Distributed optimization through Newton-Raphson consensus

Damiano Varagnolo

joint work with Luca Schenato and Filippo Zanella

School of Electrical Engineering - KTH Royal Institute of Technology

June 7, 2012 – Netcon Meeting

Distributed optimization

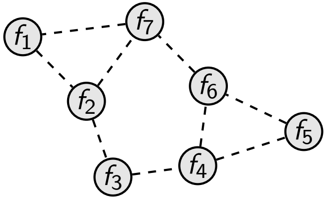
Problem formulation

$$\begin{array}{ll} \text{minimize} & f(x) = \sum_{i=1}^{N} f_i(x) \\ \text{s.t.} & g(x) \leq 0 \\ & x \in \mathcal{X} \end{array}$$

convexity assumptions

Multi-agents scenario

cooperation to find the optimum



Our position in literature

- primal based
- unconstrained convex
- uses second-order approximations
- uses 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

Simplified scalar scenario

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

Corresponding solution

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

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

And for generic convex local cost functions?

For quadratics ...

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

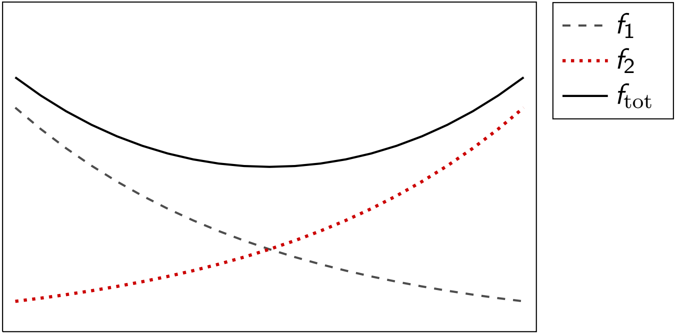
with

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

