DISTRIBUTED OPTIMIZATION WITH UNBOUNDED DELAYS

Panayotis Mertikopoulos[1]

joint with

Zhengyuan Zhou[2] Nick Bambos[3] Peter Glynn[3] Yinyu Ye[3]

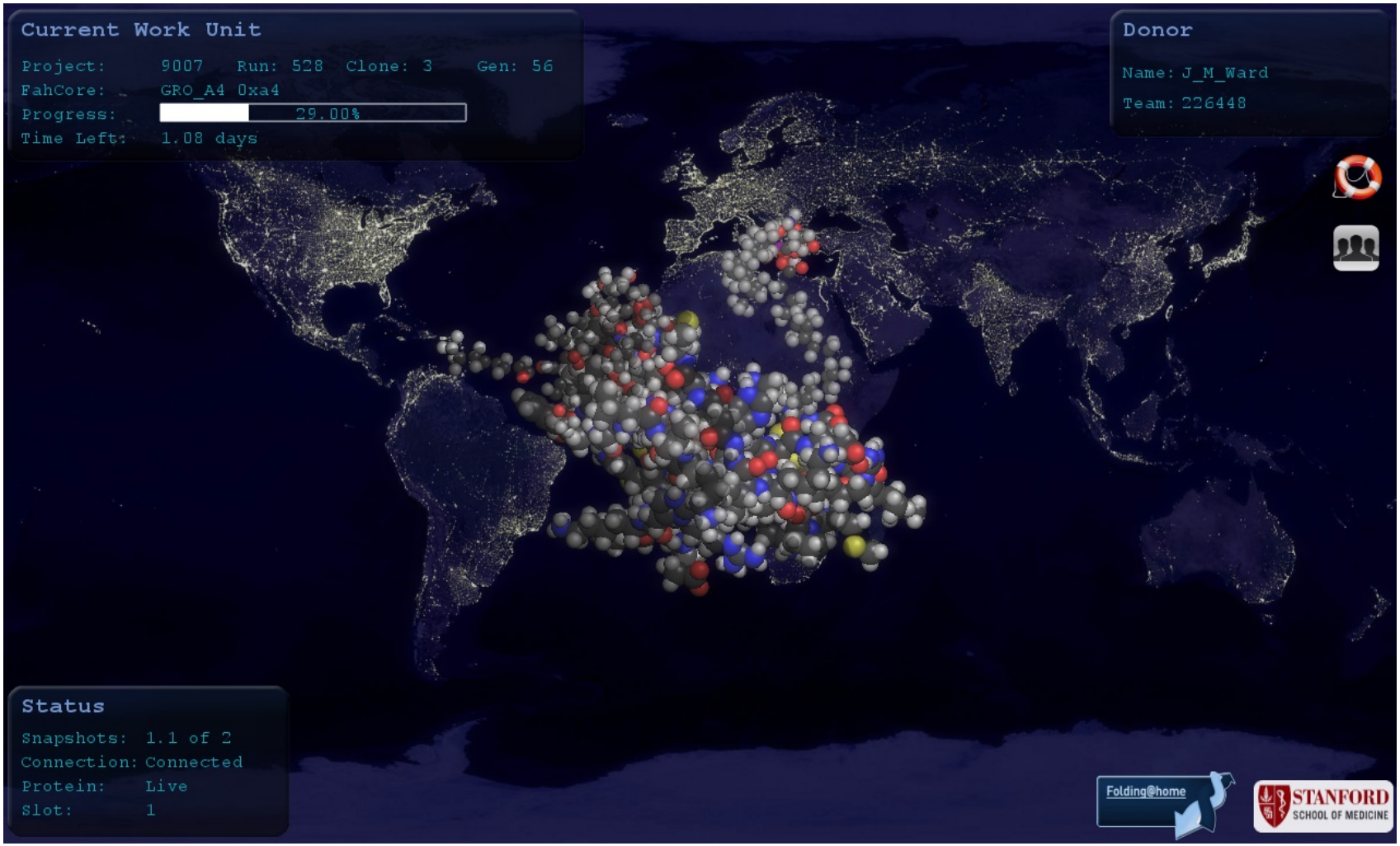
[1]CNRS / LIG / POLARIS

[2]NYU Stern

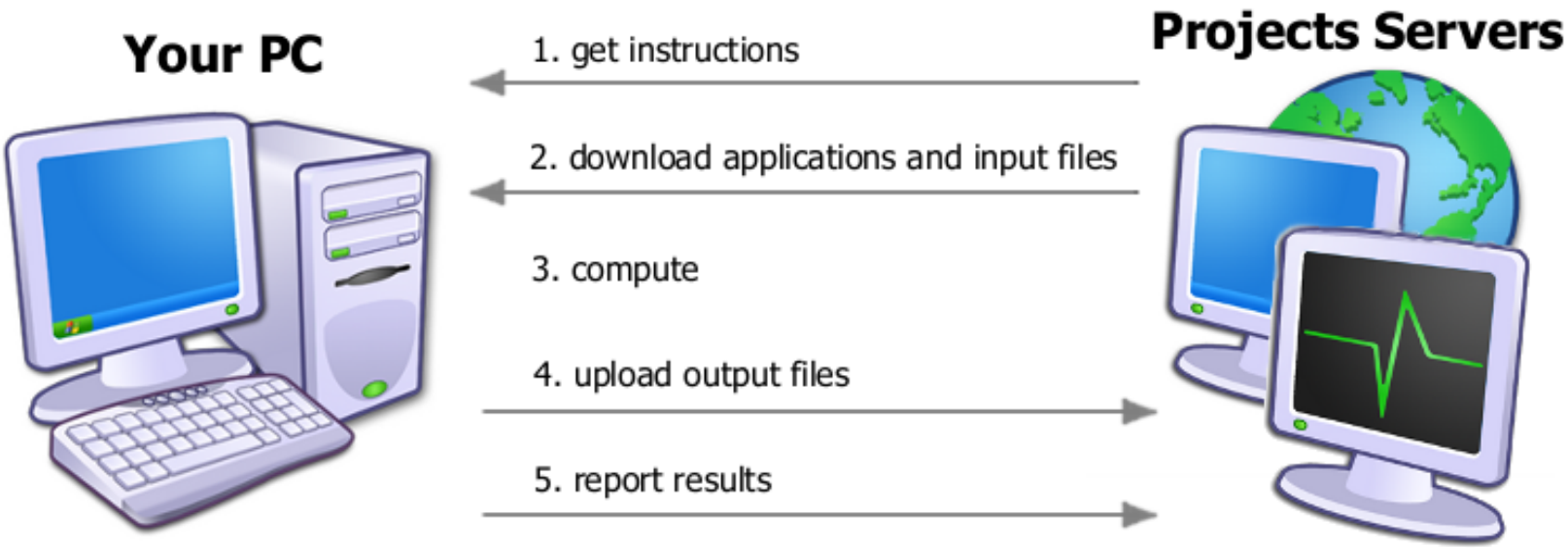
[3]Stanford University

WAX workshop – May 27, 2021

Distributed computing



Distributed computing, cont'd



Stochastic/Distributed optimization

Stochastic optimization:

$$\begin{aligned} \text{minimize} \quad & f(x) = \mathbb{E}[F(x;\omega)] \\ \text{subject to} \quad & x \in \mathcal{X} \end{aligned} \tag{Opt}$$

where:

- $\mathcal{X}$ is a closed, convex subset of $\mathbb{R}^d$
- $F(x;\omega)$ is a random function (depending on ω)

Stochastic/Distributed optimization

Stochastic optimization:

$$\begin{aligned} \text{minimize} \quad & f(x) = \mathbb{E}[F(x;\omega)] \\ \text{subject to} \quad & x \in \mathcal{X} \end{aligned} \tag{Opt}$$

where:

- $\mathcal{X}$ is a closed, convex subset of $\mathbb{R}^d$
- $F(x;\omega)$ is a random function (depending on ω)

Distributed optimization: $f(x) = \frac{1}{N}\sum_{i=1}^{N} f_i(x)$ $\quad [F(x;\omega_i) = f_i(x)]$

Stochastic/Distributed optimization

Stochastic optimization:

$$\begin{aligned} \text{minimize} \quad & f(x) = \mathbb{E}[F(x;\omega)] \\ \text{subject to} \quad & x \in \mathcal{X} \end{aligned} \tag{Opt}$$

where:

