Distributed convex optimization: a consensus-based Newton-Raphson approach

Damiano Varagnolo

joint work with A. Cenedese, G. Pillonetto, L. Schenato, F. Zanella

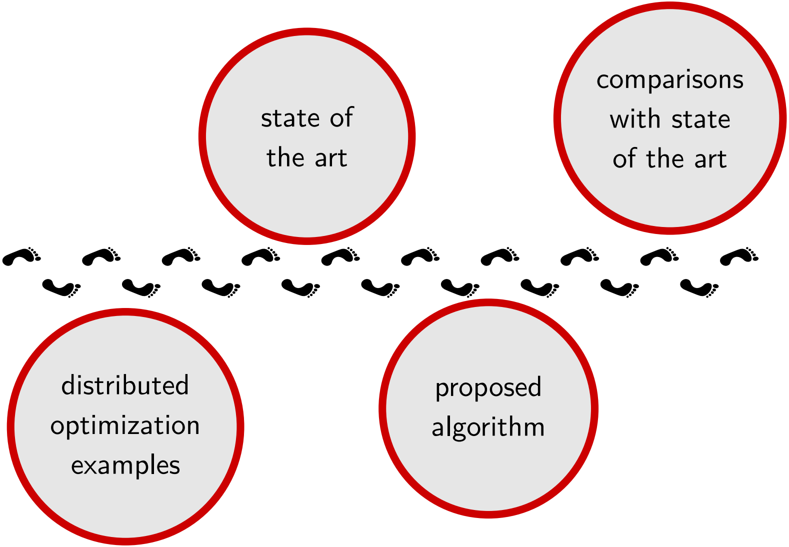
Department of Information Engineering - University of Padova

December 14$^{\text{th}}$, 2011 – 50$^{\text{th}}$ IEEE CDC

UNIVERSITAS STUDII PADUANI MCCXXII

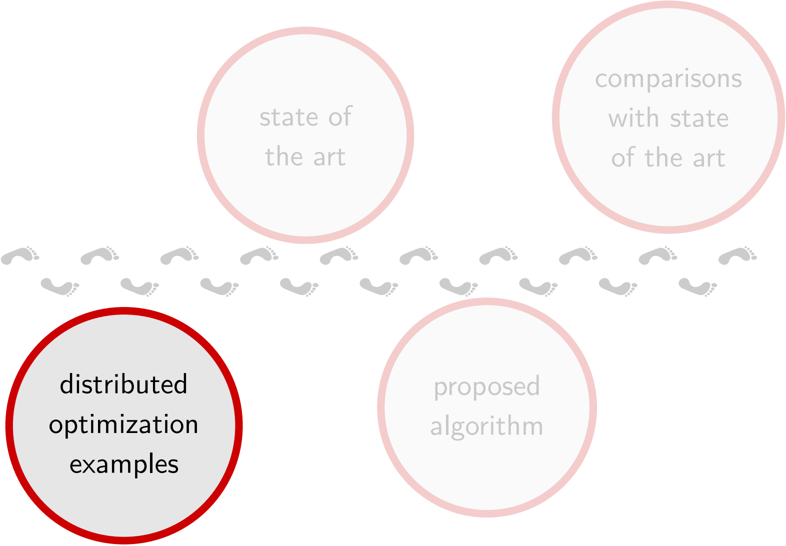
DIPARTIMENTO
DI INGEGNERIA
DELL'INFORMAZIONE

This talk



state of the art
comparisons with state of the art
distributed optimization examples
proposed algorithm

This talk



state of the art
comparisons with state of the art
distributed optimization examples
proposed algorithm

Distributed convex optimization and its importance

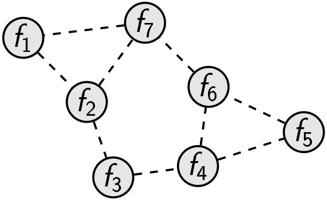
A general problem ...

$$\begin{aligned} \text{minimize} \quad & f(x) = \textstyle\sum_{i=1}^{N} f_i(x) \\ \text{subject to} \quad & g(x) \leq 0 \\ & x \in \mathcal{X} \end{aligned}$$

under ***convexity assumptions***

... motivated by multi-agents scenarios

Networked system where neighbors cooperate to find the optimum



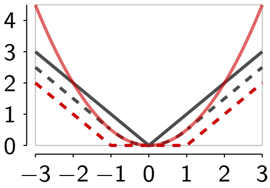
Distribution optimization - Example 1

Regression in sensor networks

(e.g. when estimation = optimization of a cost function)

