Online Stochastic DCA With Applications to Principal Component Analysis

Hoai An Le Thi, Hoang Phuc Hau Luu, and Tao Pham Dinh

Abstract—Stochastic algorithms are well-known for their performance in the era of big data. In this article, we study nonsmooth stochastic Difference-of-Convex functions (DC) programs—the major class of nonconvex stochastic optimization, which have a variety of applications in diverse domains, in particular, machine learning. We propose new online stochastic algorithms based on the state-of-the-art DC Algorithm (DCA)—a powerful approach in nonconvex programming framework, in the online context of streaming data continuously generated by some (unknown) source distribution. The new schemes use the stochastic approximations (SAs) principle; deterministic quantities of the standard DCA are replaced by their noisy estimators constructed using newly arriving samples. The convergence analysis of the proposed algorithms is studied intensively with the help of tools from modern convex analysis and martingale theory. Finally, we study several aspects of the proposed algorithms on an important problem in machine learning: the expected problem in principal component analysis (PCA).

Index Terms—Difference of Convex functions (DC) programming, DC algorithm (DCA), nonconvex optimization, online stochastic DCA (osDCA), principal component analysis (PCA).

NOMENCLATURE

\((\Omega, \mathcal{M}, \mathbb{P}), \mathbb{E}\)

Probability space and expectation.

\(Z, P_Z, \Xi\)

Random variable, its distribution, and its support.

\(R_l\)

Rademacher average.

\(\text{dom } \theta, \partial \theta\)

Effective domain and subdifferential of \(\theta\).

\(P_k, (P_k)_{k \in \mathbb{N}}\)

Information up to time \(k\) and filtration.

\(\rho(\theta), \theta^*\)

Modulus of strong convexity and conjugate of \(\theta\).

I. INTRODUCTION

In the era of big data, stochastic optimization is a means to resolve many associated challenges including data processing, storage bottleneck, noisy measurements, high dimensionality, and so on. A typical stochastic optimization problem takes the form

\[
\min \left\{ F(w) := \mathbb{E}(f(w, Z)) : w \in S \subset \mathbb{R}^n \right\} \tag{1}
\]

where \(Z\) is a random vector whose probability distribution is, in general, unknown, and \(f\) is a loss function defined on \(\mathbb{R}^m\). There is a vast literature that studies this problem in the convex setting, that is, when \(f(\cdot, Z)\) is convex for all \(Z\), and \(S\) is a convex set. When either \(f(\cdot, Z)\) or \(S\) is nonconvex [resp. \(f(\cdot, Z)\) is nonsmooth], (1) is a nonconvex (resp. nonsmooth) stochastic optimization problem. Just as with deterministic optimization, there is a source of difficulty when shifting from convex to nonconvex setting. Nonsmooth and nonconvex stochastic optimization, much more difficult than convex and/or smooth optimization, plays a central role in many practical applications. However, algorithms for nonsmooth and nonconvex stochastic problems remain rare.

In this article, we are concerned with the so-called stochastic Difference of Convex functions (DC) program, the major class of nonconvex stochastic optimization, which has the form of (1) with \(f\) being a DC function, namely

\[
\min \left\{ F(w) = \mathbb{E}(g(w, Z) - h(w, Z)) : w \in S \right\} \tag{2}
\]

where \(Z\) is a random vector determined in some complete probability space \((\Omega, \mathcal{M}, \mathbb{P})\) \((Z : \Omega \rightarrow \mathbb{R}^n)\), while \(S \subset \mathbb{R}^m\) is a nonempty, compact, and convex set, \(g(\cdot, Z)\) and \(h(\cdot, Z)\) are lower semicontinuous convex functions for all \(Z\), satisfying some technical conditions described later. The framework of the problem (2) is very general in two aspects. First, the underlying distribution of \(Z\) is arbitrary, which makes it able to treat any involved random variable. As a special case, when \(Z\) is uniformly distributed over a finite set, we obtain a large-sum problem

\[
\min_{w \in S} \left\{ F(w) = \frac{1}{N} \sum_{i=1}^{N} g(w, z_i) - \frac{1}{N} \sum_{i=1}^{N} h(w, z_i) \right\} \tag{3}
\]

Second, in (2), \(g\) and \(h\) are possibly nonsmooth, resulting in a very large class of stochastic nonsmooth, nonconvex DC programs that comprises most real-world problems [1].

Our study is mainly motivated by the role of stochastic DC programs in machine learning and by new challenges that emerged from optimization problems arising in machine learning.

First, let us mention that machine learning is a mine of DC programs, and various machine learning problems can be formulated in the form of the stochastic DC program (2).
Indeed, formally, a machine learning task results in an optimization problem that minimizes a loss function \( f(w, Z) \) on a training set of examples, quantifying the goodness of the prediction model parameterized by \( w \) using the data point \( Z \). In many cases, a regularizer is joined with the loss to encourage some structure of solutions such as sparsity or low rank. When the loss function and the regularizer are DC, this learning problem is a DC program. A typical example of DC regularization arises in learning with sparsity where the zero-norm (the zero-norm of a vector is the number of its nonzero components) is used for modeling sparsity. The major approach to deal with the zero-norm is nonconvex approximation where the zero-norm is replaced by its nonconvex approximation functions, and it is proved in [2] that all existing nonconvex approximations (e.g., capped \( \ell_1 \), log-sum penalty, minimax concave penalty, etc.) are DC. Examples of DC loss functions are numerous. For instance, classification or regression in supervised learning such as robust logistic regression [3], truncated linear regression [4], robust support vector machines [5], [6], robust support vector regression [7], phase retrieval [8], the nonconvex subproblem of Uzawa Algorithm for Neyman–Pearson classification [9], or again neural network with rectified linear activation unit (ReLU) activations [10] (combined with popular used loss such as cross-entropy, Huber loss, square error, LogCosh loss), and so on. Several other unsupervised/semisupervised learning problems possessing DC structures include principal component analysis (PCA) [11], t-Distributed Stochastic Neighbor Embedding [12], Positive Unlabeled learning [13], Semisupervised support vector machines [14], and so on.

If we consider the above learning problem in the context of stochastic data, say, \( Z \) is a random vector with an (unknown) probability distribution, then we have to minimize the expectation loss \( F(w) = \mathbb{E}(f(w, Z)) \) (and possibly with a regularizer), this optimization problem has the form of (2). Especially, the success of deep learning brought to the forefront optimization problems which are typically nonconvex (most of them are DC programs).

Apart from learning issues, (2) is a common model of many other applications in divers domains, to name a few—finance, energy, logistics, robotics.

Second, the challenges of big data (the huge available data volume, the velocity, and the uncertainty of data) require the design of fast and scalable stochastic optimization algorithms capable of learning online.

Our work addresses these challenges of big data and nonconvex optimization. Our goal is to provide fast and scalable nonconvex stochastic optimization algorithms that are well-suited to huge volume data in high-dimensional, stream data (i.e., in online contexts), and all this with arbitrary distribution random data. The backbone of our approach is DC programming and DC algorithm (DCA), a famous approach in nonconvex programming frameworks. This choice is justified by the fact that DCA is a powerful arsenal to tackle nonconvexity and nonsmoothness, as well as scalability. In fact, in a deterministic optimization context, DC programming, and DCA constitute a quite logical and natural extension of modern convex analysis/programming to nonsmooth nonconvex analysis/programming, sufficiently large to cover most real-world nonsmooth nonconvex programs, but not too broad to explore/exploit the powerful arsenal of convex analysis/programming. This theoretical and algorithmic philosophy was first introduced in 1985 by Pham Dinh Tao and widely developed by Le Thi Hoai An and Pham Dinh Tao since 1993 to become now classic and increasingly popular (see [15], [16], [17], [18] and a comprehensive review on 30 years of developments of DC programming and DCA in [1]). It is widely recognized that DCA is one of the rare algorithms to efficiently solve large-scale nonconvex and nonsmooth programs [1]. Thanks to the pervasiveness of DC programming and the flexible principle of DC reformulations, DCA recovers almost all standard methods in convex and nonconvex programming. Also, the flexibility and simplicity of DCA make the method a powerful tool to be employed in various applications in applied sciences (see [1] and references therein).

**Paper Contributions:** We investigate DCA to tackle the main challenge of the problem (2) that is the nonconvexity, and DCA is introduced in an online mode to deal with streaming data. Our contributions are multiple.

First, we design a generic online stochastic DCA (osDCA) scheme for solving the problem (2) in the most general setting: the function \( F(w) = G(w) - H(w) \) with \( G(w) := \mathbb{E}(g(w, Z)) \) and \( H(w) := \mathbb{E}(h(w, Z)) \), that is, both \( G \) and \( H \) are expectation functions (the problem setting will be described in more detail in Section IV). We combine DCA with stochastic approximations (SAs) in the online fashion: at each iteration, new data is used to construct an SA of \( G \) and one of a subgradient of \( H \), and the solution is updated via DCA. Since the upd at steps of the proposed algorithms require new fresh samples from the distribution of \( Z \), we refer to our algorithm as osDCA. The proposed osDCA enjoys a double benefit of an online algorithm: it is suitable to perform streaming data that comes from an unknown distribution.