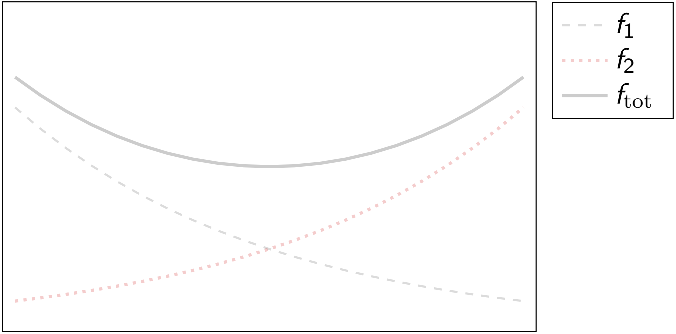
... so let's check

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

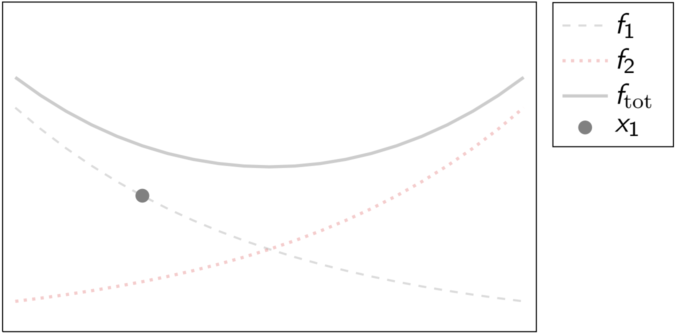
Graphical interpretation



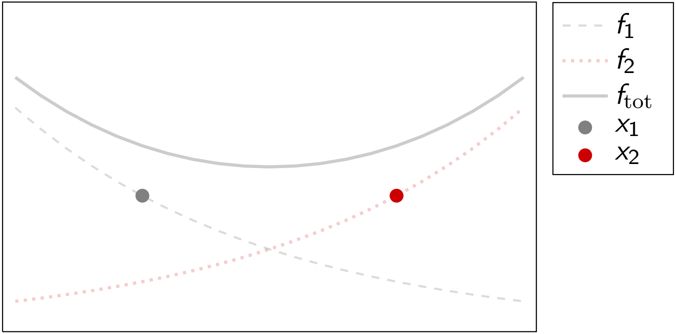
Graphical interpretation



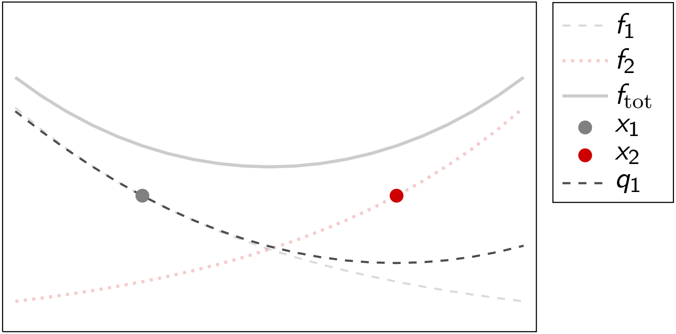
Graphical interpretation



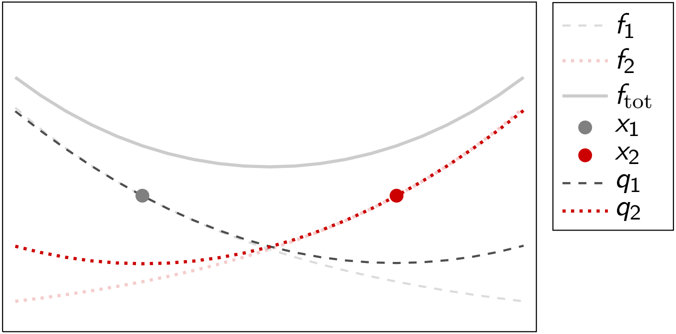
Graphical interpretation



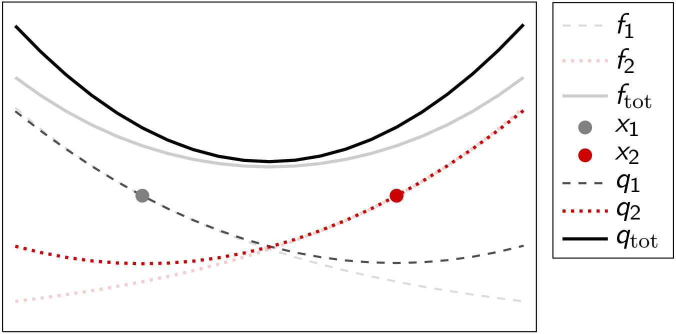
Graphical interpretation



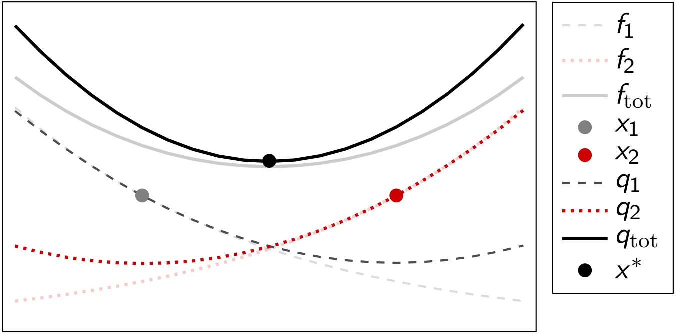
Graphical interpretation



Graphical interpretation



Graphical interpretation



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

intuition: it is a Newton-Raphson approximation

The complete algorithm – synchronous case

1. quadratic approximations update:
   - $g_i(k) := f_i''(x_i(k))x_i(k) - f_i'(x_i(k))$
   - $h_i(k) := f_i''(x_i(k))$
2. quadratic approximations mixing (av. consensus, 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. guesses updates ($\div$ component-wise):
   - $\boldsymbol{x}(k+1) = (1-\varepsilon)\boldsymbol{x}(k) + \varepsilon \dfrac{\boldsymbol{y}(k+1)}{\boldsymbol{z}(k+1)}$

