Asynchronous Delay-Aware Accelerated Proximal Coordinate Descent for Nonconvex Nonsmooth Problems

Ehsan Kazemi*, **Liqiang Wang**
Department of Computer Science, University of Central Florida
ehsan_kazemy@knights.ucf.edu, lwang@cs.ucf.edu

Abstract

Nonconvex and nonsmooth problems have recently attracted considerable attention in machine learning. However, developing efficient methods for the nonconvex and nonsmooth optimization problems with certain performance guarantee remains a challenge. Proximal coordinate descent (PCD) has been widely used for solving optimization problems, but the knowledge of PCD methods in the nonconvex setting is very limited. On the other hand, the asynchronous proximal coordinate descent (APCD) recently have received much attention in order to solve large-scale problems. However, the accelerated variants of APCD algorithms are rarely studied. In this paper, we extend APCD method to the accelerated algorithm (AAPCD) for nonsmooth and nonconvex problems that satisfies the sufficient descent property, by comparing between the function values at proximal update and a linear extrapolated point using a delay-aware momentum value. To the best of our knowledge, we are the first to provide stochastic and deterministic accelerated extension of APCD algorithms for general nonconvex and nonsmooth problems ensuring that for both bounded delays and unbounded delays every limit point is a critical point. By leveraging Kurdyka-Łojasiewicz property, we will show linear and sublinear convergence rates for the deterministic AAPCD with bounded delays. Numerical results demonstrate the practical efficiency of our algorithm in speed.

Introduction

For many machine learning and data mining applications, efficiently solving the optimization problem with nonsmooth regularization is important. In this paper, we focus on the following composite optimization problem of machine learning model with nonsmooth regularization term as

$$\min_{x\in\mathbb{R}^m} F(x) = f(x) + g(x) \tag{1}$$

where $f : \mathbb{R}^m \to \mathbb{R}$ captures the empire risk which is smooth and possibly nonconvex, and $g : \mathbb{R}^m \to \mathbb{R}$, corresponding to the regularization term, reduces to a finite-sum

$$g(x) = \sum_{j=1}^{m} g_j(x_j) \tag{2}$$

where each g_j can be nonconvex.

Many problems on (1) correspond to convex model that can be efficiently optimized by first order algorithm, in particular accelerated proximal gradient (APG) methods which is proven to be efficient for the class of convex functions. However, many real applications require the problems to be nonconvex. The nonconvexity might originate either from function $f(x)$ or the regularization function. This type of problems is popular in machine learning, for example, sparse logistic regression (Liu, Chen, and Ye 2009), and sparse multi-class classification (Blondel, Seki, and Uehara 2013). On the other hand regarding the nonsmooth regularization terms, proximal gradient methods often address solving optimization problems with nonsmoothness. The proximal operator is defined as following

$$\text{Prox}_{\eta g_j}(y) = \arg\min_{x\in\mathbb{R}^m} \frac{1}{2\eta}\|x-y\|^2 + g_j(x_j)$$

where $\eta > 0$, and $\|\cdot\|$ is l_2-norm. If the proximal operator does not have an analytic solution, an algorithm should be used to solve the proximal operator which might be inexact. In this paper we consider only algorithms which use exact proximal mapping.

While the new algorithms for problem (1) provide both good theoretical convergence and empirical performances, the investigations on them were mainly conducted in the sequential setting. In the current big data era, we need to design algorithms to deal with very large scale problems (m is large). In this case, we need to eliminate sequential updates which usually take too much costly idle time. This necessitates parallel computation which will not use synchronization to wait for all others and share their updates. Recently asynchronous parallelization have received huge successes due to its potential to vastly speed up algorithms (Dean et al. 2012; Recht et al. 2011). We design and analyze an asynchronous parallel implementations of the accelerated proximal coordinate descent algorithms with bounded and unbounded delays for nonconvex nonsmooth problems, which is not well studied in the literature, to the best of our knowledge.

Contributions

The main contributions of this paper are summarized as follows. We first propose the basic stochastic and deterministic variants of asynchronous accelerated proximal coordinate

* Corresponding author.

descent algorithm for nonconvex problems. By construction of Lyapunov functions, we show that the limit points of the sequences generated by AAPCD are critical points of the problem (1) for both bounded delays and unbounded delays. This is one of the first convergence results for a method with acceleration which alleviates the bottleneck of unbounded delays for nonsmooth nonconvex functions. The convergence studies for AAPCD, through a novel perspective, characterize the stepsize based on the momentum parameter. This fills the void in previous analyses such as (Li and Lin 2015; Yao et al. 2017), where the effect of the exact value of the momentum parameter on the acceleration of convergence were not observed. As the stability of the algorithm is highly affected by asynchronism, by allowing negative momentum for high staleness values we will show the reduction in the objective function will be increased significantly and accelerates convergence. In particular, we characterize the momentum parameter in the sense that increasing the stepsize would involve decreasing of the momentum parameter, while it will provide comparable asymptotic convergence in terms of the violation of first-order optimality conditions. We will show that by requiring momentum, a fixed stepsize could be chosen for unbounded delays.

By leveraging different cases of Kurdyka-Łojasiewicz property of the objective function, we establish the linear and sub-linear convergence rates of the function value sequence generated by the deterministic AAPCD with bounded deterministic delays and they match the synchronous results. In all the cases investigated in this paper, the independence assumption between blocks and delays is avoided.

We provide numerical experiments to demonstrate the performance of our stochastic AAPCD algorithm on various large-scale real-world datasets. The results outperforms other asynchronous stochastic algorithms reported in literature such as ASCD (Liu et al. 2015) and AASCD (Fang, Huang, and Lin 2018). It also shows that AAPCD can achieve good speedup on large-scale real-world datasets and provide significantly faster convergence to a reasonable accuracy than competing options, while still providing favorable asymptotic accuracy.