Second, we propose two other osDCA schemes for the cases where we can compute directly (i.e., the SA procedure is not used) the function \( G \) (the second osDCA) or \( H \) (the third osDCA). In terms of quality, these two algorithms are more advantageous than the first osDCA, as the exact expectation usually implies better solutions (for instance, less noisy) than its approximation. However, while the first osDCA can be employed in any case, the use of the latter two is not always possible and usually requires additional knowledge about \( Z \) (see 1) and 2) of Remark 3 below).

Third, we rigorously study the convergence analysis of the three proposed algorithms. Basically, with the appropriate rising rate of samples across iterations, the proposed algorithms guarantee—with probability one—to subsequentially converge to critical points.

Fourth, we study the practical performance of the proposed algorithms via the expected problem of PCA (E-PCA). Numerical experiments have been conducted carefully on various real-world and synthetic datasets to study the proposed algorithms’ behaviors in different aspects.
II. RELATED WORKS

Stochastic optimization has been investigated thoroughly for convex problems since the seminal work [19], where the well-known stochastic gradient descent (SGD) was introduced, which really opened a door in numerical optimization for large-scale problems [20], [21]. In nonconvex settings, stochastic algorithms remain rare, and most of them require the objective to be smooth or partially smooth Overall, there are four approaches to tackling nonconvex stochastic problems. The first, inspired by the aforementioned SGD, includes stochastic (proximal) (sub)gradient-based methods which are mainly developed for smooth or weakly convex objective functions [8], [22], [23]. The second is stochastic majorization-minimization (MM) for partially smooth objective [24], [25], which iteratively minimizes a stochastic convex surrogate to obtain an updated optimization variable. Here, the authors considered either a smooth error function or a smooth surrogate, which implies—in the case of DC programs—one DC component should be smooth. The third is stochastic successive convex approximation [26], [27] (mainly for smooth objective functions) that is similar to stochastic MM where the sequence of approximation functions are convex but not necessary upper bounds of sample objective functions. The fourth is stochastic DCA (SDCA) for stochastic DC programs—a substantially large class to cover almost all real-world nonconvex optimization problems [1]. Initial works in this approach include [13], [28], [29], [30], and [31] that consider some special classes of DC programs such as large-sum and/or (partially) smooth, as well as [32] working on a very general class of stochastic nonsmooth DC programs. To extend beyond the DC programming framework, Metel and Takeda [33] used Moreau envelope which is a DC function to approximate a nonsmooth, nonconvex regularizer, and then developed an SDCA for solving the resulting problem. It should be noted that, as indicated in [1], while the (stochastic) MM proposes a general idea to majorize the objective function, (stochastic) DCA gives the simplest and the closest convex surrogate, thanks to DC structures of the objective. Furthermore, usual choices of surrogates of MM result in DCA versions [1].

So far, there are very few algorithms for the general setting (2). To our knowledge, this article [32] is the first work dealing with (2) where both DC components are nonsmooth. In that article, several SDCA schemes are proposed in the aggregated update style. That is, all past information (sample realizations) is used to construct subproblems. These algorithms, therefore, need to store all samples in the computer memory during the computational process. In the present work, we investigate osDCA for the general problem (2) to deal with fast streaming data where we do not need to store samples all the time. Furthermore, thanks to the online mechanism, our proposed algorithms have the adaptive ability which is a great advantage over the SDCA schemes proposed in [32]. Numerical experiments justify this claim.

III. PRELIMINARIES

A. Outline of DC Programming and DCA

In this section, we briefly introduce DC programming and DCA. Let $\Gamma_0(\mathbb{R}^n)$ denote the convex cone of all lower semicontinuous proper convex functions on $\mathbb{R}^n$. The standard DC program takes the form

$$a := \inf \{ f(x) = g(x) - h(x) : x \in \mathbb{R}^n \}, \quad (P_{DC})$$

where $g, h \in \Gamma_0(\mathbb{R}^n)$. Such a function $f$ is called DC, $g - h$ is DC decomposition, while $g$ and $h$ are DC components of $f$. Note that, a DC program with closed convex constraint $x \in C$ can be equivalently written as a standard DC program in such a way that $f = (g + \chi_C) - h$, where $\chi_C$ is the indicator function of $C$. For a convex function $\theta$ defined on $\mathbb{R}^n$ and a convex set $C$, the modulus of strong convexity of $\theta$ on $C$, denoted by $\rho(\theta, C)$ or $\rho(\theta)$ if $C = \mathbb{R}^n$, is given by

$$\rho(\theta, C) = \sup\{ \mu \geq 0 : \theta - (\mu/2)\|\cdot\|^2 \text{ is convex on } C \}.$$  

Moreover, a function $\theta$ is said to be strongly convex on $C$ if $\rho(\theta, C) > 0$. The subdifferential of $\theta$ at $x_0 \in \text{dom } \theta$, denoted by $\partial \theta(x_0)$, is defined by

$$\partial \theta(x_0) = \{ y \in \mathbb{R}^n : \theta(x) \geq \theta(x_0) + \langle x - x_0, y \rangle, \forall x \in \mathbb{R}^n \}.$$  

The conjugate function $\theta^*$ of $\theta \in \Gamma_0(\mathbb{R}^n)$ is defined by

$$\theta^*(y) = \sup\{ \langle x, y \rangle - \theta(x) : x \in \mathbb{R}^n \}.$$  

A point $x^*$ is called a critical point, or a generalized Karush–Kuhn–Tucker (KKT) point of $(P_{DC})$ if $\partial g(x^*) \cap \partial h(x^*) \neq \emptyset$, or equivalently $0 \in \partial g(x^*) - \partial h(x^*)$, while it is called a strongly critical point of $g - h$ if $0 \notin \partial h(x^*) \subset \partial g(x^*)$.

DCA is based on local optimality conditions and duality in DC programming, which introduces the nice and elegant concept of approximating a DC program by a sequence of convex ones: at each iteration $k$, DCA approximates the second DC component $h$ by its affine minorization $h_k(x) = h(x^k) + (x - x^k, y^k)$ by taking $y^k \in \partial h(x^k)$ and then solves the resulting convex subprogram to get $x^{k+1}$. The standard DCA is formally described as follows.

**Standard DCA**

**Initialization:** Let $x^0 \in \text{dom } \partial h$ and $k = 0$.

repeat

1. Compute the subgradient $y^k \in \partial h(x^k)$.

2. Solve the following convex program:

$$x^{k+1} \in \arg \min \{ g(x) - h_k(x) : x \in \mathbb{R}^n \}.$$  

3. Set $k = k + 1$.

until Stopping criterion.

Convergence properties of the standard DCA and its complete theoretical foundation in the DC programming framework can be found in [15], [16], and [17]. For instance, it is especially worth mentioning that the sequence $\{x^k\}$ generated by DCA has the following properties.

1) The sequence $\{(g - h)(x^k)\}$ is decreasing.
2) If $(g - h)(x^{k+1}) = (g - h)(x^k)$, then $x^k$ and $x^{k+1}$ are critical points of $(P_{DC})$ and DCA terminates at the $k$th iteration.
3) If $\rho(g) + \rho(h) > 0$, then $\sum_{k=1}^{\infty} \| x^{k+1} - x^k \|^2$ converges.
4) If the optimal value $\alpha$ of the problem $(P_{DC})$ is finite and the sequences $\{x^k\}$ and $\{y^k\}$ are bounded, then every limit point $\hat{x}$ of $\{x^k\}$ is a critical point of $g - h$.  

Authorized licensed use limited to the terms of the applicable license agreement with IEEE. Restrictions apply.
B. Some Notions in Probability Theory

1) History of a Stochastic Process: Given a stochastic process \( \mathcal{X} = \{X_t\}_{t=1}^{\infty} \), we define the history up to time \( k \) of \( \mathcal{X} \) by \( \mathcal{P}_k = \{X_1, X_2, \ldots, X_k\} \), where \( \sigma(X_1, X_2, \ldots, X_k) \) is the sigma algebra generated by random variables \( \{X_1, X_2, \ldots, X_k\} \). The sequence of increasing sigma algebras \( \{\mathcal{P}_k\} \) is called filtration.

2) Rademacher Average: For a set of points \( \{z_1, z_2, \ldots, z_l\} \subseteq \mathbb{Z} \), the Rademacher average \( R_l(z, z') \) is defined as \( R_l(z, z') = \mathbb{E}_z \sup_{w \in \mathbb{Z}} \sum_{i=1}^{l} \sigma_i g(w, z_1) \), where \( \sigma_i \) are i.i.d. random numbers such that \( \sigma_i \in \{-1, 1\} \) with \( \mathbb{P}(\sigma_i = 1) = \mathbb{P}(\sigma_i = -1) = 1/2 \). The Rademacher average of a family of functions \( \{g(\cdot, z) : z \in \mathbb{Z}\} \), denoted by \( R_l(g, \mathbb{Z}) \), is defined as \( R_l(g, \mathbb{Z}) = \sup_{z_1, z_2, \ldots, z_l} R_l(g, z_1, \ldots, z_l) \).

IV. Proposed Methods

This section develops osDCA schemes for solving the problem (2) which can be described as follows.