Towards an asynchronous version ...

$$N(k) = \begin{bmatrix} 0 & & & & & \\ & 1 & & & & \\ & & 0 & & & \\ & & & 0 & & \\ & & & & 0 & \\ & & & & & 0 \end{bmatrix}$$

Towards an asynchronous version . . .

$$N(k) = \begin{bmatrix} 0 & & & & & \\ & 1 & & & & \\ & & 0 & & & \\ & & & 0 & & \\ & & & & 0 & \\ & & & & & 0 \end{bmatrix} \qquad E(k) = \begin{bmatrix} 0 & & & & & \\ & 1 & & & & \\ & & 0 & & & \\ & & & 0 & & \\ & & & & 1 & \\ & & & & & 0 \end{bmatrix}$$

Towards an asynchronous version ...

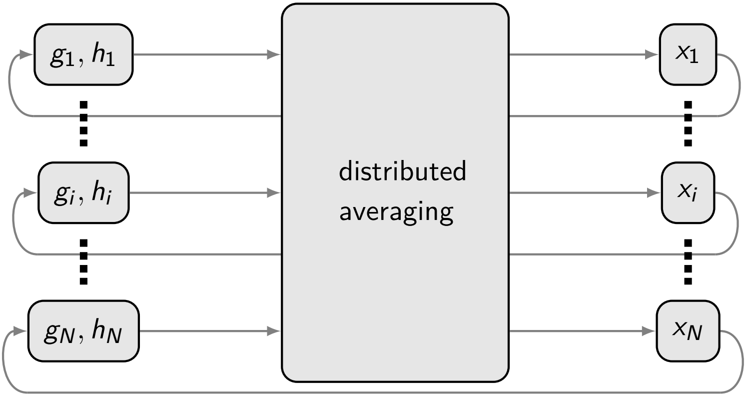
$$N(k) = \begin{bmatrix} 0 & & & & & \\ & 1 & & & & \\ & & 0 & & & \\ & & & 0 & & \\ & & & & 0 & \\ & & & & & 0 \end{bmatrix} \qquad E(k) = \begin{bmatrix} 0 & & & & & \\ & 1 & & & & \\ & & 0 & & & \\ & & & 0 & & \\ & & & & 1 & \\ & & & & & 0 \end{bmatrix}$$

$$P(k) = \begin{bmatrix} 0 & & & & & \\ & 1-\alpha & & & \alpha & \\ & & 0 & & & \\ & & & 0 & & \\ & \alpha & & & 1-\alpha & \\ & & & & & 0 \end{bmatrix}$$

The complete algorithm – asynchronous case

1. quadratic approximations update:
   - $g_i(k) := f_i''(x_i(k))x_i(k) - f_i'(x_i(k))$
   - $h_i(k) := f_i''(x_i(k))$
2. quadratic approximations mixing:
   - $\boldsymbol{y}(k+1) = P(k)\left[\boldsymbol{y}(k) + E(k)\Big(\boldsymbol{g}(k) - \boldsymbol{g}(k-1)\Big)\right]$
   - $\boldsymbol{z}(k+1) = P(k)\left[\boldsymbol{z}(k) + E(k)\Big(\boldsymbol{h}(k) - \boldsymbol{h}(k-1)\Big)\right]$
3. guesses updates:
   - $\boldsymbol{x}(k+1) = \boldsymbol{x}(k) + \varepsilon N(k)\left(\dfrac{\boldsymbol{y}(k+1)}{\boldsymbol{z}(k+1)} - \boldsymbol{x}(k)\right)$

