Fast-Lipschitz Optimization for Non-convex and Convex Problems

Bellairs
McGill University

February 18-th, 2013

Carlo Fischione

ACCESS Linnaeus Center, Electrical Engineering
KTH Royal Institute of Technology
Stockholm, Sweden
web: http://www.ee.kth.se/~carlofi
e-mail: carlofi@kth.se

Optimization is pervasive over networks

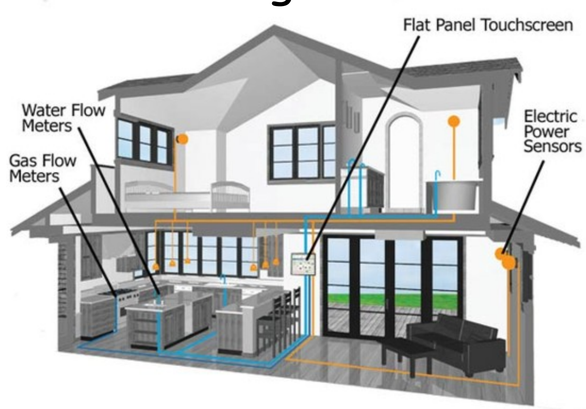
Parallel computing

Environmental monitoring

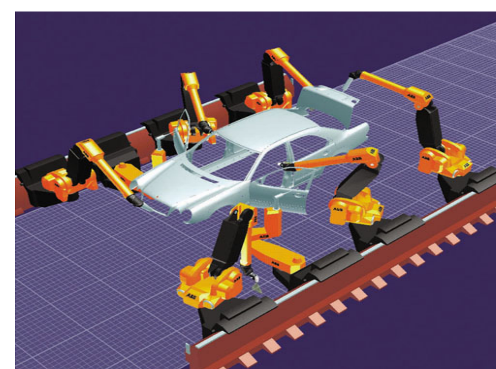
Intelligent transportation systems

Smart grids

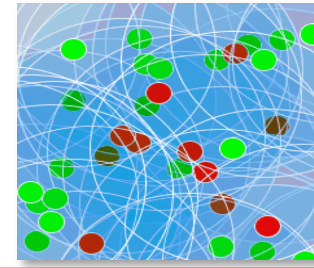
Smart buildings



Industrial control

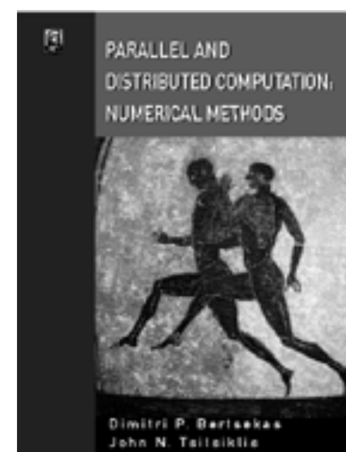


Optimization over networks



- Optimization needs fast solver algorithms of low complexity
  - Time-varying networks, little time to compute solution
  - Distributed computations
  - E.g., networks of parallel processors, cross layer networking, distributed detection, estimation, content distribution, ...

- Parallel and distributed computation
  - Fundamental theory for optimization over networks
  - Drawback over energy-constrained wireless networks: the cost for communication not considered

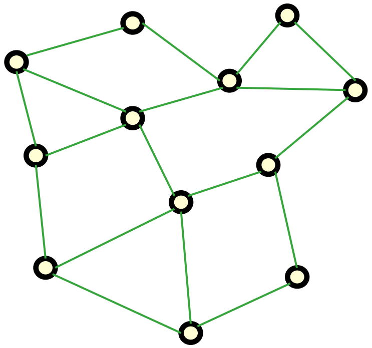


- An alternative theory is needed
  - In a number of cases, Fast-Lipschitz optimization

Outline

- **Motivating example: distributed detection**
- Definition of Fast-Lipschitz optimization
- Computation of the optimal solution
- Problems in canonical form
- Examples
- Conclusions

Distributed binary detection



$$\Gamma_i(s) = w_i(s) \quad \text{if} \quad H_0$$
$$\Gamma_i(s) = E + w_i(s) \quad \text{if} \quad H_1$$
measurements at node i

$$T_i = \frac{1}{S}\sum_{s=1}^{S} \Gamma_i(s) \gtreqless x_i$$
hypothesis testing with S measurements and threshold x_i

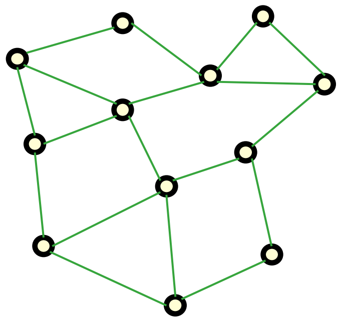
$$P_{\text{fa}}^{(i)}(x_i) = \Pr[T_i > x_i | H_0]$$
probability of false alarm

$$P_{\text{md}}^{(i)}(x_i) = \Pr[T_i \leq x_i | H_1]$$
probability of misdetection

- A threshold minimizing the prob. of false alarm maximizes the prob. of misdetection.
- How to choose optimally the thresholds when nodes exchange opinions?

Threshold optimization in distributed detection

$$
\begin{aligned}
\min_{\boldsymbol{x}} \quad & \sum_{i=1}^{n} P_{\mathrm{fa}}^{(i)}(x_i) \\
\text{s.t.} \quad & \sum_{j=1}^{n} b_{i,j} P_{\mathrm{md}}^{(j)}(x_j) \leq c_i, \quad i = 1, \ldots, n\,, \\
& 0 \preceq \boldsymbol{x} \preceq E\mathbf{1}\,.
\end{aligned}
$$



- How to solve the problem by distributed operations among the nodes?
- The problem is convex
  - Lagrangian methods (interior point) can be applied
  - Drawback: too many message passing (Lagrangian multipliers) among nodes to compute iteratively the optimal solution
- An alternative method: Fast-Lipschitz optimization

C. Fischione, "Fast-Lipschitz Optimization with Wireless Sensor Networks Applications", *IEEE TAC*, 2011

Outline

- Motivating example: distributed detection
- **Definition of Fast-Lipschitz optimization**
- Computation of the optimal solution
- Problems in canonical form
- Examples
- Conclusions

