Experimental Design Approaches to Maximum Stress Prediction for Lightweight Structure Designs

Yining Wang

Tuesday 19th December, 2017

Abstract

Background. One important question in optimizing the structure of a manufactured object and aiming for a lightweight design is to predict, given a particular shape hypothesis, the largest stress an object experiences under certain external force configurations. An optimization procedure can then be carried out to synthesize the lightest weight structure that can withstand maximum stress for all possible external force configurations. The problem proves to be computationally very challenging, as the critical force configuration creating the largest stress on the object changes as the structure evolves along the optimization path and thus, approximate computations of maximum stress under external forces are necessary.

Objectives. Use as few sample force positions as possible to predict the maximum stress under external forces applied to arbitrary parts on the structure surface.

Data. We use the data of three structures (Fig. 1) conveniently denoted as Fertility, RockingChair and Shark. Discretization of structures into finite elements yields thousands of surface elements on which external forces can be applied, and the maximum stress across the entire volumetric mesh is sought. The data come from (Ulu et al., 2017).

Methods. We use computationally tractable methods for optimal design in linear models, introduced in (Wang et al., 2016; Allen-Zhu et al., 2017). Both methods select a pre-determined number of sample points on the structure surface using only the graph Laplacians of the boundary element mesh, without inspecting the stress response that requires computationally expensive finite-element analysis (FEA).

Results. On the three test structures, our proposed algorithm pipeline 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× speed-up compared to the brute-force approach that performs FEA on the entire surface mesh, which consists of 4,000 to 6,000 potential force nodes. It also achieves a 5× speed-up over existing work (Ulu et al., 2017) and is also much simpler to implement.

Keywords: Lightweight structure design, experimental design, linear regression

DAP Committee members:

Aarti Singh ⟨*aarti@cs.cmu.edu*⟩ (Machine Learning Department);

Levent Burak Kara ⟨*lkara@cmu.edu*⟩ (Department of Mechanical Engineering);
Approved by DAP Committee Chair: _____

1 Introduction

As additive manufacturing technologies (also commonly known as 3D printing) become increasingly popular in recent years, the question of optimization in lightweight structure designs has received much research attention. Common approach in formalizing such an optimization problem is to model the external forces as known and fixed quantities. However, in many real world applications, the external forces, contact locations and magnitudes may change significantly. In order to guarantee that the resulting structure is structurally sound under all possible force configurations, the maximum stress experienced within the current shape hypothesis under the worst-case force configuration is to be predicted at each optimization step. The predicted maximum stress is then compared against a pre-specified tolerance threshold (i.e., material yield strength) and the structure design is then updated accordingly.

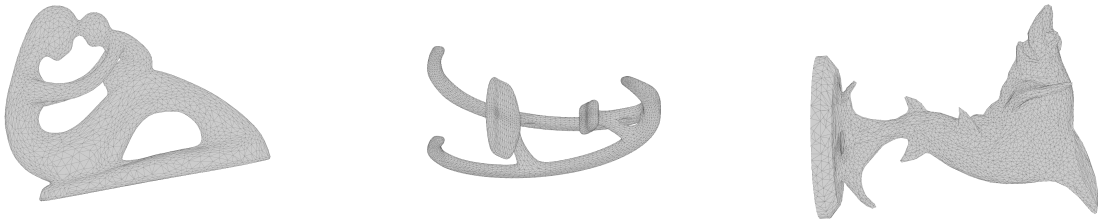
Finite-element analysis (FEA) is the standard technique that computes the stress distribution for a given external force node and the maximum stress suffered can then be subsequently obtained. However, FEA is generally computationally expensive, and performing the FEA on every external force configuration possible is out of the question for most structures. Approximation methods that reduce the total number of FEA runs are mandatory to make the stress prediction and structure optimization problem feasible in practice.

Ulu et al. (2017) initiated the research of applying machine learning techniques to the maximum stress prediction problem. In particular, a small subset of force locations were sub-sampled and the stress response on these locations are computed by FEA. Afterwards, a Principle Component Analysis (PCA) quadratic regression model was built on the sub-sampled data points, which was then used to predict the stress distribution on the other force configurations not sampled. Empirical results show that, with additional post-processing steps, the machine learning based approach gives accurate predictions of the maximum stress while using much fewer number of FEAs.

In this paper, we improve over the methods in (Ulu et al., 2017) to further reduce the number of FEAs needed without much sacrifice in the prediction accuracy of the maximum structure stress. We first propose a simplified algorithm pipeline, which replaces the PCA quadratic regression model in (Ulu et al., 2017) by a simple linear regression model built on the top eigenvectors of the graph Laplacian of the full boundary mesh, and substitutes the local search procedure with a simple ranking post-processing step. Furthermore, we apply the computationally tractable experimental design methods developed in (Wang et al., 2016; Allen-Zhu et al., 2017) to select the sub-sampled training set, which improves the naive training set selection algorithm in (Ulu et al., 2017) that only maximizes the pairwise geodesic distance between the selected force nodes. Our experimental results suggest that the newly proposed algorithm significantly improves over existing work, with approximately 5× fewer FEAs required compared to (Ulu et al., 2017) and up to 100× compared against the brute-force approach.

2 Problem formulation and methods

In this section we formally formulate the stress prediction problem in lightweight structure design and present a simple pipeline algorithm that produces such a prediction using a small collection



(a) Fertility (b) RockingChair (c) Shark

Figure 1: The three test structures.

of subsampled surface forces. We then introduce the methods developed in (Wang et al., 2016; Allen-Zhu et al., 2017) that subsample the surface elements in a principled way.

