EURO Journal on Computational Optimization 13 (2025) 100123

Contents lists available at ScienceDirect

EURO Journal on Computational Optimization

journal homepage: www.elsevier.com/locate/ejco

Randomized block coordinate DC algorithm

Hoomaan Maskan[a,*], Paniz Halvachi[c], Suvrit Sra[b], Alp Yurtsever[a]

[a] Umeå University, Umeå, Sweden
[b] Technical University of Munich, Munich, Germany
[c] Independent Researcher

HIGHLIGHTS

- Development of a novel variant of the difference of convex algorithm (DCA) that incorporates randomized coordinate descent updates.
- Design of a block coordinate Expectation Maximization (EM) method through the connection between DCA and the EM algorithm.
- Establishment of a non-asymptotic $\mathcal{O}(n/k)$ convergence rate in expectation for the proposed block-coordinate DCA.

ARTICLE INFO

Keywords:
Difference of convex
Block coordinate methods
Nonconvex optimization

ABSTRACT

We introduce an extension of the Difference of Convex Algorithm (DCA) in the form of a randomized block coordinate approach for problems with separable structure. For n coordinate-blocks and k iterations, our main result proves a non-asymptotic convergence rate of $O(n/k)$ in expectation, with respect to a stationarity measure based on a Forward-Backward envelope. Furthermore, leveraging the connection between DCA and Expectation Maximization (EM), we propose a randomized block coordinate EM algorithm.

1. Introduction

In nonconvex optimization, many problems find their abstraction through the lens of Difference of Convex (DC) programming, which involves minimizing an objective function represented as the difference of convex functions. This paper focuses on the following DC programming template:

$$\min_{x\in\mathcal{M}} \quad \phi(\boldsymbol{x}) := f(\boldsymbol{x}) + g(\boldsymbol{x}) - h(\boldsymbol{x}), \tag{1}$$

where f, g, and h are real-valued lower semi-continuous convex functions defined on the closed convex set $\mathcal{M} \subseteq \mathbb{R}^m$, and throughout, we assume that a finite solution $\boldsymbol{x}_\star \in \mathcal{M}$ exists, satisfying $\phi_\star := \phi(\boldsymbol{x}_\star) \leq \phi(\boldsymbol{x})$ for all $\boldsymbol{x} \in \mathcal{M}$. We assume that the function f is smooth (*i.e.,* its gradient is Lipschitz continuous), and that the function g and the set $\mathcal{M}$ exhibit a block-separable structure in relation to the coordinates of $\boldsymbol{x}$ (see Section 3). We do not impose the smoothness assumption on the terms g and h.

A large and popular class of algorithms for solving problem (1), introduced in [1], comprises instances of the DCA method. These algorithms rely on the simple yet effective idea of approximating $\phi(\boldsymbol{x})$ with the convex surrogate function obtained by using the first-order Taylor approximation of the concave component $-h(\boldsymbol{x})$. Starting from a feasible initial point $\boldsymbol{x}^0 \in \mathcal{M}$, DCA solves a sequence of convex optimization subproblems defined as

$$\min_{x\in\mathcal{M}} \hat{\phi}(\boldsymbol{x}, \boldsymbol{x}^k) := f(\boldsymbol{x}) + g(\boldsymbol{x}) - h(\boldsymbol{x}^k) - \langle \boldsymbol{v}^k, \boldsymbol{x} - \boldsymbol{x}^k \rangle, \tag{2}$$

where $\boldsymbol{x}^k$ represents the current estimate and $\boldsymbol{v}^k \in \partial h(\boldsymbol{x}^k)$ denotes a subgradient of h at $\boldsymbol{x}^k$.

* Corresponding author.
Email address: hoomaan.maskan@umu.se (H. Maskan).

https://doi.org/10.1016/j.ejco.2025.100123

Available online 26 November 2025
2192-4406/© 2025 The Author(s). Published by Elsevier B.V. on behalf of Association of European Operational Research Societies (EURO). This is an open access article under the CC BY license (http://creativecommons.org/licenses/by/4.0/).

Observe that the function $\hat{\phi}$ is a global upper bound on ϕ since h is convex; as a result, DCA can be viewed as a variant of the Majorization–Minimization (MM) algorithm [2].

The general template of DCA does not specify how, or to what degree of precision, we should solve these subproblems, and various instances can be derived by employing different techniques for their resolution. Motivated by its strong empirical performance, wide applicability, simplicity, and its interpretability, DCA has been extensively studied.

With the recent focus on machine learning and nonconvex optimization problems, along with the increasing complexity and scale of these problems, the demand for more efficient and scalable algorithms for nonconvex optimization continuously grows. A particular technique for addressing this challenge is the 'coordinate descent' (CD) approach. The main idea in CD is to minimize the multivariate objective function by updating one variable (or a *block* of variables) at every iteration, while keeping the others fixed. This approach often leads to simple subproblems that can be solved more efficiently, particularly for large-scale optimization problems commonly encountered in data science and related fields.

DC programming and DCA have been widely used in machine learning applications. Examples include kernel selection [3], sparse principal component analysis [4], discrepancy estimation in domain adaptation [5], training neural networks [6], and many more. However, to our knowledge, there is currently no true CD variant of DCA with convergence guarantees. Our main goal in this paper is to address this gap. With this in mind, we summarize now the key contributions of this paper:

- Our primary contribution is the development of a novel variant of the DCA method that incorporates randomized CD updates. We refer to this algorithm as the Block Coordinate DC algorithm (BDCA). We analyze BDCA and demonstrate that it converges in expectation to a first-order stationary point of the problem at a non-asymptotic rate of $\mathcal{O}(n/k)$, for n blocks and k iterations. Importantly, our guarantees do not necessitate the functions g or h to be smooth.
- DCA relates to the well known Expectation Maximization (EM) algorithm when dealing with exponential family of distributions [7,8]. Building on this connection, we introduce a randomized block coordinate EM method, referred to as Block EM.

2. Related work

For a comprehensive survey on the variants and applications of DCA, we refer to [8]. Here, we focus specifically on the variants involving block-coordinate or alternating updates.

2.1. Related work on BDCA

Pham Dinh et al. [9] recently proposed an alternating DC algorithm for partial DC problems involving two variables, where the objective retains a DC form when one variable is fixed. The algorithm is shown to converge to a Fréchet/Clarke critical point of the objective function under the Kurdyka-Łojasiewicz (KL) property, with non-asymptotic convergence guarantees.

More recently, Yuan [10] proposed a block-coordinate method for DC problems; however, this method differs from DCA, in that it solves a proximal subproblem instead of the DCA subproblem in (2). Furthermore, their analysis is based on the so-called *Luo-Tseng error bound* assumption, which posits that the (sub)gradients of the objective function converge to zero in norm continuously as they approach a stationary point. This assumption limits the applicability of these results in nonsmooth settings, as most nonsmooth functions of interest exhibit discontinuous changes in subgradient norm.

DCA is also known as the convex-concave procedure (CCCP) when both the convex and concave terms are differentiable. Recently, Yurtsever and Sra [11] established an equivalence between CCCP and the Frank-Wolfe algorithm applied to an epigraph reformulation of the original DC problem. As a result, they showed a $\mathcal{O}(1/k)$ non-asymptotic convergence rate for CCCP.

Our analysis differs from theirs in several key aspects. While they transfer guarantees from the Frank-Wolfe connection, we provide a direct proof. We consider the DCA template with potentially nonsmooth terms, which are managed through their subgradients. They demonstrate convergence in terms of

$$\max_{x\in\mathcal{M}} \left\{ f(\boldsymbol{x}^k) - f(\boldsymbol{x}) - \langle \nabla h(\boldsymbol{x}^k), \boldsymbol{x}^k - \boldsymbol{x} \rangle \right\}$$

which is not amenable to our block-coordinate analysis unless f is also separable. Consequently, we identify a new gap function (see Section 3) suitable for measuring proximity to first-order stationarity, which is an essential component of our analysis.

2.2. Related work on block EM

Kumar and Schmidt [12] analyzed the convergence guarantees of the EM algorithm from a majorization–minimization perspective. Assuming that the surrogate optimization problems are strongly convex, they established that EM converges at a rate of $\mathcal{O}(1/k)$ in terms of the squared gradient norm of the negative log-likelihood. However, this strong convexity assumption may not hold in general, even for exponential family distributions.

More recently, Kunstner et al. [13] studied the non-asymptotic convergence of EM for exponential family distributions by drawing a connection to mirror descent, establishing a convergence rate of $\mathcal{O}(1/k)$ in Kullback-Leibler divergence and linking it to first-order stationarity through Bregman divergences.

2.3. Other work on nonconvex BC methods

Block Successive Upperbound Minimization algorithm was proposed to minimize an approximated upperbound of any nonconvex and/or nonsmooth objective function by Razaviyayn et al. [14]. This method considers optimizing a sequence of approximate objective functions. They showed the asymptotic convergence of their method for quasi-convex functions or objectives with compact level sets. Also, they assume that each subproblem of BSUM has a unique solution.

Among recent studies in nonconvex optimization using block-coordinate methods, Xu and Yin [15] proposed a block prox-linear method. Assuming Lipschitz smoothness of the differentiable parts, they proved the convergence of a subsequence of their algorithm to a stationary point. Further, assuming the KL condition, they showed that the entire sequence converges to a critical point and estimated its asymptotic convergence rate.

Aubry et al. [16] proposed a mixture of maximum block improvement and a sequential minimization method for continuously differentiable nonconvex problems, with asymptotic first-order optimality guarantees and an application to the wireless sensor allocation problem.

More recent studies targeted a non-asymptotic convergence analysis. In the realm of nonconvex composite optimization, Chorobura and Necoara [17] considered objectives that include two nonconvex non-separable terms where one of them has Lipschitz gradients while the other function is continuously differentiable. They showed a convergence rate of $\mathcal{O}(n/\sqrt{k})$ in the sense of expected gradient norm when utilizing a randomized selection of the blocks. Their cyclic scheme achieves a rate of $\mathcal{O}(n^2/\sqrt{k})$ in the same sense.

Cai et al. [18] recently proposed a cyclic coordinate descent algorithm with variance reduction. Their objective considers a block separable and another separable sum function whose proximal operator is efficiently computable. Their analysis assumes a nonstandard Lipschitz gradient assumption defined with respect to block-dependent semi-norms and achieves a rate of $\mathcal{O}(n/k)$ in the sense of the distance of the objective's subdifferential to zero.

Deng and Lan studied a stochastic block mirror descent algorithm [19] and, more recently, proposed a randomized coordinate proximal subgradient method [20] applicable to nonconvex problem templates. Gao et al. [21] developed a randomized Bregman coordinate descent algorithm for nonconvex functions under relative smoothness. In a related direction, Faust et al. [22] provided a Bregman divergence interpretation of DCA.

Finally, Latafat et al. [23] proposed a proximal coordinate method for nonconvex composite optimization. Their formulation complements ours in terms of separability assumptions. They assume the smooth component to be separable, whereas the nonsmooth term is not. However, unlike our method, which updates coordinates through local subproblems involving only g_i for the selected coordinate blocks, their algorithm involves proximal mapping of the full g before updating a randomly selected coordinate.

3. Block coordinate DCA

We begin by establishing notation and restating our assumptions for the model problem.

We assume that the function f is L-smooth, meaning that its gradient is Lipschitz continuous with a constant $L \geq 0$. In addition, we assume that the function g and set $\mathcal{M}$ exhibit a block separable structure in relation to the coordinates of $\boldsymbol{x}$:

$$g(\boldsymbol{x}) = \sum_{i=1}^{n} g_i(\boldsymbol{D}_i \boldsymbol{x}),$$

where $\boldsymbol{D}_i$ represents $(m_i \times m)$ dimensional row subsets of the identity matrix $\boldsymbol{I}$, serving as the selection operator for a non-overlapping partition of the coordinates into n blocks:

$$\sum_{i=1}^{n} \boldsymbol{D}_i^\top \boldsymbol{D}_i = \boldsymbol{I} \quad \text{and} \quad \sum_{i=1}^{n} m_i = m.$$

Similarly, $\mathcal{M}$ can be decomposed as $\mathcal{M}_1 \times \cdots \times \mathcal{M}_n$, where the components $\mathcal{M}_i \in \mathbb{R}^{m_i}$, such that

$$\boldsymbol{x} \in \mathcal{M} \iff \boldsymbol{D}_i \boldsymbol{x} \in \mathcal{M}_i \quad \text{for} \quad i = 1, \ldots, n.$$

For the ease of presentation, we define the following coordinate vector notation:

$$x_i = \boldsymbol{D}_i \boldsymbol{x}, \quad \boldsymbol{x}_i = \boldsymbol{D}_i^\top \boldsymbol{D}_i \boldsymbol{x}, \quad \text{and} \quad \bar{\boldsymbol{x}}_i = (\boldsymbol{I} - \boldsymbol{D}_i^\top \boldsymbol{D}_i)\boldsymbol{x}.$$

Here, $x_i \in \mathbb{R}^{m_i}$ represents the i^{th} coordinate block of $\boldsymbol{x} \in \mathbb{R}^m$. The vector $\boldsymbol{x}_i$ is an extension of x_i to $\mathbb{R}^m$ with all other blocks padded with zeros. Conversely, $\bar{\boldsymbol{x}}_i$ is the complement of $\boldsymbol{x}_i$, containing zeros in the i^{th} block and ensuring that $\boldsymbol{x}_i + \bar{\boldsymbol{x}}_i = \boldsymbol{x}$.

The foundation of DCA is the surrogate convex objective obtained by linearizing the concave term in the vicinity of the current estimate $\boldsymbol{x}^k$, as defined in (2). To derive a block coordinate variant of DCA, we minimize this surrogate function along specific directions determined by randomly chosen coordinate blocks. The algorithm operates as follows:

1 Start from a feasible initial point $\boldsymbol{x}^0 \in \mathcal{M}$.
2 For $k = 1, \ldots, K$, update the estimate as follows:
   (a) Choose a coordinate block i_k uniformly at random from the set $\{1, \ldots, n\}$.
   (b) Keep the values fixed for all coordinates except i_k, resulting in $\bar{\boldsymbol{x}}_{i_k}^{k+1} = \bar{\boldsymbol{x}}_{i_k}^{k}$.
   (c) Update the i_k^{th} block by minimizing the surrogate objective (2) along these coordinates.

Algorithm 1 Block coordinate DC algorithm (BDCA).

