



UNIVERSITÀ DEGLI STUDI
DI **CAGLIARI**



Introduction to Open Multi-Agent Systems – Part III

Descent algorithms via operator theory

Diego Deplano*

*Department of Electrical and Electronic Engineering, University of Cagliari, Italy

SIDRA Ph.D. Summer School • July 13–15, 2026 • Bertinoro, Italy

Outline

- 1 Introduction
- 2 Operator theory for open systems
- 3 The gradient descent (GD) algorithm
- 4 The proximal gradient (PG) algorithm
- 5 The distributed GD (DGD) algorithm
- 6 The DGD in open multi-agent systems
- 7 Related results in the current state-of-the-art

Outline

- 1 Introduction
- 2 Operator theory for open systems
- 3 The gradient descent (GD) algorithm
- 4 The proximal gradient (PG) algorithm
- 5 The distributed GD (DGD) algorithm
- 6 The DGD in open multi-agent systems
- 7 Related results in the current state-of-the-art

$$\begin{array}{ll} \min_x & f(x) \\ \text{s.t.} & x \in \mathcal{X}. \end{array}$$

$$\begin{array}{ll} \min & f(x) \\ \text{s.t.} & x \in \mathcal{X}. \end{array}$$

- $x \in \mathbb{R}^p$ is the decision variable;
- $f : \mathbb{R}^p \rightarrow \mathbb{R}$ is the objective function;
- $\mathcal{X} \subseteq \mathbb{R}^p$ is the admissible decision space (constraints).

In a multi-agent system, each agent holds:

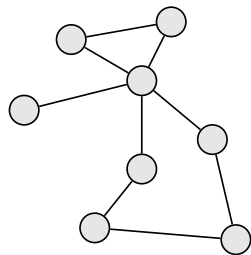
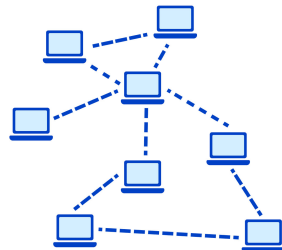
- $x_i \in \mathbb{R}^p$ is a local decision variable;
- $f_i : \mathbb{R}^p \rightarrow \mathbb{R}$ is a local objective function;
- $\mathcal{X}_i \subseteq \mathbb{R}^p$ is a local admissible decision space (constraints);

so that the optimization problem becomes **distributed**:

$$\begin{aligned} \min_{\{x_i\}_{i \in \mathcal{V}_k}} \quad & \sum_{i \in \mathcal{V}_k} f_i(x_i) \\ \text{s.t.} \quad & x_i \in \mathcal{X}_i, \quad \forall i \in \mathcal{V}_k \text{ (local constraints),} \\ & x_i = x_j, \quad \forall (i, j) \in \mathcal{E}_k \text{ (consensus constraints).} \end{aligned}$$

These slides focus on networks of agents interacting according to an **undirected, connected** graph $\mathcal{G}_k = (\mathcal{V}_k, \mathcal{E}_k)$:

- $\mathcal{V}_k$ is the time-varying set of nodes (grey circles) modeling the agents;
- $\mathcal{E}_k \subseteq \mathcal{V}_k \times \mathcal{V}_k$ is the set of undirected edges (black lines) modeling the flow of information among agents.



Given the centralized optimization problem,

$$\min_{x \in \mathcal{X}} f(x),$$

let $\mathcal{X}^*$ be the set of feasible minimizers of f over a nonempty set $\mathcal{X}$:

$$\mathcal{X}^* := \{x^* \in \mathcal{X} : f(x^*) \leq f(x), \forall x \in \mathcal{X}\}.$$

In most nontrivial problems, a solution $x^* \in \mathcal{X}^*$ cannot be computed explicitly, since no closed-form formula is available. Consequently, one aims to **design iterative algorithms** that converge to a solution $x^* \in \mathcal{X}^*$, starting from an initial guess $x_0 \in \mathcal{X}$, namely:

$$\forall k \in \mathbb{N} : \quad x(k+1) = \mathsf{T}(x(k)), \quad x(0) = x_0,$$

where:

- $k \in \mathbb{N}$ is the iteration counter;
- $x(k) \in \mathbb{R}^p$ is the current iterate;
- $x(k+1) \in \mathbb{R}^p$ is the next iterate;
- $\mathsf{T} : \mathbb{R}^p \rightarrow \mathbb{R}^p$ is the update rule (an operator) describing the iterative algorithm.

Then, the algorithm works if the iterates converge toward the set of feasible minimizers, i.e.,

$$\lim_{k \rightarrow \infty} x(k) \in \mathcal{X}^*.$$

This set of slides presents fundamental notions on centralized and distributed optimization algorithms and discusses them by exploiting results on **operator theory**.

Given the iteration

$$\forall k \in \mathbb{N}: \quad x(k+1) = \mathsf{T}(x(k)), \quad x(0) = x_0,$$

The central questions in optimization are:

- ➊ How can we design the operator T such that its repeated application converges?
- ➋ How can we ensure that it converges to the desired optimal set of solutions $\mathcal{X}^*$?

Outline

- 1 Introduction
- 2 Operator theory for open systems**
- 3 The gradient descent (GD) algorithm
- 4 The proximal gradient (PG) algorithm
- 5 The distributed GD (DGD) algorithm
- 6 The DGD in open multi-agent systems
- 7 Related results in the current state-of-the-art

Intuition: The execution of the iterative algorithm starting from a minimizer $x_0 = x^* \in \mathcal{X}^*$ should return the same x^* because it is already a solution, namely:

$$x^* = T(x^*), \quad \text{where } T \text{ is the operator governing the iteration of the algorithm.}$$

This means that x^* is a *fixed point* of the operator T .

Definition in eq. (1.68) of [R1]

Consider a self-operator $T : \mathcal{X} \rightarrow \mathcal{X}$. A point $\hat{x} \in \mathcal{X}$ is a fixed point if

$$T(\hat{x}) = \hat{x}.$$

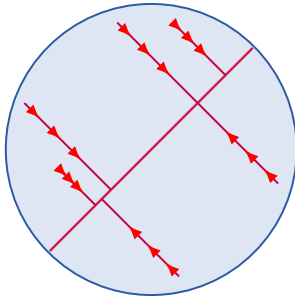
The set of fixed points is denoted by $\text{fix}(T) = \{x \in \mathcal{X} : T(x) = x\}$.

[R1] Bauschke HH and Combettes PL (2017). Convex analysis and monotone operator theory in Hilbert spaces, second ed., CMS books in mathematics, Springer, Cham.

Idea: Design an operator T whose fixed points are the minimizers x^* of f .

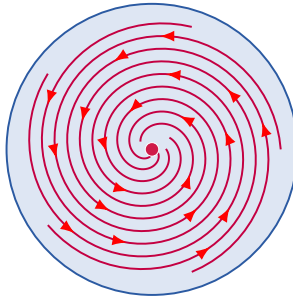
Question: How? What properties should the algorithm have?

The trajectory eventually reaches the set of fixed points



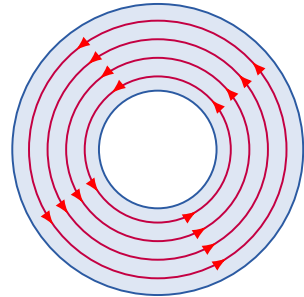
$\text{fix}(\mathbf{T})$ is a set, not a singleton

The trajectory eventually reaches the unique fixed point



$\text{fix}(\mathbf{T}) = \{x^*\}$ is a singleton

The trajectory never reaches a fixed point



$\text{fix}(\mathbf{T}) = \emptyset$ is empty

Definitions 1.47, 4.1(ii), 4.1(iii) in [R1]

An operator $T : \mathcal{X} \rightarrow \mathcal{Y}$ with $\mathcal{X} \subseteq \mathcal{Y} \subseteq \mathbb{R}^p$ is called “ ℓ -Lipschitz” if, for some $\ell \geq 0$ and for all $x, y \in \mathcal{X}$, it satisfies

$$\|T(x) - T(y)\| \leq \ell \|x - y\|. \quad (1)$$

If we let $\text{Lip}(T)$ be the minimum (or infimum) constant $\ell \geq 0$ that satisfies (1), i.e.,

$$\text{Lip}(T) := \sup_{x \neq y} \frac{\|T(x) - T(y)\|}{\|x - y\|}$$

then:

- T is “ ℓ -contractive” if $\text{Lip}(T) \leq \ell$ for some $\ell \in (0, 1)$;
- T is “nonexpansive” if $\text{Lip}(T) \leq 1$.

[R1] Bauschke HH and Combettes PL (2017). Convex analysis and monotone operator theory in Hilbert spaces, second ed., CMS books in mathematics, Springer, Cham.

Note: contractivity $\implies$ nonexpansiveness

Let us define the following notions of distance:

- Distance between two points: $d(x, y) = \|x - y\|$;
- Distance between a point and a set: $d(x, \mathcal{Y}) = \inf_{y \in \mathcal{Y}} \|x - y\|$;
- Distance between two sets: $d(\mathcal{X}, \mathcal{Y}) = \inf_{x \in \mathcal{X}} d(x, \mathcal{Y})$.

Definition 5 in [R2]

An operator $T: \mathcal{X} \rightarrow \mathcal{Y}$ with $\mathcal{X} \subseteq \mathcal{Y} \subseteq \mathbb{R}^p$ is called “ γ -paracontractive” if, for some $\gamma \in (0, 1)$ and for all $x \in \mathcal{X} \setminus \text{fix}(T)$, it satisfies

$$d(T(x), \text{fix}(T)) \leq \gamma \cdot d(x, \text{fix}(T)).$$

Equivalently, to highlight the similarity with contractivity:

$$\inf_{\hat{x} \in \text{fix}(\mathbf{T})} \|\mathbf{T}(x) - \hat{x}\| \leq \gamma \cdot \inf_{\hat{x} \in \text{fix}(\mathbf{T})} \|x - \hat{x}\|. \quad (2)$$

[R2] D. Deplano, N. Bastianello, M. Franceschelli, K.H. Johansson (2026). Optimization and learning in Open Multi-Agent Systems, IEEE Transactions on Automatic Control.

Note: contractivity $\implies$ paracontractivity

Definition 4.33 in [R1]

An operator $T : \mathcal{X} \rightarrow \mathcal{Y}$ with $\mathcal{X} \subseteq \mathcal{Y} \subseteq \mathbb{R}^p$ is called “ α -averaged” if there exists a nonexpansive operator R such that

$$T(x) = (1 - \alpha)x + \alpha R(x), \quad \forall x \in \mathcal{X}. \quad (3)$$

[R1] Bauschke HH and Combettes PL (2017). Convex analysis and monotone operator theory in Hilbert spaces, second ed., CMS books in mathematics, Springer, Cham.

Proposition 4.35 in [R1]

An operator $T : \mathcal{X} \rightarrow \mathcal{Y}$ with $\mathcal{X} \subseteq \mathcal{Y} \subseteq \mathbb{R}^p$ is “ α -averaged” if and only if, for some $\alpha \in (0, 1)$ and for all $x, y \in \mathcal{X}$, it satisfies

$$\|T(x) - T(y)\|^2 \leq \|x - y\|^2 - \frac{1 - \alpha}{\alpha} \|(x - y) - (T(x) - T(y))\|^2.$$

[R1] Bauschke HH and Combettes PL (2017). Convex analysis and monotone operator theory in Hilbert spaces, second ed., CMS books in mathematics, Springer, Cham.

What to do when the operator is nonexpansive?

Theorem 5.15(iii) in [R1] (Krasnoselskij iteration)

For a self-operator $T : \mathcal{X} \rightarrow \mathcal{X}$, consider the following iteration:

$$x(k+1) = (1 - \alpha)x(k) + \alpha T(x(k)), \quad \alpha \in (0, 1).$$

If T is nonexpansive, then the iteration is averaged and, in turn, converges to a fixed point of T , provided $\text{fix}(T) \neq \emptyset$.

[R1] Bauschke HH and Combettes PL (2017). Convex analysis and monotone operator theory in Hilbert spaces, second ed., CMS books in mathematics, Springer, Cham.

Question: How about the convergence rate?

Answer: averagedness of the iteration does not give an explicit convergence rate.

Proposition 4.44 in [R1]

Let operators $R, F : \mathcal{X} \rightarrow \mathcal{Y}$ with $\mathcal{X} \subseteq \mathcal{Y} \subseteq \mathbb{R}^p$ be averaged with constants α_R and α_F , respectively. Then, their composition operator $T = R \circ F$ is α -averaged with

$$\alpha = \frac{\alpha_R + \alpha_F - 2\alpha_R\alpha_F}{1 - \alpha_R\alpha_F}.$$

[R1] Bauschke HH and Combettes PL (2017). Convex analysis and monotone operator theory in Hilbert spaces, second ed., CMS books in mathematics, Springer, Cham.

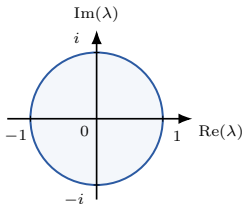
A graphical intuition of where the eigenvalues of linear operators lie

$$\mathcal{A} : x \mapsto Ax,$$

$$\sigma_{\max}(A) = \max_{\|x\|=1} \|Ax\|,$$

$$\sigma_{\max}^{\perp}(A) := \max_{\|x\|=1 \wedge x \perp \ker(A-I)} \|Ax\|$$

Nonexpansive operator

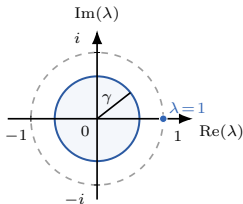


inside or on the unit disk

$$|\lambda| \leq 1$$

$$\sigma_{\max}(A) \leq 1$$

Paracontractive operator



$\lambda = 1$ allowed,
rest inside disk

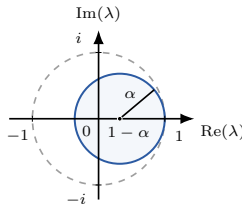
$$|\lambda| \leq \gamma < 1 \text{ for } \lambda \neq 1$$

$$\lambda = 1 \text{ is allowed}$$

$$\sigma_{\max}^{\perp}(A) \leq \gamma < 1$$

$$\lambda = 1 \text{ is semisimple}$$

Averaged operator

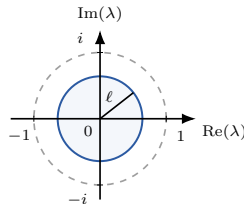


a disk tangent to 1

$$|\lambda - (1 - \alpha)| \leq \alpha$$

$$\sigma_{\max}(A - (1 - \alpha)I) \leq \alpha$$

Contractive operator



strictly inside the unit disk

$$|\lambda| \leq \ell < 1$$

$$\sigma_{\max}(A) \leq \ell < 1$$

Remark. For linear operators it holds nonexpansive $\supsetneq$ paracontractive $\supsetneq$ averaged $\supsetneq$ contractive, while this is not the case for general nonlinear operators.

Theorems 4.29 and 1.50(i) in [R1]:

For a self-operator $T : \mathcal{X} \rightarrow \mathcal{X}$, the following statements hold:

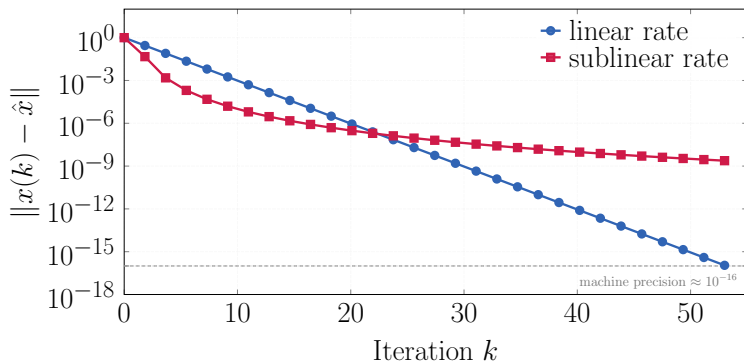
- If T is nonexpansive, then **one must assume** $\text{fix}(T) \neq \emptyset$;
- If $T = (1 - \alpha)I + \alpha R$, then by construction $\text{fix}(T) = \text{fix}(R)$, which **need not be a singleton**;
- If T is ℓ -contractive, then there exists a **unique fixed point** $\text{fix}(T) = \{\hat{x}\}$;
- If T is γ -paracontractive, then $\text{fix}(T)$ **need not be a singleton**.

