A globally convergent difference-of-convex algorithmic framework and application to log-determinant optimization problems

Chaorui Yao*   Xin Jiang†

June 3, 2023

Abstract

The difference-of-convex algorithm (DCA) is a conceptually simple method for the minimization of (possibly) nonconvex functions that are expressed as the difference of two convex functions. At each iteration, DCA constructs a global overestimator of the objective and solves the resulting convex subproblem. Despite its conceptual simplicity, the theoretical understanding and algorithmic framework of DCA needs further investigation. In this paper, global convergence of DCA at a linear rate is established under an extended Polyak–Łojasiewicz condition. The proposed condition holds for a class of DC programs with a bounded, closed, and convex constraint set, for which global convergence of DCA cannot be covered by existing analyses. Moreover, the DCProx computational framework is proposed, in which the DCA subproblems are solved by a primal–dual proximal algorithm with Bregman distances. With a suitable choice of Bregman distances, DCProx has simple update rules with cheap per-iteration complexity. As an application, DCA is applied to several fundamental problems in network information theory, for which no existing numerical methods are able to compute the global optimum. For these problems, our analysis proves the global convergence of DCA, and more importantly, DCProx solves the DCA subproblems efficiently. Numerical experiments are conducted to verify the efficiency of DCProx.

1   Introduction

In this paper, we consider the difference-of-convex (DC) programming problems of the form

$$\begin{array}{ll} \text{minimize} & f(x) = g(x) - h(x) \\ \text{subject to} & x \in \mathcal{C}, \end{array} \tag{1}$$

where the functions g, h are convex and differentiable, and $\mathcal{C}$ is a bounded, convex set. This general problem covers a wide variety of applications in machine learning, information theory, statistics, and other fields [1–4]. A conceptually simple method for solving (1) takes the iterations

$$x^{(k+1)} \in \operatorname*{argmin}_{x \in \mathcal{C}} \left(g(x) - (h(x^{(k)}) + \langle \nabla h(x^{(k)}), x - x^{(k)} \rangle) \right); \tag{2}$$

i.e., one replaces h with its first-order Taylor expansion at $x^{(k)}$ and then solves the resulting convex subproblem. Due to the DC structure of (1), the iteration (2) will be referred to as the

---

*Department of Electrical and Computer Engineering, UCLA. Email: chaorui@ucla.edu.

†Department of Industrial and Systems Engineering, Lehigh University. Email: xjiang@lehigh.edu.

difference-of-convex algorithm (DCA). This paper studies the convergence theory of DCA and numerical methods for the subproblem in (2), and discusses applications of DCA in a class of matrix optimization problems involving log-determinant functions.

DC programming and DCA have been extensively studied since the late 1990s [1,2,5,6]. Classical convergence analyses for DCA establish asymptotic convergence to a first-order stationary point [2, 5], and convergence to a global optimum has not been investigated until very recently [7–9]. However, existing analyses for global convergence only consider the case where $\mathcal{C}$ is the entire space, and cannot handle closed and bounded constraint sets. Problems with such more complicated constraint sets arise from network information theory, functional analysis, statistics, *etc.*; see [4,7, 10]. For these problems, only the global minimizer is useful for practical reasons, but unfortunately global convergence of DCA has not been established for them.

Despite the limited global convergence guarantees, numerical methods for the subproblem in (2) are also lacking for the aforementioned applications. For these problems (and many others), the constraint set $\mathcal{C}$ is often expensive to project on, which prevents the direct use of first-order methods based on projections and gradient evaluations. Therefore, the structure of $\mathcal{C}$ (as well as the objective) should be further exploited, and customized subproblem solvers should be developed for the class of DC programs in the form (1).

In view of the above difficulties, this paper studies the conceptual DC algorithm from both theoretical and algorithmic aspects. From the theoretical angle, linear global convergence of DCA is established under a new set of conditions. The proposed assumption extends the recently proposed DC PL condition [9] to account for a bounded, closed, convex constraint set, and is shown to be satisfied in several classes of matrix optimization problems involving log-determinant functions.

In the computational phase, the DCProx algorithmic framework is developed to extend the scope of nonconvex problems numerically solvable by DCA. In DCProx, primal–dual proximal algorithms with Bregman distances are applied to exploit the subproblem structure at each DCA iteration, and solve those subproblems that are not suitable for projected gradient methods. With a suitable choice of Bregman distances, the inner iterations in DCProx are simple (Bregman) projections and have cheap per-iteration complexity. As an application, DCProx is applied to several fundamental problems in information theory. For these problems, DCProx is shown to converge to the global optimum, and has promising numerical performance.

The rest of the paper is organized as follows. Section 2 reviews the technical background and related work. The DCProx framework is developed in Section 3.1, and in Section 3.2, global convergence of DCA is established under a new set of conditions. Section 4 presents several applications, for which DCA converges to the global optimum and customized DCProx has simple update rules with cheap per-iteration complexity. Section 5 includes numerical results and Section 6 concludes the paper.

2 Background and related work

In this section, we discuss some background knowledge on the difference-of-convex algorithm (DCA) and primal–dual first-order proximal algorithms, along with the review on related work. These two methods are the basic ingredients of the presented algorithmic framework.

We define $[n] = \{1, 2, \ldots, n\}$ for $n \in \mathbb{N}$. We denote by $\mathbb{S}^n$ the set of symmetric $n \times n$ matrices, and by $\mathbb{S}^n_+$ the set of symmetric, positive semidefinite (PSD) $n \times n$ matrices. The inequality $X \succeq 0$ means that X is PSD. The notation $\langle x, y \rangle = x^T y$ is used to denote the inner product of vectors x

and y, and $\|x\|_2 = \langle x, x \rangle^{1/2}$ is the Euclidean norm. Other norms will be distinguished by subscripts. The convex conjugate of f is defined as $f^*(y) = \sup_x (\langle x, y \rangle - f(x))$.

2.1 The difference-of-convex algorithm (DCA)

DCA (2) can be viewed as a variant of the majorization–minimization (MM) algorithm [11] because at each iteration, it builds a global overestimator of the original problem and solves the resulting simpler subproblem. The constructed overestimator (known as *surrogate* in MM methods) is able to retain all the information from the convex component and only linearizes the concave portion. Extensions and variants of (2) exist for various kinds of problems. The current name DCA follows its original paper [12], and the same iteration (2) is also called the *concave–convex procedure* (CCCP) by different researchers; see, *e.g.*, [13–15]. The relation and difference between DCA and CCCP have been discussed in, *e.g.*, [1, 2], and in this paper, we use the term DCA to refer to the iteration (2) applied to the problem (1).

Asymptotic convergence of DCA to a first-order stationary point follows naturally from the analysis of MM methods [11]. Direct analyses of DCA also exist, with different assumptions on the problem; see, *e.g.*, [5, 16]. However, global convergence of DCA has not been studied until very recently [7–9], and is of particular interest here due to the nature of the applications studied in this paper. Global convergence of DCA can be established when the DC program (1) exhibits additional properties. For example, if the objective in (1) is geodesically convex on a manifold $\mathcal{M}$ and $\mathcal{C} \subseteq \mathcal{M}$, DCA (2) converges to a global optimum at a linear rate [7]. In addition, the classical Polyak–Łojasiewicz (PL) condition is also studied and extended for DC programming [8, 9]. A locally Lipschitz, continuously differentiable function f is said to satisfy the PL condition if there exists $\mu > 0$ such that

$$\mu(f(x) - f(x^\star)) \leq \tfrac{1}{2}\|\nabla f(x)\|_2^2, \quad \text{for all } x \in \mathbf{dom}\, f, \tag{3}$$

where $x^\star$ is a global minimizer of f. If $\mathcal{C} = \mathbb{R}^d$, g, h are locally Lipschitz continuous, and f satisfies the PL condition, then DCA (2) converges linearly to a global optimum [8]. The squared Euclidean distance in the classical PL condition can also be replaced with a Bregman distance (generated by g^* or h^*), yielding the so-called DC PL condition in [9]. Then linear global convergence is established for DCA if $\mathcal{C} = \mathbb{R}^d$, g, h are continuously differentiable, and f satisfies the DC PL condition [9].

