Asynchronous distributed optimization using a randomized ADMM algorithm

Walid Hachem

CNRS; Telecom ParisTech

Joint work with
P. Bianchi, Ph. Ciblat and F. Iutzeler

Problem Statement

N computing agents, each having a private function $f_n : \mathbb{R}^K \to \bar{\mathbb{R}}$

Problem 1: Solve the minimization problem $\inf_{x \in \mathbb{R}^K} \sum_{n=1}^{N} f_n(x)$

Distributed iterative implementation: each agent updates a local estimate of the parameter and communicates it to its neighbors. Estimates ought to converge to same minimizer.

Assumption: The f_n are proper, lower semicontinous, and **convex** (notation: $f_n \in \Gamma$). A minimizer of Problem 1 exists.

A simple example from the field of signal processing

Network of N sensors.

- Y_n = random observation of sensor n,
- $x_\star$ = unknown parameter to be estimated.

Likelihood function

$$l(Y_1, \ldots, Y_N; x) = l_1(Y_1; x) \times \cdots \times l_N(Y_N; x) \quad \text{(independence)}.$$

Maximum likelihood estimate

$$\hat{x} = \arg\min_x \sum_{n=1}^{N} -\log l_n(Y_n; x).$$

Two classes of algorithms

- Local subgradient descent (or variants) + averaging with neighbors. Conceptually simple but often slow to converge.
- Dual space techniques:

  Area of active research in convex optimization theory,
  Often easy to parallelize or to distribute,
  Better convergence properties than the former,
  ADMM is one of the most popular.

Problem Statement

The Alternating Direction Method of Multipliers (ADMM) algorithm
- ADMM presentation
- Parallel implementation
- Distributed synchronous implementation

Monotone operator theory

Asynchronous ADMM

Classical description of the ADMM

$$p = \inf_{z=Mx} \left(f(x) + g(z)\right), \quad f, g \in \Gamma.$$

Given $\rho > 0$, the augmented Lagrangian is

$$\mathcal{L}_\rho(x, z; \lambda) = f(x) + g(z) + \langle \lambda, Mx - z \rangle + \frac{\rho}{2} \left\| Mx - z \right\|^2$$

ADMM :

$$\begin{aligned}
x_{k+1} &\in \arg\min_x \mathcal{L}_\rho(x, z_k; \lambda_k) \\
&= \arg\min_x f(x) + \langle \lambda_k, Mx \rangle + \frac{\rho}{2} \left\| Mx - z_k \right\|^2, \\
z_{k+1} &\in \arg\min_z \mathcal{L}_\rho(x_{k+1}, z; \lambda_k) \\
&= \arg\min_z g(z) - \langle \lambda_k, z \rangle + \frac{\rho}{2} \left\| Mx_{k+1} - z \right\|^2, \\
\lambda_{k+1} &= \lambda_k + \rho\left(Mx_{k+1} - z_{k+1}\right).
\end{aligned}$$

The Alternating Direction Method of Multipliers (ADMM) algorithm

- ADMM presentation
- Parallel implementation
- Distributed synchronous implementation

Reformulation of Problem 1

Assume that all agents are connected to a central scheduler. We look for a parallel implementation of Problem 1 (*e.g.* [Boyd *et.al* 11]).

Set $K = 1$ for now on, and let

$$\begin{array}{rccc} f: & \mathbb{R}^N & \longrightarrow & \bar{\mathbb{R}} \\ & x = (x(1), \dots, x(N)) & \longmapsto & f(x) = \sum_1^N f_n(x(n)) \end{array}$$

$$\begin{array}{rccc} g: & \mathbb{R}^N & \longrightarrow & \bar{\mathbb{R}} \\ & z & \longmapsto & g(z) = \imath_{\mathrm{span}(\mathbf{1}_N)}(z) \end{array}$$

where $\imath$ is the indicator function

$$\imath_C(x) = \begin{cases} 0 & \text{if } x \in C \\ \infty & \text{if not.} \end{cases}$$

Equivalent formulation of Problem 1: $\displaystyle \inf_{x=z \in \mathbb{R}^N} (f(x) + g(z))$.

ADMM : Parallel implementation

At iteration k,

- Write $x_k = (x_k(1), \ldots, x_k(N))$,
- $z_k = \bar{z}_k \mathbf{1}_N$ since domain of g is $\mathrm{span}(\mathbf{1}_N)$,
- Write $\lambda_k = (\lambda_k(1), \ldots, \lambda_k(N))$.

ADMM : Parallel implementation

At iteration k,

- Write $x_k = (x_k(1), \dots, x_k(N))$,
- $z_k = \bar{z}_k \mathbf{1}_N$ since domain of g is $\mathrm{span}(\mathbf{1}_N)$,
- Write $\lambda_k = (\lambda_k(1), \dots, \lambda_k(N))$.