Input: Starting point $\boldsymbol{x}^1 \subseteq \mathcal{M}$, and total number of iterations K.
for $k = 1$ **to** K **do**
  Select a subgradient $\boldsymbol{v}^k \in \partial h(\boldsymbol{x}^k)$
  Choose i_k from $\{1, \dots, n\}$ uniformly at random
  Find $x_{i_k}^{k+1} \in argmin_{x_{i_k} \in \mathcal{M}_{i_k}} f(\boldsymbol{x}_{i_k} + \bar{\boldsymbol{x}}_{i_k}^k) + g_{i_k}(x_{i_k}) - \langle v_{i_k}^k, x_{i_k} \rangle$
  Set $\boldsymbol{x}^{k+1} = \boldsymbol{x}_{i_k}^{k+1} + \bar{\boldsymbol{x}}_{i_k}^k$
end for
Output: $\boldsymbol{x}^{k+1}$

Step 2c in this algorithm description amounts to solving the following subproblem:

$$\min_{\boldsymbol{x} \in \mathcal{M}} \quad \hat{\phi}(\boldsymbol{x}, \boldsymbol{x}^k) \quad \text{subj.to} \quad \boldsymbol{x} = \boldsymbol{x}_{i_k} + \bar{\boldsymbol{x}}_{i_k}^k,$$

which is clearly equivalent to solving the following:

$$\min_{x_{i_k} \in \mathcal{M}_{i_k}} \quad \hat{\phi}(\boldsymbol{D}_{i_k}^\top x_{i_k} + \bar{\boldsymbol{x}}_{i_k}^k, \boldsymbol{x}^k).$$

To simplify the presentation, we introduce the notation $\hat{\phi}_i : \mathbb{R}^{m_i} \times \mathbb{R}^m \to \mathbb{R}$ for the surrogate objective restricted to the ith coordinate block:

$$\begin{aligned}\hat{\phi}_i(x_i, \boldsymbol{x}^k) &:= \hat{\phi}(\boldsymbol{D}_i^\top x_i + \bar{\boldsymbol{x}}_i^k, \boldsymbol{x}^k) \\ &= \hat{\phi}(\boldsymbol{x}_i + \bar{\boldsymbol{x}}_i^k, \boldsymbol{x}^k) \\ &= f(\boldsymbol{x}_i + \bar{\boldsymbol{x}}_i^k) + g_i(x_i) - g_i(x_i^k) + g(\boldsymbol{x}^k) - h(\boldsymbol{x}^k) - \langle \boldsymbol{v}_i^k, x_i - x_i^k \rangle,\end{aligned}$$

where $\boldsymbol{v}_i^k \in \partial_i h(\boldsymbol{x}^k)$. Here, only the terms in blue are relevant for solving the subproblem, as the remaining terms are constant with respect to x_i. Building on this discussion, we propose the Block Coordinate DC Algorithm (BDCA), as outlined in Algorithm 1.

3.1. Convergence guarantees for BDCA

In this section we analyze the convergence guarantees of BDCA. Specifically, we establish an upper bound on a stationarity measure for BDCA and use it to prove the convergence rate of Algorithm 1. For clarity of exposition, we first develop the analysis assuming h is differentiable, which yields convergence to first-order stationary points. We then show that the same analysis extends to the non-differentiable case, where the guarantee weakens to convergence to critical points.

Definition 1. We use the following 'gap' function as a measure of closeness to a first order stationary point:

$$\text{gap}_{\mathcal{M}}^L(\boldsymbol{y}) = \max_{\boldsymbol{x} \in \mathcal{M}} \left\{ \langle \nabla f(\boldsymbol{y}) - \nabla h(\boldsymbol{y}), \boldsymbol{y} - \boldsymbol{x} \rangle + g(\boldsymbol{y}) - g(\boldsymbol{x}) - \frac{L}{2}\|\boldsymbol{x} - \boldsymbol{y}\|^2 \right\}.$$

To the best of our knowledge, this measure of stationarity has not been explored in DC formulations. It is closely related to the Forward-Backward envelope (see Section 2.1 in [24] and references therein). The next lemma shows that this is a suitable measure for first-order stationarity.

Lemma 1. *Let $\mathcal{M} \subseteq \mathbb{R}^m$ be a closed convex set, and let $f, g, h : \mathcal{M} \to \mathbb{R}$ be lower-semicontinuous convex functions, where f is L-smooth and h is continuously differentiable. Then, $\text{gap}_{\mathcal{M}}^L(\boldsymbol{y}) \geq 0$ for all $\boldsymbol{y} \in \mathcal{M}$, and $\text{gap}_{\mathcal{M}}^L(\boldsymbol{y}) = 0$ if and only if $\boldsymbol{y}$ is a first-order stationary point of problem (1).*

Proof. The first statement is straightforward, since the expression being maximized becomes zero when we set $\boldsymbol{x} = \boldsymbol{y}$, and $\boldsymbol{y} \in \mathcal{M}$.

For the second statement, let us recall the definition of first-order stationarity. A point $\boldsymbol{y}$ is said to be a first-order stationary point of problem (1) if the following condition holds:

$$0 \in \nabla f(\boldsymbol{y}) + \partial g(\boldsymbol{y}) - \nabla h(\boldsymbol{y}) + \mathcal{N}_{\mathcal{M}}(\boldsymbol{y}). \tag{3}$$

Here, ∂g is the subdifferential of g and $\mathcal{N}_{\mathcal{M}}$ represents the normal cone of $\mathcal{M}$.

If (3) holds, it implies there exists $\boldsymbol{u} \in \partial g(\boldsymbol{y})$ such that

$$\begin{aligned}& -\nabla f(\boldsymbol{y}) - \boldsymbol{u} + \nabla h(\boldsymbol{y}) \in \mathcal{N}_{\mathcal{M}}(\boldsymbol{y}) \qquad (4)\\ \iff & \langle \nabla f(\boldsymbol{y}) + \boldsymbol{u} - \nabla h(\boldsymbol{y}), \boldsymbol{y} - \boldsymbol{x} \rangle \leq 0, \ \forall \boldsymbol{x} \in \mathcal{M} \\ \overset{\text{convex } g}{\implies} & \langle \nabla f(\boldsymbol{y}) - \nabla h(\boldsymbol{y}), \boldsymbol{y} - \boldsymbol{x} \rangle + g(\boldsymbol{y}) - g(\boldsymbol{x}) \leq 0, \\ \implies & \langle \nabla f(\boldsymbol{y}) - \nabla h(\boldsymbol{y}), \boldsymbol{y} - \boldsymbol{x} \rangle + g(\boldsymbol{y}) - g(\boldsymbol{x}) - \tfrac{L}{2}\|\boldsymbol{x} - \boldsymbol{y}\|^2 \leq 0, \\ \iff & \text{gap}_{\mathcal{M}}^L(\boldsymbol{y}) \leq 0,\end{aligned}$$

for all $\boldsymbol{x} \in \mathcal{M}$, where the second line follows from the definition of the normal cone. Combining this with the first statement, which asserts that $\text{gap}^L_{\mathcal{M}}(\boldsymbol{y}) \geq 0$ for all $\boldsymbol{y} \in \mathcal{M}$, we conclude that $\text{gap}^L_{\mathcal{M}}(\boldsymbol{y}) = 0$ if $\boldsymbol{y}$ is a first-order stationary point.

Next, we show that if $\text{gap}^L_{\mathcal{M}}(\boldsymbol{y}) = 0$, then $\boldsymbol{y}$ is a first-order stationary point. Suppose $\text{gap}^L_{\mathcal{M}}(\boldsymbol{y}) = 0$. Then,

$$\langle \nabla f(\boldsymbol{y}) - \nabla h(\boldsymbol{y}), \boldsymbol{x} - \boldsymbol{y} \rangle + g(\boldsymbol{x}) - g(\boldsymbol{y}) + \frac{L}{2}\|\boldsymbol{x} - \boldsymbol{y}\|^2 \geq 0,$$

for all $\boldsymbol{x} \in \mathcal{M}$. Consider $\boldsymbol{x} = \boldsymbol{y} + \alpha \boldsymbol{d}$ for an arbitrary feasible direction $\boldsymbol{d}$ (unit norm) and step-size $\alpha > 0$:

$$\alpha \langle \nabla f(\boldsymbol{y}) - \nabla h(\boldsymbol{y}), \boldsymbol{d} \rangle + g(\boldsymbol{y} + \alpha \boldsymbol{d}) - g(\boldsymbol{y}) + \frac{L}{2}\alpha^2 \geq 0, \quad \forall \alpha \boldsymbol{d} \,:\, \boldsymbol{y} + \alpha \boldsymbol{d} \in \mathcal{M}.$$

By dividing both sides by α and taking the limit as $\alpha \to 0^+$, we obtain

$$\langle \nabla f(\boldsymbol{y}) - \nabla h(\boldsymbol{y}), \boldsymbol{d} \rangle + \langle \boldsymbol{u}, \boldsymbol{d} \rangle \geq 0, \qquad \forall \boldsymbol{d} \,:\, \lim_{\alpha \to 0^+} (\boldsymbol{y} + \alpha \boldsymbol{d}) \in \mathcal{M},$$

for some $\boldsymbol{u} \in \partial g(\boldsymbol{y})$. Since $\mathcal{M}$ is a closed and convex set, we have $\lim_{\alpha \to 0^+}(\boldsymbol{y} + \alpha \boldsymbol{d}) \in \mathcal{M}$ for all $\boldsymbol{d} = \boldsymbol{x} - \boldsymbol{y}$ such that $\boldsymbol{x} \in \mathcal{M}$. Consequently, we have

$$\langle \nabla f(\boldsymbol{y}) - \nabla h(\boldsymbol{y})\boldsymbol{x} - \boldsymbol{y} \rangle + \langle \boldsymbol{u}, \boldsymbol{x} - \boldsymbol{y} \rangle \geq 0,$$

for all $\boldsymbol{x} \in \mathcal{M}$. This is equivalent to (4), thus concludes the proof. □

Remark 2. We can relate the gap function in Definition 1 to the Moreau-Yosida envelope: $\text{gap}^L_{\mathcal{M}}(\boldsymbol{y}) = 0$ if and only if $\boldsymbol{y}$ is a fixed point of the proximal mapping:

$$\boldsymbol{y} = \text{prox}_{\frac{1}{L}g + I_{\mathcal{M}}}\left((\boldsymbol{y} - \tfrac{1}{L}\left(\nabla f(\boldsymbol{y}) - \nabla h(\boldsymbol{y})\right)\right)$$

where $I_{\mathcal{M}}$ is the indicator function of the set $\mathcal{M}$. Moreover, $\text{gap}^L_{\mathcal{M}}(\boldsymbol{y})$ provides a stronger measure than the proximal gradient mapping residual, satisfying

$$\text{gap}^L_{\mathcal{M}}(\boldsymbol{y}) \geq \frac{L}{2}\left\|\boldsymbol{y} - \text{prox}_{\frac{1}{L}g + I_{\mathcal{M}}}\left((\boldsymbol{y} - \tfrac{1}{L}\left(\nabla f(\boldsymbol{y}) - \nabla h(\boldsymbol{y})\right)\right)\right\|^2.$$

The proof of this connection is deferred to the appendix.

We can now present the main convergence result.

Theorem 3. *Suppose $\boldsymbol{x}^1, \ldots, \boldsymbol{x}^K$ is a sequence generated by BDCA for solving problem (1). Then, the following bound holds:*

$$\min_{k \in \{1,\ldots,K\}} \mathbb{E}\left[\text{gap}^L_{\mathcal{M}}(\boldsymbol{x}^k)\right] \leq \frac{n}{K}\left(\phi(\boldsymbol{x}^1) - \phi^\star\right).$$

Proof. We begin by noting that $\forall x_{i_k} \in \mathcal{M}_{i_k}$, we have

$$\begin{aligned}
\phi(\boldsymbol{x}^{k+1}) &\leq f(\boldsymbol{x}^{k+1}) + g(\boldsymbol{x}^{k+1}) - h(\boldsymbol{x}^k) - \langle \nabla h(\boldsymbol{x}^k), \boldsymbol{x}^{k+1} - \boldsymbol{x}^k \rangle \\
&= f(\boldsymbol{x}^{k+1}) + g(\boldsymbol{x}^{k+1}) - h(\boldsymbol{x}^k) - \langle \nabla_{i_k} h(\boldsymbol{x}^k), x_{i_k}^{k+1} - x_{i_k}^k \rangle \\
&= \hat{\phi}(\boldsymbol{x}_{i_k}^{k+1} + \bar{\boldsymbol{x}}_i^k, \boldsymbol{x}^k) = \hat{\phi}_{i_k}(x_{i_k}^{k+1}, \boldsymbol{x}^k) \leq \hat{\phi}_{i_k}(x_{i_k}, \boldsymbol{x}^k) \\
&= f(\bar{\boldsymbol{x}}_{i_k}^k + \boldsymbol{x}_{i_k}) + g(\boldsymbol{x}^k) + g_{i_k}(x_{i_k}) - g_{i_k}(x_{i_k}^k) - h(\boldsymbol{x}^k) - \langle \nabla_{i_k} h(\boldsymbol{x}^k), x_{i_k} - x_{i_k}^k \rangle.
\end{aligned}$$

Here, the first inequality holds due to the convexity of h, while the second inequality follows from the definition of $x_{i_k}^{k+1}$ in the update rule of BDCA. Adding $f(\boldsymbol{x}^k)$ to both sides and rearranging, we get

$$f(\boldsymbol{x}^k) - f(\bar{\boldsymbol{x}}_{i_k}^k + \boldsymbol{x}_{i_k}) + g_{i_k}(x_{i_k}^k) - g_{i_k}(x_{i_k}) + \langle \nabla_{i_k} h(\boldsymbol{x}^k), x_{i_k} - x_{i_k}^k \rangle \leq \phi(\boldsymbol{x}^k) - \phi(\boldsymbol{x}^{k+1}).$$

Since the function f is L-smooth, we have

$$f(\bar{\boldsymbol{x}}_{i_k}^k + \boldsymbol{x}_{i_k}) \leq f(\boldsymbol{x}^k) + \langle \nabla_{i_k} f(\boldsymbol{x}^k), x_{i_k} - x_{i_k}^k \rangle + \frac{L}{2}\|x_{i_k} - x_{i_k}^k\|^2.$$

Combining the last two inequalities, we obtain

$$\langle \nabla_{i_k} f(\boldsymbol{x}^k) - \nabla_{i_k} h(\boldsymbol{x}^k), x_{i_k}^k - x_{i_k} \rangle + g_{i_k}(x_{i_k}^k) - g_{i_k}(x_{i_k}) - \frac{L}{2}\|x_{i_k} - x_{i_k}^k\|^2 \leq \phi(\boldsymbol{x}^k) - \phi(\boldsymbol{x}^{k+1}). \tag{5}$$

