Exact and heuristic solution approaches for the D-Optimality problem

Gabriel Ponte
Universidade Federal do Rio de Janeiro
Centro de Tecnologia, Rio de Janeiro, CEP: 21941-90
gabrielponte@poli.ufrj.br

Marcia Fampa
Universidade Federal do Rio de Janeiro
PESC/COPPE, Rio de Janeiro, CEP: 21941-90
fampa@cos.ufrj.br

Jon Lee
University of Michigan
Ann Arbor, MI 48109, USA
jonxlee@umich.edu

RESUMO

O problema de D-otimalidade é um problema de otimização inteira não-linear fundamental e desafiador, com origem na estatística. Dado um conjunto finito de pontos para experimentos (ou seja, especificações de experimentos) e um orçamento para o número de experimentos a serem realizados, o problema busca a escolha do conjunto ótimo de pontos, de modo a minimizar a variância generalizada da estimativa de parâmetros de mínimos quadrados para um modelo linear baseado nos pontos escolhidos. Investigamos heurísticas de busca local para este problema, bem como um algoritmo branch-and-bound baseado em sua relaxação contínua. Por fim, a teoria da dualidade de otimização convexa é aplicada no desenvolvimento de desigualdades válidas para o problema, que podem reduzir o intervalo de possíveis valores inteiros para as variáveis.

PALAVRAS CHAVE. D-otimalidade. Programa não-linear inteiro. Branch-and-cut. Busca local.

PM , OC , EST&MP

ABSTRACT

The D-optimality problem is a fundamental and challenging integer nonlinear optimization problem coming from statistics. Among a finite set of design points (i.e., specifications of experiments), and a budget for the number of experiments to carry out, it seeks to choose the optimal set of design points, so as to minimize the generalized variance of least-squares parameter estimates for a linear model based on the chosen design points. We investigate local-search heuristics for this problem as well as a branch-and-bound algorithm based on its continuous relaxation. Finally, convex optimization duality theory is applied in the development of valid inequalities for the problem that can reduce the range of possible integer values for the variables.

KEYWORDS. D-optimality. Integer nonlinear program. Branch-and-cut. Local search.

PM , OC , EST&MP

1. Introduction

We consider the D-Optimality problem formulated as

$$\max \left\{ \operatorname{ldet} \sum_{\ell \in N} x_\ell v_\ell v_\ell^\top \ : \ \mathbf{e}^\top x = s,\ 0 \leq x \leq u,\ x \in \mathbb{Z}^n \right\}, \tag{D-Opt}$$

where $v_\ell \in \mathbb{R}^m$, for $\ell \in N := \{1, \ldots, n\}$, and $u \in \mathbb{R}^n$. D-Opt is a fundamental problem in statistics. Defining $A := (v_1, v_2, \ldots, v_n)^\top$, we consider the least-squares regression problem $\min_{\theta \in \mathbb{R}^m} \|A\theta - y\|_2$, where y is an arbitrary response vector. We assume that A has full column rank, and so there is unique solution to the least-squares problem. But we consider a situation where each v_ℓ corresponds to a costly experiment, which could be carried out between 0 and u_ℓ times. Overall, we have a budget to carry out a total of s experiments, and so we specify the choices by the n-vector x in D-Opt. For a given feasible solution $\hat{x}$, we define $A_{\hat{x}}$ to be a matrix that has $\hat{x}_\ell$ copies of $v_\ell^\top$ as its rows. This leads to the reduced least-squares problem $\min_{\theta \in \mathbb{R}^m} \|A_{\hat{x}}\theta - y\|_2$. The generalized variance of the least-squares parameter estimator $\hat{\theta}$ is inversely proportional to $\det \sum_{\ell \in N} \hat{x}_\ell v_\ell v_\ell^\top$, and so D-Opt corresponds to picking the set of experiments to minimize the generalized variance of the least-squares parameter estimator $\hat{\theta}$ (see Fedorov [1972], for example). There is a large literature on heuristic algorithms for D-Opt and its variations. Welch [1982] was the first to approach D-Opt with an exact branch-and-bound algorithm, employing a bound based on Hadamard's inequality and another based on continuous relaxation (apparently without using state-of-the art NLP solvers of that time). Ko et al. [1998] proposed a spectral bound (also see Lee and Lind [2020]) and analytically compared it with the Hadamard bound.

We present local-search heuristics to obtain a feasible solution for D-Opt. We also investigate the solution of the problem by applying a branch-and-bound algorithm based on the convex continuous relaxation of D-Opt. We consider the construction of valid inequalities to D-Opt, based on a lower bound for D-Opt and on the knowledge of a feasible solution for the Lagrangian dual of the continuous relaxation. In our numerical experiments, we compare the different proposed heuristics, we investigate the ability of the branch-and-bound algorithm to obtain the optimal solution of the tested instances, and the ability of the valid inequalities to reduce the range of feasible values of the variables in D-Opt, and even to fix them in the root node of the branch-and-bound algorithm.

A similar solution approach has been successfully applied to the related maximum entropy sampling problem (MESP) (see Anstreicher et al. [1999]; Anstreicher [2018, 2020]), where given the covariance matrix C of a Gaussian random n-vector, one searches for a subset of s random variables which maximizes the "information" (measured by "differential entropy") (see Shewry and Wynn [1987]; Caselton and Zidek [1984]; Lee [2012], for example).