Related Works

Proximal Gradient Algorithms: Proximal gradient methods for nonsmooth regularization are among the most important methods for solving composite optimization problems. There have been accelerated exact proximal gradient variants. Specifically, for convex problems, the authors in (Beck and Teboulle 2009) displayed basic accelerated proximal gradient (APG) method which extends Nesterov's accelerated methods for solving single smooth convex function (Nesterov 1983). They proved that APG displays the non-asymptotic convergence rate $O(\frac{1}{k^2})$, where k is the number of iterations.

For extensions to nonconvex settings, (Ghadimi and Lan 2016) studied the condition that only the regularization term could be nonconvex, and proved the convergence rate of APG method. (Boţ, Csetnek, and László 2016) established the convergence of proximal method when $f(x)$ and $g(x)$ could be nonconvex. (Li and Lin 2015) focused on first-order algorithms and by exploiting KL property they proved that APG algorithm can converge to a stationary point in different rates. Recently, in (Gu, Huo, and Huang 2016) and (Li et al. 2017) several accelerated proximal methods were studied, and sublinear and linear rates under different cases of the KL property for nonconvex problems were provided.

In addition to the above proximal gradient methods, several stochastic optimization methods were developed for solving composite problems see, e.g., proximal stochastic coordinate descent prox-SCD (Shalev-Shwartz and Tewari 2011), prox-SVRG (Xiao and Zhang 2014), prox-SAGA (Defazio, Bach, and Lacoste-Julien 2014), prox-SDCA (Shalev-Shwartz and Zhang 2014). Under the assumption that the regularization term is block separable, (Richtárik and Takáč 2014) developed a randomized block-coordinate descent method. An accelerated variant of this method is studied in (Lin, Lu, and Xiao 2015). All these stochastic methods require convexity of f, or even stronger assumptions.

For nonconvex problems, (Ghadimi and Lan 2016) generalized an accelerated SGD method to solve nonconvex but smooth minimization problems. Stochastic variance reduction methods for nonconvex problems were investigated in (Allen-Zhu and Hazan 2016; Reddi et al. 2016a). Furthermore, proximal variance reduction methods for general nonconvex, nonsmooth problems are proposed in (Reddi et al. 2016b; Allen-Zhu 2017). Then, (Xu and Yin 2015) proposed a block stochastic gradient method for nonconvex and nonsmooth problems.

Asynchronous Coordinate Descent: The asynchronous computation is much more efficient than the synchronous computation. More recently, asynchronous parallel methods have been successfully applied to accelerate many optimization algorithms including stochastic coordinate descent (Liu et al. 2015). We briefly review the works which are closely related to ours as follows. ASCD can provide linear and sublinear convergence rates (Liu et al. 2015; Avron, Druinsky, and Gupta 2015). Similar results were established for asynchronous SGD (Recht et al. 2011), and stochastic variance reduction algorithms (Reddi et al. 2015; Leblond, Pedregosa, and Lacoste-Julien 2017). A study of ASCD for unbounded delays has been performed in (Sun, Hannah, and Yin 2017), however the results are restricted only to Lipschitz differentiable functions. Some asynchronous algorithms particularly outperform conventional ones. In (Meng et al. 2016), authors integrated momentum acceleration and variance reduction techniques to accelerate asynchronous SGD. Several accelerated schemes for asynchronous coordinate descent and SVRG using momentum compensation techniques were proposed in (Fang, Huang, and Lin 2018). Recently, (Hannah, Feng, and Yin 2018) analyzed an asynchronous accelerated block coordinate descent algorithm with optimal complexity which converges linearly to a solution for strongly convex functions.

However, to the best of our knowledge, there is no study on the asynchronous parallel versions of accelerated proximal coordinate descent algorithms for nonconvex nonsmooth objective functions.

Preliminaries and Assumptions

We describe our asynchronous accelerated proximal coordinate descent for nonconvex problems in Algorithm 1. Compared to the regular proximal coordinate descent step, AAPCD takes an extra linear extrapolation step depending on the value of the current ages of $\hat{y}^k$, which is called also delay and denoted by d_k. In order to compute the delay d_k, we use a scalar counter to denote the weights at iteration k, starting from $k = 0$, and with each update we increment the counter by one. We allow each worker to record the iteration i when reading the weights and we let k to denote the iteration when the same worker updating the weights. Then the actual delay d_k is $d_k = k - i$. If delay is greater than the threshold T_1, we consider adding negative momentum to extrapolate a new iterate. We further show that adding such a momentum for large delays have the effect of decreasing Lyapunov function over iterations. For acceleration, AAPCD only accepts the new extrapolated iterate when the objective function value is sufficiently decreased. It is important to note that the threshold T_1 can adaptively change during the iterations. From practical point of view there is a need to know how to select the parameter T_1. We will address this question later when we present the analyses of convergence. It will be shown that accumulation points of sequences generated by AAPCD will converge to stationary points of F. In the step 5 of Algorithm 1, at iteration k, the block gradient $\nabla_{j_k} f$ is computed at the delay iterate $\hat{y}^k$, which is assumed to be some earlier state of y^k in the shared memory with the delay d_k. The delay iterate $\hat{y}^k$ can be formulated as

$$\hat{y}^k = y^k - \sum_{h \in I(k)} (y^{h+1} - y^h) \tag{3}$$

where $I(k) \in \{k-1, \ldots, k-d_k\}$ is a subset of previous iterations. From the proximal update for AAPCD, we have $x_j^{k+1} = y_j^k$ for $j \neq j_k$. We also assume $\beta = \max_k\{\beta_k \geq 0\}$ and $\beta' = \max_k\{\beta_k < 0\}$. We let Γ_k^r be the set of iterations from k to r with $d_k > T_1$ and $y^{k+1} = v^k$, ${\Gamma^c}_k^r$ denote the set of iterations from k to r with $d_k \leq T_1$ and $y^{k+1} = v^k$, and ${\Gamma^0}_k^r$ denote the set of iterations from k to r with $y^{k+1} = x^{k+1}$.

By studying different cases of KL property we will show that AAPCD will decrease the function value properly at the initial point. For the deterministic AAPCD with deterministic bounded staleness, we prove the linear and sublinear convergence rate by exploiting different cases of KL property.