2.1 Problem formulation

Discretized structure has n_S nodes on the boundary $\mathcal{S}$ while the entire volumetric mesh $\mathcal{W}$ contains n_W nodes. Out of the n_S boundary nodes, there are contact regions $\mathcal{F} \subseteq \mathcal{S}$ with $n_F < n_S$ nodes upon which external forces might be applied. For example, in the structures depicted in Fig. 1, the objects are fixed from the bottom of the supporting platform (e.g., in the first figure) and corresponding regions are excluded from $\mathcal{F}$ as no external/user forces are expected there.

Given $\mathcal{S}, \mathcal{F}, \mathcal{W}$ and the current shape hypothesis, it is possible to calculate a *stress distribution* over $\mathcal{W}$ for a particular force in $\mathcal{W}$ using finite-element analysis (FEA), mathematically formulated as

$$\sigma : \mathcal{F} \times \mathcal{W} \to \mathbb{R}_+.$$

More specifically, running one FEA on a force node $f \in \mathcal{F}$ one obtains $\sigma(f, \cdot)$, which includes the data of $\sigma(f, w)$ for all $w \in \mathcal{W}$.

Given the stress distribution $\sigma(f, \cdot)$ for every force node $f \in \mathcal{F}$, the objective is then to calculate the *maximum* stress on the weak region $\mathcal{W}$ incurred by the worst-case external force, mathematically defined as

$$\sigma^* := \max_{f \in \mathcal{F}} \sigma^*(f) := \max_{f \in \mathcal{F}} \max_{w \in \mathcal{W}} \sigma(f, w).$$

2.2 The algorithm pipeline

A naive method to obtain the maximum stress σ^* is to compute the stress distribution $\sigma(f, \cdot)$ for every force node f in the force region $\mathcal{F}$. However, as FEAs are computationally heavy [1] and the number of force configurations n_F is large (in the range of 5 to 6 thousands), such a brute-force

[1] In our experiments, the fastest first-order FEA on *each element* f typically takes several minutes to complete, and much longer if higher-order approximations are used.

method is infeasible in real-world applications and approximate computations of the maximum stress σ^* is mandatory.

In this paper, we introduce the following algorithm pipeline that efficiently computes σ^* using linear Laplacian smoothing on a small subset of subsampled force nodes in $\mathcal{F}$. Our algorithmic pipeline is a great simplification of that developed in (Ulu et al., 2017) but yields much more competitive results.

Subsampling of force nodes. The algorithm pipeline starts with sampling a small subset of the contact region $\mathcal{F}_L \subseteq \mathcal{F}$ such that $\mathcal{F}_L$ contains $n_{FL} \ll n_F$ force nodes. The subsampling can be accomplished by simple uniform or leverage score sampling (Drineas et al., 2008; Spielman & Srivastava, 2011), or more sophisticated methods such as the k-means algorithm (Ulu et al., 2017) and the computationally efficient algorithms proposed in (Wang et al., 2016; Allen-Zhu et al., 2017). After $\mathcal{F}_L$ is obtained, FEA is performed on the n_{FL} subsampled force nodes to obtain the critical stress map $\sigma(f, \cdot)$ as well as the maximum stress $\sigma^*(f)$ for all $f \in \mathcal{F}_L$.

Linear Laplacian Smoothing Let $\mathbf{F} \in \mathbb{R}^{n_F \times n_F}$ be the *adjacency matrix* on $\mathcal{F}$, where $\mathbf{F}(f, f')$ is the percentage of forces imposed on f' for a small circular force region centered at f if f, f' are neighboring elements or close to each other in $\mathcal{F}$, and $\mathbf{F}(f, f') = 0$ otherwise. Sec. 3 in (Ulu et al., 2017) gives a complete description of the construction of $\mathbf{F}$. Let $\mathbf{L} \in \mathbb{R}^{n_F \times p}$ be the top-p eigenvectors of the *graph Laplacian* of $\mathcal{F}$ or $\mathbf{F}$ which captures the spectral properties of the mesh graph constructed on $\mathcal{F}$, where p is a small integer that is set as $p = 15$ in this paper. We refer the readers to the book of Chung (1997) for an introduction to the graph Laplacian and its properties. Denote also $\boldsymbol{\sigma}^* = [\sigma^*(f)]_{f \in \mathcal{F}} \in \mathbb{R}_+^{n_F}$ as the maximum stress responses for all force configurations in $\mathcal{F}$. We build the following *linear regression* model:

$$\boldsymbol{\sigma}^* = \overline{\mathbf{F}}\mathbf{L}\beta_0 + \boldsymbol{\varepsilon}, \tag{2.1}$$

where $\overline{\mathbf{F}}$ is the mean-centered adjacency matrix of $\mathbf{F}$ so that each column of $\overline{\mathbf{F}}$ sums to 0, β_0 is a p-dimensional unknown vector that models the linear map, and $\boldsymbol{\varepsilon} \in \mathbb{R}^{n_F}$ represents modeling noise for each force node.