Block schematic representation



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

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

need just uniformly exponentially converging av. consensus

Convergence properties - 1/3

Hypotheses on the local costs

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

Convergence properties - 2/3

Theorem

uniform activation$^{(1)}$ $\Rightarrow$ ***global convergence***$^{(2)}$

Convergence properties - 2/3

Theorem

uniform activation$^{(1)}$ $\Rightarrow$ ***global convergence***$^{(2)}$

(1): on the long run all the nodes are activated
the same number of times

Convergence properties - 2/3

Theorem

uniform activation$^{(1)}$ $\Rightarrow$ ***global convergence***$^{(2)}$

(1): on the long run all the nodes are activated the same number of times

(2): for every open ball B_r centered in $\boldsymbol{x}^*$

Convergence properties - 2/3

Theorem

uniform activation$^{(1)}$ $\Rightarrow$ ***global convergence***$^{(2)}$

(1): on the long run all the nodes are activated the same number of times

(2): for every open ball B_r centered in $\boldsymbol{x}^*$ exists $\overline{\varepsilon}_r > 0$ s.t.

Convergence properties - 2/3

Theorem

uniform activation$^{(1)}$ $\Rightarrow$ ***global convergence***$^{(2)}$

(1): on the long run all the nodes are activated
the same number of times

(2): for every open ball B_r centered in $\boldsymbol{x}^*$
exists $\bar{\varepsilon}_r > 0$ s.t.
for all $\varepsilon < \bar{\varepsilon}_r$

Convergence properties - 2/3

Theorem

uniform activation$^{(1)}$ $\Rightarrow$ ***global convergence***$^{(2)}$

(1): on the long run all the nodes are activated the same number of times

(2): for every open ball B_r centered in $\boldsymbol{x}^*$
exists $\bar{\varepsilon}_r > 0$ s.t.
for all $\varepsilon < \bar{\varepsilon}_r$
exist $c_r, \gamma_\varepsilon > 0$ s.t.

$$\|\boldsymbol{x}_k - \boldsymbol{x}^*\| \leq \|\boldsymbol{x}_0 - \boldsymbol{x}^*\| \cdot c_r e^{-\gamma_\varepsilon k} \qquad \forall \boldsymbol{x}_0 \in B_r$$

Convergence properties - 3/3

Theorem

$$\textit{\textbf{persistent activation}}^{(1)} \Rightarrow \textit{\textbf{local convergence}}^{(2)}$$

Convergence properties - 3/3

Theorem

persistent activation$^{(1)}$ $\Rightarrow$ ***local convergence***$^{(2)}$

(1): bounded intercommunication intervals

Convergence properties - 3/3

Theorem

persistent activation$^{(1)}$ $\Rightarrow$ ***local convergence***$^{(2)}$

(1): bounded intercommunication intervals

(2): exists an open ball B_0 centered in $\boldsymbol{x}^*$ s.t.

Convergence properties - 3/3

Theorem

persistent activation$^{(1)} \Rightarrow$ ***local convergence***$^{(2)}$

(1): bounded intercommunication intervals

(2): exists an open ball B_0 centered in $\boldsymbol{x}^*$ s.t.
exists $\overline{\varepsilon} > 0$ s.t.

Convergence properties - 3/3

Theorem

persistent activation$^{(1)}$ $\Rightarrow$ ***local convergence***$^{(2)}$

(1): bounded intercommunication intervals

(2): exists an open ball B_0 centered in $\boldsymbol{x}^*$ s.t.
exists $\overline{\varepsilon} > 0$ s.t.
for all $\varepsilon < \overline{\varepsilon}$

Convergence properties - 3/3

Theorem

persistent activation$^{(1)}$ $\Rightarrow$ ***local convergence***$^{(2)}$

(1): bounded intercommunication intervals

(2): exists an open ball B_0 centered in $\boldsymbol{x}^*$ s.t.
exists $\overline{\varepsilon} > 0$ s.t.
for all $\varepsilon < \overline{\varepsilon}$
exist $c, \gamma_\varepsilon > 0$ s.t.

$$\|\boldsymbol{x}_k - \boldsymbol{x}^*\| \leq \|\boldsymbol{x}_0 - \boldsymbol{x}^*\| \cdot c e^{-\gamma_\varepsilon k} \qquad \forall \boldsymbol{x}_0 \in B_0$$