The Fast-Lipschitz optimization

$$
\begin{aligned}
\max_{\boldsymbol{x}} \quad & f_0(\boldsymbol{x}) \\
\text{s.t.} \quad & x_i \leq f_i(\boldsymbol{x}) \,, \quad i = 1, \ldots, l \\
& x_i = h_i(\boldsymbol{x}) \,, \quad i = l+1, \ldots, n \\
& \boldsymbol{x} \in \mathscr{D} \,,
\end{aligned}
$$

$$f_0(\boldsymbol{x}) : \mathscr{D} \to \mathbb{R} \,,$$

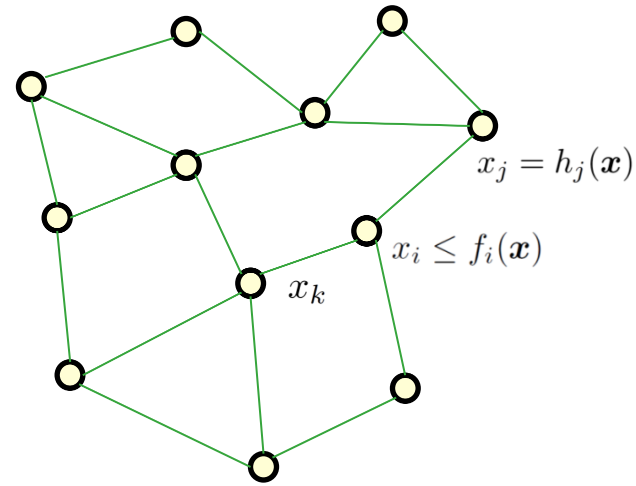
$$f_i(\boldsymbol{x}) : \mathscr{D} \to \mathbb{R} \,, \qquad i = 1, \ldots, l$$

$$h_i(\boldsymbol{x}) : \mathscr{D} \to \mathbb{R} \,, \qquad i = l+1, \ldots, n$$

$\mathscr{D} \subset \mathbb{R}^n$ nonempty compact set containing the vertexes of the constraints

Computation of the solution

$$\begin{aligned} \max_{\boldsymbol{x}} \quad & f_0(\boldsymbol{x}) \\ \text{s.t.} \quad & x_i \le f_i(\boldsymbol{x}) \,, \quad i = 1, \dots, l \\ & x_i = h_i(\boldsymbol{x}) \,, \quad i = l+1, \dots, n \\ & \boldsymbol{x} \in \mathscr{D} \,, \end{aligned}$$



Network of n nodes

- Centralized optimization
  - Problem solved by a central processor

- Distributed optimization
  - Decision variables and constraints are associated to nodes that cooperate to compute the solution in parallel

Pareto Optimal Solution

$$\begin{aligned} \max_{\boldsymbol{x}} \quad & f_0(\boldsymbol{x}) \\ \text{s.t.} \quad & x_i \leq f_i(\boldsymbol{x}), \quad i = 1, \ldots, l \\ & x_i = h_i(\boldsymbol{x}), \quad i = l+1, \ldots, n \\ & \boldsymbol{x} \in \mathscr{D}, \end{aligned}$$

Definition : Consider the following set

$$\begin{aligned} \mathscr{A} = \{ & \boldsymbol{x} \in \mathscr{D} : x_i \leq f_i(\boldsymbol{x}), i = 1, \ldots, l, \\ & x_i = h_i(\boldsymbol{x}), i = l+1, \ldots, n \}, \end{aligned}$$

and let $\mathscr{B} \in \mathbb{R}^l$ be the image set of $f_0(\boldsymbol{x})$, namely $f_0(\boldsymbol{x}) : \mathscr{A} \to \mathscr{B}$. Then, we make the natural assumption that the set $\mathscr{B}$ is partially ordered in a natural way, namely if $\boldsymbol{x}, \mathbf{y} \in \mathscr{B}$ then $\boldsymbol{x} \succeq \mathbf{y}$ if $x_i \geq y_i \ \forall i$ (e.g., $\mathbb{R}^l_+$ is the ordering cone).

Definition (Pareto Optimal): A vector $\boldsymbol{x}^*$ is called a Pareto optimal (or an Edgeworth-Pareto optimal) point if there is no $\boldsymbol{x} \in \mathscr{A}$ such that $f_0(\boldsymbol{x}) \succeq f_0(\boldsymbol{x}^*)$ (i.e., if $f_0(\boldsymbol{x}^*)$ is the maximal element of the set $\mathscr{B}$ with respect to the natural partial ordering defined by the cone $\mathbb{R}^l_+$).

Notation

$$\mathbf{F}(\boldsymbol{x}) = \begin{bmatrix} F_1(\boldsymbol{x}) \\ F_2(\boldsymbol{x}) \\ \vdots \\ F_n(\boldsymbol{x}) \end{bmatrix} \quad F_i(\boldsymbol{x}) : \mathbb{R}^n \to \mathbb{R} \ \forall i$$

$$\nabla \mathbf{F}(\boldsymbol{x}) = \begin{bmatrix} \frac{dF_1(\boldsymbol{x})}{dx_1} & \frac{dF_2(\boldsymbol{x})}{dx_1} & \cdots & \frac{dF_n(\boldsymbol{x})}{dx_1} \\ \frac{dF_1(\boldsymbol{x})}{dx_2} & \frac{dF_2(\boldsymbol{x})}{dx_2} & \cdots & \frac{dF_n(\boldsymbol{x})}{dx_2} \\ \vdots & \vdots & \ddots & \cdots \\ \frac{dF_1(\boldsymbol{x})}{dx_n} & \frac{dF_2(\boldsymbol{x})}{dx_n} & \cdots & \frac{dF_n(\boldsymbol{x})}{dx_n} \end{bmatrix}$$

Gradient

$$|\nabla \mathbf{F}(\boldsymbol{x})|_\infty = \max_j \sum_{i=1}^n \left| \frac{dF_i(\boldsymbol{x})}{dx_j} \right|$$