2.2 Primal–dual proximal algorithms with Bregman distances

The design of proximal splitting algorithms has recently become an active research area, and proximal methods have been widely applied throughout science and engineering; see [17, 18] for recent surveys. In general, proximal methods iteratively decompose a large-scale optimization problem into smaller, simpler problems and then solve them separately. Thus, proximal methods are suitable for large-scale (or even huge-scale) convex optimization problems that cannot be handled by general-purpose solvers.

An important pillar for proximal algorithms is the (Euclidean) *proximal operator* (or *proximal mapping*) of a closed convex function $f \colon \mathbb{R}^d \to \mathbb{R}$, which is defined as

$$\mathbf{prox}_f(u) = \operatorname*{argmin}_x \left(f(x) + \tfrac{1}{2}\|x - u\|_2^2 \right). \tag{4}$$

3

The minimizer in the definition always exists and is unique for all u, as long as f is closed and convex [19]. The squared Euclidean distance used in (4) can also be replaced with a generalized distance d, of which an important example is the Bregman distance. Let ϕ be a convex function with a domain that has a nonempty interior, and assume ϕ is continuous on $\mathbf{dom}\,\phi$ and continuously differentiable on $\mathbf{int}(\mathbf{dom}\,\phi)$. The *Bregman distance* [20] generated by the *kernel function* ϕ is

$$d(x, y) = \phi(x) - \phi(y) - \langle \nabla \phi(y), x - y \rangle,$$

with $\mathbf{dom}\,d = \mathbf{dom}\,\phi \times \mathbf{int}(\mathbf{dom}\,\phi)$. The corresponding *Bregman proximal operator* is defined as:

$$\mathbf{prox}_f^\phi(u, a) = \underset{x}{\operatorname{argmin}} \left(f(x) + \langle a, x \rangle + d(x, u) \right).$$

It is assumed that the minimizer in the definition is unique and lies in $\mathbf{int}(\mathbf{dom}\,\phi)$ for all a and all $u \in \mathbf{int}(\mathbf{dom}\,\phi)$. In all the applications studied in this paper, the Bregman proximal operator is used with specific combinations of f and ϕ, and its existence and uniqueness follow directly from a closed-form solution. The use of Bregman distances can make the corresponding proximal operator easier to compute, and/or help build a more accurate local optimization model around the current iterate of a proximal algorithm [21]. We refer interested readers to [22–25] for more applications of Bregman proximal methods.

In this paper, proximal algorithms are applied to convex problems of the form

$$\text{minimize} \quad F(u) + G(\mathcal{A}u), \tag{5}$$

with optimization variable u, where F, G are closed convex functions and $\mathcal{A}$ is a linear mapping. A well-known proximal method for solving this type of problem is the Bregman primal–dual hybrid-gradient (PDHG) algorithm [26, 27]:

$$u^{(t+1)} = \mathbf{prox}_{\tau F}^{\phi_{\mathrm{p}}}(u^{(t)}, \tau \mathcal{A}^* v^{(t)}) \tag{6a}$$

$$v^{(t+1)} = \mathbf{prox}_{\sigma G^*}^{\phi_{\mathrm{d}}}(v^{(t)}, -\sigma \mathcal{A}(2u^{(t+1)} - u^{(t)})), \tag{6b}$$

where $\mathcal{A}^*$ is the adjoint operator of $\mathcal{A}$, and ϕ_{p} and ϕ_{d} are two kernel functions in the primal and dual spaces, respectively. It is assumed (without loss of generality) that ϕ_{p} and ϕ_{d} are 1-strongly convex with respect to norms $\|\cdot\|_{\mathrm{p}}$ and $\|\cdot\|_{\mathrm{d}}$:

$$d_{\mathrm{p}}(u, u') \geq \tfrac{1}{2}\|u - u'\|_{\mathrm{p}}^2, \qquad d_{\mathrm{d}}(v, v') \geq \tfrac{1}{2}\|v - v'\|_{\mathrm{d}}^2$$

for all $(u, u') \in \mathbf{dom}\,d_{\mathrm{p}}$ and $(v, v') \in \mathbf{dom}\,d_{\mathrm{d}}$. The stepsizes σ, τ must satisfy $\sigma\tau\|\mathcal{A}\|^2 \leq 1$. Here the operator norm $\|\mathcal{A}\|$ is defined as

$$\|\mathcal{A}\| = \sup_{u \neq 0, v \neq 0} \frac{\langle v, Au \rangle}{\|v\|_{\mathrm{d}}\|u\|_{\mathrm{p}}} = \sup_{u \neq 0} \frac{\|\mathcal{A}u\|_{\mathrm{d},*}}{\|u\|_{\mathrm{p}}} = \sup_{v \neq 0} \frac{\|\mathcal{A}^* v\|_{\mathrm{p},*}}{\|v\|_{\mathrm{d}}},$$

where $\|\cdot\|_{\mathrm{p},*}$ and $\|\cdot\|_{\mathrm{d},*}$ are the dual norm of $\|\cdot\|_{\mathrm{p}}$ and $\|\cdot\|_{\mathrm{d}}$, respectively. Line search techniques also exist for choosing adaptive stepsizes [21, 28, 29]. With suitable stepsizes, the iterates $(u^{(k)}, v^{(k)})$ converge to a primal and dual optimum of the problem (5), and the averaged iterates converge at a rate of $O(1/k)$ [21, 27, 30]. Successful applications of Bregman PDHG range from signal processing [25], optimal transport [31] to sparse semidefinite programming [29].

Algorithmic Framework 1 DCProx
1: **for** $k = 1, 2, \ldots,$ **do**
2:     **for** $t = 1, 2, \ldots,$ **do**
3:         Compute the Bregman PDHG updates:

$$u^{(t+1)} \leftarrow \underset{u \in \mathcal{C}_1}{\operatorname{argmin}} \, g(u) + \langle \mathcal{A}^* v^{(t)} - \nabla h(x^{(k)}), u \rangle + \tfrac{1}{\tau} d_{\mathrm{p}}(u, u^{(t)}) \tag{7a}$$

$$v^{(t+1)} \leftarrow \underset{v}{\operatorname{argmin}} \, \delta_{\mathcal{C}_2}^*(v) - \langle \mathcal{A}(2u^{(t+1)} - u^{(t)}), v \rangle + \tfrac{1}{\sigma} d_{\mathrm{d}}(v, v^{(t)}). \tag{7b}$$

4:     **end for**
5:     Update DCA iterate $x^{(k+1)} \leftarrow \hat{u}$, where $(\hat{u}, \hat{v})$ is the limit point of Bregman PDHG.
6: **end for**

3   DCProx algorithmic framework and global convergence of DCA

We now present our theoretical and algorithmic contributions to DCA. When DCA is applied to the problem (1) (with a nontrivial constraint set $\mathcal{C}$), global convergence cannot be obtained directly from existing results, and the subproblem (2) cannot be solved efficiently by first-order methods based on projections and gradient evaluations. However, when $\mathcal{C}$ exhibits special structure (*e.g.*, it is written as the intersection of simple convex sets), primal–dual proximal methods can be used to split the nontrivial set constraints and each (primal/dual) update becomes a simple (Bregman) projection. In addition, the recently proposed DC PL condition [9] is extended to account for a closed and bounded set $\mathcal{C}$, and global convergence is established for DCA when applied to (1).