Algorithm:

$$x_{k+1}(n) = \arg\min_x f_n(x) + \lambda_k(n)x + \frac{\rho}{2}(x - \bar{z}_k)^2 \quad \text{for } n = 1, \dots, N$$

ADMM : Parallel implementation

At iteration k,

- Write $x_k = (x_k(1), \ldots, x_k(N))$,
- $z_k = \bar{z}_k \mathbf{1}_N$ since domain of g is span$(\mathbf{1}_N)$,
- Write $\lambda_k = (\lambda_k(1), \ldots, \lambda_k(N))$.

Algorithm:

$$x_{k+1}(n) = \arg\min_x f_n(x) + \lambda_k(n)x + \frac{\rho}{2}(x - \bar{z}_k)^2 \quad \text{for } n = 1, \ldots, N$$

$$\bar{z}_{k+1} = \frac{1}{N}\sum_1^N x_{k+1}(n), \quad \text{projection of } x_{k+1} \text{ on domain of } g$$

ADMM : Parallel implementation

At iteration k,

- Write $x_k = (x_k(1), \ldots, x_k(N))$,
- $z_k = \bar{z}_k \mathbf{1}_N$ since domain of g is $\text{span}(\mathbf{1}_N)$,
- Write $\lambda_k = (\lambda_k(1), \ldots, \lambda_k(N))$.

Algorithm:

$$x_{k+1}(n) = \arg\min_x f_n(x) + \lambda_k(n)x + \frac{\rho}{2}(x - \bar{z}_k)^2 \quad \text{for } n = 1, \ldots, N$$

$$\bar{z}_{k+1} = \frac{1}{N} \sum_1^N x_{k+1}(n), \quad \text{projection of } x_{k+1} \text{ on domain of } g$$

$$\lambda_{k+1}(n) = \lambda_k(n) + \rho(x_{k+1}(n) - \bar{z}_{k+1}) \quad \text{for } n = 1, \ldots, N$$

The Alternating Direction Method of Multipliers (ADMM) algorithm

ADMM presentation

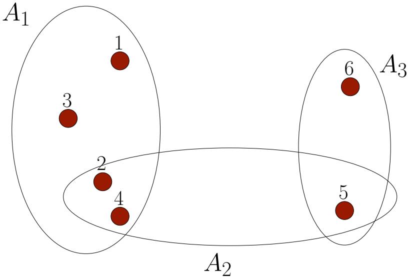
Parallel implementation

Distributed synchronous implementation

Reformulation of Problem 1 on an example

Idea of [Ribeiro *et.al* 08]. Let $A_1, \ldots, A_L$ be a collection of subsets of the set $\mathcal{A} = \{1, \ldots, N\}$ of agents.

Example with $N = 6$ and $L = 3$:



Problem

$$\inf_{x \in \mathbb{R}^6} f(x) + \imath_{\mathrm{span}(\mathbf{1}_4)} \begin{pmatrix} x(1) \\ x(2) \\ x(3) \\ x(4) \end{pmatrix} + \imath_{\mathrm{span}(\mathbf{1}_3)} \begin{pmatrix} x(2) \\ x(4) \\ x(5) \end{pmatrix} + \imath_{\mathrm{span}(\mathbf{1}_2)} \begin{pmatrix} x(5) \\ x(6) \end{pmatrix}$$

is equivalent to Problem 1.

Reformulation of Problem 1

$$
\begin{array}{llll}
g: & \mathbb{R}^{|A_1|} \times \cdots \times \mathbb{R}^{|A_L|} & \longrightarrow & \bar{\mathbb{R}} \\
 & z = (z^1, \ldots, z^L) & \longmapsto & g(z) = \sum_1^L \imath_{\mathrm{span}(\mathbf{1}_{|A_\ell|})}(z^\ell)
\end{array}
$$

Let

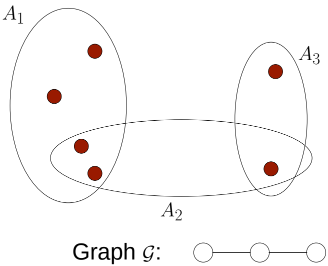
$$
M = \begin{pmatrix} S_{A_1} \\ \vdots \\ S_{A_L} \end{pmatrix}
$$

where S_{A_ℓ} is the matrix that selects the components of x belonging to A_ℓ.

> **Problem 2:** Find $\inf\limits_{z=Mx} f(x) + g(z)$.