Norm infinity: sum along a row

$$|\nabla \mathbf{F}(\boldsymbol{x})|_1 = \max_j \sum_{i=1}^n \left| \frac{dF_j(\boldsymbol{x})}{dx_i} \right|$$

Norm 1: sum along a column

Qualifying conditions

Now that we have introduced basic notation and concepts, we give some conditions for which a problem is Fast-Lipschitz

Qualifying conditions

1.a $\nabla f_0(x) \succ 0$, *i.e., $f_0(x)$ is strictly increasing,*

1.b $|\nabla \mathbf{F}(x)|_\infty < 1$,

and either

2.a $\nabla_j F_i(x) \geq 0 \quad \forall i, j$,

or

3.a $\nabla_i f_0(x) = \nabla_j f_0(x)$,

3.b $\nabla_j F_i(x) \leq 0 \quad \forall i, j$,

3.c $|\nabla \mathbf{F}(x)|_1 < 1$,

or

4.a $f_0(x) \in \mathbb{R}$,

4.b $|\nabla \mathbf{F}(x)|_1 \leq \dfrac{\delta}{\delta + \Delta}$,

$$\delta = \min_{i, \boldsymbol{x} \in \mathscr{D}} \nabla_i f_0(x),$$

$$\Delta = \max_{i, \boldsymbol{x} \in \mathscr{D}} \nabla_i f_0(x).$$

$$\begin{aligned}
\max_{\boldsymbol{x}} \quad & f_0(\boldsymbol{x}) \\
\text{s.t.} \quad & x_i \leq f_i(\boldsymbol{x}), \quad i = 1, \ldots, l \\
& x_i = h_i(\boldsymbol{x}), \quad i = l+1, \ldots, n \\
& \boldsymbol{x} \in \mathscr{D},
\end{aligned}$$

$$\mathbf{f}(\boldsymbol{x}) = [f_1(\boldsymbol{x}), f_2(\boldsymbol{x}), \ldots, f_l(\boldsymbol{x})]^T$$

$$\mathbf{h}(\boldsymbol{x}) = [h_{l+1}(\boldsymbol{x}), h_{l+2}(\boldsymbol{x}), \ldots, h_n(\boldsymbol{x})]^T$$

$$\mathbf{F}(\boldsymbol{x}) = [F_i(\boldsymbol{x})] = [\mathbf{f}(\boldsymbol{x})^T \ \mathbf{h}(\boldsymbol{x})^T]^T$$

Functions may be non-convex

Outline

- Motivating example: distributed detection
- Definition of Fast-Lipschitz optimization
- **Computation of the optimal solution**
- Problems in canonical form
- Examples
- Conclusions

Optimal Solution

$$\begin{aligned} \max_{\boldsymbol{x}} \quad & f_0(\boldsymbol{x}) \\ \text{s.t.} \quad & x_i \le f_i(\boldsymbol{x}), \quad i = 1, \ldots, l \\ & x_i = h_i(\boldsymbol{x}), \quad i = l+1, \ldots, n \\ & \boldsymbol{x} \in \mathscr{D}, \end{aligned}$$

Theorem: Let an F-Lipschitz optimization problem be feasible. Then, the problem admits a unique optimum $\boldsymbol{x}^* \in \mathscr{D}$ given by the solution of the set of equations

$$\begin{aligned} x_i^* &= f_i(\boldsymbol{x}^*) \quad i = 1, \ldots, l \\ x_i^* &= h_i(\boldsymbol{x}^*) \quad i = l+1, \ldots, n\,. \end{aligned}$$

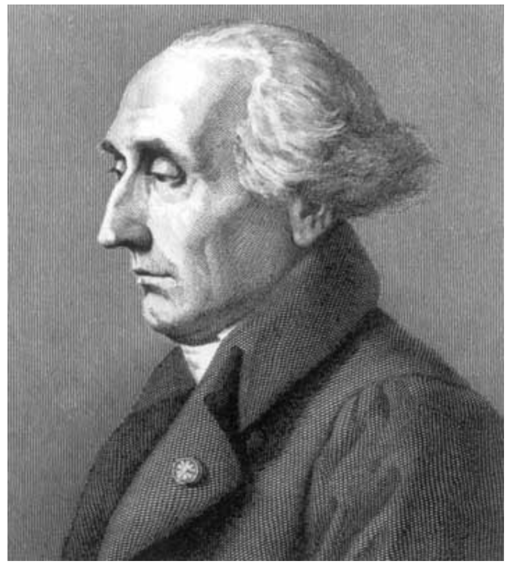
- The Pareto optimal solution is just given by a set of (in general non-linear) equations.
- Solving a set of equations is much easier than solving an optimization problem by traditional Lagrangian methods!

Lagrangian methods

- Let's have a closer look at the Lagrangian methods, which are normally used to solve optimization problems

- Lagrangian methods are the essential to solve, for example, convex problems

Lagrangian methods



G. L. Lagrange, 1736-1813

"The methods I set forth require neither constructions nor geometric or mechanical considerations. They require only algebraic operations subject to a systematic and uniform course"

Lagrangian methods

$$\begin{aligned}\max_{\boldsymbol{x}} \quad & f_0(\boldsymbol{x}) \\ \text{s.t.} \quad & x_i \le f_i(\boldsymbol{x})\,, \quad i = 1, \dots, l \\ & x_i = h_i(\boldsymbol{x})\,, \quad i = l+1, \dots, n \\ & \boldsymbol{x} \in \mathscr{D}\,,\end{aligned}$$

Theorem: Consider a feasible F-Lipschitz problem. Then, the KKT conditions are necessary and sufficient.

