2021 60th IEEE Conference on Decision and Control (CDC)
December 13-15, 2021. Austin, Texas

On Robustness of the Normalized Random Block Coordinate Method for Non-Convex Optimization

Berkay Turan César A. Uribe Hoi-To Wai Mahnoosh Alizadeh

2021 60th IEEE Conference on Decision and Control (CDC) | 978-1-6654-3659-5/21/$31.00 ©2021 IEEE | DOI: 10.1109/CDC45484.2021.9682846

***Abstract*— Large-scale optimization problems are usually characterized not only by large amounts of data points but points living in a high-dimensional space. Block coordinate methods allow for efficient implementations where steps can be made (block) coordinate-wise. Many existing algorithms rely on trustworthy gradient information and may fail to converge when such information becomes corrupted by possibly adversarial agents. We study the setting where the partial gradient with respect to each coordinate block is arbitrarily corrupted with some probability. We analyze the robustness properties of the normalized random block coordinate method (NRBCM) for non-convex optimization problems. We prove that NRBCM finds an $\mathcal{O}(1/\sqrt{T})$-stationary point after T iterations if the corruption probabilities of partial gradients with respect to each block are below $1/2$. With the additional assumption of gradient domination, faster rates are shown. Numerical evidence on a logistic classification problem supports our results.**

I. INTRODUCTION

For high-dimensional optimization problems, block coordinate methods are shown to be efficient when the full gradient is expensive to compute [1]–[4]. Theoretical studies establish global convergence guarantees for deterministic [5], [6] and randomized [7]–[9] block coordinate methods. In addition to strong theoretical properties, a recent empirical study demonstrates that block-normalized gradient methods help accelerate the training of neural networks [10].

Existing methods assume that the gradients are *trustworthy*. However, due to numerous reasons, such as computational errors at the machines or data corruption, the gradients might become corrupted. Besides, distributed implementations of these methods are gaining traction and require reliable communication between the machines. However, unreliable communication might occur due to noisy wireless communication, or more importantly, due to man-in-the-middle adversarial attacks [11]. In man-in-the-middle attacks, an adversary can take over network sub-systems and arbitrarily alter the information stored and communicated between the machines to prevent convergence to the optimal solution, i.e., Byzantine attacks [12]. In these situations, methods that use raw gradient information may fail to converge, as an erroneous gradient can have an arbitrarily large effect on the algorithm.

B. Turan and M. Alizadeh are with Dept. of ECE, UCSB, Santa Barbara, CA, USA. C. A. Uribe is with Dept. of ECE, Rice University, TX, USA. H. T. Wai is with Dept. of SEEM, CUHK, Shatin, Hong Kong. This work is supported by UCOP Grant LFR-18-548175, NSF grant #1847096, and CUHK Direct Grant #4055113.. E-mails: bturan@ucsb.edu, cauribe@rice.edu, htwai@se.cuhk.edu.hk, alizadeh@ucsb.edu

In this paper, we adopt a random and adversarial corruption model, where the gradients are arbitrarily corrupted with some probability. We do not limit the corruption level. We highlight the robustness properties of the normalized random block coordinate method (NRBCM) in this setting. The NRBCM performs a block coordinate update with an adaptive step size scaled as the reciprocal of the partial gradient norm with respect to the corresponding block. We show that this method avoids large updates that corrupted gradients might cause by discarding the norm and only preserving the directional information of the gradient. This allows the algorithm to converge without any modification, even in the presence of corrupted gradients.

Our contributions can be summarized as follows:

- We prove that if the corruption probabilities of partial gradients with respect to each block are below $1/2$, the NRBCM finds an $\mathcal{O}(1/\sqrt{T})$-stationary point after T iterations for smooth (possibly non-convex) cost functions.
- For a family of cost functions satisfying a gradient domination condition, we prove that the NRBCM can: 1) either converge to a $\mathcal{O}(\gamma)$ neighborhood of the optimal solution at a linear rate with constant step sizes proportional to γ, 2) or converge to the optimal solution at a rate $\mathcal{O}(\log(t)/t)$ with decreasing step sizes at $\mathcal{O}(1/t)$, where t is the iteration index.
- We provide numerical evidence that for multi-class logistic classification task on the MNIST dataset, the NRBCM is robust to the modeled corruption.

Related work: Besides the literature on block coordinate descent type methods, our work has connections to the literature on (1) normalized gradient method and (2) optimization under corruption.

1) *Normalized gradient method:* Normalized gradient method is a well-established algorithm for convex [13], [14] and quasi-convex optimization [15]. More importantly, normalized updates for non-convex optimization [16] is gaining traction since, for non-convex objectives, the magnitude of the gradient provides less information about the value of the function, while the direction still indicates the direction of steepest descent. An important benefit of this was shown to be the fast evasion of saddle points [17] since the normalized updates will not diminish around the saddle points. A variant of normalized gradient methods is the gradient clipping technique used for privacy [18] and robustness [19].

2) *Optimization under corruption:* This line of work aims to develop optimization algorithms for learning and distributed optimization under various corruption models

[20], [21]. Adversarial learning literature commonly studies classification [22] and linear regression [23] tasks, where the corruption is due to data manipulation. For example, [24] considers a probabilistic corruption scenario called p-tampering, where the adversary is restricted to choose valid tampered data with correct labels.

On the other hand, the literature on robust distributed optimization either studies a *bounded corruption* model, e.g., due to noise [25], quantization [26], [27], and inexact oracles [28], or, studies an *arbitrary adversarial corruption* model while assuming that the adversary is only able to manipulate a certain *fraction of agents* or data samples [29]–[31]. It has been shown that in a distributed setup with multiple agents participating in the optimization process, when the majority of the agents are trustworthy, adversarial corruption can be filtered out via robust aggregation [32]–[36].

