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


Alexander Tyurin — KAUST, AIRI, Skoltech

Marta Pozzi — KAUST, Pavia

Ivan Ilin — KAUST

Peter Richtárik — KAUST


King Abdullah University of Science and Technology

NEURAL INFORMATION PROCESSING SYSTEMS

## 1. Distributed Stochastic Homogeneous Optimization

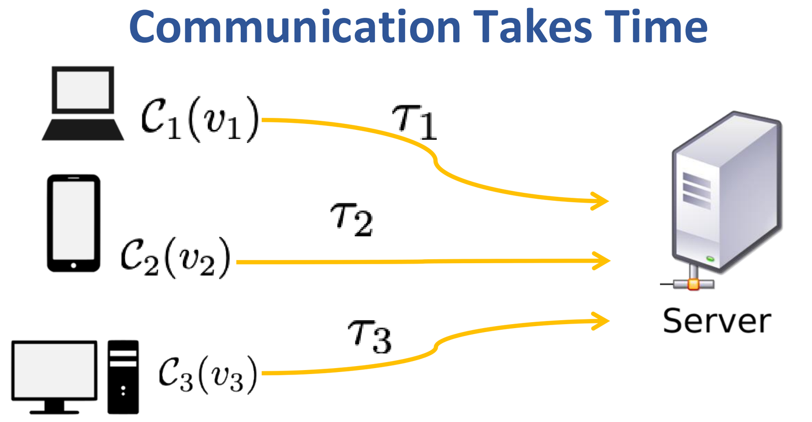
We consider the nonconvex smooth optimization problem

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

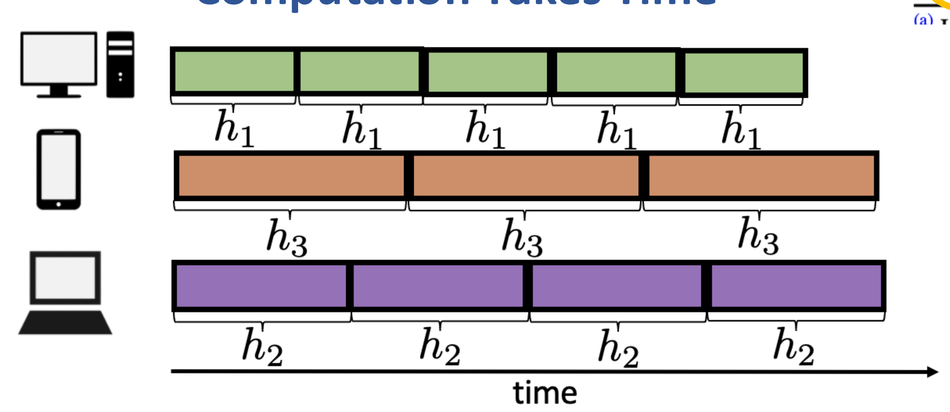
## 2. Setup

- $n$ *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

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

## 5. Unbiased Compressors

$$\mathcal{C}_i \in \mathbb{U}(\omega_i) \qquad \begin{array}{l} \mathbb{E}\left[\mathcal{C}_i(v)\right] = v \quad \forall v \in \mathbb{R}^d \\ \mathbb{E}\left[\|\mathcal{C}_i(v) - v\|^2\right] \leq \omega_i \|v\|^2 \quad \forall v \in \mathbb{R}^d \end{array}$$

**Example: *RandK* compressor**

$$d = 5; K = 1 \qquad \begin{pmatrix} 4 \\ -7 \\ 2 \\ 1 \\ -3 \end{pmatrix} \xrightarrow{\mathcal{C}_i} 5 \times \begin{pmatrix} 0 \\ 0 \\ 2 \\ 0 \\ 0 \end{pmatrix}$$