In the following we first introduce some tools for analyzing asynchronous algorithms, and then describe the assumptions on the problem (1) that we assume in this paper.

For analysis of the stochastic algorithm, we let $\mathcal{F}_k$ denote the sigma algebra generated by $\{y^0, \ldots, y^k\}$. We denote the total expectation by $\mathbb{E}$ and the expectation over the stochastic variable d_k by $\mathbb{E}_{d_k}$. Function $g(x)$ is lower semicontinuous at point x_0 if $\liminf_{x \to x_0} g(x) \geq g(x_0)$. Throughout this paper, we assume each g_j in problem (1) is lower semicontinuous. A point $x \in \mathbb{R}^m$ is said a critical point of function F if $0 \in \partial F(x)$. The following Uniformized KL property is a powerful tool to analyze the first order descent algorithms.

Algorithm 1 Asynchronous Accelerated Proximal Coordinate Decent (AAPCD)

1: **Input:** The stepsize η, threshold T_1
2: **Initialize:** $y^0 \in \mathbb{R}^m$
3: **for** $k = 0, 1, \ldots, R$ **do**
4: Randomly choose j_k from $\{1, \ldots, m\}$
5: $x_{j_k}^{k+1} = \text{Prox}_{j_k, \eta g_{j_k}}(y^k - \eta \nabla_{j_k} f(\hat{y}^k))$ and $x_j^{k+1} = y_j^k$ for $j \neq j_k$
6: **if** $d_k \leq T_1$ **then** choose $\beta_k > 0$
7: $v_{j_k}^k = x_{j_k}^{k+1} + \beta_k(x_{j_k}^{k+1} - y_{j_k}^k)$ and $v_j^k = y_j^k$ for $j \neq j_k$
8: **else** choose $\beta_k < 0$
9: $v_{j_k}^k = x_{j_k}^{k+1} + \beta_k(x_{j_k}^{k+1} - y_{j_k}^k)$ and $v_j^k = y_j^k$ for $j \neq j_k$
10: **if** $F(x^{k+1}) \leq F(v^k)$ **then**
11: $y_{j_k}^{k+1} = x_{j_k}^{k+1}$
12: **else**
13: $y_{j_k}^{k+1} = v_{j_k}^k$
14: **Output:** y_{R+1}

Definition 1 (Uniformized KL Property)**.** *A function $f : \mathbb{R}^m \to (-\infty, \infty]$ is said to satisfy the Uniformized KL property if for every compact set $\Omega \subset dom\ \partial f$ on which f is constant, there exists $\epsilon, \gamma \in (0, +\infty]$ and $\phi \in \Phi_\gamma$, such that for all $\hat{u} \in \Omega$ and all $u \in \{u \in \mathbb{R}^m : dist_\Omega(u) < \epsilon\} \cap \{u \in \mathbb{R}^m : f(\hat{u}) < f(u) < f(\hat{u}) + \gamma\}$, the following inequality holds*

$$\phi'(f(u) - f(\hat{u}))dist_{\partial f(u)}(0) \geq 1$$

where Φ_γ stands for a class of function $\phi : [0, \gamma) \to \mathbb{R}^+$ satisfying: (1) ϕ is concave and C^1 on $(0, \gamma)$; (2) ϕ is continuous at 0, $\phi(0) = 0$; and (3) $\phi'(x) > 0$, for all $x \in (0, \gamma)$.

By (Bolte, Sabach, and Teboulle 2014, Lemma 6), if function f is lower semicontinuous and satisfies KL property at every point of Ω, then it satisfies the Uniformized KL property. All semi-algebraic functions satisfy the KL property. Specially, the desingularising function $\phi(t)$ of semi-algebraic functions can be chosen to take the form $\phi(t) = \frac{C}{\theta} t^\theta$ with $\theta \in (0, 1]$. In particular, typical semi-algebraic functions include real polynomial functions, $\|x\|_p$ with $p \geq 0$, rank, etc.

We make the following assumptions on the problem (1) in this paper.

Assumption 1. *Function f and each g_j are proper and lower semicontinuous; $\inf_{x \in \mathbb{R}^m} F(x) > -\infty$; the sublevel set $\{x \in \mathbb{R}^d : F(x) \leq \alpha\}$ is bounded for all $\alpha \in \mathbb{R}$.*

Assumption 2. *Function f is continuously differentiable and the gradient ∇f is L-Lipschitz continuous.*

To prove the limit points of $\{y^k\}$ generated by AAPCD are stationary points, we need a new assumption:

Assumption 3. *For AAPCD, it is assumed that there exists $K \in \mathbb{N}$ such that for all $k \in \mathbb{N}$, we have $\{1, \ldots, m\} \subseteq \{j_{k+1}, \ldots, j_{k+K}\}$.*

The goal of our paper is to provide a comprehensive analysis for AAPCD for both bounded and unbounded delays to justify the overall advantages of AAPCD.

AAPCD with Bounded Delays

In this section we analyze the convergence of Algorithms 1 for bounded delays, i.e., we assume $d_k \leq \tau$ for all k and for a fixed number τ. Define the Lyapunov function G as

$$G(x^k) := G(x^k, y^k, \ldots, y^{k-\tau}) = F(x^k) + \xi_k$$

where the sequence $\{\xi_k\}_{k\in\mathbb{N}}$, defined by

$$\xi_k := \frac{L^2\tau}{2C} \sum_{h=k-\tau+1}^{k} (h-k+\tau) \left\|y^h - y^{h-1}\right\|^2$$

with $C > 0$ is a constant to be determined later. In the lemma below, we present an inequality which states for a proper stepsize, AAPCD can provide sufficient descent in our Lyapunov function.

Lemma 1. *Suppose Assumption 2 hold. Given $\eta > 0$, we have*

$$\mathbb{E}[G(x^{k+1})] \leq \mathbb{E}[G(y^k)] - \left(\frac{1}{2\eta} - \frac{L}{2} - L\tau(1+\beta_k)\right) \mathbb{E}\left\|x^{k+1} - y^k\right\|^2. \quad (4)$$

We characterize the convergence of AAPCD. Our first result characterizes the behavior of the limit points of the sequence generated by AAPCD. Based on the lemma, we show that the sequence $\{y^k\}$ generated by AAPCD approaches critical points of the general nonconvex problem (1).

