We study the D-optimal Data Fusion (DDF) problem, which aims to select new data points, given an existing Fisher information matrix, so as to maximize the logarithm of the determinant of the overall Fisher information matrix. We show that the DDF problem is NP-hard and has no constant-factor polynomial-time approximation algorithm unless P = NP. Therefore, to solve the DDF problem effectively, we propose two convex integer-programming formulations and investigate their corresponding complementary and Lagrangian-dual problems. We also develop scalable randomized-sampling and local-search algorithms with provable performance guarantees. Leveraging the concavity of the objective functions in the two proposed formulations, we design an exact algorithm, aimed at solving the DDF problem to optimality. We further derive a family of submodular valid inequalities and optimality cuts, which can significantly enhance the algorithm performance. Finally, we test our algorithms using real-world data on the new phasor-measurement-units placement problem for modern power grids, considering the existing conventional sensors. Our numerical study demonstrates the efficiency of our exact algorithm and the scalability and high-quality outputs of our approximation algorithms.

Key words: Data fusion, Fisher information matrix, D-optimality, maximum-entropy sampling, approximation algorithm, exact algorithm, submodular inequality, optimality cut

1. Introduction

We study the following D-optimal Data Fusion problem

\[ z^* := \max_{S \subseteq [n], |S| = s} \ l\det \left( C + \sum_{i \in S} a_i a_i^\top \right), \tag{DDF} \]

where l\det denotes the natural logarithm of the determinant, the positive-definite matrix \( C \in S^d_{++} \) is an existing Fisher Information Matrix (FIM), the columns \( \{a_i \in \mathbb{R}^d\}_{i \in [n]} \) of \( A \in \mathbb{R}^{d \times n} \) represent \( n \) candidate \( d \)-dimensional data points to be selected, and the positive integer \( s \in [n] := \{1, 2, \ldots, n\} \) denotes the number of data points to be selected. In DDF, the FIM comprises two parts: \( C \) and \( \sum_{i \in S} a_i a_i^\top \), corresponding to the information obtained from existing data and new selected points, respectively. Therefore, the goal of DDF is to maximize the information gain by integrating new data points with conventional data from other sources. It is worth remarking that DDF differs
from the classic D-optimal design problem in the following aspects: (i) the conventional D-optimal design problem does not have the existing FIM $\mathbf{C}$; (ii) the D-optimal design problem typically assumes that $s \geq d$, while in DDF, it is possible that $s < d$. Hence, the existing results for D-optimal design in [SX20, MSTX19, NST19, PFL22] do not directly apply to DDF.

Below, we describe some interesting examples for which the proposed DDF can be applicable.

- **Sensor Fusion**: Modern sensor networks often involve multi-type sensors working collectively for a specific monitoring task (see [Var12, Zha95]). When installing new (possibly high-end) sensors, in order to achieve economical and effective operation, it is desirable to maximize the overall information obtained by fusing a small number of new sensors with the existing ones. Using the D-optimality criterion, widely-used in sensor placement, the corresponding optimal sensor-fusion problem is equivalent to DDF. In particular, $\mathbf{C} \in \mathbb{S}^d_+$ represents the FIM of the existing sensors installed at an $n$-node network. The additional FIM components introduced by new sensors can be shown to be equivalent to adding their corresponding rank-one matrices into the existing FIM (see, e.g., [LNI11, KG11, YKBB13]). In this problem, the dimension $d$ of sensor measurements is equal to the number of sensor locations, i.e., $d = n$, and we must have $s \leq d = n$. We test a sensor fusion problem in power systems in our numerical study section.

- **Active Learning**: In the big-data era, there are much more unlabeled data than labeled ones, where the latter are often expensive to acquire. Recently, researchers have been proactively working on active learning to select a small subset of unlabeled data points to label, with the hope of achieving a desired learning goal (e.g., classification accuracy) using fewer training data and/or achieving smaller labeling costs. Active learning enables the learners to iteratively select the most informative data points by exploiting labeled ones and exploring the unlabeled ones, which is also known as sequential experimental design in statistics. Thus, given that the FIM of the existing labeled data points is positive-definite, the D-optimal new-data selection problem can be formulated in the form of DDF (see [He09, LC21, MAL19, YBT06]). At each active learning iteration, the number of selected unlabeled data points is often much smaller than the dimension, especially for high-dimensional data, and can even be set to one for the sake of computational efficiency and stable convergence (see [WP17]).

- **Regularized D-optimal Design**: The regularized experimental design that arises from the linear models with a regularization penalty has been recently studied in [Tan20, MAL19], which admits the same form as our DDF. For regularized D-optimal design, its FIM often involves an additional positive-definite matrix $\mathbf{C}$ (e.g., an identity matrix in ridge-regularized D-optimal design).

### 1.1. Maximum-Entropy Sampling Problem (MESP)

DDF is closely related to the maximum-entropy sampling problem

$$\max_{S \subseteq [n], |S| = s} \det \mathbf{C}_{S,S} \quad \text{(MESP)}$$

where $\mathbf{C} \in \mathbb{S}^n_+$ has rank at least $s$, and $\mathbf{C}_{S,S}$ denotes the principal submatrix of $\mathbf{C}$ indexed by $S$.

Since [SW87], MESP has seen a variety of applications in statistics and information theory, as well as algorithmic advances (see [FL22a]). Below, we demonstrate that MESP is a special case of DDF with $n = d$ when the covariance matrix of MESP is positive-definite. This result implies that DDF is NP-hard (see [KLQ95]) and cannot be approximated by any polynomial-time algorithm within $\Omega(n \log(c))$ with some constant $c > 1$ unless $P = NP$ according to [CMI13]. Therefore, to efficiently solve DDF, this paper focuses on developing exact and near-optimal approximation algorithms.

In fact, as we will see below, DDF is also a special case of MESP.

**Theorem 1** For a covariance matrix $\mathbf{C} \in \mathbb{S}^n_{++}$, MESP is a special case of DDF.
Proof. Let \( F \) be the Cholesky factor of the positive semidefinite matrix \( (C/\lambda_{\min}(C) - I_n) \). Then, \( C/\lambda_{\min}(C) = I_n + F^\top F \), and for any \( S \subseteq [n], |S| = s \), we can verify that
\[
\text{ldet} C_{S,S} = \text{ldet} ((C/\lambda_{\min}(C))_{S,S}) - s \log(1/\lambda_{\min}(C))
\]
\[
= \text{ldet} ((I_n + F^\top F)_{S,S}) - s \log(1/\lambda_{\min}(C))
\]
\[
= \text{ldet} (I_n + \sum_{i=1}^n x_i f_i f_i^\top) - s \log(1/\lambda_{\min}(C)),
\]
where \( f_i \in \mathbb{R}^n \) is the \( i \)-th column of \( F \) and \( x \) is the 0/1 characteristic vector of \( S \subseteq [n] \), i.e., \( x_j = 1 \) if \( j \in S \); \( x_j = 0 \) if \( j \in [n] \setminus S \). The result follows because the first term in (1) is the objective function of MESP for the covariance matrix \( C \), and the last term is the objective function of a particular DDF added to a constant. \( \square \)

**Theorem 2** DDF is a special case of MESP.

Proof. Let \( C \in \mathbb{S}_+^d \) and \( a_i \in \mathbb{R}^d \), for \( i \in [n] \), be the input data for DDF. Let \( C^{1/2} \) be the square root of \( C \) and \( B := \{b_1, \ldots, b_n\} \in \mathbb{R}^d \times n \), where \( b_i := C^{-\frac{1}{2}} a_i \), for \( i \in [n] \). Then, for any \( S \subseteq [n], |S| = s \), we can verify that
\[
\text{ldet} \left( C + \sum_{i \in S} a_i a_i^\top \right) = \text{ldet} C + \text{ldet} (I_d + \sum_{i \in S} C^{-\frac{1}{2}} a_i a_i^\top C^{-\frac{1}{2}})
\]
\[
= \text{ldet} C + \text{ldet} (I_d + \sum_{i \in S} b_i b_i^\top)
\]
\[
= \text{ldet} C + \text{ldet} (I_n + B^\top B)_{S,S}.
\]
The result follows because the first term in (2) is the objective function of DDF for the existing FIM \( C \), set \( \{a_i \in \mathbb{R}^d\}_{i \in [n]} \) of \( n \) candidate data points, and \( s \) number of data points to be selected; and the last term in (2) is the objective function of a particular MESP added to a constant. \( \square \)

**1.2. Relevant literature**

In this subsection, we survey the relevant literature on exact and approximation algorithm for solving DDF or its variants including the regularized D-optimal design and MESP.

**Exact Algorithms:** As a special case of DDF, [He09] and [MAL19] studied the regularized D-optimal design with number of selected data being \( s = 1 \) and derived a closed-form optimal solution by using the formula for the determinant of a rank-one change to a symmetric matrix. A series of research works aimed at solving MESP, another special case of DDF, to optimality by a branch-and-bound algorithm have been conducted in [KLQ95, Lee98, AFLW96, AFLW99, LW03, HLW01, AL04, BL07, Ans18, Ans20, CFL21, CFL22], [CHB08, CHB13] and also [FL22b] instead considered outer-approximation approaches, and [LX20a] developed a branch-and-cut algorithm based on (sub)gradient inequalities. In contrast to previous approaches, we propose a new algorithm by deriving (sub)gradient inequalities and submodular inequalities from two equivalent convex integer-programming formulations of DDF, and exploring probing techniques to obtain optimality cuts. The submodular inequalities and the optimality cuts are numerically demonstrated to significantly strengthen the exact algorithm proposed, which is an enhancement of the LP/NLP branch-and-bound algorithm from [QG92].

**Approximation Algorithms:** Besides exact algorithms, more scalable yet effective approximation algorithms have also attracted attention, such as greedy, local-search, and randomized-sampling algorithms. Although the objective function of DDF is submodular and non-decreasing, it can be negative. Thus, the existing performance guarantees (e.g., \((1 - 1/e)\)-approximation ratio) for nonnegative non-decreasing submodular maximization cannot be directly applied [NWF78]. On the other hand, it has been shown that the randomized-sampling and local-search algorithms can be successfully applied to the D-optimal design and MESP to generate provably near-optimal solutions [LX20a, MSTX19, SX20, Nik15, NST19]. Motivated by these recent breakthroughs, we tailor these
two approximations algorithms to DDF, and we develop efficient implementations and theoretical guarantees. In particular, our approximation bounds of the sampling and local-search algorithms are invariant with respect to $s \leftrightarrow n - s$, outperforming the state-of-the-art ones [Nik15, LX20a] for MESP when $s > n/2$. The detailed comparison can be found in Section 3.

1.3. Summary of the organization and contributions

(i) We derive two convex integer-programming (CIP) formulations of DDF, termed R-DDF and M-DDF. We also study the Lagrangian-dual of their continuous relaxations and establish their optimality gaps.

(ii) We develop complementary formulations of R-DDF and M-DDF, based on the fact that DDF can be interpreted as excluding $n - s$ less informative data points from a cardinality-$n$ dataset. We show that R-DDF shares the same continuous relaxation value as that of its complement and M-DDF does not.

(iii) Exploring the two CIPs and the Lagrangian-dual of their continuous relaxations, we establish the theoretical performance guarantees of proposed local-search and randomize-sampling algorithms for solving DDF, as displayed in Table 1. We remark that each CIP provides us a different analysis of approximation bounds and some bounds in Table 1 are invariant with respect to $s$ and $n - s$, due to the complementary formulations.

(iv) Exploring the two CIPs, we succeed to derive closed-form (sub)gradient and submodular based valid inequalities, which are used on an enhancement of the LP/NLP branch-and-bound (B&B) algorithm proposed in [QG92], for solving DDF. In [MFR20] a numerical comparison is presented between the main algorithms in the literature for convex mixed-integer nonlinear-programming (MINLP), and the LP/NLP B&B presents an excellent performance in the comparison.

(v) We investigate probing techniques to derive optimality cuts to strengthen DDF from both primal and dual perspectives, where our probing schemes are effective and easy-to-implement, based on tight Lagrangian dual bounds and near-optimal approximation algorithms. Our numerical study confirms the effectiveness of the optimality cuts.

(vi) All the analyses and results can be extended to the regularized D-optimal design problem and MESP with a positive-definite sampling covariance matrix.