3.1   The DCProx framework

Very often in applications, the constraint set $\mathcal{C}$ is nontrivial, and projection onto $\mathcal{C}$ is expensive. Thus, most projection-and-gradient-based first-order methods are not efficient in solving the subproblem in (2), and special structure of $\mathcal{C}$ should be leveraged to design a customized subproblem solver. In many applications, the set $\mathcal{C}$ can be naturally written as the intersection of two (or several) simple convex sets, and projection onto each component set is much easier than projection on $\mathcal{C}$. (This is true for all the applications in Section 4, and in particular, Section 4.3 provides an example in which $\mathcal{C}$ consists of an arbitrary number of component sets.) This motivates the use of primal–dual proximal algorithms (*e.g.*, PDHG [26, 27]) as the subproblem solver. For example, if $\mathcal{C} = \mathcal{C}_1 \cap \mathcal{C}_2$, where $\mathcal{C}_1$ and $\mathcal{C}_2$ are convex sets, the subproblem (2) can be reformulated in the form of (5) with

$$F(u) = g(u) - \langle \nabla h(x^{(k)}), u \rangle + \delta_{\mathcal{C}_1}(u), \qquad G(v) = \delta_{\mathcal{C}_2}(v), \qquad \mathcal{A} = \mathrm{Id},$$

where $\delta_{\mathcal{C}_i}$ is the indicator function of $\mathcal{C}_i$ and Id is the identity operator. Therefore, with Bregman PDHG (6) as the subproblem solver, the proposed DCProx algorithmic framework is summarized in Framework 1. (The term Prox indicates that proximal methods exploit a splitting of $\mathcal{C}$.) It is called an *algorithmic framework* rather than an algorithm because in principle, ϕ_{p} and ϕ_{d} could be any valid kernel functions. The choice of Bregman kernels largely depends on problem structure, and a suitable choice of kernels can significantly improve the efficiency of DCProx. The distance d_{d} in the dual space is often chosen as the squared Euclidean distance ($\phi_{\mathrm{d}} = \tfrac{1}{2} \| \cdot \|_2^2$), and then the dual

update (7b) simply involves the Euclidean projection on $\mathcal{C}_2$. Moreover, when we choose $\phi_{\mathrm{p}} = g$, the primal update (7a) reduces to

$$u^{(t+1)} = \underset{u \in \mathcal{C}_1}{\operatorname{argmin}} \, (1 + \tfrac{1}{\tau}) g(u) + \big\langle \tfrac{\tau}{1+\tau}(\mathcal{A}^* v^{(t)} - \nabla h(x^{(k)})) - \tfrac{1}{\tau} \nabla g(u^{(t)}), u \big\rangle$$

$$= \underset{u \in \mathcal{C}_1}{\operatorname{argmin}} \, d_g(u, \tilde{u}), \quad \text{where } \nabla g(\tilde{u}) = \tfrac{\tau}{1+\tau}(\mathcal{A}^* v^{(t)} - \nabla h(x^{(k)})) - \tfrac{1}{\tau} \nabla g(u^{(t)}).$$

With this choice of the kernel, the primal update is simply a Bregman projection on $\mathcal{C}_1$. (Note that $x^{(k)}$ is the kth DCA iterate while $(u^{(t)}, v^{(t)})$ is the tth PDHG iterate. The vector $\tilde{u}$ is used only for demonstration purpose, and is not computed explicitly in practice.)

3.2 Global convergence for DCA

While the presented DCProx framework makes DCA tractable for a wider range of DC problems (with a nontrivial convex constraint set), we show that for these problems, DCA converges to a global optimum at a linear rate. (Note that the presented analysis focuses on the DCA iterations (2), and is independent of the subproblem solver.) All the assumptions for (1) are listed below, and global convergence of DCA for this kind of problems are not addressed by existing analyses [7–9].

Assumption 1. The following assumptions are made for the problem (1).

1. The functions g, h are closed, convex, and continuously differentiable on their respective domains.

2. The constraint set $\mathcal{C}$ is bounded, closed, and convex, with $\mathcal{C} \subseteq \operatorname{\mathbf{dom}} g \cap \operatorname{\mathbf{dom}} h$.

3. The problem (1) has a global minimizer $x^\star$ with a finite optimal value $f^\star = f(x^\star)$.

4. The DCA iteration (2) is solvable for any point $x^{(k)} \in \mathcal{C}$.

5. There exists $\mu > 0$ such that for some $r > 0$,

$$\mu(f(x) - f^\star) \le d_{h^*}(\nabla g(x) + y, \nabla h(x)), \quad \text{for all } x \in \mathcal{C}, \ y \in N_\mathcal{C}(x) \text{ and } \|y\|_2 \le r \qquad (8)$$

where $N_\mathcal{C}(x)$ is the normal cone of $\mathcal{C}$ at x.

The differentiability of g and h implies that $\operatorname{\mathbf{dom}} f$ is an open set, and existing analyses [8,9] that (implicitly) assume $\mathcal{C} = \mathbb{R}^d$ cannot handle a closed and bounded constraint set. Assumption 1.4 is an extension of the recently proposed DC PL condition [9, Definition 1] and the classical PL condition. (A similar condition is proposed in [32] for the mirror descent method.) In particular, (8) replaces the squared Euclidean distance in the PL condition (3) with a Bregman distance, and carefully handles the closedness and boundedness of $\mathcal{C}$ that is often ignored in the DCA literature. The constant $r > 0$ is needed to bound $\|y\|_2$ when y is on the boundary of $\mathcal{C}$, and is used only for theoretical analysis. Assumption 1.4 might (implicitly) impose some requirements on the constant r, but its actual value does not need to be known in practice. With Assumption 1, we can prove a linear convergence rate on the function value.

Theorem 1. *With Assumption 1, the DCA iteration* (2) *satisfies*

$$f(x^{(k+1)}) - f^\star \le \frac{1}{1 + \mu}(f(x^{(k)}) - f^\star).$$

6

Proof. From [Assumption 1](#), $\|\nabla g(x)\|_2$ and $\|\nabla h(x)\|_2$ are bounded for all $x \in \mathcal{C}$, and then it is assumed that the constant r satisfies $r > \|\nabla g(x)\|_2 + \|\nabla h(x)\|_2$ for all $x \in \mathcal{C}$. The optimality condition of DCA implies that there exists $\hat{y} \in N_{\mathcal{C}}(x^{(k+1)})$ and $\|\hat{y}\|_2 \leq r$ such that $\nabla h(x^{(k)}) = \nabla g(x^{(k+1)}) + \hat{y}$. Then it follows from (8) that

$$
\begin{aligned}
&\mu(f(x^{(k+1)}) - f^\star) \\
&\leq d_{h^*}(\nabla g(x^{(k+1)}) + \hat{y}, \nabla h(x^{(k+1)})) \\
&= d_{h^*}(\nabla h(x^{(k)}), \nabla h(x^{(k+1)})) \\
&= h^*(\nabla h(x^{(k)})) - h^*(\nabla h(x^{(k+1)})) - \langle x^{(k+1)}, \nabla h(x^{(k)}) - \nabla h(x^{(k+1)})\rangle \\
&= \langle \nabla h(x^{(k)}), x^{(k)}\rangle - h(x^{(k)}) - \langle \nabla h(x^{(k+1)}), x^{(k+1)}\rangle + h(x^{(k+1)}) \\
&\quad - \langle x^{(k+1)}, \nabla h(x^{(k)}) - \nabla h(x^{(k)})\rangle \\
&= d_h(x^{(k+1)}, x^{(k)}) \\
&= f(x^{(k)}) - f(x^{(k+1)}) - d_g(x^{(k)}, x^{(k+1)}) \\
&\leq (f(x^{(k)}) - f^\star) - (f(x^{(k+1)}) - f^\star).
\end{aligned}
$$

Thus, the desired inequality follows. $\qquad\square$

4 Application to log-determinant optimization problems

In this section, we apply DCProx to several classes of problems arising from network information theory [4, 10, 33, 34]. Each of these problems is shown to have a unique global solution [4], but so far we are not aware of any optimization algorithm that is guaranteed to find the optimum. In comparison, with [Theorem 1](#) we show that DCA converges to the unique global optimum for all the problems of interest. Moreover, for all the studied applications, projected gradient methods might not be applicable to the subproblems in (2) due to a nontrivial $\mathcal{C}$. In comparison, DCProx is able to solve these subproblems by splitting the nontrivial constraint set $\mathcal{C}$, and in particular, the use of a suitable Bregman distance further improves the efficiency of DCProx.

4.1 Two receiver vector Gaussian broadcast channel with private messages

Consider the optimization problem

$$
\begin{aligned}
&\text{minimize} \quad -\log\det(X + \Sigma_1) + \lambda \log\det(X + \Sigma_2) \\
&\text{subject to} \quad 0 \preceq X \preceq C,
\end{aligned}
\tag{9}
$$

where the variable is $X \in \mathbb{S}^n$, and the coefficients are $\Sigma_1, \Sigma_2 \in \mathbb{S}_{++}^n$, $C \in \mathbb{S}_+^n$, and $\lambda > 1$. This problem arises from various applications in network information theory, wireless communication, *etc.*; see, *e.g.*, [33, 35]. In certain applications, the unique global optimum of the problem evaluates the capacity region of two receiver vector Gaussian broadcast channels with private messages [4]. Due to the nature of its origin, only the global solution of (9) is useful for the underlying application, and other first-order stationary points are of limited interest.