Due to the modeling differences on the corruption, none of the previous works address the problem we study in this paper. We allow *arbitrary adversarial corruption* in a centralized setup, which prevents robust aggregation to create gradient estimates. Closest to our setup is our previous work in [37], which studies robustness of normalized subgradient method in a randomly corrupted subgradient setting. However, [37] studies a full gradient type method for constrained convex optimization problems satisfying a certain acute angle condition, whereas this work considers a block coordinate descent type method for unconstrained non-convex optimization problems.

Paper Organization: The remainder of the paper is organized as follows. In Section II, we formalize the problem setup. In Section III, we describe the NRBCM (Algorithm 1) and analyze its convergence in a randomly corrupted gradient setting for smooth (possibly non-convex) cost functions. In Section IV, we provide a numerical study demonstrating the robustness of the NRBCM.

Notations. Unless otherwise specified, $\|\cdot\|$ denotes the standard Euclidean norm. Given a positive integer $q > 0$, $[q]$ denotes the set of integers $\{1, 2, \dots, q\}$. The abbreviation *a.s.* indicates almost sure convergence.

II. PROBLEM SETUP

We consider the general unconstrained optimization problem

$$f^\star = \min_{x\in\mathbb{R}^d} f(x), \tag{1}$$

where $f : \mathbb{R}^d \to \mathbb{R}$ is a continuously differentiable cost function of a decision vector $x \in \mathbb{R}^d$. We assume that the optimal solution set of (1) is nonempty. We partition the decision vector into q blocks as:

$$x = [x(1)^T x(2)^T \dots x(q)^T]^T, \tag{2}$$

where $x(i) \in \mathbb{R}^{d_i}$ with $d_i > 0$ and $\sum_i d_i = d$. Using the notation in [7], it is useful to define matrices $U_i \in \mathbb{R}^{d\times d_i}$ such that

$$[U_1 U_2 \dots U_q] = I. \tag{3}$$

With this notation, $x(i) = U_i^T x$ and $x = \sum_{i\in[q]} U_i x(i)$. Lastly, we define the partial gradient with respect to the decision variables in the i'th block as:

$$\nabla_i f(x) = U_i^T \nabla f(x),\ \forall i \in [q]. \tag{4}$$

Random block coordinate descent methods [7] are practical for iteratively solving (1). In these methods, at each iteration t, the algorithm generates a random integer $i_t \in [q]$ with the distribution

$$P(i_t = i) = \phi_i,\ \forall i \in [q], \tag{5}$$

where $\sum_{i=1}^q \phi_i = 1$ and $\Phi = [\phi_1, \phi_2, \dots, \phi_q]$ is a probability vector for the sampling distribution. Upon selecting i_t, the algorithm receives the feedback $\nabla_{i_t} f(x_t)$ and updates the i_t'th block, i.e. $x_t(i_t)$, according to the feedback.

Such methods however assume that the feedback is a trustworthy gradient information and might fail to converge when the feedback becomes corrupted, as one single corrupted feedback can have an arbitrarily large effect. In this paper, we consider the case where at each iteration t, the gradient information corresponding to block i_t is corrupted with probability p_{i_t}, potentially due to an adversarial attack. Therefore at each iteration t, the feedback corresponding to the i_t'th block is determined as:

$$h_{i_t,t} = \begin{cases} \nabla_{i_t} f(x_t) & \text{with probability } 1 - p_{i_t}, \\ U_{i_t}^T b_t & \text{with probability } p_{i_t}, \end{cases} \tag{6}$$

where the corrupted feedback b_t is arbitrary. We note that this model encompasses all the cases where the feedback can become corrupted (e.g., communication errors, computational errors, corrupted data, adversarial manipulation) since we set no restrictions on b_t.

The following section will describe the normalized random block coordinate method and state its convergence guarantees in a randomly corrupted gradient setting defined by (6) for smooth (possibly non-convex) cost functions.

III. ROBUSTNESS OF NORMALIZED RANDOM BLOCK COORDINATE METHOD

We study the normalized random block coordinate method (NRBCM) and show that it can be used to solve (1) in the random corruption setting defined by (6). The intuition behind this is that the feedback is restricted to contain only directional information by normalization. This allows us to limit the corrupted gradients' potential by not allowing arbitrarily large updates, which would have been possible without normalization.

We summarize NRBCM in Algorithm 1. At each iteration t, the algorithm selects a random block $i_t \in [q]$ according to (5). Given x_t and i_t, the algorithm receives the feedback $h_{i_t,t}$ according to (6). Then, it computes the normalized vector $U_{i_t} h_{i_t,t}/\|h_{i_t,t}\|$ as the update direction and moves the iterate x_t along that direction with step size $\gamma_{i_t,t}$. Here, U_{i_t} simply produces a d-dimensional vector by adding zeros to d_{i_t}-dimensional $h_{i_t,t}/\|h_{i_t,t}\|$ so that the update operation is feasible.

Algorithm 1 Normalized Random Block Coordinate Method

Input: Initialize $x_0 \in \mathbb{R}^d$, step sizes $\{\gamma_{i,t}\}_{\forall i \in [q]}$, sampling probability vector Φ, and T

1: **for** $t = 0$ to $T-1$ **do**
2: Select a random block $i_t \in [q]$ according to (5).
3: Given x_t and i_t, receive $h_{i_t,t}$ according to (6).
4: Update $x_{t+1} = x_t - \gamma_{i_t,t} \frac{U_{i_t} h_{i_t,t}}{\|h_{i_t,t}\|}$, where $\frac{U_{i_t} h_{i_t,t}}{\|h_{i_t,t}\|} = 0$ if $\|h_{i_t,t}\| = 0$.
5: **end for**

Before presenting the convergence results, we need to state the following technical assumption on the block coordinatewise smoothness of f:

Assumption 1. *We assume that the gradient of f is block coordinatewise Lipschitz continuous with constants $\{L_i\}_{i\in[q]}$:*

$$\|\nabla_i f(x + U_i^T h_i) - \nabla_i f(x)\| \leq L_i \|h_i\|, \tag{7}$$

for any $i \in [q], h_i \in \mathbb{R}^{d_i}$.