Let $\boldsymbol{\sigma}_L^* = [\sigma^*(f)]_{f \in \mathcal{F}_L}$ be the maximum stress responses on the subsampled force nodes in $\mathcal{F}_L$, which can be calculated from the results of FEAs carried out on all $f \in \mathcal{F}_L$. Let also $[\mathbf{FL}]_L$ be the corresponding n_{FL} rows in the $n_F \times p$ matrix $\mathbf{FL}$. An ordinary least squares (OLS) estimator $\widehat{\beta}$ can be calculated using $\boldsymbol{\sigma}_L^*$ and $[\overline{\mathbf{F}}\mathbf{L}]_L$ as follows:

$$\widehat{\beta} \in \arg\min_{\beta \in \mathbb{R}^p} \left\| \boldsymbol{\sigma}_L^* - [\overline{\mathbf{F}}\mathbf{L}]_L \beta \right\|_2^2. \tag{2.2}$$

Afterwards, a preliminary prediction of maximum stress $\widehat{\boldsymbol{\sigma}}^*$ can be obtained by applying the OLS estimate $\widehat{\beta}$:

$$\widehat{\boldsymbol{\sigma}}^* = \overline{\mathbf{F}}\mathbf{L}\widehat{\beta}. \tag{2.3}$$

Force node ranking and final predictions Given the maximum stress prediction $\widehat{\boldsymbol{\sigma}}^*$ in Eq. (2.3), it is tempting to directly report $\max_{f\in\mathcal{F}} \widehat{\sigma}^*(f)$ as the final prediction of the maximum stress σ^* corresponding to the worst-case force. However, our empirical results suggest that the *absolute estimation error* $\|\widehat{\boldsymbol{\sigma}}^* - \boldsymbol{\sigma}^*\|_\infty$ is in fact quite large, which makes the direct predicting approach less desriable. This phenomenon was also observed in (Ulu et al., 2017), who showed that a quadratic regression model on dimensionality-reduced data has a similar large error in terms of predicting the absolute value of the maximum stress response.

On the other hand, while the absolute estimation error $\|\widehat{\boldsymbol{\sigma}}^* - \boldsymbol{\sigma}^*\|_\infty$ can be large, we observe that the *relative positions* of the force nodes which result in maximum stress as predict ed by $\widehat{\boldsymbol{\sigma}}^*$ are quite accurate. In other words, if a force at node $f \in \mathcal{F}$ is predicted to result in large maximum stress as indicated by $\widehat{\boldsymbol{\sigma}}^*$, then it is likely to cause large stress in the ground-truth response $\boldsymbol{\sigma}^*$ as well. This observation motivates us to consider a "ranking" approach for making the final stress predictions.

In particular, for a parameter $k \ll n_F$, we define $\widehat{\mathcal{F}}_K$ as the top k force nodes in $\mathcal{F}$ that have the largest stress response according to $\widehat{\boldsymbol{\sigma}}^*$. We then perform FEA on every force node in $\widehat{\mathcal{F}}_K$, with the maximum stress response denoted as

$$\widehat{\boldsymbol{\sigma}}^*_K = \left[\max_{w\in\mathcal{W}} \sigma(f, w)\right]_{f\in\widehat{\mathcal{F}}_K}.$$

The predicted maximum stress is then calculated as $\widetilde{\sigma}^* := \max_{f\in\widehat{\mathcal{F}}_K} \widehat{\boldsymbol{\sigma}}^*_K(f)$. Compared to the fully $\widehat{\sigma}^*$ based approach, the new predictor $\widehat{\sigma}^*_K$ is much more accurate as fresh FEAs are run on the predicted "top" force nodes; therefore, as long as the relative force positions resulting in maximum stress as indicated by $\widehat{\boldsymbol{\sigma}}^*$ are sufficiently accurate the predicted maximum stress $\widetilde{\sigma}^*_K$ is very accurate. On the other hand, when K is small then very few number of new FEAs are needed for the final prediction, which is only a small computational overhead incurred on the entire algorithm pipeline.

2.3 Force node subsampling by computationally tractable experimental design

In this section we give further details on the algorithms proposed in (Wang et al., 2016; Allen-Zhu et al., 2017) that select the contact region subset $\mathcal{F}_L \subseteq \mathcal{F}$, corresponding to the first step in our algorithm pipeline. Both algorithms are based on the (optimal) experimental design question in the statistics literature, and draw tools from the areas of convex optimization and spectral graph sparsification to achieve computational tractability.

An overview of optimal experimental design. The question of selecting a small number of design points from a large design pool so as to maximize the statistical efficiency on the optimized design is a old topic in statistics, usually termed as *experimental design* or *optimal design* (Pukelsheim, 2006; Chaloner & Verdinelli, 1995). More specifically, for the linear regression model $\boldsymbol{\sigma}^* = \mathbf{X}\beta_0 + \boldsymbol{\varepsilon}$ specified in Eq. (2.1), where $\mathbf{X} = \overline{\mathbf{F}}\mathbf{L} = (x_1, \cdots, x_{n_F}) \subseteq \mathbb{R}^p$, the optimal design problem can be formulated as a combinatorial optimization problem as follows:

$$\min_{s_1,\cdots,s_n} \Phi\left(\sum_{i=1}^n s_i x_i x_i^\top\right) \quad s.t.\ \ s_i \in \{0,1\},\ \ \sum_{i=1}^n s_i \le n_{FL}. \tag{2.4}$$

Here in Eq. (2.4), the binary variables $\{s_i\}_{i=1}^n$ represent the selected design points (force regions) in $\mathcal{F}_L$, and the objective function Φ is the *optimality criterion* that reflects certain aspects of the desired statistical efficiency. As in our problem the *prediction error* $\mathbf{X}(\widehat{\beta}-\beta_0)$ is of primary concern in our problem, the most relevant criteria are the V- and G-optimality:

- *V-optimality*: $\Phi_V(A) = \frac{1}{n}\mathrm{tr}(XA^{-1}X^\top)$;
- *G-optimality*: $\Phi_G(A) = \max_{1\le i\le n} x_i^\top A^{-1} x_i$.

