Proceedings of Machine Learning Research vol 99:1–49, 2019 32nd Annual Conference on Learning Theory

Combinatorial Algorithms for Optimal Design

Vivek Madan VMADAN7@GATECH.EDU
Mohit Singh MOHIT.SINGH@ISYE.GATECH.EDU
Uthaipon (Tao) Tantipongpipat TAO@GATECH.EDU
Georgia Institute of Technology

Weijun Xie WXIE@VT.EDU
Virginia Polytechnic Institute and State University

Editors: Alina Beygelzimer and Daniel Hsu

Abstract

In an optimal design problem, we are given a set of linear experiments $v_1, \ldots, v_n \in \mathbb{R}^d$ and $k \geq d$, and our goal is to select a set or a multiset $S \subseteq [n]$ of size k such that $\Phi((\sum_{i\in S} v_i v_i^\top)^{-1})$ is minimized. When $\Phi(M) = Determinant(M)^{1/d}$, the problem is known as the D-optimal design problem, and when $\Phi(M) = Trace(M)$, it is known as the A-optimal design problem. One of the most common heuristics used in practice to solve these problems is the local search heuristic, also known as the Fedorov's exchange method (Fedorov, 1972). This is due to its simplicity and its empirical performance (Cook and Nachtrheim, 1980; Miller and Nguyen, 1994; Atkinson et al., 2007). However, despite its wide usage no theoretical bound has been proven for this algorithm. In this paper, we bridge this gap and prove approximation guarantees for the local search algorithms for D-optimal design and A-optimal design problems. We show that the local search algorithms are asymptotically optimal when $\frac{k}{d}$ is large. In addition to this, we also prove similar approximation guarantees for the greedy algorithms for D-optimal design and A-optimal design problems when $\frac{k}{d}$ is large.

Keywords: Optimal Design, Experimental Design, D-optimal design, A-optimal design, Fedorov Exchange, Local Search, Greedy Algorithm.

1. Introduction

Optimal experimental design (Pukelsheim, 2006) lies at the intersection of statistics and optimization where the goal is to pick a subset of statistical trials to perform from a given set of available trials. Linear models are one of the most widely used and well-studied models in the area (Federer et al., 1955; Pukelsheim, 2006; Atkinson et al., 2007). The goal is to learn an unknown parameter $\theta^\star \in \mathbb{R}^d$ from a set of linear experiments $\{v_1, \ldots, v_n\}$ where each $v_i \in \mathbb{R}^d$. If the i^{th} experiment is performed, we observe $y_i = \langle v_i, \theta^\star \rangle + \eta_i$ where η_i is a small error introduced in the experiment. Given an integer $k \leq n$, the optimization problem involves picking k vectors out of n to ensure the unknown parameter θ^* can be deduced as accurately as possible.

By assuming the error vector η_i is a gaussian noise, the maximum likelihood estimate for $\theta^\star$, call it $\hat{\theta}$, is obtained via minimizing the least square error over the set S of performed experiments, i.e. $\hat{\theta} = \text{argmin}_{\theta\in\mathbb{R}^d} \sum_{i\in S} \|v_i^\top \theta - y_i\|_2^2$. The error in estimation $\hat{\theta} - \theta^\star$ is distributed as Gaussian with mean zero. If the variance for each η_i is 1 (which can be assumed by normalization), then the covariance matrix $(\sum_{i\in S} v_i v_i^\top)^{-1}$. Optimal design consists of minimizing a function $\Phi\left((\sum_{i\in S} v_i v_i^\top)^{-1}\right)$ where $\Phi(M) = \det(M)^{\frac{1}{d}}$ for D-optimal design and $\Phi(M) = \text{tr}(M)$ for A-optimal design.

© 2019 V. Madan, M. Singh, U.(. Tantipongpipat & W. Xie.

D-DESIGN: Given a set of vectors $v_1, \ldots, v_n \in \mathbb{R}^d$ for some $d \in \mathbb{N}$, and a parameter $k \geq d$, our goal is to find a set or a multiset $S \subseteq [n]$ of size k such that $\det\left(\sum_{i \in S} v_i v_i^\top\right)^{1/d}$ is maximized[1]. Here, $\det(M)$ denote the determinant of the matrix M.

A-DESIGN: Given a set of vectors $v_1, \ldots, v_n \in \mathbb{R}^d$ for some $d \in \mathbb{N}$, and a parameter $k \geq d$, our goal is to find a set or a multiset $S \subseteq [n]$ of size k such that $\operatorname{tr}\left(\left(\sum_{i \in S} v_i v_i^\top\right)^{-1}\right)$ is minimized. Here, $\operatorname{tr}(M)$ denote the trace of the matrix M.

When selecting a multiset, we refer to the problem as optimal design with repetitions and when selecting a set, we refer to the problem as optimal design without repetitions. Statistically, D-DESIGN objective aims to minimize the volume of the confidence ellipsoid and the A-DESIGN objective aims to minimize the expected length square of the error vector $\hat{\theta} - \theta^*$. Several other objective functions such as E-design, G-design, and I-design have also been studied in literature (Atkinson et al., 2007).

One of the classical optimization methods that is used for optimal design problems is the local search heuristic which is also called the Fedorov's exchange method (Fedorov, 1972) (see also Mitchell and Miller Jr (1970)). The method starts with any set of k experiments from the given set of n experiments and aims to exchange one of the design vectors if it improves the objective. The ease in implementing the method as well as its efficacy in practice makes the method widely used (Nguyen and Miller, 1992) and implemented in statistics softwares such as SAS (see Atkinson et al. (2007), Chapter 13). Moreover, there has been considerable study on heuristically improving the performance of the algorithm. Surprisingly, theoretical analysis of this classical algorithm has not been performed despite its wide usage. In this paper, we bridge this gap and give theoretical guarantees on the performance of local search heuristic for D and A-optimal design problems. In addition to local search, we analyze the greedy heuristic for the D and A-optimal design problems.

1.1. Our Results and Contributions

Our main contribution is to prove worst case bounds on the performance of simple local search algorithm (also known as Fedorov Exchange method) and greedy algorithms. Our results also give worst case performance guarantee on the variants of local search algorithm.

Our first result is for the D-optimal design problem where we show the following guarantee. We consider both settings when the design vectors are allowed to be repeated in the solution and when they are not allowed to be repeated.

Theorem 1 *For any $\epsilon > 0$, the local search algorithm returns a $(1+\epsilon)$-approximate solution for D-DESIGN with or without repetitions whenever $k \geq d + \frac{d}{\epsilon}$.*

Our analysis method crucially uses the convex relaxation for the D-DESIGN problem. In recent works, the convex relaxation has been studied extensively and various rounding algorithms have been designed (Wang et al. (2016); Allen-Zhu et al. (2017); Singh and Xie (2018); Nikolov et al. (2019)). Solving the convex relaxation is usually the bottleneck in the running time of all these algorithms. Our results differ from this literature in that we only use the convex relaxation for the analysis of the local search heuristic. The algorithm does not need to solve the convex program (or even formulate it). We use the *dual-fitting* approach to prove the guarantee. We also remark

1. Since $\det(M^{-1}) = 1/\det(M)$, for notational convenience, we consider an equivalent formulation of D-DESIGN where instead of minimizing $\det((\sum_{i \in S} v_i v_i^\top)^{-1})^{1/d}$, we maximize $\det(\sum_{i \in S} v_i v_i^\top)^{1/d}$.

the above guarantee improves on the best previous bound which had an additional additive term of $\frac{1}{\epsilon^2}\log\frac{1}{\epsilon}$ in the requirement on the size of k.

We also consider the natural greedy algorithm for D-DESIGN problem. Indeed this algorithm has also been implemented and tested in empirical studies (see for example Atkinson et al. (2007), Chapter 12) and is referred to as the forward procedure algorithm. The algorithm is initialized to a small set of experiments and new experiments are added greedily. We show that the guarantee is slightly specific to the initialized set. If the initialized set is a local optimum set of size d, we obtain the following result. Again we employ the dual-fitting approach to prove the bounds.

Theorem 2 *For any $\epsilon > 0$, the greedy algorithm for D-DESIGN with repetitions returns a $(1+\epsilon)$-approximate solution whenever $k \geq \Omega\left(\frac{d}{\epsilon}\left(\log\frac{1}{\epsilon} + \log\log d\right)\right)$.*

A-DESIGN. While the simple combinatorial algorithms have tight asymptotic guarantee for D-DESIGN, we show that a similar guarantee *cannot* be proven for A-DESIGN. Indeed, there are examples where local optimum can be arbitrarily bad as compared to the optimum solution as we show in Section 3.3. We note that the bad local optima arise due to presence of long vectors among design vectors. In particular, we show that this is the *only* bottleneck to obtain an asymptotic guarantee on the performance of the local search algorithm. Moreover, we show a combinatorial iterative procedure to truncate the length of all the vectors while ensuring that the value of the optimal solution does not change significantly. This allows us to obtain a modified local search procedure with the following guarantee.

Theorem 3 *The modified local search algorithm for A-DESIGN with repetitions returns a $(1+\epsilon)$-approximate solution whenever $k = \Omega\left(\frac{d}{\epsilon^4}\right)$.*

We note that the above asymptotic guarantee does not match the best approximation algorithms (Nikolov et al., 2019) for A-DESIGN as was the case of D-DESIGN. Nonetheless, it specifically points why local search algorithm performs well in practice as has been noted widely (Atkinson et al., 2007).

We also consider the natural greedy algorithm for the A-DESIGN problem, which again requires truncating the length of all vectors. As in D-DESIGN problem, the guarantee depends on the initialized set. If the initialized set is a local optimum set of size cd for an absolute constant c, we obtain the following guarantee.

Theorem 4 *The modified greedy algorithm for A-DESIGN with repetitions returns a $(1+\epsilon)$-approximate solution whenever $k \geq \Omega\left(\frac{d}{\epsilon^3}\log^2\frac{1}{\epsilon}\right)$.*

Approximate Local Search: Theorem 1 and 3 show that the local search for D-DESIGN and modified local search for A-DESIGN yield $(1+\epsilon)$-approximation algorithm. But, as are typical of local search algorithms, they are usually not polynomial time algorithms. However, the standard fix is to make local improvements only when the objectives improves by a factor of $1+\delta$. With appropriately chosen δ, this implies a polynomial running time at the cost of a slight degradation in the approximation guarantee. We show that under the same assumption on parameter k, approximate local search for D-DESIGN and modified approximate local search for A-DESIGN yield $(1+2\epsilon)$-approximation when δ is small enough and take polynomially many iterations.

Theorem 5 *The $(1+\delta)$-approximate local search algorithm for D-DESIGN with repetitions returns a $(1+2\epsilon)$-approximate solution whenever $k \geq d + \frac{d}{\epsilon}$ and $\delta < \frac{\epsilon d}{2k}$, and the algorithm runs in polynomial time.*

Theorem 6 *The modified $(1+\delta)$-approximate local search algorithm for A-DESIGN with repetitions returns a $(1+2\epsilon)$-approximate solution whenever $k = \Omega\left(\frac{d}{\epsilon^4}\right)$ and $\delta < \frac{\epsilon d}{2k}$, and the algorithm runs in polynomial time.*

We note that approximate local optimum sets are sufficient for initialization of greedy algorithms, implying that greedy algorithms run in polynomial time.

1.2. Related Work

As we remarked earlier, experimental design is a classical problem and has attracted significant attention throughout the years. We refer the reader to Pukelsheim (2006) for a broad survey on the experimental design. Here, we mention the results known for the problems discussed in this paper.

D-DESIGN: When experiments can be picked fractionally, D-DESIGN reduces to the natural convex program which can be solved efficiently (Sagnol and Harman (2015)). In contrast, when experiments need to be chosen integrally as in this paper, D-DESIGN is NP-hard (Welch (1982)). Hence, there has been a series of approximation algorithms known for the problem. Bouhtou et al. (2010) gave a $\frac{n}{k}$-approximation algorithm based on rounding the solution of the natural convex program. Wang et al. (2016) improved the approximation ratio to $(1+\epsilon)$ when $k \geq \frac{d^2}{\epsilon}$. Allen-Zhu et al. (2017) gave a $(1+\epsilon)$-approximation algorithm when $k = \Omega\left(\frac{d}{\epsilon^2}\right)$. Singh and Xie (2018) improved this result and gave $(1+\epsilon)$-approximation algorithm when repetitions are not allowed and $k = \Omega\left(\frac{d}{\epsilon} + \frac{1}{\epsilon^2}\log\frac{1}{\epsilon}\right)$, and $(1+\epsilon)$-approximation when repetitions are allowed and $k \geq \frac{2d}{\epsilon}$. Our results improve on these bounds as they achieve $(1+\epsilon)$-approximation when $k \geq d + \frac{d}{\epsilon}$.

A-DESIGN: As in case of D-DESIGN, A-DESIGN reduces to solving the natural convex program which can be done efficiently when experiments are picked fractionally. On the other hand, when experiments are picked integrally as in this paper, A-DESIGN is NP-hard (Nikolov et al. (2019)). Several of the results mentioned above for D-DESIGN work in more generality and in particular for A-DESIGN as well. For instance, algorithm by Avron and Boutsidis (2013) gives $\frac{n-d+1}{k-d+1}$-approximation ratio for A-DESIGN as well. Algorithm by Wang et al. (2016) gives $(1+\epsilon)$-approximation ratio when $k \geq \frac{d^2}{\epsilon}$. Algorithm by Allen-Zhu et al. (2017) gives $(1+\epsilon)$-approximation ratio when $k = \Omega\left(\frac{d}{\epsilon^2}\right)$. Recently, Nikolov et al. (2019) showed d-approximation for A-DESIGN when $k = d$, $(1+\epsilon)$-approximation when repetitions are not allowed and $k = \Omega\left(\frac{d}{\epsilon} + \frac{1}{\epsilon^2}\log\frac{1}{\epsilon}\right)$, and $(1+\epsilon)$-approximation when repetitions are allowed and $k \geq \frac{(1+\epsilon)(d-1)}{\epsilon}$. On the hardness side, Nikolov et al. (2019) showed that A-DESIGN is APX-hard for $k = d$; there is no c-approximation for some constant $c > 1$.

Other variants of optimal design have been studied such as E-DESIGN problem where our goal is to select set $S \subseteq [n]$ of size k such that the minimum eigenvalue of $\sum_{i \in S} v_i v_i^\top$ is maximized. E-DESIGN is also known to be an NP-hard problem (Çivril and Magdon-Ismail (2009)). Algorithm by Avron and Boutsidis (2013) gives $d \cdot \frac{n-d+1}{k-d+1}$-approximation algorithm. Wang et al. (2016) gave $(1+\epsilon)$-approximation algorithm when $k \geq \frac{d^2}{\epsilon}$. Allen-Zhu et al. (2017) improved this result and gave $(1+\epsilon)$-approximation algorithm when $k = \Omega\left(\frac{d}{\epsilon^2}\right)$.

1.3. Organization

In Section 2, we analyze the local search algorithm for D-DESIGN and prove Theorem 1. In Section 3, we analyze the modified local search algorithm for A-DESIGN and prove Theorem 3. Sections A and B include details and proofs deferred from the main body of the paper. We present approximate local search algorithms for D-DESIGN and A-DESIGN and their analysis in Sections C and D, respectively, proving Theorems 5 and 6. Greedy algorithms and their analysis for D-DESIGN and A-DESIGN are presented in Sections E and F, respectively, which prove Theorems 2 and 4.

2. Local Search for D-DESIGN

We first give the local search algorithm for D-DESIGN with repetitions.

2.1. Local Search Algorithm

Algorithm 1 Local search algorithm for D-DESIGN

Input: $V = \{v_1, \ldots, v_n\}$ where $v_i \in \mathbb{R}^d, d \leq k \in \mathbb{N}$.
Let I be any (multi)-subset of $[1, n]$ of size k such that $X = \sum_{i \in I} v_i v_i^\top$ is non-singular matrix.
while $\exists i \in I, j \in [1, n]$ such that $\det\left(X - v_i v_i^\top + v_j v_j^\top\right) > \det(X)$ **do**
  $X = X - v_i v_i^\top + v_j v_j^\top$
  $I = I \setminus \{i\} \cup \{j\}$
end while
Return (I, X)

2.2. Relaxations

To prove the performance of local search algorithm, presented earlier as Theorem 1, we use the convex programming relaxation for the D-DESIGN problem. We first describe these relaxations in Figure 3 (see Chapter 7 of Boyd and Vandenberghe (2004)). Let ϕ_f^{D} denote the be the common optimum value of (D-REL) and its dual (D-REL-DUAL). Let $I^\star$ denote the indices of the vector in the optimal solution and let $\phi^{\mathrm{D}} = \det\left(\sum_{i \in I^\star} v_i v_i^\top\right)^{\frac{1}{d}}$ be its objective. Observe that $\phi_f^{\mathrm{D}} \geq \log \phi^{\mathrm{D}}$. Theorem 1 now follows from the following result.

Theorem 7 *Let X be the solution returned by Algorithm 1. Then,*

$$\det(X) \geq \left(\frac{k-d+1}{k}\right)^d e^{d \cdot \phi_f^{\mathrm{D}}}$$

and therefore,

$$\det(X)^{\frac{1}{d}} \geq \frac{k-d+1}{k} \cdot \phi^{\mathrm{D}}.$$

Before we prove Theorem 7, we begin with a few definitions. Let (I, X) be the returned solution of the algorithm. Let V_I be the $d \times |I|$ matrix whose columns are v_i for each $i \in I$. Observe that $X = V_I V_I^\top$ and X is invertible since $\det(X) > 0$ at the beginning of the algorithm and $\det(X)$ only increases in later iterations. We let $\tau_i = v_i^\top X^{-1} v_i$ for any $1 \leq i \leq n$. Observe that if $i \in I$, then τ_i is the leverage score of row v_i with respect to the matrix $V_I^\top$. We also let $\tau_{ij} = v_i^\top X^{-1} v_j$ for any $1 \leq i, j \leq n$.

$$\begin{aligned} \max_{x\in\mathbb{R}^n} \quad & \frac{1}{d}\log\det\left(\sum_{i=1}^n x_i v_i v_i^\top\right) & & \\ & \sum_{i=1}^n x_i \leq k & & \\ & x_i \geq 0 & & i\in[1,n] \end{aligned}$$

(*a*) Convex relaxation (D-REL) for D-DESIGN

$$\begin{aligned} \min_{\substack{\mu\in\mathbb{R}\\ Y\in\mathbb{R}^{d\times d}}} \quad & \frac{1}{d}\log\det(Y) + \frac{k}{d}\mu - 1 & & \\ & \mu - v_i^\top Y^{-1} v_i \geq 0 & & i\in[1,n] \\ & Y \succeq 0 & & \end{aligned}$$

(*b*) Dual (D-REL-DUAL) of (D-REL)

Figure 3: Convex Relaxation and its Dual for the D-DESIGN problem

Notations: For convenience, we summarize the notations used in this section.

- ϕ_f^{D} is the common optimum value of (D-REL) and its dual (D-REL-DUAL).
- $I^\star \subseteq [1,n]$ is the set of indices of the vectors in the optimal solution.
- $\phi^{\mathrm{D}} = \det\left(\sum_{i\in I^\star} v_i v_i^\top\right)^{\frac{1}{d}}$, the integral optimum value of D-DESIGN
- $I \subseteq [1,n]$, $X = \sum_{i\in I} v_i v_i^\top$ is the solution returned by the algorithm.
- For $1 \leq i \leq n$, $\tau_i = v_i^\top X^{-1} v_i$.
- For $1 \leq i,j \leq n$, $\tau_{ij} = v_i^\top X^{-1} v_j$.

The following lemma states standard properties about leverage scores of vectors with respect to the PSD matrix $X = \sum_{i\in I} v_i v_i^\top$ (see for example Drineas et al. (2012)). These results hold even when X is not an output from a local search algorithm and the proof is included in the appendix.

Lemma 8 *Let $v_1,\ldots,v_n \in \mathbb{R}^d$ and $I \subseteq [n]$. For any matrix $X = \sum_{i\in I} v_i v_i^\top$, we have:*

1. *For any $i \in I$, we have $\tau_i \leq 1$. Moreover, for any $i \in I$, $\tau_i = 1$ if and only if $X - v_i v_i^\top$ is singular.*
2. *We have $\sum_{i\in I} \tau_i = d$.*
3. *For any $1 \leq j \leq n$, we have $\sum_{i\in I} \tau_{ij}\tau_{ji} = \tau_j$.*
4. *For any $1 \leq i,j \leq n$, we have $\tau_{ij} = \tau_{ji}$ and $\tau_{ij} \leq \sqrt{\tau_i\tau_j}$.*

We now prove an upper bound on τ_j for the local optimal solution. This lemma utilizes the local optimality condition crucially.

Lemma 9 *For any $j \in [1,n]$, $\tau_j \leq \frac{d}{k-d+1}$.*

Before we prove the lemma, we complete the proof of Theorem 7 using Lemma 9.

Proof [Theorem 7] We construct a feasible solution to the (D-REL-DUAL) of the objective value at most $\frac{1}{d}\log\det(X) + \log\frac{k}{k-d+1}$. This would imply that

$$\phi_f^{\mathrm{D}} \le \frac{1}{d}\log\det(X) + \log\frac{k}{k-d+1}$$

which proves the first part of the theorem. The second part follows since $\phi_f^{\mathrm{D}} \ge \log\phi^{\mathrm{D}}$.