In our convergence analysis, we will use the block version of the standard descent lemma (e.g., [1, Proposition A.24]) for block coordinatewise smooth functions:

Lemma 1 (Block Descent Lemma [6, Lemma 3.2]). *Suppose that f is a continuously differentiable function over $\mathbb{R}^d$ satisfying* (7)*. Let $u, v \in \mathbb{R}^d$ be two vectors which differ only in the i'th block, that is, there exists an $h \in \mathbb{R}^{d_i}$ such that $v - u = U_i h$. Then*

$$f(v) \leq f(u) + \langle \nabla f(u), v - u \rangle + \frac{L_i}{2}\|u - v\|^2. \tag{8}$$

We can now present the main technical result on the convergence of NRBCM for block coordinatewise smooth cost functions in a randomly corrupted gradient setting:

Theorem 1. *Suppose that f is a continuously differentiable function over $\mathbb{R}^d$ for which Assumption 1 holds. Let $f_{lb} \leq f^\star$ be a known lower bound for the optimal value of* (1) *and $\bar{p}_i \geq p_i$, $\forall i \in [q]$, be known upper bounds on the corruption probabilities. If $\bar{p}_i < 1/2$ for all $i \in [q]$, then Algorithm 1 with parameters*

$$\phi_i = \frac{L_i^{1/2}/(1-2\bar{p}_i)}{\sum_{j=1}^q L_j^{1/2}/(1-2\bar{p}_j)}, \ \forall i \in [q], \tag{9}$$

$$\gamma_{i,t} = \sqrt{\frac{2(f(x_0) - f_{lb})}{T L_i}}, \ \forall i \in [q], \forall t, \tag{10}$$

produces iterates that satisfy the following:

$$\frac{1}{T}\sum_{t=0}^{T-1} \mathbb{E}[\|\nabla f(x_t)\|] \leq \sqrt{\frac{2(f(x_0) - f_{lb})}{T}} \sum_{i=1}^q \frac{L_i^{1/2}}{(1-2\bar{p}_i)} \tag{11}$$

Proof: We will first prove the convergence result for general ϕ_i and $\gamma_{i,t}$, and then prove that the parameters in (9) and (10) minimize the upper bound. Starting with Lemma 1:

$$f(x_{t+1}) \leq f(x_t) + \langle \nabla f(x_t), x_{t+1} - x_t \rangle + \frac{L_{i_t}}{2}\|x_{t+1} - x_t\|^2 \tag{12}$$

$$= f(x_t) + \langle \nabla f(x_t), -\gamma_{i_t,t} \frac{U_{i_t} h_{i_t,t}}{\|h_{i_t,t}\|} \rangle + \frac{L_{i_t}\gamma_{i_t,t}^2}{2} \tag{13}$$

$$= f(x_t) - \gamma_{i_t,t}(1 - Y_{i_t,t}) \langle \nabla f(x_t), \frac{U_{i_t} U_{i_t}^T \nabla f(x_t)}{\|U_{i_t}^T \nabla f(x_t)\|} \rangle - \gamma_{i_t,t} Y_{i_t,t} \langle \nabla f(x_t), \frac{U_{i_t} U_{i_t}^T b_t}{\|U_{i_t}^T b_t\|} \rangle + \frac{L_{i_t}\gamma_{i_t,t}^2}{2}, \tag{14}$$

where $Y_{i_t,t}$ is the Bernoulli random variable indicating whether the gradient for updating the i_t'th block at iteration t is corrupted or not, where $Y_{i_t,t} = 0$ corresponds to the event that the gradient is trustworthy. Next, we observe that both $U_{i_t} U_{i_t}^T \nabla f(x_t)$ and $U_{i_t} U_{i_t}^T b_t$ have non-zero entries only at the i_t'th block and hence rewrite (14) as:

$$f(x_{t+1}) \leq f(x_t) - \gamma_{i_t,t}(1 - Y_{i_t,t}) \langle \nabla_{i_t} f(x_t), \frac{U_{i_t}^T \nabla f(x_t)}{\|U_{i_t}^T \nabla f(x_t)\|} \rangle - \gamma_{i_t,t} Y_{i_t,t} \langle \nabla_{i_t} f(x_t), \frac{U_{i_t}^T b_t}{\|U_{i_t}^T b_t\|} \rangle + \frac{L_{i_t}\gamma_{i_t,t}^2}{2} \tag{15}$$

$$\overset{(a)}{\leq} f(x_t) - \gamma_{i_t,t}(1 - Y_{i_t,t})\|\nabla_{i_t} f(x_t)\| + \gamma_{i_t,t} Y_{i_t,t} \|\nabla_{i_t} f(x_t)\| + \frac{L_{i_t}\gamma_{i_t,t}^2}{2} \tag{16}$$

$$= f(x_t) - \gamma_{i_t,t}(1 - 2Y_{i_t,t})\|\nabla_{i_t} f(x_t)\| + \frac{L_{i_t}\gamma_{i_t,t}^2}{2}, \tag{17}$$

where (a) uses the Cauchy-Schwarz inequality. Next, we take expectation conditioned on x_t:

$$\mathbb{E}[f(x_{t+1})|x_t] \leq f(x_t) - \sum_{i\in[q]} \phi_i \gamma_{i,t}(1 - 2p_i)\|\nabla_i f(x_t)\| + \sum_{i\in[q]} \frac{L_i \phi_i \gamma_{i,t}^2}{2} \tag{18}$$

$$\leq f(x_t) - \sum_{i\in[q]} \phi_i \gamma_{i,t}(1 - 2\bar{p}_i)\|\nabla_i f(x_t)\| + \sum_{i\in[q]} \frac{L_i \phi_i \gamma_{i,t}^2}{2}. \tag{19}$$