A. Problem Setting

Let \( P_Z \) be the probability distribution of \( Z \) on \( \mathbb{R}^n \) and \( \Xi = \text{supp}(P_Z) \) be the support of \( P_Z \). By definition, a point \( x \in \mathbb{R}^n \) is in support of \( P_Z(N_i) > 0 \), for all neighborhood \( N_i \) of \( x \). Since a measure "lives" in its support, we only need to work in \( \Xi \) instead of \( \mathbb{R}^n \). For instance, a Dirac measure \( \delta_a \) concentrating at a single point \( a \) admits a support containing only one point \( a \); a discrete measure \( \mu = \sum_{i \in A} \beta_i \delta_{a_i} \) with \( \beta_i > 0 \) admits a support \( \{a_1, a_2, \ldots, a_l\} \). A basic property of \( \Xi \) is that it is closed in \( \mathbb{R}^n \). Moreover, \( P_Z(\Xi^c) = 0 \) since \( \mathbb{R}^n \) is the topological Hausdorff space and \( P_Z \) is a Radon measure in \( \mathbb{R}^n \). Therefore, only the values of \( g \) and \( h \) on \( S \times \Xi \) matter. For simplicity of presentation, we assume that \( dom g = dom h = S \times \Xi \). That is, the value of \( g \) and \( h \) outside \( S \times \Xi \) is set to \(+\infty\). This implies that \( dom G = dom H = S \) [recall that \( G(\omega) := \mathbb{E}(g(\omega, Z)) \) and \( H(\omega) := \mathbb{E}(h(\omega, Z)) \)]. Here, we use the convention \(+\infty - (+\infty) = +\infty\). Moreover, \( g \) and \( h \) are assumed to be bounded below and Borel measurable, and \( g(w, Z), h(w, Z) \) are integrable for all \( w \in S \). It is noted that the Borel sigma algebra on \( \mathbb{R} \cup \{+\infty\} \) is generated by the order topology of \( \mathbb{R} \cup \{+\infty\} \). As \( g(w, Z) \) and \( h(w, Z) \) are convex, lower semicontinuous for all \( z \in \Xi \), so are \( G \) and \( H \). In the sequel, we will consider the problem (2) in its DC form

\[
\min \{F(w) = G(w) - H(w) : w \in S\}.
\]

Moreover, we need some mild assumptions as follows.

**Assumption 1:**
1) For all \( z \in \Xi \), dom \( \tilde{h}(\cdot, z) = S \).
2) \( \tilde{h} := \text{inf}_{z \in \Xi} \rho((g(\cdot, z), 0)) > 0 \).
3) There exists a Borel measurable selector \( \tau \) such that \( \forall \omega \in S, \tau(w, z) \in \text{dom} h(w, z) \) where \( \tau \) is \( L^2 \) uniformly bounded in the sense that there exists a Borel measurable function \( \tilde{\tau} \) such that \( \tilde{\tau}(Z)^2 \) is integrable and \( \forall \omega \in S, \tau \in \Xi, \|\tau(w, z)\| \leq \tilde{\tau}(z) \).

4) \( \sup_{z \in \Xi} |F(w, z)| < +\infty \).

**Remark 1:** It is observed that assumptions 1), 3), and 4) are mild. On another hand, thanks to the regularization technique introduced in [16], Assumption 1-2) is easily fulfilled by adding an \( L_2 \) regularizer to both DC components.

**Assumption 2:**
1) There exists a Borel measurable function \( \tilde{g} : \mathbb{R}^n \to \mathbb{R} \) such that \( \tilde{g}(Z) \) is integrable and \( |g(w, z)| \leq \tilde{g}(z) \), \( \forall w \in S, z \in \Xi \).

2) \( R_k(g, \Xi) \leq N_k/k^a \), with \( N_k > 0 \) and \( a > 0 \).

It is noteworthy that Assumption 2-2) holds for various cases described as follows [34].

**Case 1 (Holder functions):** \( g(\cdot, z), z \in \Xi \): Let \( D \) be the length of a cube in \( \mathbb{R}^n \) containing the compact set \( S \). Suppose that \( \exists M, L > 0 \) and \( \gamma \in (0, 1) \) such that the following conditions hold.

1) \( |g(w, z)| \leq M, \forall w \in S, z \in \Xi \).

2) \( |g(x, z) - g(y, z)| \leq L \|x - y\|^\gamma, \forall x, y \in S, z \in \Xi \).

Then, for any \( a \in (0, 1/2) \), \( R_k(g, \Xi) \leq N_k/k^a \), where

\[
N_k = LD^2 m^\gamma + \frac{M \gamma^a}{\sqrt{(1-2a)\gamma}}.
\]

**Case 2 (Holder functions):** \( g(w, \cdot), w \in S \): Suppose that \( \Xi \) is compact, let \( D \) be the length of a cube in \( \mathbb{R}^n \) that contains \( \Xi \). Suppose that there exists \( M, L, \gamma > 0 \) such that the following conditions hold.

1) \( |g(w, z)| \leq M, \forall w \in S, z \in \Xi \).

2) \( |g(w, u) - g(w, v)| \leq L \|u - v\|^\gamma, \forall w \in S, u, v \in \Xi \).

Then, \( R_k(g, \Xi) \leq N_k/k^a \), where \( N_k = M + L D^2 n^{\gamma/2} \) and \( a = \gamma/(2\gamma + n) \).

**Case 3 (discrete set \( \Xi \)):** Suppose that the number of elements of \( \Xi \) is finite, say \( |\Xi| = N_\Xi \). Furthermore, assume that there exists \( M > 0 \) such that \( |g(w, z)| \leq M, \forall w \in S, z \in \Xi \). Then, \( R_k(g, \Xi) \leq M (N_\Xi/k)^{1/2} \), hence, \( a = 1/2 \).

It turns out that Assumption 2 is not strong, hence a class of functions meeting the criteria is wide to cover many problems arising in practice. In three cases of Rademacher complexity presented above, though \( a \) in case 2 can be very small in the high-dimensional regime, which makes our next algorithm impractical, the other two cases have \( a = 1/2 \) or arbitrarily near to \( 1/2 \), which are appropriate sample rates in practice. It should be stressed that the Rademacher complexity measures the richness of a class of functions. Therefore, roughly speaking, the function \( g \) must be quite "simple" in this Rademacher sense. This criterion naturally fulfills our demand to control the variability of SAs made on \( g \).

B. Online Stochastic DCA Schemes

We introduce an osDCA scheme described in Algorithm 1. Algorithm 1 is well defined with probability 1. To be more specific, the set of events that makes Algorithm 1 work is \( V = \bigcap_{k=1}^{\infty} \bigcap_{i=1}^{\infty} (Z_{k,i} \in \Xi) \) and hence \( \mathbb{P}(V) = 1 \). We denote \( Z_k = Z_{k, \text{last}}, \sigma = \sigma(Z_0, Z_1, \ldots, Z_{k-1}, w^0, w^1, \ldots) \), and \( \mathcal{P}_k = \mathcal{P} \). We observe that \( \{w^k\}_{k=0}^{\infty} \) is a predictable process and \( \{t^k\}_{k=0}^{\infty} \) is an adapted process with respect to the filtration \( \{\mathcal{P}_k\}_{k=0}^{\infty} \). The convergence results of Algorithm 1 are presented in theorem 1.

**Theorem 1:** Under Assumptions 1 and 2, let \( \beta = \min\{\alpha, 1\} \), if the sequence of sample sizes \( \{n_k\} \) satisfies \( \sum_{k=1}^{\infty} n_k^{-\beta} < +\infty \), the iterations of Algorithm 1 satisfy the following conditions.
Algorithm 1 osDCA

Initialization: Choose $w^0 \in S$ and a sequence of sample sizes $\{n_k\}$, set $k = 0$.

repeat

1. Draw independently $n_k$ samples $Z_{k,1}, \ldots, Z_{k,n_k}$ from the distribution of $Z$ in such a way that they are also independent of the past.
2. Compute $t^k = (1/n_k) \sum_{i=1}^{n_k} 1(w^k, Z_{k,i})$.
3. Compute $w^{k+1}$, an optimal solution to the problem
   $$\min \left\{ \frac{1}{n_k} \sum_{i=1}^{n_k} g(w, Z_{k,i}) - (t^k, w) : w \in \mathbb{R}^m \right\}. \quad (5)$$

4. Set $k = k + 1$.

until Stopping criterion.

1. There exists $F^\infty$ integrable such that $F(w^k) \to F^\infty$ almost surely (a.s.).
2. $\sum_{k=1}^{\infty} \|w^{k+1} - w^k\|^2 < +\infty$ a.s.
3. There exists a measurable set $\mathcal{L} \subset \Omega$ with $\mathbb{P}(\mathcal{L}) = 1$ such that for each $\omega \in \mathcal{L}$, every limit point of $\{w^k(\omega)\}$ is a critical point of $F = G - H$.

The proof of Theorem 1 is deferred to the Appendix.

Remark 2: 1) The algorithm only uses samples at the current time to update the solution (past samples are no longer used). Therefore, even if the distribution of $Z$ changes at a certain time (suppose that, due to some real-world events, $Z$ becomes $Z'$ at the iteration $k$), the algorithm will automatically solve the problem (2) with $Z$ being replaced by $Z'$. Indeed, the current solution $w^k$ can be considered as the initial point for restart, the algorithm continues operating based on new samples from the distribution of $Z'$. Theorem 1 is still valid, and the subsequential convergence with probability one to DC critical points of the DC problem associated with the new distribution is guaranteed. This is indeed an advantage of the osDCA. In contrast, intuitively, stochastic algorithms using aggregated updates (still using old samples to compute the current solution) barely have this kind of adaptivity. We will conduct numerical experiments to study this aspect.

