

# Shadowheart SGD: Distributed Asynchronous SGD with Optimal Time Complexity Under Arbitrary Computation and Communication Heterogeneity



Alexander Tyurin KAUST, AIRI, Skoltech  
 Marta Pozzi KAUST, University of Pavia  
 Ivan Ilin KAUST  
 Peter Richtárik KAUST

## 1. Distributed Stochastic Homogeneous Optimization

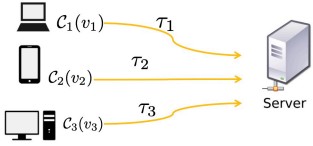
We consider the nonconvex smooth optimization problem

$$\min_{x \in \mathbb{R}^d} \left\{ f(x) := \mathbb{E}_{\xi \sim \mathcal{D}_\xi} [f(x; \xi)] \right\}.$$

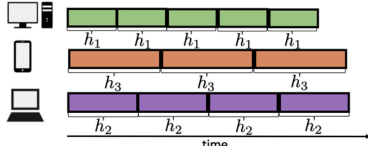
## 2. Setup

- $n$  workers/nodes are able to compute stochastic gradients  $\nabla f(x; \xi)$  of  $f$ , in parallel and asynchronously, and it takes (at most)  $h_i$  seconds for worker  $i$  to compute a single stochastic gradient;
- the workers are connected to a server which acts as a communication hub;
- the workers can communicate with the server in parallel and asynchronously; it takes (at most)  $\tau_i$  seconds for worker  $i$  to send a compressed message to the server; compression is performed via applying lossy communication compression to the communicated message (a vector from  $\mathbb{R}^d$ );

### Communication Takes Time



### Computation Takes Time



## 3. Assumptions

- Assumption 1.1.**  $f$  is differentiable &  $L$ -smooth, i.e.,  $\|\nabla f(x) - \nabla f(y)\| \leq L \|x - y\|, \forall x, y \in \mathbb{R}^d$ .
- Assumption 1.2.** There exist  $f^* \in \mathbb{R}$  such that  $f(x) \geq f^*$  for all  $x \in \mathbb{R}^d$ . We define  $\Delta := f(x^0) - f^*$ , where  $x^0 \in \mathbb{R}^d$  is a starting point of all algorithms we consider.
- Assumption 1.3.** For all  $x \in \mathbb{R}^d$ , the stochastic gradients  $\nabla f(x; \xi)$  are unbiased, and their variance is bounded by  $\sigma^2 \geq 0$ , i.e.,  $\mathbb{E}_\xi[\nabla f(x; \xi)] = \nabla f(x)$  and  $\mathbb{E}_\xi[\|\nabla f(x; \xi) - \nabla f(x)\|^2] \leq \sigma^2$ .

## 4. Goal

Find a (possibly random) vector  $x \in \mathbb{R}^d$  such that

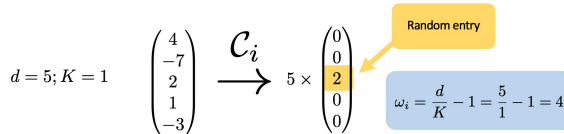
$$\mathbb{E} [\|\nabla f(x)\|^2] \leq \varepsilon$$

## 5. Unbiased Compressors

$$C_i \in \mathcal{U}(w_i) \quad \mathbb{E} [C_i(v)] = v \quad \forall v \in \mathbb{R}^d$$

$$\mathbb{E} [\|C_i(v) - v\|^2] \leq \omega_i \|v\|^2 \quad \forall v \in \mathbb{R}^d$$

**Example: RandK compressor**



## 6. Shadowheart SGD is an Optimal Compressed Method

Table 1: Time Complexities of Centralized Distributed Algorithms. Assume that it takes at most  $h_i$  seconds to worker  $i$  to calculate a stochastic gradient and  $\tau_i$  seconds to send one coordinate/float to server. Abbreviations:  $L$  = smoothness constant,  $\varepsilon$  = error tolerance,  $\Delta = f(x^0) - f^*$ ,  $n$  = # of workers,  $d$  = dimension of the problem. We take the RandK compressor with  $K = 1$  (as an example) in QSGD and Shadowheart SGD.

