Robust Projection-Based Distributed Gradient-Descent Algorithm: A Fully-Distributed (Re-)design

LabEx EnergyAlps: Kick-off meeting

Mohamed Maghenem *in Collaboration with T. Bazizi and P. Frasca*

17/11/2025

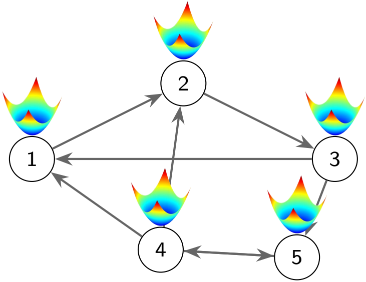
Problems Formulation

The problem $\min_{x \in \mathbb{R}^n} \left(F(x) := \sum_{i=1}^{N} f_i(x) \right)$ can be recast to

$$\min_{x_1,\ldots,x_N \in \mathbb{R}^n} \sum_{i=1}^{N} f_i(x_i) \text{ s.t. } x_1 = \ldots = x_N.$$

In a distributed framework :

- Each $f_i : \mathbb{R}^n \to \mathbb{R}$ is assigned to a computation unit.
- Every unit seeks a minimum of f_i within the set of minimizers X_i.
- The exchange is key so that the minimum is consensual.



Challenge : Dynamically design every unite minimizer x_i s.t.

$$\lim_{t \to +\infty} x_1(t) = \ldots = \lim_{t \to +\infty} x_N(t) \in \{X_1 \cap \ldots \cap X_N\}.$$

Distributed/Federated Learning

Consider the optimization problem

$$\min_{\theta_1,\ldots,\theta_N\in\mathbb{R}^n} \sum_{i=1}^{N} \ell_i(\theta_i,(u_1^i,y_1^i),\ldots,(u_{m_i}^i,y_{m_i}^i)) \text{ s.t. } \theta_1=\ldots=\theta_N.$$

Here, every unit i has access to the local data-set $\{(u_k^i,y_k^i)\}_{k=1}^{m_i}$. Hence, following a distributed-optimization framework, every unit i

- minimizes its local cost function ℓ_i using $\{(u_k^i,y_k^i)\}_{k=1}^{m_i}$.
- shares only θ_i, the local estimate of a global solution, with some other neighboring agents.

When the exchange happens through **servers**, we talk about **federated learning**. Otherwise, it is **distributed learning**.

Some Literature Review

Here is a candidate distributed-optimization algorithm :

$$\dot{x}_i = \underbrace{-\nabla f_i(x_i)}_{\text{Optimality}} + \gamma_i \underbrace{\sum_{j=1}^{N} a_{ij}(x_j - x_i)}_{\text{consensus}}, \quad \gamma_i > 0, \ \forall i \in \{1, 2, \ldots, N\},$$

∇f_i is the gradient of f_i, and $a_{ij} \geq 0$ represents the communication weight between agent i and j.

- Usually the descent rate γ_i is a predefined time-varying signal **[P. Lin, et al, IEEE TAC, 2025]** and **[Nedic, et al, IEEE TAC, 2009]**.

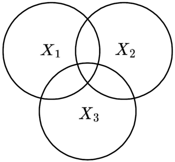
- Alternative algorithms, when the γ_is are constant, include :

- A PI control strategy : **[Yang, et al, Automatica, 2018]**.
- Second-order algorithm : **[Lu and Tang, IEEE TAC, 2012]**.

Projection-Based Formulation

Assume that :

- Each local cost function f_i has a convex set X_i of minimizers.
- The intersection of X_is is nonempty.