The V-optimality Φ_V measures the variance of the prediction averaged over all design points x_i, and the G-optimality Φ_G measures the maximum prediction variance. In this paper we opt for the V-optimality Φ_V for computational reasons, because Φ_V is differentiable and hence easier to optimize by first-order optimization methods.

A continuous relaxation. As direct optimization of the combinatorial problem in Eq. (2.4) is difficult, we instead consider a *continuous relaxation* of Eq. (2.4), which was also described and analyzed in (Wang et al., 2016; Allen-Zhu et al., 2017; Joshi & Boyd, 2009):

$$\min_{\pi_1,\cdots,\pi_n} \Phi\left(\sum_{i=1}^n \pi_i x_i x_i^\top\right) \quad s.t.\ 0\le\pi_i\le 1,\ \sum_{i=1}^n \pi_i \le n_{FL}. \tag{2.5}$$

The only difference between Eqs. (2.5) and (2.4) is that the optimization variables in Eq. (2.5) are no longer constrained to take integer values. In addition, because the Φ_V objective is *convex*, the optimization problem in Eq. (2.5) becomes a standard convex continuous optimization problem and can be solved using conventional convex optimization methods. In particular, we use the projected gradient descent ((Bubeck et al., 2015, Sec. 3.1), see also Nesterov (2013)) to optimize Eq. (2.5), which is observed to converge fast in practice (Wang et al., 2016). We attach the details of the projected gradient descent algorithm in the appendix.

Sparsifying the continuous solution. The optimal solution π^* to the continuously relaxed problem in Eq. (2.5) is in general a dense vector, and cannot be used directly to obtain a subset $\mathcal{F}_L \subseteq \mathcal{F}$ with only n_{FL} force nodes. A simple fix is to first "normalize" the solution vector π^* such that $\sum_{i=1}^n \pi_i^* = \sum_{f\in\mathcal{F}} \pi(f)^* = 1$, and then *sample* n_{FL} force nodes without replacement from $\mathcal{F}$, where each force node $f \in \mathcal{F}$ is sampled with probability $\pi(f)^*$. This was the approach adopted in (Wang et al., 2016), and the resulting design subset $\mathcal{F}_L$ enjoys theoretical guarantees too if n_{FL} is not too small compared to p.

Allen-Zhu et al. (2017) proposed a more sophisticated algorithm that turns the optimal continuous solution π^* into a "sparse" design set. The algorithm starts with the empty set $\emptyset$ and greedily add elements $f \in \mathcal{F}$ into the design set $\mathcal{F}_L$, according to a carefully designed potential function. The greedy algorithm has the advantage of being completely deterministic, thus reducing the uncertainty in sampling methods caused by statistical fluctuation. Allen-Zhu et al. (2017) also shows that the greedy algorithm enjoys better theoretical approximation guarantees compared to the sampling method. We attach the details of the greedy algorithm in the appendix.

Table 1: Statistics of the testing structures

	n_W	n_S	n_F	σ^* [MPa]
Fertility	11221	4494	3914	6.37
RockingChair	15191	5918	5348	37.0
Shark	14152	5757	4281	26.2

3 Experiments

3.1 Data descriptions and methods

We use three test structures (Fertility, RockingChair and Shark) to evaluate the performance of our proposed algorithm pipeline. Table 1 gives a description of statistics of the three structures, including n_S, n_W, n_F and the maximum stress σ^*.

Within our algorithm pipeline, we consider 5 different methods that subsample the force nodes $\mathcal{F}_L$, which is arguably the most important step in our algorithmic framework. The first two methods, Uniform and Levscore, are the relatively simple uniform and leverage score sampling methods that serve as baselines. The third method K-means produces force samples that are as uniform as possible on the structure surface, and was the subsampling method used in previous work (Ulu et al., 2017). The last two methods correspond to the Sampling and Greedy algorithms described in Sec. 2.3 in this paper.

1. Uniform: the sample set $\mathcal{F}_L$ is obtained by sampling without replacement each force node $f \in \mathcal{F}$ uniformly at random, until n_{FL} samples are obtained;
2. Levscore: the sample set $\mathcal{F}_L$ is obtained by sampling without replacement each force node $f \in \mathcal{F}$ with probability proportional to its *leverage score*, defined as $x(f)^\top(\mathbf{X}^\top\mathbf{X})^{-1}x(f)$, until n_{FL} samples are obtained;
3. K-means: the sample set $\mathcal{F}_L$ consists of n_{FL} force nodes in $\mathcal{F}$ such that the geodesic distance between the closest force nodes in $\mathcal{F}_L$ 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 sample set $\mathcal{F}_L$ is obtained by sampling without replacement each force node $f \in \mathcal{F}$ with probability $\pi(f)^*$ until n_{FL} samples are obtained, where π^* is the optimal solution to Eq. (2.5);
5. Greedy: the sample set $\mathcal{F}_L$ is obtained by the greedy selection algorithm described in Sec. 2.3 of this paper and (Allen-Zhu et al., 2017, Sec. 3), which is based on the optimal solution π^* to Eq. (2.5).