Section 5.2, Proposition 5.16, and Theorem 1.50(iii) in [R1]

For a self-operator $T : \mathcal{X} \rightarrow \mathcal{X}$, consider the iteration $x(k+1) = T(x(k))$. Then, the following statements hold:

- If T is nonexpansive, then the iteration **may not** converge to a fixed point;
- If T is α -averaged, then the iteration converges to a fixed point in $\text{fix}(T)$ if $\text{fix}(T) \neq \emptyset$;
- If T is ℓ -contractive, then the iteration converges **linearly with rate ℓ** to the unique fixed point $\text{fix}(T) = \{\hat{x}\}$, in the sense that $\|x(k) - \hat{x}\| \leq \ell^k \|x(0) - \hat{x}\|$;
- If T is γ -paracontractive, then the iteration converges **linearly with rate γ** to the set of fixed points $\text{fix}(T)$ if $\text{fix}(T) \neq \emptyset$, in the sense that $\inf_{\hat{x} \in \text{fix}(T)} \|x(k) - \hat{x}\| \leq \gamma^k \cdot \inf_{\hat{x} \in \text{fix}(T)} \|x(0) - \hat{x}\|$.

[R1] Bauschke HH and Combettes PL (2017). Convex analysis and monotone operator theory in Hilbert spaces, second ed., CMS books in mathematics, Springer, Cham.



Remark: both converge to machine precision eventually, but the sublinear rate just takes exponentially longer. For instance, in the above plot:

- Linear rate r^k with $r = 0.5$: reaches machine precision at $k \approx 50$;
- Polynomial decay $1/k^d$ with $d = 5$: reaches machine precision at $k \approx 10^4$ (10 thousand iterations)

In an **open multi-agent system**, the dimension of the network state $x(k)$ may change at each step k . In turn, the update

$$x(k+1) = \mathsf{T}_k(x(k))$$

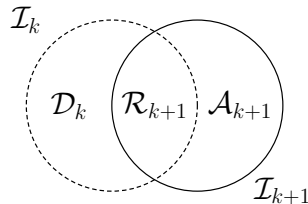
is ruled by a so-called **open operator** $\mathsf{T}_k : \mathcal{X}_k \rightarrow \mathcal{X}_{k+1}$ where $\mathcal{X}_k$ and $\mathcal{X}_{k+1}$ may have different dimensions and/or label composition.

Let $\mathcal{I}_k \neq \emptyset$ be a finite set of labels indexing the elements of $\mathcal{X}_k$, i.e.,

$$x = \begin{bmatrix} \vdots \\ x_i \\ \vdots \end{bmatrix} \in \mathcal{X}_k, \quad \text{with } i \in \mathcal{I}_k.$$

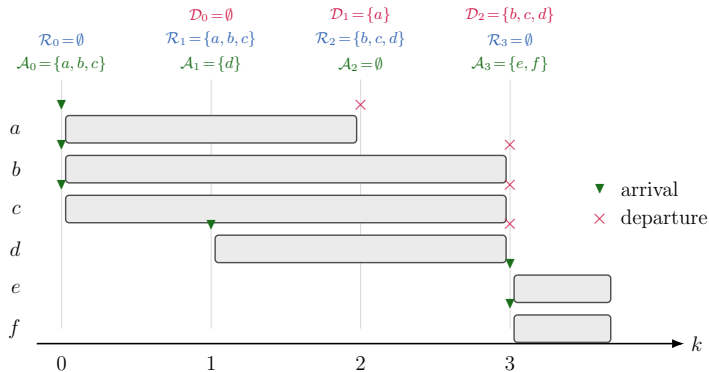
It is useful to identify the following subsets:

- **Remaining labels** $\mathcal{R}_{k+1} = \mathcal{I}_{k+1} \cap \mathcal{I}_k$: labels present both at time k and time $k+1$;
- **Arriving labels** $\mathcal{A}_{k+1} = \mathcal{I}_{k+1} \setminus \mathcal{I}_k$: labels present at time $k+1$ but not at time k ;
- **Departing labels** $\mathcal{D}_k = \mathcal{I}_k \setminus \mathcal{I}_{k+1}$: labels present at time k but not at time $k+1$.



Example 1: Consider the sets of labels $\mathcal{I}_0 = \{a, b, c\}$, $\mathcal{I}_1 = \{a, b, c, d\}$, $\mathcal{I}_2 = \{b, c, d\}$, $\mathcal{I}_3 = \{e, f\}$. Then:

- (at the initial step 0) there are no departing labels, $\mathcal{A}_0 = \{a, b, c\}$, and $\mathcal{R}_0 = \emptyset$;
- (from step 0 to step 1) $\mathcal{D}_0 = \emptyset$, $\mathcal{A}_1 = \{d\}$, $\mathcal{R}_1 = \{a, b, c\}$;
- (from step 1 to step 2) $\mathcal{D}_1 = \{a\}$, $\mathcal{A}_2 = \emptyset$, $\mathcal{R}_2 = \{b, c, d\}$;
- (from step 2 to step 3) $\mathcal{D}_2 = \{b, c, d\}$, $\mathcal{A}_3 = \{e, f\}$, $\mathcal{R}_3 = \emptyset$.



Thus, the iteration of the open operator T_k can be written component-wise for each $i \in \mathcal{I}_{k+1}$, as follows:

$$x_i(k+1) = T_{i,k}(x(k)) = \begin{cases} F_{i,k}(x(k)) & \text{if } i \in \mathcal{R}_{k+1} \\ x_{i,k+1}^A & \text{if } i \in \mathcal{A}_{k+1}, \end{cases} \quad (4)$$

where $x_{i,k+1}^A$ is the initialization of the component x_i associated with the arriving label i at step $k+1$.

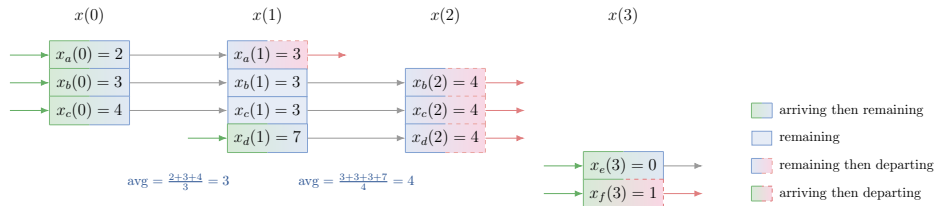
If the components of the operator remain unchanged at step k , that is, $\mathcal{I}_k = \mathcal{I}_{k+1}$, then the operator $F_k : \mathcal{X}_k \rightarrow \mathcal{X}_k$ rules the so-called **standard iteration**:

$$x(k+1) = F_k(x(k)), \quad \text{when } \mathcal{I}_k = \mathcal{I}_{k+1}. \quad (5)$$

Example 2. Consider the open operator T_k ruled by the averaging operator $F_{i,k} : x \mapsto F_{i,k}x$ defined by

$$F_{i,k} = \frac{1}{|\mathcal{I}_k|} \mathbf{1}_{|\mathcal{I}_k|}^\top, \quad \forall i \in \mathcal{I}_k.$$

If the set of labels evolves as in Example 1, then a possible open sequence generated by the iteration is:



- (from step 0 to step 1) all components associated with remaining labels update their value to the average of $x(0)$, that is $(2 + 3 + 4)/3 = 3$; label d arrives, and component $x_d(1)$ is initialized at 7;
- (from step 1 to step 2) all components associated with remaining labels update their value to the average of $x(1)$, that is $(3 + 3 + 3 + 7)/4 = 4$; label a departs, and component $x_a(2)$ is left out;
- (from step 2 to step 3) labels b, c, d depart, and components $x_b(3), x_c(3), x_d(3)$ are left out; labels e, f arrive, and components $x_e(3), x_f(3)$ are initialized at 0, 1, respectively.

Definition 1 in [R2]

Consider the iteration of an open operator $T_k : \mathcal{X}_k \rightarrow \mathcal{X}_{k+1}$ with component-wise form as in (4), and let $\hat{\mathcal{X}}_k$ be the set of fixed points of the operator $F_k : \mathcal{X}_k \rightarrow \mathcal{X}_k$ ruling the standard iteration, i.e.,

$$\hat{\mathcal{X}}_k := \{x \in \mathcal{X}_k \mid x = F_k(x)\}. \quad (6)$$

Then, the sequence $\{\hat{\mathcal{X}}_k\}_{k \in \mathbb{N}}$ is called the “trajectory of sets of interest” (TSI) of the operator T_k .

Remark. If each operator F_k has a unique equilibrium point $\hat{\mathcal{X}}_k = \{\hat{x}_k\}$, then the sequence $\{\hat{x}_k\}_{k \in \mathbb{N}}$ of these points forms the *trajectory of points of interest* (TPI).

[R2] D. Deplano, N. Bastianello, M. Franceschelli, K.H. Johansson (2026). Optimization and learning in Open Multi-Agent Systems, IEEE Transactions on Automatic Control.

Example 3. Consider the iteration described in Example 2. Then, the TSI is given by

$$\hat{\mathcal{X}}_k = \{\alpha \mathbf{1} \in \mathcal{X}_k \mid \alpha \in \mathbb{R}\}.$$

Indeed, at each $k \in \mathbb{N}$, the set of fixed points of the standard iteration ruled by $F_k : x \mapsto F_k x$ consists of the eigenspace of F_k corresponding to the (unique) unit eigenvalue, which is spanned by the vector of ones because F_k is row-stochastic.

Definition 2 in [R2]

Consider the open sequence $\{x(k) \in \mathcal{X}_k\}_{k \in \mathbb{N}}$ generated by the iteration of an open operator whose TSI is $\{\hat{\mathcal{X}}_k \subseteq \mathcal{X}_k\}_{k \in \mathbb{N}}$. The open sequence “converges” to the TSI within a radius $R \geq 0$ if

$$\limsup_{k \rightarrow \infty} \frac{d(x(k), \hat{\mathcal{X}}_k)}{\sqrt{|\mathcal{I}_k|}} \leq R.$$

If $R = 0$, then the sequence converges asymptotically to the TSI.

[R2] D. Deplano, N. Bastianello, M. Franceschelli, K.H. Johansson (2026). Optimization and learning in Open Multi-Agent Systems, IEEE Transactions on Automatic Control.

Question: Why do we need the normalization factor $\sqrt{|\mathcal{I}_k|}$?

Answer: Because without normalization, iterations of an open operator with increasing number of components may produce sequences diverging from the TSI, even if the distance between new components and their TSI counterparts remains bounded.

Example 4. Consider the iteration described in Example 2 and assume that the set of components is completely replaced at each step and its cardinality increases by 2 at each step, namely, $\mathcal{D}_k = \mathcal{I}_k$ and $|\mathcal{A}_{k+1}| = |\mathcal{I}_k| + 2$. A possible open sequence generated by the iteration $x(k+1) = \mathbf{T}_k(x(k))$ is

$$x(0) = \begin{bmatrix} +1 \\ -1 \end{bmatrix}, \quad x(1) = \begin{bmatrix} +1 \\ +1 \\ -1 \\ -1 \end{bmatrix}, \quad x(2) = \begin{bmatrix} +1 \\ +1 \\ +1 \\ -1 \\ -1 \\ -1 \end{bmatrix}, \quad \dots,$$

where half of the components are initialized at +1 and the other half at -1. The point $\hat{x}_k \in \hat{\mathcal{X}}_k$ in the TSI attaining the minimum distance from $x(k)$ is the null vector $\hat{x}_k = \mathbf{0}_{2(k+1)}$ and therefore

$$d(x(k), \hat{\mathcal{X}}_k) = d(x(k), \hat{x}_k) = d(x(k), \mathbf{0}_{2(k+1)}) = \sqrt{2(k+1)},$$

which, as $k \rightarrow \infty$, **diverges even though the new components have bounded distance of 1 from their corresponding components in the TSI**. Instead, when normalization is considered, it is possible to find a finite upper bound

$$\limsup_{k \rightarrow \infty} \frac{d(x(k), \hat{\mathcal{X}}_k)}{\sqrt{2(k+1)}} = 1.$$

We overload the distance notation by introducing the following definition of **open distance**.

Definition 3 in [R2]

Let $\mathcal{I}_1, \mathcal{I}_2$ be two finite sets of labels and let $\mathcal{X}_1, \mathcal{X}_2$ be two spaces labeled by $\mathcal{I}_1, \mathcal{I}_2$, respectively. If $\mathcal{I}_1 \cap \mathcal{I}_2 \neq \emptyset$, the “open distance” $d: \mathcal{X}_1 \times \mathcal{X}_2 \rightarrow \mathbb{R}_+$ between points $x \in \mathcal{X}_1$ and $y \in \mathcal{X}_2$ is defined by

$$d(x, y) = \|\tilde{x} - \tilde{y}\|_2, \quad \text{where} \quad \begin{cases} \tilde{x} &= [x_i \text{ for } i \in \mathcal{I}_1 \cap \mathcal{I}_2], \\ \tilde{y} &= [y_i \text{ for } i \in \mathcal{I}_1 \cap \mathcal{I}_2]. \end{cases}$$

If $\mathcal{I}_1 \cap \mathcal{I}_2 = \emptyset$, then $d(x, y) = 0$.

[R2] D. Deplano, N. Bastianello, M. Franceschelli, K.H. Johansson (2026). Optimization and learning in Open Multi-Agent Systems, IEEE Transactions on Automatic Control.

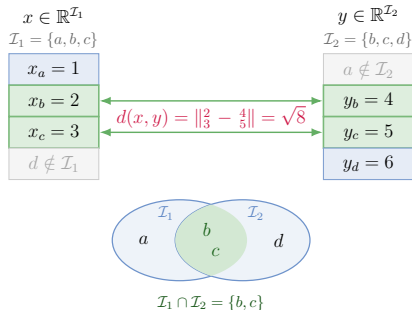
Example 5. Consider the sets of labels

$\mathcal{I}_1 = \{a, b, c\}$ and $\mathcal{I}_2 = \{b, c, d\}$ and let

$$x = \begin{bmatrix} x_a \\ x_b \\ x_c \end{bmatrix} = \begin{bmatrix} 1 \\ 2 \\ 3 \end{bmatrix} \in \mathbb{R}^{\mathcal{I}_1}, \quad y = \begin{bmatrix} y_b \\ y_c \\ y_d \end{bmatrix} = \begin{bmatrix} 4 \\ 5 \\ 6 \end{bmatrix} \in \mathbb{R}^{\mathcal{I}_2}.$$

Since the common components are those labeled by b and c , the distance between them is given by

$$d(x, y) = \left\| \begin{bmatrix} x_b \\ x_c \end{bmatrix} - \begin{bmatrix} y_b \\ y_c \end{bmatrix} \right\| = \left\| \begin{bmatrix} 2 \\ 3 \end{bmatrix} - \begin{bmatrix} 4 \\ 5 \end{bmatrix} \right\| = \sqrt{8}.$$



Definition 4 in [R2]

Let $\mathcal{I}_1, \mathcal{I}_2$ be two finite sets of labels and let $\mathcal{X}_1, \mathcal{X}_2$ be two spaces labeled by $\mathcal{I}_1, \mathcal{I}_2$, respectively. If $\mathcal{I}_1 \cap \mathcal{I}_2 \neq \emptyset$:

- the projection of a point $x \in \mathcal{X}_1$ onto $\mathcal{X}_2$ is a set of points $y \in \mathcal{X}_2$ of minimum distance from x , given by the **projection operator** defined as

$$\text{proj}(x, \mathcal{X}_2) = \{y^* \in \mathcal{X}_2 : d(x, y^*) = \inf_{y \in \mathcal{X}_2} d(x, y)\};$$

- the **shadow distance** $d_{\text{SH}} : \mathcal{X}_1 \times \mathcal{X}_2 \rightarrow \mathbb{R}_+$ between the labeled spaces $\mathcal{X}_1$ and $\mathcal{X}_2$ is defined by

$$d_{\text{SH}}(\mathcal{X}_1, \mathcal{X}_2) = \sup_{z \in \mathcal{X}_1 \cup \mathcal{X}_2} d(\text{proj}(z, \mathcal{X}_1), \text{proj}(z, \mathcal{X}_2)).$$

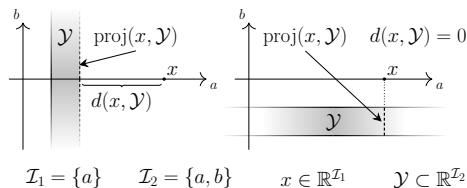
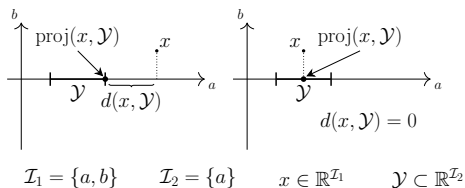
[R2] D. Deplano, N. Bastianello, M. Franceschelli, K.H. Johansson (2026). Optimization and learning in Open Multi-Agent Systems, IEEE Transactions on Automatic Control.