We select $\gamma_{i,t} = \gamma/(\phi_i(1-2\bar{p}_i))$ for some γ, which will be determined later, and rearrange:

$$\sum_{i\in[q]} \gamma \|\nabla_i f(x_t)\| \leq f(x_t) - \mathbb{E}[f(x_{t+1})|x_t] + \sum_{i\in[q]} \frac{\gamma^2 L_i}{2\phi_i(1-2\bar{p}_i)^2}. \tag{20}$$

Noting that $\|\nabla f(x_t)\| \leq \sum_{i\in[q]} \|\nabla_i f(x_t)\|$, we take expectation with respect to x_t and divide both sides by γ:

$$\mathbb{E}[\|\nabla f(x_t)\|] \leq \frac{\mathbb{E}[f(x_t) - f(x_{t+1})]}{\gamma} + \sum_{i\in[q]} \frac{\gamma L_i}{2\phi_i(1-2\bar{p}_i)^2}. \tag{21}$$

We sum both sides from $t=0$ to $T-1$, divide by T, and note that $f(x_T) \geq f^\star \geq f_{lb}$ to obtain:

$$\frac{1}{T}\sum_{t=0}^{T-1}\mathbb{E}[\|\nabla f(x_t)\|] \leq \frac{f(x_0)-f_{lb}}{\gamma T} + \sum_{i\in[q]}\frac{\gamma L_i}{2\phi_i(1-2\bar{p}_i)^2}. \tag{22}$$

The RHS of the above inequality is minimized for $\gamma = \sqrt{2(f(x_0)-f_{lb})/(T\sum_{i=1}^{q}\frac{L_i}{\phi_i(1-2\bar{p}_i)^2})}$ as:

$$\frac{1}{T}\sum_{t=0}^{T-1}\mathbb{E}[\|\nabla f(x_t)\|] \leq \sqrt{\frac{2(f(x_0)-f_{lb})\sum_{i=1}^{q}\frac{L_i}{\phi_i(1-2\bar{p}_i)^2}}{T}}. \tag{23}$$

In order to minimize the bound in (23) with respect to Φ, we need to solve the following convex optimization problem:

$$\min_{\{\phi_i\}_{i\in[q]}} \quad \sum_{i=1}^{q}\frac{L_i}{\phi_i(1-2\bar{p}_i)^2} \tag{24}$$

$$\text{subject to} \quad \sum_{i=1}^{q}\phi_i = 1. \tag{25}$$

Let λ be the dual variable associated with the constraint (25) and write the Lagrangian as:

$$\mathcal{L}(\{\phi_i\}_{i\in[q]},\lambda) = \sum_{i=1}^{q}\frac{L_i}{\phi_i(1-2\bar{p}_i)^2} + \lambda\sum_{i=1}^{q}\phi_i - \lambda. \tag{26}$$

The first order optimality condition requires:

$$\frac{\partial\mathcal{L}\{\phi_i\}_{i\in[q]},\lambda)}{\partial\phi_i} = -\frac{L_i}{\phi_i^2(1-2\bar{p}_i)^2} + \lambda = 0, \tag{27}$$

and therefore $\phi_i \propto \frac{L_i^{1/2}}{(1-2\bar{p}_i)}$, $\forall i \in [q]$. Accordingly, we get the optimal solution as

$$\phi_i = \frac{L_i^{1/2}/(1-2\bar{p}_i)}{\sum_{j=1}^{q}L_j^{1/2}/(1-2\bar{p}_j)}, \ \forall i\in[q]. \tag{28}$$

Plugging the expression for ϕ_i into (23) yields the desired result in (11). Lastly, plugging (28) into $\gamma_{i,t} = \gamma/(\phi_i(1-2\bar{p}_i))$ with $\gamma = \sqrt{2(f(x_0)-f_{lb})/(T\sum_{i=1}^{q}\frac{L_i}{\phi_i(1-2\bar{p}_i)^2})}$ results in step sizes given by (10). ∎

According to Theorem 1, there exists a point $\tilde{x} \in \{x_0,...,x_{T-1}\}$, generated by Algorithm 1, such that

$$\mathbb{E}\|\nabla f(\tilde{x})\| = \mathcal{O}\Big(\frac{1}{\sqrt{T}}\Big).$$

We established that $\{\phi_i\}_{i\in[q]}$ in (9) are indeed optimal in the sense that they minimize the upper bound on $T^{-1}\sum_{t=0}^{T-1}\mathbb{E}[\|\nabla f(x_t)\|]$. In particular, (9) states that the probability of choosing block i is proportional to $L_i^{1/2}$, and inversely proportional to $(1-2\bar{p}_i)$. Firstly, the proportionality to L_i^{α} for some $\alpha\in\mathbb{R}$ is consistent with the literature on random coordinate descent methods [7]. In our case, ignoring the dependency on $(1-2\bar{p}_i)$, $\phi_i = \mathcal{O}(L_i^{1/2})$ along with $\gamma_{i,t} = \mathcal{O}(L_i^{-1/2})$ leads to an algorithm that in expectation moves along the full gradient to achieve best convergence rates. Secondly, inverse proportionality to $(1-2\bar{p}_i)$ implies that blocks prone to corruption with higher probability should be selected with higher probability, in other words, more often. Intuitively, a block update that can be corrupted with higher probability requires larger number of updates so that trustworthy gradients dominate the corrupted gradients in the long run. Analytically, this choice establishes that in expectation, the update direction is along the full gradient.