2) Our algorithm and the convergence analysis can be extended to deal with the more general problem whose the random variables inside the first and the second DC components are not necessarily the same, that is, $F(w) = \mathbb{E}(g(w, Z)) + \mathbb{E}(h(w, Z))$. At the iteration $k$, we approximate values of $G$ and the subgradients of $H$ by using $n_k$ independent random samples from the distribution of $Z$ and $\tilde{n}_k$ independent random samples from the distribution of $\tilde{Z}$, respectively. The sample size sequences $\{n_k\}$ and $\{\tilde{n}_k\}$ need to increase in such a way that $\sum_{k=1}^{\infty} n_k^{-1} < \infty$ and $\sum_{k=1}^{\infty} \tilde{n}_k^{-1} < \infty$.

Next, we will discuss two scenarios where one can directly compute (without stochastic approximation) values of $G$ or $H$. Since the information of $G$ (resp. $H$) can be achieved, we will modify Algorithm 1 to exploit this advantage. Note that these two schemes are not special cases of Algorithm 1, but they will coincide with Algorithm 1 in some cases [see Remark 3-1].

1) $G$ Can Be Directly Computed Without Approximation: In this case, $G$ does not need to be stochastically approximated, we replace the approximation of $G$ in step 3 of Algorithm 1 by its true value, which results in Algorithm 2.

Algorithm 2 osDCA With Exact $G$

Algorithm 1, in which the problem (5) in step 3 is replaced by $\min\{G(w) - (t^k, w) : w \in \mathbb{R}^m\}$.

With this algorithm, we obtain stronger convergence results since $G$ is computed exactly. Note that, in the convergence results of Algorithm 1, we impose Assumption 2 to control the variance of the stochastic estimator of $G$. To study the convergence of Algorithm 2, we do not need such an assumption. Furthermore, in Assumption 1, we replace the convexity condition $\rho_h + \inf_{z \in \mathbb{R}^m} \rho(h(z)) > 0$ by the weaker one $\rho_h + \rho_G > 0$, which gives rise to a milder assumption called Assumption 1'. We obtain the convergence theorem 2 whose proof is similar to that of Theorem 1.

Theorem 2: Under Assumptions 1' and 2, if the sequence of sample sizes $\{n_k\}$ satisfies $\sum_{k=1}^{\infty} n_k^{-1} < +\infty$, then the iterations of Algorithm 2 satisfy the following conditions.

1) There exists $F^\infty$ integrable such that $F(w^k) \to F^\infty$ a.s.
2) $\sum_{k=1}^{\infty} \|w^{k+1} - w^k\|^2 < +\infty$ a.s.
3) There exists a measurable set $\mathcal{L} \subset \Omega$ with $\mathbb{P}(\mathcal{L}) = 1$ such that for each $\omega \in \mathcal{L}$, every limit point of $\{w^k(\omega)\}$ is a critical point of $F = G - H$.

2) $H$ Can Be Directly Computed Without Approximation: In this case, we replace the stochastic estimator of the subgradient of $H$ in Algorithm 1 with the true subgradient of $H$ to obtain the following algorithm.

Algorithm 3 osDCA With Exact $H$

Algorithm 3, in which step 2 is replaced by
2. Compute $t^k = \partial H(w^k)$.

Since we work directly on $H$, we replace Assumption 1-1) by dom $\partial H = S$. Likewise, Assumption 1-3) is replaced by: there exists $M > 0$ such that $\forall w \in S$, $\forall t \in \partial H(w) : \|t\| \leq M$. These modifications bring about a new set of assumptions called Assumption 1". We obtain the following convergence results whose proof is similar to that of Algorithm 1.

Theorem 3: Under Assumptions 1" and 2, if the sequence of sample sizes $\{n_k\}$ satisfies $\sum_{k=1}^{\infty} n_k^{-1} < +\infty$, the iterations of Algorithm 3 satisfy:

1) There exists $F^\infty$ integrable such that $F(w^k) \to F^\infty$ a.s.
2) $\sum_{k=1}^{\infty} \|w^{k+1} - w^k\|^2 < +\infty$ a.s.
3) There exists a measurable set $\mathcal{L} \subset \Omega$ with $\mathbb{P}(\mathcal{L}) = 1$ such that for each $\omega \in \mathcal{L}$, every limit point of $\{w^k(\omega)\}$ is a critical point of $F = G - H$.

Remark 3: 1) When $g(w, z)$ [resp. $h(w, z)$] does not depend on $z$, the function $G$ (resp. $H$) is deterministic, that is, it is directly determined without approximation, hence Algorithm 2 (resp. Algorithm 3) coincides with Algorithm 1. In some cases, thanks to the flexibility of
DC decomposition of \( f(w, z) := g(w, z) - h(w, z) \), one can choose a suitable DC decomposition so that either \( g \) or \( h \) does not depend on \( z \). For instance, if \( f(·, z) \) are \( L \)-smooth with the same constant \( L \) for all \( z \in \mathcal{Z} \), \( f \) has the following DC decomposition: 
\[
f(w, z) = ((L/2)\|w\|^2 - ((L/2)\|z\|^2 - f(w, z)).
\]
In another hand, if there exists a convex function \( \phi(w) \) such that \( f(w, z) + \phi(w) \) are convex for all \( z \in \mathcal{Z} \) (in particular, when \( \phi(w) = (\kappa/2)\|w\|^2 \), \( f(·, z) \) is weakly convex), \( f \) has the following DC decomposition: 
\[
f(w, z) = f(w, z) + \phi(w) - \phi(w).
\]
2) Apart from the use of the above appropriate DC decomposition \( g - h \) in (2) so that \( G \) or \( H \) to be a deterministic function and thereafter Algorithm 2 or 3 is applicable, the exact computation of \( G \) and \( H \) requires additional knowledge about \( Z \). For example, suppose that the distribution of \( Z \) is unknown but the distribution of its measurement, \( \mu(Z) \), is known (\( P_\mu \)). For instance, \( \mu \) can be the magnitude \( \|Z\| \), a partial observation \( Z^* \) (\( Z^* \) is a part of the vector \( Z \)), the difference \( Z_1 - Z_2 \) [where \( Z = (Z_1, Z_2) \)], and so on. In such cases, if \( g(w, ·) \) is a function with respect to \( \mu, \) that is, \( g(w, z) = u(w, \mu(z), \) the expectation \( \mathbb{E}(g(w, z)) = \int u(w, \mu) dP_\mu \) which is possibly computed. Similar arguments apply to \( h(w, z) \).

3) In big data analytics, large-sum problems play a key role. We consider the following large-sum objective function:
\[
F(w) = \sum_{i=1}^{N} a_i f_i(w) = \sum_{i=1}^{N} a_i g_i(w) - \sum_{i=1}^{N} a_i h_i(w)
\]
where \( g_i, h_i \) are convex, \( a_i \geq 0 \) for all \( i = 1, N \) and \( \sum_{i=1}^{N} a_i = 1 \). The function \( F \) can be rewritten as \( F(w) = \mathbb{E}(g_i(w)) - \mathbb{E}(h_i(w)) \), where \( I \) is a random index with \( \mathbb{P}(I = i) = a_i \). In this case, the distribution of \( I \) is known completely. However, as \( N \) can be very large, we may still need to apply osDCA schemes. Furthermore, since \( I \) is known, we have full freedom to choose Algorithm 1, Algorithm 2, or Algorithm 3 to apply, which leads to—in general—three distinctive algorithms. The practical tradeoff between these algorithms would be which DC component (or none of them) is cheaper to be computed directly.

V. APPLICATIONS: SOLVING THE EXPECTED PCA

PCA is arguably one of the most successful tools for dimensionality reduction. In this section, we will apply osDCA schemes to the E-PCA to study the generalization capacity of the proposed methods.

A. osDCA Schemes for Solving Expected PCA

We consider the following E-PCA as follows [11]:
\[
\min \left\{ -\frac{1}{2} \mathbb{E}((w, Z)^2) : \|w\| \leq 1 \right\}, \quad (\text{E-PCA})
\]
where \( Z \) is a normalized random vector, that is, \( \|Z\| = 1 \), with unknown distribution. The situation in which we are interested is the data obtained online. The problem (E-PCA) can be considered as the theoretical problem of the classic PCA [and—vice versa—the classic PCA is the empirical problem of (E-PCA)]. The problem (E-PCA) aims to generalize the compressing capacity of the classical PCA on unseen data.

First, we see that the problem (E-PCA) is nonconvex and it can be formulated as a DC problem of the form (4), where
\[
G(w) = \frac{\lambda}{2} \|w\|^2, \quad H(w) = \mathbb{E}
\]
\[
\frac{1}{2} \|w\|^2 + \frac{1}{2} (w, Z)^2
\]
\[
S = \{ w \in \mathbb{R}^m : \|w\| \leq 1 \} \quad \text{and} \quad \lambda > 0.
\]
Although we have a very natural DC decomposition with \( G(w) = 0 \), \( H(w) = \mathbb{E}((1/2)(w, Z)^2) \), here we add \( (\lambda/2)\|w\|^2 \) to both DC components to fulfill Assumption 1-2). Since the values \( G \) are directly obtained without approximation, Algorithm 1 coincides with Algorithm 1. We call this scheme osDCA-1, where the \( k \)th iteration is described as follows.