4.1.1 DCA and its global convergence

The problem (9) is a DC program in the form of (1) with

$$g(X) = -\log\det(X + \Sigma_1), \qquad h(X) = -\lambda\log\det(X + \Sigma_2), \qquad \mathcal{C} = \{X \in \mathbb{S}^n \mid 0 \preceq X \preceq C\}.$$

In this problem, the functions g, h are strongly convex on the bounded set $\mathcal{C}$, and have an Lipschitz continuous gradient on $\operatorname{int}\mathcal{C}$. (But they are not strongly convex nor L-smooth on their natural domains.) This fact helps to establish the inequality (8) for (9), and then global convergence of DCA is provided in Proposition 1.

Proposition 1. *Problem (9) satisfies Assumption 1. DCA for (9) converges to its minimizer at a linear rate.*

Proof. It suffices to show that the problem (9) satisfies the inequality (8). Within the bounded, convex set $\mathcal{C} = \{X \in \mathbb{S}^n \mid 0 \preceq X \preceq C\}$, the function $g(X) = -\log\det(X + \Sigma_1)$ has a Lipschitz continuous gradient and $h(X) = -\lambda\log\det(X + \Sigma_2)$ is strongly convex. Thus, there exists two positive constants $\alpha, \beta > 0$ such that for all $X \in \mathcal{C}$,

$$g(X) - g(X^\star) \le \tfrac{\alpha}{2}\|X - X^\star\|_F^2$$
$$h(X) - h(X^\star) \ge \tfrac{\beta}{2}\|X - X^\star\|_F^2,$$

where $X^\star$ is the global optimum for (9). Thus it follows that

$$f(X) - f(X^\star) \le \tfrac{\alpha - \beta}{2}\|X - X^\star\|_F^2 \tag{10}$$

for all $X \in \mathcal{C}$. (It implies from the optimality of $X^\star$ that $\alpha > \beta$.)

On the other hand, the convex conjugate h^* is also strongly convex on the bounded, convex set $\mathcal{C}$. Thus, there exists some $\gamma > 0$ such that $d_{h^*}(U, V) \ge \tfrac{\gamma}{2}\|U - V\|_F^2$; see, *e.g.*, [21]. Hence, the right-hand side of (8) is bounded below by

$$
\begin{aligned}
d_{h^*}(\nabla g(X) + Y, \nabla h(X)) &\ge \tfrac{\gamma}{2}\|\nabla g(X) + Y - \nabla h(X)\|_F^2 \\
&= \tfrac{\gamma}{2}\|\nabla g(X) + Y - \nabla h(X) - (\nabla g(X^\star) + Y^\star - \nabla h(X^\star))\|_F^2 \\
&\ge \tfrac{\gamma}{2}\big|\|\nabla f(X) - \nabla f(X^\star) + (Y - Y^\star)\|_F^2,
\end{aligned}
\tag{11}
$$

where $Y^\star \in N_{\mathcal{C}}(X^\star)$ satisfies $\nabla g(X^\star) + Y^\star = \nabla h(X^\star)$. Following the same argument in the first part, we can see that there exists $\kappa_{\max} > \kappa_{\min} > 0$ such that $\kappa_{\min}\|X - X^\star\|_F \le \|\nabla f(X) - \nabla f(X^\star)\|_F \le \kappa_{\max}\|X - X^\star\|_F$ for all $X \in \mathcal{C}$. In addition, for all $Y \in N_{\mathcal{C}}(X)$ and $\|Y\|_F \le r$, there exists $\xi_{\max} > \xi_{\min} > 0$ such that $\xi_{\min} \le \|Y - Y^\star\|_F \le \xi_{\max}$. Thus, it follows that

$$
\begin{aligned}
&\|\nabla f(X) - \nabla f(X^\star) + (Y - Y^\star)\|_F \\
&\ge \big|\|\nabla f(X) - \nabla f(X^\star)\|_F - \|Y - Y^\star\|_F\big| \\
&\ge \max\{\kappa_{\min}\|X - X^\star\|_F - \xi_{\max}, \xi_{\min} - \kappa_{\max}\|X - X^\star\|_F\}.
\end{aligned}
\tag{12}
$$

Clearly, there exists $\eta > 0$ such that

$$\eta\|X - X^\star\|_F \le \xi_{\min} - \kappa_{\max}\|X - X^\star\|_F. \tag{13}$$

Combining (10)–(13), we conclude that there exists $\mu = (\alpha - \beta)/(\eta^2\gamma) > 0$ such that

$$\mu(f(X) - f(X^\star)) \le d_{h^*}(\nabla g(X) + Y, \nabla h(X)). \qquad \square$$

4.1.2 DCProx: Bregman PDHG as the subproblem solver

The kth DCA iteration (2) for problem (9) involves the convex subproblem

$$\begin{aligned}
\text{minimize} \quad & -\log\det(U + \Sigma_1) + \lambda\langle(X^{(k)} + \Sigma_2)^{-1}, U\rangle \\
\text{subject to} \quad & 0 \preceq U \preceq C
\end{aligned} \tag{14}$$

with optimization variable $U \in \mathbb{S}^n$. Projection onto the constraint set $\mathcal{C}$ is much more expensive than a PSD projection, which prevents the direct use of first-order methods based on projections and gradient evaluations. Nevertheless, the constraint set can be written as $\mathcal{C} = \mathbb{S}^n_+ \cap \mathcal{C}_0$ with $\mathcal{C}_0 = \{U \in \mathbb{S}^n \mid U \preceq C\}$, and projection on each component set ($\mathbb{S}^n_+$ and $\mathcal{C}_0$) is much easier. This observation motivates the use of primal–dual proximal algorithms, and when Framework 1 is applied, the inner iterations (7) (with $\phi_{\mathrm{p}} = g$ and $\phi_{\mathrm{d}} = \frac{1}{2}\|\cdot\|_2^2$) reduce to

$$U^{(t+1)} = \operatorname*{argmin}_{U \in \mathbb{S}^n_+} -(1 + \tfrac{1}{\tau})\log\det(U + \Sigma_1) + \langle V^{(t)} + \lambda(X^{(k)} + \Sigma_2)^{-1} + \tfrac{1}{\tau}(U^{(t)} + \Sigma_1)^{-1}, U\rangle$$

$$V^{(t+1)} = \mathbf{prox}_{\sigma\delta^*_{\mathcal{C}_0}}(V^{(t)} + \sigma(2U^{(t+1)} - U^{(t)})),$$

Dual update Recall $\mathbf{prox}_{\delta_{\mathcal{C}_0}} = \Pi_{\mathcal{C}_0}$ is the projection operator, and the V-update reduces to

$$V^{(t+1)} = \Pi_{\mathbb{S}^n_+}\big(\sigma C - V^{(t)} - \sigma(2U^{(t+1)} - U^{(t)})\big) - \big(\sigma C - V^{(t)} - \sigma(2U^{(t+1)} - U^{(t)})\big);$$

i.e., given the eigen-decomposition of $\widehat{V} = \sigma C - V^{(t)} - \sigma(2U^{(t+1)} - U^{(t)})$, the next iterate $V^{(t+1)}$ simply sets the positive eigenvalues of $\widehat{V}$ to zero and takes the absolute values of the negative ones.

Primal update If the squared Euclidean distance is used in the U-update, the computation of the proximal mapping needs an eigen-decomposition and solving a number of n quadratic equations (see, *e.g.*, [36, §6.5]). In comparison, with $\phi_{\mathrm{p}}(X) = g(X)$, we can avoid solving a large number of quadratic equations, and the U-update has a closed-form expression given by Proposition 2.