3.2 Evaluation measures

Suppose $\widetilde{\sigma}_K^*$ is the predicted maximum stress, which by definition is always less than or equal to the ground truth σ^*. The performance of an algorithm is evaluated by the smallest integer K required such that $\sigma^* \leq (1+\delta)\widetilde{\sigma}_K^*$, where $\delta \geq 0$ is a pre-specified error tolerance parameter. The setting

Table 2: Results for the structure **Fertility**. Randomized algorithm (UNIFORM, LEVSCORE and SAMPLING) are run for 10 independent trials and the median performance is reported.

	$n_{FL}=$	25	50	100	150	200	250	300	Total FEAs
$\delta=0$	UNIFORM	316.5	149	78.5	37.5	98.5	42.5	39	178.5 ($n_{FL}=100$)
	LEVSCORE	252.5	54.5	73.5	68.5	42.5	31	13.5	104.5 ($n_{FL}=50$)
	K-MEANS	237	**25**	61	82	57	**17**	**16**	75 ($n_{FL}=50$)
	SAMPLING	210.5	148.5	51	30	35.5	34	26.5	151 ($n_{FL}=100$)
	GREEDY	**12**	26	**13**	**7**	**11**	25	**33**	**37** ($n_{FL}=25$)
$\delta=0.05$	UNIFORM	285	80.5	52	10	63	10	10	130.5 ($n_{FL}=50$)
	LEVSCORE	175	26.5	55.5	59	17	10	7	76.5 ($n_{FL}=50$)
	K-MEANS	144	**2**	19	22	14	**2**	**2**	52 ($n_{FL}=50$)
	SAMPLING	202	113	10	**7**	11	8	6	110 ($n_{FL}=100$)
	GREEDY	**4**	3	**4**	**7**	**5**	**2**	6	**29** ($n_{FL}=25$)
$\delta=0.1$	UNIFORM	87.5	37.5	13	7	15.5	7	6.5	87.5 ($n_{FL}=50$)
	LEVSCORE	59	13	15	14	13	8	6	63 ($n_{FL}=50$)
	K-MEANS	46	**1**	7	20	8	**1**	**1**	51 ($n_{FL}=50$)
	SAMPLING	88	25	7.5	6	7.5	6.5	4	75 ($n_{FL}=50$)
	GREEDY	**4**	**1**	**1**	**4**	**1**	**1**	4	**29** ($n_{FL}=25$)

Table 3: Results for the structure **RockingChair**. Randomized algorithm (UNIFORM, LEVSCORE and SAMPLING) are run for 10 independent trials and the median performance is reported.

	$n_{FL}=$	25	50	100	150	200	250	300	Total FEAs
$\delta=0$	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 ($n_{FL}=100$)
	K-MEANS	4013	4400	4573	4301	4320	4620	4757	4038 ($n_{FL}=25$)
	SAMPLING	672.5	282	38.5	38	38	**36**	**36**	138.5 ($n_{FL}=100$)
	GREEDY	**36**	**35**	208	**35**	**36**	**36**	**36**	**61** ($n_{FL}=25$)
$\delta=0.05$	UNIFORM	404	466	201.5	20	88	93.5	18	170 ($n_{FL}=150$)
	LEVSCORE	444	192.5	20	**18.5**	**18**	**18**	**18**	120 ($n_{FL}=100$)
	K-MEANS	285	466	**14**	24	26	161	195	114 ($n_{FL}=100$)
	SAMPLING	540	268	21.5	20.5	20.5	20	20	121.5 ($n_{FL}=100$)
	GREEDY	**20**	**19**	200	20	20	20	20	**45** ($n_{FL}=25$)
$\delta=0.1$	UNIFORM	399	384.5	135.5	15.5	60.5	75.5	14	165.5 ($n_{FL}=150$)
	LEVSCORE	437	145	14	14	**14**	**14**	**14**	114 ($n_{FL}=100$)
	K-MEANS	258	395	**5**	**13**	16	140	184	105 ($n_{FL}=100$)
	SAMPLING	535	237	16	16	16	16	16	116 ($n_{FL}=100$)
	GREEDY	**14**	**14**	175	16	16	**14**	**14**	**39** ($n_{FL}=25$)

Table 4: Smallest K such that $(1+\delta)\widetilde{\sigma}_K^* \geq \sigma^*$ for the structure **Shark**. Randomized algorithm (UNIFORM, LEVSCORE and SAMPLING) are run for 10 independent trials and the median performance is reported.

	$n_{FL}=$	25	50	100	150	200	250	300	Total FEAs
$\delta=0$	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** ($n_{FL}=25$)
$\delta=0.01$	UNIFORM	568.5	341	131.5	156	15	**4**	4.5	215 ($n_{FL}=200$)
	LEVSCORE	416	**4**	**4**	**4**	**4**	**4**	**4**	54 ($n_{FL}=50$)
	K-MEANS	129	84	**4**	**4**	**4**	**4**	**4**	104 ($n_{FL}=100$)
	SAMPLING	872.5	69	**4**	**4**	**4**	**4**	**4**	104 ($n_{FL}=100$)
	GREEDY	**4**	115	**4**	**4**	**4**	**4**	**4**	**29** ($n_{FL}=25$)
$\delta=0.05$	UNIFORM	323	172.5	52	75	10	**4**	4.5	151 ($n_{FL}=100$)
	LEVSCORE	225	**4**	**4**	**4**	**4**	**4**	**4**	54 ($n_{FL}=50$)
	K-MEANS	129	80	**4**	**4**	**4**	**4**	**4**	104 ($n_{FL}=100$)
	SAMPLING	391.5	52.5	**4**	**4**	**4**	**4**	**4**	102.5 ($n_{FL}=50$)
	GREEDY	**4**	115	**4**	**4**	**4**	**4**	**4**	**29** ($n_{FL}=25$)

of $\delta = 0$ asks for exact identification of the maximum stress caused by the worst-case external force application, while $\delta > 0$ allows for a small error margin in the prediction of $\widetilde{\sigma}_K^*$, which can be subsequently compensated by using a more conservative mass distribution when producing the final structure designs.