1) Receive \( n_k \) samples \( Z_{k,1}, \ldots, Z_{k,n_k} \).
2) Compute \( t^k = \frac{\lambda}{\alpha} w^k + (1/\alpha) \sum_{i=1}^{n_k} (w^k, Z_{k,i})Z_{k,i} \).
3) Update \( w^{k+1} = \frac{1}{\lambda} t^k + \frac{\alpha}{\lambda} w^k \) if \( \|t^k\| \leq \lambda \), otherwise.

Second, it is well known that if a function \( \theta \) has \( L \)-Lipschitz continuous gradient, then \( (L/2)\|w\|^2 - \theta \) and \( (L/2)\|w\|^2 + \theta \) are convex. Therefore, we have another DC decomposition for the problem (E-PCA) as follows:
\[
G(w) = \mathbb{E}
\]
\[
\frac{L}{2} \|w\|^2 - \frac{1}{2} (w, Z)^2
\]
\[
H(w) = \mathbb{E}
\]
\[
\frac{L}{2} \|w\|^2 + \frac{1}{2} (w, Z)^2.
\]
Since \( G \) and \( H \) remain unknown, we apply Algorithm 1 to this DC problem. Obviously, the family \( \{g(·, z) : \|z\| = 1\} \) is uniformly Lipschitz and uniformly bounded by a constant, therefore, the rate \( r \) in Assumption 2 can be chosen arbitrarily in \((0, 1/2)\). With this setup, we obtain a second scheme called osDCA-2 whose the \( k \)th iteration is described as follows.

1) Receive \( n_k \) samples \( Z_{k,1}, Z_{k,2}, \ldots, Z_{k,n_k} \).
2) Compute the stochastic gradient
\[
t^k = L w^k + \frac{1}{n_k} \sum_{i=1}^{n_k} (w^k, Z_{k,i})Z_{k,i}.
\]
3) Solve the following convex program to get \( w^{k+1} \):
\[
\min_{w \in S} \left\{ \frac{L}{2} \|w\|^2 - \frac{1}{2m_k} \sum_{i=1}^{n_k} (w, Z_{k,i})^2 - (t^k, w) \right\}.
\]
The problem (8) is convex and can be solved by an existing convex optimization solver. Thanks to the efficiency of DCA, we use again it to solve (8), via the following “false” DC decomposition:
\[
\tilde{g}(w) = \frac{L}{2} \|w\|^2, \quad \tilde{h}(w) = \frac{1}{2m_k} \sum_{i=1}^{n_k} (w, Z_{k,i})^2 + (t^k, w).
\]
We get a simple DCA scheme where the solutions of convex subproblems have closed-form. The (deterministic) DCA takes the current solution \( w^0 \) as the initial point, then operates until the stopping criterion which is \( \|u^{k+1} - u^k\| < \epsilon \) is met, where \( \epsilon > 0 \) is the error tolerance.
B. Numerical Experiments

1) Datasets: The numerical experiments are conducted on standard machine learning datasets on LIBSVM. The information of the used datasets is described in Table I. The samples of each dataset are normalized as $\|x_i\| = 1$.

Furthermore, to test the adaptive ability of osDCA schemes, we generate a synthetic dataset that consists of two subdatasets [training set (200 000 × 500), test set (500 000 × 500)] and [training set (200 000 × 500), test set (200 000 × 500)], in which the generating mechanism is described in Section V-B3.

2) Comparative Algorithms: We compare our algorithms with two versions of projected stochastic subgradient (PSS) method [8]—an online algorithm for weakly convex objective functions, four SDCA schemes [32] proposed for nonconvex, nonsmooth DC programs, and two benchmark algorithms for solving online PCA including the AdaOja [35] and MaxAP [36].

3) Experimental Setup and Results: The numerical experiments comprise five parts. The first experiment is a comparison between algorithms, the second experiment studies our algorithms’ behaviors when the DC decomposition of the problem varies, the third experiment compares between convex solvers for solving subproblems, the fourth experiment studies the adaptive capacity of osDCA schemes, and the fifth experiment compares directly the differences in the practical performance of the three proposed osDCA schemes.

a) Experiment 1: We compare osDCA schemes with two versions of PSS (constant stepsize policy and diminishing stepsize policy), four SDCA schemes, AdaOja, and MaxAP. First, we run cross-validation ten times on each dataset to find the best hyperparameters for the comparative algorithms. To be specific, for the PSS with constant stepsize, we cross-validate to find the best stepsizes

$$\alpha_k = c/k$$

where $c$ is searched from $\{0.001, 0.005, 0.01, 0.015, 0.02\}$; Meanwhile, for the PSS with adaptive stepsize $\alpha_k = c/k$, $c$ is searched from $\{4, 5, 6, \ldots , 15\}$. On the other hand, the AdaOja automatically tunes its stepsize based on the gradient information. The MaxAP uses diminishing stepsize $\alpha_k = c/k$ where $c$ is chosen from $\{1, 2, \ldots , 10\}$. For the four SDCA schemes, it should be stressed that SDCA1 and SDCA3 require the first DC component of the objective to be explicitly defined, meanwhile, SDCA2 and SDCA4 can handle the unknown first DC component. Therefore, we apply SDCA1 and SDCA3 to (6) with $\lambda = 10^{-6}$ (small $\lambda$ yields good results [32]); meanwhile, SDCA2 and SDCA4 are applied to (7) where $L = 1$. We use the sequence of equal weights for all four SDCA schemes. On the other hand, based on the theoretical analysis, the parameters of osDCA schemes are chosen as follows. For the osDCA-1, we choose the sequence of sample sizes as $n_k = k^2$, and $\lambda = 1$ which is a neutral number. For the osDCA-2, the sequence of sample sizes is chosen as $n_k = k^3$, the Lipschitz smoothness constant $L = 1$, and the tolerance error in solving subproblems $\epsilon = 10^{-5}$.

Next, the training dataset and the validation dataset are merged together and are fed to all algorithms (one pass—to meet the online setting context and guarantee the independence of samples). As a preprocessing step, each training dataset is randomly shuffled before each run. The starting points are also randomly initialized in $S$. The performance of our algorithms is measured on the test set to guarantee their generalization capability. To enhance visualization, we report the suboptimality graph $F(w_0) - F(w^*)$ averaging over 20 runs, where $F(w^*)$ is the optimal value found on each test set by computing the largest eigenvalue of the covariance matrix. Furthermore, we classify osDCA-1, SDCA1, SDCA3 in one group and osDCA-2, SDCA2, and SDCA4 in another group [since the former three use the DC decomposition (6) and the latter three use (7)] to plot them in two different figures.

All experiments are performed on a PC Intel(R) Core(TM) i7-8700 CPU @3.20 GHz of 16-GB RAM.

Figs. 1 and 2 illustrate the performance of all algorithms (we only display six out of 12 datasets). Moreover, to show the final effect of the learned system, we report in Table II, the average reconstruction errors (on test sets) when reconstructing compressed data from the learned eigenspace.

b) Comparisons between osDCA schemes and PSS, AdaOja, MaxAP: Our algorithms take a very short amount of time to pass through the training sets while obtaining really small suboptimality values, say $10^{-4}$–$10^{-5}$ in most of the cases. In contrast, the PSS with constant stepsize and the AdaOja converge slower and usually cannot obtain comparable objective values. On the other hand, PSS with diminishing stepsize performs very well and obtains similar suboptimality to osDCA schemes, where the differences are negligible. Finally, the MaxAP performs quite poorly in this experiment.

c) Comparisons between osDCA and SDCA: About the solutions’ quality, osDCA schemes and SDCA1, SDCA3, and SDCA4 are similar in all cases. Meanwhile, the SDCA2 is slightly worse in some cases (e.g., letter, SensIT Vehicle). On the other hand, the osDCA-2 usually obtains such solutions at a faster rate than SDCA2 and SDCA4.

Overall, among the top six best algorithms (osDCA-1,2, SDCA1,3,4, PSS diminishing stepsize), the time to pass through the entire training datasets (only considered in datasets where the computational time is significant) of osDCA-1 is the shortest, osDCA-2 is the second-best. In comparison, SDCA1, which is the fastest algorithm among all SDCA schemes,
TABLE II
AVERAGE RECONSTRUCTION ERRORS OF ALL STUDIED ALGORITHMS

| Algorithm          | osDCA-1 | osDCA-2 | SDCA1 | SDCA2 | SDCA3 | SDCA4 | PSS-dimi | PSS-const | AdaOja | MaxAP |
|--------------------|---------|---------|-------|-------|-------|-------|----------|-----------|--------|-------|
| letter             | 0.4412  | 0.4412  | 0.4412| 0.4413| 0.4412| 0.4412| 0.4412   | 0.4413    | 0.4423 | 0.4490|
| YearPredictionMSD  | 0.1782  | 0.1782  | 0.1782| 0.1782| 0.1782| 0.1782| 0.1782   | 0.1783    | 0.1783 | 0.1785|
| SensIT Vehicle     | 0.7054  | 0.7054  | 0.7054| 0.7050| 0.7054| 0.7054| 0.7054   | 0.7058    | 0.7212 |       |
| shuttle             | 0.1130  | 0.1130  | 0.1130| 0.1130| 0.1130| 0.1130| 0.1130   | 0.1130    | 0.1131 | 0.1140|
| covtype             | 0.1096  | 0.1096  | 0.1096| 0.1096| 0.1096| 0.1096| 0.1096   | 0.1097    | 0.1097 | 0.1096|
| SUSY               | 0.4550  | 0.4550  | 0.4550| 0.4550| 0.4550| 0.4550| 0.4550   | 0.4551    | 0.4550 | 0.4566|
| codrna             | 0.0151  | 0.0151  | 0.0151| 0.0151| 0.0151| 0.0151| 0.0151   | 0.0152    | 0.0152 | 0.0157|
| madelon            | 0.0037  | 0.0037  | 0.0037| 0.0037| 0.0037| 0.0037| 0.0037   | 0.0038    | 0.0038 | 0.0047|
| mushroom           | 0.6228  | 0.6032  | 0.6032| 0.6043| 0.6031| 0.6031| 0.6039   | 0.6044    | 0.6069 | 0.6082|
| satimage          | 0.5478  | 0.5478  | 0.5478| 0.5478| 0.5478| 0.5478| 0.5478   | 0.5479    | 0.5489 | 0.5567|

Fig. 1. Performance of osDCA-1 compared with SDCA1, SDCA3, two versions of PSS, AdaOja, and MaxAP. (a) SensIT Vehicle. (b) shuttle. (c) letter. (d) YearPredictionMSD. (e) a7a. (f) SUSY.