Theorem 1. *Let Assumptions 1-3 hold for the problem* (1). *Then with stepsize $\eta < \frac{1}{L+2LT_1(1+\beta)}$, and the momentum $-1 < \beta_k < \frac{1}{L\tau}(\frac{1}{2\eta} - \frac{L}{2}) - 1$ the sequence $\{y^k\}$ generated by AAPCD satisfies*

1. *$\{y^k\}$ is an almost surely bounded sequence and $\mathbb{E}\left\|y^{k+1} - y^k\right\| \to 0$.*
2. *The set of limit points of $\{y^k\}$ forms a compact set, on which function F is a constant F^* and the sequences $\{F(y^k)\}$ and $\{G(y^k)\}$ converge to F^*.*
3. *All the limit points of $\{y^k\}$ are critical points of F, and $\mathbb{E}[dist_{\partial F(y^k)}(0)] = o(\frac{1}{\sqrt{k}})$.*

Remark 1. *The connectedness and compactness of the set Ω of the limit points of $\{y^k\}$ is implied from $E\left\|y^{k+1} - y^k\right\| \to 0$. Theorem 1 also states that the objective function on Ω containing the critical points remains constant.*

Remark 2. *Equation* (4) *shows that the selection of negative β_k for substantial staleness values would increase Lyapunov function reduction over an iteration. In the light of the bounds for the momentum term β_k in Theorem 1, we could realize an estimation of an upper bound for the threshold T_1 in AAPCD algorithm. The staleness bound T_1 should be large enough to allow positive β_k. For example if $\beta = \frac{1}{2}$, then we should have, $\frac{1}{L\tau}(\frac{1}{2\eta} - \frac{L}{2}) - 1 \geq \frac{1}{2}$. Thus, by choosing $\eta = \frac{1}{L+4LT_1(1+\beta)}$, we obtain $T_1 \geq \frac{3\tau}{4(1+\beta)} = \frac{\tau}{2}$.*

The compact set Ω satisfies the requirements of the Uniformized KL property, and hence can be utilized to show the decrease of function values, depending on a certain exponent θ defined below.

Theorem 2. *Let the conditions of Theorem 1 hold. Suppose that F satisfies the Uniformized KL property with desingularising function ϕ of the form $\phi(t) = \frac{e}{\theta}t^\theta$. Let $F(x) = F^*$ for all of the limit points of $\{x^k\}$ in AAPCD, and denote $r_k = F(x^k) - F^*$. Then the sequence $\{r_k\}$ for k large enough satisfies*

1. *If $\theta = 1$, and x_0 is chosen such that $r_0 < \frac{1}{b_1 e^2}$, then r_k reduces to zero in finite steps;*
2. *If $\theta = \frac{1}{2}$, then $\mathbb{E}[r_{k+1}] \leq \frac{b_1 e^2}{1+b_1 e^2}\mathbb{E}[r_0]$,*

where

$$b_1 = \frac{2(\frac{1}{\eta} + L)^2(K+1) + 2L^2T_1(1+\beta) + 2L^2T(1+\beta'')}{\left(\frac{1}{2\eta} - \frac{L}{2} - L\tau(1+\beta)\right)}$$

with $\beta'' = \max\{\beta', 0\}$.

Remark 3. *As $\beta'' \leq 0$, the contribution of the delays greater than T_1 in the factor b_1, i.e., $2L^2T(1+\beta'')$ decreases, which indicates acceleration is possible with negative momentum term.*

AAPCD with Unbounded Delays

In this section, we allow the delay d_k to be an unbounded stochastic variable, and extremely large delays in our algorithm are permitted. Depending on some limitations on the distribution of d_k, we can still prove convergence. For unbounded delay analysis, one approach is to consider a new bound for the distribution of the end-behavior of d_k to decay sufficiently fast as the iterations progress.

We emulate this solution in the following. In particular, we define fixed parameters p_j related to probabilities of the delay such that $p_j \geq P(j(k) = j)$, for all k, and $c_k := \sum_{t=1}^{\infty} t(t+k)p_{t+k}$ with $\sum_{k=0}^{\infty} c_k < \infty$. For instance, we note that if p_j have the probability distributions with decay bound $p_j = O(j^{-t})$, $t > 4$, then $\sum_{k=0}^{\infty} c_k$ is finite.

We define a more involved Lyapunov function G as

$$G(x^k) := G(x^k, y^k, \ldots, y^0) = F(x^k) + \xi_k \quad (5)$$

where to simplify the presentation, we define ξ_k which encompasses all terms

$$\xi_k := \frac{L^2}{2C} \sum_{h=1}^{k} c_{k-h} \left\|y^h - y^{h-1}\right\|^2.$$

where $\frac{1}{C} > 0$ is a contraction rate to be defined later.

Lemma 2. *Under Assumption 1, for any $\eta > 0$, we have*

$$\mathbb{E}[G(x^{k+1})] \leq \mathbb{E}[G(y^k)] - \left(\frac{1}{2\eta} - \frac{L}{2} - L(1+\beta_k)\sqrt{c_0}\right) \mathbb{E}\left\|x^{k+1} - y^k\right\|^2. \quad (6)$$

Now we characterize the behavior of the limit points of the sequence generated by AAPCD with unbounded delays.