Remark. The concept of shadow distance is essential because it allows us to formulate the following version of the triangle inequality:

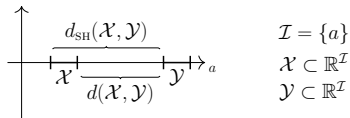
$$d(z, \mathcal{X}) \leq d(z, \mathcal{Y}) + d_{\text{SH}}(\mathcal{X}, \mathcal{Y}). \quad (7)$$

Graphical examples

Graphical examples of projections:



Graphical example of shadow distance:



Theorem 1 in [R2]

Consider a sequence of finite label sets $\{\mathcal{I}_k\}_{k \in \mathbb{N}}$ and spaces $\{\mathcal{X}_k\}_{k \in \mathbb{N}}$ labeled by those sets. Consider the iteration $x(k)$ ruled by an open operator $T_k : \mathcal{X}_k \rightarrow \mathcal{X}_{k+1}$, and split the state $x(k) = [x_k^R; x_k^A]$ into two vectors:

- $x_k^R \in \mathbb{R}^{\mathcal{R}_k}$ is the vector of the remaining components;
- $x_k^A \in \mathbb{R}^{\mathcal{A}_k}$ is the vector of the arriving components.

Assume that

- (a) F_k is paracontractive with $\gamma \in (0, 1)$;
- (b) the TSI has bounded variation $B \geq 0$, i.e., $d_{\text{SH}}(\hat{\mathcal{X}}_{k+1}, \hat{\mathcal{X}}_k) / \sqrt{|\mathcal{I}_{k+1}|} \leq B$;
- (c) the arrival process is bounded with $H \geq 0$, i.e., $\exists \hat{x}_k \in \text{proj}(x_k^R, \hat{\mathcal{X}}_k)$ s.t. $d(x_k^A, \hat{x}_k) / \sqrt{|\mathcal{I}_k|} \leq H$;
- (d) the departure process is bounded with $\beta \in (\gamma, 1)$, i.e., $\sqrt{|\mathcal{I}_{k+1}|} / \sqrt{|\mathcal{I}_k|} \geq \beta$.

Then, the open sequence $\{x(k) \in \mathcal{X}_k\}_{k \in \mathbb{N}}$ converges linearly with rate $\theta = \gamma/\beta \in (0, 1)$ to the TSI within a radius

$$R = \frac{B + H}{1 - \theta}, \quad (8)$$

according to the following pointwise upper bound

$$\frac{d(x(k), \hat{\mathcal{X}}_k)}{\sqrt{|\mathcal{I}_k|}} \leq \theta^k \frac{d(x(0), \hat{\mathcal{X}}_0)}{\sqrt{|\mathcal{I}_0|}} + \frac{1 - \theta^k}{1 - \theta} (B + H).$$

[R2] D. Deplano, N. Bastianello, M. Franceschelli, K.H. Johansson (2026). Optimization and learning in Open Multi-Agent Systems, IEEE Transactions on Automatic Control.

Proof (part 1). For any two consecutive steps $k-1, k$ with $k \in \mathbb{N}$, we have:

$$\begin{aligned}
 d(x(k), \hat{\mathcal{X}}_k) &\stackrel{(i)}{\leq} d(x(k), \hat{x}_k) = \sqrt{d^2(x_k^R, \hat{x}_k) + d^2(x_k^A, \hat{x}_k)} \\
 &\leq d(x_k^R, \hat{x}_k) + d(x_k^A, \hat{x}_k) \stackrel{(i)}{=} d(x_k^R, \hat{\mathcal{X}}_k) + d(x_k^A, \hat{x}_k) \\
 &\stackrel{(i)}{\leq} d(x_k^R, \hat{\mathcal{X}}_k) + \sqrt{|\mathcal{I}_k|}H, \\
 &\stackrel{(ii)}{\leq} d(x_k^R, \hat{\mathcal{X}}_{k-1}) + d_{\text{SH}}(\hat{\mathcal{X}}_k, \hat{\mathcal{X}}_{k-1}) + \sqrt{|\mathcal{I}_k|}H, \\
 &\stackrel{(iii)}{\leq} d(x_k^R, \hat{\mathcal{X}}_{k-1}) + \sqrt{|\mathcal{I}_k|}(B + H), \\
 &\stackrel{(iv)}{\leq} d(F_{k-1}(x(k-1)), \hat{\mathcal{X}}_{k-1}) + \sqrt{|\mathcal{I}_k|}(B + H), \\
 &\stackrel{(v)}{\leq} \gamma d(x(k-1), \hat{\mathcal{X}}_{k-1}) + \sqrt{|\mathcal{I}_k|}(B + H),
 \end{aligned}$$

where (i) holds by assumption (c) for some $\hat{x}_k \in \text{proj}(x_k^R, \hat{\mathcal{X}}_k)$; (ii) holds by the triangle inequality in (7) for the shadow distance; (iii) holds by assumption (b); (iv) holds because the vector x_k^R is entirely contained in $F_{k-1}(x(k-1))$ by definition; (v) holds by assumption (a).

Proof (part 2). Thus, we have shown that

$$\frac{d(x(k), \hat{\mathcal{X}}_k)}{\sqrt{|\mathcal{I}_k|}} \leq \gamma \frac{d(x(k-1), \hat{\mathcal{X}}_{k-1})}{\sqrt{|\mathcal{I}_k|}} + B + H.$$

This, by assumption (d), becomes

$$\frac{d(x(k), \hat{\mathcal{X}}_k)}{\sqrt{|\mathcal{I}_k|}} \leq \frac{\gamma}{\beta} \frac{d(x(k-1), \hat{\mathcal{X}}_{k-1})}{\sqrt{|\mathcal{I}_{k-1}|}} + B + H,$$

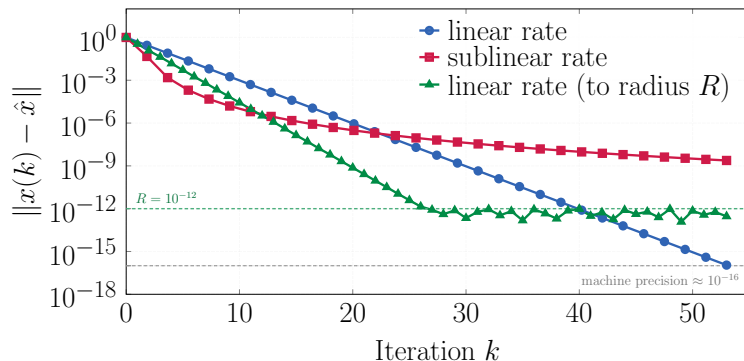
By iterating over $k \in \mathbb{N}$, we get

$$\frac{d(x(k), \hat{\mathcal{X}}_k)}{\sqrt{|\mathcal{I}_k|}} \leq \left(\frac{\gamma}{\beta}\right)^k \frac{d(x(0), \hat{\mathcal{X}}_0)}{\sqrt{|\mathcal{I}_0|}} + (B + H) \sum_{i=0}^{k-1} \left(\frac{\gamma}{\beta}\right)^i.$$

Since, as $k \rightarrow \infty$, the term $(\gamma/\beta)^k$ tends to 0 and the geometric sum equals $(1 - \theta^k)/(1 - \theta)$ with limit $1/(1 - \gamma/\beta)$, we have

$$\limsup_{k \rightarrow \infty} \frac{d(x(k), \hat{\mathcal{X}}_k)}{\sqrt{|\mathcal{I}_k|}} \leq \frac{B + H}{1 - \frac{\gamma}{\beta}} = R.$$

This concludes the proof.



Remark: almost always, an open multi-agent system never converges to machine precision because the join/leave events persistently change the equilibrium points of the system.

Outline

- 1 Introduction
- 2 Operator theory for open systems
- 3 The gradient descent (GD) algorithm**
- 4 The proximal gradient (PG) algorithm
- 5 The distributed GD (DGD) algorithm
- 6 The DGD in open multi-agent systems
- 7 Related results in the current state-of-the-art

We consider unconstrained optimization problems:

$$\min_{x \in \mathbb{R}^p} f(x).$$

A solution to the problem can be computed by means of the gradient descent algorithm

$$x(k+1) = x(k) - \rho \nabla f(x(k))$$

if the objective function f is:

- proper;
- differentiable;
- **convex**;
- **L-smooth**.

Section 1.6 in [R1]

A function $f : \mathcal{X} \rightarrow \mathbb{R} \cup \{+\infty\}$ is proper if its domain is nonempty and it never attains $-\infty$.

[R1] Bauschke HH and Combettes PL (2017). Convex analysis and monotone operator theory in Hilbert spaces, second ed., CMS books in mathematics, Springer, Cham.

Convexity: the definition (for functions)

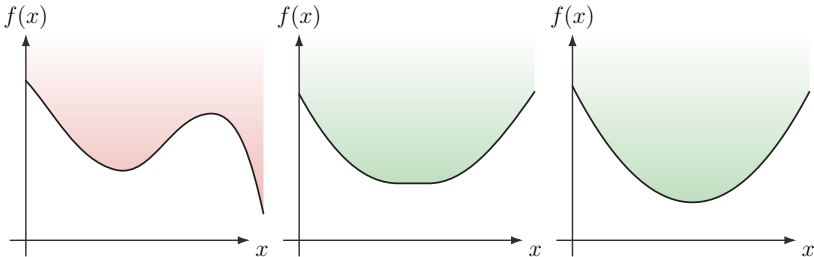
Definition 8.1 in [R1]

A function $f: \mathcal{X} \rightarrow \mathbb{R} \cup \{+\infty\}$ with $\mathcal{X} \subseteq \mathbb{R}^p$ is convex if its epigraph

$$\text{epi}(f) = \{(x, \xi) \in \mathbb{R}^p \times \mathbb{R} : f(x) \leq \xi\}$$

is a convex subset of $\mathbb{R}^p \times \mathbb{R}$.

[R1] Bauschke HH and Combettes PL (2017). *Convex analysis and monotone operator theory in Hilbert spaces*, second ed., CMS books in mathematics, Springer, Cham.



Convexity: a necessary and sufficient condition

Propositions 8.4 and 10.8 in [R1]

A function $f: \mathcal{X} \rightarrow \mathbb{R} \cup \{+\infty\}$ is convex if and only if

$$f(\alpha x + (1 - \alpha)y) \leq \alpha f(x) + (1 - \alpha)f(y), \quad \forall \alpha \in (0, 1), \forall x, y \in \mathcal{X}.$$

Moreover:

- f is strictly convex if the above holds strictly with “ $<$ ”;
- f is m -strongly convex if and only if $f(\cdot) - \frac{m}{2} \|\cdot\|^2$ is convex.

[R1] Bauschke HH and Combettes PL (2017). *Convex analysis and monotone operator theory in Hilbert spaces*, second ed., CMS books in mathematics, Springer, Cham.

Differentiability of functions

Definition 17.1 in [R1]

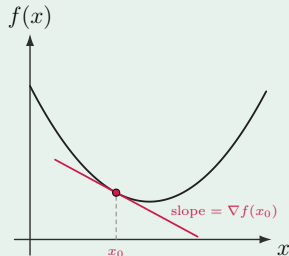
A proper function $f : \mathcal{X} \rightarrow \mathbb{R} \cup \{+\infty\}$ with $\mathcal{X} \subseteq \mathbb{R}^p$ is differentiable at $x \in \mathcal{X}$ if for every direction $v \in \mathbb{R}^p$ the limit

$$f'(x, v) = \lim_{h \rightarrow 0^\pm} \frac{f(x + hv) - f(x)}{h},$$

exists, is finite, and it is linear in v . If f is differentiable at every $x \in \mathcal{X}$, then we simply say that f is differentiable. For differentiable functions f , the **gradient operator** $\nabla f : \mathcal{X} \rightarrow \mathbb{R}^p$ is defined by

$$\nabla f(x) = \begin{bmatrix} f'(x, e_1) \\ \vdots \\ f'(x, e_i) \\ \vdots \\ f'(x, e_p) \end{bmatrix} \quad \text{where} \quad e_i = \begin{bmatrix} 0 \\ \vdots \\ 0 \\ \underbrace{1}_{i\text{-th element}} \\ 0 \\ \vdots \\ 0 \end{bmatrix}.$$

[R1] Bauschke HH and Combettes PL (2017). Convex analysis and monotone operator theory in Hilbert spaces, second ed., CMS books in mathematics, Springer, Cham.



The gradient descent algorithm via the forward step operator

Idea: Design an operator T whose fixed point is the minimizer x^* of f .

A possible choice is the gradient descent operator

$$T(x) = x - \rho \nabla f(x), \quad \rho > 0,$$

which, for convex f , satisfies

$$\nabla f(x^*) = 0 \quad \Leftrightarrow \quad T(x^*) = x^*.$$

Thus, minimizers of f are fixed points of T , and vice versa.

Does it work? (Is the operator contractive or averaged?)

Relation between convexity and monotonicity for differentiable functions

Definition 22.1 in [R1]

An operator $T : \mathcal{X} \rightarrow \mathbb{R}^p$ is strongly monotone with constant $\beta > 0$ if, for all $x, y \in \mathcal{X}$,

$$(\mathsf{T}(x) - \mathsf{T}(y))^\top (x - y) \geq \beta \|x - y\|^2.$$

If the above holds with $\beta = 0$, then T is simply monotone.

[R1] Bauschke HH and Combettes PL (2017). Convex analysis and monotone operator theory in Hilbert spaces, second ed., CMS books in mathematics, Springer, Cham.

Proposition 17.7 (i) $\Leftrightarrow$ (iii) in [R1]

Consider a differentiable function $f: \mathcal{X} \rightarrow \mathbb{R}$ on an open convex set $\mathcal{X}$. Then, f is (strongly) convex if and only if ∇f is (strongly) monotone, i.e.,

$$(\nabla f(x) - \nabla f(y))^\top (x - y) \geq m \|x - y\|^2, \quad \forall x, y \in \mathcal{X}.$$

[R1] Bauschke HH and Combettes PL (2017). Convex analysis and monotone operator theory in Hilbert spaces, second ed., CMS books in mathematics, Springer, Cham.