Sketch of the proofs

1. rewrite the algorithm to highlight two-time scales dynamics
2. analyze separately fast and slow dynamics
   (discrete version of standard singular perturbation analysis)
3. analysis of boundary layer:
   - requires an exponentially convergent average consensus
   - use discrete converse Lyapunov theorems
4. analysis of reduced system:
   - exploit averaging to remove the dependency on $N(k)$'s
   - massage av. consensus equations + exploit smoothness assumptions on the f_i's to obtain a Lyapunov function

Properties

Good:

- easy implementation
- "small" computational requirements
- inherits qualities of consensus:
  - small topological knowledge requirements
  - robust to numerical errors and communication noise

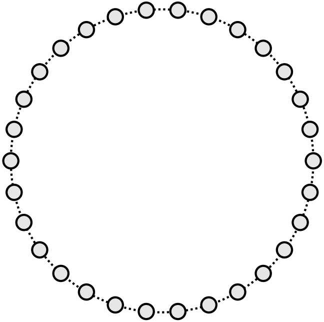
Bad:

strong assumptions:

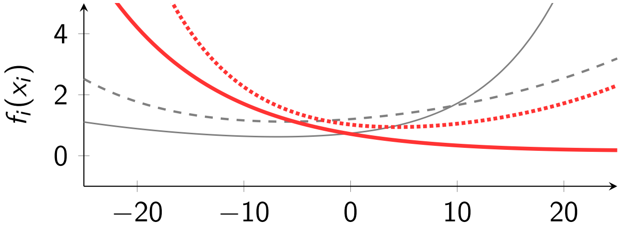
- $f_i \in \mathcal{C}^2(\mathbb{R})$
- f_i strictly convex
- f_i' and f_i'' bounded

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

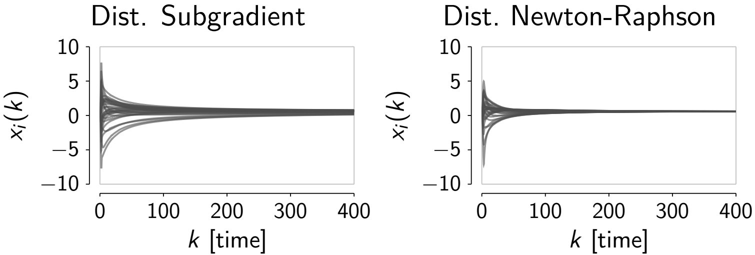


Comparisons with a Distributed Subgradient

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

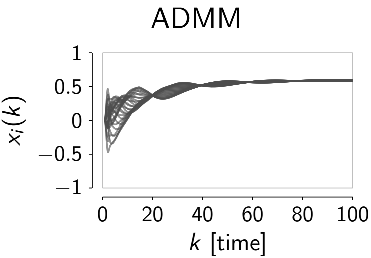


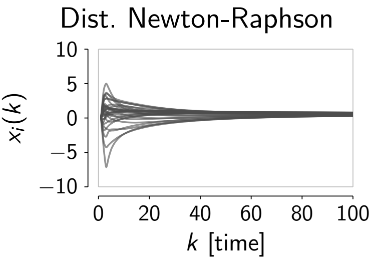
Comparisons with (an) ADMM

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

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

Numerical comparison





Conclusions and future works

The algorithm we proposed ...

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

Conclusions and future works

Principal open problems

- analytically characterize the convergence speeds for specific functions and graphs
  (with comparisons to other methods)

- relax the assumptions
  (strict convexity, $\mathcal{C}^2$, ...)

- tune ε on-line

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 optimization through Newton-Raphson consensus

Damiano Varagnolo

joint work with Luca Schenato and Filippo Zanella

School of Electrical Engineering - KTH Royal Institute of Technology

June 7, 2012 – Netcon Meeting

entirely written in LaTeX2ε using Beamer and Tik Z