<table>
<thead>
<tr>
<th>Table 1 Approximation bounds for the local-search and sampling algorithms</th>
</tr>
</thead>
<tbody>
<tr>
<td>Local-Search Algorithm 1</td>
</tr>
<tr>
<td>Sampling Algorithm 2</td>
</tr>
</tbody>
</table>

$\bar{s} := \min\{s, n - s\}$; $\sigma_{\text{max}} := \max_{i \in [n]} a_i^\top C^{-1} a_i$; $\delta = \lambda_{\text{max}}(A^\top C^{-1} A)$

Notation: The following notation is used throughout the paper. We use bold lower-case letters (e.g., $x$) and bold upper-case letters (e.g., $X$) to denote vectors and matrices, respectively, and we use corresponding non-bold letters (e.g., $x_i$) to denote their components. We let $\mathbb{R}^n_+$ denote the set of all $n$-dimensional nonnegative vectors. Given positive integers $s < n$, we let $[n] := \{1, 2, \cdots, n\}$, $[s, n] := \{s, s + 1, \cdots, n\}$, and $\bar{s} := \min\{s, n - s\}$. Further, we let $\mathcal{Z}_s$ denote the collection of feasible solutions satisfying the cardinality constraint, i.e., $\mathcal{Z}_s := \{x \in \{0, 1\}^n : \sum_{i \in [n]} x_i = s\}$. We let $\mathcal{S}_s$ (resp., $\mathcal{S}_s^+$) denote the cone of $n \times n$ symmetric positive semidefinite (resp., definite) matrices. We let $C^\frac{1}{2}$ denote the square root of matrix $C$, i.e., $C^\frac{1}{2} C^\frac{1}{2} = C$. We let $C^\dagger$ denote the Moore-Penrose pseudo-inverse of $C$. We let $I_n$ denote the $n \times n$ identity matrix, and we let $e_i \in \mathbb{R}^n$ denote the $i$-th standard-unit vector. We let $\mathbf{1}$ denote a vector with all entries being 1. For $x \in \mathbb{R}^n$, we denote its support by $\text{supp}(x) \subseteq [n]$. We let $|S|$ denote the cardinality of a finite set $S$. Overloading the notation $\binom{n}{k}$ for the number of $k$-subsets of an $s$-element set, given a set $S$, we let $\binom{s}{k}$ denote
the collection of all the cardinality-$k$ subsets of $S$. Given an $m \times n$ matrix $X$ and two subsets $S \subseteq [m], T \subseteq [n]$, we let $X_{S,T}$ denote the submatrix of $X$ with rows and columns indexed by sets $S, T$, respectively, and we let $X_S$ denote the submatrix of $X$ with columns indexed by $S$. Given a symmetric matrix $X$, we let $\lambda_{\text{min}}(X), \lambda_{\text{max}}(X)$ denote the least and greatest eigenvalues of $X$, respectively. For $X, Y \in \mathbb{R}^{n \times n}$, we let $X \circ Y$ denote their Hadamard product, and $\text{tr} \, X$ denote the trace of $X$. Additional notation is introduced as needed.

Remark 3 Throughout the paper, considering the existing FIM $C$, the $n$ candidate $d$-dimensional data points $\{a_i \in \mathbb{R}^d\}_{i \in [n]}$ to be selected in DDF, and $A := [a_1, \ldots, a_n] \in \mathbb{R}^{d \times n}$, we use the definitions: $b_i := C^{-\frac{1}{2}} a_i$, for $i \in [n], B := [b_1, \ldots, b_n] \in \mathbb{R}^{d \times n}, q_i := (C + AA^\top)^{-\frac{1}{2}} a_i$, for $i \in [n]$; we let $V^\top V$ be the the Cholesky factorization of $I_n + B^\top B$, and let $v_i$ be the $i$-th column of $V$, for $i \in [n]$.

Organization: The remainder of the paper is organized as follows. In Section 2, we develop two CIP formulations for DDF, and their corresponding complementary problems and Lagrangian duals. In Section 3, we develop and analyze two approximation algorithms. In Section 4, we present our exact algorithmic approach with the introduction of valid submodular inequalities and optimality cuts. In Section 5, we present numerical results on a real-world application in power systems. Finally, Section 6 contains brief conclusions.

2. Two convex integer-programming formulations

Next, we present two convex integer-programming (CIP) formulations for DDF, termed R-DDF and M-DDF, as well as their complementary problems and the Lagrangian-dual of their relaxations.

2.1. First CIP formulation: R-DDF

To formulate DDF as a mathematical program, we introduce the binary variable $x_i = 1$, if the $i$-th data point is selected, and 0 otherwise, for each $i \in [n]$. Our first formulation of DDF is as follows.

**Proposition 4** DDF is equivalent to

$$z^* := \text{ldet} \, C + \max_{x \in \mathbb{Z}_n} \left\{ \text{ldet} \left( I_d + \sum_{i \in [n]} x_i b_i b_i^\top \right) \right\}. \quad \text{(R-DDF)}$$

**Proof.** Let $x$ be the 0/1 characteristic vector of $S \subseteq [n]$. Then, we have

$$\det \left( C + \sum_{i \in S} a_i a_i^\top \right) = \det \left( C + \sum_{i \in [n]} x_i a_i a_i^\top \right) = \det(C) \det \left( I_d + \sum_{i \in [n]} x_i b_i b_i^\top \right).$$

The result follows from the definition of $b_i$ (see Remark 3) and by taking the logarithm on both sides of the above equation. \qed

We refer to the problem as R-DDF because the objective function is similar to that of the regularized D-optimal design problem. The following result presents the Lagrangian dual of the continuous relaxation of R-DDF.

**Proposition 5** The Lagrangian dual of the continuous relaxation of R-DDF is

$$\bar{z}_R := \text{ldet} \, C + \min_{\Lambda \in \mathbb{S}^n_{++}, \nu, \mu \in \mathbb{R}^n_+} \left\{ -\text{ldet}(\Lambda) + \text{tr} \, \Lambda + s \nu + \sum_{i \in [n]} \mu_i - d : \Lambda b_i \leq \nu + \mu_i, \; i \in [n] \right\}. \quad (3)$$
Proof. First, we introduce an auxiliary matrix variable $X \in \mathbb{S}^d_{++}$ and reformulate R-DDF as

$$\text{lndet} C + \max_{x \in \mathbb{S}^d_{++}} \left\{ \text{lndet}(X) : I_d + \sum_{i \in [n]} x_i b_i b_i^\top \succeq X \right\}.$$ 

Then we derive the Lagrangian dual of the above maximization problem over $x, X$. Let $\Lambda \in \mathbb{S}^d_{++}$, $\nu \in \mathbb{R}$, $v \in \mathbb{R}^n_+$, $\mu \in \mathbb{R}^n_+$ denote the Lagrangian multipliers. The Lagrangian function $L$ is

$$L(x, X, \Lambda, \nu, v, \mu) = \text{lndet}(X) + \text{tr} \Lambda - \text{tr}(X \Lambda) + \sum_{i \in [n]} (b_i^\top \Lambda b_i - \nu + v_i - \mu_i) x_i + sv + \sum_{i \in [n]} \mu_i.$$ 

Maximizing $L$ over $(x, X)$ yields

$$\Lambda = X^{-1}, \ b_i^\top \Lambda b_i - \nu + v_i - \mu_i = 0, \forall i \in [n].$$ 

Then the Lagrangian dual problem can be obtained by plugging the above result into $L$, removing $v$, and minimizing $L$ over $(\Lambda, \nu, \mu)$.

We further derive the gradient, Hessian, and Lipschitz constant of the objective function of R-DDF, which allow us to use first- or second-order methods (e.g., Frank-Wolfe algorithm) to compute $\hat{z}_R$ with a proven convergence rate. We demonstrate the following result.

**Proposition 6** For any $x \in [0, 1]^n$, the gradient $g \in \mathbb{R}^n$ and the Hessian $H \in \mathbb{R}^{n \times n}$ of the objective function in the continuous relaxation of R-DDF are

$$g(x) := [b_1^\top X^{-1} b_1, \ldots, b_n^\top X^{-1} b_n], \quad H(x) := -(B^\top X^{-1} B) \circ (B^\top X^{-1} B), \text{ and } H(x) \succeq -\delta^2 I_n,$$

where $X := I_d + \sum_{i \in [n]} x_i b_i b_i^\top$ and $\delta := \lambda_{\max}(B^\top B)$.

Proof. For any $Y \in \mathbb{S}^d_{++}$, it is well-known that the function $\text{lndet}(Y)$ has first- and second-order derivatives: $Y^{-1}$ and $-Y^{-1} \partial Y Y^{-1}$. Thus, the gradient and Hessian of the objective function over $x$ in R-DDF can be derived.

According to [Joh74], the least eigenvalue of Hessian matrix $H(x)$ satisfies

$$\lambda_{\min}(H(x)) = -\lambda_{\max}(-H(x)) \geq -\lambda_{\max}^2(B^\top X^{-1} B) \geq -\lambda_{\max}^2(B^\top B),$$

where the last inequality is due to $X \succeq I_d$.

We adopt the well-known Frank-Wolfe algorithm to compute $\hat{z}_R$ in the numerical study. Due to Proposition 6 and [LX20a, Theorem 4], the Frank-Wolfe algorithm admits a convergence rate of $O(\delta^2 \lambda_{\max}^2(B^\top B)/\kappa)$, where $\kappa$ denotes the number of iterations.

An alternative interpretation of DDF is via excluding $n - s$ ineffective data points, which leads to the complementary formulation of R-DDF and the resulting Lagrangian dual problem below.

**Proposition 7** R-DDF is equivalent to

$$z^* = \text{lndet}(C + AA^\top) + \max_{x \in \mathbb{Z}_{n-s}} \left\{ \text{lndet} \left( I_d - \sum_{i \in [n]} x_i q_i q_i^\top \right) \right\}, \quad \text{(R-DDF-comp)}$$

and the Lagrangian dual of the continuous relaxation of R-DDF-comp is

$$\hat{z}_R = \text{lndet}(C + AA^\top) + \min_{\Lambda \in \mathbb{S}^d_{++}, \nu, \mu \in \mathbb{R}^n_+} \left\{ -\text{lndet} \Lambda + \text{tr} \Lambda + (n-s)\nu + \sum_{i \in [n]} \mu_i - d : -q_i^\top \Lambda q_i \leq \nu + \mu_i, \ i \in [n] \right\}. \quad (4)$$
Proof. For any \( x \in \mathbb{Z}_s \), the objective function of R-DDF can be written as

\[
\text{ldet} \left( C + \sum_{i \in [n]} x_i a_i a_i^\top \right) = \text{ldet} \left( C + AA^\top - \sum_{i \in [n]} (1 - x_i) a_i a_i^\top \right)
\]
\[
= \text{ldet}(C + AA^\top) + \text{ldet} \left( I_d - \sum_{i \in [n]} (1 - x_i)(C + AA^\top)^{-\frac{1}{2}} a_i a_i^\top (C + AA^\top)^{-\frac{1}{2}} \right).
\]

Then, replacing variable \( x \) by \( 1 - x \) and considering the definition of \( q_i \) (see Remark 3), we arrive at the equivalent formulation of R-DDF given by R-DDF-comp.

Following the similar derivation as Proposition 5, we can also derive the Lagrangian dual of the continuous relaxation of R-DDF-comp.

We remark that (i) the greatest eigenvalue of \( P \in \mathbb{S}_+^n \) is strictly less than one, so the matrix \( I_d - \sum_{i \in [n]} q_i q_i^\top \) in R-DDF-comp is always positive-definite and the objective value is finite; (ii) the continuous relaxation value of R-DDF does not vary with the complementary transformation, but considering the two Lagrangian dual problems (3) and (4) together, we obtain an approximation bound that is better than by considering only one of the Lagrangian duals, as discussed in Section 3.

### 2.2. Second CIP formulation: M-DDF

#### Lemma 8
\([\text{Nik15}, \text{Lemma 13}]\)

Let \( \lambda \in \mathbb{R}_+^n \) with \( \lambda_1 \geq \lambda_2 \geq \cdots \geq \lambda_n \), and let \( 0 < s \leq n \). There exists a unique integer \( k \), with \( 0 \leq k < s \), such that

\[
\lambda_k > \frac{1}{s-k} \sum_{i \in [k+1,n]} \lambda_i \geq \lambda_{k+1},
\]

with the convention \( \lambda_0 = +\infty \).

#### Lemma 9
Let \( \lambda \in \mathbb{R}_+^n \) with \( \lambda_1 \geq \lambda_2 \geq \cdots \geq \lambda_t > \lambda_{t+1} = \cdots = \lambda_s = \cdots = \lambda_n = 0 \). Then, the \( k \) satisfying (5) is precisely \( t \).

Proof. The result is immediate because

\[
\frac{1}{s-t} \sum_{i \in [t+1,n]} \lambda_i = \lambda_{t+1}.
\]

#### Definition 10
\([\text{Nik15}]\)

For a matrix \( X \in \mathbb{S}_+^n \) with eigenvalues \( \lambda_1 \geq \cdots \geq \lambda_n \geq 0 \), we let

\[
f(X) = \prod_{i \in [k]} \lambda_i \times \left( \frac{1}{s-k} \sum_{i \in [k+1,n]} \lambda_i \right)^{s-k},
\]

where \( k < s \) is unique non-negative integer satisfying (5).

It has been shown in [Nik15] that \( f(\cdot) \) is a concave function. With the notation above, we are ready to define M-DDF.

#### Theorem 11
DDF is equivalent to the following CIP formulation

\[
z^* = \text{ldet} C + \max_{x \in \mathbb{Z}_s} \left\{ \log f \left( \sum_{i \in [n]} x_i v_i v_i^\top \right) \right\}.
\]

(M-DDF)
Proof. It is sufficient to prove that the objective functions of M-DDF and R-DDF are equal for any $x \in \mathbb{Z}_n$. So, let $S := \text{supp}(x)$ for an $x \in \mathbb{Z}_n$. Clearly $|S| = s$, and thus we have

$$
\det \left( I_d + \sum_{i \in [n]} x_i b_i b_i^\top \right) = \det(I_d + B_S B_S^\top) = \det(I_n + B_S^\top B_S) = \det(I_n + B^\top B)_{S,S}$$

$$= \det(V_S^\top V_S) = f \left( \sum_{i \in [n]} x_i v_i v_i^\top \right),$$

where the second equation is because $B_S B_S^\top$ and $B_S^\top B_S = (B^\top B)_{S,S}$ have the same non-zero eigenvalues, and the last one follows from the definition of $v_i$ (see Remark 3) and from Lemma 9. This completes the proof.