In addition to the convergence guarantees highlighted by Theorem 1, Algorithm 1 can achieve a faster rate of convergence for a family of cost functions that are called *gradient dominated*. In particular, we borrow the definition from [38] and state the following assumption:

Assumption 2. *We assume that f satisfies the $(1,\mu)$-Gradient Domination condition, i.e., there exists $\mu>0$ s.t.*

$$\|\nabla f(x)\| \geq \mu(f(x)-f^\star), \quad \forall x\in\mathbb{R}^d. \tag{29}$$

In the literature, functions that satisfy the above assumption are also referred to as *gradient dominated of order* $p=\infty$ [39], [40]. Examples of cost functions that meet this assumption are log barrier functions, e.g., $f(x) = -\log((x-a)(b-x))$ for $b>a$, and exponential functions, e.g., $f(x)=\exp(c|x|)$ for some $c>0$. For this type of functions, the next theorem states the convergence result of NRBCM, under both decreasing and constant step size schemes:

Theorem 2. *Suppose that f is a continuously differentiable function over $\mathbb{R}^d$ for which Assumptions 1 and 2 hold. Let $\bar{p}_i \geq p_i$, $\forall i\in[q]$, be known upper bounds on the corruption probabilities. If $\bar{p}_i < 1/2$ for all $i\in[q]$, then Algorithm 1 with parameters*

$$\phi_i = \frac{L_i^{1/2}/(1-2\bar{p}_i)}{\sum_{j=1}^{q}L_j^{1/2}/(1-2\bar{p}_j)}, \ \forall i\in[q], \tag{30}$$

generates iterates that have the follwing properties depending on the choice of $\gamma_{i,t}$:

1) If $\gamma_{i,t} = \gamma/(\phi_i(1-2\bar{p}_i))$ for some $\gamma\in(0,1/\mu)$, then:

$$\mathbb{E}[f(x_T)-f^\star] \leq (f(x_0)-f^\star)(1-\gamma\mu)^T + \frac{\gamma}{2\mu}\sum_{i=1}^{q}\frac{L_i^{1/2}}{(1-2\bar{p}_i)}. \tag{31}$$

2) If $\gamma_{i,t} = 1/(\phi_i(1-2\bar{p}_i)\mu(t+1))$, then:

$$\mathbb{E}[f(x_T)-f^\star] \leq \frac{1+\log T}{2\mu^2 T}\sum_{i=1}^{q}\frac{L_i^{1/2}}{(1-2\bar{p}_i)}, \tag{32}$$

and $\lim_{t\to\infty} f(x_t) = f^\star, \quad a.s.$

The proof can be found in Appendix A. Theorem 2 shows that under Assumption 2, Algorithm 1 exhibits a faster convergence rate than in the general setting with smooth objective functions.

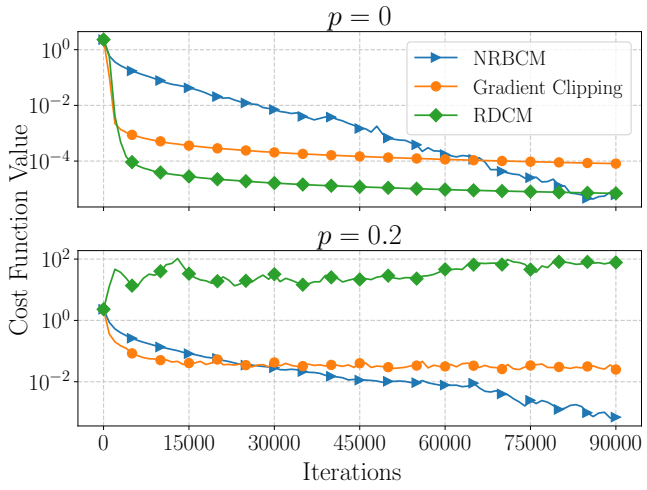

Fig. 1. Convergence performance of NRBCM, RDCM, and gradient clipping in logistic classification task on MNIST dataset for corruption probabilities $p = 0$ (top) and $p = 0.2$ (bottom).

With constant step sizes $\gamma_{i,t} = \gamma/(\phi_i(1-2\bar{p}_i))$, $\forall i \in [q]$, Eq. (31) shows that the sequence $\{\mathbb{E}[f(x_t)] - f^\star\}_{t\geq 0}$ geometrically approaches the interval:

$$\left[0,\ \frac{\gamma}{2\mu}\sum_{i=1}^{q}\frac{L_i^{1/2}}{(1-2\bar{p}_i)}\right] \tag{33}$$

i.e., it finds a solution in the $\mathcal{O}(\gamma/\mu)$-neighborhood of an optimal solution in expectation. Note that for any $\epsilon > 0$, one can select $\gamma = \mathcal{O}(\epsilon\mu(\sum_{i=1}^{q} L_i^{1/2}/(1-2\bar{p}_i))^{-1})$ such that the algorithm finds an ϵ-optimal solution. With a decreasing step size of $\gamma_{i,t} = \mathcal{O}(1/t)$, Eq. (32) shows that in expectation, Algorithm 1 converges to an optimal solution at the rate of $\mathcal{O}(\log(t)/t)$, in terms of the differences in objective value to $f^\star$. Note that a direct application of (11) and (29) would only yield a convergence rate of $\mathcal{O}(1/\sqrt{t})$.

In the next section, we present the numerical study on the robustness of NRBCM.

IV. NUMERICAL STUDY

We study the robustness of Algorithm 1 on multi-class logistic classification using the MNIST dataset [41]. The task is to determine $m = 10$ linear classifiers in order to separate $N = 60000$ $d = 784$-dimensional image vectors. The problem can be stated as:

$$\min_{x\in\mathbb{R}^{m\times d}} -\frac{1}{N}\sum_{i=1}^{N}\log\frac{e^{x_{y_i}A_i^T}}{\sum_{j=1}^{m}e^{x_{y_j}A_j^T}} \tag{34}$$

where $A \in \mathbb{R}^{N\times d}$ is the matrix containing N data vectors in its rows (A_i denotes the i'th row of A) and $y \in \{0,1,\ldots,9\}^N$ is the vector containing the N associated classes. The decision parameter consists of m vectors with dimension d, where each vector corresponds to a class (hence, x_{y_i} corresponds to the y_i'th row of x, where y_i is the class of i'th data vector).