Let us denote by $\mathbb{E}_k$ the conditional expectation with respect to the random selection of i_k, given all the random choices in the previous iterations. Then,

$$\begin{aligned}
&\mathbb{E}_k\left[\langle \nabla_{i_k} f(\boldsymbol{x}^k) - \nabla_{i_k} h(\boldsymbol{x}^k), x_{i_k}^k - x_{i_k} \rangle + g_{i_k}(x_{i_k}^k) - g_{i_k}(x_{i_k}) - \frac{L}{2}\|x_{i_k} - x_{i_k}^k\|^2\right] \\
&= \frac{1}{n}\sum_{i=1}^n \left(\langle \nabla_i f(\boldsymbol{x}^k) - \nabla_i h(\boldsymbol{x}^k), x_i^k - x_i \rangle + g_i(x_i^k) - g_i(x_i) - \frac{L}{2}\|x_i - x_i^k\|^2\right) \\
&= \frac{1}{n}\left(\langle \nabla f(\boldsymbol{x}^k) - \nabla h(\boldsymbol{x}^k), \boldsymbol{x}^k - \boldsymbol{x} \rangle + g(\boldsymbol{x}^k) - g(\boldsymbol{x}) - \frac{L}{2}\|\boldsymbol{x} - \boldsymbol{x}^k\|^2\right).
\end{aligned}$$

Combining this with (5) gives the following bound, which holds for all $\boldsymbol{x} \in \mathcal{M}$:

$$\langle \nabla f(\boldsymbol{x}^k) - \nabla h(\boldsymbol{x}^k), \boldsymbol{x}^k - \boldsymbol{x} \rangle + g(\boldsymbol{x}^k) - g(\boldsymbol{x}) - \frac{L}{2}\|\boldsymbol{x} - \boldsymbol{x}^k\|^2 \leq n\phi(\boldsymbol{x}^k) - n\mathbb{E}_k\left[\phi(\boldsymbol{x}^{k+1})\right]. \tag{6}$$

Now, we maximize this inequality over $\boldsymbol{x} \in \mathcal{M}$ to get

$$\mathrm{gap}^L_{\mathcal{M}}(\boldsymbol{x}^k) \leq n\phi(\boldsymbol{x}^k) - n\mathbb{E}_k\left[\phi(\boldsymbol{x}^{k+1})\right].$$

Then, we take the expectation of both sides over the random choices in all iterations:

$$\mathbb{E}\left[\mathrm{gap}^L_{\mathcal{M}}(\boldsymbol{x}^k)\right] \leq n\mathbb{E}\left[\phi(\boldsymbol{x}^k)\right] - n\mathbb{E}\left[\phi(\boldsymbol{x}^{k+1})\right]. \tag{7}$$

Finally, we average this inequality over $k = 1, \ldots, K$:

$$\frac{1}{K}\sum_{k=1}^{K}\mathbb{E}\left[\mathrm{gap}^L_{\mathcal{M}}(\boldsymbol{x}^k)\right] \leq \frac{n}{K}\left(\phi(\boldsymbol{x}^1) - \mathbb{E}\left[\phi(\boldsymbol{x}^{K+1})\right]\right) \leq \frac{n}{K}\left(\phi(\boldsymbol{x}^1) - \phi^\star\right). \tag{8}$$

We complete the proof by noting that the minimum of $\mathrm{gap}^L_{\mathcal{M}}(\boldsymbol{x}^k)$ over $k = 1, \ldots, K$ is smaller than or equal to the average gap. □

Remark 4. The smoothness assumption of f is not required by Algorithm 1 itself, but only for the analysis. In particular, smoothness allows us to bound each block update of f by a quadratic model,

$$\langle \nabla_i f(\boldsymbol{x}^k), x_i - x_i^k \rangle + \tfrac{L}{2}\|x_i - x_i^k\|^2 \;\leq\; f(\boldsymbol{x}^k) - f(\bar{\boldsymbol{x}}_i^k + \boldsymbol{x}_i),$$

which effectively provides a block-separable structure for the analysis. This step is essential in our proof but does not play any role in the algorithm itself.

Remark 5. The rate in Theorem 3 is consistent with existing results for CD methods in convex optimization. For instance, Theorem 1 in [25] shows that $\mathbb{E}[f(\boldsymbol{x}^k)] - f^\star \leq \frac{2nL\|\boldsymbol{x}^1 - \boldsymbol{x}_\star\|^2}{k}$ for the CD method in unconstrained smooth convex minimization. In terms of the first-order stationarity condition, this implies

$$\mathbb{E}\left[\|\nabla f(\boldsymbol{x}^k)\|^2\right] \leq 2L\mathbb{E}\left[f(\boldsymbol{x}^k) - f^\star\right] \leq \frac{4nL^2\|\boldsymbol{x}^1 - \boldsymbol{x}_\star\|^2}{k}.$$

Theorem 3 yields a comparable result, since in this setting $\phi = f$ satisfies $f(\boldsymbol{x}^1) - f(\boldsymbol{x}^\star) \leq \frac{L}{2}\|\boldsymbol{x}^1 - \boldsymbol{x}^\star\|^2$, and the gap takes the form of $\mathrm{gap}^L(\boldsymbol{x}^k) = \frac{1}{2L}\|\nabla f(\boldsymbol{x}^k)\|^2$. Thus, we obtain

$$\min_{k\in\{1,\ldots,K\}}\mathbb{E}\left[\|\nabla f(\boldsymbol{x}^k)\|^2\right] \leq \frac{nL^2\|\boldsymbol{x}^1 - \boldsymbol{x}_\star\|^2}{K}.$$

Identifying the gap function in Definition 1 as a measure of first-order stationarity is an important technical component of our analysis. Notably, many standard stationarity measures are not suitable for nonsmooth nonconvex optimization problems, motivating recent efforts to explore and reassess alternative definitions [26,27]. For example, the distance measure

$$\mathrm{dist}(0, \nabla f(\boldsymbol{y}) + \partial g(\boldsymbol{y}) - \nabla h(\boldsymbol{y}) + \mathcal{N}_{\mathcal{M}}(\boldsymbol{y})),$$

or the 'Frank-Wolfe gap' using a subgradient $\boldsymbol{u} \in \partial g(\boldsymbol{y})$

$$\max_{\boldsymbol{x}\in\mathcal{M}}\langle \nabla f(\boldsymbol{y}) + \boldsymbol{u} - \nabla h(\boldsymbol{y}), \boldsymbol{y} - \boldsymbol{x}\rangle$$

are intractable, meaning that no algorithm can guarantee finding a point satisfying ϵ-closeness in these notions within a finite number of iterations.

Our results suggest that the challenge of identifying appropriate stationarity measures stems from the nonsmooth convex term, while the nonsmooth concave term remains benign. Although it may initially seem surprising, this can be explained by the fact that the concave term does not invalidate the smoothness-based quadratic upper bound, since

$$f(\boldsymbol{y}) \leq f(\boldsymbol{x}) + \langle \nabla f(\boldsymbol{x}), \boldsymbol{y} - \boldsymbol{x}\rangle + \frac{L}{2}\|\boldsymbol{x} - \boldsymbol{y}\|^2, \qquad -h(\boldsymbol{y}) \leq -h(\boldsymbol{x}) - \langle \nabla h(\boldsymbol{x}), \boldsymbol{y} - \boldsymbol{x}\rangle,$$

for all $\boldsymbol{x}$ and $\boldsymbol{y}$. Summing these two inequalities, we see that $f(\boldsymbol{y}) - h(\boldsymbol{y})$ behaves similarly to a smooth function in the analysis, as the quadratic upper bound is often the only property required from the smoothness assumption.

3.1.1. Guarantees under non-differentiable concave terms

When h is non-differentiable, we replace $\nabla h(\boldsymbol{y})$ in the definition of the gap with a subgradient $\boldsymbol{v} \in \partial h(\boldsymbol{y})$, and define

$$\mathrm{gap}^L_{\mathcal{M}}(\boldsymbol{y}) = \max_{\boldsymbol{x}\in\mathcal{M}}\min_{\boldsymbol{v}\in\partial h(\boldsymbol{y})}\left\{\langle \nabla f(\boldsymbol{y}) - \boldsymbol{v}, \boldsymbol{y} - \boldsymbol{x}\rangle + g(\boldsymbol{y}) - g(\boldsymbol{x}) - \frac{L}{2}\|\boldsymbol{x} - \boldsymbol{y}\|^2\right\}.$$

This choice ensures the gap remains well-defined, as it is sufficient that the condition holds for one subgradient of h at $\boldsymbol{y}$. When h is differentiable, this definition reduces to Definition 1.

The result below extends Lemma 1 to the setting where h is non-differentiable; the proof follows similar arguments and is deferred to the appendices.

Lemma 6. *Let $\mathcal{M} \subseteq \mathbb{R}^m$ be a closed convex set, and let $f, g, h : \mathcal{M} \to \mathbb{R}$ be lower-semicontinuous convex functions, where f is L-smooth. Then $\mathrm{gap}^L_{\mathcal{M}}(\boldsymbol{y}) \geq 0$ for all $\boldsymbol{y} \in \mathcal{M}$, and $\mathrm{gap}^L_{\mathcal{M}}(\boldsymbol{y}) = 0$ if and only if $\boldsymbol{y}$ is a critical point of problem (1), defined by the condition*

$$\left(\nabla f(\boldsymbol{y}) + \partial g(\boldsymbol{y}) + \mathcal{N}_{\mathcal{M}}(\boldsymbol{y})\right) \cap \partial h(\boldsymbol{y}) \neq \emptyset. \tag{9}$$

Theorem 3 continues to hold in the non-differentiable setting. The proof follows identically, except that $\nabla_{i_k} h(\boldsymbol{x}^k)$ is replaced by the i_k-th component (or block) of a subgradient $\boldsymbol{v}^k \in \partial h(\boldsymbol{x}^k)$, denoted by $v^k_{i_k}$. Any subgradient can be used, since the gap is defined via a minimization over the subdifferential set. The proof proceeds without further modification.

3.1.2. Faster rates under a generalized Polyak-Lojasiewicz condition

In the same spirit as [28], it is possible to prove a linear convergence rate for BDCA, conditioned on a generalized Polyak-Łojasiewicz (PL) inequality defined in terms of our gap function.

Theorem 7. *Assume that ϕ satisfies the generalized PL inequality, i.e., there exists $\mu \in (0, L]$ such that*

$$\phi(\boldsymbol{x}) - \phi(\boldsymbol{x}^*) \leq \frac{L}{\mu} \mathrm{gap}^L_{\mathcal{M}}(\boldsymbol{x}) \quad \textit{for all } \boldsymbol{x} \in \mathcal{M}. \tag{10}$$

Suppose $\boldsymbol{x}^1, \ldots, \boldsymbol{x}^K$ is a sequence generated by BDCA for solving problem (1). Then, the following bound holds:

$$\mathbb{E}\left[\phi(\boldsymbol{x}^{k+1})\right] - \phi(\boldsymbol{x}^*) \leq \left(1 - \frac{\mu}{nL}\right)^{k+1} \left(\mathbb{E}\left[\phi(\boldsymbol{x}^0)\right] - \phi(\boldsymbol{x}^*)\right).$$

Proof. The proof starts by following the same steps as that of Theorem 3. Then, rearranging (7), subtracting $\phi(\boldsymbol{x}^*)$, and applying (10), we obtain

$$\mathbb{E}\left[\phi(\boldsymbol{x}^{k+1})\right] - \phi(\boldsymbol{x}^*) \leq \mathbb{E}\left[\phi(\boldsymbol{x}^k)\right] - \phi(\boldsymbol{x}^*) - \frac{1}{n}\mathbb{E}\left[\mathrm{gap}^L_{\mathcal{M}}(\boldsymbol{x}^k)\right] \leq \left(1 - \frac{\mu}{nL}\right)\left(\mathbb{E}\left[\phi(\boldsymbol{x}^k)\right] - \phi(\boldsymbol{x}^*)\right).$$

Iterating this inequality concludes the proof. □

4. Block EM algorithm

The Expectation-Maximization (EM) algorithm is a cornerstone in statistical analysis for finding maximum likelihood estimates in the presence of incomplete data; it is widely used in applications such as mixture models, latent variable models, data imputation, and clustering [29,30]. EM shares strong connections with DCA [7,8]. In this section, we leverage these connections to develop a block EM algorithm based on our proposed BDCA.

Consider the negative log-likelihood minimization problem:

$$\min_{\boldsymbol{\theta}} \ \mathcal{L}(\boldsymbol{\theta}) := -\sum_{\boldsymbol{x} \in \mathcal{X}} \log P(\boldsymbol{x}|\boldsymbol{\theta}). \tag{11}$$

Note the change in notation: here, $\boldsymbol{x}$ denotes data points, while the decision variable $\boldsymbol{\theta} \in \Theta \subseteq \mathbb{R}^m$ represents the parameters of the underlying distribution. The dataset $\mathcal{X}$ has support size N, and $\mathcal{L}$ is the negative log-likelihood function. The distribution $P(\boldsymbol{x}|\boldsymbol{\theta})$ is defined through a hidden variable $\boldsymbol{y}$:

$$P(\boldsymbol{x}|\boldsymbol{\theta}) = \sum_{\boldsymbol{y}} P(\boldsymbol{x}, \boldsymbol{y}|\boldsymbol{\theta}),$$

where $P(\boldsymbol{x}, \boldsymbol{y}|\boldsymbol{\theta})$ denotes the complete-data likelihood. The presence of hidden variables complicates (11), making standard optimization techniques inapplicable.

The EM algorithm, for solving this problem, iteratively alternates between two steps – Expectation (E-step) and Maximization (M-step) – to improve the likelihood of the model parameters until convergence:

$$Q(\boldsymbol{\theta}; \boldsymbol{\theta}^k) = \mathbb{E}_{P(\boldsymbol{y}|\boldsymbol{x}, \boldsymbol{\theta}^k)}\left[\log P(\boldsymbol{x}, \boldsymbol{y}|\boldsymbol{\theta})\right] \quad \text{(E-step)}$$

$$\boldsymbol{\theta}^{k+1} = \arg\max_{\boldsymbol{\theta} \in \Theta} Q(\boldsymbol{\theta}; \boldsymbol{\theta}^k) \quad \text{(M-step)}$$

Here, $P(\boldsymbol{y}|\boldsymbol{x}, \boldsymbol{\theta}^k)$ is the posterior distribution of the latent variables given the observed data and the current parameter estimate $\boldsymbol{\theta}^k$. In the E-step, the expected value of the complete-data log-likelihood is computed with respect to the conditional distribution of the latent variables given the observed data and the current parameter estimates. In the M-step, this expected log-likelihood is maximized to update the parameters.

For the broad class of natural exponential family distributions, EM is a specific instance of DCA.