- $\mathcal{X}$ is a closed, convex subset of $\mathbb{R}^d$
- $F(x;\omega)$ is a random function (depending on ω)

Distributed optimization: $f(x) = \frac{1}{N}\sum_{i=1}^{N} f_i(x)$ $\quad [F(x;\omega_i) = f_i(x)]$

Key challenges:

- Computation of expectation/sum impossible
- Gradients of f inaccessible but gradients of F cheap

The master-worker architecture

Master-Worker architecture

Require: Master and workers $i = 1, \ldots, N$

1: **repeat**

2: **Master**:

(a) Receive gradient update from worker i

(b) Update state

(c) Send updated state to worker i

3: **Workers**:

(a) Receive state

(b) Compute i.i.d. gradient

(c) Send gradient update to master

4: **until** end

Stochastic gradient descent

Distributed asynchronous stochastic gradient descent (DASGD)

Require: Initial state $Y_0 \in \mathbb{R}^d$, step-size sequence γ_n

1: $n \leftarrow 0$
2: **repeat**
3: $\quad X_n = \Pi(Y_n)$
4: $\quad Y_{n+1} = Y_n - \gamma_{n+1} \nabla F(X_{s_n}; \omega_{s_n+1})$
5: $\quad n \leftarrow n + 1$
6: **until** end
7: **return** solution candidate X_n

Glossary:

- n: global counter
- s_n: iteration from which gradient of round n originates
- $d_{s_n} = n - s_n$: delay of iteration s_n (could become huge over time)
- X_n: global state
- Π: projection to $\mathcal{X}$

State of the art

What is known:

- **Sublinear delays, convex, deterministic:** convergence if $d_n = o(n)$

  [Bertsekas & Tsitsiklis, 1997]

- **Bounded delays, convex, stochastic:** mean convergence if $\sup_n d_n < \infty$

  [Agarwal & Duchi, 2011; Recht et al., 2011; Mania et al., 2017;…]

- **Bounded delays, nonconvex, stochastic:** mean gradient gap if $\sup_n d_n < \infty$

  [Lian et al., 2015]

Limitations:

- High probability / almost sure convergence?
- Large delays?
- Convergence in nonconvex objectives?

Assumptions

Assumption (Regularity)

1. Differentiability: $F(x;\omega)$ *is differentiable in* x *for* $\mathbb{P}$*-almost all* ω
2. Bounded second moments: $\mathbb{E}[\|\nabla F(x;\omega)\|^2] < \infty$
3. Lipschitz smoothness: $\nabla f(x) = \mathbb{E}[\nabla F(x;\omega)]$ *is Lipschitz on* $\mathcal{X}$

Assumption (Variational Coherence)

F *is variationally coherent in the mean:*

$$\mathbb{E}[\langle \nabla f(x;\omega), x - x^* \rangle] > 0 \qquad \text{(VC)}$$

for all $x^* \in \arg\min f$ *and all* $x \notin \arg\min f$.

Our results

The method's step-size plays a crucial role:

1. Bounded delays: $\sup_n d_n < \infty \quad \rightsquigarrow \quad \sum_{n=1}^{\infty} \gamma_n^2 < \infty, \sum_{n=1}^{\infty} \gamma_n = \infty$
2. Linearly growing delays: $d_n = \mathcal{O}(n) \quad \rightsquigarrow \quad \gamma_n \propto 1/(n \log n)$
3. Polynomially growing delays: $d_n = \mathcal{O}(n^q) \quad \rightsquigarrow \quad \gamma_n \propto 1/(n \log n \log\log n)$

Our results

The method's step-size plays a crucial role:

1. Bounded delays: $\sup_n d_n < \infty \quad \rightsquigarrow \quad \sum_{n=1}^{\infty} \gamma_n^2 < \infty, \sum_{n=1}^{\infty} \gamma_n = \infty$
2. Linearly growing delays: $d_n = \mathcal{O}(n) \quad \rightsquigarrow \quad \gamma_n \propto 1/(n \log n)$
3. Polynomially growing delays: $d_n = \mathcal{O}(n^q) \quad \rightsquigarrow \quad \gamma_n \propto 1/(n \log n \log\log n)$