The matrix in the objective of M-DDF is of order $n$, while that of R-DDF is of order $d$. We may consider choosing between them in practice, based on the two parameters $n$ and $d$ and on the continuous-relaxation bounds.

**Proposition 12** ([LX20a]) The Lagrangian dual of the continuous relaxation of M-DDF is

$$
\tilde{z}_M := \text{ldet} C + \min_{\Lambda \in \mathbb{S}_{++}^s, \nu, \mu \in \mathbb{R}^n_+} \left\{ -\log \det \Lambda + s \nu + \sum_{i \in [n]} \mu_i - s : v_i^\top \Lambda v_i \leq \nu + \mu_i, \; i \in [n] \right\}, \tag{6}
$$

where the function $\det(\cdot)$ denotes the product of $s$ least eigenvalues.

We note that the continuous relaxation value $\tilde{z}_M$ of M-DDF can be computed efficiently via the Frank-Wolfe algorithm (see [LX20a]). Similar to R-DDF, M-DDF also admits a complementary formulation, which has been widely-studied in the MESP literature (e.g., see [AFLW96, AFLW99, Ans20, CFL21, CFL22, FL22a]).

According to the identity

$$
\det(I + B^\top B)_{S,S} = \det(I_n + B^\top B) \det((I_n + B^\top B)^{-1})_{[n] \setminus S, [n] \setminus S}
$$

for any subset $S \subseteq [n]$, see [JH85, Section 0.8.4], we can derive the complementary formulation for M-DDF and then the Lagrangian dual of its continuous relaxation:

**Proposition 13** M-DDF is equivalent to

$$
z^* = \text{ldet}(C + AA^\top) + \max_{x \in \mathbb{Z}_{n-s}} \left\{ f \left( \sum_{i \in [n]} x_i v_i v_i^\top \right) \right\}, \tag{M-DDF-comp}
$$

and the Lagrangian dual of the continuous relaxation of M-DDF-comp is

$$
\tilde{z}_M := \text{ldet}(C + AA^\top) + \min_{\Lambda \in \mathbb{S}_{++}^{n-s}, \nu, \mu \in \mathbb{R}^n_+} \left\{ -\log \det \Lambda + (n-s) \nu + \sum_{i \in [n]} \mu_i - (n-s) : v_i^\top \Lambda v_i \leq \nu + \mu_i, \; i \in [n] \right\}. \tag{7}
$$

Contrary to Proposition 7, we will observe in our numerical study in Section 5, that the Lagrangian dual problem (6) is not equivalent to (7). Furthermore, the complementary problem M-DDF-comp and the Lagrangian dual (7) motivate us to further improve the approximation bounds of the local-search and sampling algorithms in [LX20a], as shown in the next section.

The two alternative formulations R-DDF and M-DDF, together with their complementary problems, often provide us with distinct continuous-relaxation solutions, which can help improve the analyses of the approximation algorithms. Each formulation has its own advantage under different
circumstances. For example, if existing data contain more accurate information, i.e., if the existing FIM $C$ dominates the overall FIM matrix, then we recommend R-DDF because its continuous relaxation provides a tighter upper bound. On the other hand, if the information from new data points is more valuable, i.e., the effect of $C$ is negligible, then M-DDF tends to yield a stronger continuous relaxation bound. Our theoretical analyses and numerical study will further confirm these phenomena.

3. Two approximation algorithms for DDF

Motivated by our two CIP formulations of DDF, we investigate simple and scalable approximation algorithms (i.e., local-search and randomized-sampling algorithms) for providing near-optimal selections of the new data points.

3.1. A local-search algorithm

In this subsection, we study a local-search algorithm for DDF, which has been successfully applied to many combinatorial optimization problems (see, for example, [SX20, LX20a, LX20b]). The algorithm runs as follows: (i) first, we start with a cardinality-$s$ subset $\hat{S} \subseteq [n]$; (ii) next, we swap one element from the set $\hat{S}$ with one from the unchosen set $[n] \setminus \hat{S}$, and we update the chosen set if the swapping strictly increases the objective value; and (iii) the algorithm terminates when there is no improvement. Motivated by R-DDF, we provide an efficient implementation of the local-search algorithm, as shown in Algorithm 1, with time complexity of $O(d^2 s(n-s))$ at the for-loop (i.e., Steps 5-11). Specifically, at Step 6, the strict improvement $\det(I_d + X - b_i b_i^\top + b_j b_j^\top) > \det(I_d + X)$, can be efficiently computed as

$$\det(I_d + X - b_i b_i^\top + b_j b_j^\top) > \det(I_d + X)$$

$$\iff \det(I_d + X)(1 + b_j^\top \Lambda_b)\left[1 + b_j^\top (I_d + X - b_i b_i^\top)^{-1}b_j\right] > \det(I_d + X)$$

$$\iff b_j^\top \Lambda_b - b_i^\top \Lambda_b b_i^\top \Lambda_b + b_i^\top \Lambda_b b_i^\top \Lambda_b > b_i^\top \Lambda_b,$$

which follows from the Sherman–Morrison formula. The update of matrix $\Lambda$ at Step 8 also follows from Sherman–Morrison formula, to avoid calculations of inverses from scratch.

Algorithm 1 Local-Search Algorithm

1: **Input:** vectors $\{b_i \in \mathbb{R}^n\}_{i \in [n]}$ and a positive integer $s \in [n]$
2: Initialize a cardinality-$s$ subset $\hat{S} \subseteq [n]$, matrix $X := \sum_{i \in \hat{S}} b_i b_i^\top$, and matrix $\Lambda := (I_d + X)^{-1}$
3: Compute $b_j^\top \Lambda b_j$ for each $j \in [n]$ and $b_i^\top \Lambda b_i$ for each $i \in \hat{S}, j \in [n] \setminus \hat{S}$
4: do
5: for each pair $(i, j) \in \hat{S} \times ([n] \setminus \hat{S})$
6: if $b_j^\top \Lambda b_j - b_i^\top \Lambda b_i b_i^\top \Lambda b_j + (b_i^\top \Lambda b_i)^2 > b_i^\top \Lambda b_i$
7: Update $\hat{S} := \hat{S} \cup \{j\} \setminus \{i\}$ and $X := X - b_i b_i^\top + b_j b_j^\top$
8: Compute $\Lambda_i := \Lambda + \frac{\Lambda b_i b_i^\top \Lambda}{1 - b_i^\top \Lambda b_i}$, and update $\Lambda := \Lambda_i - \frac{\Lambda b_i b_i^\top \Lambda}{1 + b_j^\top \Lambda b_j}$
9: Update $b_i^\top \Lambda b_i$ for each $l \in [n]$, and update $b_i^\top \Lambda b_j$ for each $l' \in \hat{S}, j' \in [n] \setminus \hat{S}$
10: end if
11: end for
12: while there is still an improvement (i.e., Step 6 is true for some pair $(i, j)$)
13: **Output:** $\hat{S}$

We use the proposed Lagrangian-dual of R-DDF, M-DDF, and their complements R-DDF-comp, M-DDF-comp to theoretically guarantee the quality of the output of Algorithm 1 by constructing feasible dual solutions:
Theorem 14 Let $\tilde{S}$ denote the output of the local-search Algorithm 1, and let $\bar{s} := \min\{s, n - s\}$, then the set $\tilde{S}$ yields a $\min\{d \log(1 + (\bar{s}/d)\sigma_{\text{max}}^2/(1 + \sigma_{\text{max}})), \bar{s} \log(\bar{s})\}$-approximation bound for DDF, i.e.,

$$\det(C + \sum_{i \in \tilde{S}} a_i a_i^\top) \geq z^* - \min\left\{d \log\left(1 + \frac{\bar{s}\sigma_{\text{max}}^2}{d(1 + \sigma_{\text{max}})}\right), \bar{s} \log(\bar{s})\right\},$$

where the constant $\sigma_{\text{max}} := \max_{i \in [n]} a_i^\top C^{-1} a_i$.

Proof. The approximation bound attains the minimum of $d \log(1 + (\bar{s}/d)\sigma_{\text{max}}^2/(1 + \sigma_{\text{max}}))$ and $\bar{s} \log(\bar{s})$, where they are derived based on R-DDF, M-DDF, and their complementary problems, respectively. Exploring the local optimality of the output solution $\tilde{S}$, we can show that

$$z^* \leq \tilde{z}_R \leq \det(C + \det(I_d + \sum_{i \in \tilde{S}} b_i b_i^\top) + d \log\left(1 + \frac{\bar{s}\sigma_{\text{max}}^2}{d(1 + \sigma_{\text{max}})}\right),$$

$$z^* \leq \tilde{z}_M \leq \log\left\{\sum_{i \in \tilde{S}} v_i v_i^\top + \det(C) + s \min\{\log(s), \log(n - s - \frac{n}{s})\}\right\},$$

$$z^* \leq \tilde{z}_M^c \leq \log\left\{\sum_{i \in \tilde{S}} v_i v_i^\top + \det(C) + (n - s) \min\{\log(n - s), \log(s - \frac{n}{n-s})\}\right\}.$$

The detailed proof can be found in Appendix A.1. \qed

We make the following remarks concerning Theorem 14:

(i) Either approximation bound with the ‘min’ in Theorem 14 is invariant with $s$ and $n - s$ by leveraging the two CIPs and their complements;

(ii) The second approximation bound attains zero when $\bar{s} = 1$, implying that the output solution of the local-search Algorithm 1 is optimal, and when $s = 1$, we have $\tilde{z}_M = z^*$, and when $s = n - 1$, we have $\tilde{z}_M^c = z^*$, as summarized in Corollary 15;

(iii) The second approximation bound (i.e., $\bar{s} \log(\bar{s})$) of the local-search Algorithm 1 improves on the one for MESP (i.e., $s \log(s - (s - 1)/s(2s - n)_+))$ derived by [LX20a], when the covariance matrix of MESP is positive-definite. Figure 1 illustrates the comparisons of two approximation bounds (i.e., our new bound $\bar{s} \log(\bar{s})$ versus the existing bound $s \log(s - (s - 1)/s(2s - n)_+)$) with $n = 10, 50$, where the greatest improvement of our bound over that in [LX20a] is indicated by a black dashed line. We see that when $s > n/2$, our new bound provides the local-search Algorithm 1 with a tighter performance guarantee;

(iv) The first approximation bound involving the constant $\sigma_{\text{max}}$ will be discussed after Theorem 17, along with that of the sampling algorithm which is also derived based on R-DDF;

(v) Another side product of Theorem 14 is to provide DDF with the optimality gaps of three proposed Lagrangian dual bounds: $\tilde{z}_R$, $\tilde{z}_M$, and $\tilde{z}_M^c$, as presented in Corollary 16.

Corollary 15 When $s = 1$ and $s = n - 1$, we have $\tilde{z}_M = z^*$ and $\tilde{z}_M^c = z^*$, respectively. In both cases, the local-search Algorithm 1 returns an optimal solution.

Corollary 16 The continuous relaxation values $\tilde{z}_R$ of R-DDF, $\tilde{z}_M$ of M-DDF, and $\tilde{z}_M^c$ of the complementary M-DDF (M-DDF-comp) satisfy

$$z^* \leq \tilde{z}_R \leq z^* + n \log\left(1 + \frac{\bar{s}\sigma_{\text{max}}^2}{n(1 + \sigma_{\text{max}})}\right);$$

$$z^* \leq \tilde{z}_M \leq z^* + s \log\left(s - \frac{n}{n-s}(2s - n)_+\right);$$

$$z^* \leq \tilde{z}_M^c \leq z^* + (n - s) \log\left(n - s - \frac{n-s}{n-s}(n - 2s)_+\right).$$

Proof. The proof follows from that of Theorem 14. \qed

As mentioned before, R-DDF and its complement R-DDF-comp have the same continuous relaxation value, whereas M-DDF and its complement M-DDF-comp do not. This fact is captured by Corollary 16, where we demonstrate a symmetric optimality gap of $\tilde{z}_R$ and non-symmetric gaps of $\tilde{z}_M$ and $\tilde{z}_M^c$. 
3.2. A randomized-sampling algorithm

In this subsection, we study a randomized-sampling algorithm which relies on the optimal continuous-relaxation solutions of R-DDF, M-DDF, and M-DDF-comp. Given an optimal continuous-relaxation solution $b^x$ of either problem, our Algorithm 2 samples a cardinality-$s$ subset $S \subseteq [n]$ with appropriate probability.

**Algorithm 2** Sampling Algorithm

1. **Input:** An optimal solution $b^x \in [0, 1]^n$ of the continuous relaxation of R-DDF, M-DDF, or M-DDF-comp and a positive integer $s \in [n]$
2. For a cardinality-$s$ subset $S \subseteq [n]$, its probability to be chosen is $P[\tilde{S} = S] = \frac{\prod_{i \in S} \tilde{x}_i}{\sum_{S' \subseteq (n)} \prod_{i \in S'} \tilde{x}_i}$.
3. **Output:** Random set $\tilde{S}$

The sampling procedure has $O(n \log n)$ complexity, and its detailed efficient implementation can be found in [SX20, Section 3.1]. The approximation bound of the output of Algorithm 2 depends on the choice of the relaxation, as presented in Theorem 17.