Hence, a solution to $\min_{x \in \mathbb{R}^n} \sum_{i=1}^{N} f_i(x)$ must belong to $\cap_{i=1}^{N} X_i$.
Then, the following equivalence holds

$$\min_{x_1,\ldots,x_N \in \mathbb{R}^n} \sum_{i=1}^{N} f_i(x_i) \quad \equiv \quad \min_{x_1,\ldots,x_N \in \mathbb{R}^n} \frac{1}{2} \sum_{i=1}^{N} |x_i - \Pi_{X_i}(x_i)|^2,$$

$$\text{subject to} \quad x_1 = \ldots = x_N \qquad\qquad \text{subject to} \quad x_1 = \ldots = x_N.$$

$\Pi_{X_i}(x_i) := \operatorname{argmin}_{y \in X_i} |x_i - y|$ is the projection of x_i on X_i.

Projection-Based Gradient Algorithm

The new local cost functions are $f_i(\cdot) := |\cdot|^2_{X_i}/2$, verifying

$$\nabla|x_i - \Pi_{X_i}(x_i)|^2 = 2(x_i - \Pi_{X_i}(x_i)).$$

Then, the gradient-descent algorithm

$$\dot{x}_i = -\nabla f_i(x_i) + \gamma_i \sum_{j=1}^{N} a_{ij}(x_j - x_i) \qquad i \in \{1, 2, ..., N\}$$

is reformulated as a projection-based gradient-descent algorithm

$$\dot{x}_i = -[x_i - \tilde{\Pi}_{X_i}(x_i)] + \gamma_i \sum_{j=1}^{N} a_{ij}(x_j - x_i) \qquad i \in \{1, 2, ..., N\},$$

where $\tilde{\Pi}_{X_i}(x_i)$ is a corrupted projection of x_i onto X_i.

Projection Computation

We model the corrupted projection as

$$\tilde{\Pi}_{X_i}(x_i) := \Pi_{X_i}(x_i) + \zeta_i p_i,$$

- The coefficient $\zeta_i \in (0,1)$ represents a tunable precision.
- $p_i \in \mathbb{R}^n$ stands for the worst-case projection error.
- The algorithm in **[Usmanova, et al, ICML, 21]** provides such a projection $\tilde{\Pi}_{X_i}(x_i)$ within $O(\log(1/\zeta_i))$ computation steps.

As a result, we recover the class of perturbed multi-agent systems

$$\dot{x}_i = \gamma_i \sum_{j=1}^{N} a_{ij}(x_j - x_i) - (x_i - \Pi_{X_i}(x_i)) + \zeta_i p_i \qquad i \in \{1, ..., N\}.$$

Handling Bounded Perturbations

$$\dot{x}_i = \gamma_i \sum_{j=1}^{N} a_{ij}(x_j - x_i) - (x_i - \Pi_{X_i}(x_i)) + \zeta_i p_i, \quad i \in \{1, ..., N\}.$$

- When $p_i \equiv 0$ the problem is solved in **[Shi, et al, IEEE TAC, 13]**.
- When $p_i \not\equiv 0$, the problem is solved using predefined signals (γ_i, ζ_i) while assuming vanishing p_i **[Lou, et al, IEEE TAC, 16]**.

Under unknown bounded perturbations, guaranteeing

$$\lim_{t \to +\infty} x_1(t) = ... = \lim_{t \to +\infty} x_N(t) \in X_1 \cap ... \cap X_N$$

is impossible. However, we can get arbitrarily close !

Adaptive Framework

$$\dot{x}_i = \gamma_i \sum_{j=1}^{N} a_{ij}(x_j - x_i) - (x_i - \Pi_{X_i}(x_i)) + \zeta_i p_i, \quad i \in \{1, ..., N\}.$$

We design $\{(\gamma_i, \zeta_i)\}_{i=1}^{N}$ as

$$\dot{\gamma}_i = h_i(x_i, \gamma_i, \{(x_j, \gamma_j)\}_{j \in \mathcal{N}_i}, \varepsilon), \quad \dot{\zeta}_i = g_i(x_i, \zeta_i, \{(x_j, \zeta_j)\}_{j \in \mathcal{N}_i}, \varepsilon).$$