Theorem (Zhou, M, Bambos, Glynn & Ye, ICML 2018; coherent)

- Assume: **A1** (*regularity*); **A2** (*coherence*)
- Then: *DASGD converges to a solution of* (Opt) *with probability* 1

Our results

The method's step-size plays a crucial role:

1. Bounded delays: $\sup_n d_n < \infty \quad \rightsquigarrow \quad \sum_{n=1}^{\infty} \gamma_n^2 < \infty, \sum_{n=1}^{\infty} \gamma_n = \infty$
2. Linearly growing delays: $d_n = \mathcal{O}(n) \quad \rightsquigarrow \quad \gamma_n \propto 1/(n \log n)$
3. Polynomially growing delays: $d_n = \mathcal{O}(n^q) \quad \rightsquigarrow \quad \gamma_n \propto 1/(n \log n \log\log n)$

Theorem (Zhou, M, Bambos, Glynn & Ye, ICML 2018; coherent)

- Assume: **A1** (*regularity*); **A2** (*coherence*)
- Then: *DASGD converges to a solution of* (Opt) *with probability* 1

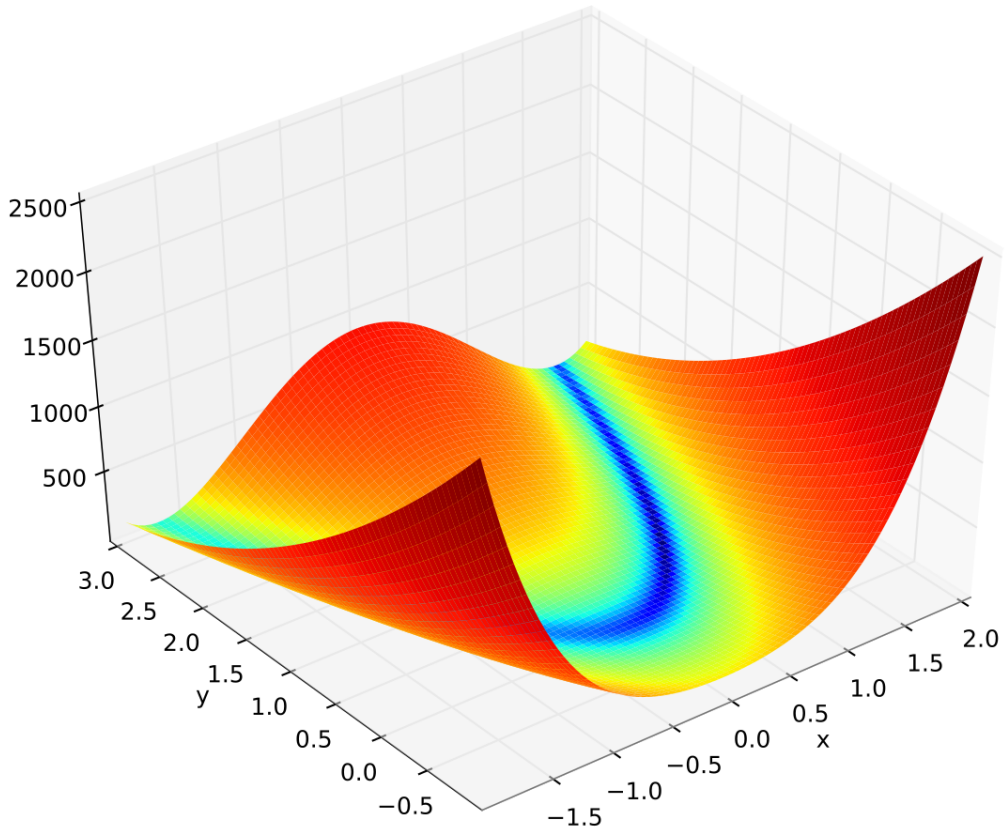
Theorem (Zhou, M, Bambos, Glynn & Ye, MOR 2021; non-convex)

- Assume: **A1** (*regularity*)
- Then: *DASGD converges to a solution of* (Opt) *with probability* 1