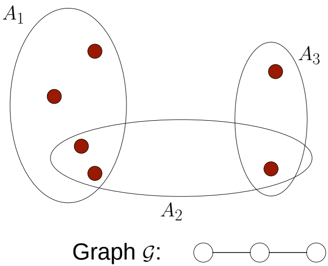
Reformulation of Problem 1

Let $\mathcal{G} = (\{1, \ldots, L\}, \mathcal{E})$ be the graph with edges $\{\ell, m\} \in \mathcal{E}$ if $A_\ell \cap A_m \neq \emptyset$.
Our example:



Reformulation of Problem 1

Let $\mathcal{G} = (\{1, \ldots, L\}, \mathcal{E})$ be the graph with edges $\{\ell, m\} \in \mathcal{E}$ if $A_\ell \cap A_m \neq \emptyset$.
Our example:



If $\cup A_\ell = \mathcal{A}$ and the graph $\mathcal{G}$ is **connected** as we shall always suppose, then Problems 1 and 2 are **equivalent**.

Distributed synchronous ADMM

- Write $x_k = (x_k(1), \ldots, x_k(N))$,
- $z_k = \begin{bmatrix} \bar{z}_k^1 \mathbf{1}_{|A_1|} \\ \vdots \\ \bar{z}_k^L \mathbf{1}_{|A_L|} \end{bmatrix} \in$ domain of g at any moment k,
- Write $\lambda_k = (\lambda_k^1, \ldots, \lambda_k^L)$ and $\lambda_k^\ell = (\lambda_k^\ell(n_1), \ldots, \lambda_k^\ell(n_{|A_\ell|})) \in \mathbb{R}^{|A_\ell|}$, Indices n_i being those of the agents belonging to A_ℓ (non zero columns of S_{A_ℓ}).

Distributed synchronous ADMM

- Write $x_k = (x_k(1), \dots, x_k(N))$,
- $z_k = \begin{bmatrix} \bar{z}_k^1 \mathbf{1}_{|A_1|} \\ \vdots \\ \bar{z}_k^L \mathbf{1}_{|A_L|} \end{bmatrix} \in$ domain of g at any moment k,
- Write $\lambda_k = (\lambda_k^1, \dots, \lambda_k^L)$ and $\lambda_k^\ell = (\lambda_k^\ell(n_1), \dots, \lambda_k^\ell(n_{|A_\ell|})) \in \mathbb{R}^{|A_\ell|}$, Indices n_i being those of the agents belonging to A_ℓ (non zero columns of S_{A_ℓ}).

Algorithm:

Distributed synchronous ADMM

- Write $x_k = (x_k(1), \ldots, x_k(N))$,
- $z_k = \begin{bmatrix} \bar{z}_k^1 \mathbf{1}_{|A_1|} \\ \vdots \\ \bar{z}_k^L \mathbf{1}_{|A_L|} \end{bmatrix} \in$ domain of g at any moment k,
- Write $\lambda_k = (\lambda_k^1, \ldots, \lambda_k^L)$ and $\lambda_k^\ell = (\lambda_k^\ell(n_1), \ldots, \lambda_k^\ell(n_{|A_\ell|})) \in \mathbb{R}^{|A_\ell|}$, Indices n_i being those of the agents belonging to A_ℓ (non zero columns of S_{A_ℓ}).

Algorithm:

$$x_{k+1}(n) \;=\; \arg\min_x f_n(x) + \sum_{\ell : n \in A_\ell} x \lambda_k^\ell(n) + \frac{\rho}{2}\left(x - \bar{z}_k^\ell\right)^2 \quad \text{for } n = 1, \ldots, N,$$

Distributed synchronous ADMM

- Write $x_k = (x_k(1), \dots, x_k(N))$,
- $z_k = \begin{bmatrix} \bar{z}_k^1 \mathbf{1}_{|A_1|} \\ \vdots \\ \bar{z}_k^L \mathbf{1}_{|A_L|} \end{bmatrix} \in$ domain of g at any moment k,
- Write $\lambda_k = (\lambda_k^1, \dots, \lambda_k^L)$ and $\lambda_k^\ell = (\lambda_k^\ell(n_1), \dots, \lambda_k^\ell(n_{|A_\ell|})) \in \mathbb{R}^{|A_\ell|}$, Indices n_i being those of the agents belonging to A_ℓ (non zero columns of S_{A_ℓ}).

Algorithm:

$$\begin{aligned} x_{k+1}(n) &= \arg\min_x f_n(x) + \sum_{\ell : n \in A_\ell} x \lambda_k^\ell(n) + \frac{\rho}{2}\Big(x - \bar{z}_k^\ell\Big)^2 \quad \text{for } n = 1, \dots, N, \\ \bar{z}_{k+1}^\ell &= \frac{1}{|A_\ell|} \sum_{n \in A_\ell} x_{k+1}(n) \quad \text{for } \ell = 1, \dots, L, \end{aligned}$$

Distributed synchronous ADMM

- Write $x_k = (x_k(1), \ldots, x_k(N))$,
- $z_k = \begin{bmatrix} \bar{z}_k^1 \mathbf{1}_{|A_1|} \\ \vdots \\ \bar{z}_k^L \mathbf{1}_{|A_L|} \end{bmatrix} \in$ domain of g at any moment k,
- Write $\lambda_k = (\lambda_k^1, \ldots, \lambda_k^L)$ and $\lambda_k^\ell = (\lambda_k^\ell(n_1), \ldots, \lambda_k^\ell(n_{|A_\ell|})) \in \mathbb{R}^{|A_\ell|}$, Indices n_i being those of the agents belonging to A_ℓ (non zero columns of S_{A_ℓ}).