**Theorem 17** Let $\tilde{x}_D, \tilde{x}_M, \tilde{x}_M^c$ denote optimal continuous-relaxation solutions of R-DDF, M-DDF, and M-DDF-comp, respectively. Suppose that Algorithm 2 generates random sets $\tilde{S}_D, \tilde{S}_M, \tilde{S}_M^c$ with $\tilde{x}_D, \tilde{x}_M, \tilde{x}_M^c$ as inputs, respectively, then we have

(i) $\log \mathbb{E} \left[ \det \left( C + \sum_{i \in \tilde{S}_D} a_i a_i^\top \right) \right] \geq z^* + n \log(x_{\min}) - n \log(1 + \delta)$, where $x_{\min} > 0$ is the least non-zero entry in $\tilde{x}_D$ and $\delta := \lambda_{\max}(A^\top C^{-1} A)$;

(ii) $\log \mathbb{E} \left[ \det \left( C + \sum_{i \in \tilde{S}_M} a_i a_i^\top \right) \right] \geq z^* - s \log \left( \frac{s}{n} \right) - \log \left( \binom{n}{s} \right)$;

(iii) $\log \mathbb{E} \left[ \det \left( C + \sum_{i \in \tilde{S}_M^c} a_i a_i^\top \right) \right] \geq z^* - (n - s) \log \left( \frac{n-s}{n} \right) - \log \left( \binom{n}{n-s} \right)$;

(iv) Algorithm 2 can be derandomized as a polynomial-time algorithm with the same performance guarantees.

We establish the approximation bounds for Algorithm 2, using the solutions of R-DDF, M-DDF, and M-DDF-comp. The detailed proof can be found in Appendix A.2. We further remark:
(i) The performance of Algorithm 2 depends on the quality of the continuous-relaxation solution \( \hat{x} \), i.e., a tighter continuous relaxation bound yields better sampling results. It can be also seen that the running time of Algorithm 2 is dependent of the dimensionality of the data, i.e., \( d \);

(ii) Using the optimal continuous-relaxation solutions from M-DDF and M-DDF-comp, the output from Algorithm 2 can be at most \( s \log(\bar{s}/n) + \log(\binom{n}{\bar{s}}) \) away from the optimal value (recall that \( \bar{s} := \min\{s, n-s\} \)), which improves the approximation bound \((s \log(\bar{s}/n) + \log(\binom{n}{\bar{s}}))\) of Algorithm 2 for solving MESP in \([LX20a]\). The comparison between these two bounds is displayed in Figure 2; and

(iii) Similar to the corollaries of Theorem 14, Theorem 17 also implies the optimality of Algorithm 2 for two special cases in Corollary 18 and alternative optimality gaps of the proposed continuous relaxation values in Corollary 19.

---

**Corollary 18** When \( s = 1 \) and \( s = n - 1 \), Algorithm 2 returns an optimal solution when using \( \hat{x}_M \) and \( \hat{x}^c_M \) as the input, respectively.

*Proof.* According to Theorem 17, if \( s = 1 \) and \( \hat{x}_M \) is used as the input, the approximation bound of Algorithm 2 is equal to 0. When \( s = n - 1 \), the same argument follows. \( \square \)

**Corollary 19** The continuous relaxation values \( \bar{z}_R \) of R-DDF, \( \bar{z}_M \) of M-DDF, and \( \bar{z}^c_M \) of M-DDF-comp satisfy

\[
\begin{align*}
z^* &\leq \bar{z}_R \leq z^* - n \log(x_{\min}) + (n - s) \log(1 + \delta); \\
z^* &\leq \bar{z}_M \leq z^* + s \log(\binom{n}{s}) + \log(\binom{n}{s}); \\
z^* &\leq \bar{z}^c_M \leq z^* + (n - s) \log(\binom{n-s}{s}) + \log(\binom{n}{n-s});
\end{align*}
\]

*Proof.* The proof follows directly from that of Theorem 17. \( \square \)

Note that the theoretical optimality gaps of the three continuous relaxation values in Corollary 16 and Corollary 19 are not comparable; thus, taking the minimum of both values yields a better optimality gap. Specifically, for the continuous relaxation value \( \bar{z}_R \), the two optimality gaps from Corollary 16 and Corollary 19 depend on parameters \( \sigma_{\max}, \delta, \) and \( x_{\min} \) and thus are not comparable. The explicit comparison of alternative optimality gaps for \( \bar{z}_M \) and \( \bar{z}^c_M \) with \( n = 10, 50 \) can be found in Figure 3, where the blue diamond line and red circle line represent the results in Corollary 16 and Corollary 19, respectively. We observe that (i) for either continuous relaxation value, the optimality
gap in Corollary 19 is tighter than that of Corollary 16, except one case; and (ii) if \( s \leq n/2 \), then \( \tilde{z}_M \) outperforms \( \tilde{z}_c \) and is a tighter upper bound for DDF.

Note that in Theorem 14 and Theorem 17, Corollary 16 and Corollary 19, the magnitudes of \( \sigma_{\text{max}} \) and \( \delta \) depend on the contained information difference of new and existing data. If the existing data are very informative, i.e., \( \sigma_{\text{max}} \) and \( \delta \) tend to be small, and the proposed approximation bounds based on R-DDF get tighter. Hence, for Algorithm 2, we recommend applying the continuous-relaxation solution of R-DDF if the existing data are more informative; otherwise, applying M-DDF and its complement.

4. Our exact algorithmic approach for DDF

To solve DDF to optimality, we propose an enhancement on the LP/NLP B&B algorithm proposed in [QG92]. We first formulate DDF as

\[
    z^* = \text{ldet} \ C + \max_{z \in \mathbb{R}, \ x \in \mathbb{Z}} \left\{ z : z \leq \text{ldet} \left( I_d + \sum_{i \in [n]} x_i b_i b_i^\top \right), z \leq f \left( \sum_{i \in [n]} x_i v_i v_i^\top \right) \right\}, \quad \text{(DDF-MINLP)}
\]

for which the optimal value of the continuous relaxation is the best bound for DDF given by the continuous relaxations of both formulations R-DDF and M-DDF.
Using the concavity of the objective functions of both formulations, LP/NLP B&B considers, for a given $S \subseteq [n]$, the following linear relaxations of the two nonlinear inequalities in DDF-MINLP:

\[
\text{(Linearization) } \quad z \leq \det [B(S)] - \sum_{i \in S} b_i^T [B(S)]^{-1} b_i + \sum_{i \in [n] \setminus S} \rho_i(N \setminus \{i\})(1 - x_i) + \sum_{i \in [n] \setminus S} \rho_i(S) x_i,
\]

\[
\text{(Submodular) } \quad z \leq f[V(S)] - \sum_{i \in S} v_i^T g(S) v_i + \sum_{i \in [n] \setminus S} v_i^T g(S) v_i x_i,
\]

where for any subset $T \subseteq [n]$, $g(T) := V^\top(T) + 1/\lambda_{\min}(V(T)) [I_d - V(T) V^\top(T)]$ is a subgradient of $f(\cdot)$ at $V(T) := \sum_{i \in T} v_i v_i^\top$, according to Proposition 2 in [LX20a].

Then, at iteration $\ell$ of LP/NLP B&B, a mixed-integer linear-programming (MILP) problem $M^\ell$ is solved. The so-called master problem $M^\ell$ is a relaxation of DDF-MINLP obtained by replacing the nonlinear inequalities in $z$ by the inequalities in (9), constructed for all $S$ in a given set $S^\ell$ of linearization points represented by elements of $\binom{[n]}{s}$. $M^\ell$ is solved by a branch-and-bound algorithm, and every time a feasible solution $\bar{x}$ of $M^\ell$ is obtained during the execution of the algorithm, supp($\bar{x}$) is included in $S^\ell$. In [MFR20], the authors highlight that LP/NLP B&B can be efficiently implemented in advanced MILP packages (e.g., CPLEX, Gurobi) with the use of lazy constraints and callback functions. As the set of linearization points increases at each iteration of LP/NLP B&B, the solution values of $M^\ell$ form a non-increasing sequence of upper bounds for DDF-MINLP and the algorithm stops when the difference between this upper bound and the best known lower bound is small enough (see [QG92]).

### 4.1. Submodular cuts

Next, we propose a first enhancement on LP/NLP B&B for DDF-MINLP. We note that, because the objective functions of R-DDF and M-DDF are monotone (due to (2)) and submodular (via the Hadamard-Fischer inequalities: see, for example, [JH85, Section 7.8, Problem 14]), according to the results in [WN99, AA11], the following submodular linear inequalities are valid for DDF-MINLP, for any $S \subseteq [n]$:

\[
\text{(Submodular) } \quad z \leq \det [B(S)] - \sum_{i \in S} \rho_i(N \setminus \{i\})(1 - x_i) + \sum_{i \in [n] \setminus S} \rho_i(S) x_i,
\]

where for any $T \subseteq [n]$ and $i \in [n] \setminus T$, we define the difference function $\rho_i(T) := \det [B(T) + b_i b_i^\top] - \det [B(T)]$ with $B(T) := I_d + \sum_{i \in T} b_i b_i^\top$.

Moreover, by using the identity for rank-one update of the determinant of a symmetric matrix, we have the closed-form expression for the difference function $\rho_i(T) = \log[1 + b_i^T [A(T)]^{-1} b_i]$. Similarly, for any subset $T \subseteq [n]$ and $i \in T$, we have $\rho_i(T \setminus \{i\}) = -\log[1 - b_i^T [A(T)]^{-1} b_i]$. Hence, the constraint coefficients involved with the difference functions can be easily computed.

Then, to tighten the MILP relaxation of DDF-MINLP at LP/NLP B&B, besides including the standard linearization inequalities (9) in $M^\ell$, we also include the submodular inequalities (10), for each set $S$ added to $S^\ell$.

### 4.2. Optimality cuts

We consider choosing one or multiple data points and fixing their corresponding binary variables in DDF-MINLP to either one or zero, and then probing the restricted DDF to derive effective optimality cuts on these binary variables, which can help significantly reduce the size of the feasible region of DDF while maintaining the optimal value. Specifically, suppose that sets $S^1, S^0 \subseteq [n]$
denote the index set of data points being selected (i.e., \( x_i = 1 \) for each \( i \in S^1 \)) and being discarded (i.e., \( x_i = 0 \) for each \( i \in S^0 \)), respectively. Then a restricted problem of DDF is defined as

\[
\text{(Restricted DDF)} \quad z(S^1, S^0) := \max_{S \subseteq [n] \setminus (S^1 \cup S^0), |S| = s - |S^1|} \left( C + \sum_{i \in S^1} a_i a_i + \sum_{i \in S} a_i a_i \right),
\]

where sets \( S^1, S^0 \) are disjoint and set \( S^1 \) is of cardinality no larger than \( s \).

Clearly, if \( S^1 = S^0 = \emptyset \), then \( z(S^1, S^0) = z^* \) and the formulation is equivalent to DDF. Otherwise, if \( z(S^1, S^0) < z^* \), then at least one constraint built on sets \( S^1, S^0 \) is violated by an optimal solution and we thus obtain an optimality cut for DDF-MINLP that cuts off a subset of sub-optimal solutions. This result is summarized below.

**Theorem 20** For any two disjoint sets \( S^1, S^0 \) with \( |S^1| \leq s \) and \( S^1 \cup S^0 \subseteq [n] \), if \( z(S^1, S^0) < z^* \), then at least one of the two inequalities below is an optimality cut of DDF-MINLP.

\[
\sum_{i \in S^1} x_i \leq |S^1| - 1, \quad \sum_{j \in S^0} x_j \geq 1, \quad \forall x \in \mathcal{Z}_s.
\]

**Proof.** Given an optimal solution \( x^* \) of DDF-MINLP, if \( x^* \) satisfies the constraints \( \sum_{i \in S^1} x_i^* = |S^1| \) and \( \sum_{j \in S^0} x_j^* = 0 \), then \( \text{supp}(x^*) \) will be feasible to the restricted problem (11) with the same objective value \( z^* \), which contradicts \( z(S^1, S^0) < z^* \). Therefore, \( x^* \) must violate at least one of the equality constraints above and using the fact that \( x^* \) is binary, we complete the proof. \( \square \)

For the result of Theorem 20, we remark that (i) if either \( S^1 \) or \( S^0 \) is empty, then the inequality in (12) based on the non-empty set must be an optimality cut; (ii) if one of sets \( S^1 \) and \( S^0 \) is singleton and the other one is empty, the inequalities in (12) recover the well-known variable-fixing ones \([FL10]\): and (iii) if both sets \( S^1 \) and \( S^0 \) are non-empty, the optimality cuts in (12) can be enforced via disjunctive programming. The optimality cuts are effective at reducing the search space and significantly improve the LP/NLP B&B algorithm as shown in our numerical results.