3.3 Results

We report the smallest K recovered for $\widetilde{\sigma}_K^*$ to be lower bounded by $\sigma^*/(1+\delta)$ in Tables 2, 3 and 4 for the algorithms mentioned in Sec. 3.1 and all three testing structures. We report the performance for $\delta \in \{0, 0.01, 0.05, 0.1\}$ settings, and the sizes of the sub-sampled training set n_{FL} ranges from 25 to 300, with larger n_{FL} generally resulting in more accurate prediction (and hence smaller K) but computationally more expensive. The total number of FEAs needed by an algorithm is the sum of n_{FL} and K. For non-deterministic algorithms (UNIFORM, LEVSCORE and SAMPLING), we run the algorithm for 10 times and report the median performance (K) out of the 10 independent runs.

Tables 2, 3 and 4 suggest that both the K-MEANS and the GREEDY algorithms outperform their competitors for most structures and 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 n_{FL} is very small (e.g., $n_{FL} = 25$), while its competitors become much unstable for such a small n_{FL} setting and require large K values to

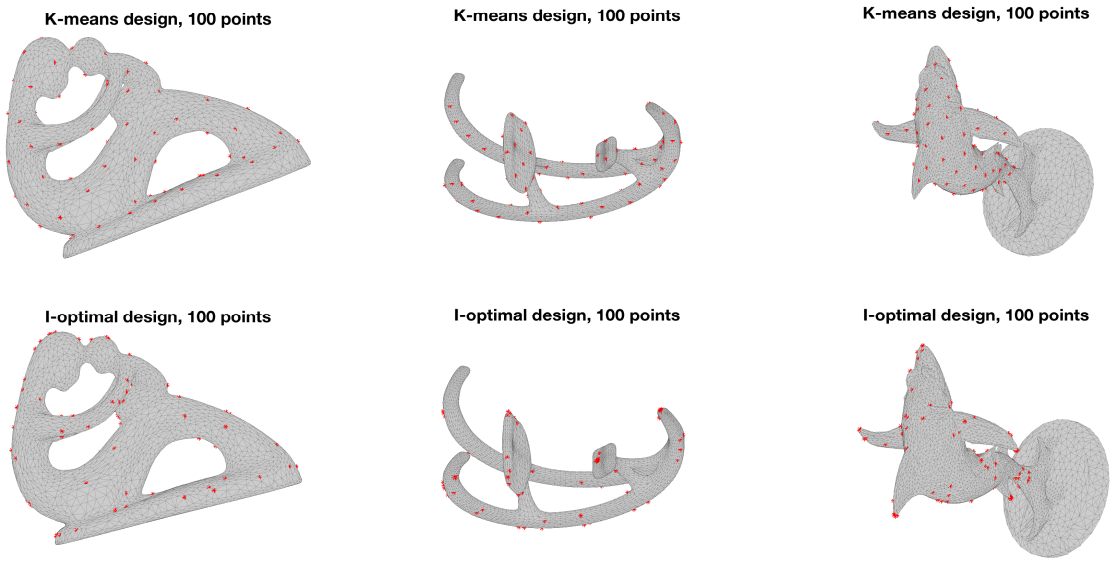


Figure 2: Sampled force nodes ($\mathcal{F}_L$) using the `K-means` algorithm (top row) versus the `Greedy` algorithm (bottom row) with $n_{FL} = 100$.

compensate for the prediction error in the first two stages. 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 all three tables.

4 Discussion

In Figs. 2 and 3 we plot the sub-sampled force nodes (i.e., $\mathcal{F}_L$) of the GREEDY algorithm with $n_{FL} = 100$ and $n_{FL} = 200$ points, respectively, and compare with the samples obtained by the K-MEANS algorithm considered in (Ulu et al., 2017). 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 seperately:

- 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 learnt linear model is more accurate in predicting the maximum stress, and therefore fewer FEAs are required to achieve a certain error tolerance level in $\widetilde{\sigma}^*_K$.
- ROCKINGCHAIR: for this structure, the GREEDY algorithm produces more samples on the