Algorithm:

$$
\begin{aligned}
x_{k+1}(n) &= \arg\min_x f_n(x) + \sum_{\ell : n \in A_\ell} x \lambda_k^\ell(n) + \frac{\rho}{2}\left(x - \bar{z}_k^\ell\right)^2 \quad \text{for } n = 1, \ldots, N, \\
\bar{z}_{k+1}^\ell &= \frac{1}{|A_\ell|} \sum_{n \in A_\ell} x_{k+1}(n) \quad \text{for } \ell = 1, \ldots, L, \\
\lambda_{k+1}^\ell(n) &= \lambda_k^\ell(n) + \rho(x_{k+1}(n) - \bar{z}_{k+1}^\ell) \quad \text{for } n = 1, \ldots, N \text{ and for } \ell : n \in A_\ell.
\end{aligned}
$$

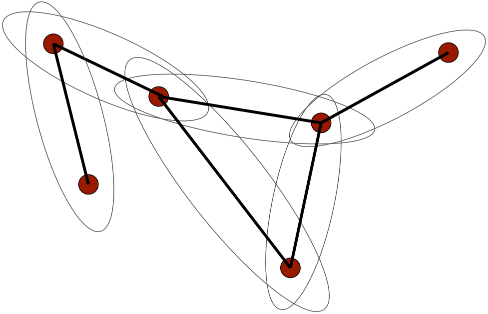
Algorithm execution

At clock tick $k+1$,

- Every agent computes $x_{k+1}(n)$,
- Members of a set A_ℓ belong to a connected communication network. They send their updates $x_{k+1}(n)$ to a device (possibly one of them) who computes the average $\bar{z}^\ell_{k+1}$. This average is then broadcasted to the members of A_ℓ,
- The $\{\lambda^\ell_k(n)\}_{n \in A_\ell}$ are local to agents. Each is updated by the agent according to the third equation.

A simple example

Communication network between agents represented by a connected non oriented graph with no self loops $G = (\mathcal{A}, E)$
Set $L = |E|$. Any $\{m, n\} \in E$ (notation $m \sim n$) is a set A_ℓ.



A simple example

We identify the index ℓ of set $A_\ell = \{m, n\} \in E$ with $\{m, n\}$.
All Agents perform updates

$$x_{k+1}(n) = \arg\min_x f_n(x) + \sum_{m \sim n} x \lambda_k^{m,n}(n) + \frac{\rho}{2}\left(x - \bar{z}_k^{m,n}\right)^2$$

A simple example

We identify the index ℓ of set $A_\ell = \{m, n\} \in E$ with $\{m, n\}$.
All Agents perform updates

$$x_{k+1}(n) = \arg\min_x f_n(x) + \sum_{m\sim n} x\lambda_k^{m,n}(n) + \frac{\rho}{2}\left(x - \bar{z}_k^{m,n}\right)^2$$

All agents m and n such that $m \sim n$ exchange the values of $x_{k+1}(m)$ and $x_{k+1}(n)$. They compute

$$\bar{z}_{k+1}^{m,n} = \frac{x_{k+1}(m) + x_{k+1}(n)}{2}$$

and

$$\lambda_{k+1}^{m,n}(n) = \lambda_k^{m,n}(n) + \rho\frac{x_{k+1}(n) - x_{k+1}(m)}{2}$$
$$\lambda_{k+1}^{m,n}(m) = \lambda_k^{m,n}(m) + \rho\frac{x_{k+1}(m) - x_{k+1}(n)}{2}$$

Duality

Consider the **primal problem**:

$$p = \inf_x \left(f(x) + g(Mx)\right), \quad f, g \in \Gamma$$

where M is a $T \times N$ matrix.
Let

$$\begin{array}{cccc} f^* : & \mathbb{R}^N & \longrightarrow & \mathbb{R} \\ & \phi & \longmapsto & f^*(\phi) = \sup_{x \in \mathbb{R}^N} \left(\langle x, \phi \rangle - f(x)\right) \end{array}$$

be the **Legendre-Fenchel Transform** of f. Similar definition for g.
The **dual problem** is

$$p^* = -\inf_{\lambda \in \mathbb{R}^T} \left(f^*(-M^*\lambda) + g^*(\lambda)\right)$$

If a qualification condition holds, the duality gap is zero ($p = p^*$), and the dual problem is attained. We also assume the primal problem is attained (existence of a saddle point).