Albeit being effective, a common criticism of the probing technique in mixed-integer programming is its computation expense \([ABG^{+}20]\), e.g., the optimal values \( z(S^1, S^0) \) and \( z^* \) in Theorem 20 may not be easily computable. Motivated by our near-optimal approximation algorithms and strong Lagrangian dual bounds for DDF, a compromise is that if an upper bound for \( z(S^1, S^0) \) is less than a lower bound for \( z^* \) (denoted by \( z^b \)), then the conclusion in Theorem 20 holds. Besides, following the spirit of the two Lagrangian duals for the continuous relaxations of DDF in Section 2, the restricted problem (11) also admits two alternative upper bounds as follows, corresponding to Lagrangian dual problems (3) and (6), respectively.

\[
\hat{z}_R(S^1, S^0) := \log \det C + \min_{\Lambda \in S^1_{++}, \nu, \mu \in \mathbb{R}_{++}} \left\{ - \log \det \Lambda + \text{tr} \Lambda + \nu + \sum_{i \in [n]} \mu_i - d \right. \\
+ \left( \sum_{j \in S^1} (b_j^\top \Lambda b_j - \nu - \mu_j) \right) + \left( \sum_{i \in S^0} \mu_i : b_i^\top \Lambda b_i \leq \nu + \mu_i, \forall i \in [n] \setminus \{S^1 \cup S^0\} \right) \\
\hat{z}_M(S^1, S^0) := \log \det C + \min_{\Lambda \in S^1_{++}, \nu, \mu \in \mathbb{R}_{++}} \left\{ - \log \det \Lambda + s \nu + \sum_{i \in [n]} \mu_i - s \right. \\
+ \left( \sum_{j \in S^1} (v_j^\top \Lambda v_j - \nu - \mu_j) \right) + \left( \sum_{i \in S^0} \mu_i : v_i^\top \Lambda v_i \leq \nu + \mu_i, \forall i \in [n] \setminus \{S^1 \cup S^0\} \right) \right\}.
\]

We observe that for some appropriately selected sets \( S^1 \) and \( S^0 \), the Lagrangian dual bounds (13) can be smaller than the lower bound of DDF, i.e., \( \hat{z}_R(S^1, S^0) < z^b \) or \( \hat{z}_M(S^1, S^0) < z^b \). Our selection strategy of sets \( S^1 \) and \( S^0 \) is a unification of the primal and dual perspectives, with an aim of reducing values: \( \hat{z}_R(S^1, S^0) \) and \( \hat{z}_M(S^1, S^0) \), which is discussed below.
(i) **Primal:** Given an optimal solution \( \hat{x} \) to the continuous relaxation of R-DDF or M-DDF, we let \( S^1 \subseteq \{ i : \hat{x}_i \leq \xi_0, \forall i \in [n] \} \) with \( \xi_0 \in [0,1] \) being a positive number close to 0 and \( S^0 \subseteq \{ i : \hat{x}_i \geq \xi^1, \forall i \in [n] \} \) with \( \xi^1 \in [0,1] \) being close to 1. In this case, we expect a big reduction on the restricted Lagrangian dual bounds in (13), when compared to the unrestricted bounds \( \hat{z}_R \) and \( \hat{z}_I \). Our numerical experiments suggest that this selection strategy performs very well in exploring appropriate subset \( S^1 \) to construct an optimality cut.

(ii) **Dual:** Using Lagrangian dual formulations (3) and (6), we can ensure that the Lagrangian dual bounds of restricted DDF problem (11) decrease by at least a given threshold. According Lagrangian dual formulations in (13), given an optimal dual solution \( (\hat{\Lambda}, \hat{\nu}, \hat{\mu}) \) of (3) or (6), we can verify that the corresponding restricted Lagrangian dual bound achieves a reduction of at least \( \sum_{j \in S^1} (\hat{b}_j^T \hat{\Lambda} \hat{b}_j - \hat{\nu} - \hat{\mu}_j) - \sum_{l \in S^0} \hat{\mu}_l \) or \( \sum_{j \in S^1} (\hat{v}_j^T \hat{\Lambda} \hat{v}_j - \hat{\nu} - \hat{\mu}_j) - \sum_{l \in S^0} \hat{\mu}_l \), compared with the original optimal dual value (see Proposition 21 below). This inspires us to identify sets \( S^1 \) and \( S^0 \) satisfying \( \sum_{j \in S^1} (\hat{b}_j^T \hat{\Lambda} \hat{b}_j - \hat{\nu} - \hat{\mu}_j) < 0 \) or \( \sum_{j \in S^1} (\hat{v}_j^T \hat{\Lambda} \hat{v}_j - \hat{\nu} - \hat{\mu}_j) < 0 \), and \( \sum_{l \in S^0} \hat{\mu}_l > 0 \), such that each restricted Lagrangian dual problems in (13) yields a smaller upper bound for restricted DDF (11) than the original Lagrangian dual value. It is worth mentioning that we can warm-start the solution procedure of the restricted dual problems (13) by using the optimal solution \( (\hat{\Lambda}, \hat{\nu}, \hat{\mu}) \) of the original Lagrangian dual problem (3) or (6). We also observe in the numerical study that the dual selection strategy is good at exploring an appropriate subset \( S^0 \) to construct an optimality cut.

**Proposition 21** Let \( (\hat{\Lambda}, \hat{\nu}, \hat{\mu}) \) be a feasible solution of Lagrangian dual problem (3) (resp., (6)) with the objective value \( z^{ub} \). Then, \( (\hat{\Lambda}, \hat{\nu}, \hat{\mu}) \) is also feasible to its corresponding restricted Lagrangian dual problem in (13) and the resulting objective value is equal to

\[
z^{ub} + \sum_{j \in S^1} (\hat{b}_j^T \hat{\Lambda} \hat{b}_j - \hat{\nu} - \hat{\mu}_j) - \sum_{l \in S^0} \hat{\mu}_l \quad (\text{resp., } z^{ub} + \sum_{j \in S^1} (\hat{v}_j^T \hat{\Lambda} \hat{v}_j - \hat{\nu} - \hat{\mu}_j) - \sum_{l \in S^0} \hat{\mu}_l).}
\]

**Proof.** If \( (\hat{\Lambda}, \hat{\nu}, \hat{\mu}) \) is a feasible solution of Lagrangian dual problem (3), it is easy to check that it is also feasible to the first optimization problem in (13) whose objective value is \( z^{ub} + \sum_{j \in S^1} (\hat{b}_j^T \hat{\Lambda} \hat{b}_j - \hat{\nu} - \hat{\mu}_j) - \sum_{l \in S^0} \hat{\mu}_l \). Similarly, for the Lagrangian dual problem (6), the same result holds by replacing \( \hat{b}_i \) by \( \hat{v}_i \) for all \( i \in [n] \).

We note that (i) our selection strategies are easy-to-implement because each continuous relaxation of DDF and its corresponding Lagrangian dual problem can be efficiently solved by the primal-dual Frank-Wolfe algorithm with a sublinear rate of convergence, (ii) the primal and dual selection strategies do not dominate each other and are complementary as shown in our numerical experiments, and (iii) all the analyses and selection strategies can be directly extended to complementary formulations of DDF.

### 5. Numerical study on sensor fusion in power systems

In this section, we present a real-world sensor-fusion problem in power systems, which can be formulated as DDF. We tested the proposed formulations and algorithms with varying-scale instances. All the experiments were conducted in Python 3.6 with calls to Gurobi 9.0 on a PC with 2.8 GHz Intel Core i5 processor and 8G of memory. All times reported are wall-clock times.

In power systems, phasor measurement units (PMUs) are the most accurate and high-speed time-synchronized devices used to measure phasors of bus voltages and currents in an electric grid (see [Nuo01]). PMUs have broad applications, including state estimation, security assessment, system monitoring, and wide-area control (see [DLRCTP10, The17]). In particular, reliable state estimation is an essential component of managing a modern energy system, aiming at determining the true voltages at all buses, based on available measurements and information that consist of observed voltage angles, power flows, and injections (see [TVC+10]).
Before the advent of PMUs, the conventional sensors, including supervisory control and data acquisition (SCADA) meter readings, were widely used to perform state estimation in power systems (see [Mon00]). It is recognized that, in many cases, one barely equips a power grid with a sufficient number of PMUs to achieve the state estimation fully (see [ZCTP06]), due to the budget and resource constraints. Therefore, it is important to make an intelligent choice of fusing PMUs and SCADA when deciding the PMU locations out of non-reference buses, in order to collect maximum information to best improve the state estimation. According to the PMU and SCADA measurement model in [LNI11], for an $n$-bus power system, the overall Fisher information matrix (FIM) is defined as

$$C + \sum_{i \in S} \frac{1}{\sigma^2_i} e_i e_i^\top,$$

where matrix $C \in \mathbb{S}^{n \times n}$ denotes the FIM obtained from conventional sensors and is positive-definite, the set $S \subseteq [n]$ denotes the bus locations of installed PMUs, and for each $i \in [n]$, $\sigma_i$ denotes the standard variance of PMU measurements at $i$-th bus. The FIM (14) of the sensor fusion problem in power systems can reduce to the objective matrix in DDF by letting $a_i = 1/\sigma_i e_i$ for each $i \in [n]$ and $d = n$. When employing D-optimality as the information selection criterion of the sensor fusion problem, based on the FIM (14), it follows that DDF exactly formulates this sensor fusion problem (see [LNI11]).

### 5.1. A comparison of continuous relaxations: IEEE 118- and 300-bus instances

From our theoretical analysis, we see that a tight continuous-relaxation bound for DDF has an important role in the implementation of Algorithm 2 and the derivation of optimality cuts. Therefore, we first investigated the three continuous-relaxation bounds of R-DDF, M-DDF, and M-DDF-comp, i.e., $\hat{z}_R$, $\hat{z}_M$, and $\hat{z}^c_M$, respectively, using two IEEE benchmark instances with 118 and 300 buses (see [AKFFS09]) of the PMU placement problem that provide the matrix $C$ in DDF. To compare the three alternative continuous-relaxation values, we conducted a controlled experiment with respect to the PMU standard deviations $\{\sigma_i\}_{i \in [n]}$, where large and small PMU standard deviations separately represent the two cases where either the existing sensors or the new sensors are more accurate for state estimation.

We used the Frank-Wolfe algorithm to compute the three upper bounds on the optimal value of DDF. The computational results for the two instances are displayed in Figure 4 and Figure 5, where the optimality gap is equal to the difference between an upper bound and a lower bound for DDF returned by our local-search Algorithm 1. We note that the Frank-Wolfe algorithm and our local-search Algorithm 1 are very efficient, and their computational time is negligible (i.e., less than one minute), so we do not report them.

For the 118-bus instance (Figure 4), we consider cases where the number of installed PMUs $s \in \{10, 15, \ldots, 105\}$, for $n = 118$, in order to compare the upper bounds for a wide range of the parameter $s$. In Figure 4(a), we sample the PMU standard deviations $\{\sigma_i\}_{i \in [n]}$ as independent uniform random variables in the range $[0, 100]$, and the new sensors contribute less to the state estimation than the existing ones. We see that the continuous-relaxation value $\hat{z}_R$ is smaller than both $\hat{z}_M$ and $\hat{z}^c_M$ in most cases in Figure 4(a). By contrast, in Figure 4(b), we sample the PMU standard deviations as $n$ independent uniform random variables in the range $[0, 0.01]$, assuming new more accurate sensors. In this setting, we see that the continuous-relaxation values $\hat{z}_M$ and $\hat{z}^c_M$ are much smaller than $\hat{z}_R$, so we use a pair of vertical axes to illustrate their performance. The comparison results parallel our theoretical findings in Section 2.

The comparison of the three upper bounds is also illustrated in Figure 5 for the 300-bus instance and the conclusions are the same. Thus, both theoretical analyses and numerical comparisons in Figure 4 and Figure 5 demonstrate that the continuous relaxation of M-DDF is more stable and tighter than that of R-DDF when the new sensors are subject to smaller measurement variances.
In addition, we observe in both figures that $\tilde{z}_M$ tends to be stronger than $\tilde{z}_c^c$ if $s$ is small; otherwise $\tilde{z}_M$ is stronger. In practice, PMUs are much more accurate than other sensors (see [ZTW+19]), and the measurement error is controlled within the range of $[-0.6^\circ, 0.6^\circ]$; and the number of PMUs to be installed is usually small due to budget constraints, i.e., $s$ is small. Thus, we identify the continuous-relaxation value $\tilde{z}_M$ of M-DDF as a better upper bound for the PMU placement problem in power systems that will be used in the following numerical study, where we also set the PMU standard deviation to $0.02^\circ$, a known PMU standard error in the literature (see [ZMST10]).

![Figure 4](image1.png)
(a) Large PMU standard deviation
(b) Small PMU standard deviation

Figure 4  Comparison between continuous-relaxation bounds on IEEE 118-bus instance

![Figure 5](image2.png)
(a) Large PMU standard deviation
(b) Small PMU standard deviation

Figure 5  Comparison between continuous-relaxation bounds on IEEE 300-bus instance

5.2. Testing submodular and optimality cuts: IEEE 118- and 300-bus instances

Next, we tested the submodular and optimality cuts introduced in Section 4. For our experiments, the proposed primal and dual selection strategies of subsets $S^1$ and $S^0$ only depend on the continuous relaxation of M-DDF, and its Lagrangian dual, due to their strength for the power-system instances. Similarly, for the selected subsets and their resulting restricted problems (11), we only computed the Lagrangian dual value $\hat{\tilde{z}}_M(S^1, S^0)$ in (13) to check whether this value is less than a
lower bound returned by the local-search Algorithm 1. That is, for a pair \((S^1, S^0)\), we applied the Frank-Wolfe algorithm with a warm start to compute the restricted dual bound \(\tilde{z}_M(S^1, S^0)\), unless the optimality cuts based on \((S^1, S^0)\) could be directly obtained using Proposition 21.

We considered different settings to generate a pair \((S^1, S^0)\) to construct optimality cuts, including: (a) \(S^1 = \emptyset\), \(S^0\) is singleton; (b) \(S^1\) is singleton, \(S^0 = \emptyset\); (c) \(S^1 = \emptyset\), \(S^0\) has 2 elements; (d) \(S^1\) has 2 elements, \(S^0 = \emptyset\); and (e) both \(S^1\) and \(S^0\) are singletons. The subsets are set to be small due to the effectiveness of their corresponding optimality cuts and tightness of the continuous-relaxation bounds. Besides, except setting (e), as one of subsets is empty, we did not need auxiliary binary variables to construct the optimality cuts (12). We used both the primal and dual strategies for selecting the subsets \(S^1\) and \(S^0\), because each one has its own advantage.