Let $Y = \alpha X, \mu = \max_{1\le j\le n} v_j^\top Y^{-1} v_j = \frac{1}{\alpha}\max_{j\in[1,n]} v_j^\top X^{-1} v_j$ where $\alpha > 0$ will be fixed later. Then, (Y,μ) is a feasible solution of (D-REL-DUAL). Hence,

$$\begin{aligned}\phi_f^{\mathrm{D}} &\le \frac{1}{d}\log\det(\alpha X) + \frac{k}{d}\cdot\frac{1}{\alpha}\max_{j\in[1,n]} v_j^\top X^{-1} v_j - 1\\ &\le \log\alpha + \frac{1}{d}\log\det(X) + \frac{k}{d\alpha}\cdot\frac{d}{k-d+1} - 1 \qquad \text{(Lemma 9)}\end{aligned}$$

Setting $\alpha = \frac{k}{k-d+1}$, we get

$$\phi_f^{\mathrm{D}} \le \log\frac{k}{k-d+1} + \frac{1}{d}\log\det(X) + 1 - 1 = \log\frac{k}{k-d+1} + \frac{1}{d}\log\det(X)$$

as required. ■

We now prove Lemma 9.

Proof [Lemma 9] Since X is a symmetric matrix, X^{-1} is also a symmetric matrix and therefore $\tau_{ij} = \tau_{ji}$ for each i, j. We first show that the local optimality condition implies the following claim:

Claim 1 *For any $i \in I$ and $1 \le j \le n$, we have $\tau_j - \tau_i\tau_j + \tau_{ij}\tau_{ji} \le \tau_i$.*

Proof Let $i \in I, j \in [1,n]$. By local optimality of I,

$$\det(X - v_i v_i^\top + v_j v_j^\top) \le \det(X).$$

Next we cite the following lemma for a determinant formula.

Lemma 10 *(Matrix Determinant Lemma, Harville (1997)) For any invertible matrix $A \in \mathbb{R}^{d\times d}$ and $a, b \in \mathbb{R}^d$,*

$$\det(A + ab^\top) = \det(A)(1 + b^\top A^{-1} a)$$

Applying the Lemma twice to $\det(X - v_i v_i^\top + v_j v_j^\top)$, the local optimality condition implies that

$$\begin{aligned}\det(X) \ge \det(X - v_i v_i^\top + v_j v_j^\top) &= \det(X + v_j v_j^\top)(1 - v_i^\top (X + v_j v_j^\top)^{-1} v_i)\\ &= \det(X)(1 + v_j^\top X^{-1} v_j)(1 - v_i^\top (X + v_j v_j^\top)^{-1} v_i)\end{aligned}$$

Hence, $(1 + v_j^\top X^{-1} v_j)(1 - v_i^\top (X + v_j v_j^\top)^{-1} v_i) \le 1$. Applying Sherman-Morrison formula, we get

$$\begin{aligned}(1 + v_j^\top X^{-1} v_j)\left(1 - v_i^\top\left(X^{-1} - \frac{X^{-1} v_j v_j^\top X^{-1}}{1 + v_j^\top X^{-1} v_j}\right) v_i\right) &\le 1\\ (1 + \tau_j)\left(1 - \tau_i + \frac{\tau_{ij}\tau_{ji}}{1 + \tau_j}\right) &\le 1\\ (1 - \tau_i)(1 + \tau_j) + \tau_{ij}\tau_{ji} &\le 1\\ \tau_j - \tau_i\tau_j + \tau_{ij}\tau_{ji} &\le \tau_i.\end{aligned}$$

This finishes the proof of Claim 1. ■

Now summing the inequality in Claim 1 over all $i \in I$, we get

$$\sum_{i \in I} (\tau_j - \tau_i \tau_j + \tau_{ij} \tau_{ji}) \leq \sum_{i \in I} \tau_i.$$

Applying Lemma 8, we obtain that $k\tau_j - d\tau_j + \tau_j \leq d$. Rearranging, we obtain that

$$\tau_j \leq \frac{d}{k - d + 1}$$

as desired. ■

2.3. D-DESIGN without Repetitions

Due to space constraints, we defer the proof of local search for D-DESIGN without repetitions to the appendix.

3. Local Search for A-DESIGN

In this section, we prove the performance of modified local search, presented earlier as Theorem 3. As remarked earlier, we need to modify the instance to cap the length of the vectors before applying the local search procedure. This is done in Section 3.1. We show that the value of any feasible solution only increases after capping. Moreover, the value of the natural convex programming relaxation increases by at most a small factor. We then analyze that the local search algorithm applied to vectors of short length returns a near optimal solution. Combining these facts give a complete analysis of modified local search for A-DESIGN in Section 3.2 which implies Theorem 3.

3.1. Capping Vectors

Algorithm 2 Capping vectors length for A-DESIGN

Input: $V = \{v_1, \ldots, v_n\} \subseteq \mathbb{R}^d$, parameter Δ.
while $\exists i \in [1, n], ||v_i||_2^2 > \Delta$ **do**
  $t = \text{argmax}_{i \in [n]} ||v_i||_2$.
  For $j \in [1, n], v_j = \left(I_d - \frac{1}{2} \frac{v_t v_t^\top}{||v_t||_2^2} \right) v_j$
end while
For $j \in [1, n], u_j = v_j$.
Return $U = \{u_1, \ldots, u_n\} \subseteq \mathbb{R}^d$

The algorithm to cap the length of input vectors is given in Algorithm 2. In each iteration, it considers the longest vector v_t. If the length of this vector (and thus every vector) is at most Δ, then it returns the current updated vectors. Else, it scales down all the vectors along the direction of the longest vector. Here, I_d denotes the d-by-d identity matrix.

Before we give the guarantee about the algorithm, we introduce the convex program for the A-DESIGN problem in Figure 6 (see Chapter 7 of Boyd and Vandenberghe (2004)). For any input

$$\begin{aligned} & A\text{-}\mathbf{REL}(V) \\ \min_{x\in\mathbb{R}^n} \quad & \operatorname{tr}\left(\left(\sum_{i=1}^{n} x_i v_i v_i^\top\right)^{-1}\right) \\ & \sum_{i=1}^{n} x_i \leq k \\ & x_i \geq 0 \quad i\in[n] \end{aligned}$$

(*a*) Convex relaxation A-REL(V) for A-DESIGN

$$\begin{aligned} & A\text{-}\mathbf{REL\text{-}DUAL}(V) \\ \max_{\substack{\lambda\in\mathbb{R}^n \\ Y\in\mathbb{R}^{d\times d}}} \quad & 2\operatorname{tr}\left(Y^{1/2}\right) - k\lambda \\ & \lambda - v_i^\top Y v_i \geq 0 \quad i\in[n] \\ & Y \succeq 0 \end{aligned}$$

(*b*) Dual A-REL-DUAL(V) of A-REL(V)

Figure 6: Convex Relaxation and its Dual for the A-DESIGN problem

vectors $V = \{v_1, \ldots, v_n\}$, the primal program is A-REL(V) and the dual program is A-REL-DUAL(V). We index these convex programs by input vectors V as we will analyze their objectives when the input vectors change by the capping algorithm. We let $\phi_f^{\mathrm{A}}(V)$ denote the (common) optimal objective value of both convex programs with input vectors V.

We prove the following guarantee about Algorithm 2. The proof along with some intuition of Algorithm 2 appears in the appendix.

Lemma 11 *For any input vectors $V = \{v_1, \ldots, v_n\} \subseteq \mathbb{R}^d$ and $k \geq d$, if $k \geq 15$ then the capping algorithm returns a set of vectors $U = \{u_1, \ldots u_n\}$ such that*

1. $\|u_i\|_2^2 \leq \Delta$ *for all $i \in [n]$.*
2. *For any (multi-)set $S \subseteq [n]$,* $\operatorname{tr}\left(\left(\sum_{i\in S} v_i v_i^\top\right)^{-1}\right) \leq \operatorname{tr}\left(\left(\sum_{i\in S} u_i u_i^\top\right)^{-1}\right)$.
3. $\phi_f^{\mathrm{A}}(U) \leq \left(1 + \frac{3000\cdot d}{k}\right)\left(\phi_f^{\mathrm{A}}(V) + \frac{135\cdot d}{\Delta}\right)$.

Lemma 11 states that if an algorithm returns a good solution from capped vectors, then the objective remains small after we map the solution back to the original (uncapped) input vectors. Moreover, by choosing a sufficiently large capping length Δ, we may bound the increase in optimal value of the natural convex programming relaxation after capping by a small factor. Optimizing for Δ is to be done later.

3.2. Local Search Algorithm

We now consider the local search algorithm with the capped vectors. The performance of the algorithm is stated as follows.

Theorem 12 *Let (I, X) be the solution returned by Algorithm 3. If $||u_i||_2^2 \leq \Delta$ for all $i \in [n]$,*

$$\operatorname{tr}(X^{-1}) \leq \phi_f^{\mathrm{A}}(U)\left(\left(1 - \frac{d-2}{k}\right) - \sqrt{\frac{\Delta\phi_f^{\mathrm{A}}(U)}{k}}\right)^{-1}.$$

Algorithm 3 Local search algorithm for A-DESIGN with capped vectors

Input: $U = \{u_1, \dots, u_n\} \subseteq \mathbb{R}^d, d \le k \in \mathbb{N}$.
Let I be any (multi)-subset of $[1, n]$ of size k such that $X = \sum_{i \in I} u_i u_i^\top$ is nonsingular.
while $\exists i \in I, j \in [1, n]$ such that $\operatorname{tr}\left((X - u_i u_i^\top + u_j u_j^\top)^{-1}\right) < \operatorname{tr}(X^{-1})$ **do**
  $X = X - u_i u_i^\top + u_j u_j^\top$
  $I = I \setminus \{i\} \cup \{j\}$
end while
Return (I, X)

The proof of Theorem 12 is deferred to the appendix. We now analyze the modified local search algorithm presented as Algorithm 4 with input vectors $V = \{v_1, \dots, v_n\}$ which may contain vectors with long length using Theorem 12. Let $I^\star$ be the set of indices of the vectors in the optimal solution of A-DESIGN with input vector set V and let $\phi^{\mathrm{A}}(V) = \operatorname{tr}\left(\left(\sum_{i \in I^\star} v_i v_i^\top\right)^{-1}\right)$ be its objective. Observe that $\phi_f^{\mathrm{A}}(V) \le \phi^{\mathrm{A}}(V)$.

Algorithm 4 Modified local search algorithm for A-DESIGN

Input: $V = \{v_1, \dots, v_n\}, d \le k \in \mathbb{N}$.
Let $\Delta = \frac{d}{\epsilon^2 \phi^{\mathrm{A}}(V)}$.
Let $U = \{u_1, \dots, u_n\}$ be the output of Vector Capping Algorithm 2 with input (V, Δ).
Let $I \subseteq [1, n], X = \sum_{i \in I} u_i u_i^\top$ be the output of Local Search Algorithm 3 with input (U, k).
Return I.

Theorem 13 *For input vectors $V = \{v_1, \dots, v_n\}$ where $v_i \in \mathbb{R}^d$ and parameter k, let I be the solution returned by Algorithm 4. If $k \ge \frac{2d}{\epsilon^4}$ and $\epsilon \le 0.001$, then*

$$\operatorname{tr}\left(\left(\sum_{i \in I} v_i v_i^\top\right)^{-1}\right) \le (1 + \epsilon)\phi^{\mathrm{A}}(V).$$

The $(1 + \epsilon)$-approximation of Algorithm 4 is achieved by setting an appropriate capping length Δ and combining the guarantees from Lemma 11 and Theorem 12.
Proof By Theorem 12,

$$\begin{aligned}
\operatorname{tr}\left(\left(\sum_{i \in I} u_i u_i^\top\right)^{-1}\right) &\le \phi_f^{\mathrm{A}}(U) \left(1 - \frac{d-2}{k} - \sqrt{\frac{\Delta \phi_f^{\mathrm{A}}(U)}{k}}\right)^{-1} \\
&= \phi_f^{\mathrm{A}}(U) \left(1 - \frac{\epsilon^4}{2} + \frac{\epsilon^4}{d} - \epsilon \sqrt{\frac{\phi_f^{\mathrm{A}}(U)}{2\phi^{\mathrm{A}}(V)}}\right)^{-1}
\end{aligned}$$

The last inequality follows since $k \ge \frac{2d}{\epsilon^4}$ and $\Delta = \frac{d}{\epsilon^2 \phi^{\mathrm{A}}(V)}$. By Lemma 11,

$$\phi_f^{\mathrm{A}}(U) \le \left(1 + 1500\epsilon^4\right) \left(\phi_f^{\mathrm{A}}(V) + 135\epsilon^2 \phi_f^{\mathrm{A}}(V)\right).$$

Since $\phi_f^{\mathsf{A}}(V) \leq \phi^{\mathsf{A}}(V)$, we get $\phi_f^{\mathsf{A}}(U) \leq (1+1500\epsilon^4)(1+135\epsilon^2)\phi^{\mathsf{A}}(V)$. Substituting in the equation above, we get

$$\begin{aligned}\operatorname{tr}\left(\left(\sum_{i\in I} u_i u_i^\top\right)^{-1}\right) &\leq \phi^{\mathsf{A}}(V)\frac{(1+1500\epsilon^4)(1+135\epsilon^2)}{1-\frac{\epsilon^4}{2}+\epsilon^4/d-\epsilon\sqrt{(1+1500\epsilon^4)(1+135\epsilon^2)/2}} \\ &\leq (1+\epsilon)\phi^{\mathsf{A}}(V)\end{aligned}$$

where the last inequality follows from the fact that $\epsilon < 0.001$. By Lemma 11, we also have that $\operatorname{tr}\left(\left(\sum_{i\in I} v_i v_i^\top\right)^{-1}\right) \leq \operatorname{tr}\left(\left(\sum_{i\in I} u_i u_i^\top\right)^{-1}\right)$. Hence,

$$\operatorname{tr}\left(\left(\sum_{i\in I} v_i v_i^\top\right)^{-1}\right) \leq (1+\epsilon)\phi^{\mathsf{A}}(V).$$

This finishes the proof of Theorem 13. ■

Algorithm 4 requires the knowledge of the optimum solution value $\phi^{\mathsf{A}}(V)$. We can guess this value efficiently by performing a binary search. The details appear in the appendix.

3.3. Instances with Bad Local Optima

In this section, we show that preprocessing input vectors to the A-DESIGN problem is required for the local search algorithm to have any approximation guarantee. This is because a locally optimal solution can give an arbitrarily bad objective value compared to the optimum. Hence, this requirement applies regardless of implementations of the local search algorithm. We summarize the result as follows.

Theorem 14 *For any $k \geq d \geq 2$, there exists an instance of A-DESIGN, either with or without repetitions, such that a locally optimal solution has an arbitrarily bad approximation ratio.*

We note that any instance to A-DESIGN with repetitions can be used for A-DESIGN without repetitions by making k copies of each input vector. Therefore, it is enough to show example of instances only in A-DESIGN with repetitions. For each i, let e_i be the unit vector in the i^{th} dimension. In this section, N is a real number tending to infinity, and the $A(N) \sim B(N)$ notation indicates that $\lim_{N\to\infty} \frac{A(N)}{B(N)} = 1$. All asymptotic notions such as big-Oh are with respect to $N \to \infty$. We first show the bad instance when $k \geq d = 2$. Though $d = 2$ seems a small case to consider, the calculation presented is central to prove the main theorem later.

Lemma 15 *There exists an instance of A-DESIGN for $k \geq d = 2$, with repetitions, such that a locally optimal solution has an arbitrarily bad approximation ratio.*

The construction in Lemma 15 can be generalized to $d > 2$ dimensions by adding a vector with an appropriate length to each additional dimension. The proof of Theorem 14 appears in the appendix. We now prove the Lemma.

Proof Let $v_1 = [1; \frac{1}{N^2}], v_2 = [1; -\frac{1}{N^2}]$, $w_1 = [N^4; \frac{1}{N}], w_2 = [N^4; -\frac{1}{N}]$, and let the input of A-DESIGN be these four vectors. We first make straightforward calculations, summarized as the following claim.

Claim 2 *Let p, q be positive integers. Then,*

$$\operatorname{tr}\left(\left(pv_1v_1^\top + qv_2v_2^\top\right)^{-1}\right) = \frac{p+q}{4pq}N^4 + O(1) \tag{1}$$

$$\operatorname{tr}\left(\left(pv_1v_1^\top + qv_2v_2^\top + w_1w_1^\top\right)^{-1}\right) = \frac{1}{p+q}N^4 + O(N) \tag{2}$$

$$\operatorname{tr}\left(\left(pv_1v_1^\top + qv_2v_2^\top + w_2w_2^\top\right)^{-1}\right) = \frac{1}{p+q}N^4 + O(N) \tag{3}$$

$$\operatorname{tr}\left(\left(w_1w_1^\top + w_2w_2^\top\right)^{-1}\right) = \frac{N^2}{2} + O(N^{-8}) \tag{4}$$

Proof We will repeatedly use the formula $\operatorname{tr}\left(\begin{bmatrix} a & b \\ c & d \end{bmatrix}^{-1}\right) = \frac{a+d}{ad-bc}$. We have

$$\begin{aligned}\operatorname{tr}\left(\left(pv_1v_1^\top + qv_2v_2^\top\right)^{-1}\right) &= \operatorname{tr}\left(\begin{bmatrix} p+q & (p-q)N^{-2} \\ (p-q)N^{-2} & (p+q)N^{-4} \end{bmatrix}^{-1}\right) \\ &= \frac{p+q+(p+q)N^{-4}}{(p+q)^2N^{-4} - (p-q)^2N^{-4}} = \frac{p+q}{4pq}N^4 + O(1)\end{aligned}$$

$$\begin{aligned}\operatorname{tr}\left(\left(pv_1v_1^\top + qv_2v_2^\top + w_1w_1^\top\right)^{-1}\right) &= \operatorname{tr}\left(\begin{bmatrix} N^8+p+q & N^3+(p-q)N^{-2} \\ N^3+(p-q)N^{-2} & N^{-2}+(p+q)N^{-4} \end{bmatrix}^{-1}\right) \\ &= \frac{N^8+O(1)}{(p+q)N^4+O(N)} = \frac{1}{p+q}N^4 + O(N)\end{aligned}$$

The calculation for $\operatorname{tr}\left(\left(pv_1v_1^\top + qv_2v_2^\top + w_2w_2^\top\right)^{-1}\right)$ is symmetric. Finally, we have

$$\operatorname{tr}\left(w_1w_1^\top + w_2w_2^\top\right)^{-1} = \operatorname{tr}\left(\begin{bmatrix} 2N^8 & 0 \\ 0 & 2N^{-2} \end{bmatrix}^{-1}\right) = \frac{N^2}{2} + \frac{1}{2N^8}$$

finishing the proof. ■

We now continue the proof of Lemma 15. Let $p = \lfloor \frac{k}{2} \rfloor, q = \lceil \frac{k}{2} \rceil$ and consider the solution S which has p and q copies of v_1 and v_2 respectively. By Claim 2, the current objective of S is $\operatorname{tr}\left(\left(pv_1v_1^\top + qv_2v_2^\top\right)^{-1}\right) \sim \frac{k}{4pq}N^4$ and the objective of $S \setminus \{v_i\} \cup \{w_j\}$ for any pair $i, j \in \{1, 2\}$ is $\frac{1}{p+q-1}N^4 + O(N) \sim \frac{1}{k-1}N^4$. As $\frac{k}{4pq}N^4 \geq \frac{k}{k^2-1}N^4 > \frac{1}{k-1}N^4$ for $k \geq 2$, S is locally optimal.

However, consider another solution S^* which picks p and q copies of w_1 and w_2. Since $\operatorname{tr}\left(w_1w_1^\top + w_2w_2^\top\right)^{-1} = O(N^2)$, by monotonicity of $\operatorname{tr}((\cdot)^{-1})$ under Loewner ordering, we must have that the objective given by S^* is also at most $O(N^2)$, which is a $\Theta(N^2)$-factor smaller than the objective value of S. The result follows because N tends to infinity. ■

4. Acknowledgement

Vivek Madan, Mohit Singh, and Uthaipon (Tao) Tantipongpipat are supported by NSF grant 1717947.

References

Zeyuan Allen-Zhu, Yuanzhi Li, Aarti Singh, and Yining Wang. Near-optimal design of experiments via regret minimization. In Doina Precup and Yee Whye Teh, editors, *Proceedings of the 34th International Conference on Machine Learning, ICML 2017, Sydney, NSW, Australia, 6-11 August 2017*, volume 70 of *Proceedings of Machine Learning Research*, pages 126–135. PMLR, 2017.

Anthony Atkinson, Alexander Donev, and Randall Tobias. *Optimum experimental designs, with SAS*, volume 34. Oxford University Press, 2007.

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

Dennis S Bernstein. *Matrix mathematics: Theory, facts, and formulas with application to linear systems theory*, volume 41. Princeton university press Princeton, 2005.

Mustapha Bouhtou, Stephane Gaubert, and Guillaume Sagnol. Submodularity and randomized rounding techniques for optimal experimental design. *Electronic Notes in Discrete Mathematics*, 36:679–686, 2010.