In Sec. 2, we present the continuous relaxation for D-Opt and its Lagrangian dual problem. In Sec. 3, we present valid inequalities for D-Opt based on the duality theory for convex optimization. In Sec. 4, we propose local-search heuristics for D-Opt. In Sec. 5, we discuss our numerical experiments, and in Sec. 6, we present our final remarks and describe future work.

Notation. We let $\mathbb{S}^n$ (resp., $\mathbb{S}^n_+$, $\mathbb{S}^n_{++}$) denote the set of symmetric (resp., positive-semidefinite, positive-definite) matrices of order n. We let $\mathbf{diag}(x)$ denote the $n \times n$ diagonal matrix with diagonal elements given by the components of $x \in \mathbb{R}^n$. We denote an all-ones vector by $\mathbf{e}$ and an identity matrix by I. For matrices A and B, $A \bullet B := \operatorname{Trace}(A^\top B)$ is the matrix dot-product. For matrix A, we denote row i by $A_{i\cdot}$ and column j by $A_{\cdot j}$.

2. Convex relaxation and duality

In this section we construct the convex continuous relaxation of D-Opt and its Lagrangian, which will be used for the construction of valid inequalities for the problem.

We define $A := (v_1, v_2, \ldots, v_n)^\top$ and we note that

$$\textstyle\sum_{\ell \in N} x_\ell v_\ell v_\ell^\top = A^\top \mathbf{diag}(x) A\,.$$

Then, we introduce the convex relaxation of D-Opt given by

$$\max \left\{ \operatorname{ldet}\left(A^\top \mathbf{diag}(x) A\right) \,:\, \mathbf{e}^\top x = s,\, 0 \le x \le u,\, x \in \mathbb{R}^n \right\}. \tag{1}$$

Let $F(x) \in \mathbb{S}^m$ be defined by $F(x) := A^\top \mathbf{diag}(x) A$. Then, (1) can be rewritten as

$$\begin{array}{ll} \max & \operatorname{ldet} W, \\ \text{s.t.} & F(x) - W = 0, \\ & \mathbf{e}^\top x = s, \\ & 0 \le x \le u. \end{array}$$

In order to formulate the Lagrangian dual problem of (1), we consider the Lagrangian function

$$\mathcal{L}(W, x, \lambda, \theta, \nu, \Lambda) = \operatorname{ldet} W + \lambda^\top (u - x) + \theta^\top x + \nu(s - \mathbf{e}^\top x) + \Lambda \bullet (F(x) - W)$$

with $\text{dom}\mathcal{L} = \mathbb{S}^m_{++} \times \mathbb{R}^n \times \mathbb{R}^n \times \mathbb{R} \times \mathbb{S}^m$, and the corresponding dual function

$$\mathcal{L}^*(\lambda, \theta, \nu, \Lambda) := \sup_{W \succ 0, x} \mathcal{L}(W, x, \lambda, \theta, \nu, \Lambda). \tag{2}$$

The Lagrangian dual problem of (1) is:

$$\min\{\mathcal{L}^*(\lambda, \theta, \nu, \Lambda) \,:\, \lambda \ge 0,\ \theta \ge 0\}.$$

In the following, we consider $F(x) = F_1 x_1 + F_2 x_2 + \ldots + F_n x_n$, where $F_\ell := A_{\ell.}^\top A_{\ell.}$, for $\ell \in N$. As $\operatorname{ldet}(\cdot)$ is a concave function in the positive definite cone, the supremum in (2) occurs where the gradient of the Lagrangian function is equal to zero. Then, the Lagrangian dual may be written as

$$\begin{array}{ll} \min & \operatorname{ldet} W + \lambda^\top (u - x) + \theta^\top x + \nu(s - \mathbf{e}^\top x) + \Lambda \bullet (F(x) - W), \\ \text{s.t.} & W^{-1} - \Lambda = 0, \\ & \Lambda \bullet F_i - \lambda_i + \theta_i - \nu = 0, \quad i \in N, \\ & W \succ 0, \lambda \ge 0, \theta \ge 0. \end{array} \tag{3}$$

Theorem 1. *The following equation is valid for all feasible solutions of (3):*

$$\lambda^\top (u - x) + \theta^\top x + \nu(s - \mathbf{e}^\top x) + \Lambda \bullet (F(x) - W) = \lambda^\top u + \nu s - m.$$

Proof. From the first constraint in (3) we have that $W = \Lambda^{-1}$, and from the group of constraints for all $i \in N$, we have

$$\begin{array}{l} \Lambda \bullet F_i - \lambda_i + \theta_i - \nu = 0 \Rightarrow \\ \Lambda \bullet F_i x_i - \lambda_i x_i + \theta_i x_i - \nu x_i = 0 \Rightarrow \\ \Lambda \bullet \sum_{i \in N} F_i x_i - \lambda^\top x + \theta^\top x - \nu \mathbf{e}^\top x = 0. \end{array}$$

So,

$$\begin{array}{l} \lambda^\top (u - x) + \theta^\top x + \nu(s - \mathbf{e}^\top x) + \Lambda \bullet (F(x) - W) \\ \quad = \lambda^\top u + \nu s - \Lambda \bullet W + \left(\Lambda \bullet \sum_{i \in N} F_i x_i - \lambda^\top x + \theta^\top x - \nu \mathbf{e}^\top x\right) \\ \quad = \lambda^\top u + \nu s - m. \end{array}$$

□

Finally, considering Thm. 1, the Lagrangian dual of (1) can be formulated as

$$\begin{array}{ll} \min & -\operatorname{ldet}\Lambda + \lambda^\top u + \nu s - m, \\ \text{s.t.} & \Lambda \bullet F_i - \lambda_i + \theta_i - \nu = 0, \quad i \in N, \\ & \Lambda \succ 0, \lambda \geq 0, \theta \geq 0. \end{array} \tag{4}$$

3. Valid inequalities

In Thm. 2, we show how to construct valid inequalities for D-Opt based on the knowledge of a lower bound and a feasible solution for the dual problem (4).