Proposition 2. *Consider the optimization problem*

$$\begin{aligned}
\text{minimize} \quad & -\log\det(X + \Sigma) + \langle A, X\rangle \\
\text{subject to} \quad & X \succeq 0
\end{aligned}$$

with variable $X \in \mathbb{S}^n$ and data matrices $A, \Sigma \in \mathbb{S}^n_{++}$. Suppose $\Sigma^{1/2}A\Sigma^{1/2} = Q\Lambda Q^T$ is the eigen-decomposition, with eigenvalues $\lambda_1 \geq \lambda_2 \geq \cdots \geq \lambda_n > 0$. Then the optimal solution is

$$X^\star = \Sigma^{1/2}Q\psi(\Lambda)Q^T\Sigma^{1/2}.$$

Here the scalar function ψ is defined as $\psi(\gamma) = \max\{(1 - \gamma)/\gamma, 0\}$, and $\psi(\Lambda)$ is a diagonal matrix with diagonal elements $\psi(\Lambda)_{ii} = \psi(\lambda_i)$.

Proof. With a change of variables $Y = Q^T\Sigma^{-1/2}X\Sigma^{-1/2}Q$, the objective function can be equivalently written as

$$\begin{aligned}
f(X) &= -\log\det\big(\Sigma^{1/2}(I + \Sigma^{-1/2}X\Sigma^{-1/2})\Sigma^{1/2}\big) + \langle\Sigma^{1/2}A\Sigma^{1/2}, \Sigma^{-1/2}X\Sigma^{-1/2}\rangle \\
&= -\log\det\Sigma - \log\det(I + \Sigma^{-1/2}X\Sigma^{-1/2}) + \langle\Sigma^{1/2}A\Sigma^{1/2}, \Sigma^{-1/2}X\Sigma^{-1/2}\rangle
\end{aligned}$$

$$= -\log \det \Sigma - \log \det(I + Y) + \langle \Lambda, Y \rangle.$$

We argue that the optimal Y is a diagonal matrix. To see this, suppose Y is diagonal. Then it follows that $\langle \Lambda, Y \rangle \le \langle \Lambda, VYV^T \rangle$, for any diagonal matrix Λ with nonnegative diagonal elements and any orthogonal matrix V. Hence the optimal Y has to diagonal, and f further reduces to

$$f(X) = -\log \det \Sigma + \sum_{i=1}^{n} (\lambda_i Y_{ii} - \log(1 + Y_{ii})).$$

Recall that $A, \Sigma \in \mathbb{S}_{++}^n$, and it follows that $\gamma_i > 0$ for all $i = 1, \ldots, n$. Define the scalar function $h(\gamma) = \lambda\gamma - \log(1 + \gamma)$ with domain $\mathbb{R}_+$. If $\lambda > 1$, the function h is monotone increasing on its domain, and the minimizer of h is taken at 0. If $0 < \lambda \le 1$, the function h takes its minimum at $\gamma = (1 - \lambda)/\lambda$. Hence, the optimal $Y^\star$ is a diagonal matrix with diagonal entries

$$Y_{ii}^\star = \psi(\lambda_i) = \max\left\{ \frac{1 - \lambda_i}{\lambda_i}, 0 \right\},$$

and then the optimal solution $X^\star$ is given by

$$X^\star = \Sigma^{1/2} Q \begin{bmatrix} \psi(\lambda_1) & & & \\ & \psi(\lambda_2) & & \\ & & \ddots & \\ & & & \psi(\lambda_n) \end{bmatrix} Q^T \Sigma^{1/2}. \qquad \square$$

4.2 Gaussian broadcast channel with common messages

The second nonconvex DC program of interest is a generalization of (9), and reads as

$$\begin{aligned} \text{minimize} \quad & -\beta \log \det(X + Y + \Sigma_2) + \alpha \log \det(X + Y + \Sigma_1) \\ & -\log \det(X + \Sigma_1) + \lambda \log \det(X + \Sigma_2) \\ \text{subject to} \quad & X + Y \preceq C, \ X \succeq 0, \ Y \succeq 0, \end{aligned} \tag{15}$$

where the optimization variables are $X, Y \in \mathbb{S}^n$, and the coefficients are $\Sigma_1, \Sigma_2 \in \mathbb{S}_{++}^n$, $C \in \mathbb{S}_+^n$, $\alpha \in [0, 1]$, $\beta > 0$, and $\lambda > 1$. (This choice of parameters is included in [34], and other choices in [34] give an optimization problem that is easier to solve than the presented one.) The unique solution of (15) evaluates the capacity region of two receiver vector Gaussian broadcast channels with common messages and private messages; see [34] and references therein.

The problem (15) can also be written in the form of (1), with convex functions $g, h \colon \mathbb{S}^n \times \mathbb{S}^n \to \mathbb{R}$

$$\begin{aligned} g(X, Y) &= -\beta \log \det(X + Y + \Sigma_2) - \log \det(X + \Sigma_1), \\ h(X, Y) &= -\alpha \log \det(X + Y + \Sigma_1) - \lambda \log \det(X + \Sigma_2), \end{aligned}$$

and the constraint set $\mathcal{C} = \{(X, Y) \mid X \succeq 0, Y \succeq 0, X + Y \preceq C\}$. Following the same ideas as in Proposition 1, we show that DCA applied to (15) converges to its unique global optimum.

Proposition 3. *Problem* (15) *satisfies* Assumption 1. *DCA for* (15) *converges to its minimizer at a linear rate.*

10

In the remainder of this subsection, we focus on the convex subproblem at kth DCA iteration:

$$\begin{array}{ll} \text{minimize} & -\beta \log\det(U + \widetilde{U} + \Sigma_2) + \alpha\langle(X^{(k)} + Y^{(k)} + \Sigma_1)^{-1}, U + \widetilde{U}\rangle \\ & -\log\det(U + \Sigma_1) + \lambda\langle(X^{(k)} + \Sigma_2)^{-1}, U\rangle \\ \text{subject to} & U + \widetilde{U} \preceq C, \ U \succeq 0, \ \widetilde{U} \succeq 0 \end{array} \tag{16}$$

with variables $U, \widetilde{U} \in \mathbb{S}^n$. We first make a change of variables $W := U + \widetilde{U}$, and then reformulate the problem (16) in the form of (5) with

$$\begin{aligned} F(U, W) = {}& -\beta \log\det(W + \Sigma_2) + \alpha\langle(X^{(k)} + Y^{(k)} + \Sigma_1)^{-1}, W\rangle \\ & -\log\det(U + \Sigma_1) + \lambda\langle(X^{(k)} + \Sigma_2)^{-1}, U\rangle + \delta_{\mathbb{S}^n_+}(U) + \delta_{\mathcal{C}_0}(W) \\ G = {}& \delta_{\mathbb{S}^n_+}, \quad \mathcal{C}_0 = \{W \in \mathbb{S}^n \mid W \preceq C\}, \end{aligned}$$

and the linear mapping $\mathcal{A} \colon \mathbb{S}^n \times \mathbb{S}^n \to \mathbb{S}^n$ is defined as $\mathcal{A}(U, W) = W - U$. When [Framework 1] is applied to this splitting strategy, the inner iterations (7) reduce to

$$U^{(t+1)} = \underset{U \in \mathbb{S}^n_+}{\operatorname{argmin}}\big(-\log\det(U + \Sigma_1) + \langle -V^{(t)} + \lambda(X^{(k)} + \Sigma_2)^{-1}, U\rangle + \tfrac{1}{\tau}d_1(U, U^{(t)})\big)$$

$$W^{(t+1)} = \underset{W \in \mathcal{C}}{\operatorname{argmin}}\big(-\beta \log\det(W + \Sigma_2) + \langle V^{(t)} + \alpha(X^{(k)} + Y^{(k)} + \Sigma_1)^{-1}, W\rangle + \tfrac{1}{\tau}d_2(W, W^{(t)})\big)$$

$$V^{(t+1)} = \mathbf{prox}_{\sigma\delta^*_{\mathbb{S}^n_+}}\big(V^{(t)} + \sigma(2(W^{(t+1)} - U^{(t+1)}) - (W^{(t)} - U^{(t)}))\big).$$

The V-update is simple and involves an eigen-decomposition. In the U-update, we use the Bregman distance generated by $\phi_1(V) = -\log\det(V + \Sigma_1)$, and the closed-form update rule is given by [Proposition 2]. A similar result exists for the W-update if we choose the kernel function $\phi_2(W) = -\log\det(W + \Sigma_2)$ with $\mathbf{dom}\,\phi_2 = \{W \in \mathbb{S}^n \mid W + \Sigma_2 \succ 0\}$. Then W-update minimizes

$$\tilde{f}(W) = -(\beta + \tfrac{1}{\tau})\log\det(W + \Sigma_2) + \langle V^{(t)} + \alpha(X^{(k)} + Y^{(k)} + \Sigma_1)^{-1} + \tfrac{1}{\tau}(W^{(t)} + \Sigma_2)^{-1}, W\rangle$$

subject to $W \preceq C$, and has a closed-form expression given in [Proposition 4].