Method	Time Complexity	Time Complexities in Some Regimes
Minibatch SGD	$\max_{i \in [n]} \max\{h_i, d\tau_i\} \left( \frac{\Delta}{\varepsilon} + \frac{\sigma^2 d \Delta}{n \varepsilon^2} \right)$	$\infty$ (non-robust) $\max\{h_i, d\tau_i\} \frac{\Delta}{\varepsilon}$ (equal performance)
QSGD (Alistarh et al., 2017) (Khaled and Richtárik, 2020)	$\max_{i \in [n]} \max\{h_i, \tau_i\} \left( (d+1) \frac{\Delta}{\varepsilon} + \frac{d \sigma^2 \Delta}{n \varepsilon^2} \right)$	$\infty$ (non-robust) $\geq \frac{d \sigma^2 \Delta}{n \varepsilon^2}$ (worse, e.g., when $\tau_i, d$ or $n$ large)
Rennala SGD (Tyurin and Richtárik, 2023c), Asynchronous SGD (e.g., Mishchenko et al., 2022)	$\geq \min_{i \in [n]} \max\left\{ h_{x_j}, d\tau_{x_j}, \frac{\sigma^2}{\varepsilon} \left( \sum_{i=1}^n \frac{1}{h_i \tau_i} \right)^{-1} \right\} \frac{\Delta}{\varepsilon^{(w)}}$	$< \infty$ (robust) $\geq \max\left\{ h_i, d\tau_i, \frac{h_i \sigma^2}{n \varepsilon} \right\} \frac{\Delta}{\varepsilon}$ (worse, e.g., when $\tau_i, d$ or $n$ large)
Shadowheart SGD (see Algorithm 1) (Corollary 4.3 in the paper)	$t^*(d-1, \sigma^2/\varepsilon, [h_i, \tau_i]) \frac{\Delta}{\varepsilon^{(w)}}$	$< \infty$ (robust) $\max\left\{ h_i, \tau_i, \frac{d \sigma^2}{n \varepsilon}, \sqrt{\frac{d \sigma^2 \Delta}{n \varepsilon^2}}, \frac{h_i \sigma^2}{n \varepsilon} \right\} \frac{\Delta}{\varepsilon}$
Lower Bound (Theorem 0.5 in the paper)	$t^*(d-1, \sigma^2/\varepsilon, [h_i, \tau_i]) \frac{\Delta}{\varepsilon^{(w)}}$	—

The time complexity of Shadowheart SGD is not worse than the time complexity of the competing centralized methods, and is strictly better in many regimes. We show that the complexity of Shadowheart SGDis the optimal time complexity in the family of centralized methods with compression.

<sup>(a)</sup> Upper bound time complexities are not derived for Rennala SGD and Asynchronous SGD. However, we can derive the lower bound using Theorem 0.5 from the paper with  $\omega = 0$ .  
<sup>(b)</sup> The mapping  $t^*$  is defined in Definition 3.1.  
<sup>(c)</sup> The lower bound constructed with the RandK compressor and the dimension  $d = \Theta(L\Delta/\varepsilon)$ .

### Algorithm 1 Shadowheart SGD

- Input: starting point  $x^0 \in \mathbb{R}^d$ , stepsize  $\gamma > 0$ , ratio  $\sigma^2/\varepsilon$ , computation times  $h_i > 0$ , and communication times  $\tau_i > 0$  for  $i \in [n]$
- Find equilibrium time  $t^*$  using Def. 3.1
- Set  $b_i = \lfloor \frac{t^*}{h_i} \rfloor$  and  $m_i = \lfloor \frac{t^*}{\tau_i} \rfloor$  for all  $i \in [n]$
- Find  $S_A = \{i \in [n] : b_i \wedge m_i > 0\}$
- for  $k = 0, 1, \dots, K-1$  do
- Run Alg. 2 in active workers  $S_A$
- Broadcast  $x^k, b_i, m_i$  to active workers  $S_A$
- Initialize  $g^k = 0$
- for  $i \in S_A$  in parallel do
- $w_i = \frac{\omega_i}{\varepsilon} (b_i \omega + \omega \frac{\sigma^2}{\varepsilon} + m_i \frac{\sigma^2}{\varepsilon})^{-1}$
- for  $j = 1, \dots, m_i$  do
- Receive  $C_{ij}(g_i^k)$  from worker  $i$
- $g^k = g^k + w_i C_{ij}(g_i^k)$
- end for
- end for
- $g^k = g^k / (\sum_{i=1}^n w_i m_i b_i)$
- $x^{k+1} = x^k - \gamma g^k$
- end for