Theorem 3. *Let Assumptions 1-3 hold for the problem* (1). *Then with stepsize $\eta < \frac{1}{L+2L\sqrt{c_{T_1}}(1+\beta)}$ and momentum $-1 < \beta_k < \frac{1}{L\sqrt{c_0}}(\frac{1}{2\eta} - \frac{L}{2}) - 1$, the sequence $\{y^k\}$ generated by AAPCD satisfies*

1. *$\{y^k\}$ is an almost surely bounded sequence and $\mathbb{E}[\xi_k] \to 0$.*
2. *The set of limit points of $\{y^k\}$ forms a compact set, on which the functions F is a constant F^* and $\{F(y^k)\}$ and $\{G(y^k)\}$ converge to F^*.*
3. *All the limit points of $\{y^k\}$ are critical points of F.*

Remark 4. *Lemma 2 shows that the selection of negative β_k for delays greater than T_1 would decrease Lyapunov function substantially over an iteration. The bounds for β_k in Theorem 3 imply an estimation of a lower bound for c_{T_1}. For example if $\beta = \frac{1}{2}$, then, we should have $\frac{1}{L\tau}(\frac{1}{2\eta} - \frac{L}{2}) - 1 \geq \frac{1}{2}$. Hence, by selecting $\eta = \frac{1}{L+4L\sqrt{c_{T_1}}(1+\beta)}$, we obtain $c_{T_1} \geq \frac{9c_0}{16(1+\beta)^2} = \frac{c_0}{4}$.*

Now by applying the Uniformized KL property we show Algorithm 1 decreases the objective value below that of $F(x_0)$.

Theorem 4. *Let the conditions of Theorem 3 hold and F satisfies the Uniformized KL property and the desingularising function has the form of $\phi(t) = \frac{e}{\theta}t^\theta$ with $e > 0$. We denote $r_k = F(y^k) - F^*$, where F^* is the function value on the set of limit points of $\{y^k\}$. Then for k large enough the sequence $\{r_k\}$ satisfies*

1. *If $\theta = 1$, and x_0 is chosen such that $r_0 < \frac{1}{b_1 e^2}$ then r_k reduces to zero in finite steps;*
2. *If $\theta = \frac{1}{2}$, then $\mathbb{E}[r_{k+1}] \leq \frac{b_1 e^2}{1+b_1 e^2}\mathbb{E}[r_0]$,*

where

$$b_1 = \frac{(\frac{2}{\eta^2} + 4L^2) + 4L^2 + 4L^2 c_0(1+\beta)}{\frac{1}{2\eta} - \frac{L}{2} - L(1+\beta)\sqrt{c_0}}.$$

Deterministic AAPCD

In this section, we consider deterministic unbounded delays. Specifically, deterministic AAPCD is presented in Algorithm 2. The stochastic and deterministic AAPCD differ only on how the current coordinates are selected at each iteration. For this purpose, we assume the delay variable d_k is deterministic, which allow extremely large delays in our algorithm. We will prove that a subsequence of points $\{y^k\}$ generated by deterministic AAPCD converges to a stationary point. Using KL property we will see that if x^0 is not a stationary point, Algorithm 2 decreases the objective value below that of $F(x^0)$. We also prove the rate of convergence for the deterministic algorithm with deterministic bounded delay by exploiting KL property, which is unavailable in the stochastic setting for the Lyapunov function.

As recommended in (Sun, Hannah, and Yin 2017), we set a sequence $\{\epsilon_i\}_{i\geq 0}$ and define the Lyapunov function G which encompasses all terms to control unbounded delays

$$G(x^k) := F(x^k) + \xi_k \tag{7}$$

where to simplify the presentation, we define

$$\xi_k := \frac{L^2}{2C}\sum_{h=1}^{\infty}\delta_{k-h}\left\|y^h - y^{h-1}\right\|^2$$

with $\delta_i = \sum_{j=i}^{\infty}\epsilon_j$ such that $\sum_{j=0}^{\infty}\delta_j < \infty$ and $C > 0$ to be determined later.

Algorithm 2 Deterministic AAPCD

Input: The stepsize η, threshold T_1
Initialize: $y^0 \in \mathbb{R}^m$
for $k = 0, 1, 2, \ldots, R$ **do**
  Choose j_k from $\{1, \ldots, m\}$
  $x_{j_k}^{k+1} = \text{Prox}_{j_k, \eta g_{j_k}}(y^k - \eta\nabla_{j_k} f(\hat{y}^k))$ and $x_j^{k+1} = y_j^k$ for $j \neq j_k$
  if $d_k \leq T_1$ **then** choose $\beta_k > 0$
    $v_{j_k}^k = x_{j_k}^{k+1} + \beta_k(x_{j_k}^{k+1} - y_{j_k}^k)$ and $v_j^k = y_j^k$ for $j \neq j_k$
  else choose $\beta_k < 0$
    $v_{j_k}^k = x_{j_k}^{k+1} + \beta_k(x_{j_k}^{k+1} - y_{j_k}^k)$ and $v_j^k = y_j^k$ for $j \neq j_k$
  if $F(x^{k+1}) \leq F(v^k)$ **then**
    $y_{j_k}^{k+1} = x_{j_k}^{k+1}$
  else
    $y_{j_k}^{k+1} = v_{j_k}^k$
Output: y_{R+1}

Lemma 3. *Let Assumption 2 hold. For any $\eta > 0$, we have*

$$G(x^{k+1}) \leq G(y^k) - \left(\frac{1}{2\eta} - \frac{L}{2} - \sqrt{\delta_0\mu_{d_k}}L(1+\beta_k)\right)\left\|x^{k+1} - y^k\right\|^2 \tag{8}$$

where $\mu_{d_k} = \sum_{h=0}^{d_k-1}\frac{1}{\epsilon_h}$.

For any $T \geq \liminf d_k$ which can be arbitrarily large, let S_T be the subsequence of $\mathbb{N}$ where the current delay is less than T. We will show the points x^k, $k \in S_T$, have convergence guarantees. The following theorem for unbounded deterministic delay is parallel to Theorem 3.

