA) \leqslant t^{-1} f(A)$ for $0 < t < 1$.

7 Conclusion

In this paper we introduced a computationally efficient algorithmic framework for the question of discrete optimization in experimental design problems. Our algorithm, based on a continuous relaxation and a greedy swapping rounding technique, enjoys rigorous approximation guarantees that are near-optimal. Numerical results on synthetic and real-world structure design problems confirm the effectiveness of our proposed method.

Acknowledgement We thank Adams Wei Yu for helpful discussions regarding the implementation of the entropic mirror descent solver for the continuous (convex) relaxation problem, thank Aleksandar Nikolov, Shayan Oveis Gharan, and Mohit Singh for discussions on the references, and thank Erva Ulu, Levent Burak Kara for help with the 3D lightweight structure design data. This work is supported by NSF CCF-1563918, NSF CAREER IIS-1252412, AFRL FA87501720212 and NCDMM America Makes Project #4058.

Appendix A Projection operators

We give detailed procedures in Algorithm 3 for computing the projection step (onto the probabilistic simplex with box constraints) in step 5 of Algorithm 1. The box constraint parameter b is taken to be $1/k$ in our applications. The algorithm has time complexity $O(n \log n)$.

Algorithm 3 Projection onto the probabilistic simplex with box constraint

Require: $\omega \in \Delta_n$, parameter $b \in [1/n, 1]$. $\triangleright\ \Delta_n = \{x \in \mathbb{R}^n : x_i \geqslant 0, \sum_i x_i = 1\}$

Ensure: an output $\omega' \in \Delta_n$ such that $\|\omega'\|_\infty \leqslant b$. $\triangleright\ \omega' = \arg\min_{y \in \Delta_n, y_i \leqslant b} \mathrm{KL}(y\|\omega)$

$\triangleright$ where $\mathrm{KL}(y\|\omega) := \sum_i y_i \log \frac{y_i}{\omega_i}$

Sort ω in descending order: $\omega_1 \geqslant \omega_2 \geqslant \cdots \geqslant \omega_n > 0$;
if $\omega_1 \leqslant b$ **then return** ω;
$\mathrm{KL}_1 \leftarrow b\log(b/\omega_1)$, $Z_1 \leftarrow 1 - \omega_1$, $\mathrm{KL}_{\mathrm{opt}} \leftarrow \infty$, $\eta_{\mathrm{opt}} \leftarrow 0$, and $C_{\mathrm{opt}} \leftarrow 0$;
for $q \leftarrow 2$ to n **do**
    $C \leftarrow (1 - b(q-1))/Z_{q-1}$;
    if $C > 0$ and $Cw_q \leqslant b$ and $\big(\mathrm{KL}_{q-1} + C\log(C) \cdot Z_{q-1} \leqslant \mathrm{KL}_{\mathrm{opt}}\big)$ **then**
        $\mathrm{KL}_{\mathrm{opt}} \leftarrow \mathrm{KL}_{q-1} + C\log(C) \cdot Z_{q-1}$, $\eta_{\mathrm{opt}} \leftarrow \omega_q$, $C_{\mathrm{opt}} \leftarrow C$;
    end if
    $\mathrm{KL}_q \leftarrow b\log(b/\omega_q)$, $Z_q \leftarrow Z_{q-1} - \omega_q$;
end for
Set $\omega'_i \leftarrow b$ if $\omega_i \geqslant \eta_{\mathrm{opt}}$, and $\omega'_i \leftarrow C_{\mathrm{opt}}\omega_i$ if $\omega_i < \eta_{\mathrm{opt}}$;
return ω'.

The correctness of Algorithm 3 is proved in Sec. A.4 in the full-version of this paper (Allen-Zhu et al., 2017). Due to space constraints we omit its proof in this paper.

Appendix B Gradient calculations

We give detailed gradient/sub-gradient calculations for the popular optimality criteria listed in Sec. 2.1. Recall that $\Sigma = \sum_{i=1}^n \pi_i x_i x_i^\top$. We also use $\mathcal{Q} = \{\pi : 0 \leqslant \pi_i \leqslant 1, \sum_{i=1}^n \pi_i \leqslant k, \sum_{i=1}^n \pi_i x_i x_i^\top \in \mathbb{S}_p^+\}$ for the feasible set of π, for which Σ is in the domain of f.