Numerical experiments

Test: Rosenbrock benchmark with $d = 101$ degrees of freedom

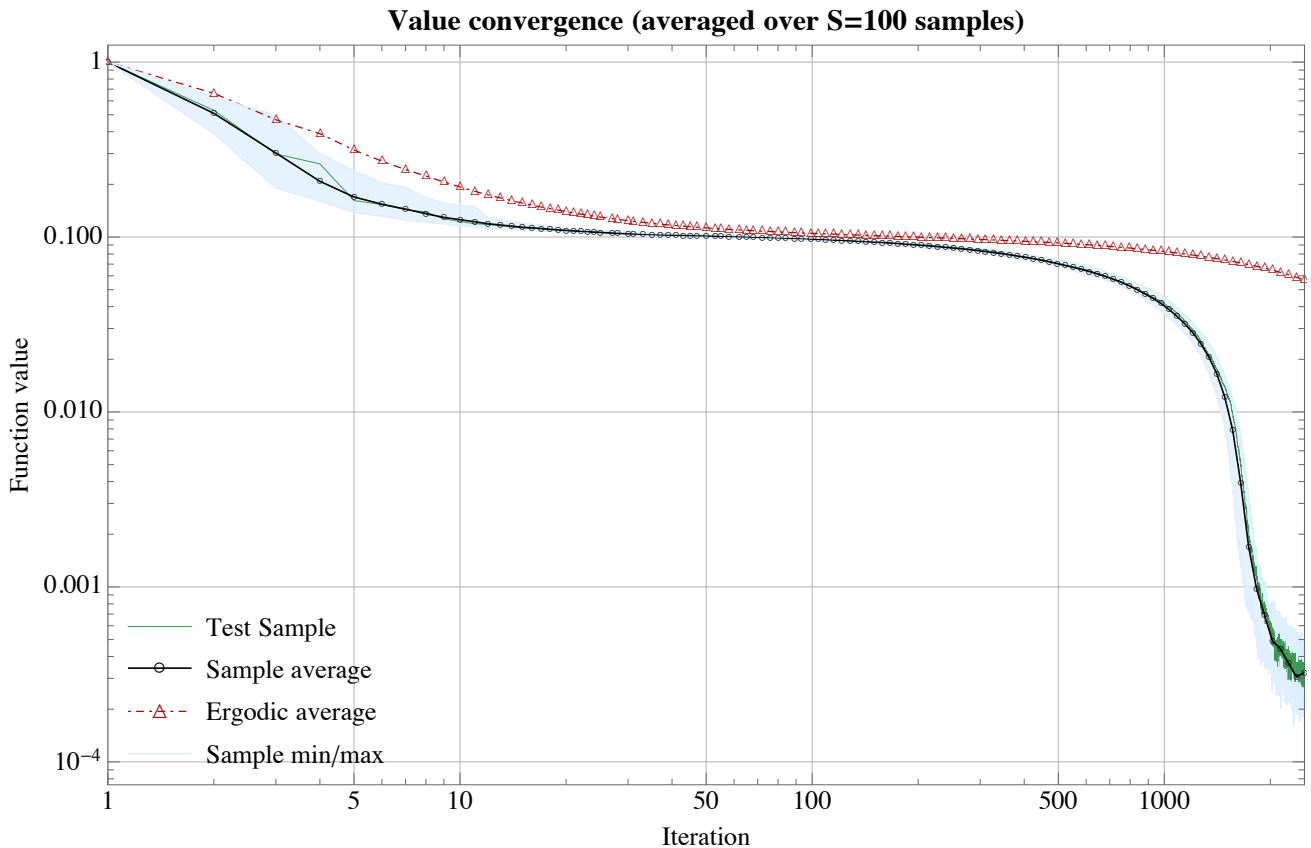
$$f(x) = \sum_{i=1}^{100} [100(x_{i+1} - x_i^2)^2 + (1 - x_i)^2],$$



Numerical experiments

Test: Rosenbrock benchmark with $d = 101$ degrees of freedom

$$f(x) = \sum_{i=1}^{100} [100(x_{i+1} - x_i^2)^2 + (1 - x_i)^2],$$



Numerical experiments

Test: Rosenbrock benchmark with $d = 101$ degrees of freedom

$$f(x) = \sum_{i=1}^{100} [100(x_{i+1} - x_i^2)^2 + (1 - x_i)^2],$$