The gradient descent algorithm: Lipschitz gradients

Additional assumption: Assume that the gradient of f is L -Lipschitz, i.e., f is L -smooth.

$$\|\nabla f(x) - \nabla f(y)\| \leq L\|x - y\|.$$

Definition 20.1 in [R1]

An operator $T : \mathcal{X} \rightarrow \mathbb{R}^P$ is β -cocoercive if, for all $x, y \in \mathcal{X}$,

$$(T(x) - T(y))^\top (x - y) \geq \beta \|T(x) - T(y)\|.$$

[R1] Bauschke HH and Combettes PL (2017). Convex analysis and monotone operator theory in Hilbert spaces, second ed., CMS books in mathematics, Springer, Cham.

Corollary 18.17 and Proposition 4.39 in [R1]

Let $f : \mathbb{R}^P \rightarrow \mathbb{R}$ be a convex, differentiable function. If f is L -smooth, then the operator ∇f is $\frac{1}{L}$ -cocoercive,

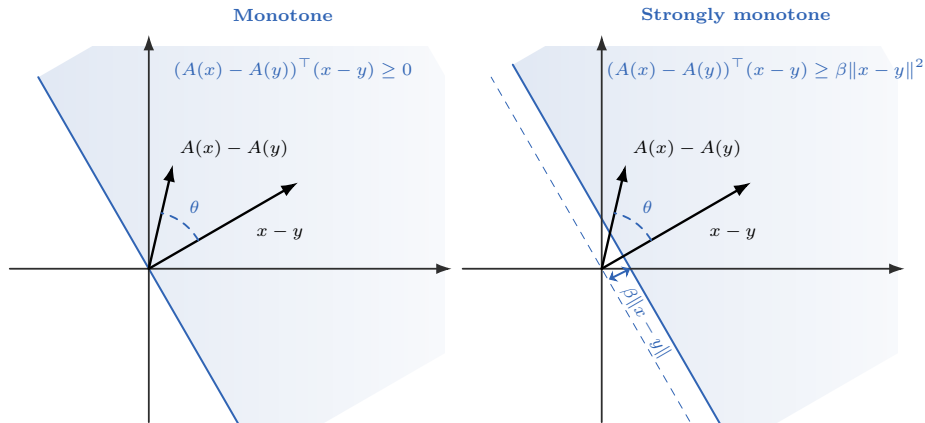
$$(\nabla f(x) - \nabla f(y))^\top (x - y) \geq \frac{1}{L} \|\nabla f(x) - \nabla f(y)\|^2. \quad (9)$$

Additionally, $I - \rho \nabla f$ is α -averaged with $\alpha = \rho L/2$ for $\rho \in (0, 2/L)$.

[R1] Bauschke HH and Combettes PL (2017). Convex analysis and monotone operator theory in Hilbert spaces, second ed., CMS books in mathematics, Springer, Cham.

Cocoercivity is sometimes called **inverse strong monotonicity** because, if ∇f is surjective, it corresponds to strong monotonicity of ∇f^{-1} . Indeed, for $z = \nabla f(x)$, $w = \nabla f(y)$: $(z - w)^\top (\nabla f^{-1}(z) - \nabla f^{-1}(w)) \geq \frac{1}{L} \|z - w\|^2$.

Graphical representation of monotonicity and strong monotonicity



$$(A(x) - A(y))^T (x - y) = \|A(x) - A(y)\| \|x - y\| \cos(\theta)$$

The gradient descent algorithm: the main result for convex functions

Theorem (Gradient Descent Algorithm)

Consider the unconstrained optimization problem

$$\min_{x \in \mathbb{R}^p} f(x),$$

and assume that $f : \mathbb{R}^p \rightarrow \mathbb{R}$ is **convex**, differentiable, L -smooth, and admits at least one minimizer. Then, the iteration

$$x(k+1) = x(k) - \rho \nabla f(x(k))$$

converges (**not necessarily at a linear rate**) to a minimizer x^* of f if $\rho \in (0, \frac{2}{L})$, i.e.,

$$\lim_{k \rightarrow \infty} x(k) \in \mathcal{X}^*, \quad \forall \rho \in \left(0, \frac{2}{L}\right),$$

where $\mathcal{X}^*$ is the set of minimizers of f .

The gradient descent algorithm: proof for convex functions

Proof (via Krasnoselskij, Theorem 5.15 in [R1]):

Step 1. Rewrite the gradient descent operator as a Krasnoselskij iteration with $\alpha \in (0, 1)$:

$$x(k+1) = x(k) - \rho \nabla f(x(k)) = (1 - \alpha) x(k) + \underbrace{\alpha \left(x(k) - \frac{\rho}{\alpha} \nabla f(x(k)) \right)}_{R(x(k))}.$$

Step 2: R is nonexpansive for $\rho \leq 2\alpha/L$ (via Corollary 18.17):

$$\begin{aligned} \|R(x) - R(y)\|^2 &= \|x - y\|^2 - \frac{2\rho}{\alpha} (\nabla f(x) - \nabla f(y))^\top (x - y) + \frac{\rho^2}{\alpha^2} \|\nabla f(x) - \nabla f(y)\|^2 \\ &\leq \|x - y\|^2 - \frac{2\rho}{\alpha L} \|\nabla f(x) - \nabla f(y)\|^2 + \frac{\rho^2}{\alpha^2} \|\nabla f(x) - \nabla f(y)\|^2 \\ &\leq \|x - y\|^2 + \underbrace{\frac{\rho}{\alpha} \left(\frac{\rho}{\alpha} - \frac{2}{L} \right)}_{\leq 0 \Leftrightarrow \alpha \geq \rho L/2} \|\nabla f(x) - \nabla f(y)\|^2 \leq \|x - y\|^2 \end{aligned}$$

Step 3: Convergence. Select $\alpha = \rho L/2$. Then, $\alpha \in (0, 1)$ if and only if $\rho \in (0, 2/L)$. Since R is nonexpansive and $\text{fix}(R) = \text{fix}(T) = \mathcal{X}^* \neq \emptyset$, by Theorem 5.15 the iteration converges to some $x^* \in \text{fix}(R) = \text{fix}(T)$.

Step 4: Fixed points are minimizers. $T(x^*) = x^* \Leftrightarrow \nabla f(x^*) = 0 \Leftrightarrow x^* \in \mathcal{X}^*$ (Fermat's rule). □

The gradient descent algorithm: strongly convex functions

Additional assumption: Assume that f is m -strongly convex.

Consequence of Theorem 18.15

Let $f : \mathbb{R}^p \rightarrow \mathbb{R}$ be an m -strongly convex, differentiable function. If f is L -smooth, then

$$(\nabla f(x) - \nabla f(y))^\top (x - y) \geq \frac{1}{L + m} \|\nabla f(x) - \nabla f(y)\|^2 + \frac{Lm}{L + m} \|x - y\|^2. \quad (10)$$

[R1] Bauschke HH and Combettes PL (2017). *Convex analysis and monotone operator theory in Hilbert spaces*, second ed., CMS books in mathematics, Springer, Cham.

Proof. Define $g(\cdot) := f(\cdot) - \frac{m}{2}\|\cdot\|^2$. Then g is convex because f is m -strongly convex and $(L - m)$ -smooth because $\nabla g(\cdot) = \nabla f(\cdot) - m(x - y)$ and, in turn, by Cauchy-Schwarz and L -smoothness of f :

$$(\nabla g(x) - \nabla g(y))^\top (x - y) = (\nabla f(x) - \nabla f(y))^\top (x - y) - m\|x - y\|^2 \leq (L - m)\|x - y\|^2. \text{ Thus:}$$

$$\begin{aligned} (\nabla g(x) - \nabla g(y))^\top (x - y) &\geq (L - m)^{-1} \|\nabla g(x) - \nabla g(y)\|^2 && \text{(cocoercivity of } \nabla g) \\ (L - m)(u - mv)^\top v &\geq \|u - mv\|^2 && u =: \nabla f(x) - \nabla f(y), v =: x - y \end{aligned}$$

$$\begin{aligned}(L-m)u^\top v - m(L-m)\|v\|^2 &\geq \|u\|^2 - 2mu^\top v + m^2\|v\|^2 \\ (L+m)u^\top v &\geq \|u\|^2 + (m(L-m) + m^2)\|v\|^2 = \|u\|^2 + mL\|v\|^2 \\ u^\top v &\geq (L+m)^{-1}(\|u\|^2 + mL\|v\|^2)\end{aligned}$$

The gradient descent algorithm: the main result for strongly convex functions

Theorem (Gradient Descent Algorithm)

Consider the unconstrained optimization problem

$$\min_{x \in \mathbb{R}^P} f(x),$$

and assume that $f: \mathbb{R}^p \rightarrow \mathbb{R}$ is m -strongly convex, differentiable, and L -smooth. Then, the iteration

$$x(k+1) = x(k) - \rho \nabla f(x(k))$$

converges (at a linear rate) to the unique minimizer x^* of f if $\rho \in (0, \frac{2}{L})$, i.e.,

$$\|x(k) - x^*\| \leq \ell^k \|x(0) - x^*\|, \quad \ell = \max\{|1 - \rho m|, |1 - \rho L|\} < 1.$$

Moreover, the best linear rate is achieved for $\rho^* = \frac{2}{L+m}$ and it is equal to

$$\ell^* = \frac{L - m}{L + m}.$$

The gradient descent algorithm: proof for strongly convex functions

Proof (via the Banach contraction theorem, Theorem 1.50(iii) in [R1]):

Step 1. Let T be the gradient iteration:

$$x(k+1) = \mathbb{T}(x(k)) = x(k) - \rho \nabla f(x(k)).$$

Step 2: T is contractive for $\rho \in (0, 2/L)$ (via Theorem 18.15):

$$\begin{aligned}\|\mathsf{T}(x) - \mathsf{T}(y)\|^2 &= \|x - y\|^2 - 2\rho(\nabla f(x) - \nabla f(y))^\top(x - y) + \rho^2\|\nabla f(x) - \nabla f(y)\|^2 \\ &\leq (1 - \frac{2\rho Lm}{L+m})\|x - y\|^2 + \rho(\rho - \frac{2}{L+m})\|\nabla f(x) - \nabla f(y)\|^2\end{aligned}$$

If $\rho \leq \frac{2}{L+m}$, use m -strong convexity:

$$\|\mathbf{T}(x) - \mathbf{T}(y)\|^2 \leq (1 - \frac{2\rho Lm}{L+m} + \rho(\rho - \frac{2}{L+m})m^2)\|x - y\|^2 = (1 - \rho m)^2\|x - y\|^2.$$

If $\rho \geq \frac{2}{L+m}$, use L -smoothness: $\|\mathsf{T}(x) - \mathsf{T}(y)\|^2 \leq (1 - \frac{2\rho Lm}{L+m} + \rho(\rho - \frac{2}{L+m})L^2)\|x - y\|^2 = (1 - \rho L)^2\|x - y\|^2$.

The contraction rate is $\ell := \max\{|1 - \rho m|, |1 - \rho L|\} < 1$ for $\rho \in (0, 2/L)$, since $0 < m \leq L$.

Step 3: Convergence. Since T is contractive, by Theorem 1.50(iii) the iteration converges to the unique $x^* \in \text{fix}(T)$, and at $\rho^* = \frac{2}{L+m}$ the contraction factor is minimized to $\ell^* = \frac{L-m}{L+m}$.

Step 4: The fixed point is the minimizer. $T(x^*) = x^* \Leftrightarrow \nabla f(x^*) = 0 \Leftrightarrow \{x^*\} = \mathcal{X}^*$ by Fermat's rule and strong convexity.

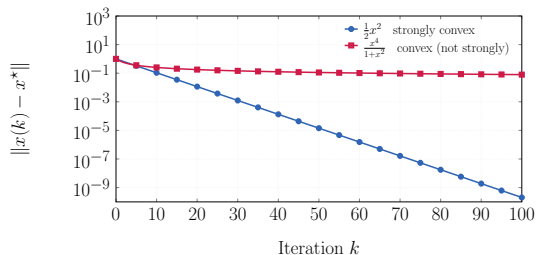
Gradient descent: numerical example 1

Strongly convex: $f(x) = \frac{1}{2}x^2$

- $\nabla f(x) = x$
- $m = 1, L = 1$
- Step-size range: $\rho \in (0, 2)$
- Chosen: $\rho = 0.2$, rate $\ell = |1 - \rho| = 0.8$
- **Linear convergence:** $|x(k)| \leq 0.8^k$

Convex, not strongly convex: $f(x) = \frac{x^4}{1+x^2}$

- $\nabla f(x) = \frac{2x^3(x^2+2)}{(1+x^2)^2}$
- $m = 0, L = 2.5$
- Step-size range: $\rho \in (0, 0.8)$
- Chosen: $\rho = 0.2$
- **Sublinear convergence:** $|x(k)| = \mathcal{O}(1/\sqrt{k})$



Both start from $x(0) = 1$ with minimizer $x^* = 0$.

Gradient descent: numerical example 2

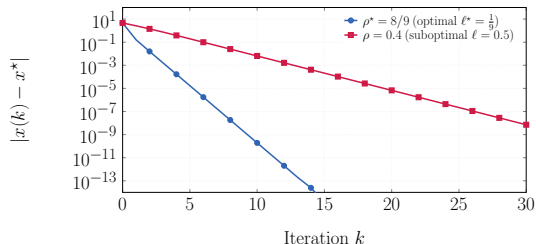
Consider the regularized logistic loss:

$$f(x) = \log(1 + e^{-x}) + \frac{1}{2}x^2$$

- $\nabla f(x) = x - \frac{1}{1 + e^x}$
- $m = 1, L = \frac{5}{4}, \rho^* = \frac{8}{9}, \ell^* = \frac{1}{9}$
- Optimality condition $\nabla f(x^*) = 0$:

$$x^* = \frac{1}{1 + e^{x^*}} \approx 0.40$$

Since there is no closed-form solution, one needs to use an iterative algorithm like GD to compute x^* .



Both start from $x(0) = 5$, $x^* \approx 0.40$.

Exercise: gd algorithm with a quadratic objective function

Exercise (30 min): Consider the following optimization problem:

$$\min_{x \in \mathbb{R}^p} f(x), \quad \text{with} \quad f(x) = \frac{1}{2} x^\top Q x + b^\top x$$

with $x \in \mathbb{R}^p$, $Q \in \mathbb{R}^{p \times p}$, $b \in \mathbb{R}^p$, under the following assumptions:

- Q is symmetric, i.e., $Q^\top = Q$;
- Q is positive semidefinite, i.e., $x^\top Q x \geq 0$ (given the above).

Tasks:

- 1 Verify that $f(x)$ is differentiable and, if it is, compute $\nabla f(x)$;
- 2 Verify that $f(x)$ is convex;
- 3 Under which additional assumption $f(x)$ is m -strongly convex? Determine the constant $m \geq 0$.
- 4 Verify that $f(x)$ is L -smooth and determine the constant $L \geq 0$;
- 5 Does L or m depend on Q and/or b ? Could you explain why?
- 6 Discuss the convergence of the gradient descent algorithm $x(k+1) = x(k) - \rho \nabla f(x(k))$.
- 7 Simulate the gradient descent algorithm with $\rho = 0.1/L$, $\rho = 1/L$, $\rho = 3/L$ and discuss the convergence.

Outline

- 1 Introduction
- 2 Operator theory for open systems
- 3 The gradient descent (GD) algorithm
- 4 The proximal gradient (PG) algorithm**
- 5 The distributed GD (DGD) algorithm
- 6 The DGD in open multi-agent systems
- 7 Related results in the current state-of-the-art

We consider unconstrained, composite optimization problems:

$$\min_{x \in \mathbb{R}^p} f(x) + g(x).$$

A solution to the problem can be computed by means of the proximal gradient algorithm

$$x(k+1) = \text{prox}_g^\rho(x(k) - \rho \nabla f(x(k)))$$

if the objective components satisfy the following conditions:

- convexity: both f and g ;
- differentiability: only f ;
- L -smoothness: only f ;
- lower semicontinuity (lsc): only g .