### Algorithm 2 Strategy of Worker $i$

- Receive  $x^k, b_i, m_i$  from server, init  $g_i^k = 0$
- for  $l = 1, \dots, b_i$  do
- Calculate  $\nabla f(x^k; \xi_{il}^k)$ ,  $\xi_{il}^k \sim \mathcal{D}_\xi$
- $g_i^k = g_i^k + \nabla f(x^k; \xi_{il}^k)$
- end for
- for  $j = 1, \dots, m_i$  do
- Send  $C_{ij}(g_i^k) \in \mathcal{C}(g_i^k; \nu_{ij}^k)$  to server,  $\nu_{ij}^k \sim \mathcal{D}_\nu, C_{ij} \in \mathcal{U}(w)$
- end for

Shadowheart SGD has the form  $x^{k+1} = x^k - \gamma g^k$ , where

$$g^k = \sum_{i=1}^n w_i \sum_{j=1}^{m_i} C_{ij} \left( \sum_{l=1}^{b_i} \nabla f(x^k; \xi_{il}^k) \right) / \sum_{i=1}^n w_i m_i b_i.$$

### Definition 3.1 (Equilibrium Time).

A mapping  $t^* : \underbrace{\mathbb{R}_{\geq 0} \times \mathbb{R}_{> 0}}_{\omega, \sigma^2/\varepsilon} \times \underbrace{(\mathbb{R}_{\geq 0} \times \mathbb{R}_{> 0})}_{(h_i, \tau_i)} \rightarrow \mathbb{R}_{> 0}$

with inputs  $\omega, \sigma^2/\varepsilon, h_1, \tau_1, \dots, h_n, \tau_n$  is called the *equilibrium time* if it is defined as follows. Find a permutation  $\pi$  that sorts the pairs  $(h_i, \tau_i)$  as  $\max\{h_{\pi_1}, \tau_{\pi_1}\} \leq \dots \leq \max\{h_{\pi_n}, \tau_{\pi_n}\}$  and find the solution  $s^*(j) \in [0, \infty]$  in  $\mathcal{S}^{\text{opt}}$

$$\left( \sum_{i=1}^j \frac{1}{2\tau_{\pi_i} \omega + \frac{4\tau_{\pi_i} h_{\pi_i} \sigma^2 \omega^2 + 2h_{\pi_i} \sigma^2}{\varepsilon}} \right)^{-1} = 8$$

for all  $j \in [n]$ . Then the mapping returns the value  $t^*(\omega, \sigma^2/\varepsilon, h_1, \tau_1, \dots, h_n, \tau_n) \equiv \min_{j \in [n]} \max\{\max\{h_{\pi_j}, \tau_{\pi_j}\}, s^*(j)\} \in [0, \infty]$ .

We shall use the short notation  $t^*(\omega, \sigma^2/\varepsilon, [h_i, \tau_i])$ .

<sup>(a)</sup> It is possible that a permutation is not unique. The result of the mapping does not depend on the choice of the permutation. See the proof of Property 4.1.

<sup>(b)</sup> For convenience, we use the projectively extended real line and define  $1/0 = \infty$ .

(a) : If  $\omega = 0$  and  $\frac{\sigma^2}{\varepsilon} = 0$ , then  $w_i = 1$

## 7. Lower Bound

Admittedly, the definition of the equilibrium time is implicit; we do not know if it is possible to give a more explicit formula in general. We provide a *lower bound* that involves the same mapping. Thus, the equilibrium time is not an ‘‘artifact’’ of our method, but is of a fundamental nature.

## 8. Adaptive Shadowheart SGD

Unfortunately, Shadowheart SGD requires  $\{\tau_i\}$  and  $\{h_i\}$  as an input. One of the main features of asynchronous methods (e.g., Rennala SGD, Asynchronous SGD) is their adaptivity to and independence from processing times. In the paper, we design Adaptive Shadowheart SGD with this feature. However, as a byproduct of this flexibility, this method has a slightly worse time complexity guarantee.