We partitioned the decision vector into $q = 8$ blocks, where $x(i) \in \mathbb{R}^{m\times\frac{d}{q}}$, $\forall i \in [q]$ and set $\phi_i = 1/q$, $\forall i \in [q]$. For comparison, we implemented the random coordinate descent method (RDCM [7]) and a block coordinate version of gradient clipping [19] along with Algorithm 1. We implemented each algorithm with constant step size $\gamma_{i_t,t} = \gamma$, $\forall t, \forall i \in [q]$. We simulated each algorithm for $\gamma = 10^{-4}$, 10^{-5}, and 10^{-6}, and picked the best-performing one. We let $p_i = p$, $\forall i \in [q]$ and set the corrupted gradient as $b_t = -((1-p)/p)\nabla f(x_t)$ at each iteration.

Figure 1 compares the performances of the algorithms for $p = 0$ and $p = 0.2$, using the value of the cost function $f(x_t)$ during training as metric. When $p = 0$, all algorithms succeed in optimizing the cost function value as expected. When $p = 0.2$, RDCM completely fails as it is not robust to corruption. On the other hand, gradient clipping prevents large updates and has satisfactory performance. Nevertheless, gradient clipping performance is limited when the parameter vector gets close to the optimal solution since the trustworthy gradients become smaller while the corrupted gradients can still have norms as big as the clipping threshold. On the other hand, NRBCM always normalizes the updates, and therefore it is robust to corrupted gradients while achieving good performance.

V. CONCLUSIONS

This paper studies the normalized random block coordinate method for non-convex optimization problems with randomly and adversarially perturbed gradients. In this corruption model, the gradient of coordinate blocks is adversarial with some probability. We show the convergence properties of the NRBCM under this corruption model. If the corruption probabilities are less than $1/2$: 1) the NRBCM generates approximate first-order stationary points at a rate $\mathcal{O}(1/\sqrt{T})$, 2) if the function has dominated gradients, a diminishing step-size guarantees convergence to an optimal solution, 3) if the function has dominated gradients, and a constant step-size is used, the NRBCM will converge linearly to an approximate first-order stationary point. Numerical results validate our theoretical findings.

Future work should study the robustness of high-order optimization algorithms, second-order stationary points, escape of saddle points, and decentralized optimization models.

REFERENCES

[1] D. P. Bertsekas, "Nonlinear programming," *Journal of the Operational Research Society*, vol. 48, no. 3, pp. 334–334, 1997.
[2] Z.-Q. Luo and P. Tseng, "Error bounds and convergence analysis of feasible descent methods: a general approach," *Annals of Operations Research*, vol. 46, no. 1, pp. 157–178, 1993.
[3] L. Grippof and M. Sciandrone, "Globally convergent block-coordinate techniques for unconstrained optimization," *Optimization methods and software*, vol. 10, no. 4, pp. 587–637, 1999.
[4] S. J. Wright, "Coordinate descent algorithms," *Mathematical Programming*, vol. 151, no. 1, pp. 3–34, 2015.
[5] A. Saha and A. Tewari, "On the nonasymptotic convergence of cyclic coordinate descent methods," *SIAM Journal on Optimization*, vol. 23, no. 1, pp. 576–601, 2013.
[6] A. Beck and L. Tetruashvili, "On the convergence of block coordinate descent type methods," *SIAM journal on Optimization*, vol. 23, no. 4, pp. 2037–2060, 2013.

