Distributed Abstract Optimization via Constraints Consensus

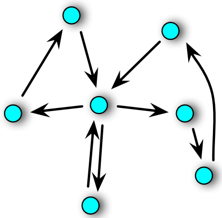
Francesco Bullo

Center for Control,
Dynamical Systems & Computation
University of California at Santa Barbara
http://motion.me.ucsb.edu

Workshop on Network Induced Constraints in Control
Stuttgart, Germany, September 29, 2009

Joint work with Giuseppe Notarstefano, University of Lecce

Preliminary #1: Consensus algorithms



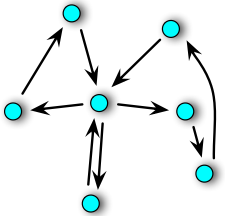
simplest distributed algorithm = linear averaging

each node contains a value x_i and repeatedly executes:

$$x_i^+ := \text{average}(x_i, \{x_j, \text{ for all in-neighbor nodes } j\})$$

each node's value converges to common value
(for strongly connected and aperiodic digraphs)

Preliminary #1: Consensus algorithms



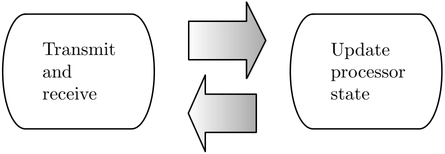
simplest distributed algorithm = linear averaging

each node contains a value x_i and repeatedly executes:

$$x_i^+ := \text{average}(x_i, \{x_j, \text{ for all in-neighbor nodes } j\})$$

each node's value converges to common value
(for strongly connected and aperiodic digraphs)

Preliminary #1: Distributed algorithms on networks



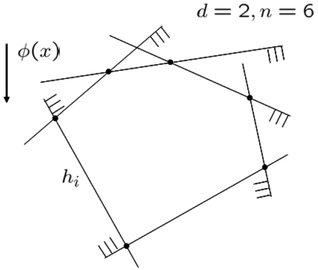
Distributed algorithm for a network of processors consists of

1. W, the **processor state set**
2. $\mathbb{A}$, the **communication alphabet**
3. $\text{stf} : W \times \mathbb{A}^n \to W$, the **state-transition map**
4. $\text{msg} : W \to \mathbb{A}$, the **message-generation map** (often identity map)

Preliminary #2: Optimization problems

Standard LP in d variables with n constraints

$$\begin{aligned} \text{minimize} \quad & c^T x \\ \text{subject to} \quad & a_i^T x \leq b_i \qquad i \in \{1, \ldots, n\} \end{aligned}$$



cost function = direction
linear inequalities = halfspace constraints

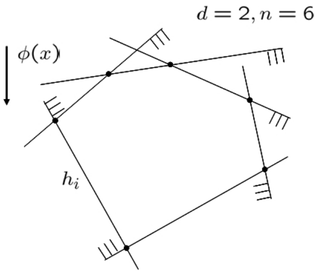
solution uniquely determined by precisely d constraints
(For special cases, use lexicographic minimum solution)

Howto setup a "distributed optimization problem" from this LP?

Preliminary #2: Optimization problems

Standard LP in d variables with n constraints

$$\begin{aligned} \text{minimize} \quad & c^T x \\ \text{subject to} \quad & a_i^T x \le b_i \qquad i \in \{1, \dots, n\} \end{aligned}$$



cost function $=$ direction
linear inequalities $=$ halfspace constraints

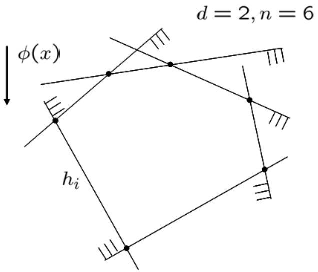
solution uniquely determined by precisely d constraints
(For special cases, use lexicographic minimum solution)

Howto setup a "distributed optimization problem" from this LP?

Preliminary #2: Optimization problems

Standard LP in d variables with n constraints

$$\begin{aligned} \text{minimize} \quad & c^T x \\ \text{subject to} \quad & a_i^T x \leq b_i \qquad i \in \{1, \dots, n\} \end{aligned}$$

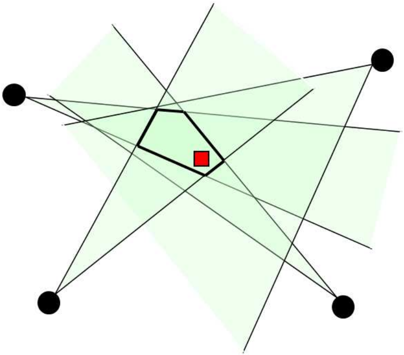


cost function = direction
linear inequalities = halfspace constraints

solution uniquely determined by precisely d constraints
(For special cases, use lexicographic minimum solution)

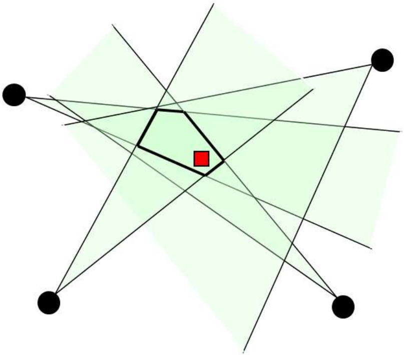
Howto setup a "distributed optimization problem" from this LP?