Splitting

Solve the **dual problem** by finding a zero of

$$-M\partial f^*(-M^*\cdot) + \partial g^*(\cdot).$$

where ∂f^* and ∂g^* are the **subdifferentials** of f^* and g^*.

Subdifferentials of convex functions are particular cases of so called **monotone operators**.

Douglas-Rachford (or **Lions-Mercier** [Lions Mercier 79]) splitting algorithm is a procedure for finding the zero of the **sum of two monotone operators**.

Applied to the two operators above, it results in the **ADMM** [Gabay 83].
$\Rightarrow$ Alternative approach to the augmented Lagrangian.

Monotone operators

A **monotone operator** on a Euclidean space X is a set-valued application $U : X \to 2^X$ such that

$$\forall (x, y),\ \forall (u, v) \in U(x) \times U(y), \quad \langle u - v, x - y \rangle \geq 0$$

- It is **maximal monotone** if it is not contained in an other monotone operator. Example: the subdifferential of a function in Γ.
- A point x is a **zero** of U if $0 \in U(x)$

The **resolvent** of U is

$$J_U = (I + U)^{-1} \quad \text{where } I \text{ is the identity operator}$$

- $\text{domain}(J_U) = X$ whenever U is maximal
- J_U is single-valued (it is a function)
- Fixed points of J_U coincide with the zeros of U: $\text{fix}(J_U) = \text{zer}(U)$.

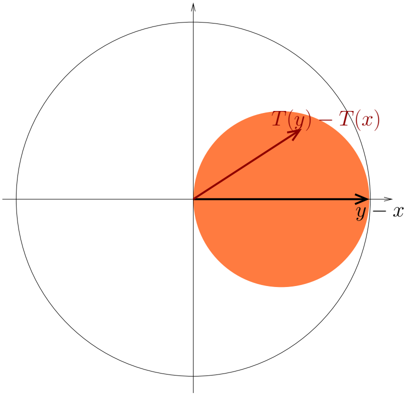
Non expansiveness

- A single valued monotone operator T is said **non expansive** if

$$\forall x, y \in \mathrm{domain}(T), \quad \|T(x) - T(y)\| \leq \|x - y\|.$$

- It is said **firmly non expansive** if

$$\forall x, y \in \mathrm{domain}(T), \quad \langle T(x) - T(y), x - y \rangle \geq \|T(x) - T(y)\|^2$$



Properties related with non expansiveness

- J is a **firmly non expansive** operator with domain $X \Leftrightarrow J$ is the **resolvent** of a maximal monotone operator.

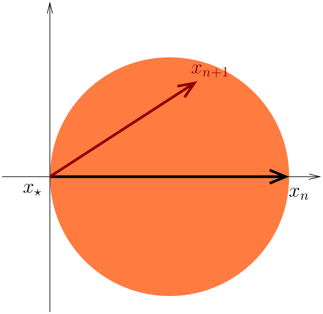
- If T is non expansive, then $\dfrac{I+T}{2}$ is firmly non expansive.

- The **reflected resolvent** (sometimes called Cayley Transform) of a monotone operator U is $R_U = 2J_U - I$.
  If U is maximal monotone, then R_U is non expansive with domain X.

The proximal point algorithm

$$x_{n+1} = J_U(x_n)$$

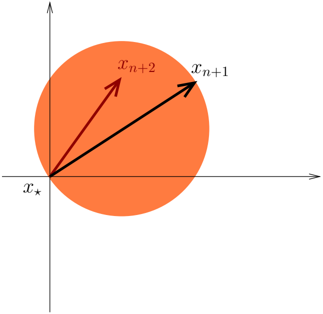
Assume that there exists $x_\star \in \mathrm{zer}(U)$



The proximal point algorithm

$$x_{n+1} = J_U(x_n)$$

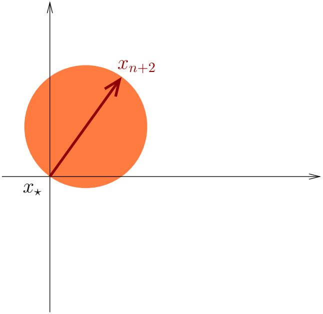
Assume that there exists $x_\star \in \mathrm{zer}(U)$



The proximal point algorithm

$$x_{n+1} = J_U(x_n)$$

Assume that there exists $x_\star \in \mathrm{zer}(U)$



$\|x_n - x_\star\|$ decreases with n

Convergence of the proximal point algorithm [Rockafellar 76]:
If U is maximal monotone and $\mathrm{zer}(U) \neq \emptyset$, then x_n converges to a point in $\mathrm{fix}(J_U) = \mathrm{zer}(U)$.

Application

$U = \partial f$ where f is a function in Γ attaining its infimum.