The well-known variable-fixing technique is a combination of settings (a) and (b), thus it can be viewed as a special case of our optimality cuts. It was originated for MESP with [AFLW96, AFLW99] and has recently been applied to MESP in [Ans18, Ans20, CFL22]. These existing works do not consider optimizing the restricted Lagrangian dual problem to strengthen the upper bound, nor explore the subsets based on the primal continuous-relaxation solution. Figure 6 presents the number of fixed variables on two IEEE instances by using our strategy (i.e., the integration of settings (a) and (b)), compared to that of [CFL22]. Because both methods manage to fix \(s\) variables to one and the remaining \(n-s\) variables to zero within one second, when \(s \leq 4\) in 118-bus instance and \(s \leq 31\) in 300-bus instances, we do not display these results.

In Figure 6, we see that the number of variables fixed to one tends to steadily increase and the number of variables fixed to zero decreases as \(s\) increases. From Figure 6(a) and Figure 6(b), we observe that our primal selection strategy successfully finds several optimal variables equal to one, but the variable-fixing by [CFL22] fails to fix any variable to one in our test instances. It is worth mentioning that the variables being fixed to one can significantly reduce the problem size and represent the best buses in power systems for installation of new PMU sensors. In Figure 6(c) and Figure 6(d), our dual strategy slightly outperforms that of [CFL22] when fixing variables to zero, and we are able to fix 13 more variables compared to that of [CFL22] for some cases. Finally, we compare the time for both methods in Figure 6(e) and Figure 6(f). We see that the overall performance of our primal and dual strategies is better than that of [CFL22], and our method takes more time but is still negligible compared to what it takes to solve DDF to optimality.

For the other three settings (c), (d), and (e) that involve two binary variables, their number of optimality cuts and overall time are illustrated in Figure 7. We see that when \(s\) increases, the number of optimality cuts tends to increase, but as seen in Figure 6, the variable-fixing gets worse, implying the fact that optimality cuts can be complementary to each other. Thus, in order to ensure the effectiveness of optimality cuts in our algorithmic framework, we may need to try various types of optimality cuts.

5.3. Exact and approximation algorithms: IEEE 118- and 300-bus instances

We tested the two IEEE instances and found that with the benefit of submodular and optimality cuts, the LP/NLP B&B can efficiently solve DDF and approximation algorithms can find high-quality solutions within one minute.

In Table 2, we compare the time and the optimality gap of the LP/NLP B&B algorithm with and without submodular and optimality cuts. For LP/NLP B&B with submodular and optimality cuts, we also present the time to generate the optimality cuts. The time to generate the submodular cuts is negligible. We used the optimality cuts based on subsets from settings (a) and (b) (i.e., variable-fixing) to reduce the problem size of DDF, and we added the other optimality cuts from settings (c), (d), and (e) to the initial MILP relaxation \(M^0\) solved in LP/NLP B&B. Columns “\#a – \#e” present the number of optimality cuts corresponding to settings (a) – (e). For each test case, the time limit was set to four hours.

Just using submodular cuts, we can solve five more cases within the time limit, and even for the cases that are not solved to optimality, the MIPgap is dramatically reduced. When additionally,
optimality cuts are included, all cases are solved to optimality within the time limit. For the four cases that took a substantial amount of time using only submodular cuts, with optimality cuts we could solve them much faster. We can see that many variables were fixed for all test cases using optimality cuts based on settings (a) and (b). According to [LNI11], it is difficult to compute the optimal value of DDF for a power system with more than 100 buses. However, our LP/NLP B&B using the submodular and optimality cuts enables us to effectively solve these cases to optimality. More details on this experiment are in Table 5, in Appendix B.
Using the optimal values of DDF in Table 2, we evaluate the bounds given by the continuous relaxation of M-DDF and the performance of the approximation algorithms on the same testing cases. The computational results are displayed in Table 3. The “gap” for M-DDF is the difference between the continuous relaxation value $\hat{z}_M$ and the optimal value $z^*$. The “gap” for the approximation algorithms is the difference between $z^*$ and the solution returned by them. For comparison purposes, we also tested the greedy algorithm for solving the PMU placement problem in power systems studied in [LNI11, LCW+12]. We feed the continuous-relaxation solution of M-DDF to Algorithm 2, and thus the time for Algorithm 2 includes that of solving the continuous relaxation. We see that the upper bound given by the solution of the continuous-relaxation of M-DDF is always close to the optimal value. We also see that Algorithm 1 consistently gives better times and gaps than Algorithm 2. Although Algorithm 1 and the greedy algorithm have the same gap for each instance, in our implementations Algorithm 1 is much faster due to the rank-one updating technique.

5.4. Scalability and near-optimality of approximation algorithms: IEEE 118-, 300-, and Polish 2383-bus instances

To better understand the overall performance of our approximation algorithms, we tested cases with a full range of selected data points (i.e., $s$) on the two IEEE instances and on a larger instance with 2383 buses, one of the largest power systems in the literature (see [ZMSG97]). Note that LP/NLP B&B is unable to solve many of the large-$s$ cases to optimality within a four-hour time limit. Thus, here we use the continuous-relaxation value $\hat{z}_M$ of M-DDF to evaluate the quality
Table 2  Impact of submodular and optimality cuts on LP/NLP B&B on IEEE instances

<table>
<thead>
<tr>
<th>n</th>
<th>s</th>
<th>LP/NLP B&amp;B</th>
<th>LP/NLP B&amp;B</th>
<th>LP/NLP B&amp;B</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td>+ submod. cuts</td>
<td>+ submod. and opt. cuts</td>
<td></td>
</tr>
<tr>
<td>MIPgap</td>
<td>time</td>
<td>MIPgap</td>
<td>time</td>
<td>MIPgap</td>
</tr>
<tr>
<td>118</td>
<td>5</td>
<td>0.00</td>
<td>11</td>
<td>0.00</td>
</tr>
<tr>
<td>118</td>
<td>10</td>
<td>0.00</td>
<td>423</td>
<td>0.00</td>
</tr>
<tr>
<td>118</td>
<td>15</td>
<td>3.18</td>
<td>-</td>
<td>0.00</td>
</tr>
<tr>
<td>118</td>
<td>16</td>
<td>4.27</td>
<td>-</td>
<td>0.00</td>
</tr>
<tr>
<td>118</td>
<td>17</td>
<td>5.19</td>
<td>-</td>
<td>0.61</td>
</tr>
<tr>
<td>118</td>
<td>18</td>
<td>4.81</td>
<td>-</td>
<td>1.34</td>
</tr>
<tr>
<td>118</td>
<td>19</td>
<td>8.47</td>
<td>-</td>
<td>2.16</td>
</tr>
<tr>
<td>118</td>
<td>20</td>
<td>11.11</td>
<td>-</td>
<td>2.20</td>
</tr>
<tr>
<td>300</td>
<td>35</td>
<td>7.76</td>
<td>-</td>
<td>0.00</td>
</tr>
<tr>
<td>300</td>
<td>40</td>
<td>83.52</td>
<td>-</td>
<td>0.00</td>
</tr>
<tr>
<td>300</td>
<td>45</td>
<td>48.06</td>
<td>-</td>
<td>0.00</td>
</tr>
<tr>
<td>300</td>
<td>50</td>
<td>52.67</td>
<td>-</td>
<td>0.88</td>
</tr>
<tr>
<td>300</td>
<td>51</td>
<td>40.41</td>
<td>-</td>
<td>0.90</td>
</tr>
<tr>
<td>300</td>
<td>52</td>
<td>140.05</td>
<td>-</td>
<td>2.44</td>
</tr>
<tr>
<td>300</td>
<td>53</td>
<td>186.13</td>
<td>-</td>
<td>3.27</td>
</tr>
<tr>
<td>300</td>
<td>54</td>
<td>168.61</td>
<td>-</td>
<td>3.55</td>
</tr>
<tr>
<td>300</td>
<td>55</td>
<td>158.67</td>
<td>-</td>
<td>4.60</td>
</tr>
<tr>
<td>300</td>
<td>56</td>
<td>143.81</td>
<td>-</td>
<td>5.29</td>
</tr>
<tr>
<td>300</td>
<td>57</td>
<td>129.59</td>
<td>-</td>
<td>4.57</td>
</tr>
</tbody>
</table>

1 MIPgap = upper bound − best feasible-solution value (both obtained by LP/NLP B&B)
2 total time in seconds; “-“: instance not solved within four hours
3 time to generate optimality cuts in seconds

of the feasible solutions obtained using the approximation algorithms. The value for “gap” in Table 4 and Figure 8 is equal to the difference between $z_M$ and the output values produced by the approximation algorithm.

We compare our proposed approximation algorithms with the greedy algorithm in Figure 8, for which we test 23 cases with number of installed PMUs from 5, 10, …, 115 and 29 cases with number of installed PMUs from 10, 20, …, 290 for the two IEEE instances with 118 and 300 buses, respectively. We see that Algorithm 1 clearly outperforms the other two, considering the gaps and times.

In Table 4, we present results where we tested a large-scale instance with 2383 buses in the Polish power system (see [ZMSG97]). The greedy algorithm is omitted in this table because it could not finish on these cases within four hours. On the other hand, our proposed approximation algorithms scale well. It is evident that Algorithm 1 performs very well in time and solution quality, dominating the performance of Algorithm 2 in both respects. Thus, we recommend using Algorithm 1 (with an efficient implementation) to solve practical PMU placement problems. We note that the gaps for Algorithm 1 could be quite small if we could compare to the optimal value. Another observation, is that the small gaps for Algorithm 1 establish the quality of the M-DDF relaxation on these cases.
### Table 3  Relaxation and approximation algorithms on IEEE instances

<table>
<thead>
<tr>
<th>n</th>
<th>s</th>
<th>M-DDF relaxation gap time</th>
<th>local-search Algorithm 1 gap time</th>
<th>sampling Algorithm 2 gap time</th>
<th>greedy algorithm gap time</th>
<th>time in seconds</th>
</tr>
</thead>
<tbody>
<tr>
<td>118</td>
<td>5</td>
<td>0.10</td>
<td>&lt;1</td>
<td>0.00</td>
<td>5</td>
<td>0.00 &lt; 1</td>
</tr>
<tr>
<td>118</td>
<td>10</td>
<td>0.16</td>
<td>&lt;1</td>
<td>0.00</td>
<td>5</td>
<td>0.00 &lt; 1</td>
</tr>
<tr>
<td>118</td>
<td>15</td>
<td>0.42</td>
<td>&lt;1</td>
<td>0.08</td>
<td>6</td>
<td>0.00 &lt; 1</td>
</tr>
<tr>
<td>118</td>
<td>16</td>
<td>0.48</td>
<td>&lt;1</td>
<td>0.32</td>
<td>6</td>
<td>0.00 &lt; 1</td>
</tr>
<tr>
<td>118</td>
<td>17</td>
<td>0.55</td>
<td>&lt;1</td>
<td>0.17</td>
<td>6</td>
<td>0.00 &lt; 1</td>
</tr>
<tr>
<td>118</td>
<td>18</td>
<td>0.57</td>
<td>&lt;1</td>
<td>0.23</td>
<td>7</td>
<td>0.00 &lt; 1</td>
</tr>
<tr>
<td>118</td>
<td>19</td>
<td>0.60</td>
<td>&lt;1</td>
<td>0.79</td>
<td>6</td>
<td>0.00 &lt; 1</td>
</tr>
<tr>
<td>118</td>
<td>20</td>
<td>0.64</td>
<td>&lt;1</td>
<td>0.87</td>
<td>6</td>
<td>0.00 &lt; 1</td>
</tr>
<tr>
<td>300</td>
<td>35</td>
<td>0.11</td>
<td>6</td>
<td>0.00</td>
<td>34</td>
<td>0.00 14</td>
</tr>
<tr>
<td>300</td>
<td>40</td>
<td>0.31</td>
<td>11</td>
<td>0.00</td>
<td>2.04</td>
<td>46</td>
</tr>
<tr>
<td>300</td>
<td>45</td>
<td>0.26</td>
<td>5</td>
<td>0.00</td>
<td>4.40</td>
<td>41</td>
</tr>
<tr>
<td>300</td>
<td>50</td>
<td>0.37</td>
<td>5</td>
<td>0.00</td>
<td>3.98</td>
<td>37</td>
</tr>
<tr>
<td>300</td>
<td>51</td>
<td>0.41</td>
<td>7</td>
<td>0.00</td>
<td>2.25</td>
<td>39</td>
</tr>
<tr>
<td>300</td>
<td>52</td>
<td>0.40</td>
<td>6</td>
<td>0.00</td>
<td>4.18</td>
<td>45</td>
</tr>
<tr>
<td>300</td>
<td>53</td>
<td>0.42</td>
<td>8</td>
<td>0.00</td>
<td>4.36</td>
<td>45</td>
</tr>
<tr>
<td>300</td>
<td>54</td>
<td>0.46</td>
<td>10</td>
<td>0.00</td>
<td>3.56</td>
<td>38</td>
</tr>
<tr>
<td>300</td>
<td>55</td>
<td>0.47</td>
<td>10</td>
<td>0.00</td>
<td>2.29</td>
<td>46</td>
</tr>
<tr>
<td>300</td>
<td>56</td>
<td>0.47</td>
<td>14</td>
<td>0.00</td>
<td>7.41</td>
<td>42</td>
</tr>
<tr>
<td>300</td>
<td>57</td>
<td>0.46</td>
<td>11</td>
<td>0.05</td>
<td>5.07</td>
<td>46</td>
</tr>
</tbody>
</table>