## 9. Bidirectional Shadowheart SGD

### Algorithm 5 Bidirectional Shadowheart SGD

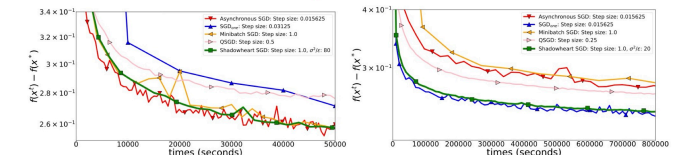
- Input: starting point  $x^0$ , stepsize  $\gamma$ , the ratio  $\sigma^2/\varepsilon$
- for  $k = 0, 1, \dots, K-1$  do
- Find the current computation speeds  $h_i^k > 0$  and communication speeds  $\tau_i^k > 0$  of the workers
- Find the equilibrium time  $t^*$  using Def. 3.1
- Set  $b_i = \lfloor \frac{t^*}{h_i} \rfloor$  and  $m_i = \lfloor \frac{t^*}{\tau_i} \rfloor$  for all  $i \in [n]$
- Find active workers  $S_A = \{i \in [n] : b_i \wedge m_i > 0\}$
- Run Alg. 6 in all workers
- Broadcast  $b_i, m_i$  to all workers
- Init  $g^k = 0$
- for  $i \in S_A$  in parallel do
- $w_i = \frac{\omega_i}{\varepsilon} (b_i \omega + \omega \frac{\sigma^2}{\varepsilon} + m_i \frac{\sigma^2}{\varepsilon})^{-1}$
- for  $j = 1, \dots, m_i$  do
- Receive  $C_{ij}(g_i^k)$  from the  $i^{\text{th}}$  worker
- $g^k = g^k + w_i C_{ij}(g_i^k)$
- end for
- end for
- $g^k = g^k / (\sum_{i=1}^n w_i m_i b_i)$
- $x^{k+1} = x^k - \gamma g^k$
- $p^{k+1} = C_{\text{SERV}}(x^{k+1} - w^k)$
- $w^{k+1} = w^k + p^{k+1}$
- Broadcast  $p^{k+1}$  to all workers
- end for
- If  $\omega = 0$  and  $\frac{\sigma^2}{\varepsilon} = 0$ , then  $w_i = 1$

### Algorithm 6 $i^{\text{th}}$ Worker’s Strategy (init all workers with $w^0 = x^0$ )

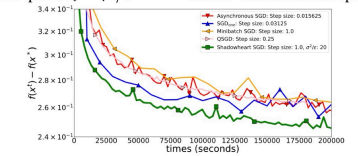
- Receive  $b_i, m_i$  from the server
- If  $b_i \wedge m_i > 0$  then
- Init  $g_i^k = 0$
- for  $l = 1, \dots, b_i$  do
- Calculate  $\nabla f(w^k; \xi_{il}^k)$ ,  $\xi_{il}^k \sim \mathcal{D}_\xi$
- $g_i^k = g_i^k + \nabla f(w^k; \xi_{il}^k)$
- end for
- for  $j = 1, \dots, m_i$  do
- Send  $C_{ij}(g_i^k) \in \mathcal{C}(g_i^k; \nu_{ij}^k)$  to the server,  $\nu_{ij}^k \sim \mathcal{D}_\nu, C_{ij} \in \mathcal{U}(w)$
- end for
- end if
- Receive  $p^{k+1}$  from the server
- $w^{k+1} = w^k + p^{k+1}$

The essential algorithmic changes of Shadowheart SGD are in red. We use compression on the server’s side using the EF21-P mechanism (Grunkowska et al., 2023). Note that  $C_{\text{SERV}}$  is from the biased compressors family (a more general family than unbiased compressors).

## 10. Experiments



(a) Experiment with computation speeds  $h_i = \sqrt{i}$  and high communications speeds  $\tau_i = \sqrt{i}/d$  (b) Experiment with computation speeds  $h_i = \sqrt{i}$  and low communications speeds  $\tau_i = \sqrt{i}/d^{1/2}$



(c) Experiment with computation speeds  $h_i = \sqrt{i}$  and medium communications speeds  $\tau_i = \sqrt{i}/d^{3/4}$

One can see that Shadowheart SGD is very robust to all regimes and has one of the best convergence rates in all experiments. Notably, in the ‘‘medium-speed communications’’ regime, where it is still expensive to send a non-compressed vector, our new method converges faster than other baseline methods.