Residuals minimization

$$\begin{array}{ll} \min_\theta & \sum_{i=1}^N \phi(y_i - \hat{y}_i) \\ \text{s.t.} & \hat{y}_i = \theta^T x_i \end{array}$$

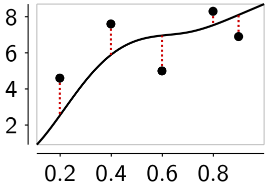


$$\phi(r) = |r|^2 \qquad \text{(least squares)}$$

$$\phi(r) = |r| \qquad \text{(least abs. deviations)}$$

$$\phi(r) = \begin{cases} 0 & \text{if } |r| < 1 \\ |r| - 1 & \text{otherwise} \end{cases} \qquad \text{(Vapnik)}$$

$$\phi(r) = \begin{cases} |r|^2 & \text{if } |r| < 1 \\ 2(|r| - 1) & \text{otherwise} \end{cases} \qquad \text{(Huber)}$$



Distribution optimization - Example 2

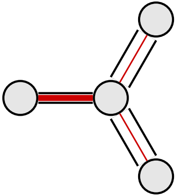
Resource allocation in wireless systems

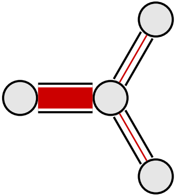
(e.g. when optimal allocation = optimization of a cost function)

Links capacity allocation [Johansson 2008]

suboptimal allocation

optimal allocation





 's width = allocated link capacity

 's width = data flux

Update

state of the art

comparisons with state of the art

distributed optimization examples

proposed algorithm

State of the art

Distributed optimization methods: 3 main categories

- Primal decompositions methods
  (e.g. distributed subgradients)
- Dual decompositions methods
  (e.g. alternating direction method of multipliers)
- Heuristic methods
  (e.g. swarm optimization, genetic algorithms)

Primal decomposition methods (distributed)

Distributed subgradient methods [Nedić Ozdaglar 2009]

$$x_i(k+1) = \mathcal{P}_{\mathcal{X}}\left[\sum_{j=1}^{N} a_{ij}(k)x_j(k) + \alpha_i(k)g_i(x_i(k))\right]$$

with

- $\sum_{j=1}^{N} a_{ij}(k)x_j(k) :=$ aver. consensus step on *local* estimates $x_j(k)$
- $g_i(x_i(k)) :=$ *local* subgradient of *local* cost $f_i(\cdot)$ at $x_i(k)$
- $\alpha_i(k) :=$ *local* stepsize

Convergence properties [Nedić Ozdaglar (2007)]

E.g., for *bounded subgradients* and $\alpha_i(k) = \alpha$ then

$$\lim \inf_{k\to+\infty} f(x_i(k)) = f^* + \text{small constant}$$

Dual decomposition methods (distributed)

Alternating Direction Method of Multipliers

[Bertsekas Tsitsiklis 1997]

$$\begin{array}{ll} \text{minimize} & f_1(x_1) + f_2(x_2) \\ \text{subject to} & A_1 x_1 + A_2 x_2 - b = 0 \end{array}$$

Augmented Lagrangian:

$$\begin{aligned} L_\rho(x_1, x_2, \lambda) \;:=\; & f_1(x_1) + f_2(x_2) \\ & + \lambda^T (A_1 x_1 + A_2 x_2 - b) \\ & + \frac{\rho}{2} \| A_1 x_1 + A_2 x_2 - b \|_2^2 \end{aligned}$$

Algorithm

1. $\boldsymbol{x_1(k+1)} = \arg\min_{\boldsymbol{x_1}} L_\rho(\boldsymbol{x_1}, x_2(k), \lambda(k))$
2. $\boldsymbol{x_2(k+1)} = \arg\min_{\boldsymbol{x_2}} L_\rho(x_1(k+1), \boldsymbol{x_2}, \lambda(k))$
3. $\lambda(k+1) = \lambda(k) + \rho\,(A_1 x_1 + A_2 x_2 - b)$