Random entry

$$\omega_i = \frac{d}{K} - 1 = \frac{5}{1} - 1 = 4$$

## 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 $\dot{\tau}_i$ 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 Rand$K$ compressor with $K = 1$ (as an example) in QSGD and Shadowheart SGD.

| Method | Time Complexity | Time Complexities in Some Regimes: $\max\{h_n, \dot{\tau}_n\} \to \infty$, $\max\{h_i, \dot{\tau}_i\} < \infty \, \forall i < n$ (the last worker is slow) | Time Complexities in Some Regimes: $h_i = h, \dot{\tau}_i = \dot{\tau} \, \forall i \in [n]$ (equal performance) |
|---|---|---|---|
| Minibatch SGD | $\max_{i \in [n]} \max\{h_i, d\dot{\tau}_i\} \left(\frac{L\Delta}{\varepsilon} + \frac{\sigma^2 L\Delta}{n\varepsilon^2}\right)$ | $\infty$ (non-robust) | $\max\{h, d\dot{\tau}, \frac{d\dot{\tau}\sigma^2}{n\varepsilon}, \frac{h\sigma^2}{n\varepsilon}\} \frac{L\Delta}{\varepsilon}$ (worse, e.g., when $\dot{\tau}$, $d$ or $n$ large) |
| QSGD (Alistarh et al., 2017) (Khaled and Richtárik, 2020) | $\max_{i \in [n]} \max\{h_i, \dot{\tau}_i\} \left(\left(\frac{d}{n} + 1\right)\frac{L\Delta}{\varepsilon} + \frac{d\sigma^2 L\Delta}{n\varepsilon^2}\right)$ | $\infty$ (non-robust) | $\geq \frac{dh\sigma^2}{n\varepsilon} \frac{L\Delta}{\varepsilon}$ (worse, e.g., when $\varepsilon$ small) |
| Rennala SGD (Tyurin and Richtárik, 2023c), Asynchronous SGD (e.g., (Mishchenko et al., 2022)) | $\geq \min_{j \in [n]} \max\left\{h_{\bar{\pi}_j}, d\dot{\tau}_{\bar{\pi}_j}, \frac{\sigma^2}{\varepsilon}\left(\sum_{i=1}^{j} \frac{1}{h_{\bar{\pi}_i}}\right)^{-1}\right\} \frac{L\Delta}{\varepsilon}$ [(a)] | $< \infty$ (robust) | $\geq \max\left\{h, d\dot{\tau}, \frac{h\sigma^2}{n\varepsilon}\right\} \frac{L\Delta}{\varepsilon}$ (worse, e.g., when $\dot{\tau}$, $d$ or $n$ large) |
| Shadowheart SGD (see Algorithm 1) (Corollary 4.3 in the paper) | $\boxed{t^*(d-1, \sigma^2/\varepsilon, [h_i, \dot{\tau}_i]_1^n) \frac{L\Delta}{\varepsilon}}$ [(c)] | $< \infty$ (robust) | $\max\left\{h, \dot{\tau}, \frac{d\dot{\tau}}{n}, \sqrt{\frac{d\dot{\tau}h\sigma^2}{n\varepsilon}}, \frac{h\sigma^2}{n\varepsilon}\right\} \frac{L\Delta}{\varepsilon}$ |
| Lower Bound (Theorem O.5 in the paper) | $\boxed{t^*(d-1, \sigma^2/\varepsilon, [h_i, \dot{\tau}_i]_1^n) \frac{L\Delta}{\varepsilon}}$ [(d)] | — | — |

**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.**

[(a)] Upper bound time complexities are not derived for Rennala SGD and Asynchronous SGD. However, we can derive the lower bound using Theorem O.5 from the paper with $\omega = 0$.
The mapping $t^*$ is defined in Definition 3.1.
The lower bound constructed with the Rand$K$ compressor and the dimension $d = \Theta(L\Delta/\varepsilon)$.