- KKT conditions:

$$\begin{aligned}& x_i - f_i(\boldsymbol{x}^*) \le 0 \quad i = 1, \dots, l \\ & x_i - h_i(\boldsymbol{x}^*) = 0 \quad i = l+1, \dots, n \\ & \lambda_i^* \ge 0 \qquad\qquad i = 1, \dots, n \\ & \lambda_i^* f_i(\boldsymbol{x}^*) = 0 \quad i = 1, \dots, n \\ & \nabla_{\boldsymbol{x}} L(\boldsymbol{x}^*, \boldsymbol{\lambda}^*) = 0\,,\end{aligned}$$

$$L(\boldsymbol{x}, \boldsymbol{\lambda}) = -f_0(\boldsymbol{x}) + \sum_{i=1}^{l} \lambda_i (x_i - f_i(\boldsymbol{x})) + \sum_{i=l+1}^{n} \lambda_i (x_i - h_i(\boldsymbol{x}))$$

Lagrangian

$$\begin{aligned}\boldsymbol{x}(k+1) &= \boldsymbol{x}(k) - \beta \nabla_{\boldsymbol{x}} L(\boldsymbol{x}(k), \boldsymbol{\lambda}(k)) \\ \boldsymbol{\lambda}(k+1) &= \boldsymbol{\lambda}(k) - \beta \nabla_{\boldsymbol{\lambda}} L(\boldsymbol{x}(k), \boldsymbol{\lambda}(k))\end{aligned}$$

Lagrangian methods to compute the solution

Lagrangian methods

$$
\begin{aligned}
\max_{\boldsymbol{x}} \quad & f_0(\boldsymbol{x}) \\
\text{s.t.} \quad & x_i \leq f_i(\boldsymbol{x}), \quad i = 1, \dots, l \\
& x_i = h_i(\boldsymbol{x}), \quad i = l+1, \dots, n \\
& \boldsymbol{x} \in \mathscr{D},
\end{aligned}
$$

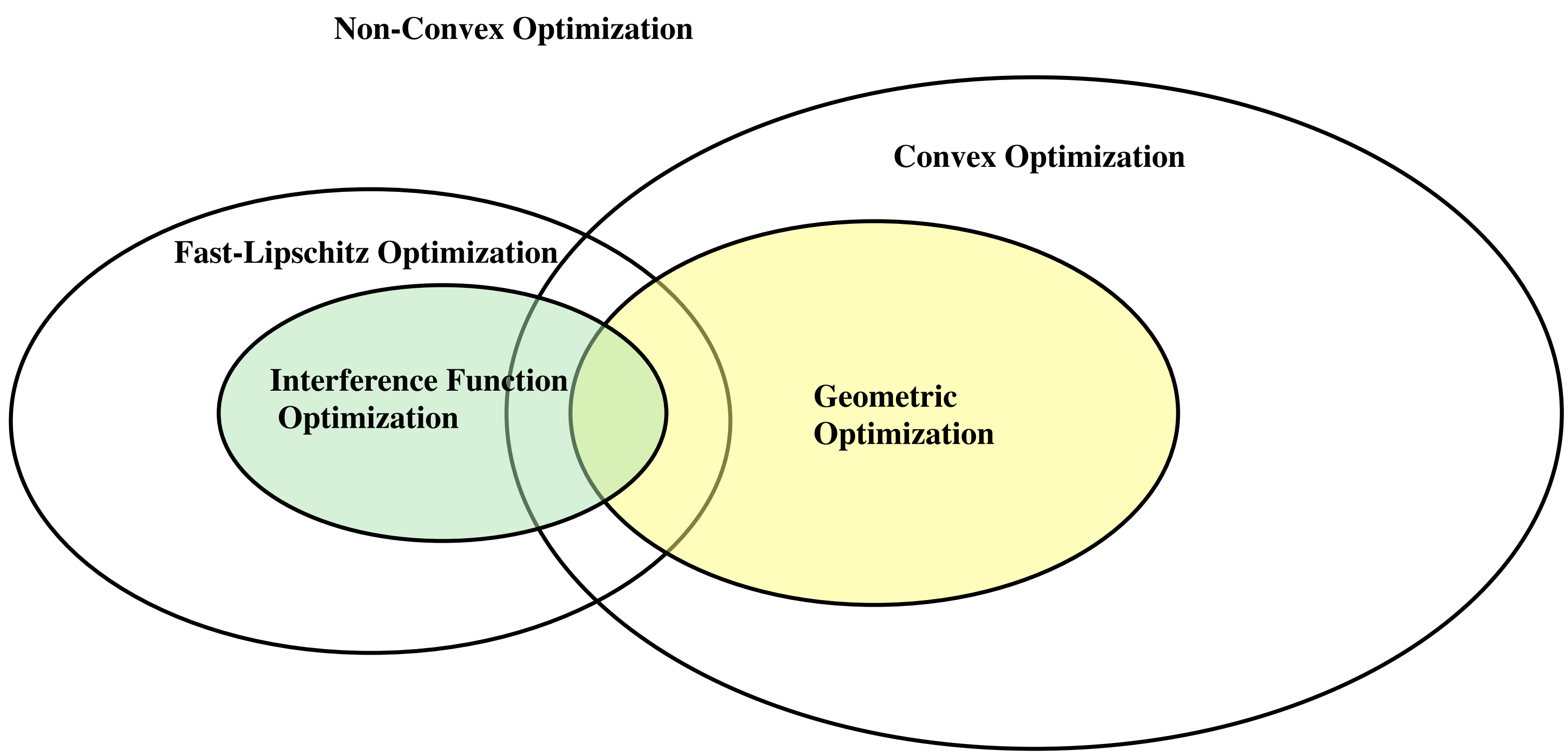
$$L(\boldsymbol{x}, \boldsymbol{\lambda}) = -f_0(\boldsymbol{x}) + \sum_{i=1}^{l} \lambda_i (x_i - f_i(\boldsymbol{x})) + \sum_{i=l+1}^{n} \lambda_i (x_i - h_i(\boldsymbol{x}))$$

Lagrangian

$$
\begin{aligned}
\boldsymbol{x}(k+1) &= \boldsymbol{x}(k) - \beta \nabla_{\boldsymbol{x}} L(\boldsymbol{x}(k), \boldsymbol{\lambda}(k)) \\
\boldsymbol{\lambda}(k+1) &= \boldsymbol{\lambda}(k) - \beta \nabla_{\boldsymbol{\lambda}} L(\boldsymbol{x}(k), \boldsymbol{\lambda}(k))
\end{aligned}
$$

- Lagragian methods need
  1. a central computation of the Lagrangian function
  2. an endless collect-and-broadcast iterative message passing of primal and dual variables
- Fast-Lipschitz methods avoid the central computation and substantially reduce the collect-and-broadcast procedure

The Fast-Lipschitz optimization



Fast-Lipschitz optimization problems can be convex, geometric, quadratic, interference-function,...