Definition 2. The natural exponential family of distributions with a parameter $\boldsymbol{\theta} \in \Theta$ is characterized by the following probability mass (or density) function:

$$P(\boldsymbol{x}, \boldsymbol{y}|\boldsymbol{\theta}) = \frac{\varphi(\boldsymbol{x}, \boldsymbol{y}) e^{\langle \boldsymbol{\theta}, T(\boldsymbol{x}, \boldsymbol{y}) \rangle}}{\sum_{\boldsymbol{x}, \boldsymbol{y}} \varphi(\boldsymbol{x}, \boldsymbol{y}) e^{\langle \boldsymbol{\theta}, T(\boldsymbol{x}, \boldsymbol{y}) \rangle}}, \tag{12}$$

where φ is a positive function (base measure) and T denotes the sufficient statistic.

It is easy to see that the problem (11) with a natural exponential family distribution (12) can be written as a DC program, since the log-likelihood function can be decomposed as $\mathcal{L} = f - h$, where the terms are given by

$$\begin{aligned} f(\boldsymbol{\theta}) &= N \log\Big(\sum_{\boldsymbol{x},\boldsymbol{y}} \varphi(\boldsymbol{x},\boldsymbol{y}) e^{\langle \boldsymbol{\theta}, T(\boldsymbol{x},\boldsymbol{y})\rangle}\Big), \\ h(\boldsymbol{\theta}) &= \sum_{x\in\mathcal{X}} \log\Big(\sum_{\boldsymbol{y}} \varphi(\boldsymbol{x},\boldsymbol{y}) e^{\langle \boldsymbol{\theta}, T(\boldsymbol{x},\boldsymbol{y})\rangle}\Big). \end{aligned} \tag{13}$$

Both terms are convex and smooth, owing to their log-sum-exp structure.

Applying DCA to this problem leads to the following update rule:

$$\boldsymbol{\theta}^{k+1} \in argmin_{\theta} \;\; N \log\left(\sum_{\boldsymbol{x},\boldsymbol{y}} \varphi(\boldsymbol{x},\boldsymbol{y}) e^{\langle \boldsymbol{\theta}, T(\boldsymbol{x},\boldsymbol{y})\rangle}\right) - \sum_{x\in\mathcal{X}} \frac{\sum_{\boldsymbol{y}} \varphi(\boldsymbol{x},\boldsymbol{y}) \langle \boldsymbol{\theta}, T(\boldsymbol{x},\boldsymbol{y})\rangle e^{\langle \boldsymbol{\theta}^k, T(\boldsymbol{x},\boldsymbol{y})\rangle}}{\sum_{\boldsymbol{y}} \varphi(\boldsymbol{x},\boldsymbol{y}) e^{\langle \boldsymbol{\theta}^k, T(\boldsymbol{x},\boldsymbol{y})\rangle}}. \tag{14}$$

Note that the dataset size N can be expressed as:

$$N = \sum_{x\in\mathcal{X},\boldsymbol{y}} \frac{\varphi(\boldsymbol{x},\boldsymbol{y}) e^{\langle \boldsymbol{\theta}^k, T(\boldsymbol{x},\boldsymbol{y})\rangle}}{\sum_{\boldsymbol{y}} \varphi(\boldsymbol{x},\boldsymbol{y}) e^{\langle \boldsymbol{\theta}^k, T(\boldsymbol{x},\boldsymbol{y})\rangle}}.$$

Also, we can write $\langle \boldsymbol{\theta}, T(\boldsymbol{x},\boldsymbol{y})\rangle = \log(e^{\langle \boldsymbol{\theta}, T(\boldsymbol{x},\boldsymbol{y})\rangle})$, and add $-N\log(\varphi(\boldsymbol{x},\boldsymbol{y}))$ to (14), as it is constant with respect to $\boldsymbol{\theta}$. This gives:

$$\boldsymbol{\theta}^{k+1} \in argmin_{\theta} \;\; \sum_{x\in\mathcal{X},\boldsymbol{y}} \frac{\varphi(\boldsymbol{x},\boldsymbol{y}) e^{\langle \boldsymbol{\theta}^k, T(\boldsymbol{x},\boldsymbol{y})\rangle}}{\sum_{\boldsymbol{y}} \varphi(\boldsymbol{x},\boldsymbol{y}) e^{\langle \boldsymbol{\theta}^k, T(\boldsymbol{x},\boldsymbol{y})\rangle}} \log\left(\sum_{\boldsymbol{x},\boldsymbol{y}} \varphi(\boldsymbol{x},\boldsymbol{y}) e^{\langle \boldsymbol{\theta}, T(\boldsymbol{x},\boldsymbol{y})\rangle}\right) - \sum_{x\in\mathcal{X},\boldsymbol{y}} \frac{\varphi(\boldsymbol{x},\boldsymbol{y}) e^{\langle \boldsymbol{\theta}^k, T(\boldsymbol{x},\boldsymbol{y})\rangle}}{\sum_{\boldsymbol{y}} \varphi(\boldsymbol{x},\boldsymbol{y}) e^{\langle \boldsymbol{\theta}^k, T(\boldsymbol{x},\boldsymbol{y})\rangle}} \log\Big(\varphi(\boldsymbol{x},\boldsymbol{y}) e^{\langle \boldsymbol{\theta}, T(\boldsymbol{x},\boldsymbol{y})\rangle}\Big),$$

or equivalently,

$$\boldsymbol{\theta}^{k+1} \in argmin_{\theta} \sum_{x\in\mathcal{X}} \sum_{\boldsymbol{y}} \frac{\varphi(\boldsymbol{x},\boldsymbol{y}) e^{\langle \boldsymbol{\theta}^k, T(\boldsymbol{x},\boldsymbol{y})\rangle}}{\sum_{\boldsymbol{y}} \varphi(\boldsymbol{x},\boldsymbol{y}) e^{\langle \boldsymbol{\theta}^k, T(\boldsymbol{x},\boldsymbol{y})\rangle}} \left[\log\left(\frac{\sum_{\boldsymbol{x},\boldsymbol{y}} \varphi(\boldsymbol{x},\boldsymbol{y}) e^{\langle \boldsymbol{\theta}, T(\boldsymbol{x},\boldsymbol{y})\rangle}}{\varphi(\boldsymbol{x},\boldsymbol{y}) e^{\langle \boldsymbol{\theta}, T(\boldsymbol{x},\boldsymbol{y})\rangle}}\right)\right], \tag{15}$$

which is exactly the EM algorithm for the exponential family. This can be seen through Definition 2 and noting that

$$P(\boldsymbol{x},\boldsymbol{y}|\boldsymbol{\theta}) = \frac{\varphi(\boldsymbol{x},\boldsymbol{y}) e^{\langle \boldsymbol{\theta}, T(\boldsymbol{x},\boldsymbol{y})\rangle}}{\sum_{\boldsymbol{x},\boldsymbol{y}} \varphi(\boldsymbol{x},\boldsymbol{y}) e^{\langle \boldsymbol{\theta}, T(\boldsymbol{x},\boldsymbol{y})\rangle}}, \quad P(\boldsymbol{y}|\boldsymbol{x},\boldsymbol{\theta}^k) = \frac{\varphi(\boldsymbol{x},\boldsymbol{y}) e^{\langle \boldsymbol{\theta}^k, T(\boldsymbol{x},\boldsymbol{y})\rangle}}{\sum_{\boldsymbol{y}} \varphi(\boldsymbol{x},\boldsymbol{y}) e^{\langle \boldsymbol{\theta}^k, T(\boldsymbol{x},\boldsymbol{y})\rangle}}.$$

We then design the Block EM method by addressing the same problem using BDCA. The resulting algorithm is presented in Algorithm 2. Notably, we can transfer the guarantees established in Theorem 3 to provide non-asymptotic convergence results for Block EM.

Corollary 8. *Suppose there exists a sequence $\boldsymbol{\theta}^k, k = 1\ldots, K$ generated by the Block EM algorithm. Then,*

$$\min_{k\in\{1,\ldots,K\}} \mathbb{E}\left[\|\nabla\mathcal{L}(\boldsymbol{\theta}^k)\|^2\right] \le \frac{2nL}{K}\big(\mathcal{L}(\boldsymbol{\theta}^1) - \mathcal{L}(\boldsymbol{\theta}^\star)\big).$$

Proof. In this setting, the gap function reduces to

$$\max_{\boldsymbol{\theta}} \langle \underbrace{\nabla f(\boldsymbol{\theta}^k) - \nabla h(\boldsymbol{\theta}^k)}_{\nabla\mathcal{L}(\boldsymbol{\theta}^k)}, \boldsymbol{\theta}^k - \boldsymbol{\theta}\rangle - \frac{L}{2}\|\boldsymbol{\theta}^k - \boldsymbol{\theta}\|^2.$$

The solution to this max problem is $\boldsymbol{\theta} = \boldsymbol{\theta}^k - \frac{1}{L}\nabla\mathcal{L}(\boldsymbol{\theta})$, leading to $\text{gap}^L(\boldsymbol{\theta}^k) = \frac{1}{2L}\|\nabla\mathcal{L}(\boldsymbol{\theta}^k)\|^2$. Substituting this into Theorem 3 yields the desired result. □

Algorithm 2 Block EM algorithm.

Input: total iterations K, initialize the probability distribution P using vector $\boldsymbol{\theta}^{(0)}$ (P belongs to the exponential family), and number of blocks n
for $k = 1$ **to** K **do**
  Randomly choose i_k in $[1, \ldots, n]$ with uniform distribution
  Update $\hat{P}^{k+1}(\boldsymbol{y}) = \frac{P(\boldsymbol{x},\boldsymbol{y}|\boldsymbol{\theta}^k)}{\sum_{\boldsymbol{y}} P(\boldsymbol{x},\boldsymbol{y}|\boldsymbol{\theta}^k)}$
  Update $\theta_{i_k}^{k+1} = \arg\min_{\theta_{i_k}} -\sum_{\boldsymbol{x}\in\mathcal{X},\boldsymbol{y}} \hat{P}^{k+1}(\boldsymbol{y}) \log P(\boldsymbol{x},\boldsymbol{y}|\theta_1^k, \ldots, \theta_{i_k}, \ldots, \theta_n^k)$
  Set $\boldsymbol{\theta}^{k+1} = \bar{\boldsymbol{\theta}}_{i_k}^k + \boldsymbol{\theta}_{i_k}^{k+1}$
end for
Output: $\boldsymbol{\theta}^{K+1}$

Remark 9. The results in Corollary 8 are consistent with the $\mathcal{O}(1/k)$ convergence rate guarantee for the standard EM algorithm, as provided by [12].

Remark 10. The standard EM algorithm achieves an $\mathcal{O}(1/k)$ convergence rate in Kullback-Leibler divergence, as shown in [13]. It is worth noting that the Lipschitz constant and the Euclidean distance generating function are not utilized in BDCA; they appear in the analysis only through the definition of the gap function. Consequently, we believe our analysis can be extended to the general setting of Bregman divergence.

5. Numerical experiments

In this section, we demonstrate the empirical performance of BDCA through numerical experiments in logistic regression with nonconvex sparsity regularization and nonconvex quadratic programming with negative ℓ_1 regularization. Additionally, we examine the performance of Block EM on the one-bit MIMO channel signal recovery problem in Appendix F. All experiments were conducted using MATLAB R2023b on an Intel(R) Xeon(R) Silver 4210R CPU @ 2.40 GHz.

5.1. Logistic regression with nonconvex sparsity regularization

The first experiment replicates the logistic regression with nonconvex sparsity regularization in [20]:

$$\min_{\boldsymbol{x}\in\mathbb{R}^m} \quad \frac{1}{N}\sum_{i=1}^{N}\log(1+e^{-b_i\langle \boldsymbol{a}_i,\boldsymbol{x}\rangle}) + \lambda(\|\boldsymbol{x}\|_1 - \|\boldsymbol{x}\|_Q) \tag{16}$$

where $\|\boldsymbol{x}\|_Q$ denotes the largest Q-norm , defined as the sum of the absolute values of the Q largest components of $\boldsymbol{x}$, and $(\boldsymbol{a}_i, b_i)$ represent the datapoints.

Logistic loss is convex and smooth, and we denote by L its smoothness constant. We compare the performance of DCA and BDCA, using the following decomposition of the objective function:

$$f(\boldsymbol{x}) = \frac{L}{2}\|\boldsymbol{x}\|^2, \quad g(\boldsymbol{x}) = \lambda\|\boldsymbol{x}\|_1,$$
$$h(\boldsymbol{x}) = \frac{L}{2}\|\boldsymbol{x}\|^2 - \frac{1}{N}\sum_{i=1}^{N}\log(1+e^{-b_i\langle \boldsymbol{a}_i,\boldsymbol{x}\rangle}) + \lambda\|\boldsymbol{x}\|_Q.$$

which results in closed form DCA subproblems of the proximal (coordinate) descent type updates.

We conduct experiments with the RCV1 ($N = 20,242, m = 47,236, s = 0.5\ \%$) and REAL-SIM ($N = 7,02,309, m = 20,958, s = 0.25\ \%$) datasets from the LIBSVM library [31]. We set $\lambda = 0.1/m$ and $\lambda = 10/m$, and use blocks of size 1000.

The results are shown in Figs. 1 and 2. In this example, BDCA and DCA perform similarly in terms of the number of full passes (iteration $\times$ block-size/m) for both the objective and the gap function (as defined in Definition 1). However, BDCA avoids solving full-scale subproblems, making it suitable for devices with limited computational power. It also offers greater potential for parallelization through asynchronous updates, which could be a valuable direction for future research.

5.2. Nonconvex quadratic programming with negative ℓ_1 regularization

The second experiment examines nonconvex quadratic programming with a negative ℓ_1 norm regularization term to promote binary solutions, formulated as follows:

$$\min_{\|\boldsymbol{x}\|_\infty\le 1} \quad \boldsymbol{x}^T\boldsymbol{Q}\boldsymbol{x} - \lambda\|\boldsymbol{x}\|_1, \tag{17}$$

where $\boldsymbol{Q}\in\mathbb{R}^{m\times m}$ is the symmetric (not necessarily positive semidefinite) cost matrix. We conduct experiments using the Gset benchmark datasets [32], setting the negative adjacency matrix as the cost matrix $\boldsymbol{Q}$. We set the regularization parameter $\lambda = \|\boldsymbol{Q}\|_F/\sqrt{n}$, where $\|\cdot\|_F$ represents the Frobenius norm.

In this formulation, we first consider the following decomposition: $f(\boldsymbol{x}) = \boldsymbol{x}^T\boldsymbol{Q}\boldsymbol{x}$, $g(\boldsymbol{x}) = 0$, $h(\boldsymbol{x}) = \lambda\|\boldsymbol{x}\|_1$, and $\mathcal{M} = [-1,1]^m$ is the separable constraint set. It is important to note that $f(\boldsymbol{x})$ is nonconvex, but this does not affect the validity of our guarantees. It is easy to verify that our guarantees hold for any smooth $f(\boldsymbol{x})$ as long as the subproblems can be solved globally. While nonconvex $f(\boldsymbol{x})$ makes the subproblems nonconvex, solving them globally along a single coordinate is tractable, as demonstrated in this example.