Stephen Boyd and Lieven Vandenberghe. *Convex optimization*. Cambridge university press, 2004.

Ali Çivril and Malik Magdon-Ismail. On selecting a maximum volume sub-matrix of a matrix and related problems. *Theoretical Computer Science*, 410(47-49):4801–4811, 2009.

R Dennis Cook and Christopher J Nachtrheim. A comparison of algorithms for constructing exact D-optimal designs. *Technometrics*, 22(3):315–324, 1980.

Michał Dereziński and Manfred K Warmuth. Subsampling for ridge regression via regularized volume sampling. *arXiv preprint arXiv:1710.05110*, 2017.

Petros Drineas, Malik Magdon-Ismail, Michael W Mahoney, and David P Woodruff. Fast approximation of matrix coherence and statistical leverage. *Journal of Machine Learning Research*, 13 (Dec):3475–3506, 2012.

Walter Theodore Federer et al. *Experimental design.* Macmillan Co., New York and London, 1955.

Valerii Vadimovich Fedorov. *Theory of optimal experiments*. Elsevier, 1972.

David A Harville. *Matrix algebra from a statistician's perspective*, volume 1. Springer, 1997.

Carl D Meyer, Jr. Generalized inversion of modified matrices. *SIAM Journal on Applied Mathematics*, 24(3):315–323, 1973.

Alan J Miller and Nam-Ky Nguyen. Algorithm as 295: A fedorov exchange algorithm for D-optimal design. *Journal of the royal statistical society. series c (applied statistics)*, 43(4):669–677, 1994.

Toby J Mitchell and FL Miller Jr. Use of design repair to construct designs for special linear models. *Math. Div. Ann. Progr. Rept.(ORNL-4661)*, 13, 1970.

Nam-Ky Nguyen and Alan J Miller. A review of some exchange algorithms for constructing discrete D-optimal designs. *Computational Statistics & Data Analysis*, 14(4):489–498, 1992.

Aleksandar Nikolov, Mohit Singh, and Uthaipon Tao Tantipongpipat. Proportional volume sampling and approximation algorithms for A-optimal design. In *Proceedings of the Thirtieth Annual ACM-SIAM Symposium on Discrete Algorithms*, pages 1369–1386. SIAM, 2019.

Friedrich Pukelsheim. *Optimal design of experiments*. SIAM, 2006.

Guillaume Sagnol and Radoslav Harman. Computing exact d-optimal designs by mixed integer second-order cone programming. *The Annals of Statistics*, 43(5):2198–2224, 2015.

Alexander Schrijver. *Theory of linear and integer programming*. John Wiley & Sons, 1998.

Mohit Singh and Weijun Xie. Approximate positive correlated distributions and approximation algorithms for D-optimal design. *To appear in SODA*, 2018.

Yining Wang, Adams Wei Yu, and Aarti Singh. On computationally tractable selection of experiments in regression models. *arXiv preprint arXiv:1601.02068*, 2016.

William J Welch. Algorithmic complexity: three np-hard problems in computational statistics. *Journal of Statistical Computation and Simulation*, 15(1):17–25, 1982.

Appendix A. Proofs from Section 2

We use the notation $\langle A, B\rangle$ for an inner product of two matrices A, B of the same size. We begin by stating the Sherman-Morrison formula that is important in our calculations. We instantiate it for symmetric matrices.

Theorem 16 *Let L be an $d \times d$ invertible matrix and $v \in \mathbb{R}^d$. Then*

$$\left(L + vv^\top\right)^{-1} = L^{-1} - \frac{L^{-1}vv^\top L^{-1}}{1 + v^\top L^{-1}v}$$

Lemma 17 *(Matrix Determinant Lemma, Harville (1997)) For any invertible matrix $L \in \mathbb{R}^{d\times d}$ and $v \in \mathbb{R}^d$,*

$$\det(L + vv^\top) = \det(L)(1 + v^\top L^{-1}v)$$

We now detail the missing proofs.

Proof [Lemma 8] Let $W = X_{-i} = X - v_iv_i^\top = \sum_{j\in I\setminus\{i\}} v_jv_j^\top$. To show $\tau_i \leq 1$, we make two cases depending on whether W is singular or not.

Case 1: W is non-singular.

$$\begin{aligned}
\tau_i &= v_i^\top (W + v_i v_i^\top)^{-1} v_i \\
&= v_i^\top \left(W^{-1} - \frac{W^{-1} v_i v_i^\top W^{-1}}{1 + v_i^\top W^{-1} v_i} \right) v_i \\
&= v_i^\top W^{-1} v_i - \frac{v_i^\top W^{-1} v_i v_i^\top W^{-1} v_i}{1 + v_i^\top W^{-1} v_i} \\
&= \frac{v_i^\top W^{-1} v_i + (v_i^\top W^{-1} v_i)^2 - (v_i^\top W^{-1} v_i)^2}{1 + v_i^\top W^{-1} v_i} \\
&= \frac{v_i^\top W^{-1} v_i}{1 + v_i^\top W^{-1} v_i} \\
&< 1.
\end{aligned}$$

Last inequality follows from the fact that $v_i^\top W^{-1} v_i > 0$ since W^{-1} is non-singular.

Case 2: W is singular. We have that X is non-singular and $W = X - v_i v_i^\top$ is a singular matrix. Let $Y^\dagger$ denote the Moore-Penrose pseudo-inverse of Y for any matrix Y. Observe that $X^\dagger = X^{-1}$. From Theorem 1 (Meyer, 1973), we have that

$$X^{-1} = W^\dagger - \frac{W^\dagger v_i v_i^\top (I - WW^\dagger)^\top}{\|(I - WW^\dagger) v_i\|_2^2} - \frac{(I - W^\dagger W)^\top v_i v_i^\top W^\dagger}{\|(I - W^\dagger W)^\top v_i\|_2^2} + \frac{(1 + v_i^\top W^\dagger v_i)(I - W^\dagger W)^\top v_i v_i^\top (I - WW^\dagger)^\top}{\|(I - W^\dagger W)^\top v_i\|_2^2 \|(I - WW^\dagger) v_i\|_2^2}$$

Now we use the fact that $(I - WW^\dagger)$ and $(I - W^\dagger W)$ are projection matrices. Since $v^\top P v = \|Pv\|_2^2$ for any projection matrix P and vector v, we obtain that

$$\begin{aligned}
v_i^\top X^{-1} v_i &= v_i^\top W^\dagger v_i - \frac{\left(v_i^\top W^\dagger v_i\right) \left(v_i^\top (I - WW^\dagger)^\top v_i\right)}{\|(I - WW^\dagger) v_i\|_2^2} - \frac{\left(v_i^\top (I - W^\dagger W)^\top v_i\right) v_i^\top W^\dagger v_i}{\|(I - W^\dagger W)^\top v_i\|_2^2} \\
&\quad + \frac{(1 + v_i^\top W^\dagger v_i) v_i^\top (I - W^\dagger W)^\top v_i v_i^\top (I - WW^\dagger)^\top v_i}{\|(I - W^\dagger W)^\top v_i\|_2^2 \|(I - WW^\dagger) v_i\|_2^2} \\
&= v_i^\top W^\dagger v_i - v_i^\top W^\dagger v_i - v_i^\top W^\dagger v_i + (1 + v_i^\top W^\dagger v_i) \\
&= 1
\end{aligned}$$

as claimed.

We now show that $\sum_{i \in I} \tau_i = d$. Indeed

$$\sum_{i \in I} \tau_i = \sum_{i \in I} v_i^\top X^{-1} v_i = \sum_{i \in I} \langle X^{-1}, v_i v_i^\top \rangle = \langle X^{-1}, \sum_{i \in I} v_i v_i^\top \rangle = \langle X^{-1}, X \rangle = d$$

Similarly, we have

$$\max \quad \frac{1}{d}\log\det\left(\sum_{i=1}^{n} x_i v_i v_i^\top\right)$$
$$\sum_{i=1}^{n} x_i \quad \leq k$$
$$1 \geq \ x_i \quad \geq 0 \qquad i \in [1,n]$$

(*a*) Convex relaxation (D-REL) for D-DESIGN

$$\min \quad \frac{1}{d}\log\det(Y) + \frac{k}{d}\mu + \frac{1}{d}\sum_{i=1}^{n}\eta_i - 1$$
$$\mu + \eta_i - v_i^\top Y^{-1} v_i \quad \geq 0 \quad i \in [1,n]$$
$$\eta_i \quad \geq 0 \quad i \in [1,n]$$
$$Y \quad \succeq 0$$

(*b*) Dual (D-REL-DUAL) of (D-REL)

Figure 9: Convex Relaxation and its Dual for the D-DESIGN problem without repetitions

$$\sum_{i\in I}\tau_{ij}\tau_{ji} = \sum_{i\in I} v_i^\top X^{-1} v_j v_j^\top X^{-1} v_i = \sum_{i\in I}\langle X^{-1} v_j v_j^\top X^{-1}, v_i v_i^\top\rangle = \langle X^{-1} v_j v_j^\top X^{-1}, \sum_{i\in I} v_i v_i^\top\rangle$$
$$= \langle X^{-1} v_j v_j^\top X^{-1}, X\rangle = v_j^\top X^{-1} v_j$$

For the last part, observe that X^{-1} is symmetric and thus $\tau_{ij} = \tau_{ji}$. Moreover,

$$\tau_{ij} = v_i^\top X^{-1} v_j = (X^{-\frac{1}{2}} v_i)^\top (X^{-\frac{1}{2}} v_j) \leq \|X^{-\frac{1}{2}} v_i\|_2 \|X^{-\frac{1}{2}} v_j\| = \sqrt{\tau_i \tau_j}$$

where the inequality follows from Cauchy-Schwarz. ■

A.1. Local Search for D-DESIGN without Repetitions

In this section, we focus on the variant of D-DESIGN where repetitions of vectors are not allowed, and show the approximation guarantee of the local search in this setting. In comparison to D-DESIGN with repetitions, the relaxation now has an upper bound on x_i and extra nonnegative variables η_i on the dual.

The local search algorithm 1 is modified by considering a swap where elements to be included in the set must not be in the current set. We prove a similar approximation ratio of the local search algorithm for the without repetition setting.

Theorem 18 *Let X be the solution returned by the local search algorithm. Then for all $k \geq d+1$,*

$$\det(X) \geq \left(\frac{k-d}{k}\right)^d e^{d\cdot\phi_f^{\mathrm{D}}}$$

and therefore,

$$\det(X)^{\frac{1}{d}} \geq \frac{k-d}{k}\cdot\phi^{\mathrm{D}}.$$

We note that in the case $k = d$, the design problem without repetition is identical to with repetition since the optimal solution must be linearly independent, and thus the bound from with repetitions of Theorem 7 applies to obtain d-approximation.

The proof of Theorem 18 is similar to D design requires a different bound on τ_j from the setting with repetitions to set a feasible dual solution, since the local search condition no longer applies to all vectors $j \in [n]$ but only for those not in output set I. We first give a bound of τ_j for $j \notin I$.

Lemma 19 *For any $j \notin S$ and any $i \in S$ such that $\tau_i < 1$,*

$$\tau_j \leq \frac{\tau_i}{1 - \tau_i}.$$

Proof We claim that the local search condition implies that for any $i \in I$ and $j \notin I$, we have

$$\tau_j - \tau_i \tau_j + \tau_{ij} \tau_{ji} \leq \tau_i. \tag{5}$$

The proof of the claim is identical to that of Claim 1. Hence, we have

$$\tau_i \geq \tau_j - \tau_i \tau_j + \tau_{ij}^2 \geq \tau_j - \tau_i \tau_j \tag{6}$$

which finishes the proof of the Lemma. ∎

We now prove the main Theorem.

Proof [Theorem 18]

As in the proof of Theorem 7, we construct a feasible solution to the (D-REL-DUAL) of the objective value of at most $\frac{1}{d} \log \det(X) + \log \frac{k}{k-d}$ which is sufficient as a proof of the theorem. Denote $\tau_{\min} = \min_{j \in I} v_j^\top Y^{-1} v_j$. Let

$$Y = \alpha X, \qquad \mu = \frac{k}{\alpha(k-d)} \tau_{\min}, \qquad \eta_j = \begin{cases} 0, & j \notin I \\ \frac{\tau_j - \tau_{\min}}{\alpha} & j \in I \end{cases}$$

where $\alpha > 0$ will be fixed later. We first check the feasibility of the solution. It is clear by definition that $\mu, \eta_j \geq 0$. For $j \notin I$, by Lemma 19, we have

$$v_j^\top Y^{-1} v_j = \frac{1}{\alpha} \cdot \tau_j \leq \frac{1}{\alpha} \cdot \frac{\tau_{\min}}{1 - \tau_{\min}} \leq \frac{1}{\alpha} \cdot \frac{k}{k-d} \tau_{\min} = \mu + \eta_j$$

where the second inequality follows from $\tau_{\min} \leq \frac{1}{k} \sum_{i \in I} \tau_i = \frac{d}{k}$. For $i \in I$, we have

$$\mu + \eta_i \geq \frac{1}{\alpha} \cdot (\tau_{\min} + \tau_i - \tau_{\min}) = v_i^\top Y^{-1} v_i$$

Therefore, the solution is dual feasible. This solution obtains the objective of $\frac{1}{d} \log \det(\alpha X) - 1 + \frac{k}{d} \mu + \frac{1}{d} \sum_{i=1}^n \eta_i$ which is equal to

$$\begin{aligned}
&= \frac{1}{d} \log \det(\alpha X) - 1 + \frac{k}{d} \frac{k}{\alpha(k-d)} \tau_{\min} + \frac{1}{\alpha d} \sum_{i \in I} (\tau_i - \tau_{\min}) \\
&= \frac{1}{d} \log \det(\alpha X) - 1 + \frac{k^2}{\alpha d(k-d)} \tau_{\min} + \frac{1}{\alpha d} (d - k \tau_{\min}) \\
&= \frac{1}{d} \log \det X + \log \alpha - 1 + \frac{1}{\alpha} \left(\frac{k}{k-d} \tau_{\min} + 1 \right) \\
&\leq \frac{1}{d} \log \det X + \log \alpha - 1 + \frac{k}{\alpha(k-d)}
\end{aligned}$$

where the last inequality is by $\tau_{\min} \leq \frac{d}{k}$. Finally, we set $\alpha = \frac{k}{k-d}$ to obtain the objective value of dual

$$\frac{1}{d}\log\det(X) + \log\frac{k}{k-d} - 1 + 1 = \frac{1}{d}\log\det(X) + \log\frac{k}{k-d}$$

as required.

■

Appendix B. Proofs from Section 3

B.1. Proof of Performance of Modified Local Search Algorithm for A-DESIGN

B.1.1. PROOF OF THEOREM 12

We first outline the proof of Theorem 12. Let (I, X) be the returned solution of the Algorithm 3. Observe that X is invertible since X is invertible at the beginning and $\mathrm{tr}(X^{-1})$ only decreases in the later iterations. Let $\tau_{ij} = u_i^\top X^{-1} u_j, h_{ij} = u_i^\top X^{-2} u_j, \tau_i = \tau_{ii}, h_i = h_{ii}$, and $\beta = tr(X^{-1})$. Since, X is a symmetric matrix, X^{-1} is also a symmetric matrix and therefore $\tau_{ij} = \tau_{ji}$ for each $i, j \in [n]$.

Notations For convenience, we restate the notations used in this section.

- V : Input to Modified Local Search Algorithm 4.
- $I^\star$: indices of the vectors in the optimal solution of A-DESIGN with input vector set V.
- $\phi^{\mathrm{A}}(V) = \mathrm{tr}\left(\left(\sum_{i\in I^\star} v_i v_i^\top\right)^{-1}\right)$.
- U : Output of Vector Capping Algorithm 2 and input to Local Search Algorithm with capped vectors 3.
- Δ : For every $i \in [1, n], ||u_i||_2^2 \leq \Delta$.
- (I, X) : Output of Local Search Algorithm with capped vectors 3 on input (U, k).
- $\phi_f^{\mathrm{A}}(U)$, and $\phi_f^{\mathrm{A}}(V)$ denote the (common) optimal value of objective values of the convex program with input vectors from V and U respectively.
- For $i, j \in [1, n], \tau_{ij} = u_i^\top X^{-1} u_j, h_{ij} = u_i^\top X^{-2} u_j$.
- For $i \in [n], \tau_i = \tau_{ii}, h_i = h_{ii}$.

Following lemma shows some standard connections between $\tau_{ij}, \tau_i, h_{ij}$ and h_i's. Proof of the lemma is presented in Section B.1.3.

Lemma 20 *We have the following.*

1. *For any $i \in I$, we have $\tau_i \leq 1$. Moreover, for any $i \in I$, $\tau_i = 1$ if and only if $X - v_i v_i^\top$ is singular.*

2. *We have* $\sum_{i\in I}\tau_i = d$.
3. *For any* $i, j \in [n]$, $h_i(1+\tau_j) - 2\tau_{ij}h_{ij} \geq 0$.
4. *For any* $j \in [n]$, *we have* $\sum_{i\in I}\tau_{ij}^2 = h_j$.
5. *We have* $\sum_{i\in I} h_i = \beta$.
6. *For any* $j \in [n]$, *we have* $\sum_{i\in I}\tau_{ij}h_{ij} = h_j$.
7. *For any* $j \in [n]$, *we have* $\tau_j \leq \sqrt{h_j}||u_j||_2$.
8. *For any* $i \in [n]$, *let* $X_{-i} = X - u_iu_i^\top$. *If* X_{-i} *is invertible, then for any* $j \in [n]$, *we have*
   - $u_j^\top X_{-i}^{-1}u_j = \frac{\tau_j+\tau_{ij}^2-\tau_i\tau_j}{1-\tau_i}$, *and*
   - $u_j^\top X_{-i}^{-2}u_j = h_j + \frac{h_i\tau_{ij}^2}{(1-\tau_i)^2} + \frac{2\tau_{ij}h_{ij}}{1-\tau_i}$.

Next lemma shows a lower bound on h_j in terms of β and $\phi_f^{\mathrm{A}}(U)$ by constructing a dual feasible solution.

Lemma 21 *We have* $\max_{j\in[n]} h_j \geq \frac{\beta^2}{k\cdot\phi_f^{\mathrm{A}}(U)}$.

Next lemma shows an upper bound on h_j in terms of β and τ_j using the local optimality condition.

Lemma 22 *For any* $j \in [n]$, $\frac{h_j}{1+\tau_j} \leq \frac{\beta}{k-d+2}$.

Before we prove these lemmas, we complete the proof of Theorem 12.

Proof [Theorem 12] By Lemma 22, for any $j \in [n]$, $\frac{h_j}{1+\tau_j} \leq \frac{\beta}{k-d+2}$. By Lemma 20, $\tau_j \leq \sqrt{h_j}||u_j||_2 \leq \sqrt{h_j\Delta}$. Hence, for any $j \in [n]$,

$$\frac{h_j}{1+\sqrt{h_j\Delta}} \leq \frac{\beta}{k-d+2}.$$

By Lemma 21, there exists $j \in [n]$ such that $h_j \geq \frac{\beta^2}{k\cdot\phi_f^{\mathrm{A}}(U)}$. Now we note the following claim.

Claim 3 $f(x) = \frac{x}{1+c\sqrt{x}}$ *is a monotonically increasing function for* $x \geq 0$ *if* $c \geq 0$.

Proof $f'(x) = \frac{1}{1+c\sqrt{x}} + x\cdot\frac{-1}{(1+c\sqrt{x})^2}\cdot\frac{c}{2\sqrt{x}} = \frac{2+c\sqrt{x}}{(1+c\sqrt{x})^2}$ which is always positive for $x \geq 0$ if $c \geq 0$. ∎

Hence, we have

$$\frac{\frac{\beta^2}{k\cdot\phi_f^{\mathrm{A}}(U)}}{1+\sqrt{\frac{\beta^2}{k\cdot\phi_f^{\mathrm{A}}(U)}\Delta}} \le \frac{\beta}{k-d+2}$$

$$\frac{k-d+2}{k}\frac{\beta}{\phi_f^{\mathrm{A}}(U)} \le 1+\sqrt{\frac{\Delta\phi_f^{\mathrm{A}}(U)}{k}}\frac{\beta}{\phi_f^{\mathrm{A}}(U)}$$

$$\left(1-\frac{d-2}{k}-\sqrt{\frac{\Delta\phi_f^{\mathrm{A}}(U)}{k}}\right)\frac{\beta}{\phi_f^{\mathrm{A}}(U)} \le 1$$

$$\mathrm{tr}(X^{-1}) = \beta \le \phi_f^{\mathrm{A}}(U)\left(1-\frac{d-2}{k}-\sqrt{\frac{\Delta\phi_f^{\mathrm{A}}(U)}{k}}\right)^{-1}.$$

This finishes the proof of Theorem 12. ■

Next, we prove Lemma 21 and Lemma 22.