- **A-optimality** $f_A(\Sigma) = \frac{1}{p}\mathrm{tr}(\Sigma^{-1})$ is differentiable on $\mathcal{Q}$, and $\frac{\partial f_A}{\partial \pi_i} = -\frac{1}{p} x_i^\top \Sigma^{-2} x_i$.
- **D-optimality** $g_D(\Sigma) = -\frac{1}{p}\log\det(\Sigma)$ is differentiable on $\mathcal{Q}$, and $\frac{\partial g_D}{\partial \pi_i} = -\frac{1}{p} x_i^\top \Sigma^{-1} x_i$.
- **T-optimality** $f_T(\Sigma) = p/\mathrm{tr}(\Sigma)$ is differentiable on $\mathcal{Q}$, and $\frac{\partial f_T}{\partial \pi_i} = -\frac{1}{p}\frac{x_i^\top x_i}{\mathrm{tr}(\Sigma)}$.
- **E-optimality** $f_E(\Sigma) = \|\Sigma^{-1}\|_2$ is *not* differentiable everywhere on $\mathcal{Q}$. Let $\sigma_{\min}$ be the least eigenvalue of Σ and $u \in \mathbb{R}^p, \|u\|_2 = 1$ be any unit vector such that $\Sigma u = \sigma_{\min} u$. Define $d_i := -|u^\top x_i|^2/\sigma_{\min}^2$ and $d = [d_1, \cdots, d_n]$. Then $d \in \partial f_E(\pi)$, where $\partial f_E(\pi)$ is the set of subgradients of f_E at π.
- **V-optimality** $f_V(\Sigma) = \frac{1}{n}\mathrm{tr}(X\Sigma^{-1}X^\top)$ is differentiable on $\mathcal{Q}$, and $\frac{\partial f_V}{\partial \pi_i} = -\frac{1}{n} x_i^\top \Sigma^{-1} X^\top X \Sigma^{-1} x_i$.
- **G-optimality** $f_G(\Sigma) = \max \mathrm{diag}(X\Sigma^{-1}X^\top)$ is *not* differentiable everywhere on $\mathcal{Q}$. Let $j = \mathrm{argmax}_{1 \leqslant i \leqslant n} e_i^\top X \Sigma^{-1} X^\top e_i$ and define $d_i := -(x_i^\top \Sigma^{-1} e_j)^2$. Then $d = [d_1, \cdots, d_n] \in \partial f_G(\pi)$, where $\partial f_G(\pi)$ is the set of subgradients of f_G at π.

References

Ageev, A. A., & Sviridenko, M. I. (2004). Pipage rounding: A new method of constructing algorithms with proven performance guarantee. *Journal of Combinatorial Optimization*, *8*(3), 307–328.

Allen-Zhu, Z., Li, Y., Singh, A., & Wang, Y. (2017). Near-optimal discrete optimization for experimental design: A regret minimization approach. *arXiv preprint arXiv:1711.05174*.

Allen-Zhu, Z., Liao, Z., & Orecchia, L. (2015). Spectral sparsification and regret minimization beyond matrix multiplicative updates. In *STOC*.

Avron, H., & Boutsidis, C. (2013). Faster subset selection for matrices and applications. *SIAM Journal on Matrix Analysis and Applications*, *34*(4), 1464–1499.

Beck, A., & Teboulle, M. (2003). Mirror descent and nonlinear projected subgradient methods for convex optimization. *Operations Research Letters*, *31*(3), 167–175.

Bian, A. A., Buhmann, J. M., Krause, A., & Tschiatschek, S. (2017). Guarantees for greedy maximization of non-submodular functions with applications. In *ICML*.

Bouhtou, M., Gaubert, S., & Sagnol, G. (2010). Submodularity and randomized rounding techniques for optimal experimental design. *Electronic Notes in Discrete Mathematics*, *36*, 679–686.

Chaloner, K., & Verdinelli, I. (1995). Bayesian experimental design: A review. *Statistical Science*, *10*(3), 273–304.

Chamon, L. F., & Ribeiro, A. (2017). Greedy sampling of graph signals. *arXiv preprint arXiv:1704.01223*.