For DCA, the subproblem becomes:

$$\min_{\|\boldsymbol{x}\|_\infty\le 1} \quad \boldsymbol{x}^T\boldsymbol{Q}\boldsymbol{x} - \lambda\langle \operatorname{sign}(\boldsymbol{x}^k), \boldsymbol{x}-\boldsymbol{x}^k\rangle,$$

which is generally NP-hard. However, using BDCA with block size 1, the subproblem simplifies to

$$\min_{x_i\in[-1,1]} \quad (\bar{\boldsymbol{x}}_i^k + \boldsymbol{x}_i)^T\boldsymbol{Q}(\bar{\boldsymbol{x}}_i^k + \boldsymbol{x}_i) - \lambda\langle \operatorname{sign}(\boldsymbol{x}^k), \bar{\boldsymbol{x}}_i^k + \boldsymbol{x}_i - \boldsymbol{x}^k\rangle.$$

Expanding this expression gives

$$x_i^{k+1} = argmin_{x_i\in[-1,1]} \ ax_i^2 + bx_i + c$$

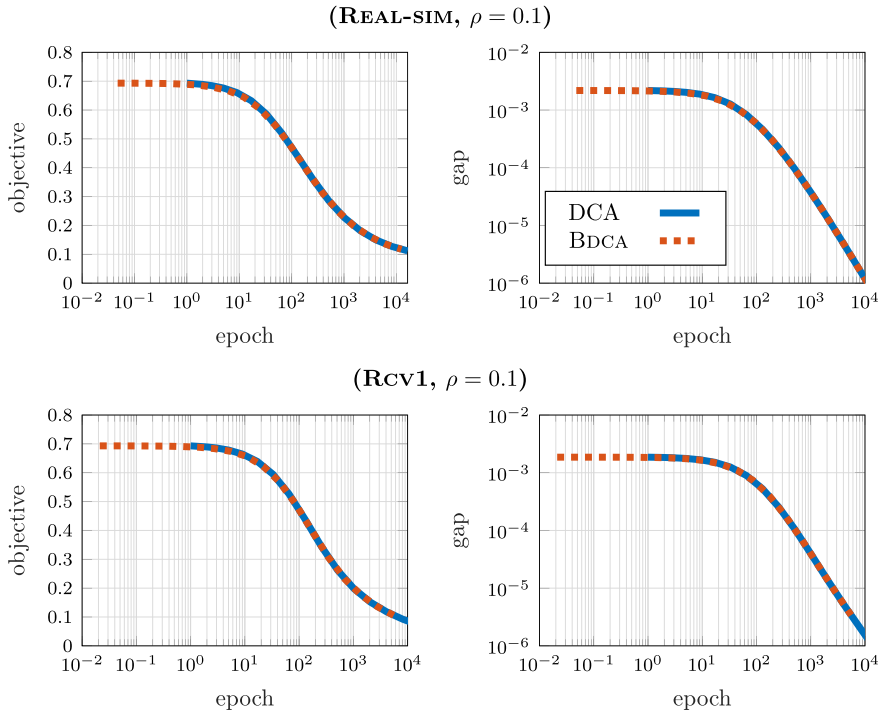

Fig. 1. Comparison of DC and BDCA for logistic regression with nonconvex sparsity regularization in (16), in terms of the objective value and the gap (as defined in Definition 1), on RCV1 and REAL-SIM datasets, with $\rho = 0.1$. Epoch represents iteration × block-size/m.

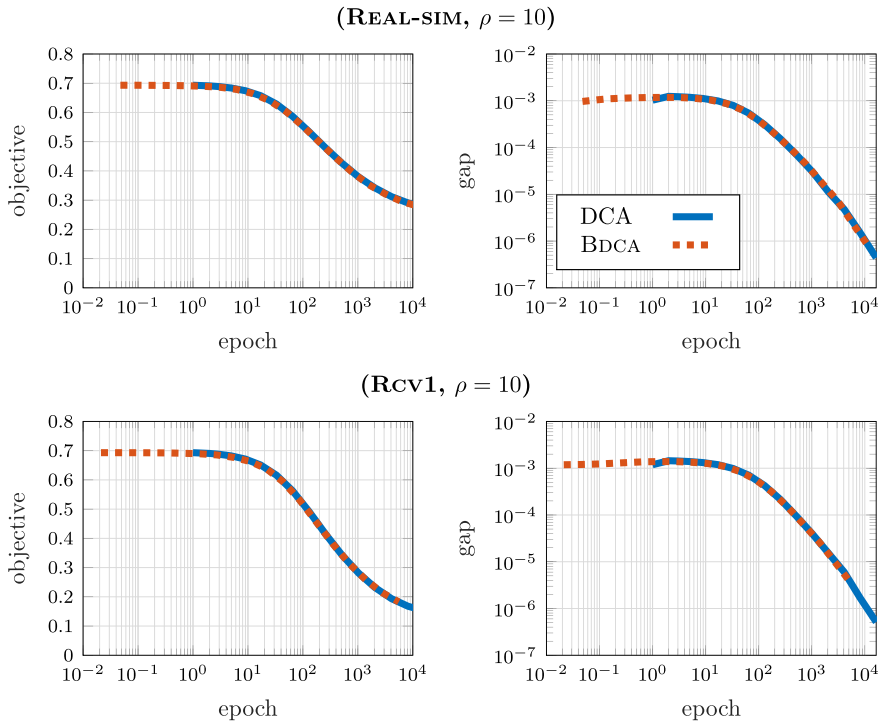

Fig. 2. Comparison of DC and BDCA for logistic regression with nonconvex sparsity regularization in (16), in terms of the objective value and the gap (as defined in Definition 1), on RCV1 and REAL-SIM datasets, with $\rho = 10$. Epoch represents iteration × block-size/m.

Table 1
Comparison of BDCA, DCA, and RCSD performance in terms of objective value and computation time in nonconvex quadratic programming with negative ℓ_1 regularization (17).

Dataset	Obj (BDCA)	Obj (DCA)	Obj (RCSD)	Time (BDCA)	Time (DCA)	Time (RCSD)
G1	−43,891.1	−43,891.1	−43,891.1	2.02	2.37	1.19
G2	−43,891.1	−43,891.1	−43,891.1	2.01	2.59	1.15
G3	−43,891.1	−43,891.1	−43,891.1	2.11	2.32	0.98
G4	−43,891.1	−43,891.1	−43,891.1	2.1	2.53	1.06
G5	−43,891.1	−43,891.1	−43,891.1	1.55	2.05	1.38
G6	−11,703.1	−11,243.1	−11,363.1	2.06	5.18	4.3
G7	−11,635.1	−11,787.1	−11,767.1	1.68	4.92	8.88
G8	−11,247.1	−11,691.1	−11,659.1	1.74	9.81	10.11
G9	−11,843.1	−11,299.1	−11,683.1	2.11	4.98	9.64
G10	−12,107.1	−11,291.1	−11,643.1	2.87	4.12	5.13
G11	−3212	−3204	−3244	0.63	3.01	0.62
G12	−3256	−3184	−3184	0.58	2.84	0.5
G13	−3260	−3132	−3196	0.6	3.05	0.56
G14	−12,128.51	−12,128.51	−12,128.51	1.27	10.31	1.99
G15	−7356.86	−6604.86	−12,052.86	2.13	15.16	4.05
G16	−6650.08	−6438.08	−6610.08	1.1	9.9	4.16
G17	−12,066.62	−7486.62	−7038.62	1.46	17.42	2.38
G18	−5556.51	−5680.51	−5596.51	0.93	14.47	2
G19	−5576.86	−5564.86	−5568.86	1.12	13.37	1.91
G20	−5538.08	−5346.08	−5482.08	1.25	7.97	1.99
G21	−5626.62	−5438.62	−5430.62	1.23	8.34	2.88
G22	−48,922.04	−48,922.04	−48,922.04	6.36	60.94	8.47
G23	−48,922.04	−48,922.04	−48,922.04	6.24	37.38	5.65
G24	−48,922.04	−18,238.04	−48,922.04	4.77	41.08	9.57
G25	−48,922.04	−17,910.04	−48,922.04	4.7	40.1	26.15
G26	−48,922.04	−48,922.04	−48,922.04	4.79	86.45	8.11
G27	−18,566.04	−18,054.04	−18,042.04	5	35.58	31.21
G28	−18,594.04	−18,154.04	−18,386.04	4.64	41.69	16.08
G29	−18,442.04	−17,842.04	−18,054.04	4.94	50.26	30.09
G30	−18,350.04	−17,958.04	−18,702.04	4.65	47.92	13.92
G31	−18,690.04	−18,074.04	−18,690.04	4.49	45.72	11.42
G32	−8172	−8076	−8236	1.37	15.26	1.95
G33	−8212	−7996	−8140	1.13	15.67	2.3
G34	−8232	−8024	−8120	1.13	16.09	2.24
G35	−16,999.82	−30,419.82	−30,419.82	2.46	74.96	10.47
G36	−30,392.32	−16,928.32	−30,392.32	3.54	65.26	10.91
G37	−30,267.86	−30,435.86	−30,435.86	4.73	63.71	14.23
G38	−30,422.11	−30,422.11	−30,422.11	4.04	132.32	8.25
G39	−13,811.82	−13,551.82	−13,651.82	2.33	41.84	12.16
G40	−13,792.32	−13,584.32	−13,912.32	2.32	56.62	16.06
G41	−13,915.86	−13,815.86	−13,683.86	2.32	45.32	11.78
G42	−14,386.11	−13,806.11	−13,862.11	2.36	52.1	14.3
G43	−24,449.9	−24,449.9	−24,449.9	0.92	4.52	1.57
G44	−24,449.9	−24,449.9	−24,449.9	0.98	4.11	1.5
G45	−24,449.9	−24,449.9	−24,449.9	0.96	3.78	1.51
G46	−24,449.9	−24,449.9	−24,449.9	0.97	3.9	1.61
G47	−24,449.9	−24,449.9	−24,449.9	0.96	4.1	1.44
G48	−13,584	−13,112	−13,472	2.96	63.35	5.12
G49	−13,512	−13,184	−13,320	2.32	213.11	5
G50	−13,536	−13,168	−13,456	2.96	45.9	5.56
G51	−15,255.73	−15,255.73	−15,255.73	0.94	18.71	1.55
G52	−15,271.77	−9083.77	−15,271.77	1.26	17.97	2.74
G53	−15,267.19	−8131.19	−8327.19	1.25	6.76	3.78
G54	−15,271.77	−8239.77	−15,271.77	1.08	8.65	3.86
G55	−21,687.45	−21,039.45	−21,863.45	18.82	280.19	72.61
G56	−21,939.45	−21,055.45	−21,579.45	14.63	359.52	63.93
G57	−20,196	−19,796	−20,100	7.62	936.13	11.47
G58	−76,335.93	−76,335.93	−76,335.93	29.66	753.15	78.43
G59	−35,019.93	−34,219.93	−34,871.93	18.88	590.34	179.67
G60	−30,462.26	−29,770.26	−30,386.26	26.31	836.66	166.26
G61	−30,094.26	−29,406.26	−29,966.26	27.97	882.6	170.17
G62	−28,296	−27,984	−28,232	13.11	2398.02	31.71
G63	−107,010.03	−61,763	−107,010.03	117.79	4218.65	354.19
G64	−49,222.03	−48,250.03	−48,670.03	43.75	1535.71	365.12
G65	−32,372	−32,156	−32,532	17.69	3468.09	32.9
G66	−36,720	−36,360	−36,704	25.39	1011.99	55.51
G67	−40,428	−40,052	−40,402	33.96	6298.46	63.5

for scalars $a = q_{ii}$, $b = 2\boldsymbol{q}_i^\top \bar{\boldsymbol{x}}_i^k - \lambda \text{sign}(x_i^k)$, and $c = (\bar{\boldsymbol{x}}_i^k)^T \boldsymbol{Q} \bar{\boldsymbol{x}}_i^k + \lambda \text{sign}(x_i^k) x_i^k$, where q_{ii} denotes the ith diagonal entry of $\boldsymbol{Q}$ and $\boldsymbol{q}_i$ is the ith column. This expression can be efficiently solved, highlighting the practical advantages of BDCA. Even in scenarios where DCA involves NP-hard subproblems, BDCA remains efficient.

Because the DCA subproblem corresponding to the above decomposition is intractable, we instead apply DCA using a different splitting. Specifically, we decompose $\boldsymbol{Q} = \boldsymbol{Q}_P + \boldsymbol{Q}_N$ where $\boldsymbol{Q}_P$ and $\boldsymbol{Q}_N$ represent the positive and negative semidefinite components of $\boldsymbol{Q}$, respectively. Under this decomposition, one could redefine $f(\boldsymbol{x}) = \boldsymbol{x}^T \boldsymbol{Q}_P \boldsymbol{x}$ and $h(\boldsymbol{x}) = \lambda \|\boldsymbol{x}\|_1 - \boldsymbol{x}^T \boldsymbol{Q}_N \boldsymbol{x}$. However, this approach requires eigendecomposition of $\boldsymbol{Q}$, which can be computationally expensive when m is large. Moreover, $\boldsymbol{Q}_P$ and $\boldsymbol{Q}_N$ may lose useful structural properties. For example, when $\boldsymbol{Q}$ is sparse, $\boldsymbol{Q}_P$ and $\boldsymbol{Q}_N$ may become dense.

We also compare the performance of RCSD from [20]. Notably, RCSD can be viewed as a special case of BDCA applied to the following reformulation of the problem:

$$\min_{\|\boldsymbol{x}\|_\infty \le 1} \quad \frac{L}{2}\|\boldsymbol{x}\|_2^2 - \lambda \|\boldsymbol{x}\|_1 - (\frac{L}{2}\|\boldsymbol{x}\|_2^2 - \boldsymbol{x}^T \boldsymbol{Q} \boldsymbol{x}),$$

where $L = 2\|\boldsymbol{Q}\|$ is the smoothness constant. We compared DCA, BDCA, and RCSD in terms of objective values at the local solutions found by the methods, and the computational time needed to reach such solutions.

Both algorithms are initialized from the same random starting point where entries are drawn from a Gaussian distribution, and run until the gap reaches 10^{-6}. The results are summarized in Table 1. Specifically, in 45 cases, BDCA finds a better local optimum than DCA, while DCA performs better in only 5 datasets. Also, in 31 cases BDCA outperforms RCSD and for 11 datasets RCSD finds a better solution. In most cases, BDCA is faster than both DCA and RCSD.