Theorem 2. *Let*

- *LB be the objective-function value of a feasible solution for D-Opt.*
- $(\hat{\Lambda}, \hat{\lambda}, \hat{\theta}, \hat{\nu})$ *be a feasible solution for* (4) *with objective-function value* $\hat{\zeta}$.

Then, for every optimal solution $x^\star$ *for D-Opt, we have:*

$$x_k^\star \leq \left\lfloor \left(\hat{\zeta} - LB\right) / \hat{\theta}_k \right\rfloor, \qquad \forall\, k \in N \text{ such that } \hat{\theta}_k > 0, \tag{5}$$

$$x_k^\star \geq u_k - \left\lfloor \left(\hat{\zeta} - LB\right) / \hat{\lambda}_k \right\rfloor, \qquad \forall\, k \in N \text{ such that } \hat{\lambda}_k > 0. \tag{6}$$

Proof. Consider D-Opt with the additional constraint $x_k \geq \alpha_k$, where $\alpha_k := \left\lfloor \left(\hat{\zeta} - \text{LB}\right) / \hat{\theta}_k \right\rfloor + 1$. Then, its dual becomes

$$\begin{array}{ll} \min & -\operatorname{ldet}\Lambda + \lambda^\top u + \nu s - m - \alpha_k \omega, \\ \text{s.t.} & \Lambda \bullet F_i - \lambda_i + \theta_i - \nu = 0, \qquad i \in N \setminus \{k\}, \\ & \Lambda \bullet F_k - \lambda_k + \theta_k - \nu + \omega = 0, \\ & \Lambda \succ 0, \lambda \geq 0, \theta \geq 0, \omega \geq 0. \end{array}$$

where ω is the new dual variable. Note that if $(\hat{\Lambda}, \hat{\lambda}, \hat{\theta}, \hat{\nu})$ is a feasible solution for (4), then $(\hat{\Lambda}, \hat{\lambda}, \hat{\theta} - \omega \mathbf{e}_k, \hat{\nu}, \omega)$ is a feasible solution of this modified dual, *as long as* $\omega \geq 0$ and $\hat{\theta}_k - \omega \geq 0$. The objective value of this modified dual is $\hat{\zeta} - \alpha_k \omega$. So, to minimize the objective value of our feasible solution of this modified dual, we set ω equal to $\hat{\theta}_k$. We conclude that $\hat{\zeta} - \alpha_k \hat{\theta}_k$ is an upper bound on the objective value of all solutions of D-Opt that satisfy $x_k \geq \alpha_k$. So, if $\hat{\zeta} - \alpha_k \hat{\theta}_k < \text{LB}$, then no optimal solution of D-Opt can have $x_k \geq \alpha_k$. Furthermore, we have

$$\begin{aligned} &\left\lfloor \left(\hat{\zeta} - \text{LB}\right) / \hat{\theta}_k \right\rfloor + 1 > \left(\hat{\zeta} - \text{LB}\right) / \hat{\theta}_k \Rightarrow \\ &\hat{\zeta} - \left(\left\lfloor \left(\hat{\zeta} - \text{LB}\right) / \hat{\theta}_k \right\rfloor + 1\right) \hat{\theta}_k < \hat{\zeta} - \left(\left(\hat{\zeta} - \text{LB}\right) / \hat{\theta}_k\right) \hat{\theta}_k = \text{LB}. \end{aligned}$$

Therefore, the inequalities (5) are satisfied by any optimal solution of D-Opt. Next, consider D-Opt with the additional constraint $x_k \leq \alpha_k$, where $\alpha_k := u_k - \left\lfloor \left(\hat{\zeta} - \text{LB}\right) / \hat{\lambda}_k \right\rfloor - 1$. The dual becomes instead,

$$\begin{array}{ll} \min & -\operatorname{ldet}\Lambda + \lambda^\top u + \nu s - m + \alpha_k \omega, \\ \text{s.t.} & \Lambda \bullet F_i - \lambda_i + \theta_i - \nu = 0, \qquad i \in N \setminus \{k\}, \\ & \Lambda \bullet F_k - \lambda_k + \theta_k - \nu - \omega = 0, \\ & \Lambda \succ 0, \lambda \geq 0, \theta \geq 0, \omega \geq 0. \end{array}$$

where ω is again the new dual variable. Note that if $(\hat{\Lambda}, \hat{\lambda}, \hat{\theta}, \hat{\nu})$ is a feasible solution for (4), then $(\hat{\Lambda}, \hat{\lambda} - \omega \mathbf{e}_k, \hat{\theta}, \hat{\nu}, \omega)$ is a feasible solution of this modified dual, *as long as* $\omega \geq 0$ and $\hat{\lambda}_k - \omega \geq 0$. The objective value of this modified dual is $\hat{\zeta} - (u_k - \alpha_k)\omega$. So, to minimize the objective value of our feasible solution of this modified dual, we set ω equal to $\hat{\lambda}_k$. We conclude that $\hat{\zeta} - (u_k - \alpha_k)\hat{\lambda}_k$ is an upper bound on objective value of all solutions of D-Opt that satisfy $x_k \leq \alpha_k$. So if $\hat{\zeta} - (u_k - \alpha_k)\hat{\lambda}_k < LB$, then no optimal solution of D-Opt can have $x_k \leq \alpha_k$. Furthermore, we have