Subdifferentiability of functions

Definition 16.1 in [R1]

The subdifferential $\partial g : \mathcal{X} \rightarrow 2^{\mathbb{R}^p}$ of a function $g : \mathcal{X} \rightarrow \mathbb{R} \cup \{+\infty\}$ is a set-valued operator defined by

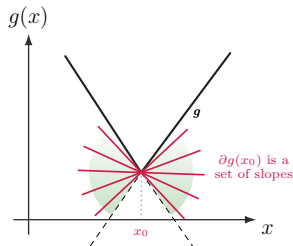
$$\partial g(x) := \{v \in \mathbb{R}^p : g(y) \geq g(x) + v^\top (y - x), \forall y \in \mathcal{X}\}.$$

We say that g is subdifferentiable at x if $\partial g(x) \neq \emptyset$, and the elements of $\partial g(x)$ are called subgradients.

[R1] Bauschke HH and Combettes PL (2017). Convex analysis and monotone operator theory in Hilbert spaces, second ed., CMS books in mathematics, Springer, Cham.

Remarks:

- If g is convex and x lies in the relative interior of its domain, then $\partial g(x) \neq \emptyset$;
- If g is convex and differentiable, then its gradient at x is the only subgradient, i.e., $\partial g(x) = \{\nabla g(x)\}$;
- A subgradient of g at x can exist even when g is not differentiable at x .



Corollary 27.6 in [R1] (Consequence of the Fermat's rule)

Let $f : \mathcal{X} \rightarrow \mathbb{R}$ and $g : \mathcal{X} \rightarrow \mathbb{R} \cup \{+\infty\}$ be proper, convex functions and assume that f is differentiable. Then, if f and g satisfy a qualification condition^o, minimizers x^* of $f + g$ satisfy

$$x^* \in \operatorname{argmin}(f + g) \quad \Longleftrightarrow \quad 0 \in (\nabla f + \partial g)(x^*).$$

[R1] Bauschke HH and Combettes PL (2017). Convex analysis and monotone operator theory in Hilbert spaces, second ed., CMS books in mathematics, Springer, Cham.

^oThe most basic qualification condition (see Cor. 27.6(b) in [R1]) is that the intersection between the relative interiors of the domains of f and g is nonempty.

Operator splitting

Our problem has therefore become that of finding a zero of the sum of the gradient and subdifferential operators, namely:

$$0 \in (\nabla f + \partial g)(x).$$

Idea: Transform this problem into a fixed-point iteration with operators constructed from ∇f , ∂g :

$$0 \in (\nabla f + \partial g)(x)$$

$$0 \in (\rho \nabla f + \rho \partial g)(x)$$

$$0 \in (\rho \nabla f + \rho \partial g \pm I)(x)$$

$$0 \in \left((I + \rho \partial g) - (I - \rho \nabla f) \right)(x)$$

$$(I - \rho \nabla f)(x) \in (I + \rho \partial g)(x)$$

$$\text{(by maximal monotonicity of } \partial g) \quad \Rightarrow \quad \underbrace{(I + \rho \partial g)^{-1}}_{J_{\rho \partial g}} (I - \rho \nabla f)(x) = x$$

where $J_{\rho \partial g} := (I + \rho \partial g)^{-1}$ is known as the **resolvent operator**.

Proposition 16.44 in [R1]

Let $g : \mathcal{X} \rightarrow \mathbb{R} \cup \{+\infty\}$ be proper, convex, and lsc. Then, the resolvent operator becomes the so-called proximal operator defined by

$$J_{\rho \partial g}(x) = \text{prox}_g^\rho(x) := \underset{y \in \mathcal{X}}{\operatorname{argmin}} \left\{ g(y) + \frac{1}{2\rho} \|x - y\|^2 \right\}.$$

[R1] Bauschke HH and Combettes PL (2017). Convex analysis and monotone operator theory in Hilbert spaces, second ed., CMS books in mathematics, Springer, Cham.

Propositions 12.28 and 4.4 in [R1]

Let $g : \mathcal{X} \rightarrow \mathbb{R} \cup \{+\infty\}$ be proper, convex, and lsc. Then the proximal operator prox_g^ρ is 1/2-averaged.

[R1] Bauschke HH and Combettes PL (2017). Convex analysis and monotone operator theory in Hilbert spaces, second ed., CMS books in mathematics, Springer, Cham.

The proximal gradient algorithm via the resolvent operator

Idea: Design an operator T whose fixed points are the minimizers x^* of $f + g$.

A possible choice is given by the proximal operator

$$T(x) = \text{prox}_g^\rho(x - \rho \nabla f(x)), \quad \rho > 0,$$

which satisfies

$$0 \in (\nabla f + \partial g)(x^*) \iff T(x^*) = x^*.$$

Thus, minimizers of $f + g$ are fixed points of T , as desired.

Does it work? (Is the operator contractive or averaged?)

The proximal gradient algorithm: the main result for convex functions

Theorem (Proximal Gradient Algorithm)

Consider the unconstrained, composite optimization problem

$$\min_{x \in \mathbb{R}^p} f(x) + g(x),$$

and assume that $f : \mathbb{R}^p \rightarrow \mathbb{R}$ is proper, lsc, **convex**, differentiable, and L -smooth, that g is proper, lsc, and convex, that $f + g$ admit at least one minimizer and satisfy a qualification condition. Then, the iteration

$$x(k+1) = \text{prox}_g^\rho(x(k) - \rho \nabla f(x(k)))$$

converges (**not necessarily at a linear rate**) to a minimizer x^* of $f + g$ if $\rho \in (0, \frac{2}{L})$, i.e.,

$$\lim_{k \rightarrow \infty} x(k) \in \mathcal{X}^*, \quad \forall \rho \in \left(0, \frac{2}{L}\right),$$

where $\mathcal{X}^*$ is the set of minimizers of $f + g$.

The proximal gradient algorithm: proof for convex functions

Proof (via composition of averaged operators):

Step 1: the gradient step is averaged. Since f is convex and L -smooth, then $I - \rho \nabla f$ is $\frac{\rho L}{2}$ -averaged for all $\rho \in (0, \frac{2}{L})$ as shown in the proof of the gradient descent algorithm.

Step 2: the proximal step is averaged. prox_g^ρ is $\frac{1}{2}$ -averaged (Prop. 12.28 and Prop. 4.4 in [R1]).

Step 3: the composition is averaged. By Prop. 4.44 in [R1], the composition $T := \text{prox}_g^\rho \circ (I - \rho \nabla f)$ is α -averaged with

$$\alpha = \frac{\alpha_1 + \alpha_2 - 2\alpha_1\alpha_2}{1 - \alpha_1\alpha_2} = \frac{2}{4 - \rho L} \in (0, 1), \quad \alpha_1 = \frac{1}{2}, \quad \alpha_2 = \frac{\rho L}{2}.$$

Step 4: convergence. Since T is averaged and $\text{fix}(T) = \mathcal{X}^* \neq \emptyset$, the iteration $x(k+1) = T(x(k))$ converges to some $x^* \in \text{fix}(T)$ (Prop. 5.16 in [R1]). Finally, by Cor. 27.6 in [R1] (Fermat's rule),

$$x^* = \text{prox}_g^\rho(x^* - \rho \nabla f(x^*)) \iff 0 \in \nabla f(x^*) + \partial g(x^*) \iff x^* \in \mathcal{X}^*.$$

The proximal gradient algorithm: the main result for strongly convex functions

Theorem (Proximal Gradient Algorithm)

Consider the unconstrained, composite optimization problem

$$\min_{x \in \mathbb{R}^p} f(x) + g(x),$$

and assume that $f : \mathbb{R}^p \rightarrow \mathbb{R}$ is proper, lsc, **m -strongly convex**, differentiable, and L -smooth, that g is proper, lsc, and convex, that $f + g$ admit at least one minimizer and satisfy a qualification condition. Then, the iteration

$$x(k+1) = \text{prox}_g^\rho(x(k) - \rho \nabla f(x(k)))$$

converges **at a linear rate** to the unique minimizer x^* of $f + g$ if $\rho \in (0, \frac{2}{L})$, i.e.,

$$\|x(k) - x^*\| \leq \ell^k \|x(0) - x^*\|, \quad \ell = \max\{|1 - \rho m|, |1 - \rho L|\} < 1.$$

Moreover, the best value of this contraction bound is achieved for $\rho^* = \frac{2}{m+L}$ and it is equal to

$$\ell^* = \frac{L - m}{L + m}.$$

The proximal gradient algorithm: proof for strongly convex functions

Proof (via the Banach contraction theorem, Theorem 1.50(iii) in [R1]):

Step 1: the gradient step $I - \rho \nabla f$ is a contraction for all $\rho \in (0, 2/L)$ with contraction rate $\ell := \max\{|1 - \rho m|, |1 - \rho L|\} < 1$ as shown in the proof of the gradient descent algorithm.

Step 2: the proximal step is nonexpansive. prox_g^ρ is averaged (Prop. 12.28 and 4.4 in [R1]), hence nonexpansive.

Step 3: the composition is a contraction. The composition of a nonexpansive and a contractive operator is a contractive operator with the same constant. Indeed, for $T := \text{prox}_g^\rho \circ (I - \rho \nabla f)$ one has

$$\begin{aligned} \|T(x) - T(y)\| &= \|\text{prox}_g^\rho(x - \rho \nabla f(x)) - \text{prox}_g^\rho(y - \rho \nabla f(y))\| \\ &\leq \|(I - \rho \nabla f)(x) - (I - \rho \nabla f)(y)\| \leq \ell \|x - y\|. \end{aligned}$$

Step 4: convergence. By the proximal fixed-point equivalence and Prop. 26.1 in [R1], $\text{fix}(T) = \mathcal{X}^*$. Since T is contractive, $\text{fix}(T) = \{x^*\}$ is a singleton; hence $\mathcal{X}^* = \{x^*\}$, and linear convergence follows from Thm. 1.50 in [R1].

Proximal gradient descent: a numerical example

Minimize $f(x) + \lambda \|x\|_1$ via proximal gradient descent:

$$x(k+1) = \text{prox}_{\rho\lambda\|\cdot\|_1}(x(k) - \rho\nabla f(x(k)))$$

with $c = (5, 3, 1, 0.5) \in \mathbb{R}^4$, $\lambda = 1.5$, $\rho = 0.2$.

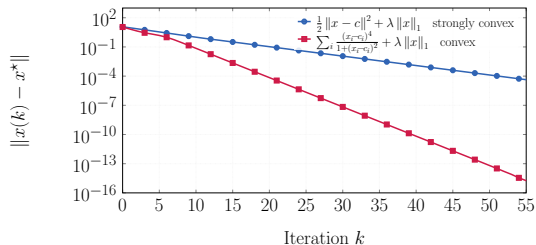
Strongly convex: $f(x) = \frac{1}{2}\|x - c\|^2$

- $x^* = \text{soft}_\lambda(c) = (3.5, 1.5, 0, 0)$
- **Linear convergence**

Convex, not strongly convex: $f(x) = \sum_i \frac{(x_i - c_i)^4}{1 + (x_i - c_i)^2}$

- $x^* = (4, 2, 0, 0)$
- **Linear convergence** (locally strongly convex at x^*)

The ℓ_1 penalty **sparsifies** the solution: components $c_3 = 1$ and $c_4 = 0.5$ are zeroed out.



Both start from $x(0) = (10, 8, 6, 4)$.

Some useful closed forms of proximal operators

These formulas are drawn from Section 6 in [R3].

Quadratic:	$f(x) = \frac{1}{2}x^\top Ax + b^\top x + c$	$\text{prox}_f^\rho(x) = (I + \rho A)^{-1}(x - \rho b)$
2-norm:	$f(x) = \ x\ _2$	$\text{prox}_f^\rho(x) = \begin{cases} \left(1 - \frac{\rho}{\ x\ _2}\right)x & \text{if } \ x\ _2 \geq \rho, \\ 0 & \text{otherwise.} \end{cases}$
Logarithm (scalar):	$f(x) = -\log(x)$	$\text{prox}_f^\rho(x) = \frac{x + \sqrt{x^2 + 4\rho}}{2}$
Absolute value (scalar)	$f(x) = x $	$\text{prox}_f^\rho(x) = \begin{cases} x - \rho & x \geq \rho \\ 0 & x \leq \rho \\ x + \rho & x \leq -\rho \end{cases}$

[R3] Parikh N and Boyd S (2014). "Proximal algorithms", Foundations and Trends in Optimization.

The special case of the indicator function

One of the main advantages of this formulation is that it allows us to deal with constrained optimization problems.

Let the indicator function be denoted by

$$\iota_{\mathcal{X}}(x) := \begin{cases} 0 & \text{if } x \in \mathcal{X} \\ +\infty & \text{otherwise.} \end{cases}$$

Then, when $g = \iota_X$, the optimization problem is equivalent to a constrained optimization problem:

$$\min_{x \in \mathcal{X}} f(x) = \min_{x \in \mathbb{R}^p} f(x) + \iota_{\mathcal{X}}(x).$$

Moreover, the proximal operator of the indicator function reduces to the Euclidean projection onto $\mathcal{X}$.

Example 12.25 in [R1]

Let $\mathcal{X}$ be a nonempty, closed, convex subset of $\mathbb{R}^p$. Then:

$$\text{prox}_{\ell_{\mathcal{X}}}^{\rho} \equiv \text{proj}_{\mathcal{X}},$$

where $\text{proj}_{\mathcal{X}}$ denotes the projection operator onto $\mathcal{X}$.

Some useful closed forms of projections

These formulas are drawn from Section 6 in [R3].

$$\text{Affine:} \quad \mathcal{X} = \{x \in \mathbb{R}^n : Ax = b\} \quad \text{proj}_{\mathcal{X}}(x) = x - A^\dagger(Ax - b)$$

$$\text{Hyperplane:} \quad \mathcal{X} = \{x \in \mathbb{R}^n : a^\top x = b\} \quad \text{proj}_{\mathcal{X}}(x) = x + \frac{b - a^\top x}{\|a\|^2} a$$

$$\text{Halfspace:} \quad \mathcal{X} = \{x \in \mathbb{R}^n : a^\top x \leq b\} \quad \text{proj}_{\mathcal{X}}(x) = x - \frac{\max\{0, a^\top x - b\}}{\|a\|^2} a$$

$$\text{Box:} \quad \mathcal{X} = \{x \in \mathbb{R}^n : \ell \leq x \leq u\} \quad (\text{proj}_{\mathcal{X}}(x))_i = \begin{cases} \ell_i & x_i \leq \ell_i \\ x_i & \ell_i \leq x_i \leq u_i \\ u_i & x_i \geq u_i \end{cases} \quad (\text{component-wise})$$

[R3] Parikh N and Boyd S (2014). "Proximal algorithms", Foundations and Trends in Optimization.

Outline

- 1 Introduction
- 2 Operator theory for open systems
- 3 The gradient descent (GD) algorithm
- 4 The proximal gradient (PG) algorithm
- 5 The distributed GD (DGD) algorithm**
- 6 The DGD in open multi-agent systems
- 7 Related results in the current state-of-the-art

Consider a network of $n \in \mathbb{N}$ agents connected according to a graph $\mathcal{G} = (\mathcal{V}, \mathcal{E})$. Each agent $i \in \mathcal{V}$ has a local cost function $f_i : \mathbb{R}^p \rightarrow \mathbb{R}$, and the goal is to minimize the sum of all local cost functions:

$$\min_{x \in \mathbb{R}^p} \sum_{i \in \mathcal{V}} f_i(x).$$

This problem can be equivalently written in a distributed way as follows:

$$\begin{aligned} \min_{\{x_i\}_{i \in \mathcal{V}}} \quad & \sum_{i \in \mathcal{V}} f_i(x_i) \\ \text{s.t.} \quad & x_i = x_j, \quad \forall (i, j) \in \mathcal{E} \text{ (consensus constraints)} \end{aligned}$$

where $x_i \in \mathbb{R}^p$ is agent i 's local copy of the decision variable. Note that, when the graph is connected, any solution to the above problem makes all agents agree on the same value, i.e., they achieve consensus, as encoded by the constraints $x_1 = \dots = x_n$.