Drawbacks of the considered algorithms

Primal based strategies

- may be slow
- may not converge to the optimum

Dual based strategies

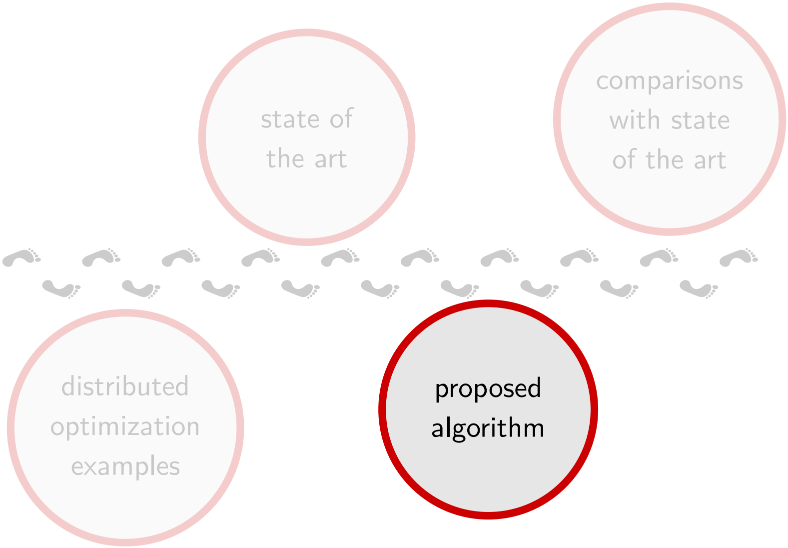
- may be computationally expensive
- require topological knowledge
- implementation to handle time-varying graphs, time delays, etc. may require effort

Motivations for our method

The algorithm that we want:

1. easy to be implemented
2. with small computational requirements
3. does not require synchronization or topology knowledge
4. assured to converge to global optimum
5. inheriting good properties of standard consensus
   convergence proofs, robustness, ...

Update


state of the art
comparisons with state of the art
distributed optimization examples
proposed algorithm

Our position in literature

How the proposed algorithm relates to other techniques?

- primal decomposition method
- uses second-order approximations

caveat:

- unconstrained convex optimization
- strong assumptions on the cost functions
  (all other algorithms can work under our hypotheses)

our contribute: better convergence speed for primal methods

Illustrative example: quadratic local cost functions

Derivation of the algorithm - step 1 on 3

Simplified scalar scenario

$$f_i(x) = \frac{1}{2} a_i (x - b_i)^2 + c_i \qquad\qquad a_i > 0$$

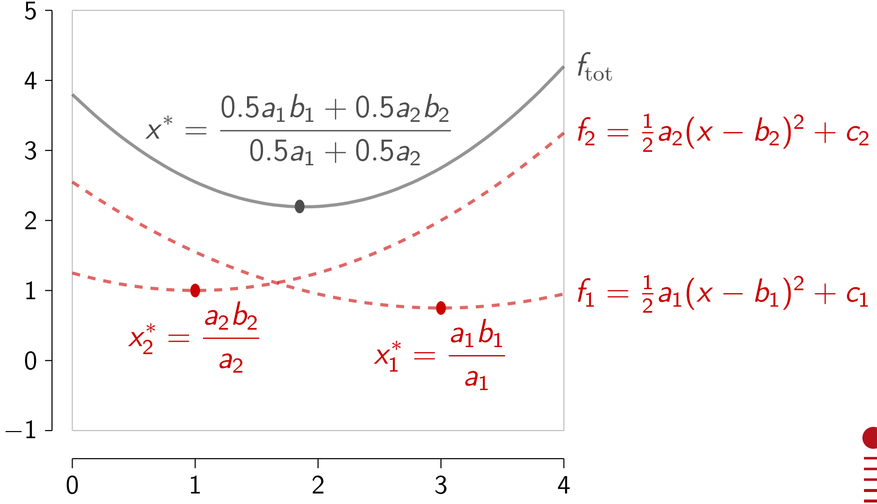
Corresponding global solution

$$x^* := \arg\min_x \sum_i f_i(x) \qquad\qquad x^* = \frac{\sum_{i=1}^{N} a_i b_i}{\sum_{i=1}^{N} a_i} = \frac{\frac{1}{N}\sum_{i=1}^{N} a_i b_i}{\frac{1}{N}\sum_{i=1}^{N} a_i}$$

i.e. ***parallel of 2 average consensus!***

Illustrative example: quadratic local cost functions

Derivation of the algorithm - step 1 on 3 - graphical interpretation



And for generic convex local cost functions?

Derivation of the algorithm - step 2 on 3

For quadratics ...

$$x^* = \frac{\frac{1}{N}\sum_{i=1}^{N} a_i b_i}{\frac{1}{N}\sum_{i=1}^{N} a_i}$$