$$\begin{aligned}
&\left\lfloor \left(\hat{\zeta} - \text{LB}\right)/\hat{\lambda}_k \right\rfloor + 1 > \left(\hat{\zeta} - \text{LB}\right)/\hat{\lambda}_k \Rightarrow \\
&\hat{\zeta} - u_k\hat{\lambda}_k + u_k\hat{\lambda}_k - \left(\left\lfloor \left(\hat{\zeta} - \text{LB}\right)/\hat{\lambda}_k \right\rfloor + 1\right)\hat{\lambda}_k < \hat{\zeta} - \left(\left(\hat{\zeta} - \text{LB}\right)/\hat{\lambda}_k\right)\hat{\lambda}_k \Rightarrow \\
&\hat{\zeta} - u_k\hat{\lambda}_k + \left(u_k - \left\lfloor \left(\hat{\zeta} - \text{LB}\right)/\hat{\lambda}_k \right\rfloor - 1\right)\hat{\lambda}_k < \text{LB}.
\end{aligned}$$

Therefore, the inequalities (6) are satisfied by any optimal solution of D-Opt. □

We note that when the right-hand side of inequality (5) (resp., (6)) is 0 (resp., u_k), we can fix x_k. We can also fix the variable if both right-hand sides are the same.

4. Local-search heuristics

In this section we describe our local-search procedures and methods to initialize them.

4.1. Initial solutions for the local-search procedures

Next, we show how we obtain initial solutions for our local-search procedures. To ensure that we start the procedures with a feasible solution for D-Opt with finite objective value, we initially construct a vector $\tilde{x} \in \{0,1\}^n$, such that $\mathbf{e}^\top \tilde{x} = m$ and $A^\top \text{diag}(\tilde{x}) A \in \mathbb{S}^m_{++}$. This is equivalent to choosing m linearly independent rows of A, and setting $\tilde{x}$ as the support vector for the selected subset of rows. We denote the set of indices of the selected rows by $\tilde{N}$. To select the rows, we use the Matlab function nsub[1] (see Fampa et al. [2021] for details). Then, we use one of the three algorithms described in the following subsections to construct a feasible solution to D-Opt from $\tilde{x}$.

4.1.1. Feasible solutions from a solution of the continuous relaxation

We present two procedures to construct a feasible solution to D-Opt from $\tilde{x}$, considering a feasible solution of (1). Let $A = U\Sigma V^\top$ be the real singular-value decomposition of A (see Golub and Van Loan [1996], for example), where $U \in \mathbb{R}^{n\times n}$, $V \in \mathbb{R}^{m\times m}$ are orthonormal matrices and $\Sigma = \text{diag}(\sigma_1, \sigma_2, \ldots, \sigma_m) \in \mathbb{R}^{n\times m}$ $(n \geq m)$ with singular values $\sigma_1 \geq \sigma_2 \geq \cdots \geq \sigma_m \geq 0$. Note that the first s columns of U are associated with the s largest singular values of A. In addition, for each $k \in N$, we have $\sum_{j\in N} U^2_{jk} = 1$ and $\sum_{j\in N} U^2_{kj} = 1$. Then, let

$$x^0_j := \textstyle\sum_{k=1}^{s} U^2_{jk}, \quad j \in N.$$

We clearly have $\mathbf{e}^\top x^0 = s$ and $0 \leq x^0_j \leq 1$, for all $j \in N$. So, x^0 is a feasible solution of (1).

Let τ be the permutation of the indices in N, such that $x^0_{\tau(1)} \geq x^0_{\tau(2)} \geq \cdots \geq x^0_{\tau(n)}$. Next, we propose two procedures to construct a feasible solution $\bar{x}$ for D-Opt, considering x^0.

- "Binary" (see Alg. 1): Let $\bar{N}$ be the first $s - m$ indices in τ that are not in $\tilde{N}$. Set $\bar{x}_j := 1$, for $j \in \bar{N}$, and $\bar{x}_j := \tilde{x}_j$, for $j \notin \bar{N}$.
- "Integer" (see Alg. 2): Let $\bar{u} = u - \tilde{x}$ and $\bar{s} = s - m$. Define, for all $j \in N$,

$$\tilde{y}_{\tau(j)} := \min\left\{\bar{u}_k, \max\left\{0, \bar{s} - \textstyle\sum_{i=1}^{j-1} \tilde{y}_{\tau(i)}\right\}\right\}.$$

  Then, set $\bar{x} := \tilde{y} + \tilde{x}$.

[1] www.mathworks.com/matlabcentral/fileexchange/83638-linear-independent-rows-and-columns-generator

Algorithm 1: Initial Binary Solution

Input: $A \in \mathbb{R}^{n\times m}$, $U \in \mathbb{R}^{n\times n}$
Output: Initial solution $\bar{x}$ of D-Opt

Initialize x^0 and $\bar{x}$ as zero vectors;
$S := \text{nsub}(A, m)$;
for $j = 1, \ldots, n$ **do**
  if $j \in S$ **then**
    $\bar{x}_j := 1$;
  else
    $x^0_j := \sum_{i=1}^{s} U^2_{ji}$;
for $j = 1, \ldots, s - m$ **do**
  $\hat{\jmath} := \text{argmax}\{x^0\}$;
  $\bar{x}_{\hat{\jmath}} := 1$;
  $x^0_{\hat{\jmath}} := 0$;