Fig. 2. Performance of osDCA-2 compared with SDCA2, SDCA4, two versions of PSS, AdaOja, and MaxAP. (a) SensIT Vehicle. (b) shuttle. (c) letter. (d) YearPredictionMSD. (e) a7a. (f) SUSY.

is 2.8–7.7 times slower than osDCA-1; PSS with diminishing stepsize is 2.9–7.1 times slower than osDCA-1.

Furthermore, it is well known that there are two main factors needed to be carefully considered when designing any DCA (or its variants), namely the DC decomposition of the problem and the convex solver for subproblems. Therefore, we consider the following experiments to study our proposed algorithms’ behaviors within these two mentioned perspectives.

d) Experiment 2: Our aim is to study the behavior of osDCA-1 when $\lambda$ varies (change the DC decomposition of the problem). It is observed that to surely fulfill the strong convexity condition $\rho_G + \rho_H > 0$, we add the regularization term $\lambda \cdot \|\cdot\|^2$ to both $G$ and $H$ components. A natural question raised is: suppose $H$ is already strongly convex, will we obtain some “optimal” performance if we do not use this regularization term? This curiosity motivates us to perform the osDCA-1 scheme with DC decomposition $g(w, z) = 0$, $h(w, z) = (1/2)(w, z)^2$. Before presenting the experimental results, let us discuss a little bit about the condition $\rho_H > 0$ in this case. We know that this condition does not always hold and it is equivalent to $\mathbb{E}(Z Z^T)$ being positive definite. By definition, the positive definiteness of $\mathbb{E}(Z Z^T)$ is equivalent to $\mathbb{E}((w^T Z)^2) > 0, \forall w \neq 0$. Therefore, this condition is violated if there exists $w_0 \neq 0$ such that $\mathbb{E}((w_0^T Z)^2) = 0$, or equivalently $w_0^T Z = 0$ a.s. In other words, the condition
Fig. 3. Performance (one run) of osDCA-1 when $\lambda \geq 0$ varies.

Fig. 4. Performance (one run) of osDCA-2 with two different convex solvers: the DCA and CPLEX. (a) SensIT Vehicle. (b) shuttle. (c) letter. (d) YearPredictionMSD.

$\rho_H > 0$ does not hold if there is a perfectly linear dependence between features of the random vector $Z$.

Fig. 3 shows the behaviors of osDCA-1 with different $\lambda > 0$ and an extreme case where $\lambda = 0$ on the YearPredictionMSD dataset. We observe that the optimal performance of osDCA-1 is achieved at some moderate values of $\lambda$, say, from 1 to 5. Besides, the quality of the performance is not monotone with respect to $\lambda$. With a large value of $\lambda$, osDCA-1 somehow gets stuck at the beginning. The performance of osDCA-1 is gradually improved as $\lambda$ decreases up to a certain value, and then the performance slightly deteriorates as $\lambda$ continues to approach 0.

e) Experiment 3: We study the performance of osDCA-2 with different convex solvers for subproblems. To be specific, besides the (deterministic) DCA used in the osDCA-2 scheme, we want to use the industrial CPLEX for solving the convex subproblems. Fig. 4 shows the difference between osDCA-2 using deterministic DCA and CPLEX for solving convex subproblems. It is observed that while the suboptimality values of these two algorithms are similar, osDCA-2 using CPLEX for the convex subproblem is faster than osDCA-2 with CPLEX.

f) Experiment 4: We study the adaptive capacity of osDCA schemes compared with SDCA schemes when there is an abrupt change in the distribution of $Z$. We describe the context of the problem as follows. We are receiving streaming data from an unknown distribution (the data is—in fact—realizations of $Z$). At a certain time, suppose that there is a real-world event that makes the distribution of $Z$ change ($Z$ becomes some $Z'$). We do not know this event (and hence the change of $Z$ is also unknown to us) and continue to receive streaming data from the changed distribution. From that time, we want to solve (2) with $Z$ being replaced by $Z'$ since the new random variable $Z'$ is more relevant than $Z$.

To this end, we generate a synthetic dataset as follows. The dataset consists of two subdatasets representing data collected before and after the abrupt change. The first subdataset includes a training set (200,000×500) and a test set (500,000×500) that are generated from multivariate normal distribution with a mean vector $0$ and a positive definite covariance matrix $\Sigma = \Lambda^T \Lambda + I$, where $\Lambda$ is randomly generated. Then, we change $\Lambda$ and generate the second subdataset consisting of a training set (200,000×500) and a test set (200,000×500). All data is then normalized as $\|z_i\| = 1$. We concatenate two training sets to create one unified training set to feed to the algorithms. Before the change, we measure the performance of each algorithm on the first test set, and after the change, we use the second test set. Fig. 5 shows the average results of 20 runs, here we separate the results into two subfigures because the running times of SDCA2, SDCA3, SDCA4 are remarkably longer than osDCA-1, osDCA-2, and SDCA1. The numerical results confirm the adaptive capacity of osDCA schemes over SDCA schemes. Indeed, after the abrupt change, osDCA schemes quickly regain suboptimality values that are as good as the ones obtained before the change. Meanwhile, SDCA schemes barely adapt to the change and decrease the suboptimality slowly.

g) Experiment 5: Finally, we compare the practical performance of the three proposed algorithms in a case on which they are all applicable, that is, $G$ and $H$ can be exactly computed. Let us suppose that the random variable $Z$ is known to have a discrete distribution over a set $\Xi$ of $N$ elements, $\mathbb{P}(Z = z_i) = a_i$ for all $z_i \in \Xi$, where $a_i > 0$ and $\sum_{i=1}^{N} a_i = 1$. We use the DC decomposition (7) to test the performance of the three algorithms. Since $a_i$ are known, we can compute $G(w)$ and $\nabla H(w)$ exactly. The probability vector $a = (a_1, a_2, \ldots, a_N)$ is randomly generated. Fig. 6 shows the difference in the practical performance of our three proposed algorithms. We observe that Algorithms 2 and 3 always obtain better solutions in comparison to Algorithm 1. This result can be explained as Algorithms 2 and 3 use more information about the DC decomposition than Algorithm 1 which approximates both DC components at each iteration.
VI. CONCLUSION

We have designed three online stochastic algorithms based on the famous DCA to handle stochastic nonsmooth, nonconvex DC programs. The first scheme stochastically approximates both DC components, while the other two are designed for the context that one of two DC components can be directly computed. Our analysis and results of numerical experiments suggest that the use of the second or the third osDCA, whenever possible, and when the calculation of exact expectations is not expensive, is highly recommended. The convergence properties of the proposed algorithms are rigorously studied, and the almost sure convergence to critical points is established. As online stochastic algorithms, the osDCA schemes gain a competitive edge when dealing with streaming data. The benefits of osDCA schemes include remedying storage burden and the ability to adapt to new changes in data distribution, as well as the efficiency and flexibility of DCA. On the other hand, it is well known that the variance of stochastic estimators of online stochastic algorithms is high, which creates difficulties in convergence analysis, especially in nonconvex and nonsmooth settings. Our algorithms’ convergence results hold, thanks to the increase in sample sizes. Moreover, the rate of this increase is determined based on the Rademacher complexity of the family of functions \( \{ g(\cdot, z) : z \in Z \} \). Nevertheless, such complexity is not always easy to compute. In future works, we would like to improve this condition and provide a better rate. To study the practical behaviors of the proposed algorithms, we conduct numerical experiments on the E-PCA with streaming data from an unknown distribution. The numerical experiments justify the efficiency of the proposed algorithms. Indeed, the proposed osDCA schemes obtain good solutions within a short time. In addition, the adaptive capacity of osDCA schemes has been confirmed: after a change in the data distribution, our algorithms quickly adapt to the new distribution. As a comparison, SDCA schemes do not have this ability. Further experimental insights confirm the important role of DC decompositions and convex solvers for subproblems in DCA. It should be noted that DCA plays a double role in these osDCA schemes, as it has also been used for solving the convex subproblems. And numerical experiments showed that the (deterministic) DCA is a very efficient and robust convex solver. Finally, via a direct comparison between the three proposed algorithms, it seems that the use of the second or the third osDCA, when possible, actually improve the performance of Algorithm 1.