1 time in seconds; “< 1”: time less than one second

### Table 4  Relaxation and approximation algorithms on the Polish instance

<table>
<thead>
<tr>
<th>n</th>
<th>s</th>
<th>M-DDF relaxation $\bar{z}_M$ time</th>
<th>local-search Algorithm 1 gap time</th>
<th>sampling Algorithm 2 gap time</th>
<th>time in seconds</th>
</tr>
</thead>
<tbody>
<tr>
<td>2383</td>
<td>75</td>
<td>198.37</td>
<td>421</td>
<td>0.98</td>
<td>9</td>
</tr>
<tr>
<td>2383</td>
<td>100</td>
<td>614.00</td>
<td>545</td>
<td>1.24</td>
<td>17</td>
</tr>
<tr>
<td>2383</td>
<td>125</td>
<td>1024.55</td>
<td>608</td>
<td>1.43</td>
<td>17</td>
</tr>
<tr>
<td>2383</td>
<td>150</td>
<td>1430.34</td>
<td>666</td>
<td>1.93</td>
<td>17</td>
</tr>
<tr>
<td>2383</td>
<td>175</td>
<td>1831.40</td>
<td>807</td>
<td>2.58</td>
<td>25</td>
</tr>
<tr>
<td>2383</td>
<td>200</td>
<td>2227.53</td>
<td>794</td>
<td>3.25</td>
<td>28</td>
</tr>
<tr>
<td>2383</td>
<td>225</td>
<td>2620.43</td>
<td>1064</td>
<td>4.08</td>
<td>31</td>
</tr>
<tr>
<td>2383</td>
<td>250</td>
<td>3010.08</td>
<td>1301</td>
<td>4.95</td>
<td>37</td>
</tr>
<tr>
<td>2383</td>
<td>275</td>
<td>3396.66</td>
<td>1333</td>
<td>5.15</td>
<td>46</td>
</tr>
<tr>
<td>2383</td>
<td>300</td>
<td>3780.59</td>
<td>1244</td>
<td>5.41</td>
<td>47</td>
</tr>
<tr>
<td>2383</td>
<td>325</td>
<td>4162.01</td>
<td>1293</td>
<td>6.11</td>
<td>45</td>
</tr>
<tr>
<td>2383</td>
<td>350</td>
<td>4540.67</td>
<td>1392</td>
<td>6.92</td>
<td>49</td>
</tr>
<tr>
<td>2383</td>
<td>375</td>
<td>4916.61</td>
<td>1510</td>
<td>8.02</td>
<td>51</td>
</tr>
</tbody>
</table>

1 time in seconds
6. Conclusion

We studied the D-optimal data fusion problem, which can be of vital importance in many fields, such as monitoring, operation, planning, control, and decision making of various environmental, structural, agricultural, food processing, and manufacturing systems. The developed exact and approximation algorithms come with theoretical performance guarantees. Our numerical study confirms the efficacy of the proposed algorithms. We expect the proposed methods can be applicable to many machine learning problems under a cardinality constraint such as sparse PCA, sparse regression, sparse matrix completion, and so on.

Acknowledgments

Y. Li and W. Xie were supported in part by NSF grants 2046426 and 2153607. 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


Appendix A. Proofs

A.1 Proof of Theorem 14

Theorem 14 Let \( \hat{S} \) denote the output of the local-search Algorithm 1, and let \( \bar{s} := \min\{s, n - s\} \), then the set \( \hat{S} \) yields a \( \min\{d \log(1 + (\bar{s}/d)\sigma_{\max}^2/(1 + \sigma_{\max})), \bar{s} \log(\bar{s})\} \)-approximation bound for DDF, i.e.,

\[
\text{ldet}\left(C + \sum_{i \in \hat{S}} a_i a_i^\top\right) \geq z^* - \min \left\{ d \log \left(1 + \frac{\bar{s} \sigma_{\max}^2}{d(1 + \sigma_{\max})}\right), \bar{s} \log(\bar{s}) \right\},
\]

where the constant \( \sigma_{\max} := \max_{i \in [n]} a_i^\top C^{-1} a_i \).

Proof. We will prove the two approximation bounds (i) \( n \log(1 + (\bar{s}/n)\sigma_{\max}^2/(1 + \sigma_{\max})) \) and (ii) \( \bar{s} \log(\bar{s}) \), using R-DDF, M-DDF, and their complements, respectively.

(i) \( d \log(1 + (\bar{s}/d)\sigma_{\max}^2/(1 + \sigma_{\max})) \)-approximation bound.

Using the Lagrangian dual problem (3) of R-DDF (R-DDF), let us first show a non-symmetric approximation bound, \( d \log(1 + (\bar{s}/d)\sigma_{\max}^2/(1 + \sigma_{\max})) \).

Given the output \( \hat{S} \) of the local-search Algorithm 1, let \( X = \sum_{i \in \hat{S}} b_i b_i^\top \) and \( \Lambda = (I_d + X)^{-1}\).

Because \( \hat{S} \) is a locally-optimal solution of DDF, for any \( i \in \hat{S} \) and \( j \in [n] \setminus \hat{S} \), following the inequalities in (8), the local-optimality conditions can be written as

\[
b_j^\top \Lambda b_j - b_i^\top \Lambda b_i b_j^\top \Lambda b_j + b_i^\top \Lambda b_i b_j^\top \Lambda b_i \leq b_i^\top \Lambda b_i.
\]

Next, we will explore the inequality (15) to construct a feasible solution to the Lagrangian dual problem (3) of the R-DDF. For any \( i \in \hat{S} \) and \( j \in [n] \setminus \hat{S} \), by dropping the non-negative term \( b_j^\top \Lambda b_j b_j^\top \Lambda b_i \) in the left-hand side of inequality (8), we obtain

\[
b_j^\top \Lambda b_j \leq \frac{1}{1 - b_i^\top \Lambda b_i} b_i^\top \Lambda b_i.
\]

On the other hand, for each \( i \in \hat{S} \), we have

\[
b_i^\top \Lambda b_i \leq b_i^\top (I_d + b_i b_i^\top)^{-1} b_i = \frac{b_i^\top b_i}{1 + b_i^\top b_i} = \frac{a_i^\top C^{-1} a_i}{1 + a_i^\top C^{-1} a_i} \leq \frac{\sigma_{\max}}{1 + \sigma_{\max}},
\]

where the last inequality is due to \( \sigma_{\max} := \max_{i \in [n]} a_i^\top C^{-1} a_i \) and the fact that the function \( h(t) = t/(1 + t) \) is non-decreasing in \( t \in \mathbb{R}_+ \).

Combining the results (16) and (17), it follows that

\[
b_j^\top \Lambda b_j \leq \frac{1}{1 - b_i^\top \Lambda b_i} b_i^\top \Lambda b_i \leq (1 + \sigma_{\max}) b_i^\top \Lambda b_i, \forall i \in \hat{S}, \forall j \in [n] \setminus \hat{S},
\]

where the second inequality is due to non-decreasing of \( h(t) = 1/(1 - t) \) over \( t \in [0, 1] \).

Let us denote \( \beta_{\min} := \min_{i \in \hat{S}} b_i^\top \Lambda b_i \). Then, according to the inequality (18), a feasible solution to the Lagrangian dual problem (3) can be constructed by

\[
\hat{A} = t \Lambda, \hat{\nu} = t(1 + \sigma_{\max}) \beta_{\min}, \hat{\mu}_i = t(1 + \sigma_{\max}) b_i^\top \Lambda b_i - \hat{\nu}, \forall i \in \hat{S}, \hat{\mu}_i = 0, \forall i \in [n] \setminus \hat{S},
\]

where \( t > 0 \) is a scalar and will be specified later.

Plugging solution \((\hat{A}, \hat{\nu}, \hat{\mu})\) to problem (3), the objective value with the scalar \( t \) satisfies

\[
\hat{z}_t \leq \min_t \{ \text{ldet} C - \text{ldet} \Lambda - d \log(t) + t \text{tr} \Lambda + t(1 + \sigma_{\max}) \text{tr}(X \Lambda) - d\}
\]

\[
= \min_t \{ \text{ldet} C - \text{ldet} \Lambda - d \log(t) + td + t\sigma_{\max} \text{tr}(X \Lambda) - d\},
\]
where the equation is from the fact \( \text{tr}((I_d + X)\Lambda) = d \).

Minimizing the left-hand side above over \( t \), the optimal scalar is \( t^* = d/(d + \sigma_{\max}\text{tr}(X\Lambda)) \) and thus the final objective value with \( t^* \) of problem (3) is equal to

\[
z^* \leq \hat{z}_R \leq \text{ldet } C + \text{ldet } (I_d + X) + d\log \left(1 + \frac{\sigma_{\max}}{d} \frac{s\sigma_{\max}}{d + \sigma_{\max}}\right),
\]

where the first inequality is from Proposition 5 and the last one is due to inequality (17).

Now we use the Lagrangian dual problem (4) of R-DDF-comp to establish a complementary approximation bound of the previous one, \( d\log(1 + (n - s)/d\sigma_{\max}^2/(1 + \sigma_{\max})) \). Because the objective function of R-DDF-comp is \( \text{ldet}(I_d - \sum_{i \in [n] \setminus \hat{S}} q_i q_i^\top) \), for any \( i \in [n] \setminus \hat{S} \) and \( j \in \hat{S} \), the local-optimality condition becomes

\[
\text{det}(I_d + X + q_j q_j^\top - q_j q_j^\top) \leq q_j^\top \Lambda q_j - q_i^\top \Lambda q_i, q_i^\top \Lambda q_i \geq q_i^\top \Lambda q_i (1 - q_i^\top \Lambda q_i),
\]

where we define \( X := -\sum_{i \in [n] \setminus \hat{S}} q_i q_i^\top \) and \( \Lambda := (I_d + X)^{-1} \) here.

Then, by dropping the term \( q_j^\top \Lambda q_j, q_i^\top \Lambda q_i \) above, we obtain

\[
-q_j^\top \Lambda q_j \leq -q_i^\top \Lambda q_i, (1 - q_i^\top \Lambda q_i) \leq -q_i^\top \Lambda q_i \frac{1}{1 + \sigma_{\max}}, \forall i \in [n] \setminus \hat{S}, \forall j \in \hat{S},
\]

where the second inequality is from plugging the expressions of \( q_j \) and \( q_i \), i.e.,

\[
q_j \Lambda q_j = a_j^\top (C + AA^\top)^{-\frac{1}{2}} \left[ I_d - (C + AA^\top)^{-\frac{1}{2}} A_{[n] \setminus \hat{S}} A_{[n] \setminus \hat{S}}^\top (C + AA^\top)^{-\frac{1}{2}} \right]^{-1}
\]

\[
(C + AA^\top)^{-\frac{1}{2}} a_j = a_j^\top \left( C + AA^\top - A_{[n] \setminus \hat{S}} A_{[n] \setminus \hat{S}}^\top \right)^{-1} a_j \leq a_j^\top (C + a_j a_j^\top)^{-1} a_j \leq \frac{\sigma_{\max}}{1 + \sigma_{\max}}.
\]

Using inequality (19), we can construct a feasible solution to the Lagrangian dual problem (4) as below

\[
\hat{\Lambda} = t\Lambda, \hat{\nu} = t \frac{1}{1 + \sigma_{\max}} \beta_{\min}, \hat{\mu}_i = -t \frac{1}{1 + \sigma_{\max}} q_i^\top \Lambda q_i - \hat{\nu}, \forall i \in [n] \setminus \hat{S}, \hat{\mu}_i = 0, \forall i \in \hat{S},
\]

where \( \beta_{\min} := -\max_{i \in [n] \setminus \hat{S}} q_i^\top \Lambda q_i \).

Analogous to analyzing the previous approximation bound, we calculate the optimal scalar \( t^* = d/[d - \sigma_{\max}/(1 + \sigma_{\max}) \text{tr}(X\Lambda)] \) and plugging the optimal scalar, the objective value of problem (4) satisfies

\[
z^* \leq \hat{z}_R \leq \text{ldet } (C + AA^\top) + \text{ldet } (I_d + X) + d\log \left(1 - \frac{s\sigma_{\max}}{d(1 + \sigma_{\max})} \text{tr}(X\Lambda)\right)
\]

\[
= \text{ldet } (C + AA^\top) + \text{ldet } (I_d + X) + d\log \left(1 + \frac{\sigma_{\max}}{d(1 + \sigma_{\max})} \sum_{i \in [n] \setminus \hat{S}} q_i^\top \Lambda q_i\right)
\]

\[
\leq \text{ldet } (C + \sum_{i \in \hat{S}} a_i a_i^\top) + d\log \left(1 + \frac{(n - s)s\sigma_{\max}^2}{d(1 + \sigma_{\max})}\right),
\]

where the last inequality is because for any \( i \in [n] \setminus \hat{S} \), we have

\[
q_i \Lambda q_i = a_i^\top \left( C + AA^\top - A_{[n] \setminus \hat{S}} A_{[n] \setminus \hat{S}}^\top \right)^{-1} a_i \leq a_i^\top C^{-1} a_i \leq \sigma_{\max}.
\]

Minimizing the two approximation bounds gives us the symmetric one, \( d\log(1 + \frac{s\sigma_{\max}}{d(1 + \sigma_{\max})}) \).
(ii) $\tilde{s} \log(\bar{s})$-approximation bound.