Chaudhuri, K., Kakade, S., Netrapalli, P., & Sanghavi, S. (2015). Convergence rates of active learning for maximum likelihood estimation. In *NIPS*.

Chernoff, H. (1953). Locally optimal designs for estimating parameters. *The Annals of Mathematical Statistics*, *24*(4), 586–602.

Condat, L. (2015). Fast projection onto the simplex and the L1-ball. *Mathematical Programming (Series A)*, *158*, 575–585.

Dhillon, P., Lu, Y., Foster, D. P., & Ungar, L. (2013). New subsampling algorithms for fast least squares regression. In *NIPS*.

Duchi, J., Shalev-Shwartz, S., Singer, Y., & Chandra, T. (2008). Efficient projections onto the L1-ball for learning in high dimensions. In *ICML*.

Fedorov, V. V. (1972). *Theory of optimal experiments*. Elsevier.

Khuri, A., Mukherjee, B., Sinha, B., & Ghosh, M. (2006). Design issues for generalized linear models: a review. *Statistical Science*, *21*(3), 376–399.

Leeds, D. D., Pyles, J. A., & Tarr, M. J. (2014). Exploration of complex visual feature spaces for object perception. *Frontiers in computational neuroscience*, *8*, 106.

Leeds, D. D., & Tarr, M. J. (2016). A method for real-time visual stimulus selection in the study of cortical object perception. *Neuroimage*, *133*, 529–548.

Li, C., Jegelka, S., & Sra, S. (2017). Polynomial time algorithms for dual volume sampling. In *Proceedings of Advances in Neural Information Processing Systems (NIPS)*.

McMahan, H. B. (2011). Follow-the-regularized-leader and mirror descent: Equivalence theorems and L1 regularization. In *AISTATS*.

Miller, A., & Nguyen, N.-K. (1994). A Fedorov exchange algorithm for d-optimal design. *Journal of the Royal Statistical Society, Series C (Applied Statistics)*, *43*(4), 669–677.

Nakamura, N., Seepaul, J., Kadane, J. B., & Reeja-Jayan, B. (2017). Design for low-temperature microwave-assisted crystallization of ceramic thin films. *Applied Stochastic Models in Business and Industry*, *33*(3), 314–321.

Nikolov, A. (2015). Randomized rounding for the largest simplex problem. In *STOC*.

Nikolov, A., & Singh, M. (2016). Maximizing determinants under partition constraints. In *Proceedings of the annual ACM symposium on Theory of computing (STOC)*.

Nikolov, A., Singh, M., & Tantipongpipat, U. T. (2018). Proportional volume sampling and approximation algorithms for A-optimal design. *arXiv preprint arXiv:1802.08318*.

Pukelsheim, F. (2006). *Optimal design of experiments*. SIAM.

Reeja-Jayan, B., Harrison, K. L., Yang, K., Wang, C.-L., Yilmaz, A., & Manthiram, A. (2012). Microwave-assisted low-temperature growth of thin films in solution. *Scientific reports*, *2*, 1003.

Singh, M., & Xie, W. (2017). Approximate positive correlated distributions and approximation algorithms for d-optimal design. In *SODA*.

Spielman, D. A., & Srivastava, N. (2011). Graph sparsification by effective resistances. *SIAM Journal on Computing*, *40*(6), 1913–1926.

Summa, M. D., Eisenbrand, F., Faenza, Y., & Moldenhauer, C. (2015). On largest volume simplices and sub-determinants. In *SODA*.

Ulu, E., Mccann, J., & Kara, L. B. (2017). Lightweight structure design under force location uncertainty. *ACM Transactions on Graphics*, *36*(4), 158.

Van der Vaart, A. W. (2000). *Asymptotic statistics*, vol. 3. Cambridge university press.

Wang, Y., Ulu, E., Singh, A., & Kara, L. B. (2018). Efficient load sampling for worst-case structural analysis under force location uncertainty. In *ASME IDETC-DAC*.

Wang, Y., Yu, W. A., & Singh, A. (2017). On computationally tractable selection of experiments in measurement-constrained regression models. *Journal of Machine Learning Research*, *18*(143), 1–41.