6. Conclusions

We investigated a general class of DC problems involving smooth and separable nonsmooth terms, and proposed a randomized block-coordinate DC algorithm (BDCA) specifically designed to solve such problems efficiently. We established convergence guarantees for BDCA with a rate of $\mathcal{O}(n/K)$, where the algorithm performs K iterations over n coordinate blocks. Leveraging the connection between the DCA and the EM method, we proposed a randomized block-coordinate EM algorithm (Block EM), where convergence guarantees are directly inherited from the DCA as a special case. DCA and EM have numerous applications in data science and machine learning; hence, scalable alternatives like BDCA and Block EM have the potential for significant impact.

CRediT authorship contribution statement

Hoomaan Maskan: Writing – review & editing, Writing – original draft, Software, Methodology, Investigation, Formal analysis. **Paniz Halvachi:** Writing – original draft, Software, Formal analysis. **Suvrit Sra:** Supervision, Methodology, Investigation, Formal analysis, Conceptualization. **Alp Yurtsever:** Writing – review & editing, Writing – original draft, Visualization, Supervision, Software, Project administration, Methodology, Formal analysis, Conceptualization.

Acknowledgement

Hoomaan Maskan and Alp Yurtsever were supported by the Wallenberg AI, Autonomous Systems and Software Program (WASP) funded by the Knut and Alice Wallenberg Foundation. Suvrit Sra acknowledges generous support from the Alexander von Humboldt Foundation. We thank the High Performance Computing Center North (HPC2N) at Umeå University for providing computational resources and valuable support during test and performance runs. The computations were enabled by resources provided by the National Academic Infrastructure for Supercomputing in Sweden (NAISS), partially funded by the Swedish Research Council through grant agreement no. 2022–06725.

Appendix A. Regularized EM

In Section 4, we discussed the relationship between the EM algorithm and DCA. To maintain simplicity and consistency with the standard EM algorithm, we focused on minimizing the negative log-likelihood function. However, it is common practice to solve regularized log-likelihood problems, where a regularization term is introduced to prevent overfitting, improve generalization, or promote specific structures in the solution [33,34]. In this context, the Regularized EM (REM) algorithm seeks to optimize the penalized negative log-likelihood problem:

$$\min_{\theta \in \Theta} \ \hat{\mathcal{L}}(\theta) := \sum_{\boldsymbol{x} \in \mathcal{X}} \sum_{\boldsymbol{y}} -\log P(\boldsymbol{x}, \boldsymbol{y}|\theta) + \lambda R(\theta), \tag{A.1}$$

where R is a convex regularization function with block-separable structure $R(\theta) = \sum_{i=1}^n R_i(\theta_i)$, and $\lambda \ge 0$ is the regularization parameter. Similarly, we assume Θ can be decomposed as $\Theta_1 \times \cdots \times \Theta_n$, where the components, are such that $\theta \in \Theta$ if and only if $\theta_i \in \Theta_i$ for all $i = 1, \dots, n$.

For the exponential family of distributions, as defined in Definition 2, we can express (A.1) as a specific instance of the problem template

$$\min_{\theta \in \Theta} \ f(\theta) + g(\theta) - h(\theta), \tag{A.2}$$

Algorithm 3 Block REM algorithm.

Input: regularization parameter λ, total iterations K, initialize the probability distribution P using vector $\theta^{(0)}$ (P belongs to the exponential family), and number of blocks n
for $k = 1$ **to** K **do**
Randomly choose i_k in $[1, \dots, n]$ with uniform distribution
Update $\hat{P}^{k+1}(\boldsymbol{y}) = \frac{P(\boldsymbol{x},\boldsymbol{y}|\theta^k)}{\sum_{\boldsymbol{y}} P(\boldsymbol{x},\boldsymbol{y}|\theta^k)}$
Update $\theta_{i_k}^{k+1} = \underset{\theta_{i_k}\in\Theta_{i_k}}{\arg\min} \sum_{\boldsymbol{x}\in\mathcal{X},\boldsymbol{y}} -\hat{P}^{k+1}(\boldsymbol{y}) \log P(\boldsymbol{x},\boldsymbol{y}|\theta_1^k, \dots, \theta_{i_k}, \dots, \theta_n^k) + \lambda R_{i_k}(\theta_{i_k})$
Set $\theta^{k+1} = \bar{\theta}_{i_k}^k + \theta_{i_k}^{k+1}$
end for
Output: θ^{K+1}

where f and h are as defined in (13), and $g(\theta) = \lambda R(\theta)$.

Employing BDCA to solve (A.2) results in the Block REM method shown in Algorithm 3. The following corollary, which is a direct consequence of Theorem 3, formulates the convergence behaviour of the Block REM method in terms of the introduced gap function (See Definition 1).

Corollary 11. *Suppose there exists a sequence $\theta^k, k = 1 \dots, K$ generated by the Block REM algorithm. Then,*

$$\min_{k\in\{1,\dots,K\}} \mathbb{E}\left[\text{gap}_\Theta^L(\theta^k)\right] \le \frac{n}{K}\left(\hat{\mathcal{L}}(\theta^1) - \hat{\mathcal{L}}^\star\right).$$

Appendix B. BDCA recovers coordinate (Sub)Gradient descent

Randomized coordinate subgradient descent method (RCSD) is a block coordinate proximal type subgradient algorithm for similar DC nonconvex problems as ours (see problem (1)). RCSD assumes an additional condition on the function g to have a simple structure [20]. Here, we briefly present RCSD and its convergence guarantee. Later, we attempt to show how BDCA recovers RCSD.

RCSD is a block coordinate subgradient method which iteratively updates some random coordinates while keeping the rest of the coordinates fixed (similar to BDCA). To formally present the method, we need block proximal mapping given by

$$\mathcal{P}_i(\bar{x}_i, y_i, \gamma_i) = argmin_{x\in\mathbb{R}^{m_i}} \left\{ \langle y_i, x\rangle + g_i(x) + \frac{\gamma_i}{2}\|\bar{x}_i - x\|_i^2 \right\}, \quad \forall y_i \in \mathbb{R}^{m_i}$$

where $\boldsymbol{\gamma} = [\gamma_1, \gamma_2, \dots, \gamma_n]$, $\|.\|_i$ is the standard Euclidean distance on $\mathbb{R}^{m_i}$, and $\mathcal{P}(\bar{\boldsymbol{x}}, \boldsymbol{y}, \boldsymbol{\gamma}) = \sum_{i=1}^n \boldsymbol{D}_i^T \boldsymbol{D}_i \mathcal{P}_i(\bar{x}_i, y_i, \gamma_i)$. The RCSD method is shown in Algorithm 4.

Next, we will present the convergence result of RCSD for uniform sampling strategy. For more detailed description of the results, see [20].

Theorem 12 (Theorem 3, [20]). *Suppose the sequence $\boldsymbol{x}^k, k = 1, \dots, K$, is generated by Algorithm 4. Assume $\gamma_i = L_i$ for L_i being block wise Lipschitz constants for function f. Then,*

$$\min_{k\in\{0,\dots,K\}} \mathbb{E}\left\{ \|G(\boldsymbol{x}^k, \nabla f(\boldsymbol{x}^k) - \boldsymbol{v}^k, \boldsymbol{\gamma})\|^2 \right\} \le \frac{2L_{max} n(\phi(\boldsymbol{x}^1) - \phi^*)}{K},$$

where $G(\boldsymbol{x}^k, \nabla f(\boldsymbol{x}^k) - \boldsymbol{v}^k, \boldsymbol{\gamma}) = \sum_{i=1}^n \boldsymbol{D}_i^T \boldsymbol{D}_i G_i(x_i^k, \nabla_i f(\boldsymbol{x}^k) - v_i^k, \gamma_i)$, $L_{max} = \max_{i\in\{1,\dots,n\}} L_i$, and

$$G_i(x_i, \nabla_i f(\boldsymbol{x}) - v_i, \gamma_i) = \gamma_i(x_i - \mathcal{P}_i(x_i, \nabla_i f(\boldsymbol{x}) - v_i, \gamma_i)).$$

Now, we will show how BDCA relates to RCSD. This result is based on the smoothness of f.

Proposition 13. *BDCA covers RCSD with uniform sampling strategy as a special instance.*

Proof. Due to smoothness of f we can write the objective function as

$$\phi(\boldsymbol{x}) = \frac{L_{max}}{2}\|\boldsymbol{x}\|^2 + g(\boldsymbol{x}) - (h(\boldsymbol{x}) + \frac{L_{max}}{2}\|\boldsymbol{x}\|^2 - f(\boldsymbol{x})).$$

Algorithm 4 RCSD algorithm.

Input: $\boldsymbol{x}^{(0)}$, total iterations K, coefficients $\boldsymbol{\gamma}$, and number of blocks n
for $k = 1$ **to** K **do**
  Randomly choose i_k in $[1, \dots, n]$ with uniform distribution
  Compute $\nabla_{i_k} f(\boldsymbol{x}^k)$ and $\boldsymbol{v}^k \in \partial h(\boldsymbol{x}^k)$
  Update $x_{i_k}^{k+1} = \mathcal{P}_{i_k}(x_{i_k}^k, \nabla_{i_k} f(\boldsymbol{x}^k) - v_{i_k}^k, \gamma_{i_k})$
  Set $\boldsymbol{x}^{k+1} = \bar{\boldsymbol{x}}_{i_k}^k + \boldsymbol{x}_{i_k}^{k+1}$
end for
Output: $\boldsymbol{x}^{K+1}$

Take $\boldsymbol{v}_{i_k}^k \in \partial_{i_k} h(\boldsymbol{x}^k)$. This gives the BDCA subproblem as

$$
\begin{aligned}
x_{i_k}^{k+1} &\in \operatorname{argmin}_{x_{i_k} \in \mathcal{M}_{i_k}} \frac{L_{max}}{2}\|x_{i_k}\|^2 + g_{i_k}(x_{i_k}) - \langle \boldsymbol{v}_{i_k}^k - \nabla_{i_k} f(\boldsymbol{x}^k) + L_{\max} x_{i_k}^k, x_{i_k}\rangle \\
&= \operatorname{argmin}_{x_{i_k} \in \mathcal{M}_{i_k}} \langle \nabla_{i_k} f(\boldsymbol{x}^k) - \boldsymbol{v}_{i_k}^k, x_{i_k}\rangle + g_{i_k}(x_{i_k}) + \frac{L_{max}}{2}\|x_{i_k} - x_{i_k}^k\|^2 \\
&= \operatorname{argmin}_{x_{i_k} \in \mathcal{M}_{i_k}} \langle \nabla_{i_k} f(\boldsymbol{x}^k) - \boldsymbol{v}_{i_k}^k, x_{i_k} - x_{i_k}^k\rangle + g_{i_k}(x_{i_k}) + \frac{L_{max}}{2}\|x_{i_k} - x_{i_k}^k\|^2 \\
&= \operatorname{argmin}_{x_{i_k} \in \mathcal{M}_{i_k}} \frac{L_{max}}{2}\left[\frac{2}{L_{max}}\langle \nabla_{i_k} f(\boldsymbol{x}^k) - \boldsymbol{v}_{i_k}^k, x_{i_k} - x_{i_k}^k\rangle\right] + g_{i_k}(x_{i_k}) + \frac{L_{max}}{2}\|x_{i_k} - x_{i_k}^k\|^2 \\
&= \operatorname{argmin}_{x_{i_k} \in \mathcal{M}_{i_k}} g_{i_k}(x_{i_k}) + \frac{L_{max}}{2}\|x_{i_k} - (x_{i_k}^k - \frac{1}{L_{max}}(\nabla_{i_k} f(\boldsymbol{x}^k) - \boldsymbol{v}_{i_k}^k))\|^2 \\
&= \operatorname{prox}_{\frac{1}{L_{max}} g_{i_k} + I_{\mathcal{M}_{i_k}}}(x_{i_k}^k - \frac{1}{L_{max}}(\nabla_{i_k} f(\boldsymbol{x}^k) - \boldsymbol{v}_{i_k}^k))
\end{aligned}
$$

which is the block proximal operator with constant $1/L_{max}$ with I_C denoting the indicator function on the set C. Application of this update for each randomly selected block i_k in BDCA gives RCSD. □

Appendix C. Stationarity in nonsmooth optimization

In this section, we illustrate through simple examples that two common measures for first-order stationarity in smooth optimization, namely the distance measure

$$\operatorname{dist}(0, \nabla f(\boldsymbol{y}) + \partial g(\boldsymbol{y}) - \nabla h(\boldsymbol{y}) + \mathcal{N}_{\mathcal{M}}(\boldsymbol{y})),$$

and the Frank-Wolfe gap (but with a subgradient $\boldsymbol{u} \in \partial g(\boldsymbol{y})$)

$$\max_{\boldsymbol{x} \in \mathcal{M}} \langle \nabla f(\boldsymbol{y}) + \boldsymbol{u} - \nabla h(\boldsymbol{y}), \boldsymbol{y} - \boldsymbol{x}\rangle$$

are intractable in nonsmooth optimization. Specifically, we show that these measures can be discontinuous at the solution and remain lower bounded even when approaching the solution arbitrarily closely.

Consider the problem where $\mathcal{M} = [-1, 1]$, $f(x) = h(x) = 0$, and $g = |x|$. The problem is convex, and the only first-order stationary point is at $x = 0$, which is also the global minimum. The subdifferential of g is and the normal cone of $\mathcal{M}$ are given by

$$
\partial g(x) = \begin{cases} \{1\} & \text{if} x > 0, \\ \{-1\} & \text{if} x < 0, \\ [-1, 1] & \text{if} x = 0, \end{cases} \qquad \mathcal{N}_{\mathcal{M}}(x) = \begin{cases} \{0\} & \text{if} x \in (-1, 1), \\ (-\infty, 0] & \text{if} x = -1, \\ [0, +\infty) & \text{if} x = 1. \end{cases}
$$

Then, the distance measure becomes

$$
\operatorname{dist}(0, \partial g(y) + \mathcal{N}_{\mathcal{M}}(y)) = \min_{\substack{u \in \partial g(y) \\ z \in \mathcal{N}_{\mathcal{M}}(y)}} |u + z - 0| = \begin{cases} 0 & \text{if} y = 0, \\ 1 & \text{otherwise}, \end{cases}
$$

indicating that the distance is 0 only at the global optimum and remains 1 at all other points, even when arbitrarily close to the global optimum.

Similarly, the Frank-Wolfe gap, even when minimized over the subdifferential set, becomes

$$
\min_{u \in \partial g(y)} \max_{x \in [-1,1]} \langle u, y - x\rangle = \begin{cases} 0 & \text{if} y = 0, \\ y + 1 & \text{if} y > 0, \\ 1 - y & \text{if} y < 0, \end{cases} \tag{C.1}
$$

which shows that, except at $y = 0$, the Frank-Wolfe gap is larger than 1 at all other points, even when arbitrarily close to the global optimum.