[7] Y. Nesterov, "Efficiency of coordinate descent methods on huge-scale optimization problems," *SIAM Journal on Optimization*, vol. 22, no. 2, pp. 341–362, 2012.
[8] P. Richtárik and M. Takáč, "Iteration complexity of randomized block-coordinate descent methods for minimizing a composite function," *Mathematical Programming*, vol. 144, no. 1, pp. 1–38, 2014.
[9] A. Patrascu and I. Necoara, "Efficient random coordinate descent algorithms for large-scale structured nonconvex optimization," *Journal of Global Optimization*, vol. 61, no. 1, pp. 19–46, 2015.
[10] A. W. Yu, L. Huang, Q. Lin, R. Salakhutdinov, and J. Carbonell, "Block-normalized gradient method: An empirical study for training deep neural network," *arXiv preprint arXiv:1707.04822*, 2017.
[11] M. Conti, N. Dragoni, and V. Lesyk, "A survey of man in the middle attacks," *IEEE Communications Surveys & Tutorials*, vol. 18, no. 3, pp. 2027–2051, 2016.
[12] L. Lamport, R. Shostak, and M. Pease, "The byzantine generals problem," in *Concurrency: the Works of Leslie Lamport*, 2019, pp. 203–226.
[13] Y. Nesterov, *Introductory Lectures on Convex Optimization*. Springer Science & Business Media, 2004, vol. 87.
[14] N. Z. Shor, *Minimization methods for non-differentiable functions*. Springer Science & Business Media, 2012, vol. 3.
[15] E. Hazan, K. Levy, and S. Shalev-Shwartz, "Beyond convexity: Stochastic quasi-convex optimization," in *Advances in Neural Information Processing Systems*, 2015, pp. 1594–1602.
[16] Y. You, J. Li, J. Hseu, X. Song, J. Demmel, and C.-J. Hsieh, "Reducing bert pre-training time from 3 days to 76 minutes," *arXiv preprint arXiv:1904.00962*, 2019.
[17] K. Y. Levy, "The power of normalization: Faster evasion of saddle points," *arXiv preprint arXiv:1611.04831*, 2016.
[18] M. Abadi, A. Chu, I. Goodfellow, H. B. McMahan, I. Mironov, K. Talwar, and L. Zhang, "Deep learning with differential privacy," in *Proceedings of the 2016 ACM SIGSAC conference on computer and communications security*, 2016, pp. 308–318.
[19] Z. Sun, P. Kairouz, A. T. Suresh, and H. B. McMahan, "Can you really backdoor federated learning?" *arXiv preprint arXiv:1911.07963*, 2019.
[20] P. Kairouz, H. B. McMahan, B. Avent, A. Bellet, M. Bennis, A. N. Bhagoji, K. Bonawitz, Z. Charles, G. Cormode, R. Cummings *et al.*, "Advances and open problems in federated learning," *arXiv preprint arXiv:1912.04977*, 2019.
[21] L. Huang, A. D. Joseph, B. Nelson, B. I. Rubinstein, and J. D. Tygar, "Adversarial machine learning," in *Proceedings of the 4th ACM workshop on Security and artificial intelligence*, 2011, pp. 43–58.
[22] A. Kurakin, I. Goodfellow, and S. Bengio, "Adversarial machine learning at scale," *arXiv preprint arXiv:1611.01236*, 2016.
[23] Y. Chen, C. Caramanis, and S. Mannor, "Robust sparse regression under adversarial corruption," in *International Conference on Machine Learning*, 2013, pp. 774–782.
[24] S. Mahloujifar, D. I. Diochnos, and M. Mahmoody, "Learning under p-tampering attacks," in *ALT*, 2018, pp. 572–596.
[25] V. Mnih and G. E. Hinton, "Learning to label aerial images from noisy data," in *Proceedings of the 29th International conference on machine learning (ICML-12)*, 2012, pp. 567–574.
[26] D. Alistarh, D. Grubic, J. Z. Li, R. Tomioka, and M. Vojnovic, "Qsgd: communication-efficient sgd via gradient quantization and encoding," in *Proceedings of the 31st International Conference on Neural Information Processing Systems*, 2017, pp. 1707–1718.
[27] A. Reisizadeh, A. Mokhtari, H. Hassani, A. Jadbabaie, and R. Pedarsani, "Fedpaq: A communication-efficient federated learning method with periodic averaging and quantization," in *International Conference on Artificial Intelligence and Statistics*. PMLR, 2020, pp. 2021–2031.
[28] O. Devolder, F. Glineur, and Y. Nesterov, "First-order methods of smooth convex optimization with inexact oracle," *Mathematical Programming*, vol. 146, no. 1, pp. 37–75, 2014.
[29] B. Biggio, B. Nelson, and P. Laskov, "Poisoning attacks against support vector machines," in *Proceedings of the 29th International Coference on International Conference on Machine Learning*, ser. ICML'12. Madison, WI, USA: Omnipress, 2012, p. 1467–1474.
[30] J. Steinhardt, P. W. Koh, and P. Liang, "Certified defenses for data poisoning attacks," in *Proceedings of the 31st International Conference on Neural Information Processing Systems*, ser. NIPS'17. Red Hook, NY, USA: Curran Associates Inc., 2017, p. 3520–3532.
[31] I. Diakonikolas, G. Kamath, D. Kane, J. Li, J. Steinhardt, and A. Stewart, "Sever: A robust meta-algorithm for stochastic optimization," in *International Conference on Machine Learning*, 2019, pp. 1596–1606.
[32] P. Blanchard, E. M. El Mhamdi, R. Guerraoui, and J. Stainer, "Machine learning with adversaries: Byzantine tolerant gradient descent," in *Proceedings of the 31st International Conference on Neural Information Processing Systems*, 2017, pp. 118–128.
[33] B. Turan, C. A. Uribe, H.-T. Wai, and M. Alizadeh, "Resilient primal–dual optimization algorithms for distributed resource allocation," *IEEE Transactions on Control of Network Systems*, vol. 8, no. 1, pp. 282–294, 2020.
[34] Y. Chen, L. Su, and J. Xu, "Distributed statistical machine learning in adversarial settings: Byzantine gradient descent," *Proceedings of the ACM on Measurement and Analysis of Computing Systems*, vol. 1, no. 2, pp. 1–25, 2017.
[35] B. Turan, C. A. Uribe, H.-T. Wai, and M. Alizadeh, "Robust distributed optimization with randomly corrupted gradients," *arXiv preprint arXiv:2106.14956*, 2021.
[36] D. Yin, Y. Chen, R. Kannan, and P. Bartlett, "Byzantine-robust distributed learning: Towards optimal statistical rates," in *International Conference on Machine Learning*. PMLR, 2018, pp. 5650–5659.
[37] B. Turan, C. A. Uribe, H.-T. Wai, and M. Alizadeh, "On robustness of the normalized subgradient method with randomly corrupted subgradients," *arXiv preprint arXiv:2009.13725*, 2020.
[38] D. J. Foster, A. Sekhari, and K. Sridharan, "Uniform convergence of gradients for non-convex learning and optimization," in *Proceedings of the 32nd International Conference on Neural Information Processing Systems*, 2018, pp. 8759–8770.
[39] A. C. Wilson, L. Mackey, and A. Wibisono, "Accelerating rescaled gradient descent: Fast optimization of smooth functions," *Advances in Neural Information Processing Systems*, vol. 32, pp. 13 555–13 565, 2019.
[40] O. Romero and M. Benosman, "Finite-time convergence in continuous-time optimization," in *International Conference on Machine Learning*. PMLR, 2020, pp. 8200–8209.
[41] Y. LeCun, L. Bottou, Y. Bengio, and P. Haffner, "Gradient-based learning applied to document recognition," *Proceedings of the IEEE*, vol. 86, no. 11, pp. 2278–2324, 1998.
[42] H. Robbins and D. Siegmund, "A convergence theorem for non negative almost supermartingales and some applications," in *Optimizing methods in statistics*. Elsevier, 1971, pp. 233–257.