Proposition 4. *Consider the optimization problem*

$$\begin{array}{ll} \text{minimize} & -\log\det(X + \Sigma) + \langle A, X\rangle \\ \text{subject to} & X \preceq C \end{array}$$

with variable $X \in \mathbb{S}^n$ and data $A, \Sigma \in \mathbb{S}^n_{++}$, $C \in \mathbb{S}^n_+$. Suppose $(C + \Sigma)^{1/2}A(C + \Sigma)^{1/2} = Q\Lambda Q^T$ is the eigen-decomposition, with eigenvalues $\lambda_1 \geq \lambda_2 \geq \cdots \geq \lambda_n > 0$. Then the optimal solution is

$$X^\star = (C + \Sigma)^{1/2}Q\psi(\Lambda)Q^T(C + \Sigma)^{1/2} - \Sigma.$$

Here the scalar function ψ is defined as $\psi(\gamma) = \min\{1/\gamma, 1\}$, and $\psi(\Lambda)$ is a diagonal matrix with diagonal elements $\psi(\Lambda)_{ii} = \psi(\lambda_i)$.

Proof. With a change of variables $Z = X + \Sigma, Y = Q^T(C + \Sigma)^{-1/2}Z(K + \Sigma)^{-1/2}Q$, the objective function can be equivalently written as

$$f(X) = -\log\det(C + \Sigma) - \langle A, \Sigma\rangle - \log\det Y + \langle\Lambda, Y\rangle,$$

and the constraint is equivalent to $Y \preceq I$. Following the argument in the proof of Proposition 2, we can show that the optimal Y is a diagonal matrix. Then f further reduces to

$$f(X) = -\log\det(C + \Sigma) - \langle A, \Sigma \rangle + \sum_{i=1}^{n} \lambda_i Y_{ii} - Y_{ii}.$$

Define the scalar function $h(\gamma) = \lambda\gamma - \log\gamma$ and parameter $\lambda \in \mathbb{R}_{++}$. The function h is decreasing on $(0, 1/\lambda)$ and increasing on $(1/\lambda, +\infty)$. Hence, the optimal $Y^\star$ is a diagonal matrix with diagonal entries

$$Y_{ii}^\star = \psi(\lambda_i) = \max\left\{\frac{1 - \lambda_i}{\lambda_i}, 0\right\},$$

and the optimal solution $X^\star$ is given by

$$X^\star = (C + \Sigma)^{1/2} Q \begin{bmatrix} \psi(\lambda_1) & & & \\ & \psi(\lambda_2) & & \\ & & \ddots & \\ & & & \psi(\lambda_n) \end{bmatrix} Q^T (C + \Sigma)^{1/2} - \Sigma. \qquad \square$$

4.3   Generalized Brascamp–Lieb inequality

The third problem arises from theoretical computer science and functional analysis [10, 37, 38]

$$\begin{aligned} \text{minimize} \quad & -\sum_{i=1}^{p} \beta_i \log\det X_i + \sum_{j=1}^{q} \alpha_j \log\det\Big(\sum_{i=1}^{p} A_{ij} X_i A_{ij}^T + \rho I_{m_j}\Big) \\ \text{subject to} \quad & 0 \preceq X_i \preceq C_i, \quad i \in [p], \end{aligned} \tag{17}$$

where the optimization variables are $X_i \in \mathbb{S}^{n_i}$, $i \in [p]$, and the coefficients are $C_i \in \mathbb{S}_+^{n_i}$ for $i \in [p]$, $A_{ij} \in \mathbb{R}^{m_j \times n_i}$ for $(i, j) \in [p] \times [q]$ with full row rank, $\alpha \in \mathbb{R}_+^q$, $\beta \in \mathbb{R}_+^p$, and $\rho > 0$. Its unique global optimum [4] computes the optimal constant for a family of inequalities in functional analysis and probability theory, including the Brascamp–Lieb inequality and the entropy power inequality [10]. In particular, when the objective in (17) reduces to

$$f_{\mathrm{BL}}(X) = -\log\det X + \sum_{j=1}^{q} \alpha_j \log\det(A_j X A_j^T)$$

(with $\mathbf{1}^T \alpha = 1$) and the constraints disappear, the problem reduces to the computation of the classical Brascamp–Lieb constant [3]. A recent paper [7] shows that the function f_{BL} is geodesically convex on $\mathbb{S}_{++}^n$ and DCA converges linearly to the global optimum when minimizing f_{BL}. However, in the general case, the objective in (17) may not be geodesically convex, and the analysis in [7] is not directly applicable. But still, the general problem (17) is a DC program in the form of (1), with

$$g(X) = -\sum_{i=1}^{p} \beta_i \log\det X_i, \qquad h(X) = -\sum_{j=1}^{q} \alpha_j \log\det\Big(\sum_{i=1}^{p} A_{ij} X_i A_{ij}^T + \rho I_{m_j}\Big),$$

and $\mathcal{C} = \{(X_1, \ldots, X_p) \mid 0 \leq X_i \leq C_i, \ i \in [p]\}$, and we have the following result.

Proposition 5. *Problem* (17) *satisfies* Assumption 1. *DCA for* (17) *converges linearly to its minimizer.*

Apart from the convergence guarantee, the subproblem at kth DCA iteration is

$$\text{minimize} \quad -\sum_{i=1}^{p} \beta_i \log \det U_i + \sum_{i=1}^{p} \Big\langle \sum_{j=1}^{q} \alpha_j A_{ij}^T \big(\sum_{i=1}^{p} A_{ij} X_i^{(k)} A_{ij}^T + \rho I_{m_j}\big)^{-1} A_{ij}, U_i \Big\rangle,$$
$$\text{subject to} \quad 0 \preceq U_i \preceq C_i, \quad i \in [p] \tag{18}$$

with variables $U_i \in \mathbb{S}^{n_i}$, $i \in [p]$, and can be solved efficiently by Bregman PDHG. To see this, we first rewrite the subproblem (18) in the form of (5): $F(\{U_i\}) = \sum_{i=1}^{p} F_i(U_i)$, $G(\{V_i\}) = \sum_{i=1}^{p} G_i(V_i)$,

$$F_i(U_i) = -\beta_i \log \det U_i + \sum_{j=1}^{q} \alpha_j \Big\langle A_{ij}^T \Big(\sum_{i=1}^{p} A_{ij} X_i^{(k)} A_{ij}^T + \rho I_{m_j}\Big)^{-1} A_{ij}, U_i \Big\rangle + \delta_{\mathbb{S}_+^{n_i}}(U_i),$$

$G_i = \delta_{\mathcal{C}_i}$ is the indicator function of $\mathcal{C}_i = \{X \in \mathbb{S}^{n_i} \mid X \preceq C_i\}$, and the linear operator $\mathcal{A} = \text{Id}$ is the identity mapping. Note that both functions F and G are separable in their respective arguments, while the original function h is not separable in $X_1, \ldots, X_p$. The tth iteration of Bregman PDHG takes the following updates for all $i \in [p]$

$$U_i^{(t+1)} = \underset{U_i \in \mathbb{S}_+^{n_i}}{\operatorname{argmin}} \Big(-\beta_i \log \det U_i + \sum_{j=1}^{q} \alpha_j \big\langle A_{ij}^T (A_{ij} X_i^{(k)} A_{ij}^T + \rho I_{m_j})^{-1} A_{ij}, U_i \big\rangle$$
$$+ \langle V_i^{(t)}, U_i \rangle + \tfrac{1}{\tau} d(U_i, U_i^{(t)}) \Big) \tag{19a}$$

$$V_i^{(t+1)} = \mathbf{prox}_{\sigma \delta_{\mathcal{C}_i}^*} (V_i^{(t)} + \sigma(2U_i^{(t+1)} - U_i^{(t)})), \tag{19b}$$

Again, each V_i-update is dominated by an eigen-decomposition. For each U_i-update, we choose the kernel function $\phi_i(X) = -\log \det X$ with domain $\mathbf{dom}\, \phi_i = \mathbb{S}_{++}^{n_i}$, and the resulting update (19a) has a closed-form expression. Recall that the limit point $\{(\widehat{U}_i, \widehat{V}_i)\}$ of Bregman PDHG (19) is the solution of the subproblem (18), and is also the $(k+1)$st DCA iteration, i.e., $X_i^{(k+1)} := \widehat{U}_i$, $i \in [p]$. The (U_i, V_i)-update in Bregman PDHG (19) can be performed in parallel while the DCA iterates $\{X_i^{(k)}\}$ are coupled in the original function h as well as its linearization in (18).