Now, if we use the gap function as in Definition 1, we have:

$$\mathrm{gap}^0_{\mathcal{M}}(y) = \max_{x\in[-1,1]} \{|y| - |x|\} = |y|,$$

which is always non-negative and becomes arbitrarily small as we approach the solution at $y = 0$.

Notably, since the examples are convex, these findings also extend to convex nonsmooth optimization problems.

Appendix D. Proof of Remark 2

We start by showing that $\mathrm{gap}^L_{\mathcal{M}}(\boldsymbol{y}) = 0$ if

$$\boldsymbol{y} = \mathrm{prox}_{\frac{1}{L}g+I_{\mathcal{M}}}\Big((\boldsymbol{y} - \tfrac{1}{L}\big(\nabla f(\boldsymbol{y}) - \nabla h(\boldsymbol{y})\big)\Big).$$

Using the definition of the proximal map we have

$$\begin{aligned}\boldsymbol{y} &= argmin_{\boldsymbol{x}\in\mathcal{M}}\ \ g(\boldsymbol{x}) + \frac{L}{2}\|\boldsymbol{x} - (\boldsymbol{y} - \tfrac{1}{L}\big(\nabla f(\boldsymbol{y}) - \nabla h(\boldsymbol{y})\big))\|^2\\ &= \mathrm{argmin}_{\boldsymbol{x}\in\mathcal{M}}\ \ g(\boldsymbol{x}) + \frac{L}{2}\|\boldsymbol{x} - \boldsymbol{y}\|^2 + \langle \boldsymbol{x} - \boldsymbol{y}, \nabla f(\boldsymbol{y}) - \nabla h(\boldsymbol{y})\rangle\\ &= \mathrm{argmax}_{\boldsymbol{x}\in\mathcal{M}}\ \ -g(\boldsymbol{x}) - \frac{L}{2}\|\boldsymbol{x} - \boldsymbol{y}\|^2 + \langle \boldsymbol{y} - \boldsymbol{x}, \nabla f(\boldsymbol{y}) - \nabla h(\boldsymbol{y})\rangle.\end{aligned}$$

From this formulation, it follows that $\mathrm{gap}^L_{\mathcal{M}}(\boldsymbol{y}) = 0$.

Next, we show that $\boldsymbol{y}$ is the fixed point of the proximal mapping when $\mathrm{gap}^L_{\mathcal{M}} = 0$. We start by the reformulating the definition of the gap function:

$$\mathrm{gap}^L_{\mathcal{M}}(\boldsymbol{y}) = L\max_{\boldsymbol{x}\in\mathcal{M}}\left\{\langle \boldsymbol{x} + \frac{1}{L}\big(\nabla f(\boldsymbol{y}) - \nabla h(\boldsymbol{y})\big) - \boldsymbol{y}, \boldsymbol{y} - \boldsymbol{x}\rangle + \frac{1}{L}(g(\boldsymbol{y}) - g(\boldsymbol{x})) + \frac{1}{2}\|\boldsymbol{x} - \boldsymbol{y}\|^2\right\}.$$

Suppose $\mathrm{gap}^L_{\mathcal{M}} = \epsilon$. Then, by definition:

$$\langle \boldsymbol{x} + \frac{1}{L}\big(\nabla f(\boldsymbol{y}) - \nabla h(\boldsymbol{y})\big) - \boldsymbol{y}, \boldsymbol{y} - \boldsymbol{x}\rangle + \frac{1}{L}g(\boldsymbol{y}) - \frac{1}{L}g(\boldsymbol{x}) + \frac{1}{2}\|\boldsymbol{x} - \boldsymbol{y}\|^2 \le \frac{\epsilon}{L}, \tag{D.1}$$

for all $\boldsymbol{x} \in \mathcal{M}$. Consider this with $\boldsymbol{x} = \mathrm{prox}_{\frac{1}{L}g+I_{\mathbf{M}}}(\boldsymbol{u})$ and $\boldsymbol{u} = \boldsymbol{y} - \frac{1}{L}\big(\nabla f(\boldsymbol{y}) - \nabla h(\boldsymbol{y})\big)$. Using non-expansiveness of the proximal operator we have

$$\langle \boldsymbol{x} - \boldsymbol{u}, \boldsymbol{y} - \boldsymbol{x}\rangle + \frac{1}{L}g(\boldsymbol{y}) - \frac{1}{L}g(\boldsymbol{x}) \ge 0. \tag{D.2}$$

Utilizing (D.2) in (D.1), we obtain the desired bound:

$$\frac{1}{2}\|\boldsymbol{y} - \mathrm{prox}_{\frac{1}{L}g+I_{\mathbf{M}}}(\boldsymbol{u})\|^2 \le \frac{\epsilon}{L}.$$

In particular, if $\epsilon = 0$, this implies $\boldsymbol{y} = \mathrm{prox}_{\frac{1}{L}g+I_{\mathcal{M}}}(\boldsymbol{u})$.

Appendix E. Proof of Lemma 6

The proof follows the same reasoning as in Lemma 1, with the extension that h is allowed to be non-differentiable.

The first statement is immediate, since the maximized expression vanishes when $\boldsymbol{x} = \boldsymbol{y}$, and $\boldsymbol{y} \in \mathcal{M}$.

For the second statement, let us first assume that condition (9) holds. Then, there exist $\boldsymbol{v}^\star \in \partial h(\boldsymbol{y})$ and $\boldsymbol{u}^\star \in \partial g(\boldsymbol{y})$ such that $\boldsymbol{v}^* - \boldsymbol{u}^* - \nabla f(\boldsymbol{y}) \in \mathcal{N}_{\mathcal{M}}(\boldsymbol{y})$, or equivalently

$$\langle \nabla f(\boldsymbol{y}) + \boldsymbol{u}^* - \boldsymbol{v}^*, \boldsymbol{y} - \boldsymbol{x}\rangle \le 0, \quad \forall \boldsymbol{x} \in \mathcal{M}.$$

Since g is convex, this implies

$$\begin{aligned}&\langle \nabla f(\boldsymbol{y}) - \boldsymbol{v}^*, \boldsymbol{y} - \boldsymbol{x}\rangle + g(\boldsymbol{y}) - g(\boldsymbol{x}) \le 0 \quad \forall \boldsymbol{x} \in \mathcal{M},\\ \Longrightarrow\ &\langle \nabla f(\boldsymbol{y}) - \boldsymbol{v}^*, \boldsymbol{y} - \boldsymbol{x}\rangle + g(\boldsymbol{y}) - g(\boldsymbol{x}) - \frac{L}{2}\|\boldsymbol{x} - \boldsymbol{y}\|^2 \le 0 \quad \forall \boldsymbol{x} \in \mathcal{M}.\end{aligned}$$

Since this inequality holds for all $\boldsymbol{x} \in \mathcal{M}$, we can maximize the left-hand side over $\boldsymbol{x}$. Moreover, since it holds for at least one subgradient $\boldsymbol{v}^* \in \partial h(\boldsymbol{y})$, we can minimize over the subdifferential set:

$$\min_{\boldsymbol{u}\in\partial g(\boldsymbol{y})}\max_{\boldsymbol{x}\in\mathcal{M}}\left\{\langle \nabla f(\boldsymbol{y}) - \boldsymbol{v}, \boldsymbol{y} - \boldsymbol{x}\rangle + g(\boldsymbol{y}) - g(\boldsymbol{x}) - \frac{L}{2}\|\boldsymbol{x} - \boldsymbol{y}\|^2\right\} \le 0.$$

Finally, by Von Neumann's minimax theorem, we can interchange the order of min and max since both $\partial g(\boldsymbol{y})$ and $\mathcal{M}$ are convex and compact sets; and the expression inside the braces is concave in $\boldsymbol{x}$ for any fixed $\boldsymbol{u} \in \partial g(\boldsymbol{y})$, and convex in $\boldsymbol{v}$ for any fixed $\boldsymbol{x} \in \mathcal{M}$. After this change, we obtain $\mathrm{gap}^L_{\mathcal{M}}(\boldsymbol{y}) \le 0$. Since the gap function is nonnegative by definition, we conclude that $\mathrm{gap}^L_{\mathcal{M}}(\boldsymbol{y}) = 0$.

We now prove the reverse direction. Suppose that $\text{gap}^L_{\mathcal{M}}(\boldsymbol{y}) = 0$. Then, there exists a subgradient $\boldsymbol{v}^\star \in \partial h(\boldsymbol{y})$ such that

$$\langle \nabla f(\boldsymbol{y}) - \boldsymbol{v}^\star, \boldsymbol{x} - \boldsymbol{y}\rangle + g(\boldsymbol{x}) - g(\boldsymbol{y}) + \frac{L}{2}\|\boldsymbol{x} - \boldsymbol{y}\|^2 \geq 0, \quad \forall \boldsymbol{x} \in \mathcal{M}.$$

Consider $\boldsymbol{x} = \boldsymbol{y} + \alpha \boldsymbol{d}$ for an arbitrary feasible direction $\boldsymbol{d}$ (unit norm) and step-size $\alpha > 0$. Then,

$$\alpha\langle \nabla f(\boldsymbol{y}) - \boldsymbol{v}^\star, \boldsymbol{d}\rangle + g(\boldsymbol{y} + \alpha\boldsymbol{d}) - g(\boldsymbol{y}) + \frac{L}{2}\alpha^2 \geq 0, \quad \forall \alpha\boldsymbol{d} \,:\, \boldsymbol{y} + \alpha\boldsymbol{d} \in \mathcal{M}.$$

Dividing by α and taking the limit as $\alpha \to 0^+$, we obtain

$$\langle \nabla f(\boldsymbol{y}) - \boldsymbol{v}^\star, \boldsymbol{d}\rangle + \langle \boldsymbol{u}^\star, \boldsymbol{d}\rangle \geq 0, \quad \forall \boldsymbol{d} \,:\, \lim_{\alpha\to 0^+} \boldsymbol{y} + \alpha\boldsymbol{d} \in \mathcal{M}.$$

for some $\boldsymbol{u}^\star \in \partial g(\boldsymbol{y})$. Since $\mathcal{M}$ is closed and convex, $\lim_{\alpha\to 0^+} \boldsymbol{y} + \alpha\boldsymbol{d} \in \mathcal{M}$ for all $\boldsymbol{d} = \boldsymbol{x} - \boldsymbol{y}$ such that $\boldsymbol{x} \in \mathcal{M}$. Consequently,

$$\langle \nabla f(\boldsymbol{y}) + \boldsymbol{u}^\star - \boldsymbol{v}^\star, \boldsymbol{x} - \boldsymbol{y}\rangle \geq 0, \quad \forall \boldsymbol{x} \in \mathcal{M},$$

which is equivalent to $-\nabla f(\boldsymbol{y}) - \boldsymbol{u}^\star + \boldsymbol{v}^\star \in \mathcal{N}_{\mathcal{M}}(\boldsymbol{y})$. Therefore, we have condition (9) satisfied.

Appendix F. Application to one-bit MIMO signal recovery

In this section, we will use the proposed Block EM method in one-bit MIMO signal recovery problem. First, we describe the problem. Later, simulation results are provided to demonstrate the performance of our method compared to the conventional EM method.

Appendix F.1. Problem description

Consider a MIMO uplink scenario where N transmitters (users) simultaneously send a signal of length W to the Base-Station (BS) which consists of M receiving antennas. The signal will undergo a frequency selective fading channel. Then, the received signal at the BS will have the form

$$\boldsymbol{R}_m = \sum_{n=1}^{N} \boldsymbol{H}_{m,n}\boldsymbol{\theta}_n + \boldsymbol{\alpha}_m \quad \text{and} \quad \boldsymbol{Y_m} = Q(\boldsymbol{R}_m), \quad \text{for} \quad m = 1, \dots, M, \tag{F.1}$$

where $\boldsymbol{R}_m \in \mathbb{C}^W$ is the received signal at mth antenna, $\boldsymbol{H}_{m,n} \in \mathbb{C}^{W\times W}$ denotes the channel circulant matrix between the $n$the user and mth receiver, $\boldsymbol{\theta}_n \in \mathbb{C}^W$ is the transmitted signal of the nth user, and $\alpha_m \in \mathbb{C}^W$ is the complex additive white Gaussian noise for the mth receiver with mean $\boldsymbol{0}$ and covariance matrix $\sigma_\alpha^2\boldsymbol{I}$. Note that each $H_{m,n}$ is constructed from the channel impulse response $h_{m,n} = \{h^0_{m,n}, h^1_{m,n}, \dots, h^L_{m,n}, 0 \dots, 0\} \in \mathbb{C}^W$ between the nth user and mth receiver with L being the number of channel taps. The operator $Q(.)$ serves as the sign operator on the real and imaginary parts of its argument. Throughout this section we assume the transmitted signal to be modulated with 16 QAM and $W \gg L$. This means that $\boldsymbol{\theta} \in \{-3, -1, +1, +3\}$. It is possible to recover such signal with EM method as [35]. Here, the aim is to use the Block EM method to recover the modulated signal $\boldsymbol{\theta}$. To do so, we represent (F.1) as

$$\boldsymbol{R} = \mathcal{H}_{cir}\boldsymbol{\theta} + \boldsymbol{\alpha} \qquad \text{and} \qquad \boldsymbol{Y} = Q(\boldsymbol{R})$$

where $\boldsymbol{R} \in \mathbb{C}^{MW\times 1}, \boldsymbol{\alpha} \in \mathbb{C}^{MW\times 1}, \boldsymbol{Y} \in \mathbb{C}^{MW\times 1}, \boldsymbol{\theta} \in \mathbb{C}^{NW\times 1}$ are the vertically stacked $\boldsymbol{R}_m, \boldsymbol{\alpha}_m, \boldsymbol{Y}_m, \boldsymbol{\theta}_n$ over m and n respectively. $\mathcal{H}_{cir} \in \mathbb{C}^{MW\times NW}$ is the block-circulant Toeplitz matrix made from $\boldsymbol{H}_{m,n}$'s. According to (F.1) we need to form the conditional probability distribution function (PDF) $p(\boldsymbol{R}|\boldsymbol{Y}, \boldsymbol{\theta})$ (note that we are treating $\boldsymbol{y}$ as the observed variable and $\boldsymbol{R}$ as the latent variable). Using the Gaussian assumption, the joint PDF $p(\boldsymbol{R}, \boldsymbol{Y}|\boldsymbol{\theta})$ becomes

$$p(\boldsymbol{R}, \boldsymbol{Y}|\boldsymbol{\theta}) = \frac{\mathbb{I}_{Q(\boldsymbol{R})}(\boldsymbol{Y})}{\pi\sigma_\alpha^2} e^{-\frac{(\boldsymbol{R} - \mathcal{H}_{cir}\boldsymbol{\theta})^2}{\sigma_\alpha^2}}$$

with $\mathbb{I}_{\mathbb{Q}(\boldsymbol{R})}(\boldsymbol{Y})$ being one when $Q(\boldsymbol{R}) = \boldsymbol{Y}$ and $(.)^2$ denoting element-wise square operator. Therefore $\hat{P}(\boldsymbol{R}) := \hat{P}(\boldsymbol{R}|\boldsymbol{Y}, \boldsymbol{\theta})$ in Block EM becomes

$$\hat{P}(\boldsymbol{R}|\boldsymbol{Y}, \boldsymbol{\theta}) = \frac{e^{-\frac{(\boldsymbol{R} - \mathcal{H}_{cir}\boldsymbol{\theta})^2}{\sigma_\alpha^2}}}{\int_{\mathcal{D}} e^{-\frac{(\boldsymbol{R} - \mathcal{H}_{cir}\boldsymbol{\theta})^2}{\sigma_\alpha^2}} d\boldsymbol{R}} \tag{F.2}$$

where $\mathcal{D}_i = (-\infty, 0)$ when $\boldsymbol{Y}_i < 0$ and $\mathcal{D}_i = (0, +\infty)$ when $\boldsymbol{Y}_i > 0$. The second update of Block EM has the objective $\mathbb{E}_{\hat{P}(\boldsymbol{R}|\boldsymbol{Y},\boldsymbol{\theta})}\left[-\log(P(\boldsymbol{R}, \boldsymbol{Y}|\theta_{i_k}))\right]$. From the structure of $-\log(P(\boldsymbol{R}, \boldsymbol{Y}|\theta_{i_k}))$ we know

$$-\log(P(\boldsymbol{R}, \boldsymbol{Y}|\theta_{i_k})) = \frac{\boldsymbol{R}^T\boldsymbol{R} - \boldsymbol{R}^T\mathcal{H}_{cir}\boldsymbol{\theta} - \boldsymbol{\theta}^T\mathcal{H}_{cir}^T\boldsymbol{R} + \boldsymbol{\theta}^T\mathcal{H}_{cir}^T\mathcal{H}_{cir}\boldsymbol{\theta}}{\sigma_\alpha^2}.$$