According to M-DDF, the output value of the local-search Algorithm 1 becomes

$$z^* \leq \tilde{z}_M \leq \log f \left( \sum_{i \in S} \mathbf{v}_i \mathbf{v}_i^\top \right) + \ln det(C) + s \min \left\{ \log(s), \log \left( n - s - \frac{n}{s} + 2 \right) \right\},$$

where the inequality results from [Theorem 7]li2020best.

Using the complementary M-DDF, we can also show

$$z^* \leq \tilde{z}_M^c \leq \log f \left( \sum_{i \in S} \mathbf{v}_i \mathbf{v}_i^\top \right) + \ln det(C) + (n - s) \min \left\{ \log(n - s), \log \left( s - \frac{n}{n - s} + 2 \right) \right\}.$$

Taking the minimum of the two bounds above and using the identity $\log f(\sum_{i \in S} \mathbf{v}_i \mathbf{v}_i^\top) + \ln det(C) = \ln det(C + \sum_{i \in S} \mathbf{a}_i \mathbf{a}_i^\top)$ complete the proof. \hfill \Box

### A.2 Proof of Theorem 17

**Theorem 17** Let $\hat{x}_D, \hat{x}_M, \hat{x}_M^c$ denote optimal continuous-relaxation solutions of R-DDF, M-DDF, and M-DDF-comp, respectively. Suppose that Algorithm 2 generates random sets $\tilde{S}_D, \tilde{S}_M, \tilde{S}_M^c$ with $\hat{x}_D, \hat{x}_M, \hat{x}_M^c$ as inputs, respectively, then we have

(i) $\log E \left[ \det \left( C + \sum_{i \in \tilde{S}_D} \mathbf{a}_i \mathbf{a}_i^\top \right) \right] \geq z^* + n \log(x_{\min}) - (n - s) \log(1 + \delta)$, where $x_{\min} > 0$ is the least non-zero entry in $\hat{x}_D$ and $\delta := \lambda_{\max}(A^\top C^{-1}A)$;

(ii) $\log E \left[ \det \left( C + \sum_{i \in \tilde{S}_M} \mathbf{a}_i \mathbf{a}_i^\top \right) \right] \geq z^* - s \log \left( \frac{2}{n} \right) - \log \left( \binom{n}{s} \right)$;

(iii) $\log E \left[ \det \left( C + \sum_{i \in \tilde{S}_M^c} \mathbf{a}_i \mathbf{a}_i^\top \right) \right] \geq z^* - (n - s) \log \left( \frac{2}{n} \right) - \log \left( \binom{n}{n-s} \right)$;

(iv) Algorithm 2 can be derandomized as a polynomial-time algorithm with the same performance guarantees.

To facilitate the analysis of approximation bounds, let us first introduce the elementary symmetric polynomial function and its relation with eigenvalues.

**Definition 22 (elementary symmetric polynomial)** For a vector $\mathbf{y} \in \mathbb{R}^n$ and an integer $k \in [n]$, let $e_k(\mathbf{y})$ denote the degree $k$ elementary symmetric polynomial, i.e.,

$$e_k(\mathbf{y}) := \sum_{S \in \binom{[n]}{k}} \prod_{i \in S} y_i.$$

For any symmetric matrix $X \in \mathbb{S}^n$ with its eigenvalue vector $\mathbf{\lambda} \in \mathbb{R}^n$ and an integer $k \in [n]$, it is well-known that $e_k(\mathbf{\lambda})$ is equal to

$$e_k(\mathbf{\lambda}) = \sum_{S \in \binom{[n]}{k}} \det(X_{S,S}). \quad (20)$$

Now we are ready to prove Theorem 17.

**Proof.** The proof can be split into four parts, corresponding to three approximation bounds and derandomization.

**Part (i)** Let $T := \text{supp}(\hat{x}_D)$ be the index set of nonzero entries in $\hat{x}_D$.

For notational convenience, we define two matrices $X \in \mathbb{S}_+^n$ and $Y \in \mathbb{S}_+^n$ as

$$X := \text{Diag}(\hat{x}_D) + Y, \quad Y := \text{Diag}^{\frac{1}{2}}(\hat{x}_D)B^\top B \text{Diag}^{\frac{1}{2}}(\hat{x}_D).$$
For each $i \in [n] \setminus T$, we can show that the $i$-th column and row vectors of $X$ and $Y$ consist of zeros as $(\tilde{x}_D)_i = 0$. It follows that both $X$ and $Y$ have rank at most $|T|$. Thus, we let $\lambda_1 \geq \cdots \geq \lambda_{|T|} \geq 0 = \lambda_{|T|+1} = \cdots = \lambda_n$ and $\beta_1 \geq \cdots \geq \beta_{|T|} \geq 0 = \beta_{|T|+1} = \cdots = \beta_n$ denote the eigenvalues of $X$ and $Y$, respectively. Furthermore, according to Weyl’s inequalities and the fact that $0 < x_{min} \leq 1$, two eigenvalue vectors $\lambda$ and $\beta$ satisfy

$$\lambda_i \geq x_{min} + \beta_i \geq x_{min}(1 + \beta_i), \forall i = 1, \cdots, |T|.$$  

(21)

Next, the exponential expectation of the objective value of R-DDF is equal to

$$\mathbb{E} \left[ \det \left( I_d + \sum_{i \in \tilde{S}_D} b_i b_i^\top \right) \right] = \sum_{S \subseteq \{n\} \setminus T} \prod_{i \in S} \det (I_d + b_i b_i^\top)$$

$$= \frac{1}{\prod_{i \in \tilde{S}_D} (\tilde{x}_D)_i} \sum_{S \subseteq \{n\} \setminus T} \prod_{i \in S} \det (I_d + b_i b_i^\top)$$

$$= \frac{1}{e_s(\tilde{x}_D)} \sum_{S \subseteq \{n\} \setminus T} \det (X_S, S) = \frac{1}{e_s(\tilde{x}_D)} e_s(\lambda) = \frac{1}{e_s(\tilde{x}_D)} \lambda_i e_s \left( \frac{1}{\lambda_i} \right)$$

$$\geq \frac{1}{e_s(\tilde{x}_D)} \lambda_i \left( \frac{|T|}{s} \right) e_n \left( \frac{1}{\lambda_i} \beta \beta_1 \cdots \beta_{|T|} \right)$$

$$= \frac{1}{e_s(\tilde{x}_D)} \lambda_i \left( \frac{|T|}{s} \right) e_n \left( x_{min}(\beta_1 + 1), \cdots, x_{min}(1 + \beta_{|T|}), 1, \cdots, 1 \right)$$

$$\geq \frac{1}{e_s(\tilde{x}_D)} \lambda_i \left( \frac{|T|}{s} \right) x_{min}^n e_n (1 + \beta) = \left( \frac{|T|}{s} \right)^s \lambda_i x_{min}^n \det (I_d + Y)$$

$$= \left( \frac{|T|}{s} \right)^s \lambda_i \left( \frac{|T|}{s} \right) x_{min}^n \det (I_d + B \text{Diag}(\tilde{x}_D) B^\top)$$

$$\geq \left( \frac{|T|}{s} \right)^s x_{min}^n \lambda_i \left( \frac{|T|}{s} \right) \exp (z_R - \text{ldet} C)$$

$$\geq \left( \frac{|T|}{s} \right)^s x_{min}^n \lambda_i \left( \frac{|T|}{s} \right) \exp (z^* - \text{ldet} C),$$

where the first inequality stems from the fact that each element of vector $\frac{1}{\lambda_i} \lambda$ is no larger than 1, the sixth equality is due to $\lambda_i = 0$ for all $i \in [|T| + 1, n]$, the second inequality is from (21), the third inequality is obtained by Maclaurin’s inequality, the last equation is from the fact that matrices $Y$ and $B \text{Diag}(\tilde{x}_D) B^\top$ have the same nonzero eigenvalues, the fourth inequality is because $\tilde{x}_D$ is an optimal solution and $x_{min} \leq 1$, and the last inequality is due to $z_R \geq z^*$. Taking logarithm on both sides of the inequality above, we obtain

$$\text{ldet} C + \log \mathbb{E} \left[ \det \left( I_n + \sum_{i \in \tilde{S}_D} b_i b_i^\top \right) \right] \geq z^* + n \log(x_{min}) + (s - |T|) \log(\lambda_1) - s \log \left( \frac{s}{|T|} \right)$$

$$\geq z^* + n \log(x_{min}) - (n - s) \log(1 + \lambda_{max}(B^\top B)),$$

where the inequality is because $\lambda_1 \leq 1 + \beta_1 \leq 1 + \lambda_{max}(B^\top B)$.

**Part (ii)** The objective value led by the output $\tilde{S}_M$ of Algorithm 2 with $\tilde{x} = \tilde{x}_M$ is bounded by

$$\log \mathbb{E} \left[ f \left( \sum_{i \in \tilde{S}_M} v_i v_i^\top \right) \right] + \text{ldet} (C) \geq \tilde{z}_M - s \log \left( \frac{s}{n} \right) - \log \left( \frac{n}{s} \right),$$
where the inequality is from [LX20a, Theorem 5] and using the fact $\tilde{z}_M \geq z^*$, we obtain the approximation bound.

**Part (iii)** Given the output $\tilde{S}_M$ of Algorithm 2 with $\mathbf{z} = \mathbf{x}_M$, the objective value of Complementary DDF satisfies

$$\log E \left[ f \left( \sum_{i \in \tilde{S}_M} v_iv_i^\top \right) \right] + \operatorname{ldet}(C) \geq \tilde{z}_M - (n-s)\log \left( \frac{n-s}{n} \right) - \log \left( \binom{n}{n-s} \right),$$

where the inequality is from [LX20a, Theorem 5] and using the fact $\tilde{z}_M \geq z^*$, we obtain the approximation bound.

**Part (iv)** According to the proof of Theorem 11, for any cardinality-$s$ subset $S \subseteq [n]$, we must have

$$\det \left( C + \sum_{i \in S} a_i a_i^\top \right) = f \left( \sum_{i \in [n]} x_i v_i v_i^\top \right),$$

The remainder of the proof follows the de-randomization procedure of [LX20a, Thm. 6]. □

### Appendix B. Supplementary numerical results

As a supplement of Table 2, we present in Table 5 detailed numerical results for LP/NLP B&B with and without submodular cuts. The results expose the quality of the lower bound (LB) and the upper bound (UB) computed by LP/NLP B&B when the instances are not solved to optimality.

<table>
<thead>
<tr>
<th>$n$</th>
<th>$s$</th>
<th>$z^*$</th>
<th>LP/NLP B&amp;B</th>
<th>LP/NLP B&amp;B + submod. cuts</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td></td>
<td>LB</td>
<td>UB</td>
</tr>
<tr>
<td>118</td>
<td>5</td>
<td>80.15</td>
<td>80.15</td>
<td>80.15</td>
</tr>
<tr>
<td>118</td>
<td>10</td>
<td>156.90</td>
<td>156.90</td>
<td>156.90</td>
</tr>
<tr>
<td>118</td>
<td>15</td>
<td>231.63</td>
<td>230.68</td>
<td>233.86</td>
</tr>
<tr>
<td>118</td>
<td>16</td>
<td>246.31</td>
<td>244.65</td>
<td>248.92</td>
</tr>
<tr>
<td>118</td>
<td>17</td>
<td>260.94</td>
<td>259.77</td>
<td>264.96</td>
</tr>
<tr>
<td>118</td>
<td>18</td>
<td>275.56</td>
<td>275.01</td>
<td>279.82</td>
</tr>
<tr>
<td>118</td>
<td>19</td>
<td>290.15</td>
<td>287.34</td>
<td>295.82</td>
</tr>
<tr>
<td>118</td>
<td>20</td>
<td>304.69</td>
<td>300.83</td>
<td>311.94</td>
</tr>
<tr>
<td>300</td>
<td>35</td>
<td>367.49</td>
<td>360.53</td>
<td>368.29</td>
</tr>
<tr>
<td>300</td>
<td>40</td>
<td>404.02</td>
<td>326.62</td>
<td>410.14</td>
</tr>
<tr>
<td>300</td>
<td>45</td>
<td>439.81</td>
<td>394.85</td>
<td>442.91</td>
</tr>
<tr>
<td>300</td>
<td>50</td>
<td>474.49</td>
<td>425.78</td>
<td>478.45</td>
</tr>
<tr>
<td>300</td>
<td>51</td>
<td>481.24</td>
<td>444.43</td>
<td>484.84</td>
</tr>
<tr>
<td>300</td>
<td>52</td>
<td>487.98</td>
<td>364.09</td>
<td>504.14</td>
</tr>
<tr>
<td>300</td>
<td>53</td>
<td>494.66</td>
<td>323.89</td>
<td>510.02</td>
</tr>
<tr>
<td>300</td>
<td>54</td>
<td>501.27</td>
<td>345.42</td>
<td>514.02</td>
</tr>
<tr>
<td>300</td>
<td>55</td>
<td>507.84</td>
<td>360.22</td>
<td>518.89</td>
</tr>
<tr>
<td>300</td>
<td>56</td>
<td>514.37</td>
<td>381.03</td>
<td>524.84</td>
</tr>
<tr>
<td>300</td>
<td>57</td>
<td>520.89</td>
<td>400.04</td>
<td>529.63</td>
</tr>
</tbody>
</table>

$^1$ MIPgap = UB − LB

$^2$ time in seconds; “−”: instance not solved within four hours