with

- $a_i b_i = f_i''(x_i)x_i - f_i'(x_i)$
- $a_i = f_i''(x_i)$

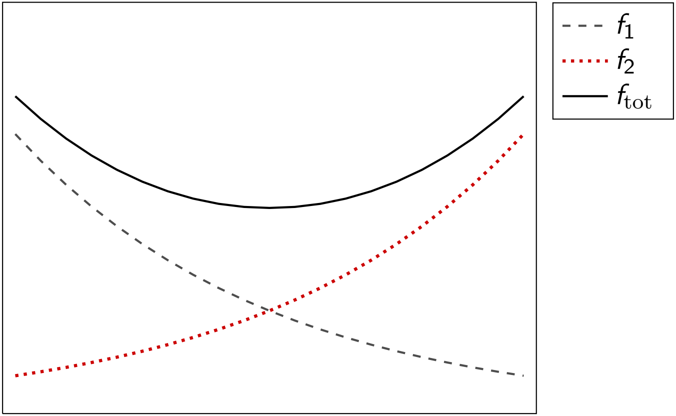
... so let's check

$$x^* \stackrel{?}{=} \frac{\frac{1}{N}\sum_{i=1}^{N} \left(f_i''(x_i)x_i - f_i'(x_i)\right)}{\frac{1}{N}\sum_{i=1}^{N} f_i''(x_i)}$$