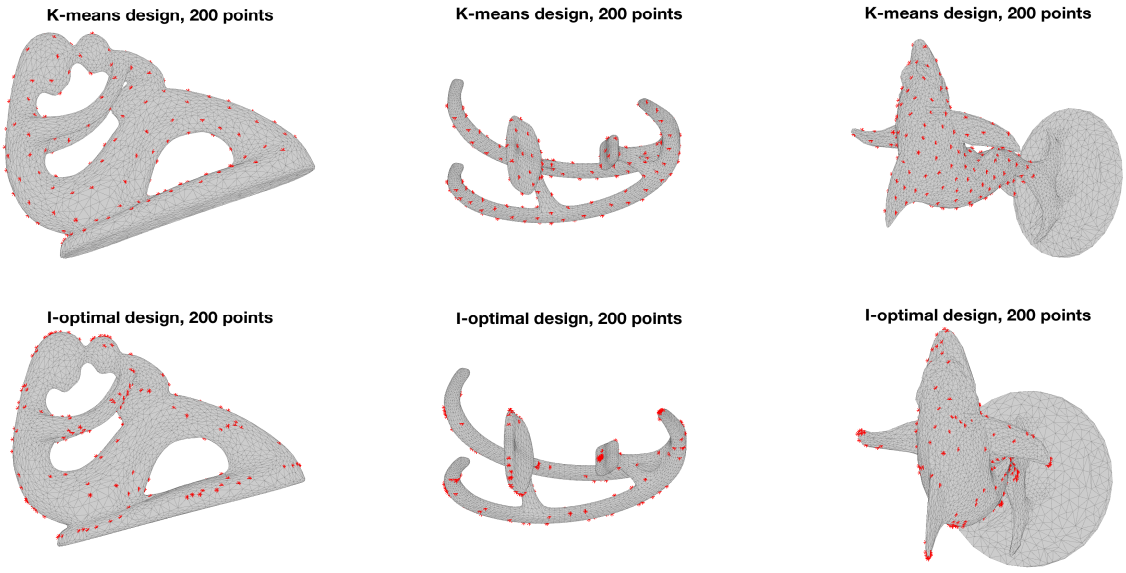


Figure 3: Sampled force nodes ($\mathcal{F}_L$) using the `K-means` algorithm (top row) versus the `Greedy` algorithm (bottom row) with $n_{FL} = 200$.

surface area of the smaller "seat" 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: this structure is rather easy to analyze, as shown by Table 4 that most sub-sampling methods can predict the maximum stress using very small number of FEAs. However, if we focus on the sample points on the wings of the shark there are some noticeable differences between K-MEANS and GREEDY: the GREEDY algorithm places more points around the tips of the wings, while samples produced by the K-MEANS algorithm are relatively uniformly distributed on the wing surfaces. This subtle difference makes GREEDY more robust in prediction accuracy when n_{FL} is very small.

Despite the significant reduction of FEA time, one important limitation of the proposed algorithm pipeline is the lack of stopping criteria. In particular, the performance parameter δ can only be evaluated *once the ground-truth maximum stress σ^* is known*, which is impractical for real-world applications. On the other hand, compared to conventional machine learning applications, performance control in the lightweight structure design problem is of vital importance because structure designs with insufficient stress tolerance may actually fail in reality, leading to devastating consequences. To overcome this issue, we suggest two possible fixes:

1. The first fix is simply to use a conservative estimate of the tolerance parameter δ, by setting $\delta = 0.2$ or even larger. Our empirical results suggest that for such a large tolerance parameter,

$K = 20$ is sufficient for almost all structures. The disadvantage of this strategy however is the loss of material efficiency, as a $\delta = 0.2$ error margin might be too conservative and require much more material mass on the structure. In addition, it is possible that for especially challenging/pathological structures, even a $\delta = 0.2$ margin might be insufficient.

2. A more principled method is to use *cross validation* to obtain an estimate on the accuracy of the predicted maximum stress $\widetilde{\sigma}_K^*$. More specifically, we start with a pre-fixed tolerance parameter δ and an initial small sample set $\mathcal{F}_L$. We then divide the $\mathcal{F}_L$ into disjoint training and testing sets, and obtain an estimated tolerance $\widehat{\delta}$. If $\widehat{\delta}$ is larger than δ desired, we conclude that the current samples are not sufficient to build a model that matches the desired accuracy, and n_{FL} is subsequently increased to obtain more data such that a more accurate model can be built. This process can be repeated until the cross-validated tolerance $\widehat{\delta}$ decreases to the specified tolerance level δ.

A The projected gradient descent algorithm

Using the V-optimality $\Phi_V(A) = \mathrm{tr}(XA^{-1}X^\top)$, the partial derivative of Φ_V with respect to π_i can be calculated as

$$\frac{\partial \Phi_V}{\partial \pi_i} = -\frac{1}{n} x_i^\top \mathbf{\Sigma}^{-1} \mathbf{X}^\top \mathbf{X} \mathbf{\Sigma}^{-1} x_i, \tag{A.1}$$

where $\mathbf{\Sigma} = \sum_{j=1}^n \pi_i x_i x_i^\top$. Let also $\Pi := \{\pi \in \mathbb{R}^n : 0 \le \pi_i \le 1, \sum_{i=1}^n \pi_i \le n_{FL}\}$ be the feasible set. The projected gradient descent algorithm can then be formulated as following:

1. **Input**: $\mathbf{X} \in \mathbb{R}^{n_F \times p}$, feasibility set Π, algorithm parameters $\alpha \in (0, 1/2]$, $\beta \in (0, 1)$;
2. **Initialization**: $\pi^{(0)} = (n_{FL}/n_F, \cdots, n_{FL}/n_F)$, $t = 0$;
3. **While** stopping criteria are not met **do the following**:
   (a) Compute the gradient $g_t = \nabla_\pi \Phi(\pi^{(t)})$ using Eq. (A.1);
   (b) Find the smallest integer $s \ge 0$ such that $\Phi(\pi) - \Phi(\pi^{(t)}) \le \alpha g_t^\top (\pi - \pi^{(t)})$, where $\pi = \mathcal{P}_\Pi(\pi^{(t)} - \beta^s g_t)$;
   (c) Update: $\pi^{(t+1)} = \mathcal{P}_\Pi(\pi^{(t)} - \beta^s g_t)$, $t \leftarrow t + 1$.