Let $\rho > 0$ and consider the iterates $x_{k+1} = J_{\rho U}(x_k) = (I + \rho \partial f)^{-1}(x_k)$.
We have $x_{k+1} + \rho \partial f(x_{k+1}) = x_k$, in other words,

$$x_{k+1} = \arg\min_w f(w) + \frac{1}{2\rho}\|w - x_k\|^2 = x_k - \rho \partial f(x_{k+1})$$

For any $\rho > 0$, the algorithm **converges to a minimum of f**.
Notice the difference with the classical subgradient.

Douglas-Rachford splitting

Problem: Find a zero of the **sum of two maximal monotone operators** $U + V$ by a procedure involving each operator **individually**.

Douglas-Rachford splitting

Problem: Find a zero of the **sum of two maximal monotone operators** $U + V$ by a procedure involving each operator **individually**.

Douglas-Rachford splitting:
Assume that $\mathrm{zer}(U + V) \neq \emptyset$. Set $\rho > 0$ and define operator

$$J_{\mathrm{DR}} = \frac{1}{2}\left(R_{\rho U} R_{\rho V} + I\right)$$

where $R_{\rho U}$ and $R_{\rho V}$ are the reflected resolvents of ρU and ρV. Then the set of fixed points of J_{DR} is not empty. For any $\zeta \in X$, the sequence $\zeta_{k+1} = J_{\mathrm{DR}}(\zeta_k)$ converges to a fixed point $\zeta_\star$ of J_{DR}, and $\lambda_\star = J_{\rho V}(\zeta_\star) \in \mathrm{zer}(U + V)$.

Douglas Rachford splitting: proof outline

- Since U is maximal monotone, $R_{\rho U}$ is non expansive with domain X. Same for V. Hence $J_{\mathrm{DR}} = 0.5(R_{\rho U} R_{\rho V} + I)$ is firmly non expansive with domain X.
  It is the resolvent of a maximal monotone operator (the so called Douglas-Rachford operator),

- Check that
  $\mathrm{zer}(U+V) = J_{\rho V}(\mathrm{fix}\, R_{\rho U} R_{\rho V}) = J_{\rho V}(\mathrm{fix}(0.5(R_{\rho U} R_{\rho V} + I)),$

- Apply the theorem of convergence of the proximal point algorithm.

ADMM as a Douglas-Rachford operator [Gabay 83] (outline)

Set

$$U = -M\partial f^*(-M^* \cdot) \quad \text{and} \quad V = \partial g^*$$

Algorithm can be rewritten

1. Input: $\zeta_k = \lambda_k + \rho z_k$ with $\lambda_k = J_{\rho V}(\zeta_k)$,
2. Set $v_{k+1} = J_{\rho U}(\lambda_k - \rho z_k)$.
3. Algorithm output: $\zeta_{k+1} = J_{\mathrm{DR}}(\zeta_k) = v_{k+1} + \rho z_k$.

- Using the identity $\partial f^* = \partial f^{-1}$, Step 2 can be translated into the update equation for x_{k+1} in Slide 6.
- $\zeta_{k+1} = v_{k+1} + \rho z_k$ at the output of Step 3 should be re-represented as $\zeta_{k+1} = \lambda_{k+1} + \rho z_{k+1}$ where $\lambda_{k+1} = J_{\rho V}(\zeta_{k+1})$. Using the identity $\partial g^* = \partial g^{-1}$, this identity gives the update equations for z_{k+1} and λ_{k+1}.

Notations

- Assume $X = X^1 \times \cdots X^L$ (cartesian product of Euclidean spaces) and write accordingly any $\zeta \in X$ as $\zeta = (\zeta^1, \ldots, \zeta^L)$.
- Let J_U be the resolvent of a maximal monotone operator U on X, and write $J_U(\zeta) = (J^1(\zeta), \ldots, J^L(\zeta))$.
- Given $\ell \in \{1, \ldots, L\}$, define

$$\bar{J}_U^{\ell}(\zeta) = \begin{pmatrix} \zeta^1 \\ \vdots \\ \zeta^{\ell-1} \\ J^{\ell}(\zeta) \\ \zeta^{\ell+1} \\ \vdots \\ \zeta^L \end{pmatrix} .$$

Random Gauss-Seidel iterations: main result

Let ξ_k be an iid random process valued in the set $\{1, \ldots, L\}$, and such that $\min_{1 \le \ell \le L} \mathbb{P}[\xi_1 = \ell] > 0$.

Theorem:
Assume U is maximal monotone. Then for any initial value ζ_0, the random sequence $\zeta_{k+1} = \bar{J}_U^{\xi_{k+1}}(\zeta_k)$ converges almost surely to an element of $\mathrm{fix}(J_U)$ whenever $\mathrm{fix}(J_U) \neq \emptyset$.

In our case, J_U will be the Douglas-Rachford resolvent.