**Algorithm 1** Shadowheart SGD

```
Input: starting point x^0 ∈ R^d, stepsize γ > 0, ratio σ²/ε, computation times h_i > 0, and communication times τ_i > 0 for i ∈ [n]
Find equilibrium time t* using Def. 3.1
Set b_i = ⌊t*/h_i⌋ and m_i = ⌊t*/τ_i⌋ for all i ∈ [n]
Find S_A = {i ∈ [n] : b_i ∧ m_i > 0}
for k = 0, 1, ..., 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 ∈ S_A in parallel do
        w_i =(a) (b_i ω + ω σ²/ε + m_i σ²/ε)^(−1)
        for j = 1, ..., 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 / (Σ_{i=1}^n w_i m_i b_i)
    x^{k+1} = x^k − γ 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, ..., b_i do
    Calculate ∇f(x^k; ξ_il^k),   ξ_il^k ~ D_ξ
    g_i^k = g_i^k + ∇f(x^k; ξ_il^k)
end for
for j = 1, ..., m_i do
    Send C_ij(g_i^k) ≡ C(g_i^k; ν_ij^k) to server,
    ν_ij^k ~ D_ν, C_ij ∈ U(ω)
end for
```

**Definition 3.1** (Equilibrium Time).
A mapping $t^* : \underbrace{\mathbb{R}_{\geq 0}}_{\omega} \times \underbrace{\mathbb{R}_{\geq 0}}_{\sigma^2/\varepsilon} \times \underbrace{(\mathbb{R}_{\geq 0} \times \mathbb{R}_{\geq 0})}_{(h_1, \tau_1)} \times \cdots \times \underbrace{(\mathbb{R}_{\geq 0} \times \mathbb{R}_{\geq 0})}_{(h_n, \tau_n)} \to \mathbb{R}_{\geq 0}$

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

$$\left(\sum_{i=1}^{j} \frac{1}{2\tau_{\pi_i}\omega + \frac{4\tau_{\pi_i} h_{\pi_i} \sigma^2 \omega}{s \times \varepsilon} + \frac{2h_{\pi_i}\sigma^2}{\varepsilon}}\right)^{-1} = s \tag{6}$$

for all $j \in [n]$. Then the mapping returns the value

$$t^*(\omega, \sigma^2/\varepsilon, h_1, \tau_1, \ldots, h_n, \tau_n) \equiv \min_{j \in [n]} \max\{\max\{h_{\pi_j}, \tau_{\pi_j}\}, s^*(j)\} \in [0, \infty]. \tag{7}$$

We shall use the short notation $t^*(\omega, \sigma^2/\varepsilon, [h_i, \tau_i]_1^n)$.

[a] 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.