Fast-Lipschitz methods

Let us see how a Fast-Lipschitz problem is solved without Lagrangian methods

Centralized optimization

$$\begin{aligned} \max_{\boldsymbol{x}} \quad & f_0(\boldsymbol{x}) \\ \text{s.t.} \quad & x_i \leq f_i(\boldsymbol{x}), \quad i = 1, \dots, l \\ & x_i = h_i(\boldsymbol{x}), \quad i = l+1, \dots, n \\ & \boldsymbol{x} \in \mathscr{D}, \end{aligned}$$

- The optimal solution is given by iterative methods to solve systems of non-linear equations (e.g., Newton methods)

$$\boldsymbol{x}(k+1) = \boldsymbol{x}(k) - \beta\left(\boldsymbol{x}(k) - \mathbf{F}(\boldsymbol{x}(k))\right)$$

$$\mathbf{f}(\boldsymbol{x}) = [f_1(\boldsymbol{x}), f_2(\boldsymbol{x}), \dots, f_l(\boldsymbol{x})]^T$$

$$\mathbf{h}(\boldsymbol{x}) = [h_{l+1}(\boldsymbol{x}), h_{l+2}(\boldsymbol{x}), \dots, h_n(\boldsymbol{x})]^T$$

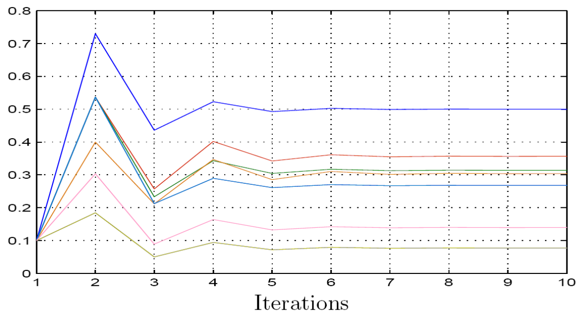
$$\mathbf{F}(\boldsymbol{x}) = [\mathbf{f}(\boldsymbol{x})^T \mathbf{h}(\boldsymbol{x})^T]^T$$

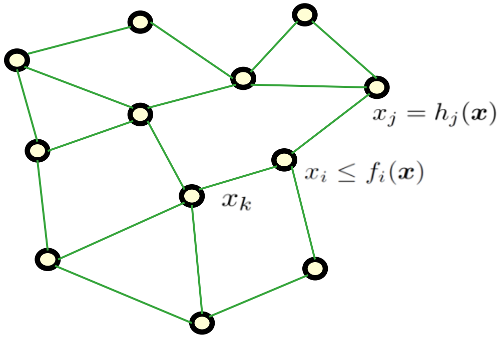
β is a matrix to ensure and maximize convergence speed

- Many other methods are available, e.g., second-order methods

Distributed optimization

$$\begin{aligned} \max_{\boldsymbol{x}} \quad & f_0(\boldsymbol{x}) \\ \text{s.t.} \quad & x_i \leq f_i(\boldsymbol{x}), \quad i = 1, \dots, l \\ & x_i = h_i(\boldsymbol{x}), \quad i = l+1, \dots, n \\ & \boldsymbol{x} \in \mathscr{D}, \end{aligned}$$





Proposition : Let $\boldsymbol{x}(0) \in$ be an initial guess of the optimal solution to a feasible F-Lipschitz problem. Let $\boldsymbol{x}^i(k) = [x_1(\tau_1^i(k)), x_2(\tau_2^i(k)), \dots, x_n^i(\tau_n(k))]$ the vector of decision variables available at node i at time $k \in \mathbb{N}_+$, where $\tau_j^i(k)$ is the delay with which the decision variable of node j is communicated to node i. Then, the following iterative algorithm converges to the optimal solution:

$$x_i(k+1) = [f_i(\boldsymbol{x}^i(k))]^{\mathscr{D}} \quad i = 1, \dots, l$$
$$x_i(k+1) = h_i(\boldsymbol{x}^i(k)) \quad i = l+1, \dots, n$$

where $k \in \mathbb{N}_+$ is an integer associated to the iterations.

Outline

- Motivating example: distributed detection
- Definition of Fast-Lipschitz optimization
- Computation of the optimal solution
- **Problems in canonical form**
- Examples
- Conclusions

Problems in canonical form

Canonical form

Bertsekas, *Non Linear Programming*, 2004

$$\begin{aligned} \min_{\boldsymbol{x}} \quad & g_0(\boldsymbol{x}) \\ \text{s.t.} \quad & g_i(\boldsymbol{x}) \leq 0\,, \quad i = 1, \ldots, l \\ & p_i(\boldsymbol{x}) = 0\,, \quad i = l+1, \ldots, n \\ & \boldsymbol{x} \in \mathscr{D}\,, \end{aligned}$$

Fast-Lipschitz form

$$\begin{aligned} \max_{\boldsymbol{x}} \quad & f_0(\boldsymbol{x}) \\ \text{s.t.} \quad & x_i \leq f_i(\boldsymbol{x})\,, \quad i = 1, \ldots, l\,, \\ & x_i = h_i(\boldsymbol{x}) \quad i = l+1, \ldots, n\,, \\ & \boldsymbol{x} \in \mathscr{D}\,, \end{aligned}$$

$$\begin{aligned} f_0(\boldsymbol{x}) &= -g_0(\boldsymbol{x})\,, & & \\ f_i(\boldsymbol{x}) &= x_i - \gamma_i g_i(\boldsymbol{x})\,, & \gamma_i &> 0 \\ h_i(\boldsymbol{x}) &= x_i - \mu_i p_i(\boldsymbol{x})\,, & \mu_i &\in \mathbb{R} \end{aligned}$$