underlying idea: use Newton-Raphson approximation

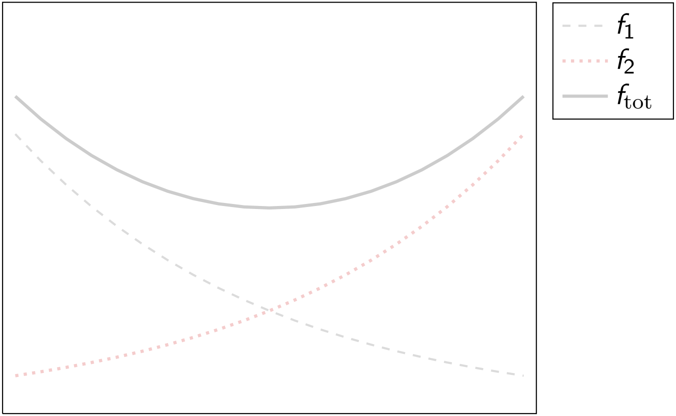
The initial idea

Derivation of the algorithm - step 2 on 3 - graphical interpretation



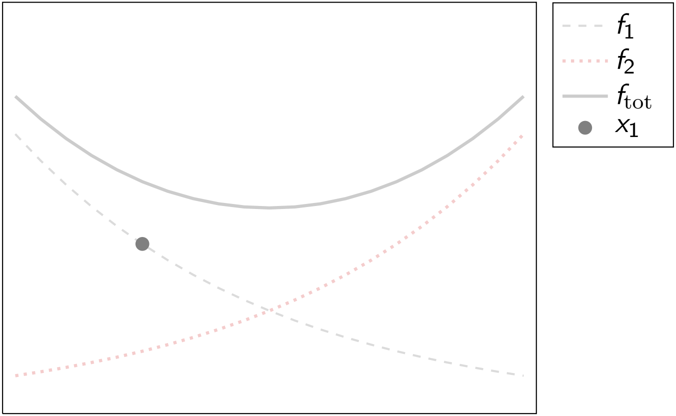
The initial idea

Derivation of the algorithm - step 2 on 3 - graphical interpretation



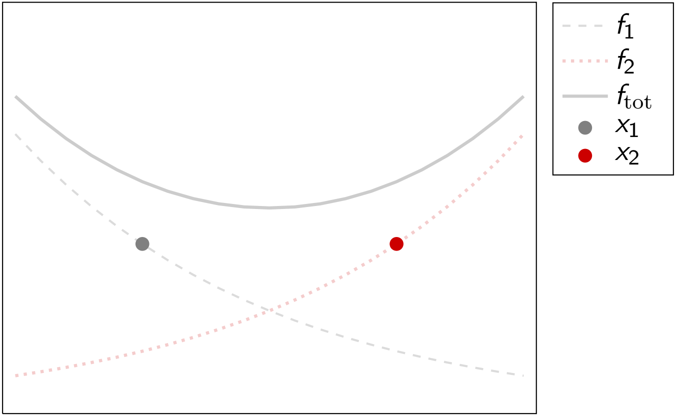
The initial idea

Derivation of the algorithm - step 2 on 3 - graphical interpretation



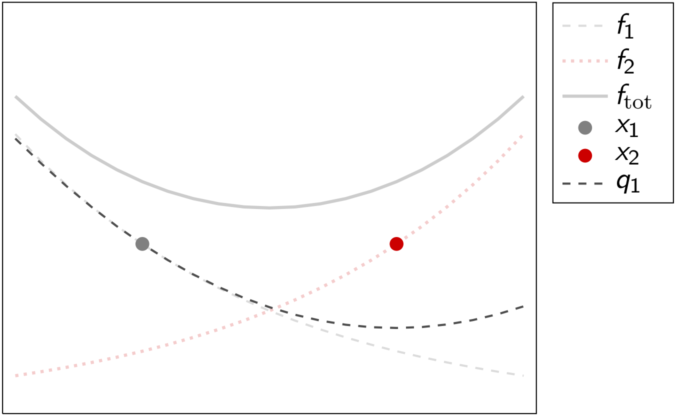
The initial idea

Derivation of the algorithm - step 2 on 3 - graphical interpretation



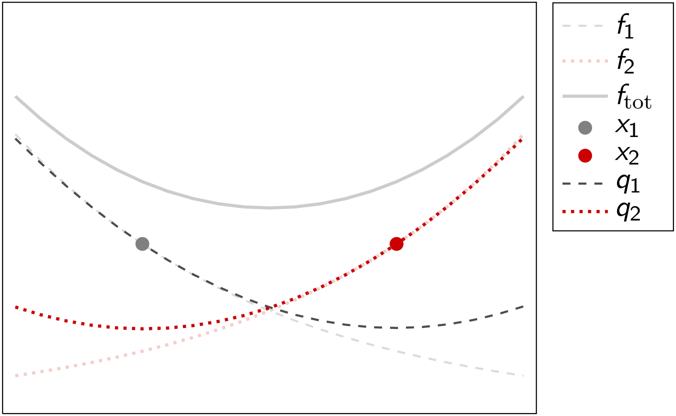
The initial idea

Derivation of the algorithm - step 2 on 3 - graphical interpretation



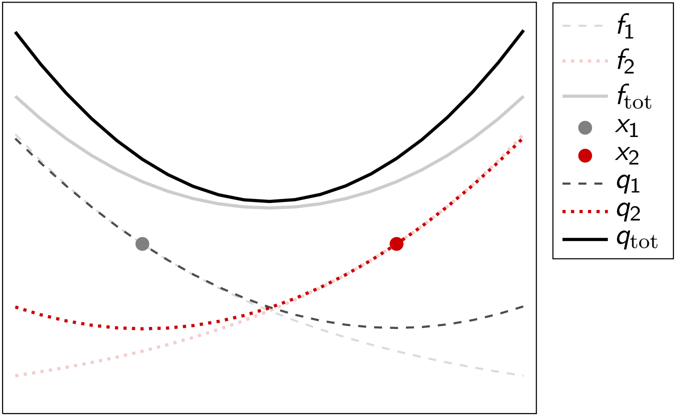
The initial idea

Derivation of the algorithm - step 2 on 3 - graphical interpretation



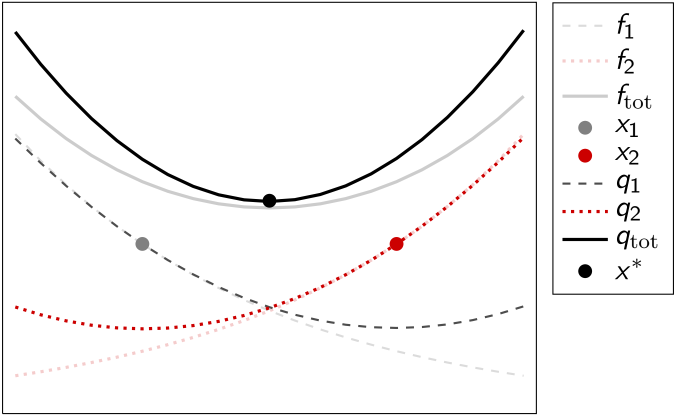
The initial idea

Derivation of the algorithm - step 2 on 3 - graphical interpretation



The initial idea

Derivation of the algorithm - step 2 on 3 - graphical interpretation



$$\text{candidate for } x^* = \frac{\frac{1}{N}\sum_{i=1}^{N} a_i b_i}{\frac{1}{N}\sum_{i=1}^{N} a_i} = \frac{\frac{1}{N}\sum_{i=1}^{N}\left(f_i''(x_i)x_i - f_i'(x_i)\right)}{\frac{1}{N}\sum_{i=1}^{N} f_i''(x_i)}$$

The initial idea

Derivation of the algorithm - step 3 on 3 - analysis of the problems

Does it work?

1. initialization:
   - $y_i(0) := f_i''(x_i(0))x_i(0) - f_i'(x_i(0))$
   - $z_i(0) := f_i''(x_i(0))$
2. ***average consensus*** (in $\|$, P doubly stochastic):
   - $\boldsymbol{y}(k+1) = P\boldsymbol{y}(k)$
   - $\boldsymbol{z}(k+1) = P\boldsymbol{z}(k)$
3. local updates: $x_i(k+1) = \dfrac{y_i(k+1)}{z_i(k+1)}$

The initial idea

Derivation of the algorithm - step 3 on 3 - analysis of the problems

Does it work?