Algorithm 2: Initial Integer Solution

Input: $A \in \mathbb{R}^{n\times m}$, $U \in \mathbb{R}^{n\times n}$
Output: Initial solution $\bar{x}$ of D-Opt

Initialize x^0 and $\bar{x}$ as zero vectors;
$S := \text{nsub}(A, m)$;
for $j = 1, \ldots, n$ **do**
  if $j \in S$ **then**
    $\bar{x}_j := 1$;
  $x^0_j := \sum_{i=1}^{s} U^2_{ji}$;
$k := m$;
while $k < s$ **do**
  $\hat{\jmath} := \text{argmax}\{x^0\}$;
  $w := u_{\hat{\jmath}} - \bar{x}_{\hat{\jmath}}$;
  if $k + w < s$ **then**
    $u_k := w$;
  else
    $u_k := s - k$;
  $\bar{x}_{\hat{\jmath}} := \bar{x}_{\hat{\jmath}} + u_k$;
  $x^0_{\hat{\jmath}} := 0$;
  $k := k + u_k$;

4.1.2. Feasible solution obtained with a greedy procedure

Proposition 1. *Let $A = U\Sigma V^\top$ be the real singular-value decomposition of A. Then the objective function of D-Opt is given by* $\text{ldet}\left(\Sigma^\top U^\top \mathbf{diag}(x) U\Sigma\right)$.

Proof.

$$\begin{aligned}\log\left(\det\left(A^\top \mathbf{diag}(x) A\right)\right) &= \log\left(\det(V)\cdot\det\left(U^\top\Sigma^\top \mathbf{diag}(x) U\Sigma\right)\cdot\det(V^\top)\right)\\ &= \log\left(\det\left(\Sigma^\top U^\top \mathbf{diag}(x) U\Sigma\right)\right).\end{aligned}$$

□

From Prop. 1, we observe that the choice of x is related to the rows of $U\Sigma$. Then, an alternative feasible solution to D-Opt can be obtained by using both matrices U and Σ. Let

$$x_j^0 := \sum_{i=1}^{m} \left(U_{ji}\Sigma_{ii}\right)^2 \quad j \in N\ .$$

Let τ be the permutation of the indices in N such that $x^0_{\tau(1)} \geq x^0_{\tau(2)} \geq \cdots \geq x^0_{\tau(n)}$. Next, we propose a procedure to construct an integer feasible solution $\bar{x}$ for D-Opt, from x^0.

- "Greedy" (see Alg. 3): Let $\bar{N}$ be the first $s - m$ indices in τ that are not in $\tilde{N}$. Set $\bar{x}_j := 1$, for $j \in \bar{N}$, and $\bar{x}_j := \tilde{x}_j$, for $j \notin \bar{N}$.

  We note that the solution x^0 is not feasible to (1), but its components may carry more information about which rows of A should be chosen in an optimal solution for D-Opt.

Algorithm 3: Initial Greedy Solution

Input: $A \in \mathbb{R}^{n\times m}$, $U \in \mathbb{R}^{n\times n}$, $\Sigma \in \mathbb{R}^{n\times m}$
Output: Initial solution $\bar{x}$ of D-Opt

```
Initialize x^0 and x̄ as zero vectors;
S := nsub(A, m);
for j = 1, ..., n do
    if j ∈ S then
        x̄_j := 1;
    else
        x^0_j := Σ_{i=1}^{m} (U_ji Σ_ii)^2;
for j = 1, ..., s − m do
    ĵ := argmax{x^0};
    x̄_ĵ := 1;
    x^0_ĵ := 0;
```

4.2. Local-search procedures

In Alg. 4, we present the local-search procedures that consider as the criterion for improvement of the given solution, the increase in the value of the objective function of D-Opt. The neighborhood of a given solution $\bar{x}$ is defined by

$$\mathcal{N}(\bar{x}) := \{y \in \mathbb{Z}^n \ :\ 0 \leq y \leq u,\ y_i = \bar{x}_i - 1,\ y_j = \bar{x}_j + 1, y_k = \bar{x}_k, k \neq i, k \neq j, \forall i, j \in N\}.$$

The three local-search procedures described next were considered in our experiments.

- "FI" (Local Search First Improvement): Starting from $\bar{x}$, the procedure visits the solution in $\mathcal{N}(\bar{x})$ with increased objective value with respect to $\bar{x}$, such that i is the least possible index, and j is the least possible index for the given i.

- "FI+" (Local Search First Improvement Plus): Starting from $\bar{x}$, the procedure visits the solution in $\mathcal{N}(\bar{x})$ with increased objective value with respect to $\bar{x}$, such that i is the least possible index, and j is selected in N, as the index that maximizes the objective value, for the given i.

- "BI" (Local Search Best Improvement): Starting from $\bar{x}$, the procedure visits the solution in $\mathcal{N}(\bar{x})$ with increased objective value with respect to $\bar{x}$, such that i and j are selected in N, as the pair of indices that maximizes the objective value.