Proof [Lemma 21] We prove the lemma by constructing a feasible solution to A-REL-DUAL(U). Let

$$Y = \gamma X^{-2}, \qquad \lambda = \max_{j\in[n]} u_j^\top Y u_j = \gamma \max_{j\in[n]} h_j$$

where $\gamma > 0$ will be fixed later. Then, (Y,λ) is a feasible solution to A-REL-DUAL(U). Hence,

$$\phi_f^{\mathrm{A}}(U) \ge 2\,\mathrm{tr}\left(\left(\gamma X^{-2}\right)^{1/2}\right) - k\gamma\max_{j\in[n]} h_j = 2\sqrt{\gamma}\beta - k\gamma\max_{j\in[n]} h_j.$$

Substituting $\gamma = \left(\frac{\beta}{k\max_{j\in[n]} h_j}\right)^2$, we get $\phi_f^{\mathrm{A}}(U) \ge \frac{\beta^2}{k\max_{j\in[n]} h_j}$. This gives us $\max_{j\in[n]} h_j \ge \frac{\beta^2}{k\phi_f^{\mathrm{A}}(U)}$ which is the desired inequality in Lemma 21. ■

Proof [Lemma 22] We start the proof by showing an inequality implied by the local optimality of the solution.

Claim 4 *For any $i\in I, j\in[n]$,*

$$h_i(1+\tau_j) - h_j(1-\tau_i) - 2\tau_{ij}h_{ij} \ge 0 \tag{7}$$

Proof For $i\in I$, let $X_{-i} = X - u_iu_i^\top$. First consider the case when X_{-i} is singular. From Lemma 20, $\tau_i = 1$ and $h_i(1+\tau_j) - 2\tau_{ij}h_{ij} \ge 0$. Hence,

$$h_i(1+\tau_j) - h_j(1-\tau_i) - 2\tau_{ij}h_{ij} \ge 0.$$

Now, consider the case when X_{-i} is non-singular. By local optimality condition, we have that for any $i\in I, j\in[n]$,

$$\beta \le \mathrm{tr}\left(\left(X_{-i}+u_ju_j^\top\right)^{-1}\right)$$

By Sherman-Morrison formula,

$$\operatorname{tr}\left(\left(X_{-i}+u_ju_j^\top\right)^{-1}\right)=\operatorname{tr}(X_{-i}^{-1})-\frac{u_j^\top X_{-i}^{-2}u_j}{1+u_j^\top X_{-i}u_j}=\operatorname{tr}(X^{-1})+\frac{u_i^\top X^{-2}u_i}{1-u_i^\top X^{-1}u_i}-\frac{u_j^\top X_{-i}^{-2}u_j}{1+u_j^\top X_{-i}u_j}$$

Hence, local optimality of I implies that for any $i\in I, j\in[n]$,

$$\beta\le\operatorname{tr}(X^{-1})+\frac{u_i^\top X^{-2}u_i}{1-u_i^\top X^{-1}u_i}-\frac{u_j^\top X_{-i}^{-2}u_j}{1+u_j^\top X_{-i}u_j}\tag{8}$$

By Lemma 20, we have $u_j^\top X_{-i}^{-1}u_j=\frac{\tau_j+\tau_{ij}^2-\tau_i\tau_j}{1-\tau_i}$ and $u_j^\top X_{-i}^{-2}u_j=h_j+\frac{h_i\tau_{ij}^2}{(1-\tau_i)^2}+\frac{2\tau_{ij}h_{ij}}{1-\tau_i}$. Substituting these and $\operatorname{tr}(X^{-1})=\beta$, $u_j^\top X^{-2}u_j=h_j$, and $u_j^\top X^{-1}u_j=\tau_j$ in equation (8), we get

$$\begin{aligned}
\beta&\le\beta+\frac{h_i}{1-\tau_i}-\frac{h_j+\frac{h_i\tau_{ij}^2}{(1-\tau_i)^2}+\frac{2\tau_{ij}h_{ij}}{1-\tau_i}}{1+\frac{\tau_j+\tau_{ij}^2-\tau_i\tau_j}{1-\tau_i}}\\
0&\le\frac{h_i}{1-\tau_i}-\frac{h_j(1-\tau_i)^2+h_i\tau_{ij}^2+2(1-\tau_i)\tau_{ij}h_{ij}}{(1-\tau_i)(1-\tau_i+\tau_j+\tau_{ij}^2-\tau_i\tau_j)}\\
0&\le\frac{h_i}{1-\tau_i}-\frac{h_i\tau_{ij}^2}{(1-\tau_i)(1-\tau_i+\tau_j+\tau_{ij}^2-\tau_i\tau_j)}-\frac{h_j(1-\tau_i)^2+2(1-\tau_i)\tau_{ij}h_{ij}}{(1-\tau_i)(1-\tau_i+\tau_j+\tau_{ij}^2-\tau_i\tau_j)}\\
0&\le\frac{h_i(1-\tau_i+\tau_j+\tau_{ij}^2-\tau_i\tau_j-\tau_{ij}^2)}{(1-\tau_i)(1-\tau_i+\tau_j+\tau_{ij}^2-\tau_i\tau_j)}-\frac{h_j(1-\tau_i)+2\tau_{ij}h_{ij}}{1-\tau_i+\tau_j+\tau_{ij}^2-\tau_i\tau_j}\\
0&\le\frac{h_i(1+\tau_j)}{1-\tau_i+\tau_j+\tau_{ij}^2-\tau_i\tau_j}-\frac{h_j(1-\tau_i)+2\tau_{ij}h_{ij}}{1-\tau_i+\tau_j+\tau_{ij}^2-\tau_i\tau_j}\\
0&\le h_i(1+\tau_j)-h_j(1-\tau_i)-2\tau_{ij}h_{ij}
\end{aligned}$$

Last inequality follows from the fact that $1-\tau_i+\tau_j-\tau_i\tau_j+\tau_{ij}^2=(1-\tau_i)(1+\tau_j)+\tau_{ij}^2>0$ which follows from the fact that $\tau_i<1$ (Lemma 20 and X_{-i} is invertible). This concludes the proof of claim 4. ∎

Next, we sum up equation (7) from claim 4 for all $i\in Z$ and get

$$(1+\tau_j)\sum_{i\in I}h_i-h_j(|I|-\sum_{i\in I}\tau_i)-2\sum_{i\in I}\tau_{ij}h_{ij}\ge0$$

By Lemma 20, $\sum_{i\in I}h_i=\beta, \sum_{i\in I}\tau_i=d$, and $\sum_{i\in I}\tau_{ij}h_{ij}=h_j$. We also know that $|I|=k$ throughout the algorithm. Substituting these in the equation above we get, $(1+\tau_j)\beta-h_j(k-d)-2h_j\ge0$ or equivalently,

$$\frac{h_j}{1+\tau_j}\le\frac{\beta}{k-d+2}.$$

This finishes the proof of Lemma 22. ∎

B.1.2. THE CAPPING ALGORITHM AND THE PROOF OF LEMMA 11

Some intuition of the capping algorithm. Section 3.3 shows an example where local search outputs a solution with very large cost, thus showing that local search does not provide any approximation algorithm. The failure of local search algorithm is the presence of extremely long vectors ($||v||_2^2$ much larger than A-optimum) which leads to "skewed" eigenvectors and eigenvalues. Moreover, we were able to show that this is the only bottleneck. That is, if all vector norms are small (compared to A-optimum), solution output by the local search algorithm has cost at most $(1+\epsilon)$ times the fractional optimum.

The capping algorithm should then satisfy the following(s): Given an instance with arbitrary length vectors, output a new instance such that

1. All vectors in the new instance have small length
2. Fractional optimum of the new instance does not increase by more than $1+\epsilon$ factor of the old fractional optimum
3. Any integral solution in the new instance can be translated into an integral solution in the old instance with the same or lower cost.

If we can get such a procedure, we run the local search on the new instance and get an integral solution with cost at most $(1+\epsilon)$ times the fractional optimum of the new solution. Combining with the properties above, we can then get an integral solution in the old instance with cost at most $(1+\epsilon)^2$ of the old fractional optimum.

We note that a more natural capping algorithm where we pick the longest vector, scale this vector down, and project all other vectors into the space orthogonal to the large vector satisfies properties (1) and (2) but not (3). That is, given an integral solution in the new instance, we can not always find an integral solution in the old instance with roughly the same cost.

We now proof of Lemma 11, which says that our capping algorithm satisfies three properties we want.

Proof [Lemma 11] For ease of notation, we consider the equivalent algorithm of Algorithm 2.

Algorithm 5 Capping vectors length for A-DESIGN

Input: $V = \{v_1, \ldots, v_n\} \subseteq \mathbb{R}^d$, parameter Δ.
For $i \in [1,n], w_i^0 := v_i, \ell = 0$.
while $\exists i \in [1,n], ||w_i^l||_2^2 > \Delta$ **do**
  $t_\ell = \text{argmax}_{i\in[1,n]} ||w_i^l||_2$.
  % For all vectors, scale the component along with w_t direction.
  For $j \in [1,n], w_j^{\ell+1} = \left(I_d - \frac{1}{2}\frac{w_{t_\ell}^\ell (w_{t_\ell}^\ell)^\top}{||w_{t_\ell}^\ell||_2^2} \right) w_j^\ell$
  $\ell = \ell + 1$.
end while
For $j \in [1,n], u_j = w_j^\ell$.
Return $U = \{u_1, \ldots, u_n\} \subseteq \mathbb{R}^d$

First observe that the length of the largest vector reduces by a constant factor and length of any vector does not increase. Thus the algorithm ends in a finite number of iterations. Observe that the first property is trivially true when the algorithm returns a solution. For the second property, we show that the objective value of any set S only increases over the iterations. In particular, we show the following claim.

Claim 5 *For any set $S \subset [n]$ and any $\ell \geq 0$,*

$$\mathrm{tr}\left(\left(\sum_{i\in S} w_i^\ell (w_i^\ell)^\top\right)^{-1}\right) \leq \mathrm{tr}\left(\left(\sum_{i\in S} w_i^{\ell+1} (w_i^{\ell+1})^\top\right)^{-1}\right)$$

Proof Let $Z = \left(I_{d\times d} - \frac{1}{2}\frac{w_{t_\ell}^\ell (w_{t_\ell}^\ell)^\top}{||w_{t_\ell}^\ell||_2^2}\right)$

$$\begin{aligned}
\mathrm{tr}\left(\left(\sum_{i\in S} w_i^{\ell+1} (w_i^{\ell+1})^\top\right)^{-1}\right) &= \mathrm{tr}\left(\left(Z\sum_{i\in S} w_i^\ell (w_i^\ell)^\top Z^\top\right)^{-1}\right) \\
&= \mathrm{tr}\left(Z^{-1}\left(\sum_{i\in S} w_i^\ell (w_i^\ell)^\top\right)^{-1} Z^{-1}\right) \\
&= \left\langle Z^{-2}, \left(\sum_{i\in S} w_i^\ell (w_i^\ell)^\top\right)^{-1}\right\rangle
\end{aligned}$$

Observe that Z has all eigenvalues 1 except for one which is $\frac{1}{2}$. Thus Z^{-1} and Z^{-2} have all eigenvalues at least one and in particular $Z^{-2} \succeq I$. Hence,

$$\mathrm{tr}\left(\left(\sum_{i\in S} w_i^{\ell+1} (w_i^{\ell+1})^\top\right)^{-1}\right) \geq \mathrm{tr}\left(\left(\sum_{i\in S} w_i^\ell (w_i^\ell)^\top\right)^{-1}\right)$$

as required. ■

To prove the last property, we aim to obtain a recursion on the objective value of the convex program over the iterations. Let $W^\ell = \{w_1^\ell, \ldots, w_n^\ell\}$ be the set of vectors at the end of ℓ^{th} iteration and let $\alpha_\ell^\star = \phi_f^{\mathrm{A}}(W^\ell)$ denote the objective value of the convex program with the vectors obtained at the end of ℓ^{th} iteration. We divide the iterations in to epochs where in each epoch the length of the maximum vector drops by a factor of 2. For ease of notation, we let $p = 0$ be the last epoch and $p = 1$ to be the second last epoch and so on. For any integer $p \geq 0$, we let $r_p := \mathrm{argmin}_\ell \max_{i\in[n]} \|w_i^\ell\|_2^2 \leq 2^p \cdot \Delta$ be the last iteration of p^{th} epoch. Thus in the p^{th} epoch the length of the largest vector is in the interval $[2^p \cdot \Delta, 2^{p+1} \cdot \Delta)$. Let T denote the first epoch and thus $r_T = 0$. Next lemma bounds the increase in the relaxation value in each iteration. The bound depends on which epoch does the iteration lies in.

Lemma 23 *For every $\ell \in [r_p, r_{p-1})$, we have*

$$\alpha_{\ell+1}^\star \leq \left(1 + \frac{2^{-3p/4}}{k}\right)\left(\alpha_l^\star + \frac{8}{2^{p/4}\Delta}\right).$$

Next lemma bounds the number of iterations in the p^{th} epoch.

Lemma 24 *For every $p \geq 1$, we have $r_{p-1} - r_p + 1 \leq \frac{8}{3}d$.*

We first see the proof of last claim of Lemma 11 using Lemma 23 and Lemma 24 and then prove these lemmas.

Using Lemmas 23 and 24, we bound the increase in relaxation value in each epoch.

Claim 6 *For every $p \geq 1$, we have*

$$\alpha^\star_{r_{p-1}} \leq \left(1 + \frac{2^{-3p/4}}{k}\right)^{\frac{8}{3}d} \left(\alpha^\star_{r_p} + \frac{64d}{3 \cdot 2^{p/4}\Delta}\right).$$

Proof From Lemma 23, we have

$$\begin{aligned}
\alpha^\star_{r_{p-1}} &\leq \left(1 + \frac{2^{-3p/4}}{k}\right)^{r_{p-1}-r_p+1} \alpha^\star_{r_p} + \frac{8}{2^{p/4}\Delta}\left(\sum_{i=1}^{r_{p-1}-r_p+1}\left(1 + \frac{2^{-3p/4}}{k}\right)^i\right) \\
&\leq \left(1 + \frac{2^{-3p/4}}{k}\right)^{r_{p-1}-r_p+1} \left(\alpha^\star_{r_p} + \frac{8}{2^{p/4}\Delta}(r_{p-1} - r_p + 1)\right) \\
&\leq \left(1 + \frac{2^{-3p/4}}{k}\right)^{r_{p-1}-r_p+1} \left(\alpha^\star_{r_p} + \frac{8}{2^{p/4}\Delta}(r_{p-1} - r_p + 1)\right) \\
&\leq \left(1 + \frac{2^{-3p/4}}{k}\right)^{\frac{8}{3}d} \left(\alpha^\star_{r_p} + \frac{64d}{3 \cdot 2^{p/4}\Delta}\right) \qquad \text{(Lemma 24)}
\end{aligned}$$

as required. ■

Solving the recurrence in Claim 6, we get a bound on the total increase in the relaxation cost throughout the algorithm.

$$\begin{aligned}
\alpha^\star_{r_0} &\leq \left(\Pi_{p=0}^{\top}\left(1 + \frac{2^{-3p/4}}{k}\right)^{\frac{8}{3}d}\right)\left(\alpha^\star_{r_T} + \sum_{p=0}^{T}\frac{64d}{3 \cdot 2^{p/4}\Delta}\right) \\
&\leq \left(\Pi_{p=0}^{\top}\left(1 + \frac{2^{-3p/4}}{k}\right)\right)^{\frac{8}{3}d}\left(\alpha^\star_{r_T} + \frac{2^{1/4}}{2^{1/4}-1}\frac{64d}{3\Delta}\right) \\
&\leq \left(\Pi_{p=0}^{\top}\left(1 + \frac{2^{-p/2}}{k}\right)\right)^{\frac{8}{3}d}\left(\alpha^\star_{r_T} + \frac{135d}{\Delta}\right) \qquad (9)
\end{aligned}$$

Claim 7 *For any $k \geq 15$,*

$$\Pi_{p=0}^{\infty}\left(1 + \frac{2^{-3p/4}}{k}\right) \leq 1 + \frac{3}{k}.$$

Proof

$$\Pi_{p=0}^{\infty}\left(1+\frac{2^{-3p/4}}{k}\right) = 1+\frac{1}{k}\sum_{p=0}^{\infty}2^{-3p/4}+\frac{1}{k^2}\sum_{p_1=0}^{\infty}\sum_{p_2=0}^{\infty}2^{-3p_1/4}2^{-3p_2/4}$$
$$+\frac{1}{k^3}\sum_{p_1=0}^{\infty}\sum_{p_2=0}^{\infty}\sum_{p_3=0}^{\infty}2^{-3p_1/4-3p_2/4-3p_3/4}\dots$$
$$=1+\frac{\sum_{p=0}^{\infty}2^{-3p/4}}{k}+\left(\frac{\sum_{p=0}^{\infty}2^{-3p/4}}{k}\right)^2+\left(\frac{\sum_{p=0}^{\infty}2^{-3p/4}}{k}\right)^3+\dots$$
$$\le 1+\frac{2.47}{k}+\left(\frac{2.47}{k}\right)^2+\left(\frac{2.47}{k}\right)^3+\dots$$
$$=\frac{1}{1-2.47/k}$$
$$\le 1+\frac{3}{k}$$

Last inequality follows since $k \ge 15$. ■

Substituting bound from claim 7 in Equation (9), we get

$$\alpha^{\star}_{r_0} \le \left(1+\frac{3}{k}\right)^{\frac{8}{3}d}\left(\alpha^{\star}_{r_T}+\frac{135d}{k}\right) \le \left(1+e^8\frac{d}{k}\right)\left(\alpha^{\star}_{r_T}+\frac{135d}{k}\right)$$

Last inequality follows from the fact that $(1+a/x)^y \le 1+e^{a}\frac{x}{y}$ if $x>y>0$ and $a \ge 1$.

By definition, $r_T = 0$. Hence, $\alpha^{\star}_0 = \alpha^{\star}_{r_T} = \phi_f^{\mathrm{A}}(V)$. Also, by definition $\alpha^{\star}_{r_0} = \phi_f^{\mathrm{A}}(U)$. Hence,

$$\phi_f^{\mathrm{A}}(U) \le \left(1+e^8\frac{d}{k}\right)\left(\phi_f^{\mathrm{A}}(V)+\frac{135d}{\Delta}\right) \le \left(1+3000\frac{d}{k}\right)\left(\phi_f^{\mathrm{A}}(V)+135\frac{d}{\Delta}\right).$$

This finishes the proof of Lemma 11. ■

To complete the missing details in the proof of Lemma 11, we now prove Lemmas 23 and 24.

Proof [Lemma 23] For simplicity of exposition, we make some simplifying assumptions. Without loss of generality, we assume that $t_\ell = 1$, i.e., the longest vector is the first vector in this iteration. Also, since trace is invariant under rotation of basis, we may assume that $w_1^{\ell} = \sqrt{\gamma}e_1$ for some non-negative number γ where $e_1 = \begin{pmatrix}1 & 0 & \dots & 0\end{pmatrix}^{\top}$ is the first standard vector. Hence,

$$w_j^{\ell+1} = \left(I_{d\times d} - \frac{1}{2}e_1e_1^{\top}\right)w_j^{\ell}.$$

Since, w_1^{ℓ} is the largest vector in this iteration and $\ell \in [r_p, r_{p-1})$, we have

$$2^p\Delta \ge \gamma > 2^{p-1}\Delta. \tag{10}$$

Let $\mathbf{x}$ be the optimal solution for A-REL$(w_1^{\ell},\dots,w_n^{\ell})$. We construct a feasible solution $\mathbf{y}$ for A-REL$(w_1^{\ell+1},\dots,w_n^{\ell+1})$ with objective at most as required in the lemma. Let $\delta \ge 0$ be a constant that will be fixed later. Let

$$y_i = \begin{cases}\frac{k}{k+\delta}(\delta+x_1) & i=1\\ \frac{k}{k+\delta}x_i & i\in[2,n]\end{cases}$$

Claim 8 *$\mathbf{y}$ is a feasible solution to A-REL$(w_1^{\ell+1},\ldots,w_n^{\ell+1})$.*

Proof Since, $\mathbf{x}$ is a feasible solution of A-REL$(w_1^{\ell},\ldots,w_n^{\ell})$, we know that $\sum_{i=1}^n x_i \le k$. Thus

$$\sum_{i=1}^n y_i = \frac{k}{k+\delta}\delta + \frac{k}{k+\delta}\sum_{i=1}^n x_i \le \frac{k}{k+\delta}\delta + \frac{k}{k+\delta}k \le k.$$

Clearly $\mathbf{y} \ge 0$ and thus it is feasible. ■

Now we bound the objective value of the solution $\mathbf{y}$. Let

$$X = \sum_{i=1}^n x_i w_i^{\ell}(w_i^{\ell})^\top, Y = \sum_{i=1}^n y_i w_i^{\ell+1}(w_i^{\ell+1})^\top.$$

Claim 9 *For any $\delta > 0$, $\mathrm{tr}(Y^{-1}) \le \frac{k+\delta}{k}\left(\mathrm{tr}(X^{-1}) + \frac{4}{\delta\gamma}\right)$.*

Before we prove Claim 9, we complete the proof of Lemma 6.