We can rewrite the consensus optimization problem as a composite problem as follows:

$$\min_{x \in \mathbb{R}^{np}} f(x) + g(x)$$

where $x = [x_1^\top, \dots, x_n^\top]^\top$ is the vector stacking all the agents' states, and

$$f(x) = \sum_{i \in \mathcal{V}} f_i(x_i), \quad g(x) = \iota_{\mathcal{X}}(x) := \begin{cases} 0 & \text{if } x \in \mathcal{X} = \{x \in \mathbb{R}^{np} : x_i = x_j, \forall (i, j) \in \mathcal{E}\}, \\ +\infty & \text{otherwise,} \end{cases}$$

where $\iota_{\mathcal{X}}$ is the indicator function. We note that:

- $f(x)$ is convex as long as the local costs $f_i(x_i)$ are convex;
- $g(x)$ is convex and lower semicontinuous if the graph is connected. Indeed, graph connectedness implies that the constraint set is equal to

$$\mathcal{X} = \{\mathbf{1}_n \otimes z : z \in \mathbb{R}^p\},$$

This is a closed linear subspace. Therefore, $\mathcal{X}$ is convex (so g is convex, see Example 8.3 in [R1]) and closed (so g is lower semicontinuous, see Example 1.25 in [R1]);

- $f(x) + g(x)$ is convex as long as both f and g are convex.

Therefore, in principle, we could apply the proximal gradient algorithm to solve the problem:

$$x(k+1) = \text{prox}_g^\rho(x(k) - \rho \nabla f(x(k)))$$

In our scenario, the consensus constraint set is affine with $A = I_{np} - \frac{1}{n} \mathbf{1}_n \mathbf{1}_n^\top \otimes I_p$ and $b = \mathbf{0}$. Moreover:

- A is idempotent, i.e., $A^2 = A$;
- A is symmetric, i.e., $A^\top = A$;
- The pseudoinverse satisfies $A^\dagger = A$.

Thus, the projection becomes:

$$\text{proj}_{\mathcal{X}}(x) = x - A^\dagger(Ax - b) = x - Ax = \frac{1}{n} (\mathbf{1}_n \mathbf{1}_n^\top \otimes I_p) x$$

The iteration becomes:

$$x(k+1) = \text{proj}_{\mathcal{X}}(x(k) - \rho \nabla f(x(k))) = \frac{1}{n} (\mathbf{1}_n \mathbf{1}_n^\top \otimes I_p) (x(k) - \rho \nabla f(x(k))),$$

which, component-wise, can be written as follows:

$$x_i(k+1) = \frac{1}{n} \sum_{j=1}^n (x_j(k) - \rho \nabla f_j(x_j(k))),$$

Is this okay?

The iteration obtained by applying the proximal gradient algorithm to the distributed problem,

$$x_i(k+1) = \frac{1}{n} \sum_{j=1}^n (x_j(k) - \rho \nabla f_j(x_j(k))),$$

requires that each node $i \in \mathcal{V}$ knows:

- the state $x_j(k)$ of all agents in the network (complete graph);
- the gradient $\nabla f_j(x_j(k))$ of all agents in the network (privacy issues);
- the number n of agents in the network (usually unknown, especially in open MAS).

This means that the above iteration **is not really distributed**. Indeed, the proximal operator of the indicator function on the consensus set corresponds to a projection onto the set, i.e., to the average of the agents' updates $x_i(k) - \rho \nabla f_i(x_i(k))$.

In other words, it would require a complete communication graph.

How can we solve this issue?

Issue 1 (complete graph and knowledge of n). The projection operation, i.e., the average of all the agents' states, can be replaced with distributed averaging over neighborhoods, namely:

$$x_i(k+1) = \sum_{j \in \mathcal{N}_i \cup \{i\}} w_{ij} (x_j(k) - \rho \nabla f_j(x_j(k))), \quad \text{where} \quad w_{ij} = \begin{cases} \varepsilon & \text{if } j \in \mathcal{N}_i, \\ 1 - \varepsilon |\mathcal{N}_i| & \text{if } j = i, \\ 0 & \text{otherwise,} \end{cases}$$

which, in compact form, can be written as follows:

$$x(k+1) = ((I - \varepsilon \mathcal{L}) \otimes I_p) (x(k) - \rho \nabla f(x(k))),$$

where $\mathcal{L}$ is the Laplacian matrix; generally, $I - \varepsilon \mathcal{L}$ can be replaced by any doubly stochastic matrix. This is called the **adapt-then-combine (ATC) algorithm** [R4].

[R4] Chen J and Sayed A.H. (2013). "Distributed Pareto Optimization via Diffusion Strategies", IEEE Journal of Selected Topics in Signal Processing.

Issue 2 (exchange of gradients). The neighborhood average gradient can be replaced with the local gradient,

$$x_i(k+1) = \sum_{j \in \mathcal{N}_i \cup \{i\}} w_{ij} x_j(k) - \rho \nabla f_i(x_i(k)),$$

which, in compact form, can be written as follows:

$$x(k+1) = ((I - \varepsilon \mathcal{L}) \otimes I_p) x(k) - \rho \nabla f(x(k)).$$

This is called the **distributed gradient descent (DGD) algorithm** [R5].

[R5] Nedic A and Ozdaglar A. (2009). "Distributed Subgradient Methods for Multi-Agent Optimization", IEEE Transactions on Automatic Control.

The main issue: Average consensus is a poor substitute for a full projection onto the consensus set, in the sense that their fixed points do not coincide with solutions of the original problem. Indeed:

$$\begin{aligned}
 x^* &= \sum_{j \in \mathcal{N}_i \cup \{i\}} w_{ij} x^* - \rho \nabla f_i(x^*) \quad \forall x^* \in \mathcal{X}^* \\
 &\iff \\
 0 &= \nabla f_i(x^*) \quad (\text{false in general})
 \end{aligned}$$

What is the actual set of fixed points? It can be computed as follows:

$$\begin{aligned}
 \hat{x} &= ((I - \varepsilon \mathcal{L}) \otimes I_p) \hat{x} - \rho \nabla f(\hat{x}) \\
 &\iff (I_{np} - (I - \varepsilon \mathcal{L}) \otimes I_p) \hat{x} + \rho \nabla f(\hat{x}) = 0 \\
 &\iff (\varepsilon \mathcal{L} \otimes I_p) \hat{x} + \rho \nabla f(\hat{x}) = 0.
 \end{aligned}$$

Hence,

$$\hat{\mathcal{X}} := \text{fix}(\mathbf{T}) = \{x \in \mathbb{R}^{np} : (\varepsilon \mathcal{L} \otimes I_p)x + \rho \nabla f(x) = 0\}.$$

Another interpretation. The DGD algorithm solves the following penalized (i.e., different) problem:

$$\min_{\{x_i\}_{i \in \mathcal{V}}} \sum_{i \in \mathcal{V}} f_i(x_i) + \frac{\varepsilon}{2\rho} x^\top (\mathcal{L} \otimes I_p) x, \quad \text{with} \quad x = [x_1^\top, \dots, x_n^\top]^\top.$$

This means that:

- The agents do not, in general, converge to the optimal solution;
- The agents do not, in general, converge to a consensus state.

The DGD algorithm: a numerical example

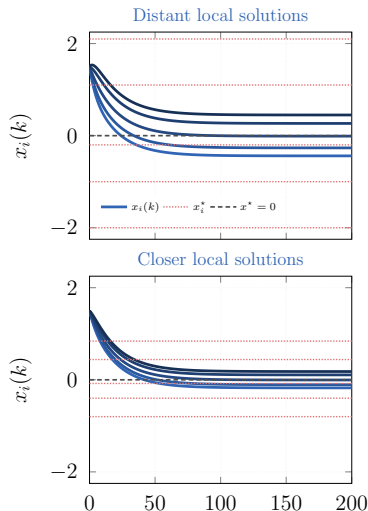
Local functions: $f_i(x) = \frac{1}{2}(x - a_i)^2$

- $n = 5$ agents on a path graph
- $\lambda_{\max}(\mathcal{L}) = 2(1 + \cos(\pi/n)) \approx 3.62$
- $x_i(0) = 45$
- $\|x(0) - \mathbf{1}_n x^*\| \approx 10^2$
- $m = L = 1$
- $\rho = 0.05$, $\varepsilon = 0.45 < 2/\lambda_{\max}(\mathcal{L})$
- $x_i^* = a_i$ (top), $x_i^* = 0.4 \cdot a_i$ (bottom)
- $x^* = \frac{1}{n} \sum_{i \in \mathcal{V}} a_i = 0$ in both cases

Two cases:

- Top: distant local solutions
- Bottom: same average, local solutions closer to x^*

Takeaway: the bias bound scales with the **dispersion** across local solutions.



Theorem (Distributed Gradient Descent Algorithm)

Consider the distributed optimization problem

$$\begin{aligned} \min_{\{x_i\}_{i \in \mathcal{V}}} \quad & \sum_{i \in \mathcal{V}} f_i(x_i) =: f(x) \\ \text{s.t.} \quad & x_i = x_j, \quad \forall (i, j) \in \mathcal{E} \end{aligned}$$

Assume that the graph is connected and that the functions $f_i : \mathbb{R}^p \rightarrow \mathbb{R}$ are **m-strongly convex**, differentiable, and L -smooth. Then, the distributed GD iteration

$$x_i(k+1) = \sum_{j \in \mathcal{N}_i \cup \{i\}} w_{ij} x_j(k) - \rho \nabla f_i(x_i(k)), \quad \text{where} \quad w_{ij} = \begin{cases} \varepsilon & \text{if } j \in \mathcal{N}_i, \\ 1 - \varepsilon |\mathcal{N}_i| & \text{if } j = i, \\ 0 & \text{otherwise,} \end{cases}$$

converges **at a linear rate** to the unique minimizer $\hat{x}$ of the penalized objective $f(x) + \frac{\varepsilon}{2\rho} x^\top (\mathcal{L} \otimes I_p) x$ if $\rho \in (0, (2 - \varepsilon \lambda_{\max}(\mathcal{L}))/L)$ and $\varepsilon < 2/\lambda_{\max}(\mathcal{L})$, where $\mathcal{L}$ is the Laplacian matrix, i.e.,

$$\|x(k) - \hat{x}\| \leq \ell^k \|x(0) - \hat{x}\|, \quad \ell = \max\{|1 - \rho m|, |1 - \rho L - \varepsilon \lambda_{\max}(\mathcal{L})|\} < 1,$$

where $\hat{x}$ is not, in general, equal to the consensus state $\mathbf{1}_n \otimes x^*$ at a minimizer x^* of the centralized objective. Nevertheless, the distance is bounded by $\|\hat{x} - \mathbf{1}_n \otimes x^*\| \leq \frac{L}{m} \sqrt{\sum_{i \in \mathcal{V}} \|x^* - x_i^*\|^2}$, with $x_i^* := \arg\min_{x \in \mathbb{R}^p} f_i(x)$.

The distributed gradient descent algorithm: proof for strongly convex functions

Proof (via recasting as a GD iteration):

Step 1: the DGD iteration is exactly the GD iteration applied to the penalized problem:

$$\min_{\{x_i\}_{i \in \mathcal{V}}} \sum_{i \in \mathcal{V}} f_i(x_i) + \frac{\varepsilon}{2\rho} x^\top (\mathcal{L} \otimes I_p) x =: F(x).$$

Indeed, $x(k+1) = ((I - \varepsilon \mathcal{L}) \otimes I_p) x(k) - \rho \nabla f(x(k)) = x(k) - \rho \underbrace{\left[\nabla f(x(k)) + \frac{\varepsilon}{\rho} (\mathcal{L} \otimes I_p) x(k) \right]}_{= \nabla F(x(k))} =: T(x(k)).$

Step 2: strong convexity of the penalized objective. Since each $f_i(x_i)$ is m -strongly convex, their sum $\sum_{i \in \mathcal{V}} f_i(x_i)$ is also m -strongly convex. Moreover, $\frac{\varepsilon}{2\rho} x^\top (\mathcal{L} \otimes I_p) x$ is convex (i.e., 0-strongly convex) because $\mathcal{L}$ is positive semidefinite for undirected graphs. Hence, the sum $F(x)$ of an m -strongly convex function and a convex function is still m -strongly convex.

Step 3: smoothness of the penalized objective. Since each f_i is L -smooth, their sum $\sum_{i \in \mathcal{V}} f_i(x_i)$ is also L -smooth. Moreover, the smoothness constant of the quadratic penalty is $L_q = \frac{\varepsilon}{\rho} \lambda_{\max}(\mathcal{L})$ because $\left\| \frac{\varepsilon}{\rho} (\mathcal{L} \otimes I_p)(x - y) \right\| \leq \frac{\varepsilon}{\rho} \|\mathcal{L} \otimes I_p\| \|x - y\| = \frac{\varepsilon}{\rho} \|\mathcal{L}\| \|x - y\| = \frac{\varepsilon}{\rho} \lambda_{\max}(\mathcal{L}) \|x - y\|$. Hence, the sum $F(x)$ of an L -smooth function and a L_q -smooth function is $L' = (L + L_q)$ -smooth.

Step 4: convergence. Applying the strongly convex GD theorem to F gives linear convergence if $0 < \rho < 2/L'$, or equivalently $\rho L + \varepsilon \lambda_{\max}(\mathcal{L}) < 2$. Thus, if $\varepsilon < 2/\lambda_{\max}(\mathcal{L})$ and $\rho \in (0, (2 - \varepsilon \lambda_{\max}(\mathcal{L}))/L)$, then convergence occurs linearly at rate $\ell = \max\{|1 - \rho m|, |1 - \rho L - \varepsilon \lambda_{\max}(\mathcal{L})|\}$.

Additional result: distance from the consensus optimizer. Let $z^* := \mathbf{1}_n \otimes x^*$, where x^* minimizes $\sum_{i \in \mathcal{V}} f_i(x)$. Since $(\mathcal{L} \otimes I_p)z^* = 0$, we have $\nabla F(z^*) = \nabla f(z^*)$. Moreover, by definition $\nabla F(\hat{x}) = 0$.