APPENDIX

Proof of Theorem 1: First, let \( \nu(w) := \mathbb{E}(\tau(w, Z)) \). Since \( \nu(w^k) \in \mathbb{E} H(w^k) \)

\[
H(w^{k+1}) \geq H(w^k) + \nu(w^k) + (w^{k+1} - w^k) + \frac{\rho H}{2} \|w^{k+1} - w^k\|^2. \tag{9}
\]

On the other hand, it follows from the definition of \( w^{k+1} \) that:

\[
\frac{1}{n_k} \sum_{i=1}^{n_k} g(w^k, Z_{k,i}) \geq \frac{1}{n_k} \sum_{i=1}^{n_k} g(w^{k+1}, Z_{k,i}) + \langle t^k, w^k - w^{k+1} \rangle + \frac{\inf_{z \in Z} \rho (g(\cdot, z))}{2} \|w^{k+1} - w^k\|^2. \tag{10}
\]

From (9) and (10), we obtain

\[
\frac{1}{n_k} \sum_{i=1}^{n_k} g(w^{k+1}, Z_{k,i}) \leq H(w^{k+1}) + \frac{1}{n_k} \sum_{i=1}^{n_k} g(w^k, Z_{k,i}) - \frac{\rho H}{2} \|w^{k+1} - w^k\|^2 + \langle t^k - \nu(w^k), w^{k+1} - w^k \rangle. \tag{11}
\]

with \( \bar{\rho} = \rho_H + \inf_{z \in Z} \rho (g(\cdot, z)). \) By taking conditional expectation with respect to \( \mathcal{P}_k \) both sides of (11), we obtain

\[
\mathbb{E} (F(w^{k+1}) - F(w^k) | \mathcal{P}_k) \leq \mathbb{E} (\langle t^k - \nu(w^k), w^{k+1} - w^k \rangle | \mathcal{P}_k) + \mathbb{E} (G(w^{k+1}) - \frac{1}{n_k} \sum_{i=1}^{n_k} g(w^{k+1}, Z_{k,i}) | \mathcal{P}_k) - \frac{\bar{\rho}}{2} \mathbb{E} (\|w^{k+1} - w^k\|^2 | \mathcal{P}_k). \tag{12}
\]

By applying Schwartz’s inequality and Holder’s inequality

\[
\mathbb{E} (\langle t^k - \nu(w^k), w^{k+1} - w^k \rangle | \mathcal{P}_k) \leq \mathbb{E} (\|t^k - \nu(w^k)\|^2 | \mathcal{P}_k)^{1/2} \mathbb{E} (\|w^{k+1} - w^k\|^2 | \mathcal{P}_k)^{1/2}. \tag{13}
\]

By using AM-GM inequality, we obtain

\[
\mathbb{E} (\|t^k - \nu(w^k)\|^2 | \mathcal{P}_k)^{1/2} \mathbb{E} (\|w^{k+1} - w^k\|^2 | \mathcal{P}_k)^{1/2} \leq \frac{1}{2\rho} \mathbb{E} (\|t^k - \nu(w^k)\|^2 | \mathcal{P}_k) + \frac{\bar{\rho}}{2} \mathbb{E} (\|w^{k+1} - w^k\|^2 | \mathcal{P}_k). \tag{14}
\]
It follows from the independence of $Z_{k,i}$ and $Z_{k,i}$ ($i \neq j$) that:

$$
\mathbb{E}(\|t^k - v(w^k)\|^2 | \mathcal{P}_k) = \frac{1}{n_k} \sum_{i=1}^{n_k} \mathbb{E}(\|t^k, Z_{k,i} - v(w^k)\|^2 | \mathcal{P}_k).
$$

We observe that

$$
\mathbb{E}(\|t^k, Z_{k,i} - v(w^k)\|^2 | \mathcal{P}_k) = \mathbb{E}_Z(\|t^k, Z\|^2) - \mathbb{V}_Z(\tau(w^k, Z)).
$$

Thus, $\mathbb{E}(\|t^k - v(w^k)\|^2 | \mathcal{P}_k) = (1/n_k) \mathbb{V}_Z(\tau(w^k, Z))$. Combining this and (12), (13), and (14), and we obtain

$$
\mathbb{E}(F(w^{k+1}) - F(w^k) | \mathcal{P}_k) \leq \frac{\mathbb{V}_Z(\tau(w^k, Z))}{2 \bar{p} n_k} + \mathbb{E}(G(w^{k+1}) - \frac{1}{n_k} \sum_{i=1}^{n_k} g(w^{k+1}, Z_{k,i}) | \mathcal{P}_k). \tag{15}
$$

Next, we make an upper bound on the right-hand side of (15). First, the (nonnegative) term $\mathbb{V}_Z(\tau(w^k, Z))$ is bounded above by $\mathbb{E}(\tau(Z)^2)$. Second, we show that

$$
\mathbb{E}\left( \sup_{w \in S} |G(w) - \frac{1}{n_k} \sum_{i=1}^{n_k} g(w, Z_{k,i})| \right) \leq 2R_{n_k}(g, \mathcal{Z}) \tag{16}
$$

To prove (16), let us first introduce “ghost samples” $Z_{k,1}, Z_{k,2}, \ldots, Z_{k,n_k}$ (similar to the arguments in [37]) that are independent of all $Z_{k,i}$ and identically distributed with $Z$. By Jensen’s inequality, we get

$$
\mathbb{E}\left( \sup_{w \in S} \left| \frac{1}{n_k} \sum_{i=1}^{n_k} g(w, Z_{k,i}) - \mathbb{E}(g(w, Z)) \right| \right) \leq \mathbb{E}\left( \left| \frac{1}{n_k} \sum_{i=1}^{n_k} \left( g(w, Z_{k,i}) - g(w, Z_{k,i}) \right) \right| \right) \leq \mathbb{E}\left( \left| \frac{1}{n_k} \sum_{i=1}^{n_k} g(w, Z_{k,i}) - \frac{1}{n_k} \sum_{i=1}^{n_k} g(w, Z_{k,i}) \right| \right) \leq \mathbb{E}\left( \sup_{w \in S} \left| \frac{1}{n_k} \sum_{i=1}^{n_k} g(w, Z_{k,i}) - \mathbb{E}(g(w, Z)) \right| \right).
$$

Now let $\sigma_1, \sigma_2, \ldots, \sigma_n$ be independent random variables with $\mathbb{P}(\sigma_i = 1) = \mathbb{P}(\sigma_i = -1) = (1/2)$ in such a way that they are also independent of $Z_{k,i}$ and $Z_{k,i}$. Then,

$$
\mathbb{E}\left( \sup_{w \in S} \left| \frac{1}{n_k} \sum_{i=1}^{n_k} g(w, Z_{k,i}) - \frac{1}{n_k} \sum_{i=1}^{n_k} g(w, Z_{k,i}^\prime) \right| \right) \leq \mathbb{E}\left( \mathbb{E}_\sigma \left( \sup_{w \in S} \left| \frac{1}{n_k} \sum_{i=1}^{n_k} \sigma_i (g(w, Z_{k,i}) - g((w, Z_{k,i}^\prime)) \right| \right) \right) \leq 2 \mathbb{E}(R_{n_k}(g, Z_{k,n_k})) \leq 2R_{n_k}(g, \mathcal{Z}) = 2R_{n_k}(g, \mathcal{Z}).
$$

Now, we establish the almost sure convergence of the sequence $\{F(w^k)\}$ as follows. The assumption 1-4 implies that there exists $R$ such that $F(w) \geq R, \forall w \in S$. Let $D(w) = F(w) - R \geq 0$ and $S_k = \left[ \mathbb{E}(D(w^{k+1}) - D(w^k)) \right] > 0$. Since $S_k$ is $\mathcal{P}_k$-measurable and by using (15) and (16), we obtain

$$
\sum_{k=1}^{\infty} \mathbb{E}(1_{S_k}(D(w^{k+1}) - D(w^k))) \leq \sum_{k=1}^{\infty} \mathbb{E}(\left| \frac{1}{2 \bar{p} n_k} \sum_{k=1}^{\infty} \mathbb{V}_Z(\tau(w^k, Z)) \right| + 2 \sum_{k=1}^{\infty} R_{n_k}(g, \mathcal{Z})) \leq \frac{1}{2 \bar{p} n_k} \sum_{k=1}^{\infty} \frac{1}{n_k} + 2 \sum_{k=1}^{\infty} \frac{1}{n_k} \leq +\infty.
$$

It follows from semimartingale convergence theorem [38] that there exists $D^\infty$ integrable such that $D(w^k) \rightarrow D^\infty$ a.s., which implies $F(w^k) \rightarrow F^\infty = D^\infty + R$ a.s.

Second, by AM-GM inequality, $(\|t^k - v(w^k)\|^2, w^{k+1} - w^k) \leq (1/\bar{p}) \|t^k - v(w^k)\|^2 + (\bar{p}/4) \|w^{k+1} - w^k\|^2$.