Problems in canonical form

$$\begin{aligned} \min_{\boldsymbol{x}} \quad & g_0(\boldsymbol{x}) \\ \text{s.t.} \quad & g_i(\boldsymbol{x}) \le 0\,, \quad i = 1,\dots,l \\ & p_i(\boldsymbol{x}) = 0\,, \quad i = l+1,\dots,n \\ & \boldsymbol{x} \in \mathscr{D}\,, \end{aligned}$$

Theorem : Consider the optimization problem in canonical form. Suppose that $\forall\ \boldsymbol{x} \in \mathscr{D}$

$$1.a \quad \nabla g_0(\boldsymbol{x}) \prec 0\,,$$

$$1.b \quad \nabla_i G_i(\boldsymbol{x}) > 0 \quad \forall i\,,$$

and either

$$2.a \quad \nabla_j G_i(\boldsymbol{x}) \le 0 \quad \forall j \ne i\,,$$

$$2.b \quad \nabla_i G_i(\boldsymbol{x}) > \sum_{j \ne i} |\nabla_i G_j(\boldsymbol{x})| \quad \forall i\,,$$

or

$$3.a \quad g_0(\boldsymbol{x}) = -c\mathbf{1}^T\boldsymbol{x} \quad c \in \mathbb{R}^+\,,$$

$$3.b \quad \nabla_j G_i(\boldsymbol{x}) \ge 0 \quad \forall j \ne i\,,$$

$$3.c \quad \nabla_i G_i(\boldsymbol{x}) > \sum_{j \ne i} |\nabla_i G_j(\boldsymbol{x})| \quad \forall i\,,$$

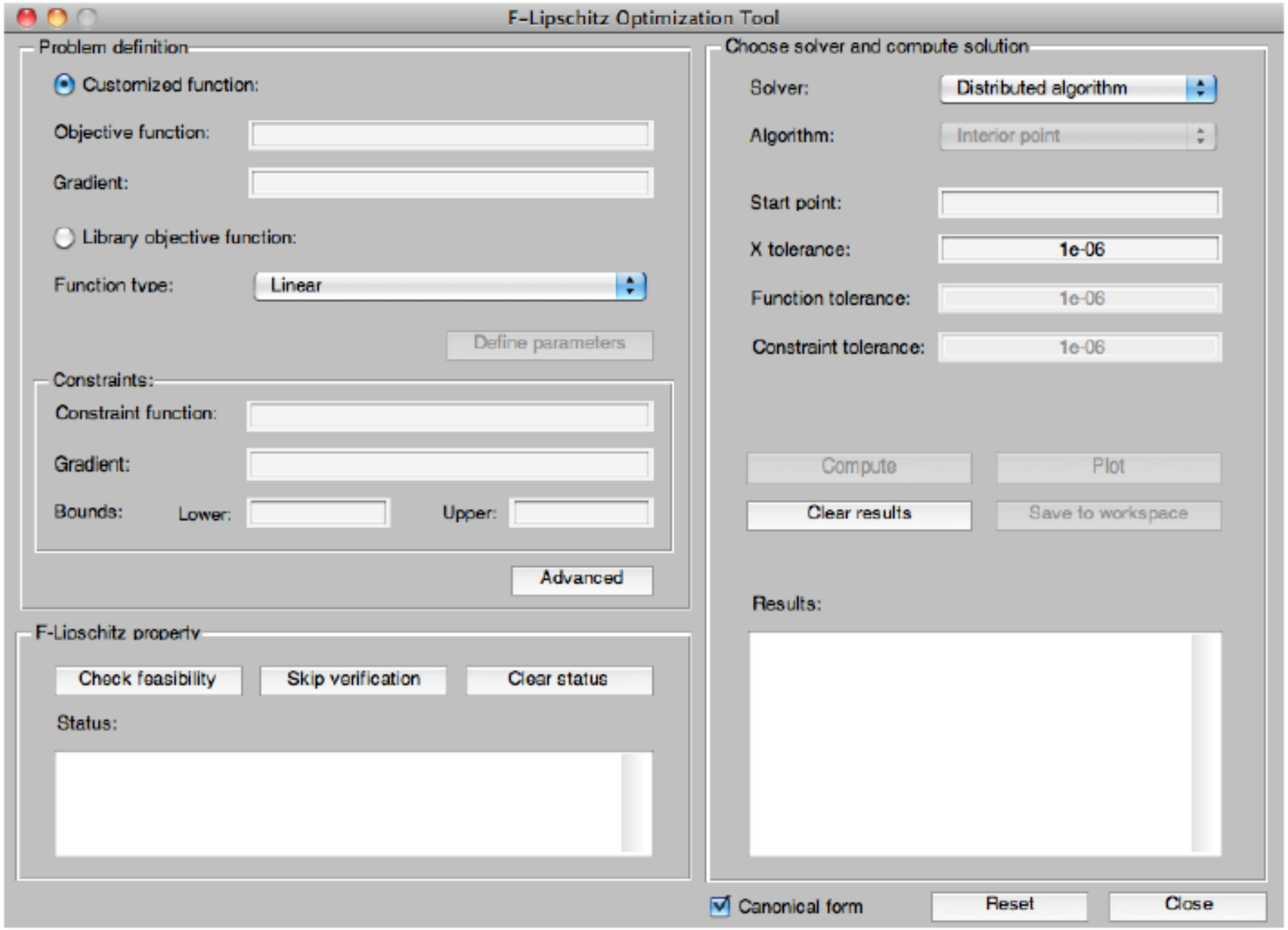
or

$$4.a \quad g_0(\boldsymbol{x}) \in \mathbb{R}\,,$$

$$4.b \quad \frac{\delta}{\delta + \Delta} \nabla_i G_i(\boldsymbol{x}) > \sum_{j \ne i} |\nabla_i G_j(\boldsymbol{x})| \quad \forall i\,.$$

Then, the problem is F-Lipschitz.