From Equation (10), we have $\gamma \ge 2^{p-1}\Delta$ and substituting $\delta = 2^{-p/2}$ in Claim 9 we get,

$$\mathrm{tr}(Y^{-1}) \le \left(1 + \frac{2^{-p/2}}{k}\right)\left(\mathrm{tr}(X^{-1}) + \frac{8}{2^{p/2}\Delta}\right).$$

Since, $\mathbf{x}$ is an optimal solution to A-REL$(w_1^{\ell},\ldots,w_n^{\ell})$, we have $\alpha_\ell^\star = \phi_f^{\mathrm{A}}(w_1^{\ell},\ldots,w_n^{\ell}) = \mathrm{tr}(X^{-1})$. Moreover, since y is a feasible solution to A-REL$(w_1^{\ell+1},\ldots,w_n^{\ell+1})$, we have

$$\alpha_{\ell+1}^\star = \phi_f^{\mathrm{A}}(w_1^{\ell+1},\ldots,w_n^{\ell+1}) \le \mathrm{tr}(Y^{-1}) \le \left(1 + \frac{2^{-p/2}}{k}\right)\left(\alpha_\ell^\star + \frac{8}{2^{p/2}\Delta}\right).$$

Hence, it only remains to show the proof of Claim 9.

Proof [Claim 9] Let $X = \sum_{i=1}^n x_i w_i^{\ell}(w_i^{\ell})^\top = \begin{bmatrix} p & \bar{q}^\top \\ \bar{q} & R \end{bmatrix}$ where $p \in \mathbb{R}, \bar{q} \in \mathbb{R}^d, R \in \mathbb{R}^{d-1\times d-1}$. Then

$$\begin{aligned}
\frac{k+\delta}{k}Y &= \delta w_1^{\ell+1}(w_1^{\ell+1})^\top + \sum_{i=1}^n x_i w_i^{\ell+1}(w_i^{\ell+1})^\top \\
&= \left(I_{d\times d} - \frac{1}{2}e_1e_1^\top\right)\left(\delta w_1^{\ell}(w_1^{\ell})^\top + \sum_{i=1}^n w_i^{\ell}(w_i^{\ell})^\top\right)\left(I_{d\times d} - \frac{1}{2}e_1e_1^\top\right)^\top \\
&= \begin{bmatrix} \frac{1}{2} & \bar{0}^\top \\ 0 & I_{(d-1)\times(d-1)} \end{bmatrix}\begin{bmatrix} p+\delta\gamma & \bar{q}^\top \\ \bar{q} & R \end{bmatrix}\begin{bmatrix} \frac{1}{2} & \bar{0}^\top \\ 0 & I_{(d-1)\times(d-1)} \end{bmatrix} \\
&= \begin{bmatrix} \frac{1}{4}(p+\delta\gamma) & \frac{1}{2}\bar{q}^\top \\ \frac{1}{2}\bar{q} & R \end{bmatrix}
\end{aligned}$$

Since X is positive definite, we must have $p > 0$, R is also positive definite and more over $p - \bar{q}^\top R^{-1}\bar{q} > 0$ (see Proposition 2.8.4 Bernstein (2005)).

Fact 1 *(Block Inversion formula) For* $A \in \mathbb{R}^{a\times a}, D \in \mathbb{R}^{d\times d}, B \in \mathbb{R}^{a\times d}, C \in \mathbb{R}^{d\times a}$ *such that* $\begin{bmatrix} A & B \\ C & D \end{bmatrix}$ *is invertible, we have*

$$\begin{bmatrix} A & B \\ C & D \end{bmatrix}^{-1} = \begin{bmatrix} (A-BD^{-1}C)^{-1} & -(A-BD^{-1}C)^{-1}BD^{-1} \\ -(D-CA^{-1}B)^{-1}CA^{-1} & (D-CA^{-1}B)^{-1} \end{bmatrix}$$

Applying block inversion formula on X, we get

$$X^{-1} = \begin{bmatrix} \frac{1}{p-\bar{q}^\top R^{-1}\bar{q}} & \cdots \\ \cdots & \left(R - \frac{1}{p}\bar{q}\bar{q}^\top\right)^{-1} \end{bmatrix}$$

Since, X is a positive semi-definite matrix, X^{-1} is also a positive semi-definite matrix. Hence, principle submatrices are positive semidefinite. In particular,

$$p - \bar{q}^\top R^{-1}\bar{q} \geq 0. \tag{11}$$

and,

$$R - \frac{1}{p}\bar{q}\bar{q}^\top \succeq 0_{(d-1)\times(d-1)} \tag{12}$$

Next, let us compute $\operatorname{tr}(X^{-1})$.

$$\operatorname{tr}(X^{-1}) = \frac{1}{p - \bar{q}^\top R^{-1}\bar{q}} + \operatorname{tr}\left(\left(R - \frac{1}{p}\bar{q}\bar{q}^\top\right)^{-1}\right) \geq \operatorname{tr}\left(\left(R - \frac{1}{p}\bar{q}\bar{q}^\top\right)^{-1}\right). \tag{13}$$

Applying block-inversion formula to $\frac{k+\delta}{k}Y$, we get

$$\left(\frac{k+\delta}{k}Y\right)^{-1} = \begin{bmatrix} \left(\frac{1}{4}(p+\delta\gamma) - \frac{1}{4}\bar{q}^\top R^{-1}\bar{q}\right)^{-1} & \cdots \\ \cdots & \left(R - \frac{1}{(p+\delta\gamma)/4}\frac{1}{4}\bar{q}\bar{q}^\top\right)^{-1} \end{bmatrix}$$

Hence,

$$\frac{k}{k+\delta}\operatorname{tr}\left(Y^{-1}\right) = \frac{4}{\delta\gamma + p - \bar{q}^\top R^{-1}\bar{q}} + \operatorname{tr}\left(\left(R - \frac{1}{p+\delta\gamma}\bar{q}\bar{q}^\top\right)^{-1}\right)$$

Claim 10

$$\frac{4}{\delta\gamma + p - \bar{q}^\top R^{-1}\bar{q}} \leq \frac{4}{\delta\gamma}$$

Proof By Equation (11), $p - \bar{q}^\top R^{-1}\bar{q} \geq 0$. Hence, the inequality trivially follows. ∎

Claim 11

$$\operatorname{tr}\left(\left(R - \frac{1}{p+\delta\gamma}\bar{q}\bar{q}^\top\right)^{-1}\right) \leq \operatorname{tr}\left(\left(R - \frac{1}{p}\bar{q}\bar{q}^\top\right)^{-1}\right)$$

Proof Since, $\delta, \alpha \geq 0$, $\frac{1}{p+\delta\gamma} \leq \frac{1}{p}$. Hence,

$$\begin{aligned}
\frac{1}{p+\delta\gamma}\bar{q}\bar{q}^\top &\preceq \frac{1}{p}\bar{q}\bar{q}^\top \\
-\frac{1}{p+\delta\gamma}\bar{q}\bar{q}^\top &\succeq -\frac{1}{p}\bar{q}\bar{q}^\top \\
R-\frac{1}{p+\delta\gamma}\bar{q}\bar{q}^\top &\succeq R-\frac{1}{p}\bar{q}\bar{q}^\top \\
\left(R-\frac{1}{p+\delta\gamma}\bar{q}\bar{q}^\top\right)^{-1} &\preceq \left(R-\frac{1}{p}\bar{q}\bar{q}^\top\right)^{-1} \\
\operatorname{tr}\left(\left(R-\frac{1}{p+\delta\gamma}\bar{q}\bar{q}^\top\right)^{-1}\right) &\leq \operatorname{tr}\left(\left(R-\frac{1}{p}\bar{q}\bar{q}^\top\right)^{-1}\right)
\end{aligned}$$

■

Applying the above two claims, we get

$$\begin{aligned}
\frac{k}{k+\delta}\operatorname{tr}\left(Y^{-1}\right) &\leq \frac{10^4}{\delta\gamma} + \operatorname{tr}\left(\left(R-\frac{1}{p}\bar{q}\bar{q}^\top\right)^{-1}\right) \\
\frac{k}{k+\delta}\operatorname{tr}\left(Y^{-1}\right) &\leq \frac{10^4}{\delta\gamma} + \operatorname{tr}(X^{-1}) \qquad (eq\ (13)) \\
\operatorname{tr}(Y^{-1}) &\leq \frac{k+\delta}{k}\left(\operatorname{tr}(X^{-1}) + \frac{10^4}{\delta\gamma}\right).
\end{aligned}$$

This finishes the proof of Claim 9. ■

Proof of Claim 9 also finishes the proof of Lemma 23.

■

Proof (Lemma 24) By definition of r_p and r_{p-1}, we know that for any $\ell \in [r_p, r_{p-1})$,

$$2^{p-1}\Delta \leq \max_{i\in[n]} ||w_i^\ell||_2^2 \leq 2^p\Delta$$

Let $M_{r_p} = I_{d\times d}, R_{r_p} = I_{d\times d}$ and for $\ell \in [r_p, r_{p-1})$, let

$$M_{\ell+1} = \left(I_{d\times d} - \frac{1}{2}\frac{w_{t_\ell}^\ell (w_{t_\ell}^\ell)^\top}{||w_{t_\ell}^\ell||_2^2}\right)M_\ell, \qquad R_{\ell+1} = M_{\ell+1}^T M_{\ell+1}.$$

For $\ell \in [r_p, r_{p-1})$, consider the potential function $\operatorname{tr}(R_\ell)$. We show the following properties about this potential function:

Claim 12 *Let M_ℓ, R_ℓ be as defined above for $\ell \in [r_p, r_{p-1})$. Then, $\operatorname{tr}(R_{r_p}) = d$ and for $\ell \in [r_p, r_{p-1})$,*

- $\operatorname{tr}(R_\ell) \geq 0$, *and*

- $\mathrm{tr}(R_{\ell+1}) \le \mathrm{tr}(R_\ell) - \frac{3}{8}$.

Using Claim 12, it is easy to see that $r_{p-1} - r_p + 1 \le \frac{8}{3}d$. Hence, to prove Lemma 24, it is enough to prove Claim 12.

Proof (Claim 12) Since, $R_{r_p} = I_{d\times d}$, $\mathrm{tr}(R_{r_p}) = d$ is trivially true. Also, for any $\ell \in [r_p, r_{p-1})$, $R_\ell = M_\ell^\top M_\ell$ which is positive semidefinite. Hence, $\mathrm{tr}(R_\ell) \ge 0$ for any $\ell \in [r_p, r_{p-1})$. For $\ell \in [r_p, r_{p-1})$,

$$R_{\ell+1} = M_{\ell+1}^\top M_{\ell+1} = M_\ell^\top \left(I_{d\times d} - \frac{1}{2}\frac{w_{t_\ell}^\ell (w_{t_\ell}^\ell)^\top}{||w_{t_\ell}^\ell||_2^2}\right)^\top \left(I_{d\times d} - \frac{1}{2}\frac{w_{t_\ell}^\ell (w_{t_\ell}^\ell)^\top}{||w_{t_\ell}^\ell||_2^2}\right) M_\ell$$

Matrix $\left(I_{d\times d} - \frac{1}{2}\frac{w_{t_\ell}^\ell (w_{t_\ell}^\ell)^\top}{||w_{t_\ell}^\ell||_2^2}\right)$ is symmetric. Hence,

$$\begin{aligned}
R_{\ell+1} &= M_\ell^\top \left(I_{d\times d} - \frac{w_{t_\ell}^\ell (w_{t_\ell}^\ell)^\top}{||w_{t_\ell}^\ell||_2^2} + \frac{1}{4}\frac{w_{t_\ell}^\ell (w_{t_\ell}^\ell)^\top}{||w_{t_\ell}^\ell||_2^2}\frac{w_{t_\ell}^\ell (w_{t_\ell}^\ell)^\top}{||w_{t_\ell}^\ell||_2^2}\right) M_\ell \\
&= M_\ell^\top \left(I_{d\times d} - \frac{w_{t_\ell}^\ell (w_{t_\ell}^\ell)^\top}{||w_{t_\ell}^\ell||_2^2} + \frac{1}{4}\frac{w_{t_\ell}^\ell (w_{t_\ell}^\ell)^\top}{||w_{t_\ell}^\ell||_2^2}\right) M_\ell \\
&= M_\ell^\top M_\ell - \frac{3}{4}\frac{(M_\ell^\top w_{t_\ell}^\ell)(w_{t_\ell}^\ell)^\top M_\ell}{||w_{t_\ell}^\ell||_2^2} = R_\ell - \frac{3}{4}\frac{(M_\ell^\top w_{t_\ell}^\ell)(M_\ell^\top w_{t_\ell}^\ell)^\top}{||w_{t_\ell}^\ell||_2^2}
\end{aligned}$$

By definition $w_{t_\ell}^\ell = M_\ell w_{t_\ell}^{r_p}$. Hence,

$$R_{\ell+1} = R_\ell - \frac{3}{4}\frac{(M_\ell^\top M_\ell w_{t_\ell}^{r_p})(M_\ell^\top M_\ell w_{t_\ell}^{r_p})^\top}{||w_{t_\ell}^\ell||_2^2} = R_\ell - \frac{3}{4}\frac{(R_\ell w_{t_\ell}^{r_p})(R_\ell w_{t_\ell}^{r_p})^\top}{||w_{t_\ell}^\ell||_2^2}$$

And the trace is

$$\mathrm{tr}(R_{\ell+1}) = \mathrm{tr}\left(R_\ell - \frac{3}{4}\frac{(R_\ell w_{t_\ell}^{r_p})(R_\ell w_{t_\ell}^{r_p})^\top}{||w_{t_\ell}^\ell||_2^2}\right) = \mathrm{tr}(R_\ell) - \frac{3}{4}\frac{||R_\ell w_{t_\ell}^{r_p}||_2^2}{||w_{t_\ell}^\ell||_2^2}$$

By Cauchy-Shwarz inequality, $||u||_2^2 \ge (v^T u)^2/||v||_2^2$. Substituting $u = R_\ell w_{t_\ell}^{r_p}$ and $v = w_{t_\ell}^{r_p}$, we get

$$\begin{aligned}
\mathrm{tr}(R_{\ell+1}) &\le \mathrm{tr}(R_\ell) - \frac{3}{4}\frac{\left((w_{t_\ell}^{r_p})^\top R_\ell w_{t_\ell}^{r_p}\right)^2}{||w_{t_\ell}^{r_p}||_2^2 \cdot ||w_{t_\ell}^\ell||_2^2} = \mathrm{tr}(R_\ell) - \frac{3}{4}\frac{\left((w_{t_\ell}^{r_p})^\top M_\ell^\top M_\ell w_{t_\ell}^{r_p}\right)^2}{||w_{t_\ell}^{r_p}||_2^2 \cdot ||w_{t_\ell}^\ell||_2^2} \\
&= \mathrm{tr}(R_\ell) - \frac{3}{4}\frac{||M_\ell w_{t_\ell}^{r_p}||_2^4}{||w_{t_\ell}^{r_p}||_2^2 ||w_{t_\ell}^\ell||_2^2} = \mathrm{tr}(R_\ell) - \frac{3}{4}\frac{||w_{t_\ell}^\ell||_2^4}{||w_{t_\ell}^{r_p}||_2^2 ||w_{t_\ell}^\ell||_2^2} \\
&= \mathrm{tr}(R_\ell) - \frac{3}{4}\frac{||w_{t_\ell}^\ell||_2^2}{||w_{t_\ell}^{r_p}||_2^2}
\end{aligned}$$

Since, $\ell \in [r_p, r_{p-1})$, $||w_{t_\ell}^\ell||_2^2 = \max_{i\in[n]} ||w_i^\ell||_2^2 \ge 2^{p-1}\Delta$. Also, by definition of r_p, $||w_{t_\ell}^{r_p}||_2^2 \le \max_{i\in[n]} ||w_i^{r_p}||_2^2 \le 2^p\Delta$. Hence,

$$\mathrm{tr}(R_{\ell+1}) \le \mathrm{tr}(R_\ell) - \frac{3}{4}\frac{2^{p-1}\Delta}{2^p\Delta} = \mathrm{tr}(R_\ell) - \frac{3}{8}.$$

as desired. ■

Hence, the proof of Lemma 24 is completed. ■

B.1.3. PROOF OF LEMMA 20

Proof [Lemma 20] Proof of first and second statement is same as that in Lemma 8. So, we start by proving that $h_i(1+\tau_j) - 2\tau_{ij}h_{ij} \geq 0$.

Claim 13 *For any $j \in [n]$, $X^{-1/2}u_j u_j^\top X^{-1/2} \preceq \tau_j I_d$.*

Proof Since, X is a symmetric matrix, X^{-1} and $X^{-1/2}$ are also symmetric matrices. Hence, if $q = X^{-1/2}u_j$, then $X^{-1/2}u_j u_j^\top X^{-1/2} = qq^\top$. Such a matrix has one non-zero eigenvalue equal to $||q||_2^2 = u_j^\top X^{-1} u_j = \tau_j$. Hence, $X^{-1/2}u_j u_j^\top X^{-1/2} \preceq \tau_j I_d$. ■

Next, we use this to derive further inequalities.

$$\begin{aligned}
X^{-1/2}u_j u_j^\top X^{-1/2} &\preceq \tau_j I_d \\
2X^{-1/2}u_j u_j^\top X^{-1/2} &\preceq 2\tau_j I_d \\
2X^{-1/2}u_j u_j^\top X^{-1/2} &\preceq (1+\tau_j) I_d && (\tau_j \leq 1, j \in [n]) \\
X^{-1/2}X^{-1/2}u_j u_j^\top X^{-1/2}X^{-3/2} &\preceq X^{-1/2}(1+\tau_j)X^{-3/2} && (X^{-1/2}, X^{-3/2} \text{ are PSD}) \\
2X^{-1}u_j u_j^\top X^{-2} &\preceq (1+\tau_j)X^{-2}
\end{aligned}$$

If $A \preceq B$, then $v^\top A v \leq v^\top B v$ for all v. Hence, $u_i^\top(2X^{-1}u_j u_j^\top X^{-2} \leq (1+\tau_j)X^{-2})u_i \leq 0$. Or in other words, $h_i(1+\tau_i) - 2\tau_{ij}h_{ij} \geq 0$.

Next, we show that $\sum_{i \in I} \tau_{ij}^2 = h_j$.

$$\begin{aligned}
\sum_{i \in I} \tau_{ij}^2 &= \sum_{i \in I} u_i^\top X^{-1} u_j u_i^\top X^{-1} u_j = \sum_{i \in I} u_i^\top X^{-1} u_j u_j^\top X^{-1} u_i \\
&= \sum_{i \in u} \langle X^{-1}u_j u_j^\top X^{-1}, u_i u_i^\top \rangle \\
&= \langle X^{-1}u_j u_j^\top X^{-1}, \sum_{i \in Z} u_i u_i^\top \rangle \\
&= \langle X^{-1}u_j u_j^\top X^{-1}, X \rangle \\
&= \langle u_j^\top X^{-1}, X X^{-1} u_j^\top \rangle \\
&= \langle u_j^\top X^{-1}, u_j^\top \rangle = u_j^\top X^{-1} u_j = h_j
\end{aligned}$$

Next, we show that $\sum_{i\in I} h_i = \beta$.

$$\begin{aligned}\sum_{i\in I} h_i &= \sum_{i\in Z} u_i^\top X^{-2} u_i \\ &= \sum_{i\in I} \langle X^{-2}, u_i u_i^\top \rangle \\ &= \langle X^{-2}, \sum_{i\in I} u_i u_i^\top \rangle = \langle X^{-2}, X\rangle \\ &= \langle X^{-1}, X^{-1}X\rangle \\ &= \langle X^{-1}, I_d\rangle = \operatorname{tr}(X^{-1})\end{aligned}$$

Next, we show that $\sum_{i\in I} \tau_{ij} h_{ij} = h_j$.

$$\begin{aligned}\sum_{i\in I} \tau_{ij} h_{ij} &= \sum_{i\in I} u_i^\top X^{-1} u_j u_i^\top X^{-2} u_j = \sum_{i\in I} u_i^\top X^{-1} u_j u_j^\top X^{-2} u_i \\ &= \sum_{i\in I} \langle X^{-1} u_j u_j^\top X^{-2}, u_i u_i^\top \rangle \\ &= \langle X^{-1} u_j u_j^\top X^{-2}, \sum_{i\in Z} u_i u_i^\top \rangle = \langle X^{-1} u_j u_j^\top X^{-2}, X\rangle \\ &= \langle u_j^\top X^{-2}, u_j^\top X^{-1} X\rangle \\ &= \langle u_j^\top X^{-2}, u_j\rangle = h_j\end{aligned}$$

Next, we show that $\tau_j \le \sqrt{h_j}||u_j||_2$.

$$\begin{aligned}\sqrt{h_j}||u_j||_2 &= \sqrt{u_j^\top X^{-2} u_j}\,||u_j||_2 \\ &= \sqrt{||X^{-1}u_j||_2^2}\,||u_j||_2 = ||X^{-1}u_j||_2 ||u_j||_2 \\ &\ge u_j^\top X^{-1} u_j = \tau_j.\end{aligned}$$

Here, the last inequality follows from Cauchy-Schwarz inequality: for any $u, v \in \mathbb{R}^d$, $u^\top v \le ||u||_2||v||_2$.