Theorem 5. *Suppose that Assumptions 1-3 hold. Then with stepsize $\eta = \frac{c}{L+2\sqrt{\delta_0\mu_{T_1}}L(1+\beta)}$ for $c \in (0,1)$, and momentum $-1 < \beta_k < \frac{\sqrt{\mu_{T_1}}}{c\sqrt{\mu_{d_k}}}(1+\beta) - 1$, we have,*

1. *$\{y^k\}$ is a bounded sequence and $\xi_k \to 0$.*
2. *The function F is constant on the set of limit points of $\{y^k\}$ and the sequences $\{F(y^k)\}$ and $\{G(y^k)\}$ converge to it.*
3. *For any subsequence S_T generated by the deterministic AAPCD, all the limit points of $\{y^k\}_{k\in S_T}$ are critical points of F.*

Remark 5. *Lemma 3 shows that the use of momentum for delayed gradient might gain no performance and have negative effects. Hence, to compensate this issue, we allow the selection of negative β_k for high staleness values to maximize the reduction of the Lyapunov function over an iteration. By taking the bounds in Theorem 5 for the momentum term β_k in to consideration, we could present an upper bound estimate for the threshold T_1 in AAPCD. The delay bound T_1 should be large enough to allow positive β_k. For example if we choose $\beta = 1$, then we should have, $\frac{\sqrt{\mu_{T_1}}}{c\sqrt{\mu_{d_k}}}(1+\beta) - 1 \geq 1$. Therefore, T_1 must be large enough such that $\mu_{T_1} \geq \frac{4c^2\mu_{d_k}}{(1+\beta)^2} = c^2\mu_{d_k}$, for all k.*

It is important to note that although Theorem 5 shows a fixed step size works for deterministic AAPCD, however, in return the upper bound for momentum is adaptive to the current delay.

In the following theorem, it turns out that a subsequence of Algorithm 2 can decrease the function value at x_0, depending on the parameter θ defined below.

Theorem 6. *Let conditions of Theorem 5 hold and that F satisfies the Uniformized KL property and the desingularising function has the form $\phi(s) = \frac{e}{\theta}t^\theta$, where $\theta \in (0, 1]$ and $e > 0$. Let $F(x) = F^*$ for all $x \in \Omega$ (the set of limit points), and denote $r_k = F(y^k) - F^*$. Then the sequence $\{r_k\}_{k\in S_T}$ for k large enough satisfies*

1. *If $\theta = 1$, and x_0 is chosen such that $r_0 < \frac{1}{b_1c^2}$ then r_k reduces to zero in finite steps;*
2. *If $\theta \in [\frac{1}{2}, 1)$, then for k large enough $r_k \le \frac{b_1e^2}{1+b_1e^2}r_0$;*
3. *If $\theta \in (0, \frac{1}{2})$, then $r_k \le \left(\frac{1}{b_2(1-2\theta)+r_0^{2\theta-1}}\right)^{\frac{1}{1-2\theta}}$*

where

$$b_1 = \frac{2(\frac{1}{\eta} + L)^2 + 3(1 + \beta)^2L^2T_1 + 2(1 + \beta'')^2L^2T}{(\frac{1}{c} - 1)\frac{L}{2}}$$

with $\beta'' = \max\{\beta', 0\}$ and $b_2 = \min(\frac{1}{b_1e^2R}, \frac{r_0^{2\theta-1}(R^{\frac{2\theta-1}{2\theta-2}}-1)}{1-2\theta})$ for a fixed number $R \in (1, \infty)$.

For the deterministic AAPCD with deterministic bounded delay T, we define $\epsilon_i = 0$ for $i > T$ and we let $\tilde{G}(x^k, y^k, \ldots, y^{k-T})$ denote the corresponding Lyapunov function. In the following $\tilde{G}(x)$ refers to $\tilde{G}(x, x, \ldots, x)$. We let Ω denote the set of stationary points of F. Since $\xi_k \to 0$, by Theorem 5, $\tilde{G}$ is constant on Ω. We can derive convergence rates for

$$r_k = \tilde{G}(y^k) - F^*. \tag{9}$$

Theorem 7. *Assume the conditions of Theorem 6, but only $\tilde{G}$ satisfies the Uniformized KL property and the desingularising function has the form $\phi(s) = \frac{e}{\theta}t^\theta$, where $\theta \in (0, 1]$ and $e > 0$. Then if the delay is bounded by T, the sequence $\{r_k\}$ for k large enough satisfies*

1. *If $\theta = 1$, then r_k reduces to zero in finite steps;*
2. *If $\theta \in [\frac{1}{2}, 1)$, then $r_k \le \left(\frac{b_1e^2}{1+b_1e^2}\right)^{\lfloor\frac{k-k_1}{T+K}\rfloor} r_{k_1}$ for k_1 large enough;*
3. *If $\theta \in (0, \frac{1}{2})$, then $r_k \le \left(\frac{1}{\lfloor\frac{k-k_0}{T+K}\rfloor b_2(1-2\theta)+r_0^{2\theta-1}}\right)^{\frac{1}{1-2\theta}}$,*

where

$$b_1 = \frac{3}{(\frac{1}{c} - 1)\frac{L}{2}}\left((\frac{1}{\eta} + L)^2 + (1 + \beta)^2L^2T_1 + (1 + \beta'')^2L^2T + 2(1 + \beta)^2L^2\mu_T\delta_0\right) \tag{10}$$

with $\beta'' = \max\{\beta', 0\}$ and $b_2 = \min(\frac{1}{b_1e^2R}, \frac{r_0^{2\theta-1}(R^{\frac{2\theta-1}{2\theta-2}}-1)}{1-2\theta})$ for a fixed number $R \in (1, \infty)$.

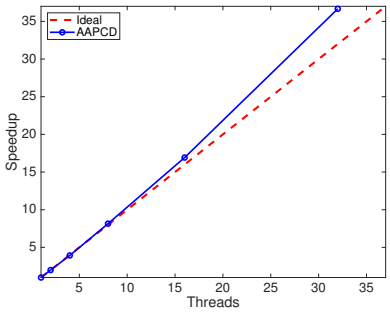

Figure 1: Speedup results of AAPCD on *rcv1* dataset.

The convergence rates in Theorem 7 match the results from (Davis 2016), but they need the independence assumption between blocks and delays. If $T = 0$ we obtain a synchronous version of the accelerated coordinate descent, and hence Theorem 7 implies the same rates as given in (Li and Lin 2015) for nonconvex functions.