1. initialization:
   - $y_i(0) := f_i''(x_i(0))x_i(0) - f_i'(x_i(0))$
   - $z_i(0) := f_i''(x_i(0))$
2. ***average consensus*** (in $\|$, P doubly stochastic):
   - $\boldsymbol{y}(k+1) = P\boldsymbol{y}(k)$
   - $\boldsymbol{z}(k+1) = P\boldsymbol{z}(k)$
3. local updates: $x_i(k+1) = \dfrac{y_i(k+1)}{z_i(k+1)}$

No, ***must provide 2 little modifications***:

The initial idea

Derivation of the algorithm - step 3 on 3 - analysis of the problems

Does it work?

1. initialization:
   - $y_i(0) := f_i''(x_i(0))x_i(0) - f_i'(x_i(0))$
   - $z_i(0) := f_i''(x_i(0))$
2. ***average consensus*** (in $\|$, P doubly stochastic):
   - $\boldsymbol{y}(k+1) = P\boldsymbol{y}(k)$
   - $\boldsymbol{z}(k+1) = P\boldsymbol{z}(k)$
3. local updates: $x_i(k+1) = \dfrac{y_i(k+1)}{z_i(k+1)}$

No, ***must provide 2 little modifications***:

- x_i changes $\Rightarrow$ must track the changing $f_i'(x_i)$ and $f_i''(x_i)$

Does it work?

1. initialization:
   - $y_i(0) := f_i''(x_i(0))x_i(0) - f_i'(x_i(0))$
   - $z_i(0) := f_i''(x_i(0))$
2. ***average consensus*** (in $\|$, P doubly stochastic):
   - $\boldsymbol{y}(k+1) = P\boldsymbol{y}(k)$
   - $\boldsymbol{z}(k+1) = P\boldsymbol{z}(k)$
3. local updates: $x_i(k+1) = \dfrac{y_i(k+1)}{z_i(k+1)}$

No, ***must provide 2 little modifications***:

- x_i changes $\Rightarrow$ must track the changing $f_i'(x_i)$ and $f_i''(x_i)$
- $x_i(k) = \dfrac{y_i(k)}{z_i(k)}$ too aggressive!! Should make it milder

The complete algorithm

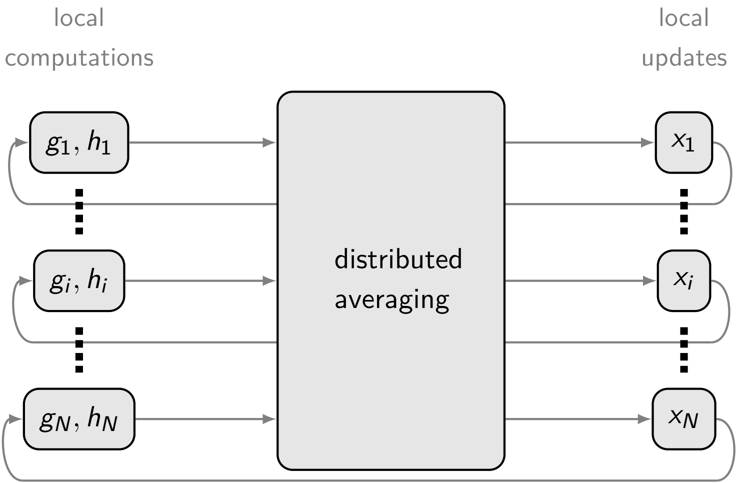
1. tracking:
   - $g_i(k) := f_i''(x_i(k))x_i(k) - f_i'(x_i(k))$
   - $h_i(k) := f_i''(x_i(k))$
2. ***average consensus*** (in $\|$, P doubly stochastic):
   - $\boldsymbol{y}(k+1) = P\left[\boldsymbol{y}(k) + \boldsymbol{g}(k) - \boldsymbol{g}(k-1)\right]$
   - $\boldsymbol{z}(k+1) = P\left[\boldsymbol{z}(k) + \boldsymbol{h}(k) - \boldsymbol{h}(k-1)\right]$
3. local updates: $x_i(k+1) = (1-\varepsilon)x_i(k) + \varepsilon\dfrac{y_i(k+1)}{z_i(k+1)}$

The complete algorithm

1. tracking:
   - $g_i(k) := f_i''(x_i(k))x_i(k) - f_i'(x_i(k))$
   - $h_i(k) := f_i''(x_i(k))$
2. ***average consensus*** (in $\|$, P doubly stochastic):
   - $\boldsymbol{y}(k+1) = P\left[\boldsymbol{y}(k) + \boldsymbol{g}(k) - \boldsymbol{g}(k-1)\right]$
   - $\boldsymbol{z}(k+1) = P\left[\boldsymbol{z}(k) + \boldsymbol{h}(k) - \boldsymbol{h}(k-1)\right]$
3. local updates: $x_i(k+1) = (1-\varepsilon)x_i(k) + \varepsilon \dfrac{y_i(k+1)}{z_i(k+1)}$