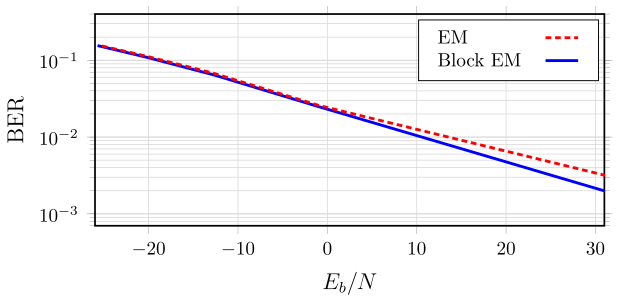

Fig. F.3. Comparison of Block EM and classical EM in terms of BER vs. E_b/N.

Note that for simplicity, we considered θ_{i_k} in θ as the optimization variable. Therefore, the first term is independent from the optimization variable and we replace it with $\hat{\boldsymbol{R}}^T\hat{\boldsymbol{R}} = \mathbb{E}_{\hat{P}(\boldsymbol{R}|\boldsymbol{Y},\theta)}[\boldsymbol{R}]^T\mathbb{E}_{\hat{P}(\boldsymbol{R}|\boldsymbol{Y},\theta)}[\boldsymbol{R}]$. Thus,

$$\mathbb{E}_{\hat{P}(\boldsymbol{R}|\boldsymbol{Y},\theta)}\left[-\log(P(\boldsymbol{R},\boldsymbol{Y}|\theta_{i_k}))\right] = \frac{\hat{\boldsymbol{R}}^T\hat{\boldsymbol{R}} - \hat{\boldsymbol{R}}^T\mathcal{H}_{cir}\theta - \theta^T\mathcal{H}_{cir}^T\hat{\boldsymbol{R}} + \theta^T\mathcal{H}_{cir}^T\mathcal{H}_{cir}\theta}{\sigma_\alpha^2} = \frac{\|\hat{\boldsymbol{R}} - \mathcal{H}_{cir}\theta\|^2}{\sigma_\alpha^2} \tag{F.3}$$

Therefore, in this problem we can evaluate the expected value of the latent variables $\boldsymbol{R}$ through (F.2) and then minimize the convex objective F.3 to recover the transmitted symbols θ. The coordinate-wise calculation of the expected value of the latent variables $\boldsymbol{R}$ will take the form

$$\hat{\boldsymbol{R}}^{k+1} := \mathbb{E}_{\hat{P}(\boldsymbol{R}|\boldsymbol{Y},\theta^k)}[\boldsymbol{R}] = \sigma_\alpha^2\left(\frac{\boldsymbol{Y}_{Re}\phi(\boldsymbol{Y}_{Re}\boldsymbol{Z}_{Re}^k/\sigma_\alpha)}{\Phi(\boldsymbol{Y}_{Re}\boldsymbol{Z}_{Re}^k/\sigma_\alpha)} + j\frac{\boldsymbol{Y}_{Im}\phi(\boldsymbol{Y}_{Im}\boldsymbol{Z}_{Im}^k/\sigma_\alpha)}{\Phi(\boldsymbol{Y}_{Im}\boldsymbol{Z}_{Im}^k/\sigma_\alpha)}\right) + \mathcal{H}_{cir}(:,i_k)\theta_{i_k}^k \tag{F.4}$$

where subscripts Re, Im specify the real and the imaginary parts, $j = \sqrt{-1}$, $\boldsymbol{Z}^k = \mathcal{H}_{cir}\theta^k$, $\phi(x) = \frac{1}{\sqrt{2\pi}}exp\{-\frac{x^2}{2}\}$, and $\Phi(x) = \int_\infty^x \phi(x')dx'$. With (F.4) at hand, we can easily update $\theta_{i_k}^{k+1}$ as

$$\theta_{i_k}^{k+1} = argmin_{\substack{\boldsymbol{x}\in\mathbb{C}^{\|i_k\|_0} \\ \boldsymbol{x}\in\{-3,-1,+1,+3\}}} \|\hat{\boldsymbol{R}}^{k+1} - \mathcal{H}_{cir}(:,i_k)\boldsymbol{x}\|_2^2. \tag{F.5}$$

Therefore, iterative updates of (F.4) and (F.5) will recover $\theta_{i_k}^{k+1}$. Note that to project to $\{-3,-1,+1,+3\}$ we need to relax the constraint set to $x \in [-3,3]$, and ensure the convergence of the updates before projection. This will lead to a small inner loop for each i_k.

Appendix F.2. Numerical experiments

We conducted numerical simulations to verify the performance of the Block EM in this problem. Specifically, we aim to study the Bit Error Rate (BER) for various bit energy to noise spectral density (E_b/N) defined d as

$$\frac{E_b}{N} = \frac{P\mathrm{Tr}\left(\mathbb{E}_{\mathcal{H}_{cir}}\left[\mathcal{H}_{cir}\mathcal{H}_{cir}^H\right]\right)}{MN\sigma_\alpha^2 B},$$

where P denotes the total transmitted power and B is number of bits in the signal constellation symbol. Consider a MIMO system employing 16-QAM with cyclic prefix omitted at the transmitter, $N = 6$ users, and $M = 64$ receiving antennas. The length of signals is $W = 32$. Channel was considered with complex Gaussian noise with mean zero and covariance matrix of $\sigma^2\boldsymbol{I}$. The length of the fading is $L = 5$ and signals undergo 4 paths to reach the receiver. The number of inner loops for projection was 10 and the stopping criteria was $\|\xi^u - \xi^{u-1}\|_2 \le \lambda\|\xi^{u-1}\|_2$ with $\lambda = 10^{-3}$, $\xi^u = \frac{1}{25}\sum_{p=25(u-1)+1}^{25u}\theta^p$. This choice of ξ^u is quite flexible and any short-term criteria to understand the behaviour of the update sequence can be applied. The maximum number of iterations for the Block EM algorithm was 10^4 and 10^3 for the EM algorithm. The algorithms were initialized at zero and the block size was 10. The results are depicted in Fig. F.3 for 50 Monte-Carlo simulations.

As shown in Fig. F.3, the Block EM performs slightly better than EM in this setting. Note that Block EM requires less computational resources than EM to maximize the likelihood function.

References

[1] P.D. Tao, E.B. Souad, Algorithms for solving a class of nonconvex optimization problems. methods of subgradients, In J.-B. Hiriart-Urruty (Ed.), Fermat Days 85: Mathematics for Optimization of North-Holland Mathematics Studies, vol. 129, North-Holland, 1986, pp. 249–271.
[2] K. Lange, MM optimization algorithms, SIAM, 2016.
[3] A. Argyriou, R. Hauser, C.A. Micchelli, M. Pontil, A DC-programming algorithm for kernel selection, in: Proceedings of the 23rd International Conference on Machine Learning, 2006, pp. 41–48.
[4] B.K. Sriperumbudur, D.A. Torres, G.R.G. Lanckriet, Sparse eigen methods by DC programming, in: Proceedings of the 24th International Conference on Machine Learning, 2007, pp. 831–838.
[5] P. Awasthi, C. Cortes, M. Mohri, Best-effort adaptation, Ann. Math. Artif. Intell. (2024) 1–46.

[6] P. Awasthi, A. Mao, M. Mohri, Y. Zhong, DC-programming for neural network optimizations, J. Glob. Optim. (2024) 1–17.
[7] A.L. Yuille, A. Rangarajan, The concave-convex procedure, Neural Comput. 15 (4) (2003) 915–936.
[8] H.A. Le Thi, T. Pham Dinh, DC programming and DCA: thirty years of developments, Math. Program. 169 (Jan 2018).
[9] T. Pham Dinh, V.N. Huynh, H.A. Le Thi, V.T. Ho, Alternating DC algorithm for partial DC programming problems, J. Glob. Optim. 82 (4) (2022) 897–928.
[10] G. Yuan, Coordinate descent methods for dc minimization: Optimality conditions and global convergence, in: Proceedings of the AAAI Conference on Artificial Intelligence, vol. 37, 2023, pp. 11034–11042.
[11] A. Yurtsever, S. Sra, CCCP is Frank-Wolfe in disguise, Adv. Neural Inf. Process. Syst. 35 (2022) 35352–35364.
[12] R. Kumar, M. Schmidt, Convergence rate of Expectation-Maximization, in: 10th NIPS Workshop on Optimization for Machine Learning, 2017, p. 98.
[13] F. Kunstner, R. Kumar, M.W. Schmidt, Homeomorphic-invariance of EM: Non-asymptotic convergence in KL divergence for exponential families via mirror descent, in: International Conference on Artificial Intelligence and Statistics, 2020.
[14] M. Razaviyayn, M. Hong, Z.-Q. Luo, A unified convergence analysis of block successive minimization methods for nonsmooth optimization, SIAM J. Optim. 23 (2) (2013) 1126–1153.
[15] Y. Xu, W. Yin, A globally convergent algorithm for nonconvex optimization based on block coordinate update, J. Sci. Comput. 72 (2) (2017) 700–734.
[16] A. Aubry, A. De Maio, A. Zappone, M. Razaviyayn, Z.-Q. Luo, A new sequential optimization procedure and its applications to resource allocation for wireless systems, IEEE Trans. Signal Process. 66 (24) (2018) 6518–6533.
[17] F. Chorobura, I. Necoara, Random coordinate descent methods for nonseparable composite optimization, SIAM J. Optim. 33 (3) (2023) 2160–2190.
[18] X. Cai, C. Song, S. Wright, J. Diakonikolas, Cyclic block coordinate descent with variance reduction for composite nonconvex optimization, in: International Conference on Machine Learning, PMLR, 2023, pp. 3469–3494.
[19] C.D. Dang, G. Lan, Stochastic block mirror descent methods for nonsmooth and stochastic optimization, SIAM J. Optim. 25 (2) (2015) 856–881.
[20] Q. Deng, C. Lan, Efficiency of coordinate descent methods for structured nonconvex optimization, in: Joint European Conference on Machine Learning and Knowledge Discovery in Databases, Springer, 2020, pp. 74–89.
[21] T. Gao, S. Lu, J. Liu, C. Chu, Randomized Bregman coordinate descent methods for non-Lipschitz optimization, arXiv preprint arXiv:2001.05202, 2020.
[22] O. Faust, H. Fawzi, J. Saunderson, A Bregman divergence view on the difference-of-convex algorithm, in: International Conference on Artificial Intelligence and Statistics, PMLR, 2023, pp. 3427–3439.
[23] P. Latafat, A. Themelis, P. Patrinos, Block-coordinate and incremental aggregated proximal gradient methods for nonsmooth nonconvex problems, Math. Program. 193 (1) (2022) 195–224.
[24] I. Fatkhullin, N. He, Taming nonconvex stochastic mirror descent with general Bregman divergence, in: International Conference on Artificial Intelligence and Statistics, PMLR, 2024, pp. 3493–3501.
[25] S.J. Wright, Coordinate descent algorithms, Math. Program. 151 (1) (2015) 3–34.
[26] J. Zhang, H. Lin, S. Jegelka, S. Sra, A. Jadbabaie, Complexity of finding stationary points of nonconvex nonsmooth functions, in: International Conference on Machine Learning, PMLR, 2020, pp. 11173–11182.
[27] G. Kornowski, O. Shamir, Oracle complexity in nonsmooth nonconvex optimization, Adv. Neural Inf. Process. Syst. 34 (2021) 324–334.
[28] H. Karimi, J. Nutini, M.W. Schmidt, Linear convergence of gradient and proximal-gradient methods under the Polyak-Łojasiewicz condition, in: ECML/PKDD, 2016.
[29] A.P. Dempster, N.M. Laird, D.B. Rubin, Maximum likelihood from incomplete data via the EM algorithm, J. R. Stat. Soc. Ser. B (Methodol.) 39 (1) (1977) 1–22.
[30] G.J. McLachlan, T. Krishnan, The EM algorithm and extensions, John Wiley & Sons, 2007.
[31] C.-C. Chang, C.-J. Lin, LIBSVM: a library for support vector machines, ACM Trans. Intell. Syst. Technol. 2 (3) (May 2011).
[32] Y. Ye, The Gset dataset, 2003. https://web.stanford.edu/.
[33] H. Li, K. Zhang, T. Jiang, The regularized EM algorithm, in: AAAI, 2005, pp. 807–812.
[34] X. Yi, C. Caramanis, Regularized EM algorithms: a unified framework and statistical guarantees, Adv. Neural Inf. Process. Syst. 28 (2015).
[35] D. Plabst, J. Munir, A. Mezghani, J.A. Nossek, Efficient non-linear equalization for 1-bit quantized cyclic prefix-free massive MIMO systems, in: 2018 15th International Symposium on Wireless Communication Systems (ISWCS), IEEE, 2018, pp. 1–6.