Application: asynchronous ADMM algorithm

Random Gauss-Seidel updates of the Douglas-Rachford resolvent made **at level of sets** A_ℓ.
Cartesian product $\mathbb{R}^{\sum |A_\ell|} = \mathbb{R}^{|A_1|} \times \cdots \times \mathbb{R}^{|A_L|}$.

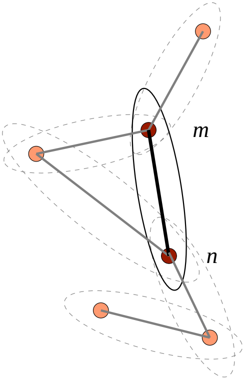
For $\xi_{k+1} = \ell$, we get

$$\zeta_{k+1} = \begin{bmatrix} \lambda_k^1 + \rho \bar{z}_k^1 \mathbf{1}_{|A_1|} \\ \vdots \\ \lambda_k^{\ell-1} + \rho \bar{z}_k^{\ell-1} \mathbf{1}_{|A_{\ell-1}|} \\ J_{\mathrm{DR}}^\ell(\lambda_k + \rho z_k) \\ \lambda_k^{\ell+1} + \rho \bar{z}_k^{\ell+1} \mathbf{1}_{|A_{\ell+1}|} \\ \vdots \\ \lambda_k^L + \rho \bar{z}_k^L \mathbf{1}_{|A_L|} \end{bmatrix}$$

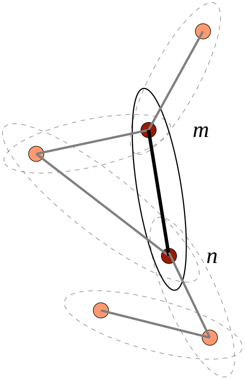
Only the $\left((x_k(n))_{n \in A_\ell}, \lambda_k^\ell, \bar{z}_k^\ell\right)$ are updated. **Agents not belonging to** A_ℓ **remain inactive**.

Implementation in the case of example above

$$A_{\xi_{k+1}} = \{m, n\}$$



Implementation in the case of example above



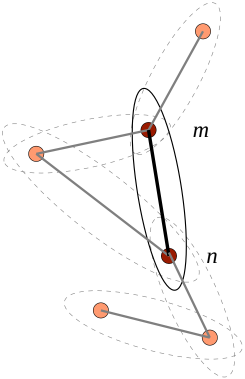
$$A_{\xi_{k+1}} = \{m, n\}$$

- Agent n computes

$$x_{k+1}(n) = \arg\min_x f_n(x) + \sum_{j \sim n} x \lambda_k^{j,n}(n) + \frac{\rho}{2}\left(x - \bar{z}_k^{j,n}\right)^2$$

and similarly for Agent m.

Implementation in the case of example above



$$A_{\xi_{k+1}} = \{m, n\}$$

- Agent n computes

$$x_{k+1}(n) = \arg\min_x f_n(x) + \sum_{j \sim n} x \lambda_k^{j,n}(n) + \frac{\rho}{2}\left(x - \bar{z}_k^{j,n}\right)^2$$

and similarly for Agent m.

- They exchange $x_{k+1}(m)$ and $x_{k+1}(n)$ and compute

$$\bar{z}_{k+1}^{m,n} = 0.5(x_{k+1}(m) + x_{k+1}(n)),$$

$$\lambda_{k+1}^{m,n}(n) = \lambda_k^{m,n}(n) + \rho \frac{x_{k+1}(n) - x_{k+1}(m)}{2}$$

$$\lambda_{k+1}^{m,n}(m) = \lambda_k^{m,n}(m) + \rho \frac{x_{k+1}(m) - x_{k+1}(n)}{2}$$

The proof

Assume $\mathbb{P}[\xi_1 = 1] = \cdots = \mathbb{P}[\xi_1 = L] = 1/L$ for simplicity.
Recalling $X = X^1 \times \cdots \times X^L$, let $\|\cdot\|_{X^\ell}$ be the norm on X^ℓ.
Let $\mathcal{F}_k = \sigma(\xi_1, \ldots, \xi_k)$.
Let $\zeta_\star$ be a fixed point of J_U.

$$\begin{aligned}
\mathbb{E}\left[L\|\zeta_{k+1} - \zeta_\star\|^2 \,\middle|\, \mathcal{F}_k\right] &= \sum_{\ell=1}^{L} \|\bar{J}_U^\ell(\zeta_k) - \zeta_\star\|^2 \\
&= \sum_{\ell=1}^{L} \Big(\|J_U^\ell(\zeta_k) - \zeta_\star^\ell\|_{X^\ell}^2 + \sum_{\substack{i=1 \\ i\neq\ell}}^{L} \|\zeta_k^i - \zeta_\star^i\|_{X^i}^2 \Big) \\
&= \|J_U(\zeta_k) - \zeta_\star\|^2 + (L-1)\|\zeta_k - \zeta_\star\|^2.
\end{aligned}$$