Thus, by also using m -strong convexity of F , which ensures that ∇F is m -strongly monotone, we get

$$m\|\hat{x} - z^*\|^2 \leq (\nabla F(\hat{x}) - \nabla F(z^*))^\top (\hat{x} - z^*) \stackrel{(*)}{\leq} \|\nabla F(\hat{x}) - \nabla F(z^*)\| \|\hat{x} - z^*\|,$$

where $(*)$ holds by the Cauchy–Schwarz inequality. If $\hat{x} \neq z^*$, dividing by $\|\hat{x} - z^*\|$ gives

$$m\|\hat{x} - z^*\| \leq \|\nabla F(\hat{x}) - \nabla F(z^*)\| = \|\nabla F(z^*)\| = \|\nabla f(z^*)\|.$$

Moreover, since $\nabla f_i(x_i^*) = 0$ and f_i is L -smooth,

$$\|\nabla f(z^*)\| = \sqrt{\sum_{i \in \mathcal{V}} \|\nabla f_i(x^*)\|^2} \leq L \sqrt{\sum_{i \in \mathcal{V}} \|x^* - x_i^*\|^2}.$$

Therefore,

$$\|\hat{x} - \mathbf{1}_n \otimes x^*\| \leq \frac{L}{m} \sqrt{\sum_{i \in \mathcal{V}} \|x^* - x_i^*\|^2}.$$

The DGD algorithm: distance from the optimizer

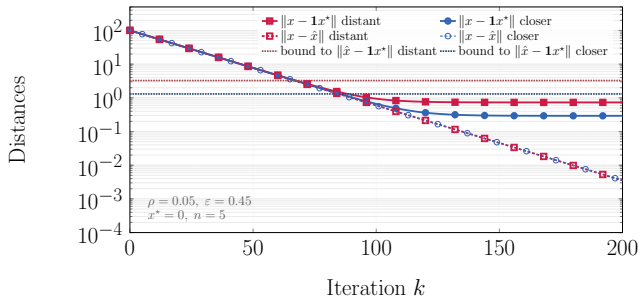
Two distances, lifted to the whole network state:

$$\|x(k) - \mathbf{1}_n x^*\| \quad \text{and} \quad \|x(k) - \hat{x}\|.$$

- $x^* = 0$ is the centralized optimizer
- $\hat{x}$ minimizes the penalized problem
- Dotted lines: upper bounds on $\|\hat{x} - \mathbf{1}_n x^*\|$
- Bound: $\frac{L}{m} \sqrt{\sum_i \|x^* - x_i^*\|^2}$

Interpretation:

- DGD approaches a biased fixed point
- $\|x(k) - \hat{x}\| \rightarrow 0$
- Smaller heterogeneity gives a smaller plateau



How to make the convergence conditions independent of spectral quantities

For an undirected graph, let

$$\Delta := \max_{i \in \mathcal{V}} |\mathcal{N}_i|.$$

Then

$$\lambda_{\max}(\mathcal{L}) \leq 2\Delta.$$

A sufficient spectral-free condition is therefore $\rho L + 2\varepsilon\Delta < 2$, namely

$$0 < \rho < \frac{2 - 2\varepsilon\Delta}{L}, \quad 0 < \varepsilon < \frac{1}{\Delta}.$$

Moreover,

$$0 < \rho < \frac{1}{L}, \quad 0 < \varepsilon < \frac{1}{2\Delta}.$$

Outline

- 1 Introduction
- 2 Operator theory for open systems
- 3 The gradient descent (GD) algorithm
- 4 The proximal gradient (PG) algorithm
- 5 The distributed GD (DGD) algorithm
- 6 The DGD in open multi-agent systems**
- 7 Related results in the current state-of-the-art

Consider an **open network** of $n_k \in \mathbb{N}$ agents, with $n_k = |\mathcal{V}_k|$, interacting according to a **time-varying graph** $\mathcal{G}_k = (\mathcal{V}_k, \mathcal{E}_k)$, connected at every k . Each agent $i \in \mathcal{V}_k$ has a local cost function $f_i : \mathbb{R}^p \rightarrow \mathbb{R}$, and the goal is to minimize the sum of all local cost functions:

$$\begin{aligned} \min_{\{x_i\}_{i \in \mathcal{V}_k}} \quad & \sum_{i \in \mathcal{V}_k} f_i(x_i) \\ \text{s.t.} \quad & x_i = x_j, \quad \forall (i, j) \in \mathcal{E}_k \text{ (consensus constraints)} \end{aligned}$$

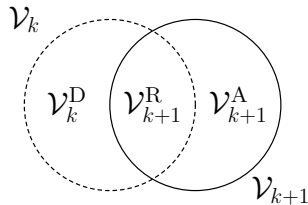
where $x_i \in \mathbb{R}^p$ is agent i 's local copy of the decision variable.

Question: What happens to the optimal solution?

Answer: It may be different at any time step k .

It is useful to identify the following subsets:

- **Remaining agents** $\mathcal{V}_{k+1}^R = \mathcal{V}_{k+1} \cap \mathcal{V}_k$: agents present both at time k and time $k+1$;
- **Arriving agents** $\mathcal{V}_{k+1}^A = \mathcal{V}_{k+1} \setminus \mathcal{V}_k$: agents present at time $k+1$ but not at time k ;
- **Departing agents** $\mathcal{V}_k^D = \mathcal{V}_k \setminus \mathcal{V}_{k+1}$: agents present at time k but not at time $k+1$.



We formalize our assumptions, which use the following notation for the set of solutions to the distributed optimization problem:

$$\mathcal{X}_k^* = \left\{ x_k^* \in \mathbb{R}^p : \sum_{i \in \mathcal{V}_k} f_i(x_k^*) = \min_{x \in \mathbb{R}^p} \sum_{i \in \mathcal{V}_k} f_i(x) \right\},$$

and for the minimizers of each local cost function,

$$\mathcal{X}_{i,k}^* = \left\{ x_{i,k}^* \in \mathbb{R}^p : f_i(x_{i,k}^*) = \min_{x \in \mathbb{R}^p} f_i(x) \right\}, \quad \forall i \in \mathcal{V}_k. \quad (11)$$

Main assumptions. For any $k \in \mathbb{N}$:

- (i) the local cost functions f_i are proper, lower semicontinuous, m -strongly convex, and L -smooth for all $i \in \mathcal{V}_k$, so that $\mathcal{X}_{i,k}^* = \{x_{i,k}^*\}$ and $\mathcal{X}_k^* = \{x_k^*\}$ are singletons;
- (ii) the distance between two consecutive global solutions x_k^* and x_{k+1}^* is upper bounded by a constant $\sigma \geq 0$;
- (iii) the distance between each local solution $x_{i,k}^*$ and the global solution x_k^* is upper bounded by $\omega \geq 0$.

DGD in open multi-agent systems can be written as follows:

$$x_i(k+1) = \begin{cases} \sum_{j \in \mathcal{N}_{i,k} \cup \{i\}} w_{ij}(k) x_j(k) - \rho \nabla f_i(x_i(k)) & \text{if } i \in \mathcal{V}_{k+1}^R, \\ x_{i,k+1}^A & \text{if } i \in \mathcal{V}_{k+1}^A, \end{cases}$$

where $w_{ii}(k) = 1 - \varepsilon |\mathcal{N}_{i,k}|$ and $w_{ij}(k) = \varepsilon$ if $j \in \mathcal{N}_{i,k}$.

Design: Several choices are possible for initializing the state $x_{i,k+1}^A$ of arriving agents:

- **Initialization to local minimizer:** $x_{i,k+1}^A = x_{i,k+1}^*$ (our choice);
- Blind initialization: $x_{i,k+1}^A = \mathbf{0}$;
- Initialization to the average of neighboring states $x_{i,k+1}^A = \frac{1}{|\mathcal{N}_{i,k}|} \sum_{j \in \mathcal{N}_{i,k}} x_j(k)$ (this requires that the arriving agents can communicate with neighbors before joining the network);
- Initialization within a bounded set $x_{i,k+1}^A \in \overline{\mathcal{X}}$ (e.g., in the case of the projected DGD);
- Other choices.

Special case of Theorem 1 in [R2]

Consider an OMAS and assume that

- (a) the standard iteration is paracontractive with $\gamma \in (0, 1)$;
- (b) the TPI has bounded variation $B \geq 0$, i.e., $d(\hat{x}_{k+1}, \hat{x}_k)/\sqrt{p|\mathcal{V}_{k+1}|} \leq B$;
- (c) the arrival process is bounded with $H \geq 0$, i.e., $d(x_{k+1}^A, \hat{x}_{k+1})/\sqrt{p|\mathcal{V}_{k+1}|} \leq H$;
- (d) the departure process is bounded with $\beta \in (\gamma, 1)$, i.e., $\sqrt{|\mathcal{V}_{k+1}|}/\sqrt{|\mathcal{V}_k|} \geq \beta$.

Then, the open sequence $\{x(k)\}_{k \in \mathbb{N}}$ converges linearly with rate $\theta = \gamma/\beta \in (0, 1)$ to the TPI within a radius

$$R = \frac{B + H}{1 - \theta}, \quad (12)$$

according to the following pointwise upper bound

$$\frac{d(x(k), \hat{x}_k)}{\sqrt{p|\mathcal{V}_k|}} \leq \theta^k \frac{d(x(0), \hat{x}_0)}{\sqrt{p|\mathcal{V}_0|}} + \frac{1 - \theta^k}{1 - \theta} (B + H).$$

[R2] D. Deplano, N. Bastianello, M. Franceschelli, K.H. Johansson (2026). Optimization and learning in Open Multi-Agent Systems, IEEE Transactions on Automatic Control.

Proof of condition (a) (the standard iteration is paracontractive with $\gamma \in (0, 1)$).

The standard iteration is simply the DGD, which we have already proved to be contractive with constant $\ell_k = \max\{|1 - \rho m|, |1 - \rho L - \varepsilon \lambda_{\max}(\mathcal{L}_k)|\}$.

Remark. To obtain a uniform upper bound $\ell < 1$ on the contraction factor ℓ_k , we need a uniform upper bound $\bar{\lambda}$ on the largest eigenvalue $\lambda_{\max}$ of the Laplacian matrix. A reasonable sufficient condition is that each agent has at most $\Delta > 0$ neighbors, because it implies $\lambda_{\max}(\mathcal{L}_k) \leq 2\Delta =: \bar{\lambda}$. Given the existence of such $\bar{\lambda} > 0$, under the stepsize conditions $0 < \varepsilon < 2/\bar{\lambda}$ and $0 < \rho < (2 - \varepsilon\bar{\lambda})/L$, one obtains the uniform bound

$$\ell = \max\{|1 - \rho m|, |1 - \rho L|, |1 - \rho L - \varepsilon \bar{\lambda}|\} < 1,$$

with $\ell_k \leq \ell$ for all $k \in \mathbb{N}$.

Conclusion: Thus, since contractivity implies paracontractivity, the standard iteration is also paracontractive with $\gamma = \ell$.

Proof of condition (b) (the TPI has bounded variation $B \geq 0$, i.e., $d(\hat{x}_{k+1}, \hat{x}_k)/\sqrt{p|\mathcal{V}_{k+1}|} \leq B$).

Since the standard iteration is contractive, at each step k there is a unique fixed point $\hat{x}_k$, and thus the OMAS admits a TPI $\{\hat{x}_k\}_{k \in \mathbb{N}}$. Moreover:

$$\begin{aligned}
 d(\hat{x}_{k+1}, \hat{x}_k) &= \sqrt{\sum_{i \in \mathcal{V}_{k+1}^R} d^2(\hat{x}_{i,k+1}, \hat{x}_{i,k})} \stackrel{(a)}{\leq} \sqrt{\sum_{i \in \mathcal{V}_{k+1}^R} (d(\hat{x}_{i,k+1}, x_{k+1}^*) + d(x_{k+1}^*, x_k^*) + d(\hat{x}_{i,k}, x_k^*))^2} \\
 &\stackrel{(b)}{\leq} \sqrt{\sum_{i \in \mathcal{V}_{k+1}} d^2(\hat{x}_{i,k+1}, x_{k+1}^*)} + \sqrt{\sum_{i \in \mathcal{V}_{k+1}^R} d^2(x_{k+1}^*, x_k^*)} + \sqrt{\sum_{i \in \mathcal{V}_k} d^2(\hat{x}_{i,k}, x_k^*)} \\
 &\stackrel{(c)}{\leq} \frac{L}{m} \sqrt{\sum_{i \in \mathcal{V}_{k+1}} d^2(x_{k+1}^*, x_{i,k+1}^*)} + \sqrt{|\mathcal{V}_{k+1}^R|} d(x_{k+1}^*, x_k^*) + \frac{L}{m} \sqrt{\sum_{i \in \mathcal{V}_k} d^2(x_k^*, x_{i,k}^*)} \\
 &\stackrel{(d)}{\leq} \sqrt{|\mathcal{V}_{k+1}|} \frac{L\omega}{m} + \sqrt{|\mathcal{V}_{k+1}|} \sigma + \sqrt{|\mathcal{V}_k|} \frac{L\omega}{m} \stackrel{(e)}{\leq} \sqrt{p|\mathcal{V}_{k+1}|} \underbrace{\frac{1}{\sqrt{p}} \left(\frac{L\omega}{m} + \sigma + \frac{L\omega}{m\beta} \right)}_B
 \end{aligned}$$

where (a) holds by the triangle inequality; (b) holds by Minkowski's inequality and by $\mathcal{V}_{k+1}^R \subseteq \mathcal{V}_{k+1}$, $\mathcal{V}_{k+1}^R \subseteq \mathcal{V}_k$; (c) holds by the previously computed bound on the distance between a fixed point $\hat{x}_k$ and the consensus state at the minimizer x_k^* ; (d) holds by assumption (ii) bounding $d(x_{k+1}^*, x_k^*) \leq \sigma$ and by assumption (iii) bounding $d(x_{i,k}^*, x_k^*) \leq \omega$; (e) holds by the boundedness of the departure process.

Proof of condition (c) (the arrival process is bounded with $H \geq 0$, i.e., $d(x_{k+1}^A, \hat{x}_{k+1})/\sqrt{p|\mathcal{V}_{k+1}|} \leq H$).

Under our initialization choice $x_{i,k+1}^A = x_{i,k+1}^*$, we have:

$$\begin{aligned}
 d(x_{k+1}^A, \hat{x}_{k+1}) &= \sqrt{\sum_{i \in \mathcal{V}_{k+1}^A} d^2(x_{i,k+1}^A, \hat{x}_{i,k+1})} = \sqrt{\sum_{i \in \mathcal{V}_{k+1}^A} d^2(x_{i,k+1}^*, \hat{x}_{i,k+1})} \\
 &\leq \sqrt{\sum_{i \in \mathcal{V}_{k+1}^A} \left(d(x_{k+1}^*, x_{i,k+1}^*) + d(x_{k+1}^*, \hat{x}_{i,k+1}) \right)^2} \\
 &\leq \sqrt{\sum_{i \in \mathcal{V}_{k+1}^A} d^2(x_{k+1}^*, x_{i,k+1}^*)} + \sqrt{\sum_{i \in \mathcal{V}_{k+1}^A} d^2(x_{k+1}^*, \hat{x}_{i,k+1})} \\
 &\leq \sqrt{\sum_{i \in \mathcal{V}_{k+1}} d^2(x_{k+1}^*, x_{i,k+1}^*)} + \sqrt{\sum_{i \in \mathcal{V}_{k+1}} d^2(x_{k+1}^*, \hat{x}_{i,k+1})} \leq \left(1 + \frac{L}{m}\right) \sqrt{\sum_{i \in \mathcal{V}_{k+1}} d^2(x_{k+1}^*, x_{i,k+1}^*)} \\
 &\leq \sqrt{|\mathcal{V}_{k+1}|} \left(1 + \frac{L}{m}\right) \omega = \underbrace{\sqrt{p|\mathcal{V}_{k+1}|} \frac{1}{\sqrt{p}} \left(1 + \frac{L}{m}\right) \omega}_H.
 \end{aligned}$$

Consider the distributed optimization problem over an open network

$$\min_{\{x_i\}_{i \in \mathcal{V}_k}} \sum_{i \in \mathcal{V}_k} f_i(x_i) =: f_k(x)$$

Assume that each graph $\mathcal{G}_k$ is connected and that the functions $f_i : \mathbb{R}^P \rightarrow \mathbb{R}$ are proper, lower semicontinuous, m -strongly convex, differentiable, and L -smooth. Consider the distributed open iteration

$$x_i(k+1) = \begin{cases} \sum_{j \in \mathcal{N}_{i,k} \cup \{i\}} w_{ij}(k)x_j(k) - \rho \nabla f_i(x_i(k)) & \text{if } i \in \mathcal{V}_{k+1}^{\text{R}}, \\ x_{i,k+1}^* & \text{if } i \in \mathcal{V}_{k+1}^{\text{A}}, \end{cases}$$

where $w_{ii}(k) = 1 - \varepsilon |\mathcal{N}_{i,k}|$ and $w_{ij}(k) = \varepsilon$ if $j \in \mathcal{N}_{i,k}$. Assume also that $\lambda_{\max}(\mathcal{L}_k) \leq \bar{\lambda}$ and that the departure process is bounded with $\beta \in (\gamma, 1)$ for all $k \in \mathbb{N}$, where $\gamma = \max\{|1 - \rho m|, |1 - \rho L|, |1 - \rho L - \varepsilon \bar{\lambda}|\} < 1$. If $\rho \in (0, (2 - \varepsilon \bar{\lambda})/L)$ and $\varepsilon < 2/\bar{\lambda}$, then the open sequence $\{x(k)\}_{k \in \mathbb{N}}$ converges at a linear rate $\theta = \gamma/\beta$ to the trajectory $\{\hat{x}_k\}_{k \in \mathbb{N}}$ of unique minimizers of the penalized objective $f_k(x) + \frac{\varepsilon}{2\rho} x^\top (\mathcal{L}_k \otimes I_p) x$ within radius R ,

$$\frac{\|x(k) - \hat{x}_k\|}{\sqrt{p|\mathcal{V}_k|}} \leq \theta^k \frac{\|x(0) - \hat{x}_0\|}{\sqrt{p|\mathcal{V}_0|}} + \underbrace{\frac{B+H}{1-\theta}}_R, \quad \text{with} \quad \begin{aligned} B &= \frac{1}{\sqrt{p}} \left(\frac{L\omega}{m} + \sigma + \frac{L\omega}{m\beta} \right), \\ H &= \frac{1}{\sqrt{p}} \left(1 + \frac{L}{m} \right) \omega. \end{aligned}$$

Application: Open Proportional Dynamic Consensus (OPDC)

Fact. The OPDC protocol [R8] is a special case of DGD with quadratic local costs.