[b] 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 $w_i = 1$

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

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

## 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 γ, the ratio σ²/ε
for k = 0, 1, ..., K − 1 do
    Find the current computation speeds h_i^k > 0 and communication speeds τ_i^k > 0 of the workers
    Find the equilibrium time t* using Def. 3.1
    Set b_i = ⌊t*/h_i^k⌋ and m_i = ⌊t*/τ_i^k⌋ for all i ∈ [n]
    Find active workers S_A = {i ∈ [n] : b_i ∧ m_i > 0}
    Run Alg. 6 in all workers
    Broadcast b_i, and m_i to all workers
    Init g^k = 0
    for i ∈ S_A in parallel do
        w_i =(a) (b_i ω + ω σ²/ε + m_i σ²/ε)^(−1)
        for j = 1, ..., m_i do
            Receive C_ij(g_i^k) from the i^th worker
            g^k = g^k + w_i C_ij(g_i^k)
        end for
    end for
    g^k = g^k / (Σ_{i=1}^n w_i m_i b_i)
    x^{k+1} = x^k − γ g^k
    p^{k+1} = C_serv(x^{k+1} − w^k)
    w^{k+1} = w^k + p^{k+1}
    Broadcast p^{k+1} to all workers
end for
```

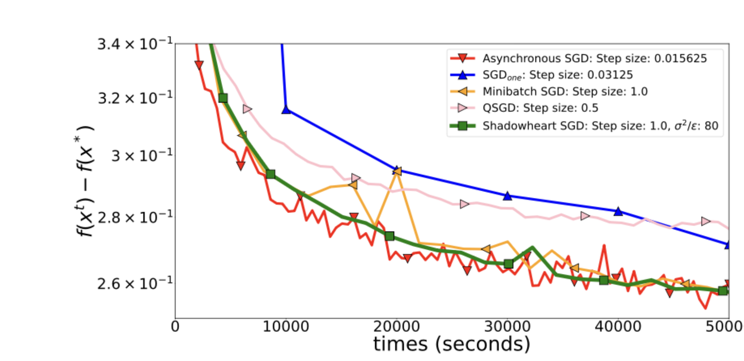
$(a)$ : If $\omega = 0$ and $\frac{\sigma^2}{\varepsilon} = 0$, then $w_i = 1$

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

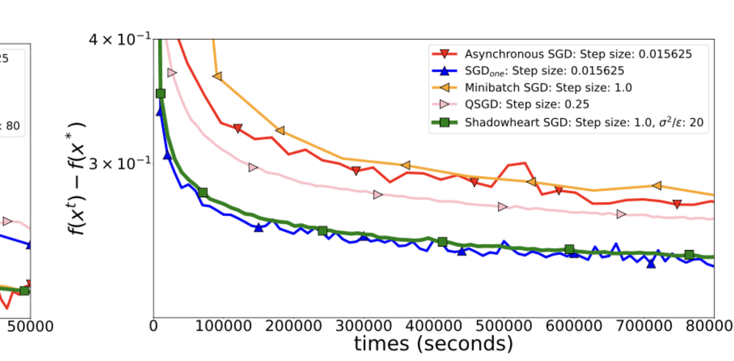
```
Receive b_i, and m_i from the server
if b_i ∧ m_i > 0 then
    Init g_i^k = 0
    for l = 1, ..., b_i do
        Calculate ∇f(w^k; ξ_il^k),   ξ_il^k ~ D_ξ
        g_i^k = g_i^k + ∇f(w^k; ξ_il^k)
    end for
    for j = 1, ..., m_i do
        Send C_ij(g_i^k) ≡ C(g_i^k; ν_ij^k) to the server,
        ν_ij^k ~ D_ν, C_ij ∈ U(ω)
    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 (Gruntkowska et al., 2023). Note that $\mathcal{C}_{\text{serv}}$ is from the biased compressors family (a more general family than unbiased compressors).

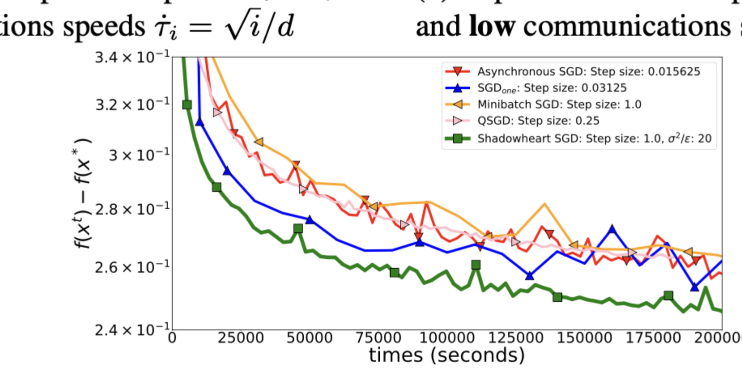
## 10. Experiments




(a) Experiment with computation speeds $h_i = \sqrt{i}$ and **high** communications speeds $\dot{\tau}_i = \sqrt{i}/d$




(b) Experiment with computation speeds $h_i = \sqrt{i}$ and **low** communications speeds $\dot{\tau}_i = \sqrt{i}/d^{1/2}$




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

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.