Fast-Lipschitz Matlab Toolbox



M. Leithe, Introducing a Matlab Toolbox for Fast-Lipschitz optimization, Master Thesis KTH, 2011

Outline

- Motivating example: distributed detection
- Definition of Fast-Lipschitz optimization
- Computation of the optimal solution
- Problems in canonical form
- **Examples**
- Conclusions

Example 1: from canonical to Fast-Lipschitz

$$\min_{x,y} \quad ae^{-x_1} + be^{-x_2} \qquad\qquad a > 0,\ b > 0$$

$$\text{s.t.} \quad x_1 - 0.5x_2 - 1 \leq 0$$

$$-x_1 + 2x_2 \leq 0$$

$$x_1 \geq 0\,, \quad x_2 \geq 0\,,$$

- The problem is both convex and Fast-Lipschitz:

$$\nabla_x(x - 0.5y - 1) = 1 > |\nabla_y(x - 0.5y - 1)| = 0.5\,,$$

Off-diagonal monotonicity

$$\nabla_y(-x + 2y) = 2 > |\nabla_x(-x + 2y)| = 1\,,$$

Diagonal dominance

- The optimal solution is given by the constraints at the equality, trivially

$$x_1 - 0.5x_2 - 1 = 0 \qquad x_1 = 4/3$$

$$-x_1 + 2x_2 = 0\,, \qquad x_2 = 2/3$$

Example 2: hidden Fast-Lipschitz

- Non Fast-Lipschitz

$$\min_{x,y,z} \quad ae^{-x} + be^{-y} + ce^{z}$$

$$\text{s.t.} \quad 2x - 0.5y + z + 3 \leq 0$$

$$-x + 2y - z^{-1} + 1 \leq 0$$

$$-3x - y + z^{-2} + 2 \leq 0$$

$$x_{\min} \leq x \leq x_{\max}, \quad y_{\min} \leq y \leq y_{\max}, \quad z_{\min} \leq z \leq z_{\max} \,,$$

- Simple variable transformation, $t = z^{-1}$, gives a Fast-Lipschitz form

$$\max_{x,y,t} \quad -ae^{-x} - be^{-y} - ce^{-t}$$

$$\text{s.t.} \quad 2x - 0.5y + t^{-1} + 3 \leq 0$$

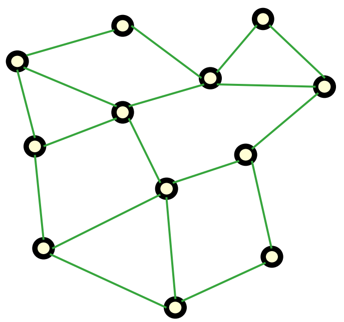
$$-0.5x + 2y - t + 1 \leq 0$$

$$-0.5x - y + t^{2} + 2 \leq 0$$

$$x_{\min} \leq x \leq x_{\max}, \quad y_{\min} \leq y \leq y_{\max}, \quad 1/z_{\max} \leq t \leq 1/z_{\min}$$

Threshold optimization in distributed detection

$$
\begin{aligned}
\min_{\boldsymbol{x}} \quad & \sum_{i=1}^{n} P_{\text{fa}}^{(i)}(x_i) \\
\text{s.t.} \quad & \sum_{j=1}^{n} b_{i,j} P_{\text{md}}^{(j)}(x_j) \le c_i, \quad i = 1, \ldots, n\,, \\
& 0 \preceq \boldsymbol{x} \preceq E\mathbf{1}\,.
\end{aligned}
$$



- How to solve the problem by parallel and distributed operations among the nodes?
- The problem is convex
  - Lagrangian methods (interior point methods) could be applied
  - Drowback: too many message passing (Lagrangian multipliers) among nodes to compute iteratively the optimal solution
- An alternative method: F-Lipschitz optimization

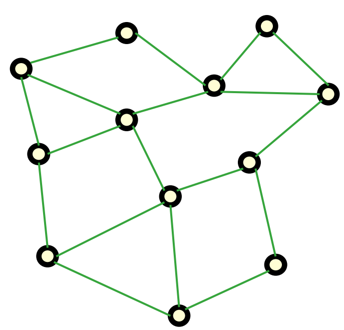
Distributed detection: Fast-Lipschitz vs Lagrangian methods

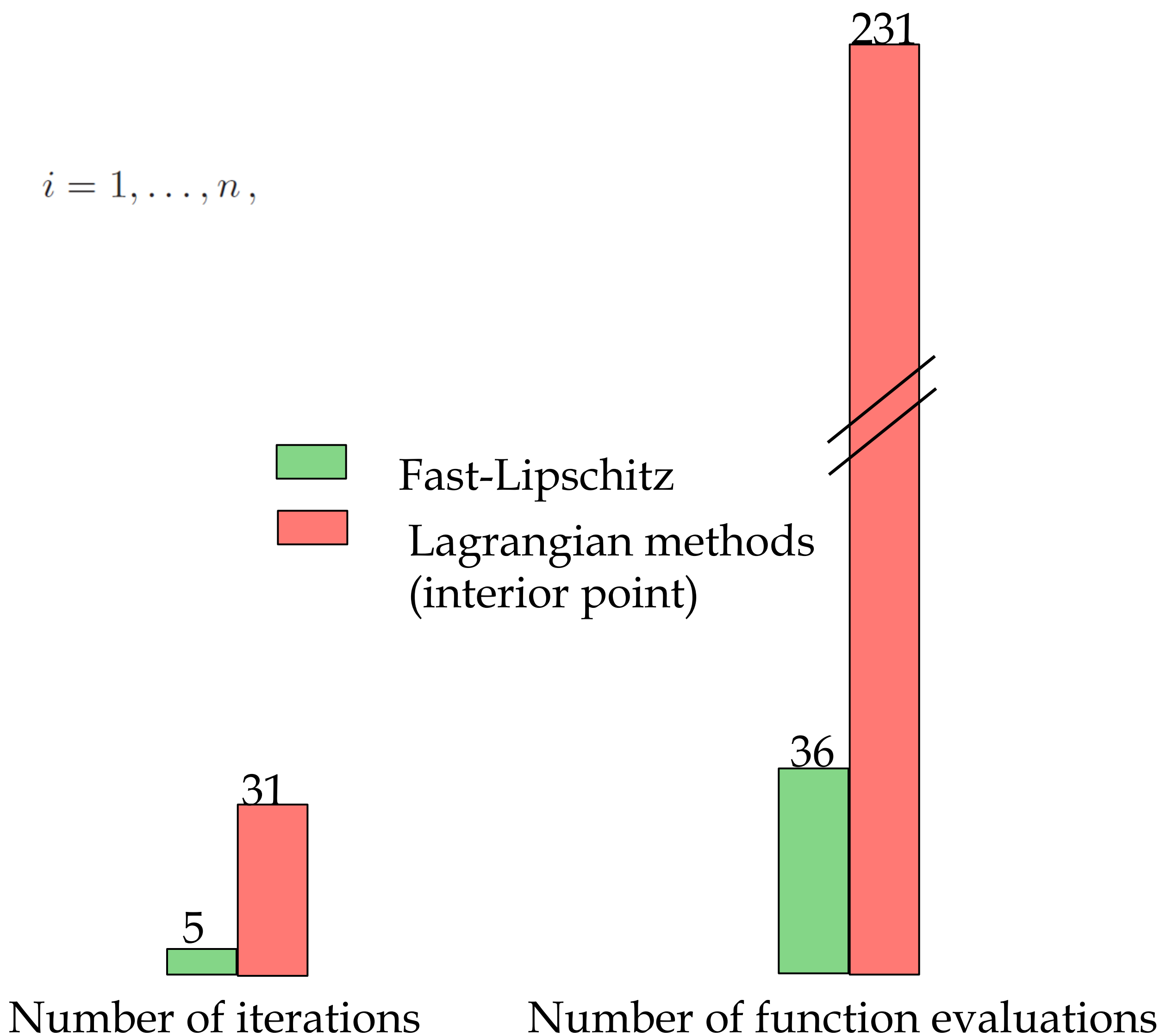
$$\min_{\boldsymbol{x}} \quad \sum_{i=1}^{n} P_{\text{fa}}^{(i)}(x_i)$$