Algorithm 4: Local search procedures

Input: A solution $\bar{x}$ of D-Opt
Output: A solution $\bar{x}$ of D-Opt, possibly updated

$x^0 := \bar{x}$;
$z^0 := \text{ldet}(A^\top \mathbf{diag}(x^0)A)$;
$flag := true$;
while $flag$ **do**
  $flag := false$;
  for $i = 1, \ldots, n$, *such that* $\bar{x}_i > 0$ **do**
    for $j = 1, \ldots, n$, *such that* $\bar{x}_j < u_j$ *e* $j \neq i$ **do**
      $x := \bar{x}$;
      $x_i := \bar{x}_i - 1$;
      $x_j := \bar{x}_j + 1$;
      $z := \text{ldet}(A^\top \mathbf{diag}(x)A)$;
      if $z > z^0$ **then**
        $x^0 := x$;
        $z^0 := z$;
        $flag := true$;
        if *"First improvement"* **then**
          break loops for i and j;
    if *"First improvement plus"* & $flag == true$ **then**
      break loop for i;
  $\bar{x} := x^0$;

5. Numerical Experiments

We have implemented the three local-search procedures described in Subsection 4.2, namely, "FI", "FI+", and "BI". We have initialized each procedure with the three methods proposed in Subsection 4.1, namely, "Binary", "Integer", and "Greedy". We have also obtained optimal solutions for the tested instances with a branch-and-bound algorithm, where the bounds were obtained with the convex continuous relaxation (1). Finally, we have investigated the effect of the valid inequalities described in Thm. 2, using the values of the optimal dual variables for the continuous relaxation (1) (at the root node of the branch-and-bound enumeration tree) and two alternatives feasible solutions for D-Opt: the best solution obtained by our local-search procedures and the optimal solution obtained with the branch-and-bound algorithm.

The algorithms proposed were coded in Julia v.1.7.1. To solve the convex relaxation (1), we apply Mosek using the Julia package MosekTools v.0.12.1. We ran the experiments on a 16-core machine (running Windows Server 2016 Standard): two Intel Xeon CPU E5-2667 v4 processors running at 3.20GHz, with 8 cores each, and 128 GB of memory.

To construct our test instances, we used the Matlab function sprand to randomly generate 15 $n \times m$ dimensional matrices A with $m := \lfloor 0.25n \rfloor$ and rank m. We generated three instances for each $n \in \{20, 30, 50, 60, 80\}$. We also used the Matlab function randi to generate a random vector u of each dimension n, with integer values between 1 and 3. Finally, we consider $s := 0.5n$.

In Table 1, we compare the solutions of three local-search procedures and their initializations. We see that "Binary" computes the best initial solutions for the vast majority of instances, and that "Integer" never computes the best initial solution. Moreover, there is no clear winner among the three local-search procedures when considering the value of the solution computed by them. We finally observe, especially for the largest instances with $n = 80$, that although "Binary leads to the

best initial solutions, when starting from them, the local-search procedures converge faster, but to a worse local optimum when comparing to use of the other initialization procedures. It indicates that the solution of "Binary" must be close to these worse local optima.

	Initial Solutions			Binary			Integer			Greedy		
n, m, s	Binary	Integer	Greedy	FI	FI+	BI	FI	FI+	BI	FI	FI+	BI
20,5,10	4.736	3.785	4.634	5.741	5.722	5.722	5.722	5.722	5.722	5.722	5.723	5.722
20,5,10	5.488	4.261	5.433	6.206	6.206	6.206	6.206	6.206	6.206	6.195	6.195	6.195
20,5,10	5.313	4.397	5.094	5.696	5.645	5.645	5.696	5.645	5.645	5.645	5.645	5.645
30,7,15	6.677	5.414	6.375	8.484	8.484	8.484	8.484	8.484	8.484	8.484	8.484	8.484
30,7,15	6.831	5.814	7.377	8.549	8.549	8.549	8.549	8.549	8.549	8.549	8.549	8.549
30,7,15	7.436	7.519	7.826	9.186	9.186	9.186	9.186	9.186	9.186	9.186	9.186	9.186
50,12,25	12.398	10.884	11.128	13.615	13.615	13.615	13.615	13.615	13.615	13.615	13.615	13.615
50,12,25	11.020	5.863	8.601	13.419	13.454	13.454	13.396	13.418	13.418	13.396	13.457	13.428
50,12,25	11.935	10.674	9.288	14.083	14.101	14.081	14.087	14.101	14.101	14.101	14.101	14.100
60,15,30	15.182	13.273	14.084	16.695	16.695	16.695	16.744	16.744	16.750	16.723	16.744	16.723
60,15,30	13.832	11.973	12.271	16.863	16.863	16.895	16.835	16.933	16.878	16.953	16.941	16.752
60,15,30	12.730	10.676	12.119	16.993	16.993	16.993	17.068	17.068	16.993	16.993	16.993	16.965
80,20,40	19.422	16.318	18.647	21.467	21.467	21.467	21.499	21.487	21.487	21.363	21.363	21.528
80,20,40	19.337	14.999	17.663	21.437	21.437	21.437	21.518	21.518	21.448	21.345	21.452	21.426
80,20,40	19.145	14.596	16.113	21.560	21.551	21.548	21.595	21.560	21.595	21.623	21.623	21.562

Table 1: Results for the local-search procedures

The elapsed times for these procedures were presented in Table 2. We see that the initialization procedures are equally fast on these instances and that the "BI" local-search converges faster than the others, on average.