5 Numerical experiments

In this section, we evaluate the performance of DCProx applied to the three DC programs in Section 4. The numerical results verify that DCProx is able to solve the three classes of nonconvex programs with only a few number of DCA iterations and an efficient subproblem solver.

The experiments are carried out in MATLAB 2021b on a server with AMD Opteron Processor 6128 CPU and 32GB memory. Since real data are not available for these problems, all the data in our experiments are synthetic, following [7, 9]. Two classes of baselines are considered. The nonlinear programming solver fmincon in Global Optimization Toolbox of Matlab is used to compare the optimality of the computed solutions. For efficiency comparison, we also replace the subproblem solver with Euclidean PDHG, and MOSEK [39]. Regarding implementation details, the line search technique for Bregman PDHG [21, 29, 31] is adopted for an adaptive choice of the stepsizes τ, σ.

Table 1 presents the results of DCProx and fmincon. Small-size problems are sufficient for comparison with fmincon, and one typical result in the experiments is presented. For all three classes of problems, fmincon might return a sub-optimal stationary point while DCProx is able to find a feasible

n	algo	obj. value	runtime		n	algo	obj. value	runtime		n	algo	obj. value	runtime
100	DCProx	-4.2736	6.82		100	DCProx	-4.2674	8.63		100	DCProx	-4.4671	13.28
	fmincon	-3.3742	10.34			fmincon	-3.6485	22.17			fmincon	-3.9126	135.84
500	DCProx	-3.5771	37.98		500	DCProx	-5.4356	54.89		500	DCProx	-3.9523	67.02
	fmincon	-1.2548	126.76			fmincon	-1.3288	262.13			fmincon	-2.5843	561.57

Table 1: Results of DCProx and fmincon for the problems (9) (left), (15) (middle), and (17) (right). Objective function values and runtime (in sec.) are reported for a typical run of the algorithms.

n	algo	avg. num. of DCA iter.	avg. num. of inner iter.	avg. runtime (in sec.)	avg. runtime per DCA iter.
500	DCProx	9.5	1735	3.63×10^2	38.23
	DCProx (Euc.)	9.5	2046	3.81×10^2	40.09
	DC-MOSEK	8.9	76	1.02×10^3	108.1
1000	DCProx	13.6	1324	1.73×10^3	127.2
	DCProx (Euc.)	13.6	1684	2.20×10^3	162.4
	DC-MOSEK	13.2	96	9.87×10^3	726.3

Table 2: Results for problem (9). Three algorithms are tested: DCProx, DCProx with Euclidean PDHG, DCA with MOSEK as subproblem solver. The results are averaged over 10 synthetic datasets.

point with a smaller objective value. This is consistent with our analysis that for these problems, DCA converges to the unique global optimum (see Theorem 1).

Tables 2 to 4 present the numerical results for (9), (15), and (17), respectively. Three algorithms are tested: DCProx, DCProx with Euclidean PDHG, and DCA with MOSEK as the subproblem solver. The results are averaged over 10 experiments (with different synthetic data). In all the experiments, DCA (the outer loop of DCProx) converges within around 10 iterations, which supports the linear convergence result in Theorem 1. DC-MOSEK has a bit fewer DCA iterations because the interior-point solver MOSEK often returns a more accurate subproblem solution than first-order PDHG. It is also observed that the customized subproblem solver in DCProx (Bregman PDHG) performs consistently better than the other two. The average number of iterations in both PDHG methods is about one to two thousand, which is typical for these primal–dual first-order proximal methods. (interior-point solver MOSEK takes much fewer iterations but has much more expensive per-iteration complexity.) The per-iteration runtime of Bregman PDHG is smaller than that of its Euclidean variant, because the former does not need to solve a number of n quadratic equations (as explained in Section 4.1). Bregman PDHG takes fewer iterations (on average) than Euclidean PDHG, which might be explained by the fact that the Bregman distance can help build a more accurate local optimization model around the current iterate [21].

6   Conclusions

We present the DCProx framework for solving several classes of difference-of-convex (DC) programming problems that have a bounded, closed, and convex constraint set. DCProx incorporates the difference-of-convex algorithm (DCA) in the outer iteration and a Bregman primal–dual proximal

n	algo	avg. num. of DCA iter.	avg. num. of inner iter.	avg. runtime (in sec.)	avg. runtime per DCA iter.
	DCProx	10.2	1273	5.71×10^2	56.07
500	DCProx (Euc.)	10.2	1496	5.63×10^2	75.83
	DC-MOSEK	9.8	93	2.32×10^3	225.1
	DCProx	12.4	1468	3.50×10^3	281.9
1000	DCProx (Euc.)	12.4	1632	4.08×10^3	313.3
	DC-MOSEK	-	-	-	-

Table 3: Results for problem (15). Three algorithms are tested: DCProx, DCProx with Euclidean PDHG, DCA with MOSEK as the subproblem solver. The results are averaged over 10 synthetic datasets. '-' indicates the experiments do not finish in 6 hours.

n	algo	avg. num. of DCA iter.	avg. num. of inner iter.	avg. runtime (in sec.)	avg. runtime per DCA iter.
	DCProx	14.7	1157.9	9.98×10^2	64.21
500	DCProx (Euc.)	14.7	1297.5	1.14×10^3	70.42
	DC-MOSEK	13.9	85.2	5.36×10^4	364.8
	DCProx	14.2	1048.7	5.74×10^3	412.6
1000	DCProx (Euc.)	14.2	1362.6	6.52×10^3	468.7
	DC-MOSEK	-	-	-	-

Table 4: Results for problem (17) ($p = q = 3$, $n_i = n$ for $i \in [p]$). Three algorithms are tested: DCProx, DCProx with Euclidean PDHG, DCA with MOSEK as the subproblem solver. The results are averaged over 10 synthetic datasets. '-' indicates the experiments do not finish in 6 hours.

algorithm (Bregman PDHG) in the inner iteration. In the theoretical aspect, the outer iteration DCA is shown to converge linearly to a global optimum under an extension of the classical Polyak–Łojasiewicz condition; from the computational perspective, Bregman PDHG solves the subproblem at each DCA iteration efficiently, with simple update rules and cheap per-iteration complexity. As an application, we consider several classes of matrix optimization problems involving log-determinant functions. For these problems, linear global convergence of DCA is established via the presented analysis, and Bregman PDHG efficiently solves the DCA subproblems by a suitable choice of Bregman distances. Numerical experiments are conducted to verify the efficiency of DCProx. Future work are needed to identify infeasible and pathological DC programs in the aforementioned applications.

References

[1] T. Lipp and S. P. Boyd, "Variations and extension of the convex–concave procedure," *Optimization and Engineering*, vol. 17, no. 2, pp. 263–287, 2016.

[2] H. A. Le Thi and T. Pham Dinh, "DC programming and DCA: thirty years of developments," *Mathematical Programming*, vol. 169, no. 1, pp. 5–68, 2018.

[3] S. Sra, N. K. Vishnoi, and O. Yildiz, "On geodesically convex formulations for the Brascamp–Lieb constant," in *Approximation, Randomization, and Combinatorial Optimization. Algorithms and Techniques (APPROX/RANDOM 2018)* (E. Blais, K. Jansen, J. D. P. Rolim, and D. Steurer, eds.), vol. 116 of *Leibniz International Proceedings in Informatics (LIPIcs)*, pp. 25:1–25:15, 2018.