$$\text{s.t.} \quad \sum_{j=1}^{n} b_{i,j} P_{\text{md}}^{(j)}(x_j) \le c_i, \quad i = 1, \ldots, n\,,$$

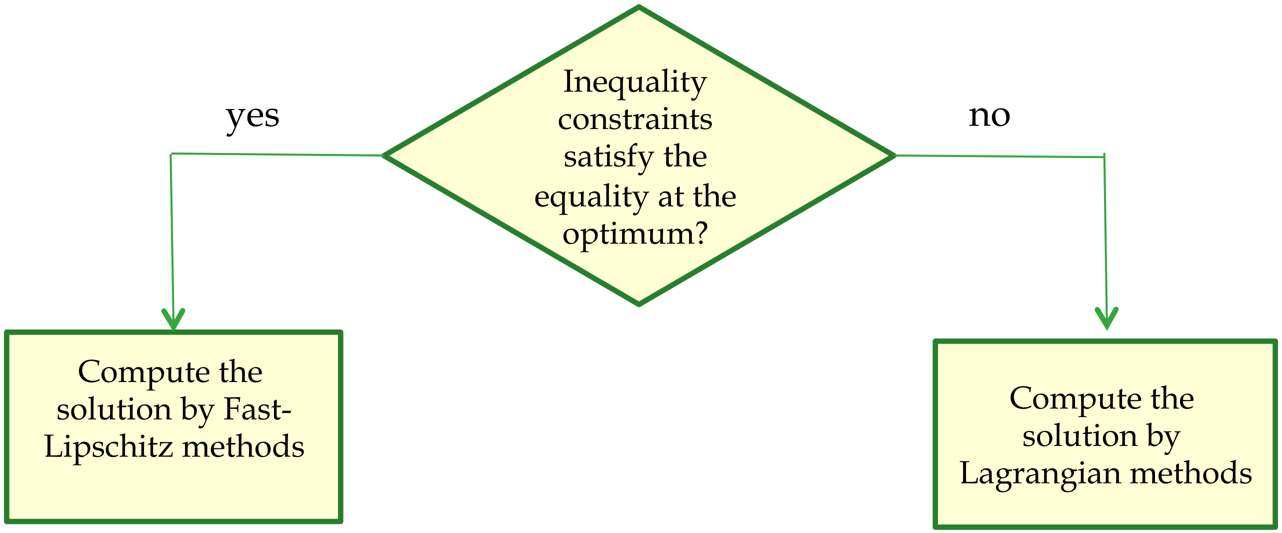
$$0 \preceq \boldsymbol{x} \preceq E\mathbf{1}\,.$$

10 nodes network





Summary



- Fast-Lipschitz optimization: a class of problems for which all the constraints are active at the optimum
- Optimum: the solution to the set of equations given by the constraints
- No Lagrangian methods, which are computationally expensive, particularly on wireless networks

Conclusions

- Existing methods for optimization over networks are too expensive

- Proposed the Fast-Lipschitz optimization
  - Application to distributed detection, many other cases

- Fast-Lipschitz optimization is a panacea for many cases, but still there is a lack of a theory for fast parallel and distributed computations

- How to generalize it for
  - static optimization?
  - dynamic optimization?
  - stochastic optimization?
  - game theoretical extensions?

Selected bibliography

- M. Jacobsson, C. Fischione, “A Comparative Analysis of the Fast-Lipschitz Convergence Speed”, IEEE CDC 2012

- M. Jacobsson, C. Fischione, “On Some Extensions of Fast-Lipschitz Optimization for Convex and Non-convex Problems”, IFAC NecSys 2012

- C. Fischione, “F-Lipschitz Optimization with Wireless Sensor Networks Applications”, *IEEE Transactions on Automatic Control*, 2011.

- A. Speranzon, C. Fischione, K. H. Johansson, A. Sangiovanni-Vincentelli, “A Distributed Minimum Variance Estimator for Sensor Networks”, *IEEE Journal on Selected Areas in Communications, special issue on Control and Communications*, Vol. 26, N. 4, pp. 609—621, May 2008.

Fast-Lipschitz Optimization for Non-convex and Convex Problems

Bellairs
McGill University

February 18-th, 2013

Carlo Fischione

ACCESS Linnaeus Center, Electrical Engineering
KTH Royal Institute of Technology
Stockholm, Sweden
web: http://www.ee.kth.se/~carlofi
e-mail: carlofi@kth.se