	Initial Solutions			Binary			Integer			Greedy		
n, m, s	Binary	Integer	Greedy	FI	FI+	BI	FI	FI+	BI	FI	FI+	BI
20,5,10	0.000	0.000	0.000	0.003	0.005	0.004	0.008	0.006	0.006	0.035	0.004	0.002
20,5,10	0.000	0.000	0.000	0.002	0.002	0.002	0.003	0.003	0.003	0.003	0.002	0.002
20,5,10	0.000	0.000	0.000	0.006	0.003	0.003	0.006	0.004	0.003	0.002	0.002	0.002
30,7,15	0.000	0.000	0.000	0.028	0.051	0.016	0.016	0.011	0.012	0.019	0.045	0.013
30,7,15	0.000	0.000	0.000	0.024	0.023	0.014	0.047	0.025	0.010	0.027	0.037	0.016
30,7,15	0.000	0.000	0.000	0.010	0.011	0.009	0.031	0.013	0.013	0.018	0.014	0.020
50,12,25	0.000	0.000	0.000	0.125	0.204	0.126	0.221	0.149	0.103	0.128	0.156	0.128
50,12,25	0.000	0.000	0.000	0.336	0.136	0.094	0.201	0.118	0.069	0.226	0.178	0.077
50,12,25	0.000	0.000	0.000	0.308	0.150	0.060	0.193	0.112	0.061	0.113	0.101	0.048
60,15,30	0.001	0.000	0.000	0.401	0.270	0.142	0.300	0.224	0.121	0.341	0.187	0.119
60,15,30	0.000	0.000	0.000	0.110	0.149	0.083	0.177	0.218	0.145	0.194	0.348	0.116
60,15,30	0.000	0.000	0.000	0.609	0.473	0.151	0.408	0.545	0.213	0.473	0.414	0.176
80,20,40	0.001	0.001	0.001	4.729	3.270	1.818	7.801	6.885	3.166	7.092	6.824	2.508
80,20,40	0.001	0.001	0.001	7.077	5.871	2.394	11.027	9.186	2.347	5.977	7.258	2.416
80,20,40	0.001	0.001	0.001	5.740	5.026	1.683	8.986	7.364	3.052	11.280	9.302	2.360

Table 2: Elapsed time for the local-search procedures (seconds)

In Table 3, we evaluate the quality of best heuristic solution obtained by the local-search procedures and the tightness of the upper bound given by the continuous relaxation, comparing their values to the optimal objective value obtained with the branch-and-bound algorithm. To better expose the comparison, besides the objective values in each case, we also present the absolute gaps between the solution of the continuous relaxation and both solutions of the local search and of the branch-and-bound on the last two columns. We see that, even for the largest instances, the local

search computes solutions with absolute gap of no more than 1% of the optimal solution value. From the last column in Table 3, we can also see that the continuous relaxation (1) gives tight upper bounds for the instances tested. Following the usual behavior for challenging combinatorial problems, the elapsed time for branch-and-bound grows fast with the dimension of the problem, and we could not solve to optimality the last two instances in the table, given a time limit of 36 hours.

	Objective value			Elapsed Time (sec)			Absolute Gap	
n, m, s	LS	BB	Continuous	LS	BB	Continuous	LS	BB
20,5,10	5.741	5.768	5.802	0.075	5.318	0.014	0.061	0.035
20,5,10	6.206	6.206	6.226	0.021	0.654	0.012	0.020	0.020
20,5,10	5.696	5.696	5.735	0.032	3.699	0.015	0.040	0.040
30,7,15	8.484	8.484	8.549	0.212	102.343	0.023	0.065	0.065
30,7,15	8.549	8.549	8.604	0.221	9.834	0.021	0.055	0.055
30,7,15	9.186	9.186	9.232	0.139	0.744	0.016	0.045	0.045
50,12,25	13.615	13.660	13.712	1.340	126.982	0.091	0.097	0.052
50,12,25	13.457	13.457	13.543	1.436	2479.589	0.043	0.086	0.086
50,12,25	14.101	14.104	14.180	1.147	1247.339	0.045	0.079	0.076
60,15,30	16.750	16.750	16.779	2.107	537.693	0.141	0.030	0.030
60,15,30	16.953	16.975	17.017	1.540	209.266	0.158	0.064	0.041
60,15,30	17.068	17.083	17.162	3.462	304.257	0.089	0.094	0.079
80,20,40	21.528	21.607	21.707	44.096	25821.883	0.464	0.179	0.100
80,20,40	21.518	-	21.670	53.554	-	0.465	0.152	-
80,20,40	21.623	-	21.789	54.796	-	0.462	0.167	-

Table 3: Local-search procedure versus branch-and-bound

In Tables 4 and 5, we investigate the effect of the valid inequalities presented in Thm. 2 when applied only at the root node of the branch-and-bound enumeration tree. In Table 4, we present the average reduction in the range of possible values for the variables x_k after computing upper bounds ub_k and lower bounds lb_k for their values with the valid inequalities, for $k \in N$. To compute the inequalities in Thm. 2, we used the values of the dual variables of the continuous relaxation (1), and two different solutions to compute the lower bound LB: the best solution obtained by the local-search procedures and the optimal solution of the branch-and-bound. We see in Table 4, that it was possible to reduce the average range in all instances tested even with the weaker lower bound. Furthermore, we see that for only one instance (with $n = 80$) it was not possible to fix variables using the inequalities. Finally, we see that using the optimal solution, we obtain better results, indicating that an improvement in our local search procedures can lead to a more effective application of our inequalities when the optimal solution is unknown. In Table 5, we show the values in which the variables were fixed. Although they were fixed at zero in most cases, we also see cases where the variables are fixed at positive values. Once more, we see an increase in the number of variables fixed when we improve the lower bound, using the optimal solution value. These results indicate a promising use of the inequalities inside a branch-and-cut framework, where the cuts would be used to change the lower and upper bounds on the variables as we go deeper in the branch-and-bound enumeration tree.