[4] C. W. K. Lau, C. Nair, and C. Yao, "Uniqueness of local maximizers for some non-convex log-determinant optimization problems using information theory," in *2022 IEEE International Symposium on Information Theory (ISIT)*, pp. 432–437, 2022.

[5] T. Pham Dinh and H. A. Le Thi, "Convex analysis approach to DC programming: Theory, algorithms and applications," *Acta mathematica vietnamica*, vol. 22, no. 1, pp. 289–355, 1997.

[6] H. A. Le Thi and T. Pham Dinh, "The DC (difference of convex functions) programming and DCA revisited with DC models of real world nonconvex optimization problems," *Annals of Operations Research*, vol. 133, pp. 23–46, 2005.

[7] M. Weber and S. Sra, "On a class of geodesically convex optimization problems solved via Euclidean MM methods," *arXiv e-prints*, vol. arXiv:2206.11426, 2022.

[8] H. Abbaszadehpeivasti, E. de Klerk, and M. Zamani, "On the rate of convergence of the difference-of-convex algorithm (DCA)," *Journal of Optimization Theory and Applications*, 2023.

[9] O. Faust, H. Fawzi, and J. Saunderson, "A Bregman divergence view on the difference-of-convex algorithm," in *Proceedings of The 26th International Conference on Artificial Intelligence and Statistics* (F. Ruiz, J. Dy, and J.-W. van de Meent, eds.), vol. 206 of *Proceedings of Machine Learning Research*, pp. 3427–3439, PMLR, 2023.

[10] V. Anantharam, V. Jog, and C. Nair, "Unifying the Brascamp–Lieb inequality and the entropy power inequality," *IEEE Transactions on Information Theory*, vol. 68, no. 12, pp. 7665–7684, 2022.

[11] Y. Sun, P. Babu, and D. P. Palomar, "Majorization–minimization algorithms in signal processing, communications, and machine learning," *IEEE Transactions on Signal Processing*, vol. 65, no. 3, pp. 794–816, 2017.

[12] T. Pham Dinh and E. B. Souad, "Algorithms for solving a class of nonconvex optimization problems. Methods of subgradients," in *Fermat Days 85: Mathematics for Optimization* (J.-B. Hiriart-Urruty, ed.), vol. 129 of *North-Holland Mathematics Studies*, pp. 249–271, North-Holland, 1986.

[13] A. L. Yuille and A. Rangarajan, "The concave–convex procedure," *Neural Computation*, vol. 15, no. 4, pp. 915–936, 2003.

[14] B. K. Sriperumbudur and G. Lanckriet, "On the convergence of the concave–convex procedure," in *Advances in Neural Information Processing Systems*, vol. 22, Curran Associates, Inc., 2009.

[15] A. Yurtsever and S. Sra, "CCCP is Frank–Wolfe in disguise," *Advances in Neural Information Processing Systems*, vol. 35, pp. 35352–35364, 2022.

[16] A. J. Smola, S. V. N. Vishwanathan, and T. Hofmann, "Kernel methods for missing variables," in *International Workshop on Artificial Intelligence and Statistics*, pp. 325–332, PMLR, Jan. 2005.

[17] P. L. Combettes and J.-C. Pesquet, "Proximal splitting methods in signal processing," in *Fixed-Point Algorithms for Inverse Problems in Science and Engineering*, Springer Optimization and Its Applications, pp. 185–212, Springer New York, 2011.

[18] L. Condat, D. Kitahara, A. Contreras, and A. Hirabayashi, "Proximal splitting algorithms for convex optimization: A tour of recent advances, with new twists," *SIAM Review*, vol. 65, no. 2, pp. 375–435, 2023.

[19] J. Moreau, "Proximité et dualité dans un espace Hilbertien," *Bulletin de la Société Mathématique de France*, vol. 93, pp. 273–299, 1965.

[20] Y. Censor and S. Zenios, *Parallel Optimization: Theory, Algorithms and Applications*. Oxford University Press, 1997.

[21] X. Jiang and L. Vandenberghe, "Bregman three-operator splitting methods," *Journal of Optimization Theory and Applications*, pp. 1–37, 2022.

[22] H. H. Bauschke, J. Bolte, and M. Teboulle, "A descent lemma beyond Lipschitz gradient continuity: first-order methods revisited and applications," *Mathematics of Operations Research*, vol. 42, no. 2, pp. 330–348, 2017.

[23] J. Bolte, S. Sabach, S. Teboulle, and Y. Vaisbourd, "First order methods beyond convexity and Lipschitz gradient continuity with applications to quadratic inverse problems," *SIAM Journal on Optimization*, vol. 28, no. 3, pp. 2131–2151, 2018.

[24] H. T. M. Chu, L. Liang, K.-C. Toh, and L. Yang, "An efficient implementable inexact entropic proximal point algorithm for a class of linear programming problems," *Computational Optimization and Applications*, 2023.

[25] H. Chao and L. Vandenberghe, "Entropic proximal operators for nonnegative trigonometric polynomials," *IEEE Transactions on Signal Processing*, vol. 66, no. 18, pp. 4826–4838, 2018.

[26] A. Chambolle and T. Pock, "A first-order primal–dual algorithm for convex problems with applications to imaging," *Journal of Mathematical Imaging and Vision*, vol. 40, no. 1, pp. 120–145, 2011.

[27] A. Chambolle and T. Pock, "On the ergodic convergence rates of a first-order primal–dual algorithm," *Mathematical Programming*, vol. 159, no. 1, pp. 253–287, 2016.

[28] Y. Malitsky and T. Pock, "A first-order primal–dual algorithm with linesearch," *SIAM Journal on Optimization*, vol. 28, no. 1, pp. 411–432, 2018.

[29] X. Jiang and L. Vandenberghe, "Bregman primal–dual first-order method and applications to sparse semidefinite programming," *Computational Optimization and Applications*, vol. 81, no. 1, pp. 127–159, 2022.

[30] L. Condat, "A primal–dual splitting method for convex optimization involving Lipschitzian, proximable and linear composite terms," *Journal of Optimization Theory and Applications*, vol. 158, no. 2, pp. 460–479, 2013.

[31] A. Chambolle and J. P. Contreras, "Accelerated Bregman primal–dual methods applied to optimal transport and Wasserstein barycenter problems," *arXiv e-prints: 2203.00802*, 2022.

[32] H. H. Bauschke, J. Bolte, J. Chen, M. Teboulle, and X. Wang, "On linear convergence of non-Euclidean gradient methods without strong convexity and Lipschitz gradient continuity," *Journal of Optimization Theory and Applications*, vol. 182, pp. 1068–1087, 2019.

[33] H. Weingarten, Y. Steinberg, and S. S. Shamai, "The capacity region of the gaussian multiple-input multiple-output broadcast channel," *IEEE Transactions on Information Theory*, vol. 52, no. 9, pp. 3936–3964, 2006.

[34] Y. Geng and C. Nair, "The capacity region of the two-receiver Gaussian vector broadcast channel with private and common messages," *IEEE Transactions on Information Theory*, vol. 60, no. 4, pp. 2087–2104, 2014.

[35] T. Liu and P. Viswanath, "An extremal inequality motivated by multiterminal information-theoretic problems," *IEEE Transactions on Information Theory*, vol. 53, no. 5, pp. 1839–1851, 2007.

[36] S. P. Boyd, N. Parikh, E. Chu, B. Peleato, and J. Eckstein, "Distributed optimization and statistical learning via the alternating direction method of multipliers," *Foundations and Trends in Machine Learning*, vol. 3, no. 1, pp. 1–122, 2011.

[37] H. J. Brascamp and E. H. Lieb, "On extensions of the Brunn–Minkowski and Prékopa–Leindler theorems, including inequalities for log concave functions, and with an application to the diffusion equation," *Journal of Functional Analysis*, vol. 22, no. 4, pp. 366–389, 1976.

[38] J. Bennett, A. Carbery, M. Christ, and T. Tao, "The Brascamp—Lieb inequalities: Finiteness, structure and extremals," *Geometric and Functional Analysis*, vol. 17, pp. 1343—1415, 2008.

[39] MOSEK ApS, *The MOSEK optimization toolbox for MATLAB manual. Version 9.3.*, 2022.