Next, we show the last two equalities. For $i \in [n]$, $X_{-i} = X - u_i u_i^\top$. Let $j \in [n]$. By Sherman-Morrison formula,

$$X_{-i}^{-1} = X^{-1} + \frac{X^{-1} u_i u_i^\top X^{-1}}{1 - u_i^\top X^{-1} u_i} = X^{-1} + \frac{X^{-1} u_i u_i^\top X^{-1}}{1-\tau_i} \tag{14}$$

Hence,

$$\begin{aligned}u_j^\top X_{-i}^{-1} u_j &= u_j^\top X^{-1} u_j + \frac{u_j^\top X^{-1} u_i u_i^\top X^{-1} u_j}{1-\tau_i} \\ &= \tau_j + \frac{u_j^\top X^{-1} u_i u_i^\top X^{-1} u_j}{1-\tau_i} \\ &= \tau_j + \frac{\tau_{ij}\cdot\tau_{ij}}{1-\tau_i} = \frac{\tau_j + \tau_{ij}^2 - \tau_i\tau_j}{1-\tau_i}\end{aligned}$$

Squaring the terms in equation (14), we get

$$X_{-i}^{-2} = X^{-2} + \frac{X^{-1}u_i u_i^\top X^{-2} u_i u_i^\top X^{-1}}{(1-\tau_i)^2} + \frac{X^{-1}u_i u_i^\top X^{-2}}{1-\tau_i} + \frac{X^{-2}u_i u_i^\top X^{-1}}{1-\tau_i}$$
$$= X^{-2} + h_i \frac{X^{-1}u_i u_i^\top X^{-1}}{(1-\tau_i)^2} + \frac{X^{-1}u_i u_i^\top X^{-2}}{1-\tau_i} + \frac{X^{-2}u_i u_i^\top X^{-1}}{1-\tau_i}$$

Hence,

$$u_j^\top X_{-i}^{-2} u_j = u_j^\top X^{-2} u_j + h_i \frac{u_j^\top X^{-1}u_i u_i^\top X^{-1} u_j}{(1-\tau_i)^2} + \frac{u_j^\top X^{-1}u_i u_i^\top X^{-2} u_j}{1-\tau_i} + \frac{u_j^\top X^{-2}u_i u_i^\top X^{-1} u_j}{1-\tau_i}$$
$$= h_j + h_i \frac{\tau_{ij} \cdot \tau_{ij}}{(1-\tau_i)^2} + \frac{\tau_{ij} h_{ij}}{1-\tau_i} + \frac{h_{ij}\tau_{ij}}{1-\tau_i}$$
$$= h_j + \frac{h_i \tau_{ij}^2}{(1-\tau_i)^2} + \frac{2\tau_{ij} h_{ij}}{1-\tau_i}$$

■

B.2. Guessing A-Optimum Value $\phi^{\mathbf{A}}(V)$

We remarked earlier that Algorithm 4 requires the knowledge of the optimum solution value $\phi^{\mathrm{A}}(V)$. We can guess this value efficiently by performing a binary search. We explain the details and the proof of the polynomial runtime of the search in this section.

Let $\alpha = \operatorname{tr}\left(\left(\sum_{i=1}^n v_i v_i^T\right)^{-1}\right)$. Since we may pick at most k copies of each vector, we have that $\phi^{\mathrm{A}}(V) \geq \operatorname{tr}\left(\left(k\sum_{i=1}^n v_i v_i^T\right)^{-1}\right) = \frac{1}{k}\alpha$. The fractional solution $x_i = \frac{k}{n}$ is feasible for A-REL(V). Hence, $\phi_f^{\mathrm{A}}(V) \leq \operatorname{tr}\left(\left(\frac{k}{n}\sum_{i=1}^n v_i v_i^T\right)^{-1}\right) = \frac{n}{k}\alpha$. Using the result in Allen-Zhu et al. (2017), we get that $\phi^{\mathrm{A}}(V) \leq (1+\epsilon)\phi_f^{\mathrm{A}}(V)$. Hence, $\phi^{\mathrm{A}}(V) \in \left[\frac{1}{k}\alpha, \frac{n(1+\epsilon)}{k}\alpha)\right]$. Hence, given an instance, we first compute α and then perform a binary search for $\phi^{\mathrm{A}}(V)$ in the interval $[\frac{1}{k}\alpha, \frac{n(1+\epsilon)}{k}\alpha]$.

Suppose the current range of the optimum is $[\ell, u]$. We guess OPT to be $\frac{\ell+u}{2}$ (use this as A-optimum $\phi^{\mathrm{A}}(V)$) and run the modified local search algorithm. We claim that if it outputs a solution with cost at most $(1+\epsilon)\frac{\ell+u}{2}$ then $\phi^{\mathrm{A}}(V)$ lies in the range $[\ell, (1+\epsilon)\frac{\ell+u}{2}]$. If it outputs a solution with cost more than $(1+\epsilon)\frac{\ell+u}{2}$, then $\phi^{\mathrm{A}}(V)$ lies in the range $[\frac{\ell+u}{2}, u]$. The first statement is trivially true. The second statement is equivalent to the following: If $\phi^{\mathrm{A}}(V)$ is less than $\frac{\ell+u}{2}$, then the algorithm outputs a solution of cost at most $(1+\epsilon)\frac{\ell+u}{2}$. Proof of this fact follows exactly the same way as the proof of Theorem 13 by substituting $\phi^{\mathrm{A}}(V)$ with $\frac{\ell+u}{2}$ everywhere. The proof still follows, since the only place we use the meaning of the $\phi^{\mathrm{A}}(V)$ value is in claiming that there exists a fractional solution with value $\phi^{\mathrm{A}}(V)$. Because $\phi^{\mathrm{A}}(V)$ is less than $\frac{\ell+u}{2}$, this statement is true with $\phi^{\mathrm{A}}(V)$ replaced by $\frac{\ell+u}{2}$.

We can guess the value of $\phi^{\mathrm{A}}(V)$ upto a factor of $1+\epsilon$ in $\log_{1+\epsilon}(n(1+\epsilon)) \leq \frac{\log(n(1+\epsilon))}{\epsilon}$ iterations. This introduces an additional multiplicative factor of $1+\epsilon$ in the approximation factor in Theorem 13. Hence, we get an approximation factor of $(1+\epsilon)(1+\epsilon) \leq (1+3\epsilon)$ and polynomial number of iterations.

B.3. Example of Instances to A-DESIGN

In this section, we give more details deferred from Section 3.3, starting with the proof of Theorem 14.

Proof [Theorem 14] The case $d = 2$ is proven in Lemma 15, so let $d \geq 3$. Let

$$v_1 = [1; \frac{1}{N^2}; 0; \ldots; 0], v_2 = [1; -\frac{1}{N^2}; 0; \ldots; 0], w_1 = [N^4; N; 0; \ldots; 0],$$

$$w_2 = [N^4; -N; 0; \ldots; 0], U = \left\{ u_i := \frac{1}{N^3} e_i : i = 3 \ldots, d \right\},$$

and let $\{v_1, v_2, w_1, w_2\} \cup U$ be the input vectors to A-DESIGN. Let $p = \lfloor \frac{k-d+2}{2} \rfloor, q = \lceil \frac{k-d+2}{2} \rceil$. Consider a solution S which picks p and q copies of v_1 and v_2, and one copy of u_i for each $i = 3, \ldots, d$. We claim that S is locally optimal.

Consider a swap of elements $S' = S \setminus \{s\} \cup \{s'\}$ where $s' \neq s$. If $s \in U$, then S' does not span full dimension. Hence, $s \in \{v_1, v_2\}$. If $s' = e_i \in U$ for some i, then the increase of eigenvalue of S' in the ith axis reduces the objective by $\Theta(N^3)$. However, by Claim 2, removing a vector s will increase the objective by $\Omega(N^4)$. Finally, if $s' \notin U$, then the swap appears within the first two dimension, so the calculation that a swap increases the objective is identical to the case $d = 2$, proven in Lemma 15. Therefore, S is locally optimal.

We now observe that the objective given by S is $\Theta(N^4)$, dominated by eigenvalues of eigenvectors spanning the first two dimension. However, consider a solution S^* which picks p and q copies of w_1 and w_2, and one copy of u_i for each $i = 3, \ldots, d$. The objective of S^* contributed by eigenvalues of eigenvectors lying in the first two dimension is $O(N^2)$ (Claim 2), so the total objective of S^* is $\Theta(N^3)$, which is arbitrarily smaller than $\Theta(N^4)$, the objective of S. ■

We also remark that the exmple of input vectors to A-DESIGN given in this section also shows that A-DESIGN objective $S \to \operatorname{tr}\left(\left(\sum_{i \in S} v_i v_i^\top\right)^{-1}\right)$ is not supermodular, making the analysis of algorithms in submodular optimization unapplicable. A set function $g : 2^U \to \mathbb{R}$ is called submodular if $g(S \cup \{u\}) - g(S) \geq g(S' \cup \{u\}) - g(S')$ for all $S \subseteq S' \subseteq U$ and $u \in U$, and g is supermodular if $-g$ is submodular. In other words, g is supermodular if the marginal loss of g by adding u is decreasing as the set S is increasing by a partial ordering "$\subseteq$". As a set increases, the marginal loss of the A-DESIGN objective not only potentially increase, but also has no upper bound.

Remark 25 *For any $d \geq 2, T > 0$, there exist sets of vectors $S \subsetneq S'$ in $\mathbb{R}^d$ and a vector $w \in \mathbb{R}^d$ such that*

$$\frac{\operatorname{tr}\left(\left(\sum_{i \in S'} vv^\top\right)^{-1}\right) - \operatorname{tr}\left(\left(\sum_{i \in S'} vv^\top + ww^\top\right)^{-1}\right)}{\operatorname{tr}\left(\left(\sum_{i \in S} vv^\top\right)^{-1}\right) - \operatorname{tr}\left(\left(\sum_{i \in S} vv^\top + ww^\top\right)^{-1}\right)} > T$$

Proof We first assume $d = 2$. Use the same definitions of vectors from Lemma 15 and set $S = \{v_1, v_2\}, S' = \{v_1, v_2, w_1\}$ and $w = w_2$. By Claim 2,

$$\operatorname{tr}\left(\left(\sum_{i \in S} vv^\top\right)^{-1}\right) - \operatorname{tr}\left(\left(\sum_{i \in S} vv^\top + ww^\top\right)^{-1}\right) = O(N)$$

and

$$\operatorname{tr}\left(\left(\sum_{i\in S'} vv^\top\right)^{-1}\right) - \operatorname{tr}\left(\left(\sum_{i\in S'} vv^\top + ww^\top\right)^{-1}\right) \geq \operatorname{tr}\left(\left(\sum_{i\in S'} vv^\top\right)^{-1}\right) - \operatorname{tr}\left(\left(w_1w_1^\top + w_2w_2^\top\right)^{-1}\right) = \Theta(N^4),$$

so the proof is done because N tends to infinity. For the case $d \geq 3$, we may pad zeroes to all vectors in the above example and add a unit vector to S, S' to each of other $d-2$ dimensions. ■

Appendix C. Approximate Local Search for D-DESIGN

While Theorem 7 proves a guarantee for every local optimum, it is not clear at all whether the local optimum solution can be obtained efficiently. Here we give a approximate local search algorithm that only makes improvements when they result in substantial reduction in the objective. We show that this algorithm is polynomial time as well results in essentially the same guarantee as Theorem 7.

Algorithm 6 Approximate Local search algorithm for D-DESIGN

Input: $V = v_1, \ldots, v_n \in \mathbb{R}^d, d \leq k \in n$, parameter $\delta > 0$.
Let I be any (multi)-subset of $[1, n]$ of size k such that $X = \sum_{i\in I} v_iv_i^\top$ is non-singular matrix.
while $\exists i \in I, j \in [1, n]$ such that $det\left(X - v_iv_i^\top + v_jv_j^\top\right) > (1+\delta)\cdot\det(X)$ **do**
  $X = X - v_iv_i^\top + v_jv_j^\top$
  $I = I \setminus \{i\} \cup \{j\}$
end while
Return (I, X)

Recall that ϕ_f^{D} denote the be the common optimum value of (D-REL) and its dual (D-REL-DUAL). $I^\star$ denote the indices of the vector in the optimal solution and $\phi^{\mathrm{D}} = \det\left(\sum_{i\in I^\star} v_iv_i^\top\right)^{\frac{1}{d}}$ be its objective. We have $\phi_f^{\mathrm{D}} \geq \log \phi^{\mathrm{D}}$. We have the following result about Algorithm 6.

Theorem 26 *Let X be the solution returned by Algorithm 6. Then,*

$$det(X) \geq e^{-k\delta}\left(\frac{k-d+1}{k}\right)^d e^{d\cdot\phi_f^{\mathrm{D}}}$$

and therefore,

$$\det(X)^{\frac{1}{d}} \geq e^{-\frac{k\delta}{d}}\frac{k-d+1}{k}\cdot\phi^{\mathrm{D}}.$$

Moreover, the running time of the algorithm is polynomial in $n, d, k, \frac{1}{\delta}$ and the size of the input.

Proof of the theorem is analogous to the proof of Theorem 7. Let (I, X) be the returned solution of the algorithm. We also let V_I denote the $d \times |I|$ matrix whose columns are v_i for each $i \in I$. Observe that $X = V_I V_I^\top$ and X is invertible since $\det(X) > 0$ at the beginning of the iteration and it only increases in later iterations. We let $\tau_i = v_i^\top X^{-1} v_i$ for any $1 \le i \le n$. Observe that if $i \in I$, then τ_i is the leverage score of row v_i with respect to the matrix $V_I^\top$. We also $\tau_{ij} = v_i^\top X^{-1} v_j$ for any $1 \le i, j \le n$. As in Theorem 7, we have some properties regarding τ_i and h_i.

Lemma 27 *We have the following.*

1. *For any $i \in I$, we have $\tau_i \le 1$. Moreover, for any $i \in I$, $\tau_i = 1$ if and only if $X - v_i v_i^\top$ is singular.*
2. *We have $\sum_{i \in I} \tau_i = d$.*
3. *For any $1 \le j \le n$, we have $\sum_{i \in I} \tau_{ij} \tau_{ji} = \tau_j$.*
4. *For any $1 \le i, j \le n$, we have $\tau_{ij} = \tau_{ji}$ and $\tau_{ij} \le \sqrt{\tau_i \tau_j}$.*

Proof of the lemma is identical to that of Lemma 8. Next, we show an upper bound on τ_j for the approximate local optimal solution.

Lemma 28 *For any $j \in [1, n]$,*

$$\tau_j \le \frac{d + \delta k}{k - d + 1}.$$

Before we prove the lemma, we complete the proof of Theorem 26.

Proof [Theorem 26] We construct a feasible solution to the (D-REL-DUAL) of the objective value of at most $\frac{1}{d} \log \det(X) + \log \frac{k}{k-d+1} + \frac{k\delta}{d}$. This would imply that

$$O_f^\star \le \frac{1}{d} \log \det(X) + \log \frac{k}{k - d + 1} + \frac{k\delta}{d}$$

which proves the first part of the theorem. The second part follows since $\phi_f^{\mathrm{D}} \ge \log \phi^{\mathrm{D}}$.

Let

$$Y = \alpha X, \qquad \mu = \max_{1 \le j \le n} v_j^\top Y^{-1} v_j = \frac{1}{\alpha} \max_{j \in [1,n]} v_j^\top X^{-1} v_j$$

where $\alpha > 0$ will be fixed later. Then, (Y, μ) is a feasible solution of (D-REL-DUAL). Hence,

$$\begin{aligned} \phi_f^{\mathrm{D}} &\le \frac{1}{d} \log \det(\alpha X) + \frac{k}{d} \cdot \frac{1}{\alpha} \max_{j \in [1,n]} v_j^\top X^{-1} v_j - 1 \\ &\le \log \alpha + \frac{1}{d} \log \det(X) + \frac{k}{d\alpha} \cdot \frac{d + k\delta}{k - d + 1} - 1 \qquad \text{(Lemma 28)} \end{aligned}$$

Setting $\alpha = \frac{k}{k-d+1}$, we get

$$\phi_f^{\mathrm{D}} \le \log \frac{k}{k - d + 1} + \frac{1}{d} \log \det(X) + 1 + \frac{k\delta}{d} - 1 = \log \frac{k}{k - d + 1} + \frac{1}{d} \log \det(X) + \frac{k\delta}{d}$$

as required. ■

Proof [Lemma 28] Since X is a symmetric matrix, X^{-1} is also a symmetric matrix and therefore $\tau_{ij} = \tau_{ji}$ for each i, j. We first show that the approximate local optimality condition implies the following claim:

Claim 14 *For any $i \in I$ and $j \in [n]$, we have*

$$\tau_j - \tau_i \tau_j + \tau_{ij}\tau_{ji} \leq \delta + \tau_i. \tag{15}$$

Proof Let $i \in I$, $j \in [n]$ and $X_{-i} = X - v_i v_i^\top$. First, consider the case when X_{-i} is singular. From Lemma 8, we have that $\tau_i = 1, \tau_{ij} = \tau_{ji} \leq \sqrt{\tau_i \tau_j} \leq 1$. Hence,

$$\tau_j - \tau_i \tau_j + \tau_{ij}\tau_{ji} \leq \tau_j - \tau_j + 1 = \tau_i \leq \delta + \tau_i$$

Now consider the case when X_{-i} is non-singular. By local optimality of I, we get that

$$\det\left(X_{-i} + v_j v_j^\top\right) \leq (1+\delta)\det\left(X_{-i} + v_i v_i^\top\right) \tag{16}$$

Claim 15 *For any invertible matrix $A \in \mathbb{R}^{d\times d}$ and $v \in \mathbb{R}^d$,*

$$\det(A + vv^\top) = \det(A)(1 + v^\top A^{-1} v)$$

Hence, local optimality of I implies that for any $i \in I, j \in [n]$,

$$\det(X_{-i})(1 + v_j^\top X_{-i}^{-1} v_j) \leq (1+\delta)\det(X_{-i})(1 + v_i^\top X_{-i}^{-1} v_i)$$

Dividing both sides by $\det(X_{-i})$, we get for each $i \in I$ and $j \in [n]$, we have $1 + v_j^\top X_{-i}^{-1} v_j \leq (1+\delta)(1 + v_i^\top X_{-i}^{-1} v_i)$ or equivalently,

$$v_j^\top X_{-i}^{-1} v_j \leq \delta + (1+\delta) v_i^\top X_{-i}^{-1} v_i.$$

From the Sherman-Morrison Formula we obtain that for any $i \in I$ and $j \in [n]$, we have

$$v_j^\top \left(X^{-1} + \frac{X^{-1} v_i v_i^\top X^{-1}}{1 - v_i^\top X^{-1} v_i}\right) v_j \leq \delta + (1+\delta) v_i^\top \left(X^{-1} + \frac{X^{-1} v_i v_i^\top X^{-1}}{1 - v_i^\top X^{-1} v_i}\right) v_i.$$

Now using the definition of τ_i, τ_j and τ_{ij}, we obtain that for any $i \in I$ and $1 \leq j \leq n$, we have

$$\tau_j + \frac{\tau_{ji}\tau_{ij}}{1 - \tau_i} \leq \delta + (1+\delta)\left(\tau_i + \frac{\tau_i^2}{1-\tau_i}\right).$$

Multiplying by $1 - \tau_i$, which is positive from Lemma 8, on both sides we obtain that for any $i \in I$ and $1 \leq j \leq n$,

$$\tau_j - \tau_i \tau_j + \tau_{ij}\tau_{ji} \leq \delta(1 - \tau_i) + (1+\delta)\tau_i = \delta + \tau_i$$

thus finishing the proof of the claim. ■

Now summing over the inequality in Claim 14 for all $i \in I$, we get

$$\sum_{i\in I} (\tau_j - \tau_i\tau_j + \tau_{ij}\tau_{ji}) \leq \sum_{i \in I} \delta + \sum_{i\in I} \tau_i.$$

Applying Lemma 8, we obtain that

$$k\tau_j - d\tau_j + \tau_j \leq \delta k + d.$$

Rearranging, we obtain that

$$\tau_j \leq \frac{d + \delta k}{k - d + 1}$$

■

Runtime Analysis. One may obtain the worst-case runtime for local search for D-design as follows. Let L be the maximum number of the length of binary string that encodes the number in each component across all input vectors v_i. Suppose we start with any solution S with nonzero determinant $\det(V_S V_S^T) = \sum_{R\subseteq S, |R|=d} \det(V_R V_R^T)$ (Cauchy-Binet), which can be done in polynomial time by finding a set of linearly independent vectors. Since $V_S V_S^\top$ is PSD, $\det(V_S V_S^T)$ is nonnegative and hence must be strictly positive, and therefore at least one term $\det(V_R V_R^T)$ is strictly positive. We now use the fact that for a square matrix A, the binary encoding length of $\det(A)$ is at most twice of the encoding length of matrix A (the exact definition of encoding length and the proof are in Theorem 3.2 of Schrijver (1998)). Since the length of $d \times d$ matrix $V_R V_R^T$ is at most $d^2 + Ld^2 \leq 2Ld^2$, the length of $\det(V_R V_R^T)$ is at most $4Ld^2$. Hence, the value of the determinant is at least 2^{-4Ld^2}.