APPENDIX

A. Proof of Theorem 2

1) We first prove the first result when $\gamma_{i,t} = \gamma/(\phi_i(1-2\bar{p}_i))$. We take expectation of both sides in (29) and continue from (21):

$$\mathbb{E}[f(x_t) - f^\star] \leq \frac{1}{\mu}\frac{\mathbb{E}[f(x_t) - f(x_{t+1})]}{\gamma} + \frac{1}{\mu}\sum_{i\in[q]}\frac{\gamma L_i}{2\phi_i(1-2\bar{p}_i)^2}. \tag{35}$$

Multiply both sides by $\gamma\mu$ and rearrange:

$$\mathbb{E}[f(x_{t+1}) - f^\star] \leq \mathbb{E}[f(x_t) - f^\star](1-\gamma\mu) + \frac{\gamma^2}{2}\sum_{i=1}^{q}\frac{L_i}{\phi_i(1-2\bar{p}_i)^2}. \tag{36}$$

Finally, telescopic summation from $t=0$ to $T-1$:

$$\begin{aligned}\mathbb{E}[f(x_T) - f^\star] &\leq (f(x_0) - f^\star)(1-\gamma\mu)^T \\ &+ \frac{\gamma^2}{2}\sum_{i=1}^{q}\frac{L_i}{\phi_i(1-2\bar{p}_i)^2}\sum_{t=0}^{T-1}\prod_{j=t+1}^{T-1}(1-\gamma\mu) \\ &= (f(x_0) - f^\star)(1-\gamma\mu)^T\end{aligned} \tag{37}$$

$$+\frac{\gamma^2}{2}\sum_{i=1}^{q}\frac{L_i}{\phi_i(1-2\bar{p}_i)^2}\sum_{t=0}^{T-1}(1-\gamma\mu)^{T-t-1} \tag{38}$$

$$=(f(x_0)-f^\star)(1-\gamma\mu)^T + \frac{\gamma^2}{2}\sum_{i=1}^{q}\frac{L_i}{\phi_i(1-2\bar{p}_i)^2}\frac{1-(1-\gamma\mu)^T}{\gamma\mu} \tag{39}$$

$$\leq (f(x_0)-f^\star)(1-\gamma\mu)^T + \frac{\gamma}{2\mu}\sum_{i=1}^{q}\frac{L_i}{\phi_i(1-2\bar{p}_i)^2}. \tag{40}$$

Setting ϕ_i given by (30) minimizes the above bound (see (24)-(28)) and gives the desired result in (31).

2) When $\gamma_{i,t}=\frac{1}{\phi_i(1-2\bar{p}_i)\mu(t+1)}$, we continue from (36) and replace γ with $1/(\mu(t+1))$:

$$\mathbb{E}[f(x_{t+1})-f^\star] \leq \mathbb{E}[f(x_t)-f^\star]\left(1-\frac{1}{t+1}\right) + \frac{1}{2\mu^2(t+1)^2}\sum_{i=1}^{q}\frac{L_i}{\phi_i(1-2\bar{p}_i)^2}. \tag{41}$$

Telescopic summation from $t=0$ to $T-1$:

$$\mathbb{E}[f(x_T)-f^\star] \tag{42}$$

$$\leq \frac{1}{2\mu^2}\sum_{i=1}^{q}\frac{L_i}{\phi_i(1-2\bar{p}_i)^2}\sum_{t=0}^{T-1}\frac{1}{(t+1)^2}\prod_{i=t+1}^{T-1}\frac{i}{i+1} \tag{43}$$

$$=\frac{1}{2\mu^2}\sum_{i=1}^{q}\frac{L_i}{\phi_i(1-2\bar{p}_i)^2}\sum_{t=0}^{T-1}\frac{1}{t+1}\frac{1}{T} \tag{44}$$

$$\leq \frac{1}{2\mu^2}\sum_{i=1}^{q}\frac{L_i}{\phi_i(1-2\bar{p}_i)^2}\frac{1+\log T}{T}. \tag{45}$$

Lastly, setting ϕ_i given by (30) minimizes the above bound (see (24)-(28)) and gives the desired result in (11).

To prove almost sure convergence of $f(x_t)$, we use the Robbins-Siegmund Theorem [42] as an auxiliary result:

Theorem 3 (Robbins-Siegmund). *Let $(V_t)_{t\geq 1}$, $(\alpha_t)_{t\geq 1}$, $(\chi_t)_{t\geq 1}$, $(\eta_t)_{t\geq 1}$ be four nonnegative $(\mathcal{F})_{t\geq 1}$-adapted processes such that $\sum_t \alpha_t<\infty$ and $\sum_t \chi_t<\infty$ almost surely. If for each $t\in\mathbb{N}$,*

$$\mathbb{E}\left[V_{t+1}|\mathcal{F}_t\right] \leq V_t(1+\alpha_t)+\chi_t-\eta_t \tag{46}$$

then $(V_t)_{t\geq 1}$ converges almost surely to a random variable V_∞ and $\sum_t \eta_t$ is finite almost surely.

We start with (20), use (29), and replace γ with $1/(\mu(t+1))$:

$$\mathbb{E}[f(x_{t+1})-f^\star|x_t] \leq (f(x_t)-f^\star)\left(1-\frac{1}{t+1}\right) + \frac{1}{2\mu^2(t+1)^2}\sum_{i=1}^{q}\frac{L_i}{\phi_i(1-2\bar{p}_i)^2}. \tag{47}$$

We now apply Theorem 3 with $V_t=f(x_t)-f^\star$, $\beta_t=0$, $\eta_t=(f(x_t)-f^\star)/(t+1)$, and $\chi_t=\frac{1}{2\mu^2(t+1)^2}\sum_{i=1}^{q}\frac{L_i}{\phi_i(1-2\bar{p}_i)^2}$ to conclude that $(f(x_t)-f^\star)$ converges almost surely and $\sum_t (f(x_t)-f^\star)/(t+1)$ is finite almost surely. In order to determine where $(f(x_t)-f^\star)$ converges as well, we use (32) to obtain:

$$\lim_{t\to\infty}\mathbb{E}\left[f(x_t)-f^\star\right]=0. \tag{48}$$

Finally, since $f(x_t)-f^\star\geq 0$, $\lim_{t\to\infty} f(x_t)-f^\star=0$ almost surely and therefore $f(x_t)$ converges to $f^\star$ almost surely.