(numerical) remark: step 2 may be substituted with asymptotical average consensus algorithms

The complete algorithm – Block schematic representation



$$g_i(k) = f_i''(x_i(k))x_i(k) - f_i'(x_i(k))$$
$$h_i(k) = f_i''(x_i(k))$$

$$x_i(k+1) = (1-\varepsilon)x_i(k) + \varepsilon \frac{y_i(k+1)}{z_i(k+1)}$$

Convergence properties

Hypotheses

- $f_i \in \mathcal{C}^2(\mathbb{R})$
- f_i' and f_i'' bounded
- f_i strictly convex
- $x^* \neq \pm\infty$
- ***null initial conditions*** (for g_i, h_i, y_i, z_i)

Thesis

- there exists a positive $\bar{\varepsilon}$ s.t. if $\varepsilon < \bar{\varepsilon}$ then

$$\lim_{k \to +\infty} \mathbf{x}(k) = x^* \mathbb{1}$$

(convergence $\propto$ as Newton-Raphson strategies over $\overline{f}$)

Sketch of the proof

importance of the proof:
gives insights on key properties

1. transform the algorithm in a continuous-time system
2. recognize the existence of a two-time scales dynamical system
3. analyze separately fast and slow dynamics
   (singular perturbation methods [Khalil (2002)])

Properties

Good qualities

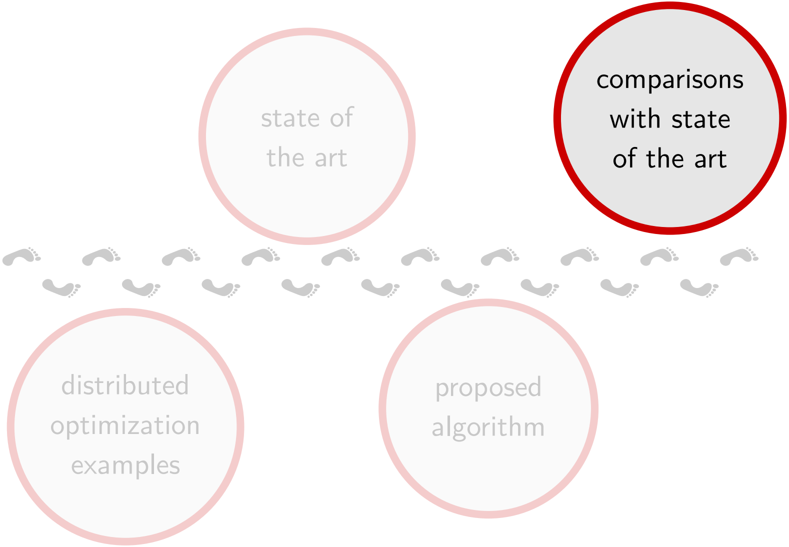
- easy to be implemented
- "small" computational requirements

Bad qualities

Up to now, requires strong assumptions:

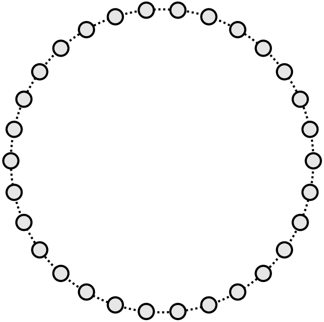
- $f_i \in \mathcal{C}^2(\mathbb{R})$
- f_i strictly convex
- f_i' and f_i'' bounded

Update


state of the art
comparisons with state of the art
distributed optimization examples
proposed algorithm

Experiments description

- circulant graph, $N = 30$
- $P = \begin{bmatrix} 0.5 & 0.25 & & & 0.25 \\ 0.25 & 0.5 & 0.25 & & \\ & \ddots & \ddots & \ddots & \\ & & 0.25 & 0.5 & 0.25 \\ 0.25 & & & 0.25 & 0.5 \end{bmatrix}$
- f_i = sum of exponentials

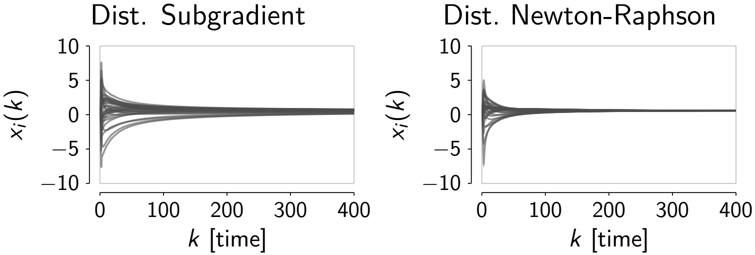


Comparisons with a Distributed Subgradient

Nedić Ozdaglar *Dist. subgr. meth. for multi-agent opt.* (2009)