Here in Steps 3(b) and 3(c), the $\mathcal{P}_\Pi(\cdot)$ is the projection operator onto the (convex) constrain set Π in Euclidean norm. More specifically, $\mathcal{P}_\Pi(\pi) := \arg\min_{z \in \Pi} \|\pi - z\|_2$. Such projection can be efficiently computed in almost linear time up to high accuracy (Wang et al. (2016), see also Duchi et al. (2008); Condat (2015); Su et al. (2012)). Also, the step 3(b) corresponds to the Amijo's rule (also known as backtracking line search) that automatically selects step sizes, a popular and efficient method for step size selection in full gradient descent methods.

B The Greedy algorithm

The Greedy algorithm was proposed in (Allen-Zhu et al., 2017) as a principled method to sparsify the continuous optimization solution π^*. The algorithm makes uses of a carefully designed *potential*

function for $i \in [n_F]$ and $\Lambda \subseteq [n_F]$:

$$\psi(i;\Lambda) := \frac{x_i^\top \mathbf{B}(\Lambda) x_i}{1+\alpha x_i^\top \mathbf{B}(\Lambda)^{1/2} x_i} \quad \text{where } \mathbf{B}(\Lambda) = \left(c\mathbf{I} + \sum_{j\in\Lambda} x_j x_j^\top \right)^{-2}, \ \operatorname{tr}(\mathbf{B}(\Lambda)) = 1. \tag{B.1}$$

Here $\alpha > 0$ is an algorithm parameter that can be tuned, and $c \in \mathbb{R}$ is the unique real number such that $\operatorname{tr}(\mathbf{B}(\Lambda)) = 1$. The exact values of c can be computed efficiently using a binary search procedure, as shown in (Allen-Zhu et al., 2017). The potential function is inspired by a regret minimization interpretation of the least singular values of sum of rank-1 matrices. Interested readers should refer to (Allen-Zhu et al., 2017; Silva et al., 2016; Allen-Zhu et al., 2015) for more details and motivations.

Based on the potential function in Eq. (B.1), the GREEDY algorithm starts with an empty set and add force nodes one by one in a greedy manner, until there are n_{FL} elements in $\mathcal{F}_L$.

1. **Input**: $\mathbf{X} \in \mathbb{R}^{n_F \times p}$, budget n_{FL}, optimal solution π^*, algorithm parameter $\alpha > 0$;
2. **Whitening**: $\mathbf{X} \leftarrow \mathbf{X}(\mathbf{X}\mathbf{\Sigma}_* \mathbf{X}^\top)^{-1/2}$, where $\mathbf{\Sigma}_* = \sum_{i=1}^n \pi_i^* x_i x_i^\top$;
3. **Initialization**: $\Lambda_0 = \emptyset$, $\mathcal{F}_L = \emptyset$;
4. **For** $t = 1$ to n_{FL} **do the following**:
   (a) Compute $\psi(i;\Lambda_{t-1})$ for all $i \notin \Lambda_{t-1}$ and select $i_t := \arg\max_{i\notin\Lambda_{t-1}} \psi(i;\Lambda_{t-1})$;
   (b) Update: $\Lambda_t = \Lambda_{t-1} \cup \{i_t\}$, $\mathcal{F}_L \leftarrow \mathcal{F}_L \cup \{i_t\}$.

Acknowledgement

I would like to thank Erva Ulu for preparing the data and offering great help in data analysis. I also thank my DAP advisors Aarti Singh and Levent Burak Kara for their valuable suggestions. This research was supported in part by NSF CCF-1563918 and AFRL FA87501720212.

References

Allen-Zhu, Z., Li, Y., Singh, A., & Wang, Y. (2017). Near-optimal design of experiments via regret minimization. In *Proceedings of the International Conference on Machine Learning (ICML)*.

Allen-Zhu, Z., Liao, Z., & Orecchia, L. (2015). Spectral sparsification and regret minimization beyond matrix multiplicative updates. In *Proceedings of Annual Symposium on the Theory of Computing (STOC)*.

Bubeck, S., et al. (2015). Convex optimization: Algorithms and complexity. *Foundations and Trends® in Machine Learning*, *8*(3-4), 231–357.

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

Chung, F. R. (1997). *Spectral graph theory*. 92. American Mathematical Soc.

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

Drineas, P., Mahoney, M. W., & Muthukrishnan, S. (2008). Relative-error CUR matrix decompositions. *SIAM Journal on Matrix Analysis and Applications*, *30*(2), 844–881.

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

Joshi, S., & Boyd, S. (2009). Sensor selection via convex optimization. *IEEE Transactions on Signal Processing*, *57*(2), 451–462.

Nesterov, Y. (2013). *Introductory lectures on convex optimization: A basic course*, vol. 87. Springer Science & Business Media.

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

Silva, M. K., Harvey, N. J., & Sato, C. M. (2016). Sparse sums of positive semidefinite matrices. *ACM Transactions on Algorithms*, *12*(1), 9.

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

Su, H., Yu, A. W., & Li, F.-F. (2012). Efficient euclidean projections onto the intersection of norm balls. In *Proceedings of International Conference on Machine Learning (ICML)*.

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

Wang, Y., Yu, W. A., & Singh, A. (2016). On computationally tractable selection of experiments in regression models. *arXiv preprints: arXiv:1601.02068*.