The optimum solution S^* of D-DESIGN attains objective $\sum_{R\subset S^*, |R|=d} \det(V_R V_R^T)$ (Cauchy-Binet). Each term $\det(V_R V_R^T)$ again has length at most $4Ld^2$, and so is at most 2^{4Ld^2}. Therefore, the optimum is at most $\binom{k}{d} \cdot 2^{4Ld^2} \leq k^d 2^{4Ld^2}$. Hence, any solution S with nonzero determinant is a $k^d 2^{8Ld^2}$-approximation. Each swap increases the objective by a multiplicative factor $1+\delta$, so the algorithm takes at most $\log_{1+\delta}(k^d 2^{8Ld^2}) \leq \frac{2}{\delta} d \log k \cdot (8Ld^2) = O(\frac{Ld^3 \log k}{\delta})$ swapping steps for $\delta < 1/2$. We may use matrix determinant lemma (for rank-one update) to compute the new determinant objective rather than recomputing it in the next iteration. The matrix determinant lemma computation takes $O(d^2)$ times, so one swapping steps takes $O(knd^2)$ time by computing all kn potential pairs of swaps. Therefore, the local search in total takes $O(\frac{Ld^3 \log k}{\delta} knd^2) = O(\frac{Lknd^5 \log k}{\delta})$ arithmetic operations.

Appendix D. Approximate Local Search for A-DESIGN

Algorithm 7 Approximate Local search algorithm for A-DESIGN

Input: $U = \{u_1, \ldots, u_n\} \subseteq \mathbb{R}^d, d \leq k \in \mathbb{N}$.
Let I be any (multi)-subset of $[1, n]$ of size k such that $X = \sum_{i\in I} v_i v_i^\top$ is non-singular.
while $\exists i \in I, j \in [1, n]$ such that $\operatorname{tr}\left((X - u_i u_i^\top + u_j u_j^\top)^{-1}\right) < (1-\delta)\operatorname{tr}(X^{-1})$ **do**
  $X = X - u_i u_i^\top + u_j u_j^\top$
  $I = I \setminus \{i\} \cup \{j\}$
end while
Return (I, X)

Recall that for any input vectors $V = \{v_1, \ldots, v_n\}$, the primal program is A-REL(V) and the dual program is A-REL-DUAL(V). We index these convex program by input vectors as we aim to analyze their objectives when the input changes by the capping algorithm. $\phi_f^{\mathrm{A}}(V)$ denote the (common) optimal value of objective values of the convex program with input vectors from V. $I^\star$ denote the indices of the vectors in the optimal solution of A-DESIGN with input vector set V and let $\phi^{\mathrm{A}}(V) = \operatorname{tr}\left(\left(\sum_{i\in I^\star} v_i v_i^\top\right)^{-1}\right)$ be its objective. Recall that $\phi_f^{\mathrm{A}}(V) \leq \phi^{\mathrm{A}}(V)$.

Similar to the local search result for A-DESIGN of Theorem 12, we can prove the following theorem:

Theorem 29 *Let X be the matrix returned by Algorithm 7. If $||u_i||_2^2 \leq \Delta$ for all $i \in [n]$,*

$$\operatorname{tr}(X^{-1}) \leq \phi_f^{\mathrm{A}}(U)\left(\left(1-\frac{d-2}{k}\right)\frac{1}{1+(k-d)\delta}-\sqrt{\frac{\Delta\phi_f^{\mathrm{A}}(U)}{k}}\right)^{-1}.$$

To prove Theorem 29, we can prove the following lemma instead of Lemma 22.

Lemma 30 *For any $j \in [n]$,*

$$\frac{h_j}{1+\tau_j} \leq \frac{\beta(1+(k-d)\delta)}{k-d+2}$$

Instead of Theorem 13, Theorem 29 now leads to the following theorem:

Theorem 31 *For input vectors $V = \{v_1, \ldots, v_n\}$ and parameter k, let $U = \{u_1, \ldots, u_n\}$ be the set of vectors returned by the Capping Algorithm 2 with vector set V and $\Delta = \frac{d}{\epsilon^2 \phi^{\mathrm{A}}(V)}$. Let (I, X) be the solution returned by Algorithm 3 with vector set U and parameter k. If $k \geq \frac{2d}{\epsilon^4}$, $\delta \leq \frac{\epsilon d}{2k}$, and$\epsilon \leq 0.001$ then,*

$$\operatorname{tr}\left(\left(\sum_{i\in I} v_i v_i^\top\right)^{-1}\right) \leq (1+2\epsilon)\phi^{\mathrm{A}}(V).$$

Proof of the theorems and lemmas are identical to the corresponding theorems and lemmas proved in Section 3. Hence, we avoid the tedious calculations in reproving these theorems.

Runtime Analysis We claim that the running times of both capping and approximate local search for A-DESIGN are polynomial in $n, d, k, \frac{1}{\delta}$ and the size of the input. The runtime analysis of approximate local search algorithm for A-DESIGN is identical to the one for D-DESIGN (with a change of objective, but the objective can still be computed ion polynomial time).

The significant change is the use of capping algorithm, which needs to be shown to ternimate in polynomial time. Let L be the maximum number of the length of binary string that encodes the number in each component across all input vectors v_i. Then $||v_i||^2 \leq \sqrt{d} \cdot 2^{2L}$ for all i's. In each iteration, the capping algorithm reduces the length of at least one vector by at least half, and hence by $n \log \frac{2^{2L}}{\Delta} = O(nL \log \frac{1}{\Delta})$ iteration of capping, all vectors have length at most Δ. As in the analysis of approximate local search for D-DESIGN, the encoding length of $\phi^{\mathrm{A}}(V)$ is polynomial in n, d, k, L, and so is $\log \frac{1}{\Delta}$ (as $\Delta = \frac{d}{\epsilon^2 \phi^{\mathrm{A}}(V)}$). Hence, the capping algorithm takes polynomial (in n, d, k, L) number of steps.

Appendix E. Greedy Algorithm for D-DESIGN

To prove Theorem 2, we again use the convex programming relaxation for the D-DESIGN problem. Recall the relaxation (D-REL) and its dual (D-REL-DUAL) shown in figure 2b. ϕ_f^{D} denote the be the common optimum value of (D-REL) and its dual (D-REL-DUAL). $I^\star$ denote the indices of the vector in the optimal solution and let $\phi^{\mathrm{D}} = \det\left(\sum_{i\in I^\star} v_i v_i^\top\right)^{\frac{1}{d}}$ be its objective. Observe that $\phi_f^{\mathrm{D}} \geq \log \phi^{\mathrm{D}}$. Now, Theorem 2 follows from the following theorem with an appropriate initialization of first d vectors which will be specified later.

Algorithm 8 Greedy algorithm for D-DESIGN

Input: $V = v_1, \ldots, v_n \in \mathbb{R}^d, d \leq k \in \mathbb{N}, S_0 \subset [n]$.
$X_0 = \sum_{j \in S_0} v_j v_j^\top$.
for $i = 1$ to $k - |S_0|$ **do**
  $j_i = \text{argmax}_{j \in [n]} \det(X + v_j v_j^\top)$
  $S_i = S_{i-1} \cup \{j_i\}, X_i = X_{i-1} + v_{j_i} v_{j_i}^\top$
end for
$I = S_{k-|S_0|}, X = X_{k-|S_0|}$
Return (I, X).

Theorem 32 *For any set of vectors $v_1, \ldots, v_n \in \mathbb{R}^d$, suppose $S_0 \subset [1, n]$ is a set of size d such that $\det\left(\sum_{i \in S_0} v_i v_i^\top\right)^{\frac{1}{d}} > \frac{d}{k}\kappa \cdot \phi^{\mathrm{D}}$ for some $\frac{1}{e} \geq \kappa > 0$ and $k \geq \frac{d}{\epsilon}\left(\log \frac{1}{\epsilon} + \log\log \frac{1}{\kappa}\right)$. Let (I, X) be the solution returned by Algorithm 8. Then,*

$$\det(X) \geq (1 - 5\epsilon)\phi^{\mathrm{D}}$$

Before we prove Theorem 32, we state and prove the following theorem, which better conveys main ideas of the proof.

Theorem 33 *For any set of vectors $v_1, \ldots, v_n \in \mathbb{R}^d$ and $k \geq \frac{d \log \frac{1}{\epsilon}}{\epsilon}$, suppose $S_0 \subset [1, n]$ is a set of size d such that $\det\left(\sum_{i \in S_0} v_i v_i^\top\right)^{\frac{1}{d}} > \frac{d}{k}\kappa \cdot \phi^{\mathrm{D}}$ for some $1 > \kappa > 0$. Let $s = \max\{d \log\log \frac{1}{\kappa}, 0\}$ and (I, X) be the solution returned by picking $k - d + s$ vectors greedily. Then,*

$$\det(X) \geq (1 - 4\epsilon)\phi^{\mathrm{D}}$$

Theorem 33 gives a bi-criteria approximation where we pick small number s of extra vectors than the budget k while obtaining near-optimal solution. These s vectors are required to improve the initial approximation $\frac{d}{k}\kappa$ to a ratio $\frac{d}{k}$ independent of n or κ.
Proof [Theorem 33] To prove this theorem, we show the following two lemmas. First lemma shows the increase in the solution value in each greedy step.

Lemma 34 *For $t \in [0, k - |S_0| - 1]$, $\det(X_{t+1}) \geq \det(X_t)\left(1 + \frac{d}{k}\frac{e^{\phi_f^{\mathrm{D}}}}{(\det(X_t))^{1/d}}\right)$*

Next lemma shows that this recursion leads to the desired bound in the theorem.

Lemma 35 *Let $\ell \geq 0$. Let $z_0, \ldots, z_{k-\ell}$ be such that for $t \in [0, k - \ell - 1]$, $z_{t+1} \geq z_t\left(1 + \frac{d}{kz_t}\right)^{1/d}$. Then,*

1. *If $z_0 < \frac{d}{k}$, then for any $s \geq d \log\log \frac{dz_0}{k}$, we have*

$$z_s \geq \frac{d}{ek}$$

2. *If $z_0 \geq \frac{d}{ek}$, then we have*

$$z_{k-\ell} \geq \frac{k - d - \ell}{k} - \frac{2d}{k}\log\frac{k}{d}$$

Proof of Theorem 33 follows from these two lemmas by defining $z_t = \frac{e^{\phi_f^{\mathrm{D}}}}{(\det(X_t))^{1/d}}$ in the bound in Lemma 34. Lemma 35 implies that for any initial κ approximation with d initial vectors to the D design problem of k vectors, $s = d \log\log \frac{1}{\kappa}$ vectors is enough to guarantee $\frac{d}{ek}$-approximation. Then, the second bound of Lemma 35 applies for the rest of the greedy algorithm. We now prove these two lemmas.

Proof [Lemma 34] By definition, $\det(X_{t+1}) = \max_{j\in[n]} \det(X_t + v_j v_j^\top)$. By Lemma 17, $\det(X_t + v_j v_j^\top) = \det(X_t)(1 + v_j^\top X_t^{-1} v_j)$. Hence,

$$\det(X_{t+1}) = \det(X_t)\left(1 + \max_{j\in[n]} v_j^\top X_t^{-1} v_j\right) \tag{17}$$

Next, we lower bound $\max_{j\in[n]} v_j^\top X_t^{-1} v_j$ by constructing a feasible solution to the (D-REL-DUAL). Let

$$Y = \alpha X_t, \qquad \mu = \max_{j\in[n]} v_j^\top Y^{-1} v_j = \frac{1}{\alpha} \max_{j\in[n]} v_j^\top X_t^{-1} v_j$$

where α will be fixed later. Then, (Y, μ) is a feasible solution of (D-REL-DUAL). Hence,

$$\phi_f^{\mathrm{D}} \le \frac{1}{d} \log\det(\alpha X_t) + \frac{k}{d} \cdot \frac{1}{\alpha} \max_{j\in[n]} v_j^\top X_t^{-1} v_j - 1$$

which implies

$$\frac{d\alpha}{k}\left(\phi_f^{\mathrm{D}} + 1 - \log\alpha - \frac{1}{d}\log\det(X_t)\right) \le \max_{j\in[n]} v_j^\top X_t^{-1} v_j$$

Setting, $\alpha = \frac{e^{\phi_f^{\mathrm{D}}}}{\det(X_t)^{1/d}}$, we get

$$\max_{j\in[n]} v_j^\top X_t^{-1} v_j \ge \frac{d}{k} \frac{e^{\phi_f^{\mathrm{D}}}}{\det(X_t)^{1/d}} \left(\phi_f^{\mathrm{D}} + 1 - \log\frac{e^{\phi_f^{\mathrm{D}}}}{\det(X_t)^{1/d}} - \frac{1}{d}\log\det(X_t)\right) = \frac{d}{k}\frac{e^{\phi_f^{\mathrm{D}}}}{\det(X_t)^{1/d}}$$

Substituting the bounds in equation (17), we get

$$\det(X_{t+1}) \ge \det(X_t)\left(1 + \frac{d}{k}\frac{e^{\phi_f^{\mathrm{D}}}}{(\det(X_t))^{1/d}}\right).$$

This finishes the proof of Lemma 34. ∎

Proof [Lemma 35] We first prove the first bound. The recursion implies that $\frac{z_{t+1}}{z_t} \ge \left(\frac{d}{kz_t}\right)^{\frac{1}{d}}$, which is equivalent to

$$\log z_{t+1} \ge \frac{1}{d}\log\frac{d}{k} + \frac{d-1}{d}\log z_t \tag{18}$$

Define $a_t := \log\frac{d}{k} - \log z_t$. If $a_u \le 0$ for any $u \le s$, then we are done because $z_s \ge z_u \ge \frac{d}{k}$. Else, we can rearrange terms to obtain

$$a_{t+1} \le \left(1 - \frac{1}{d}\right) a_t \tag{19}$$

Hence, we have

$$\begin{aligned} a_s &\le \left(1 - \frac{1}{d}\right)^s a_0 \\ &\le e^{-\frac{s}{d}} a_0 \le e^{-\frac{s}{d}} \log \frac{dz_0}{k} \\ &\le 1 \end{aligned}$$

where the last inequality follows from $s \ge \log\log \frac{dz_0}{k}$. Therefore, $\log \frac{d}{k} - \log z_s = a_s \le 1$, giving the desired bound.

To prove the second bound, the recursion is equivalent to

$$\log \frac{z_{t+1}}{z_t} \ge \frac{1}{d} \log\left(1 + \frac{d}{kz_t}\right) \tag{20}$$

It is clear that z_t is an increasing sequence in t, hence $\frac{d}{kz_t} \le \frac{d}{kz_0} = e$. We use $\log(1+x) \ge \frac{x}{e}$ for $0 \le x \le e$ (by concavity of $\log\ x$) to lower bound the right-hand-side of (20) above inequality to obtain

$$\log \frac{z_{t+1}}{z_t} \ge \frac{1}{d} \cdot \frac{d}{ekz_t} = \frac{1}{ekz_t}$$

Thus, by using $e^x \ge 1 + x$, we have $\frac{z_{t+1}}{z_t} \ge e^{\frac{1}{ekz_t}} \ge 1 + \frac{1}{ekz_t}$, which implies

$$z_{t+1} \ge z_t + \frac{1}{ek}$$

Therefore, we obtain $z_t \ge \frac{t}{ek}$ for all $t \ge 0$.

Next, we apply the bound $\log(1+x) \ge x - \frac{x^2}{2} = x\left(1 - \frac{x}{2}\right)$ whenever $0 \le x$ on the right-hand-side of (20) to obtain

$$\log \frac{z_{t+1}}{z_t} \ge \frac{1}{d}\frac{d}{kz_t}\left(1 - \frac{d}{2kz_t}\right) \ge \frac{1}{kz_t} \cdot \left(1 - \frac{2d}{t}\right)$$

where the last inequality comes from $z_t \ge \frac{t}{ek}$. Thus, applying $e^x \ge 1 + x$, we have $\frac{z_{t+1}}{z_t} \ge 1 + \frac{1}{kz_t} \cdot \left(1 - \frac{2d}{t}\right)$, which implies

$$z_{t+1} \ge z_t + \frac{1}{k} - \frac{2d}{tk} \tag{21}$$

Summing (21) from $t = d$ to $t = k - \ell - 1$ gives

$$\begin{aligned} z_{k-\ell} &\ge z_d + \frac{k-d-\ell-1}{k} - \frac{2d}{k}\left(\frac{1}{d} + \frac{1}{d+1} + \ldots + \frac{1}{k-\ell-1}\right) \\ &\ge \frac{k-d-\ell}{k} - \frac{2d}{k}\log\frac{k}{d} \end{aligned}$$

as desired. ■

Now we prove Theorem 33. We first pick s vectors greedily to guarantee that $z_s \geq \frac{d}{ek}$. (If $z_0 > \frac{d}{ek}$, then $s = 0$.) Substituting $\ell = d$ and $k \geq \frac{d \log \frac{1}{\epsilon}}{\epsilon}$ in Lemma 35 gives

$$z_{k-\ell} \geq 1 - \frac{d}{k}\left(2 + 2\log\frac{k}{d}\right)$$
$$\geq 1 - \frac{2\epsilon}{\log\frac{1}{\epsilon}}\left(1 + \log\frac{1}{\epsilon} + \log\log\frac{1}{\epsilon}\right) \geq 1 - 4\epsilon$$

where the second inequality follows from $\frac{1}{x}(1 + \log x)$ being decreasing function on $x \geq 1$, and the last inequality is by $1 + x \leq e^x$ with $x = \log\frac{1}{\epsilon}$. ■

We are now ready to prove the main theorem.

Proof [Theorem 32] The proof is identical to the proof of Theorem 33 except that, after using $s = \log\log\frac{1}{\kappa}$ vectors to obtain $\frac{d}{ek}$-approximation, we only take $k - d - s$ greedy steps instead of $k - d$ greedy steps. Hence, we set $\ell = d + s$ to the second bound of Lemma 35 to obtain

$$z_{k-\ell} \geq \frac{k - 2d - s}{k} - \frac{2d}{k}\log\frac{k}{d} = 1 - \frac{d}{k}\left(2 + 2\log\frac{k}{d}\right) - \frac{s}{k}$$

We have $1 - \frac{d}{k}\left(2 + 2\log\frac{k}{d}\right) \geq 1 - 4\epsilon$ identical to the proof of Theorem 33. By $k \geq \frac{d}{\epsilon}\log\log\frac{1}{\kappa} = \frac{s}{\epsilon}$, we have $\frac{s}{k} \leq \epsilon$, completing the proof. ■

We finally note on combinatorial algorithms for setting initial solution of size d. One may use volume sampling algorithms to achieve $\frac{n}{k}$-approximation to optimal objective in for picking d vectors (Avron and Boutsidis, 2013). Alternatively, we can perform local search on initial d vectors to obtain $d(1+\delta)$-approximation in time polynomial in $\frac{1}{\delta}$, as shown in Section C. Since we know that the relaxation gaps of A- and D- optimal design are at most $\frac{k}{k-d+1}$, we can bound the optimum values of design problems between picking d and k vectors to be at most k multiplicative factor apart (Avron and Boutsidis, 2013; Nikolov et al., 2019). The approximation ratios of two algorithms are hence n and $dk(1+\delta)$, respectively. We formalize this argument and the result with locally optimal initial set as the following statement, which proves Theorem 2.

Corollary 36 *Greedy algorithm initialized by a local optimal set of size d returns a $(1+5\epsilon)$-approximation whenever $k \geq \frac{d}{\epsilon}(\log\frac{1}{\epsilon} + \log\log d + 1)$.*

We first argue the ratio of optimum D-DESIGN values when the size of the set is d and k. Denote $\phi^{\mathrm{D}}(d), \phi^{\mathrm{D}}(k) = \phi^{\mathrm{D}}$ the optimum D-DESIGN objective $\det\left(\sum_{i\in S} v_i v_i^\top\right)^{\frac{1}{d}}$ on size d, k, respectively. Denote $\phi_f^{\mathrm{D}}(d), \phi_f^{\mathrm{D}}(k) = \phi_f^{\mathrm{D}}$ the common optimum value of (D-REL) and its dual (D-REL-DUAL) for size constraints of d, k respectively.

Claim 16 *We have*

$$\phi^{\mathrm{D}}(k) \leq k\phi^{\mathrm{D}}(d)$$

Proof Because (D-REL) is a relaxation of D-DESIGN (up to $\log$ scale), we have

$$\exp\phi_f^{\mathrm{D}}(k) \geq \phi^{\mathrm{D}}(k), \qquad \exp\phi_f^{\mathrm{D}}(d) \geq \phi^{\mathrm{D}}(d)$$

We may scale any optimal solution of (D-REL) with size k to size d by applying $x_i := \frac{d}{k} x_i$ coordinate-wise. Therefore, we have

$$\phi_f^{\mathrm{D}}(d) \geq \phi_f^{\mathrm{D}}(k) + \log \frac{d}{k}$$

Finally, we know that the integrality gap of (D-REL) is $\frac{k}{k-d+1}$. This follows from the approximation result of local search algorithm which compares the objective value of returned set to the objective to the convex relaxation. (This exact bound of the gap also follows from previous work on proportional volume sampling (Nikolov et al., 2019).) We apply this gap for size budget d to obtain

$$\exp \phi_f^{\mathrm{D}}(d) \leq d\phi^{\mathrm{D}}(d)$$

Therefore, we have

$$\phi^{\mathrm{D}}(k) \leq \exp \phi_f^{\mathrm{D}}(k) \leq \frac{k}{d} \exp \phi_f^{\mathrm{D}}(d) \leq k\phi^{\mathrm{D}}(d) \tag{22}$$

as desired. ■

Proof [Corollary 36] Theorem 1 implies that a local search solution satisfies d-approximation when budget size is d. Hence, by Claim 16, a local solution is dk-approximation compared to D-DESIGN with a size budget of k.