Result : Given $\varepsilon > 0$, we design $\{(h_i, g_i)\}_{i=1}^{N}$ such that

$$\lim_{t \to +\infty} |x_i(t) - x_j(t)| \leq \varepsilon \qquad \forall i, j \in \{1, \ldots, N\},$$
$$\lim_{t \to +\infty} |x_i(t)|_{X_i} \leq \varepsilon \qquad \forall i \in \{1, \ldots, N\}.$$

At the same time, we guarantee that

$$|\gamma_i|_\infty < \infty, \ |\zeta_i|_\infty < 1, \ \inf_{t \geq 0} \zeta_i(t) > 0 \qquad \forall i \in \{1, \ldots, N\}.$$

Adaptive Design

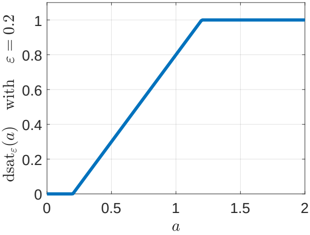
Inspired by **[Chen, et al, IEEE TAC, 2024]**, we let

$$\dot{\gamma}_i = \sum_{j=1}^{N} a_{ij}(\gamma_j - \gamma_i) + \frac{1}{2}\mathrm{dsat}_\varepsilon\left(|x_i|_{X_i}\right) + \frac{1}{2}\mathrm{dsat}_\varepsilon\left(\sum_{j\in\mathcal{N}_i} |x_j - x_i|\right),$$

$$\dot{\zeta}_i = -\sum_{j=1}^{N} a_{ij}(\zeta_i^2/\zeta_j - \zeta_i) + \frac{\zeta_i}{2}\mathrm{dsat}_\varepsilon\left(|x_i|_{X_i}\right) + \frac{\zeta_i}{2}\mathrm{dsat}_\varepsilon\left(\sum_{j\in\mathcal{N}_i} |x_j - x_i|\right),$$

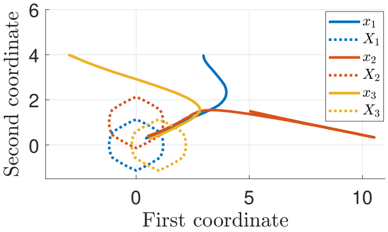
where the function $\mathrm{dsat}_\varepsilon : \mathbb{R}_{\geq 0} \to \mathbb{R}_{\geq 0}$ is given by

$$\mathrm{dsat}_\varepsilon(a) := \frac{1 + |a - \varepsilon| - |a - \varepsilon - 1|}{2}.$$

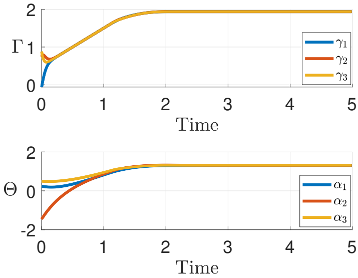


Some Simulations

Consider the distributed optimization problem with $N = 3$ agents of dimension $n = 2$.



(a) Phase portrait



(b) Evolution of γ_is and α_is

Perspectives

- Extend our framework to general perturbations, capturing false data, malicious nodes, and failures.
- Consider similar problems in discrete time.
- Consider sub-gradient, proximal, and high-order algorithms, as well as constrained problems.

Project Review

The project allowed us to

1. participate at the 2025 NOLCOS conference in Island.
2. partially support Olayo's internship at UC San Diego.
3. partially support Olayo's trip to CDC 2025 in Brazil.
4. buy a computer to a new PhD student.

Scientific production (To be presented at CDC 25)

1. On the Perturbed Projection-Based Distributed Gradient-Descent Algorithm, with T. Bazizi and P. Frasca.
2. A Necessary and Sufficient Condition for Forward Invariance in Constrained systems, with O. Reynaud and A. Hably.

Journal extensions are almost ready for submission.