OPDC update (for remaining agents $i \in \mathcal{V}_{k+1}^R$):

$$x_i(k+1) = x_i(k) - \alpha(x_i(k) - u_i(k)) - \varepsilon \sum_{j \in \mathcal{N}_{i,k}} (x_i(k) - x_j(k)),$$

where $u_i(k) \in \mathbb{R}$ is a reference signal and $\alpha \in (0, 0.5)$.

Identification with DGD. Set $\rho = 1$ and choose the local cost

$$f_i(x_i) = \frac{\alpha}{2}(x_i - u_i(k))^2 \quad \implies \quad \nabla f_i(x_i) = \alpha(x_i - u_i(k)).$$

Then the DGD iteration $x_i(k+1) = \sum_{j \in \mathcal{N}_{i,k} \cup \{i\}} w_{ij} x_j(k) - \rho \nabla f_i(x_i(k))$ becomes exactly the OPDC update.

Inherited properties:

- $m = L = \alpha \implies \kappa = L/m = 1$;
- Contraction rate: $\ell = \max\{|1 - \alpha|, |1 - \alpha - \varepsilon\bar{\lambda}|\} = 1 - \alpha$, matching [R8];
- Minimizer of f_i : $x_i^* = u_i(k)$ (the reference signal itself);
- The stability radius R result applies directly.

[R8] M. Franceschelli and P. Frasca (2021). Stability of Open Multiagent Systems and Applications to Dynamic Consensus, IEEE Transactions on Automatic Control.

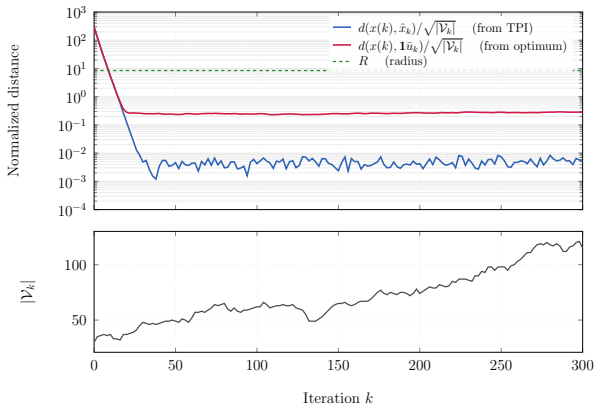
OPDC simulation: convergence to the TPI

Setup (ring graph, static inputs):

- $n(0) = 30$ agents, $u_i \sim \text{Uniform}[0, 1]$
- $\alpha = 0.3$, $\varepsilon = 0.01$, $\rho = 1$
- Arrive rate $\approx 2\%$, depart rate $\approx 1.5\%$
- Init: $x_i(0) = u_i + \text{Uniform}[-500, 500]$
- Parameters: $\gamma = 1 - \alpha = 0.7$, $\beta = 0.95$, $\theta = \gamma/\beta \approx 0.74$, $\omega = 0.54$, $\sigma = 0.04$.

Key observations:

- The radius ($R \approx 8.5$) is quite conservative, while the actual distances settle at $\approx 5 \cdot 10^{-3}$ (from the TPI) and $\approx 2.5 \cdot 10^{-1}$ (from the optimum);
- Linear convergence is visible during the initial transient;
- The gap between the two curves is the **DGD bias** (distance between the TPI and the true consensus optimizer).



Exercise: the projected DGD in OMAS

Consider the projected DGD:

$$x_i(k+1) = \begin{cases} \text{proj}_{\mathcal{X}} \left(\sum_{j \in \mathcal{N}_{i,k} \cup \{i\}} w_{ij}(k) x_j(k) - \rho \nabla f_i(x_i(k)) \right) & i \in \mathcal{V}_{k+1}^{\text{R}}, \\ x_{i,k+1}^{\text{A}} \in \mathcal{X} & i \in \mathcal{V}_{k+1}^{\text{A}}, \end{cases}$$

where $\mathcal{X} \subseteq \mathbb{R}^p$ is a closed, convex, and bounded set, and f_i are m -strongly convex and L -smooth.

Exercise (30 min): Apply the Special Case of Theorem 1 in [R2] to prove linear convergence of the above algorithm. In particular:

- Is the standard iteration still paracontractive? Explain why and discuss if there is the need of any extra assumptions or, vice versa, if some can be relaxed.
- Does the projection change the TPI? In either case, can you still prove its boundedness? Explain why and discuss if there is the need of any extra assumptions or, vice versa, if some can be relaxed.
- Is the arrival process bounded? Explain why and discuss if there is the need of any extra assumptions or, vice versa, if some can be relaxed.

Hint. You can use ChatGPT, Claude, or other generative AI. The important thing is that you can write the precise steps and be able to explain in your words.

Outline

- 1 Introduction
- 2 Operator theory for open systems
- 3 The gradient descent (GD) algorithm
- 4 The proximal gradient (PG) algorithm
- 5 The distributed GD (DGD) algorithm
- 6 The DGD in open multi-agent systems
- 7 Related results in the current state-of-the-art**

What are the current state-of-the-art results?

The following projected distributed gradient descent algorithm in open networks:

$$x_i(k+1) = \begin{cases} \text{proj}_{\mathcal{X}} \left(\sum_{j \in \mathcal{N}_{i,k} \cup \{i\}} w_{ij}(k) x_j(k) - \rho(k) \nabla f_i(x_i(k)) \right) & \text{if } i \in \mathcal{V}_{k+1}^R, \\ x_{i,k+1}^A & \text{if } i \in \mathcal{V}_{k+1}^A, \end{cases}$$

has been studied by Hayashi [R6] (in a more general setting), where:

- The step size $\rho(k)$ is time-varying and non-increasing, i.e., $\rho(k+1) \leq \rho(k)$;
- There is a projection step $\text{proj}_{\mathcal{X}}$ onto a convex and compact set $\mathcal{X}$, which is a constraint set common to all the agents;
- The initialization of arriving agents is within the common constraint set, i.e., $x_{i,k+1}^A \in \mathcal{X}$;
- The local objective functions f_i are assumed to be convex, not necessarily strongly convex.

[R6] Hayashi N (2023). "Distributed Subgradient Method in Open Multi-Agent Systems", IEEE Transactions on Automatic Control.

The regret metric. In standard online optimization, the regret over the horizon $\{0, \dots, T-1\}$ is:

$$\text{Reg}_T := \sum_{k=0}^{T-1} f_k(x(k)) - \sum_{k=0}^{T-1} f_k(x_\star), \quad x_\star := \underset{x \in \mathcal{X}}{\text{argmin}} \sum_{k=0}^{T-1} f_k(x).$$

This measures the cumulative suboptimality of the online decisions $x(k)$ against the **best fixed decision in hindsight** $x_* \in \mathcal{X}$. This metric has attracted much attention because, when it **grows sublinearly**, i.e.,

$$\lim_{T \rightarrow \infty} \frac{\text{Reg}_T}{T} = 0,$$

it means that the **average regret per step vanishes**. However, x_* need not be a solution of the objective function at a given time, so in general there exists $k \in \mathbb{N}$ such that $x_* \neq x_k^*$.

Open distributed case. With $\mathcal{V}_k$ time-varying, the aggregate objective evaluated at a common decision z is given by $f_k(z) := \sum_{j \in \mathcal{V}_k} f_j(z)$. Moreover, each agent i is active for time indices $k = t_i^{\text{in}}, \dots, t_i^{\text{out}} - 1$, with $0 \leq t_i^{\text{in}} \leq t_i^{\text{out}} \leq T$, so one can define a **local regret** as follows:

$$\text{Reg}_i := \sum_{k=t_i^{\text{in}}}^{t_i^{\text{out}}-1} (f_k(x_i(k)) - f_k(x_*)).$$

Indeed, Hayashi studies the sublinearity of Reg_i w.r.t. $T_i = t_i^{\text{out}} - t_i^{\text{in}}$ for each agent i individually.

91 / 94

Theorem 1 in [R6]

Under the above assumptions, the regret of agent i active for time indices $k = t_i^{\text{in}}, \dots, t_i^{\text{out}} - 1$ satisfies:

$$\text{Reg}_i \leq \underbrace{\frac{L^2 D_{\mathcal{X}}^2 \bar{V}}{2} \sum_{k=t_i^{\text{in}}}^{t_i^{\text{out}}-1} \rho(k)}_{\text{(I) gradient noise}} + \underbrace{\frac{2C_x^2 \bar{V}}{\rho(t_i^{\text{out}} - 1)}}_{\text{(II) transient}} + \underbrace{2C_x \left(4LD_{\mathcal{X}} + \frac{C_x}{\rho(t_i^{\text{out}} - 1)} \right) \sum_{k=t_i^{\text{in}}}^{t_i^{\text{out}}-1} \underbrace{\max\{|\mathcal{V}_k|, |\mathcal{V}_k^{\text{D}}|\}}_{M_k}}_{\text{(III) network changes}}.$$

To achieve a **sublinear bound** in $T_i = t_i^{\text{out}} - t_i^{\text{in}}$ for the regret, the terms in the bound must scale sublinearly:

- $\rho(k) \rightarrow 0$ sufficiently fast so that $\sum_{k=t_i^{\text{in}}}^{t_i^{\text{out}}-1} \rho(k) = o(T_i)$ (for Term I);
- $\rho(k)$ must not decay too fast, namely $1/\rho(t_i^{\text{out}} - 1) = o(T_i)$ (for Term II);
- $\sum_{k=t_i^{\text{in}}}^{t_i^{\text{out}}-1} M_k$ must grow slowly enough, namely $\sum_{k=t_i^{\text{in}}}^{t_i^{\text{out}}-1} M_k = o(T_i \rho(t_i^{\text{out}} - 1))$ (for Term III);

Remark. Since $\rho(k) \rightarrow 0$, the condition for Term III implies $\sum_{k=t_i^{\text{in}}}^{t_i^{\text{out}}-1} M_k = o(T_i)$. Since $M_k = \max\{|\mathcal{V}_k|, |\mathcal{V}_k^{\text{D}}|\}$, then $M_k \geq |\mathcal{V}_k|$, hence $\frac{1}{T_i} \sum_{k=t_i^{\text{in}}}^{t_i^{\text{out}}-1} |\mathcal{V}_k| \rightarrow 0$. Thus, the average network size over the active horizon of agent i **must vanish**, which is very restrictive and **cannot hold for a persistently nonempty network**.

[R6] Hayashi N (2023). "Distributed Subgradient Method in Open Multi-Agent Systems", IEEE Transactions on Automatic Control.

What are the current state-of-the-art results?

The following distributed gradient descent algorithm:

$$x_i(k+1) = \sum_{j \in \mathcal{N}_{i,k} \cup \{i\}} w_{ij}(k) x_j(k) - \rho \nabla f_i^k(x_i(k))$$

has been studied by Hendrickx and Rabbat [R7]:

- the local objectives may change **arbitrarily at every step k** (where $f_i^k = 0$ models the inactivity of an agent);
- the network has **fixed size** $|\mathcal{V}_k| = n$ because inactive agents still perform their state update.

A different interpretation. Agents may be replaced while keeping the network size fixed. A newly arriving agent may be associated with its own local cost f_i^k , possibly entirely different from that of the departing agent, while initializing its state with the state of the agent it replaces.

Remark. If $f_i^k = 0$, then it is simply convex and not m -strongly convex.

[R7] J.M. Hendrickx and M.G. Rabbat (2020). Stability of Decentralized Gradient Descent in Open Multi-Agent Systems, IEEE CDC.

Reinterpretation of Theorem 4 in [R7]

Consider an OMAS executing the DGD iteration under the following specific assumptions:

- the local objectives f_i^k are m -strongly convex and L -smooth, and may change **arbitrarily at every step** $k \in \mathbb{N}$;
- the network has **fixed size** $|\mathcal{V}_k| = n$, i.e., only **replacements** are allowed $|\mathcal{V}_{k+1}^A| = |\mathcal{V}_k^D|$;
- the new agent $i \in \mathcal{V}_{k+1}^A$ uses the state of the replaced (departing) agent $d \in \mathcal{V}_k^D$ as its own pre-update state, then performs the usual DGD step with its own local function f_i^k , namely,

$$x_{i,k+1}^A = w_{ii}(k)x_d(k) + \sum_{j \in \mathcal{N}_{i,k}} w_{ij}(k)x_j(k) - \rho \nabla f_i^k(x_d(k));$$
- the minimizers of f_i are **uniformly bounded**, i.e., $\exists r > 0$ such that $\|x_{i,k}^*\| \leq r$ for all $i \in \mathcal{V}_k$ and all $k \in \mathbb{N}$;
- $\lambda_{\max}(\mathcal{L}) \leq \bar{\lambda}$ is uniformly bounded;

If $\rho \in (0, (1 - \varepsilon \lambda_{\max}(\mathcal{L}))/L)$ and $\varepsilon < 1/\bar{\lambda}$, then there exists a finite constant

$$R = \begin{cases} (1 + \sqrt{6})\sqrt{n}r(1 + \sqrt{\kappa}) & \text{if } \kappa_{\varepsilon,\rho} < 3, \\ (1 + \sqrt{2\kappa_{\varepsilon,\rho}})\sqrt{n}r(1 + \sqrt{\kappa}) & \text{otherwise,} \end{cases}, \quad \text{with} \quad \begin{aligned} \kappa &= L/m \\ \kappa_{\varepsilon,\rho} &= (L + (\varepsilon/\rho)\bar{\lambda})/m, \end{aligned}$$

such that if $\|x(k^*)\| \leq R$ at some $k^* \in \mathbb{N}$, then $\|x(k)\| \leq R$ for all $k \geq k^*$, i.e., the ball of radius R centered at 0 is **invariant**.

[R7] J.M. Hendrickx and M.G. Rabbat (2020). Stability of Decentralized Gradient Descent in Open Multi-Agent Systems, IEEE CDC.



UNIVERSITÀ DEGLI STUDI
DI **CAGLIARI**



Introduction to Open Multi-Agent Systems – Part III

Descent algorithms via operator theory

Thank you for your attention!

Diego Deplano

Email: diego.deplano@unica.it

Webpage: <https://diegodeplano.github.io/>