The proof

Recall J_U is firmly nonexpansive. So is Operator $I - J_U$. Since $(I - J_U)\zeta_\star = 0$, we have

$$
\begin{aligned}
&\|J_U(\zeta_k) - \zeta_\star\|^2 - \|\zeta_k - \zeta_\star\|^2 \\
&\qquad = \|J_U(\zeta_k) - \zeta_k + \zeta_k - \zeta_\star\|^2 - \|\zeta_k - \zeta_\star\|^2 \\
&\qquad = \|J_U(\zeta_k) - \zeta_k\|^2 + 2\langle J_U(\zeta_k) - \zeta_k, \zeta_k - \zeta_\star\rangle \\
&\qquad = \|J_U(\zeta_k) - \zeta_k\|^2 - 2\langle (I - J_U)(\zeta_k) - (I - J_U)(\zeta_\star), \zeta_k - \zeta_\star\rangle \\
&\qquad \leq -\|J_U(\zeta^k) - \zeta_k\|^2
\end{aligned}
$$

The proof

Hence

$$\mathbb{E}\left[\|\zeta_{k+1}-\zeta_\star\|^2 \,|\, \mathcal{F}_k\right] \leq \|\zeta_k-\zeta_\star\|^2 - \frac{1}{L}\|J_U(\zeta^k)-\zeta_k\|^2 \qquad (1)$$

This shows that $\|\zeta_k-\zeta_\star\|^2$ is a nonnegative supermartingale. As such, it converges towards a random variable $0 \leq X_{\zeta_\star} < \infty$. By a separability argument, we get

Fact 1: There is a probability one set on which $\|\zeta_k-\zeta_\star\|$ converges for every fixed point $\zeta_\star$ of J_U.

Taking expectations in (1) and iterating,

$$\sum_0^\infty \mathbb{E}\left[\|J(\zeta_k)-\zeta_k\|^2\right] \leq L\|\zeta_0-\zeta_\star\|^2 < \infty.$$

By Markov's inequality and Borel Cantelli's lemma

Fact 2: $J(\zeta_k)-\zeta_k \to 0$ almost surely.

Proof

On the probability one event where Facts 1 and 2 hold,

- Sequence $\|\zeta_k\|$ is bounded since $\|\zeta_k - \zeta_\star\|$ converges.
- Since J_U is nonexpansive, it is continuous, and Fact 2 shows that accumulation points of ζ_k are fixed points of J_U.
- Assume $\boldsymbol{\zeta}_\star$ is an accumulation point. Since $\|\zeta_k - \boldsymbol{\zeta}_\star\|$ converges by Fact 1, $\lim \|\zeta_k - \boldsymbol{\zeta}_\star\| = \liminf \|\zeta_k - \boldsymbol{\zeta}_\star\| = 0$. So $\boldsymbol{\zeta}_\star$ is unique.

Simulation setting

Configuration of example above, with $\mathcal{A} = \{1, \ldots, 5\}$ and $E = \{\{1, 2\}, \{2, 3\}, \{3, 4\}, \{4, 5\}, \{5, 3\}\}$.

Behavior of

- The synchronous **distributed gradient** algorithm,
- An **asynchronous** version of the **distributed gradient**,
- The **synchronous ADMM**,
- The **asynchronous ADMM**.

with quadratic functions f_n.

Simulation results

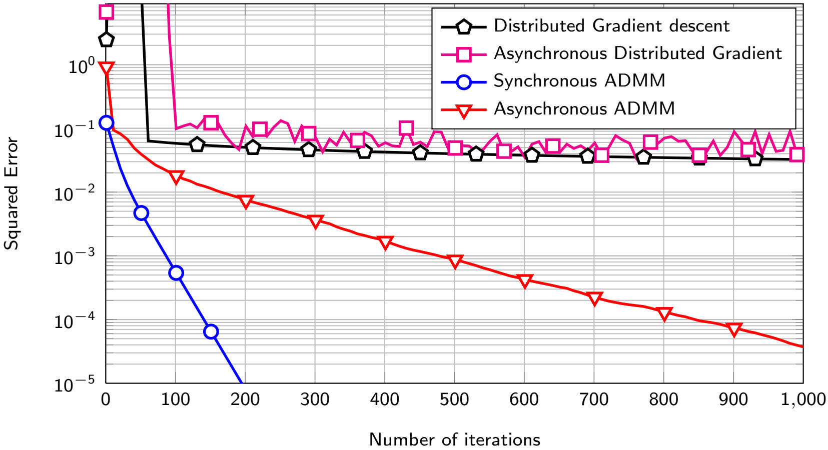


Figure: Squared error versus the number of primal updates