We now apply Theorem 32: it is sufficient to show that

$$k \geq \frac{d}{\epsilon}\left(\log \frac{1}{\epsilon} + \log \log \frac{1}{\kappa}\right) \tag{23}$$

for $\kappa = \frac{1}{d^2}$, so the result follows. ■

Appendix F. Greedy Algorithm for A-DESIGN

In this section, we prove Theorem 4. As remarked in the case of local search algorithm, we need to modify the instance to cap the length of the vectors in the case of greedy algorithm as well. This is done by Algorithm 2. As shown in Lemma 11, the value of any feasible solution only increases after capping and the value of the convex programming relaxation increases by a small factor if k is large.

We now show that the greedy algorithm run on these vectors returns a near optimal solution. For any input vectors $V = \{v_1, \ldots, v_n\}$, the primal program is A-REL(V) and the dual program is A-REL-DUAL(V). $\phi_f^{\mathrm{A}}(V)$ denotes the (common) optimal value of objective values of the convex program with input vectors from V. $I^\star$ denotes the indices of the vectors in the optimal solution of A-DESIGN with input vector set V and $\phi^{\mathrm{A}}(V) = \operatorname{tr}\left(\left(\sum_{i \in I^\star} v_i v_i^\top\right)^{-1}\right)$ be its objective. We show the following theorem about Algorithm 9 in terms of capping length Δ.

Theorem 37 *Let $||u_i||_2^2 \leq \Delta$, $S_0 \subseteq [n]$ of size $r \geq d$ such that $\operatorname{tr}\left(\left(\sum_{i \in S_0} u_i u_i^\top\right)^{-1}\right) \leq \kappa \cdot \phi^{\mathrm{A}}(U)$ for some $\kappa \geq 1$, and $\Lambda = \sqrt{\frac{\Delta \phi_f^{\mathrm{A}}(U)}{k}}$. Let (I, X) be the solution returned by Algorithm 9. Then we have*

$$\operatorname{tr}(X^{-1}) \leq \left(1 - \frac{d+r}{k} - 2\Lambda \log \frac{k \max\{\Lambda\kappa, 1\}}{d}\right)^{-1} \phi^{\mathrm{A}}(U)$$

Algorithm 9 Greedy algorithm for A-DESIGN

Input: $U = u_1, \ldots, u_n \in \mathbb{R}^d, d \le k \in \mathbb{N}, S_0 \subset [n]$.
$X_0 = \sum_{j \in S_0} u_j u_j^\top$.
for $i = 1$ to $k - |S_0|$ **do**
$j_i = \text{argmin}_{j \in [n]} \operatorname{tr}\left(\left(X + u_j u_j^\top\right)^{-1}\right)$
$S_i = S_{i-1} \cup \{j_i\}, X_i = X_{i-1} + u_{j_i} u_{j_i}^\top$
end for
$I = S_{k-|S_0|}, X = X_{k-|S_0|}$.
Return (I, X).

Similar to the analysis of local search for A-DESIGN, capping vector length is necessary to obtain theoretical guarantee. We will optimize over the length Δ later in Theorem 40.

Proof [Theorem 37] To prove the theorem, we show the following two lemmas:

Lemma 38 *For any $t \in [0, k - |S_0|]$, let $z_t = \operatorname{tr}(X_t^{-1})/\phi_f^A(U)$. Then, for any $t \in [0, k - |S_0| - 1]$,*

$$z_{t+1} \le z_t \left(1 - \frac{z_t}{k\left(1 + z_t \sqrt{\frac{\Delta \phi_f^A(U)}{k}}\right)}\right)$$

Lemma 39 *Let $\Lambda \ge 0$ and $\ell \ge 0$. Suppose $z_{t+1} \le z_t \left(1 - \frac{z_t}{k(1+z_t\Lambda)}\right)$ for all $t \ge 0$, then*

1. *If $z_0 > \frac{1}{\Lambda}$, then for any $s \ge 2\Lambda k \log(\Lambda z_0)$, we have*

$$z_s \le \frac{1}{\Lambda}$$

2. *If $z_0 \le \frac{1}{\Lambda}$, we have*

$$z_{k-\ell} \le \left(1 - \frac{d+\ell}{k} - 2\Lambda \log \frac{k}{d}\right)^{-1}$$

Proof [Lemma 38] By definition,

$$\operatorname{tr}(X_{t+1}^{-1}) = \min_{j \in [n]} \operatorname{tr}\left(\left(X_t + u_j u_j^\top\right)^{-1}\right).$$

By Sherman-Morrison formula,

$$\operatorname{tr}(X_{t+1}^{-1}) = \operatorname{tr}(X_t^{-1}) - \max_{j \in [n]} \frac{u_j^\top X_t^{-2} u_j}{1 + u_j^\top X_t^{-1} u_j}$$

Note that $u_j^\top X_t^{-1} u_j = \langle u_j, X_t^{-1} u_j \rangle$. By Cauchy-Schwarz inequality, $u_j^\top X_t^{-1} u_j$ is at most $||u_j||_2 ||X_t^{-1} u_j||_2 = ||u_j||_2 \sqrt{u_j^\top X_t^{-2} u_j}$. Since, $||u_j||_2^2 \leq \Delta$, we get $u_j^\top X_t^{-1} u_j \leq \sqrt{\Delta \cdot u_j^\top X_t^{-2} u_j}$. Hence,

$$\operatorname{tr}(X_{t+1}^{-1}) \leq \operatorname{tr}(X_t^{-1}) - \max_{j \in [n]} \frac{u_j^\top X_t^{-2} u_j}{1 + \sqrt{\Delta \cdot u_j^\top X_t^{-2} u_j}} \tag{24}$$

Next, we lower bound $\max_{j\in[n]} u_j^\top X_t^{-2} u_j$ by finding a feasible solution to A-REL-DUAL. Let,

$$Y = \gamma X_t^{-2}, \qquad \lambda = \max_{j\in[n]} u_j^\top Y u_j = \gamma \max_{j\in[n]} u_j^\top X_t^{-2} u_j$$

where $\gamma > 0$ will be fixed later. Then, (Y, λ) is a feasible solution to A-REL-DUAL(U). Hence,

$$\phi_f^{\mathrm{A}}(U) \geq 2 \operatorname{tr}\left(\left(\gamma X_t^{-2} \right)^{1/2} \right) - k\gamma \max_{j\in[n]} u_j^\top X_t^{-2} u_j$$

$$\max_{j\in[n]} u_j^\top X_t^{-2} u_j \geq \frac{1}{k\gamma} \left(2\sqrt{\gamma} \operatorname{tr}(X_t^{-1}) - \phi_f^{\mathrm{A}}(U) \right)$$

Substituting $\gamma = \left(\frac{\phi_f^{\mathrm{A}}(U)}{\operatorname{tr}(X_t^{-1})} \right)^2$, we get

$$\max_{j\in[n]} u_j^\top X_t^{-2} u_j \geq \frac{\operatorname{tr}(X_t^{-1})^2}{k \phi_f^{\mathrm{A}}(U)}.$$

As proved in Claim 3, $\frac{x}{1+c\sqrt{x}}$ is a monotonically increasing function for $x \geq 0$ if $c \geq 0$. Hence,

$$\max_{j\in[n]} \frac{u_j^\top X_t^{-2} u_j}{1 + \sqrt{\Delta \cdot u_j^\top X_t^{-2} u_j}} \geq \frac{\frac{\operatorname{tr}(X_t^{-1})^2}{k\phi_f^{\mathrm{A}}(U)}}{1 + \sqrt{\Delta \frac{\operatorname{tr}(X_t^{-1})^2}{k\phi_f^{\mathrm{A}}(U)}}}$$

Substituting $z_t = \frac{\operatorname{tr}(X_t^{-1})}{\phi_f^{\mathrm{A}}(U)}$, we get

$$\max_{j\in[n]} \frac{u_j^\top X_t^{-2} u_j}{1 + \sqrt{\Delta \cdot u_j^\top X_t^{-2} u_j}} \geq \frac{\operatorname{tr}(X_t^{-1})}{k} \frac{z_t}{1 + z_t \sqrt{\frac{\Delta \phi_f^{\mathrm{A}}(U)}{k}}}.$$

Substituting this inequality in Equation (24), we get

$$\operatorname{tr}(X_{t+1}^{-1}) \leq \operatorname{tr}(X_t^{-1}) \left(1 - \frac{z_t}{k \left(1 + z_t \sqrt{\frac{\Delta \phi_f^{\mathrm{A}}(U)}{k}} \right)} \right).$$

Substituting $z_t = \operatorname{tr}(X_t^{-1}) / \phi_f^{\mathrm{A}}(U)$ and $z_{t+1} = \operatorname{tr}(X_{t+1}^{-1}) / \phi_f^{\mathrm{A}}(U)$, we get

$$z_{t+1} \leq z_t \left(1 - \frac{z_t}{k \left(1 + z_t \sqrt{\frac{\Delta \phi_f^{\mathrm{A}}(U)}{k}} \right)} \right).$$

This finishes the proof of Lemma 38. ■

Proof [Lemma 39] We first prove the first bound. If $z_t \le \frac{1}{\Lambda}$ for any $t < s$, then we are done, so assume $z_t\Lambda \ge 1$. The recursion then implies

$$z_{t+1} \le z_t \left(1 - \frac{z_t}{k(2z_t\Lambda)}\right) = z_t \left(1 - \frac{1}{2k\Lambda}\right)$$

Therefore,

$$\begin{aligned} z_s &\le z_0 \left(1 - \frac{1}{2k\Lambda}\right)^s \\ &\le z_0 e^{-\frac{1}{2\Lambda}s} \le z_0 e^{-\log \Lambda z_0} = \frac{1}{\Lambda} \end{aligned}$$

as desired.

We now prove the second bound. Let $a_t = \frac{1}{z_t}$. Then the recursion $z_{t+1} \le z_t \left(1 - \frac{z_t}{k(1+z_t\Lambda)}\right)$ can be rewritten as

$$\frac{a_{t+1}}{a_t} \ge \left(1 - \frac{1}{k\,(\Lambda + a_t)}\right)^{-1} \tag{25}$$

Applying $\left(1 - \frac{1}{k(\Lambda+a_t)}\right)^{-1} \ge 1 + \frac{1}{k(\Lambda+a_t)}$ and rearranging terms, we obtain

$$a_{t+1} \ge a_t + \frac{a_t}{k(\Lambda + a_t)} = a_t + \frac{1}{k} - \frac{\Lambda}{k(\Lambda + a_t)} \tag{26}$$

It is obvious from (25) that a_t is an increasing sequence, and hence $a_t \ge a_0 \ge \Lambda$ for all $t \ge 0$. So (26) implies

$$a_{t+1} \ge a_t + \frac{1}{k} - \frac{\Lambda}{k(2\Lambda)} = a_t + \frac{1}{2k} \tag{27}$$

Therefore, we have $a_t \ge \frac{t}{2k}$ for all $t \ge 0$.

Using this bound $a_t \ge \frac{t}{2k}$, the recursion (26) also implies

$$a_{t+1} \ge a_t + \frac{1}{k} - \frac{\Lambda}{k(\frac{t}{2k})} = a_t + \frac{1}{k} - \frac{2\Lambda}{t} \tag{28}$$

Summing 28 from $t = d$ to $t = k - \ell - 1$ gives

$$\begin{aligned} a_{k-\ell} &\ge a_d + \frac{k-d-\ell}{k} - 2\Lambda \sum_{t=d}^{k-\ell-1} \frac{1}{t} \\ &= \frac{k-d-\ell}{k} - 2\Lambda \log \frac{k}{d} \end{aligned}$$

proving the desired bound. ■

We now prove Theorem 37. The first bound of Lemma 39 shows that with initial approximation κ, we require $s = \max\{0, 2\Lambda k \log(\Lambda\kappa)\}$ steps to ensure $\frac{1}{\Lambda}$ approximation ratio. After that, we can pick $k-r-s$ vectors. Hence, we apply the second bound of Lemma 39 with $\ell = r+s$ to get the approximation ratio of X as

$$\begin{aligned}
z_{k-\ell} &\leq \left(1 - \frac{d+r+s}{k} - 2\Lambda \log \frac{k}{d}\right)^{-1} \\
&= \left(1 - \frac{d+r}{k} - 2\Lambda \left(\log \frac{k}{d} + \max\left\{\log \Lambda\kappa, 0\right\}\right)\right)^{-1} \\
&= \left(1 - \frac{d+r}{k} - 2\Lambda \log \frac{k \max\{\Lambda\kappa, 1\}}{d}\right)^{-1}
\end{aligned}$$

proving the desired bound. ■

Next, we tune Δ in Theorem 37 and use Lemma 11 to obtain the final bound, from which Theorem 4 will follow.

Theorem 40 *For input vectors $V = \{v_1, \ldots, v_n\}$ and parameter $k \in \mathbb{N}$, let $U = \{u_1, \ldots, u_n\}$ be the set of vectors returned by the Capping Algorithm 2 with input vector set V and $\Delta = \frac{d}{\epsilon \phi^{\mathrm{A}}(V)}$. Let $S_0 \subseteq [n]$ be an initial set of size $r \geq d$ where $\operatorname{tr}\left(\left(\sum_{i \in S_0} u_i u_i^\top\right)^{-1}\right) \leq \kappa \cdot \phi^{\mathrm{A}}(U)$ for some $\kappa \geq 1$. Let (I, X) be the solution returned by Algorithm 9 with vector set U and parameter k. If $k \geq \frac{r}{\epsilon} + \frac{d\left(\log^2 \kappa + \log^2 \frac{1}{\epsilon}\right)}{\epsilon^3}$ and $\epsilon \leq 0.0001$, then*

$$\operatorname{tr}\left(\left(\sum_{i \in I} v_i v_i^\top\right)^{-1}\right) \leq (1 + 6000\epsilon)\phi^{\mathrm{A}}(V)$$

Proof By Lemma 11, substituting Δ, we have

$$\begin{aligned}
\phi_f^{\mathrm{A}}(U) &\leq \left(1 + \frac{5000d}{k}\right)\left(\phi_f^{\mathrm{A}}(V) + 150\epsilon\phi^{\mathrm{A}}(V)\right) \\
&\leq (1 + 5500\epsilon)\phi^{\mathrm{A}}(V) \qquad (29)
\end{aligned}$$

where the last inequality follows from $\phi^{\mathrm{A}}(V) \geq \phi_f^{\mathrm{A}}(V)$, $k \geq \frac{d}{\epsilon}$. and $\epsilon \leq 0.0001$. Thus, we have

$$\Lambda = \sqrt{\frac{\Delta \phi_f^{\mathrm{A}}(U)}{k}} = \sqrt{\frac{d\phi_f^{\mathrm{A}}(U)}{\epsilon k \phi^{\mathrm{A}}(V)}} \leq \sqrt{\frac{d(1+5500\epsilon)}{\epsilon k}} \leq 2\sqrt{\frac{d}{\epsilon k}}$$

Next, Theorem 37 implies that

$$\operatorname{tr}(X^{-1}) \leq \left(1 - \frac{d+r}{k} - 2\Lambda \log \frac{k \max\{\Lambda\kappa, 1\}}{d}\right)^{-1} \phi^{\mathrm{A}}(U) \qquad (30)$$

Note that

$$\begin{aligned}
2\Lambda \log \frac{k \max\{\Lambda\kappa, 1\}}{d} &\leq 2\Lambda \log \frac{k\kappa}{d} \\
&\leq 4\sqrt{\frac{d}{\epsilon k}} \log \frac{k}{d} + 4\sqrt{\frac{d}{\epsilon k}} \log \kappa
\end{aligned}$$

Since $\frac{1}{\sqrt{x}}\log x$ is a decreasing function on $x \geq 8$, applying $k \geq \frac{d\log^2\frac{1}{\epsilon}}{\epsilon^3}$, we have

$$\sqrt{\frac{d}{\epsilon k}}\log\frac{k}{d} \leq \frac{\epsilon}{\log\frac{1}{\epsilon}}\left(3\log\frac{1}{\epsilon}+\log\log\frac{1}{\epsilon}+\log 2\right) \leq 4\epsilon$$

where the last inequality follows from $\epsilon \leq 0.0001$. Also, applying $k \geq \frac{d\log^2\frac{1}{\kappa}}{\epsilon^3}$, $k \geq \frac{d\log^2\frac{1}{\epsilon}}{\epsilon^3} \geq \frac{d}{\epsilon}$, and $k \geq \frac{r}{\epsilon}$, we have

$$\sqrt{\frac{d}{\epsilon k}}\log\kappa \leq \epsilon, \qquad \frac{d}{k} \leq \epsilon, \qquad \frac{r}{k} \leq \epsilon$$

Hence, (30) implies that

$$\mathrm{tr}(X^{-1}) \leq (1-22\epsilon)^{-1}\,\phi^{\mathrm{A}}(U) \tag{31}$$

Combining (31) with Lemma 11 and (29) gives

$$\begin{aligned}\mathrm{tr}\left(\left(\sum_{i\in I} v_i v_i^\top\right)^{-1}\right) &\leq \mathrm{tr}(X^{-1}) \leq (1-22\epsilon)^{-1}(1+5500\epsilon)\phi^{\mathrm{A}}(V)\\ &\leq (1+6000\epsilon)\phi^{\mathrm{A}}(V)\end{aligned}$$

where the last inequality follows from $\epsilon \leq 0.0001$. ■

We note an efficient combinatorial algorithm of volume sampling (Avron and Boutsidis, 2013; Dereziński and Warmuth, 2017) that gives $\frac{n}{k}$-approximation to the A-DESIGN problem of selecting d vectors (note that these randomized algorithms can be derandomized, e.g. by rejection sampling). Alternatively, from our result on approximate local search algorithm for A-DESIGN in Section D, we can also initialize with $c\cdot d$ vectors for an absolute constant c and perform local search algorithm to obtain $1+0.0001+\delta$ approximation in time polynomial in $\frac{1}{\delta}$ for some small δ. Similar to Claim 16, we can relate the optimum of A-DESIGN of size budget $d \leq r \leq k$ and k to be at most factor $\frac{k}{r-d+1}$ apart (Avron and Boutsidis, 2013; Nikolov et al., 2019). Hence, the volume sampling on initial set of size d and local search on initial set of size cd give approximation ratio of n and $\frac{k}{cd-d+1}(1+0.0001+\delta) \leq \frac{k}{d}$, respectively; that is, κ can be set to n or $\frac{k}{d}$ in Theorem 40 and we adjust r accordingly. Using the local search on initial cd vectors to set the value of κ and r, we prove Theorem 4.

Proof [Theorem 4] Suppose $k \geq C\cdot\frac{d}{\epsilon^3}\log^2\frac{1}{\epsilon}$ for some absolute constant $C > 0$ to be specified later and $\epsilon \leq 0.0001$. By Theorem 40, it is sufficient to have $k \geq \frac{r}{\epsilon}+\frac{d\left(\log^2\kappa+\log^2\frac{1}{\epsilon}\right)}{\epsilon^3}$, where $\kappa = \frac{k}{d}$ and $r = cd$ by initializing the greedy algorithm with an output from an approximate local search algorithm of size cd for an absolute constant c. By checking the derivative of $f(k) := k-\frac{cd}{\epsilon}-\frac{d\left(\log^2\frac{k}{d}+\log^2\frac{1}{\epsilon}\right)}{\epsilon^3}$, $f(k)$ is increasing when $2d\log\frac{k}{d} \leq k\epsilon^3$, which is true for a large enough C. Hence, we only need to show $f(k) \geq 0$ for $k = C\cdot\frac{d}{\epsilon^3}\log^2\frac{1}{\epsilon}$. The condition $f(k) \geq 0$ is equivalent to

$$C\log^2\frac{1}{\epsilon} \geq \log^2\frac{C\log^2\frac{1}{\epsilon}}{\epsilon^3}+\log^2\frac{1}{\epsilon}+c\epsilon^2 \tag{32}$$

It is clear that $\log^2 \frac{1}{\epsilon} + c\epsilon^2 \leq \frac{C}{2} \log^2 \frac{1}{\epsilon}$ for $C \geq 3 + c$. We also have

$$\begin{aligned}
\log^2 \frac{C \log^2 \frac{1}{\epsilon}}{\epsilon^3} &= \left(\log C + 3 \log \frac{1}{\epsilon} + 2 \log \log \frac{1}{\epsilon} \right)^2 \\
&\leq \left(\log C + 5 \log \frac{1}{\epsilon} \right)^2 \\
&\leq \left(\sqrt{\frac{C}{2}} - 5 + 5 \log \frac{1}{\epsilon} \right)^2 \\
&\leq \left(\sqrt{\frac{C}{2}} \log \frac{1}{\epsilon} \right)^2
\end{aligned}$$

where we use $x \leq e^x$ for $x = \log \frac{1}{\epsilon}$, $\log C \leq \sqrt{C} - 5$ for a sufficiently large C, and $\log \frac{1}{\epsilon} \geq 1$ for the three inequalities above, respectively. Hence, we finished the proof of (32). ■