	LB given by local search $(\text{ub}_k - \text{lb}_k)/u_k$					LB given BB $(\text{ub}_k - \text{lb}_k)/u_k$				
n, m, s	0	0.333	0.5	0.666	1	0	0.333	0.5	0.666	1
20,5,10	3	0	0	0	17	3	0	0	0	17
20,5,10	7	0	0	0	13	7	0	0	0	13
20,5,10	2	0	0	0	18	2	0	0	0	18
30,7,15	8	2	0	0	20	8	2	0	0	20
30,7,15	9	0	0	0	21	9	0	0	0	21
30,7,15	13	0	0	0	17	13	0	0	0	17
50,12,25	**7**	**1**	**1**	**1**	**40**	**9**	**1**	**0**	**1**	**39**
50,12,25	5	1	4	2	38	5	1	4	2	38
50,12,25	9	1	2	1	37	9	1	2	1	37
60,15,30	10	0	0	0	50	10	0	0	0	50
60,15,30	**7**	**0**	**0**	**0**	**53**	**9**	**0**	**0**	**1**	**50**
60,15,30	7	0	0	0	53	7	0	0	0	53
80,20,40	**1**	**0**	**0**	**1**	**78**	**2**	**1**	**1**	**0**	**76**
80,20,40	0	1	1	2	76			-		
80,20,40	2	1	0	0	77			-		

Table 4: Effect of valid inequalities: average reduction in range of values for the variables

	LB given by local search u_k				LB given by BB u_k			
n, m, s	0	1	2	3	0	1	2	3
20,5,10	2	1	0	0	2	1	0	0
20,5,10	5	1	1	0	5	1	1	0
20,5,10	1	1	0	0	1	1	0	0
30,7,15	8	0	0	0	8	0	0	0
30,7,15	9	0	0	0	9	0	0	0
30,7,15	12	1	0	0	12	1	0	0
50,12,25	**7**	**0**	**0**	**0**	**9**	**0**	**0**	**0**
50,12,25	5	0	0	0	5	0	0	0
50,12,25	9	0	0	0	9	0	0	0
60,15,30	7	3	0	0	7	3	0	0
60,15,30	**7**	**0**	**0**	**0**	**9**	**0**	**0**	**0**
60,15,30	6	1	0	0	6	1	0	0
80,20,40	**1**	**0**	**0**	**0**	**2**	**0**	**0**	**0**
80,20,40	0	0	0	0			-	
80,20,40	1	1	0	0			-	

Table 5: Effect of valid inequalities: number of variables fixed at each integer value

6. Conclusions

Our numerical experiments indicate promising directions to investigate in order to improve the efficiency of the branch-and-bound algorithm to solve the D-optimality problem. As natural extension of this work, for example, we consider the introduction of the valid inequalities studied, in branch-and-cut algorithm, where the cuts would in fact, tight the lower and upper bounds on the variables as we go deeper in the branch-and-bound enumeration tree. Other initializations for our local-search procedures could also be tested, as we could observe that they had some impact on the solution to which the procedures have converged to. We should investigate initializations which consider reductions on the components of the vector u until we get a feasible solution to D-Opt.

This procedure would be complementary to the one considered in this work, where we start from a zero vector and increase its components until a feasible solution is obtained.

Acknowledgment

G. Ponte was supported in part by CNPq PIBIC scholarship 149149/2020-4. M. Fampa was supported in part by CNPq grants 305444/2019-0 and 434683/2018-3. J. Lee was supported in part by AFOSR grants FA9550-19-1-0175 and FA9550-22-1-0172.

References

Anstreicher, K., Fampa, M., Lee, J., and Williams, J. (1999). Using continuous nonlinear relaxations to solve constrained maximum-entropy sampling problems. *Mathematical Programming*, 85: 221–240.

Anstreicher, K. M. (2018). Maximum-entropy sampling and the Boolean quadric polytope. *Journal of Global Optimization*, 72(4):603–618.

Anstreicher, K. M. (2020). Efficient solution of maximum-entropy sampling problems. *Operations Research*, 68(6):1826–1835.

Caselton, W. F. and Zidek, J. V. (1984). Optimal monitoring network design. *Statistics and Probability Letters*, 2:223–227.

Fampa, M., Lee, J., Ponte, G., and Xu, L. (2021). Experimental analysis of local searches for sparse reflexive generalized inverses. *Journal of Global Optimization*, 81:1057–1093.

Fedorov, V. (1972). *Theory of optimal experiments*. Academic Press, New York-London. Translated from the Russian and edited by W. J. Studden and E. M. Klimko.

Golub, G. and Van Loan, C. (1996). *Matrix Computations (3rd Ed.)*. Johns Hopkins University Press, Baltimore, MD, USA.

Ko, C., Lee, J., and Wayne, K. (1998). Comparison of spectral and Hadamard bounds for D-optimality. In *MODA 5*, Contrib. Statist., p. 21–29. Physica, Heidelberg.

Lee, J. (2012). Maximum entropy sampling. In El-Shaarawi, A. and Piegorsch, W., editors, *Encyclopedia of Environmetrics, 2nd ed.*, p. 1570–1574. Wiley, Boston.

Lee, J. and Lind, J. (2020). Generalized maximum-entropy sampling. *INFOR: Information Systems and Operational Research*, 58(2):168–181.

Shewry, M. C. and Wynn, H. P. (1987). Maximum entropy sampling. *Journal of Applied Statistics*, 46:165–170.

Welch, W. J. (1982). Branch-and-bound search for experimental designs based on D-optimality and other criteria. *Technometrics*, 24(1):41–48.