Remark 6. *The characterization of the factor b_1 in Theorems 6 and 7 is noticeable in a particular way that the delays greater than T_1 contribute to this factor. Since $\beta'' \le 0$, it shows that applying negative momentum for high delay values could efficiently decrease the value of b_1 which results in acceleration.*

Remark 7. *The KL property of F is not necessarily sufficient to ensure that the Lyapunov function G satisfies the KL property. However, since $G - F$ is semi-algebraic and the class of semi-algebraic functions is closed under addition, it shows that G is semi-algebraic, which implies that G is a KL function.*

Numerical Results

In this section we test the efficiency of the asynchronous stochastic proximal coordinate descent algorithm with momentum acceleration. We performed binary classifications on the benchmark dataset *rcv1*. Following the practices in (Gong et al. 2013), we consider the logistic loss function with nonconvex regularization,

$$g(x) = \lambda \sum_{j=1}^{m} \min(|x_j|, \theta),$$

with $\lambda = 0.0001$, $\theta = 0.1\lambda$ and the zero vector as starting point. Figure 1 demonstrates the speedups of our algorithm. AAPCD has significant linear speedup on a parallel platform with shared memory compared to its sequential counterpart. We conduct experiments for comparing AAPCD with other asynchronous algorithms: ASCD (Liu et al. 2015), an synchronous version of doubly stochastic proximal algorithm (DSPG) (Zhao et al. 2014), AASCD (Fang, Huang, and Lin 2018). ASCD and DSPG did not utilize the momentum acceleration techniques. AASCD is an asynchronous accelerated variant of ASCD but only for convex and strongly convex functions. For all experiments we set the number of local workers to 32. We set $\lambda = 0.0001$, $\theta = 0.1\lambda$. For AAPCD, we set $\eta = 0.08$, $\beta = -0.08$ for negative momentum, $\beta = 0.8$

for positive momentum and threshold $T_1 = 0.9\tau$. All blocks are of size 1000. We set the stepsize for ASCD with $\eta = 0.06$. In AASCD we set $\eta = 0.09$, with momentum value $\theta_1 = 0.8$. For DSPG, the stepsize is $\eta = 0.03$ and mini-batch size is 200. All algorithms are terminated when the number of iterations exceeds 100. Note that we use the best tuned parameters for each method which is obtained over a refined grid to attain the best performance. Figure 2 shows the convergence of the objective function with respect to CPU time and the number of iterations. Towards the end AAPCD decreases

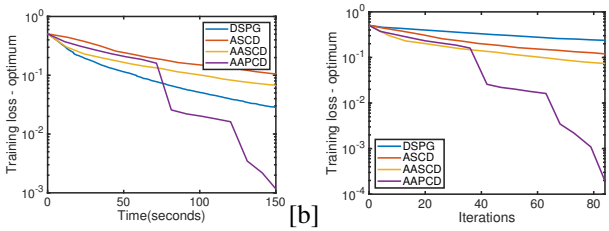

Figure 2: Figure(a) is convergence of objective value vs. time; Figure(b) is comparison of the objective function vs. iteration for different algorithms.

rapidly and needs much fewer iterations and less computing time than ASCD and AASCD to reach the same objective function values. This means that our AAPCD algorithm is very efficient and attains the best performance. Moreover AAPCD obtains a much smaller objective value by order of magnitudes compared with other algorithms. For saving space, we leave another experiment for Sigmoid loss in the supplementary materials.

Figure 3 shows AAPCD by only applying nonnegative momentum values which is slower than AAPCD, showing that linear extrapolation using negative momentum β for large delays is significantly useful.

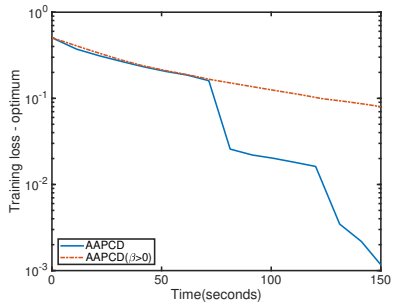

Figure 3: AAPCD versus AAPCD with momentum values $\beta > 0$.

In summary our experimental results validate that AAPCD can indeed accelerate the convergence in practice.

Conclusion

In this paper, we have studied the stochastic and deterministic asynchronous parallelization of coordinate descent algorithm with momentum acceleration for efficiently solving nonconvex nonsmooth problems. We have shown that every limit point is a critical point and proved the convergence rates for deterministic AAPCD with bounded delay. We verified the advantages of our method through numerical experiments.

Overall speaking, these asynchronous proximal algorithms can be highly efficient when being used to solve large scale nonconvex nonsmooth problems. As for future work, an extension of this study might develop the analysis in this paper to inexact proximal methods. We also plan to investigate the asynchronous parallelization of more algorithms for nonconvex nonsmooth programming for solving more complicated models.

Acknowledgements

This work is partially supported by NSF IIS-1741431 award.

References

Allen-Zhu, Z., and Hazan, E. 2016. Variance reduction for faster non-convex optimization. In *International Conference on Machine Learning*, 699–707.

Allen-Zhu, Z. 2017. Natasha: Faster non-convex stochastic optimization via strongly non-convex parameter. *arXiv preprint arXiv:1702.00763*.

Avron, H.; Druinsky, A.; and Gupta, A. 2015. Revisiting asynchronous linear solvers: Provable convergence rate through randomization. *Journal of the ACM (JACM)* 62(6):51.

Beck, A., and Teboulle, M. 2009. A fast iterative shrinkage-thresholding algorithm for linear inverse problems. *SIAM journal on imaging sciences* 2(1):183–202.

Blondel, M.; Seki, K.; and Uehara, K. 2013. Block coordinate descent algorithms for large-scale sparse multiclass classification. *Machine learning* 93(1):31–52.

Bolte, J.; Sabach, S.; and Teboulle, M. 2014. Proximal alternating linearized minimization or nonconvex and nonsmooth problems. *Mathematical Programming* 146(1-2):459–494.