Combining this inequality with (11), we get

$$
\frac{1}{\bar{p}} \|w^{k+1} - w^k\|^2 \leq F(w^k) - F(w^{k+1}) + G(w^{k+1}) - G(w^k) \leq \frac{1}{\bar{p}} \|t^k - v(w^k)\|^2.
$$

By applying Lebesgue dominated convergence theorem (see [39, Th. 4.2]), we get

$$
\frac{1}{\bar{p}} \mathbb{E}\left( \sum_{k=1}^{\infty} \|w^k - w^{k+1}\|^2 \right) \leq \mathbb{E}(F(w^1)) - \mathbb{E}(F^\infty) + \frac{M}{\bar{p}} \sum_{k=1}^{\infty} \frac{1}{n_k} + 2N_{\bar{p}} \sum_{k=1}^{\infty} \frac{1}{n_k} < \infty.
$$

Therefore, $\sum_{k=1}^{\infty} \|w^k - w^{k+1}\|^2 < +\infty$ a.s.

Third, denote by $G_k(w) = (1/n_k) \sum_{i=1}^{n_k} g(w, Z_{k,i})$. It follows from $t^k \in \partial G_k(w^{k+1})$ and $v(w^k) \in \partial H(w^k)$ that:

$$
\langle w^{k+1}, t^k \rangle = G_k(w^{k+1}) + G^*_k(v^k),
$$

$$
\langle v^k, w^k \rangle = H^*(v^k) + H^*(w^k).
$$

Then we obtain

$$
G_k(w^k) - H(w^k) \geq H^*(v(w^k)) - G^*_k(v^k) + \langle t^k - v(w^k), w^k \rangle \geq G_k(w^{k+1}) - H(w^{k+1}) + \langle t^k - v(w^k), w^k - w^{k+1} \rangle
$$

which implies

$$
G_k(w^k) - H(w^k) - H^*(v(w^k^k)) + G^*_k(v^k) \rightarrow 0 \tag{17}
$$

since $G_k(w^k) - H(w^k) \rightarrow F^\infty$, $\langle t^k - v(w^k), w^k \rangle \rightarrow 0$, and $G_k(w^{k+1}) - H(w^{k+1}) \rightarrow F^\infty$.

Hence, (17) implies $G(w^k) + G^*_k(v^k) \rightarrow 0$.

On another hand, from the definition of the conjugate functions $G^*$ and $G^*_k$, we get

$$
|G^*_k(v^k) - G^*(v^k)| \leq \sup_{x \in S} |G_k(x) - G(x)| \rightarrow 0.
$$

Thus, $G(w^k) + G^*_k(v^k) - (w^k, v(w^k)) \rightarrow 0$ a.s. Now let $\mathcal{L}$ be an intersection of sets with probability 1 gained from all a.s. true statements from the beginning of the proof,
we have \( P(\mathcal{L}) = 1 \) since there are at most countably finite statements. Let \( \omega \in \mathcal{L} \), we have \( \{w^b(\omega)\} \) and \( \{v(w^b(\omega))\} \) are bounded. Let \( w^* \in S \) be a limit point of \( \{w^b(\omega)\} \), there exists a subsequence \( \{w^b(\omega)\} \) such that \( w^b(\omega) \to w^* \). By extracting a subsequence of \( \{w^b(\omega)\} \) if necessary, we can assume that \( v(w^b(\omega)) \to v^* \), which implies \( t^b(\omega) \to v^* \). Therefore, \( G(w^b(\omega)) + G(t^b(\omega)) \to (w^*, v^*) \). Letting \( j \to +\infty \) and noting that \( \theta(w, z) = G(w) + G(z) \) is lower semicontinuous, we obtain \( G(w^*) + G(v^*) \leq (w^*, v^*) \). On the other hand, according to Young's inequality, \( G(w^*) + G(v^*) \geq (w^*, v^*) \). Therefore, \( G(w^*) + G(v^*) = (w^*, v^*) \). In other words, \( v^* \in \partial G(w^*) \). Furthermore, for each \( w \in S \), it follows from \( v(w^b(\omega)) \in \partial H(w^b(\omega)) \) that \( H(w) \geq H(w^b(\omega)) + \langle v(w^b(\omega)), w - w^b(\omega) \rangle \), which implies \( H(w) \geq H(w^*) + \langle v^*, w - w^* \rangle \). Therefore, \( v^* \in \partial H(w^*) \), and we conclude that \( w^* \) is a critical point of \( F = G - H \) since \( \partial G(w^*) \cap \partial H(w^*) \neq \emptyset \).

ACKNOWLEDGMENT

The authors are thankful to the editor and five anonymous referees who provided valuable and constructive comments that helped them improve the presentation of this article.

REFERENCES

[1] H. A. Le Thi and T. Pham Dinh, “DC programming and DCA: Thirty years of developments,” Math. Program., vol. 169, pp. 5–68, May 2018.
[2] H. A. Le Thi, T. Pham Dinh, H. M. Le, and X. T. Vo, “DC approximation method,” Ann. Oper. Res., vol. 22, no. 1, pp. 289–355, 1997.
[3] H. A. Le Thi, H. M. Le, D. N. Phan, and B. Tran, “DC approximation for minimizing a large sum of DC functions with application to training deep Boltzmann machines,” in Proc. 23rd Int. Conf. Mach. Learn., vol. 153, pp. 62–76, Apr. 2015.
[4] A. Nitanda and T. Suzuki, “Stochastic difference of convex algorithm and its application to training deep Boltzmann machines,” in Proc. 20th Int. Conf. Artif. Intell. Stat., Fort Lauderdale, FL, USA, 2017, pp. 470–478.
[5] H. A. Le Thi, T. Pham Dinh, and H. P. H. Luu, “Stochastic difference-of-convex-functions algorithms for nonconvex programming,” SIAM J. Optim., vol. 23, no. 3, pp. 2263–2293, Sep. 2012.
[6] M. Metivier, Semimartingales, Berlin, Germany: Walter de Gruyter, 1982.
[7] A. Montanari and E. Richard, “Non-negative principal component analysis: Message passing algorithms and sharp asymptotics,” IEEE Trans. Inf. Theory, vol. 62, no. 3, pp. 1458–1484, Mar. 2016.
[8] Y. Xu, Q. Qi, Q. Lin, R. Jin, and T. Yang, “Stochastic optimization for DC functions and non-smooth non-convex regularizers with non-asymptotic convergence,” in Proc. 36th Int. Conf. Mach. Learn., 2019, pp. 6942–6951.
[9] H. M. Le, H. A. Le Thi, and M. C. Nguyen, “Sparse semi-supervised support vector machines by DC programming and DCA,” Neurocomputing, vol. 153, pp. 62–76, Apr. 2015.

Authorized licensed use limited to the terms of the applicable license agreement with IEEE. Restrictions apply.
Hoai An Le Thi received the Ph.D. degree (Hons.) in optimization and the Habilitation degree from the University of Rouen, Rouen, France, in 1994 and 1997, respectively.

She is currently serving as a Professor, in an exceptional class, with the Université de Lorraine, Metz, France. She is the Co-Founder (with Pham Dinh Tao) of DC programming and DCA, an innovative approach in nonconvex programming. She has authored/coauthored more than 270 journal articles, international conference papers, and book chapters, is the coeditor of 24 books and/or special issues of international journals, and is the supervisor of 35 Ph.D. theses. Her research interests include machine learning, optimization, and operations research, and their applications in information systems and various complex industrial systems.

Dr. Le Thi is nominated as a Senior Member of the IUF (Academic Institute of France) in May 2021 and was a recipient of the 2021 Constantin Caratheodory Prize of the International Society of Global Optimization, which is awarded biannually to an individual (or a group) for fundamental contributions to theory, algorithms, and applications of global optimization. She has been the President of the Scientific Committee and the Organizing Committee as well as a member of the Scientific Committee of various international conferences and has been the Head of several regional/national/international projects.

Hoang Phuc Hau Luu received the master’s degree in applied mathematics from Sorbonne Paris Nord University, Villetaneuse, France, in 2019. He is currently pursuing the Ph.D. degree in computer science with the Université de Lorraine, Metz, France.

His research interests include nonconvex stochastic optimization, machine learning, Markov chain Monte Carlo, and computer vision.

Tao Pham Dinh received the D.Sc. degree in numerical analysis and optimization from the University of Joseph Fourier-Grenoble I, Grenoble, France, in 1981.

Since 1989, he has been appointed as a Professor at the National Institute for Applied Sciences (INSA)-Rouen, University of Normandie, Saint-Étienne-du-Rouvray, France, where he is currently a Full Professor and was the Leader of the Team “Modeling, Optimization and Operations Research” (LMI) from 1989 to 2010. He has authored more than 210 articles, is an Editor of eight books and ten special issues of international journals, and is a supervisor of 50 Ph.D. theses. His research interests include numerical analysis, optimization, and operations research (and their applications in data mining-machine learning, bioinformatics, image processing, and computer vision, cryptology, finance, mechanics, communication systems, transport-logistics, petrochemical engineering, robotics and optimal control, and supply chain and management), where he has introduced dc programming and DCA in their preliminary form in 1985 and extensively developed, with Le Thi Hoai An, these theoretical and algorithmic tools since 1994 for them to become now classic and increasingly popularized by researchers and practitioners all over the world.

Dr. Pham Dinh has been a member of the Scientific Committee of many International Conferences. He has been the Chair of many Scientific Committees and the Chair of the Organizing Committee of six international conferences.