1. $\mathbf{x}^{(c)}(k) = P\mathbf{x}(k)$ (consensus step)
2. $x_i(k+1) = x_i^{(c)}(k) - \frac{\rho}{k} f_i'\left(x_i^{(c)}(k)\right)$ (local gradient descent)

Numerical comparison

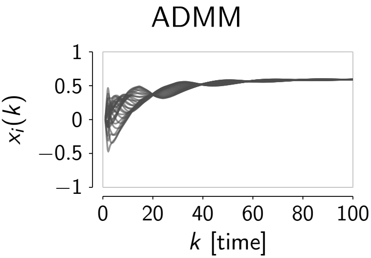


Comparisons with (an) ADMM

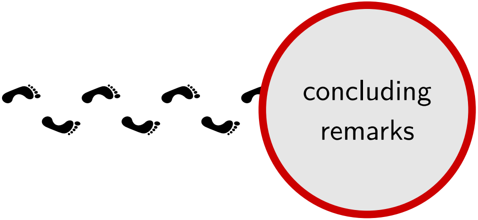
Bertsekas Tsitsiklis, *Parall. and Dist. Computation* (1997)

$$L_\rho \;:= \sum_i \Big[\; f_i\left(x_i\right) + y_i^{(\ell)}\left(x_i - z_{i-1}\right) + y_i^{(c)}\left(x_i - z_i\right) + y_i^{(r)}\left(x_i - z_{i+1}\right) + \tfrac{\delta}{2}\left|x_i - z_{i-1}\right|^2 + \tfrac{\delta}{2}\left|x_i - z_i\right|^2 + \tfrac{\delta}{2}\left|x_i - z_{i+1}\right|^2 \Big]$$

Numerical comparison



Update



Conclusions and future works

The algorithm we proposed ...

- is a distributed Newton-Raphson strategy (+)
- requires minimal network topology knowledge (+)
- requires minimal agents synchronization (+)
- is simple to be implemented (+)
- converges to global optimum under convexity and smoothness assumptions (+ / -)
- is numerically faster than subgradients (+) but slower than ADMM (-)

Conclusions and future works

Currently working on (or already performed)

- extension to multi-dimensional problems
- extension to modified Newton strategies
- analytical characterization of the convergence speed for quadratic functions and specific graphs
  (with comparisons to other methods)
- relax the assumptions
  (strict convexity, $\mathcal{C}^2$, ...)
- find automatic stepsizes tuning strategies
- propose quasi-Newton strategies

K. C. Kiwiel (2004)
Convergence of approximate and incremental subgradient methods for convex optimization
SIAM Journal on Optimization

D. P. Bertsekas (1982)
Constrained Optimization and Lagrange Multiplier Methods
Academic Press

D. P. Bertsekas and J. N. Tsitsiklis (1997)
Parallel and Distributed Computation: Numerical Methods
Athena Scientific

A. Nedić and A. Ozdaglar (2009)
Distributed subgradient methods for multi-agent optimization
IEEE Transactions on Automatic Control

B. Johansson (2008)
On Distributed Optimization in Networked Systems
Ph.D. Thesis, KTH

A. Nedić and A. Ozdaglar (2007)
On the Rate of Convergence of Distributed Subgradient Methods for Multi-agent Optimization
CDC

S. Boyd, N. Parikh, E. Chu, B. Peleato, J. Eckstein (2010)
Distributed Optimization and Statistical Learning via the Alternating Direction Method of Multipliers
Foundations and Trends in Machine Learning

M. Zargham, A. Ribeiro, A. Ozdaglar, A. Jadbabaie (2011)
Accelerated Dual Descent for Network Optimization
ACC

D. P. Bertsekas (2011)
Centralized and Distributed Newton Methods for Network Optimization and Extensions
Technical Report LIDS 2866

H. K. Khalil (2002)
Nonlinear Systems
Prentice Hall

Distributed convex optimization: a consensus-based Newton-Raphson approach

Damiano Varagnolo

joint work with A. Cenedese, G. Pillonetto, L. Schenato, F. Zanella

Department of Information Engineering - University of Padova

December 14^{th}, 2011 – 50^{th} IEEE CDC

varagnolo@dei.unipd.it

www.dei.unipd.it/∼varagnolo/

google: damiano varagnolo