Boţ, R. I.; Csetnek, E. R.; and László, S. C. 2016. An inertial forward–backward algorithm for the minimization of the sum of two nonconvex functions. *EURO Journal on Computational Optimization* 4(1):3–25.

Davis, D. 2016. The asynchronous palm algorithm for nonsmooth nonconvex problems. *arXiv preprint arXiv:1604.00526*.

Dean, J.; Corrado, G.; Monga, R.; Chen, K.; Devin, M.; Mao, M.; Senior, A.; Tucker, P.; Yang, K.; Le, Q. V.; et al. 2012. Large scale distributed deep networks. In *Advances in neural information processing systems*, 1223–1231.

Defazio, A.; Bach, F.; and Lacoste-Julien, S. 2014. Saga: A fast incremental gradient method with support for non-strongly convex composite objectives. In *Advances in neural information processing systems*, 1646–1654.

Fang, C.; Huang, Y.; and Lin, Z. 2018. Accelerating asynchronous algorithms for convex optimization by momentum compensation. *arXiv preprint arXiv:1802.09747*.

Ghadimi, S., and Lan, G. 2016. Accelerated gradient methods for nonconvex nonlinear and stochastic programming. *Mathematical Programming* 156(1-2):59–99.

Gong, P.; Zhang, C.; Lu, Z.; Huang, J.; and Ye, J. 2013. A general iterative shrinkage and thresholding algorithm for non-convex regularized optimization problems. In *International Conference on Machine Learning*, 37–45.

Gu, B.; Huo, Z.; and Huang, H. 2016. Inexact proximal gradient methods for non-convex and non-smooth optimization. *arXiv preprint arXiv:1612.06003*.

Hannah, R.; Feng, F.; and Yin, W. 2018. A2bcd: An asynchronous accelerated block coordinate descent algorithm with optimal complexity. *arXiv preprint arXiv:1803.05578*.

Leblond, R.; Pedregosa, F.; and Lacoste-Julien, S. 2017. Asaga: Asynchronous parallel saga. In *Artificial Intelligence and Statistics*, 46–54.

Li, H., and Lin, Z. 2015. Accelerated proximal gradient methods for nonconvex programming. In *Advances in neural information processing systems*, 379–387.

Li, Q.; Zhou, Y.; Liang, Y.; and Varshney, P. K. 2017. Convergence analysis of proximal gradient with momentum for nonconvex optimization. In *International Conference on Machine Learning*, 2111–2119.

Lin, Q.; Lu, Z.; and Xiao, L. 2015. An accelerated randomized proximal coordinate gradient method and its application to regularized empirical risk minimization. *SIAM Journal on Optimization* 25(4):2244–2273.

Liu, J.; Wright, S. J.; Ré, C.; Bittorf, V.; and Sridhar, S. 2015. An asynchronous parallel stochastic coordinate descent algorithm. *The Journal of Machine Learning Research* 16(1):285–322.

Liu, J.; Chen, J.; and Ye, J. 2009. Large-scale sparse logistic regression. In *Proceedings of the 15th ACM SIGKDD international conference on Knowledge discovery and data mining*, 547–556. ACM.

Meng, Q.; Chen, W.; Yu, J.; Wang, T.; Ma, Z.; and Liu, T.-Y. 2016. Asynchronous accelerated stochastic gradient descent. In *IJCAI*, 1853–1859.

Nesterov, Y. E. 1983. A method for solving the convex programming problem with convergence rate o (1/kˆ 2). In *Dokl. Akad. Nauk SSSR*, volume 269, 543–547.

Recht, B.; Re, C.; Wright, S.; and Niu, F. 2011. Hogwild: A lock-free approach to parallelizing stochastic gradient descent. In *Advances in neural information processing systems*, 693–701.

Reddi, S. J.; Hefny, A.; Sra, S.; Poczos, B.; and Smola, A. J. 2015. On variance reduction in stochastic gradient descent and its asynchronous variants. In *Advances in Neural Information Processing Systems*, 2647–2655.

Reddi, S. J.; Hefny, A.; Sra, S.; Poczos, B.; and Smola, A. 2016a. Stochastic variance reduction for nonconvex optimization. In *International conference on machine learning*, 314–323.

Reddi, S. J.; Sra, S.; Póczos, B.; and Smola, A. J. 2016b. Proximal stochastic methods for nonsmooth nonconvex finite-sum optimization. In *Advances in Neural Information Processing Systems*, 1145–1153.

Richtárik, P., and Takáč, M. 2014. Iteration complexity of randomized block-coordinate descent methods for minimizing a composite function. *Mathematical Programming* 144(1-2):1–38.

Shalev-Shwartz, S., and Tewari, A. 2011. Stochastic methods for l1-regularized loss minimization. *Journal of Machine Learning Research* 12(Jun):1865–1892.

Shalev-Shwartz, S., and Zhang, T. 2014. Accelerated proximal stochastic dual coordinate ascent for regularized loss minimization. In *International Conference on Machine Learning*, 64–72.

Sun, T.; Hannah, R.; and Yin, W. 2017. Asynchronous coordinate descent under more realistic assumptions. In *Advances in Neural Information Processing Systems*, 6182–6190.

Xiao, L., and Zhang, T. 2014. A proximal stochastic gradient method with progressive variance reduction. *SIAM Journal on Optimization* 24(4):2057–2075.

Xu, Y., and Yin, W. 2015. Block stochastic gradient iteration for convex and nonconvex optimization. *SIAM Journal on Optimization* 25(3):1686–1716.

Yao, Q.; Kwok, J. T.; Gao, F.; Chen, W.; and Liu, T.-Y. 2017. Efficient inexact proximal gradient algorithm for nonconvex problems. In *Proceedings of the 26th International Joint Conference on Artificial Intelligence*, 3308–3314. AAAI Press.

Zhao, T.; Yu, M.; Wang, Y.; Arora, R.; and Liu, H. 2014. Accelerated mini-batch randomized block coordinate descent method. In *Advances in neural information processing systems*, 3329–3337.