Preliminary #3: Target localization in sensor networks



1. each sensor/camera i provides "convex set" measurement
2. **set-membership localization** = intersection of n convex sets

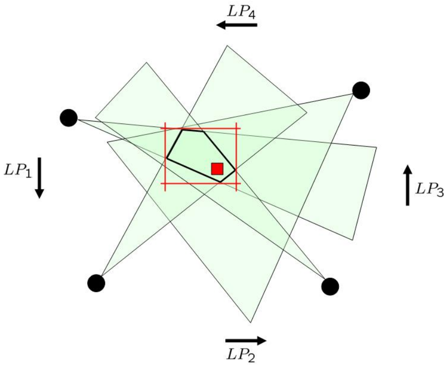
Preliminary #3: Target localization in sensor networks



1. each sensor/camera i provides "convex set" measurement
2. **set-membership localization** = intersection of n convex sets

Preliminary #3: Target localization in sensor networks

1. intersection of n convex sets $\subset$ **axis-aligned bounding box**
2. **axis-aligned bounding box** $:=$ **4 LPs wrt cardinal directions**

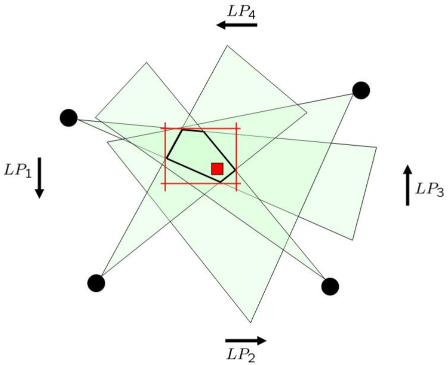


Each LP has 2 variables and

(# constraints) = (# sensors) × (# edges of each measurement)

Preliminary #3: Target localization in sensor networks

1. intersection of n convex sets $\subset$ **axis-aligned bounding box**
2. **axis-aligned bounding box** $:=$ **4 LPs wrt cardinal directions**



Each LP has 2 variables and
(# constraints) = (# sensors) × (# edges of each measurement)

Problem statement: Distributed optimization

A distributed LP

Assume

1. {direction, n halfspace constraints} is feasible LP in d variables
2. G is directed graph with n nodes, strongly connected
3. memory of node i contains {direction, ith halfspace constraint}

Design distributed algorithm so each node computes global LP solution

Dimensionality assumption

$$d \ll n$$

- network with many nodes (order n) and finite memory (order d)
- network with bounded node degree, also
- for $d \sim n$, see "Parallel Computation" by Bertsekas & Tsitsiklis

Problem statement: Distributed optimization

A distributed LP

Assume

1. $\{$direction, n halfspace constraints$\}$ is feasible LP in d variables
2. G is directed graph with n nodes, strongly connected
3. memory of node i contains $\{$direction, ith halfspace constraint$\}$

Design distributed algorithm so each node computes global LP solution

Dimensionality assumption

$$d \ll n$$

- network with many nodes (order n) and finite memory (order d)
- network with bounded node degree, also
- for $d \sim n$, see "Parallel Computation" by Bertsekas & Tsitsiklis

Problem statement: Distributed optimization

A distributed LP

Assume

1. {direction, n halfspace constraints} is feasible LP in d variables
2. G is directed graph with n nodes, strongly connected
3. memory of node i contains {direction, ith halfspace constraint}

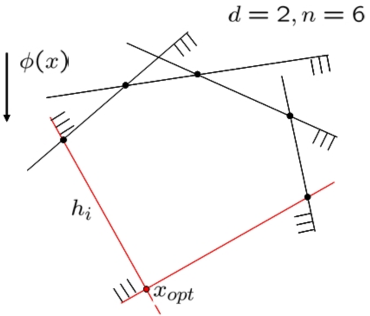
Design distributed algorithm so each node computes global LP solution

Dimensionality assumption

$$d \ll n$$

- network with many nodes (order n) and finite memory (order d)
- network with bounded node degree, also
- for $d \sim n$, see "Parallel Computation" by Bertsekas & Tsitsiklis

Simple observations



- each node knows some local constraints
- each node can solve "local LP" & compute "local active constraints"
- **achieve consensus upon "global active constraints"**

Solution: first attempt

processor state: a set of constraints $\mathcal{C}_i$ — initialized $\mathcal{C}_i := \{(a_i, b_i)\}$

message generation: transmit the set of constraints $\mathcal{C}_i$

state update rule:

1. collect all constraints

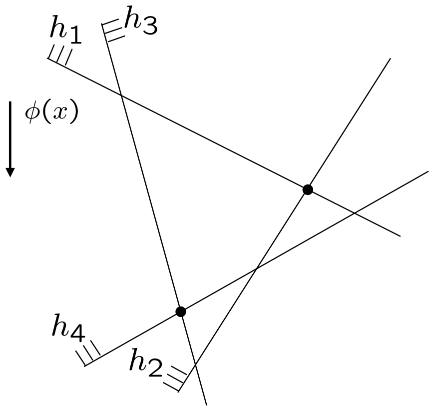
$$\mathcal{C}_{\text{tmp}} := \mathcal{C}_i \cup \Big(\cup_{\text{for all in-neighbor } j} \mathcal{C}_j \Big)$$

2. solve local LP

$$\begin{aligned} &\text{minimize} && c^T x \\ &\text{subject to} && a_k^T x \leq b_k \qquad \text{for all } (a_k, b_k) \in \mathcal{C}_{\text{tmp}} \end{aligned}$$

3. store $\mathcal{C}_i :=$ active constraints in solution of local LP

But constraints need to be re-examined!



Note: h_2 is a global active constraints, but not local:

1. $\{h_1, h_2\}$ is a basis for $\{h_1, h_2, h_3, h_4\}$, but
2. $\{h_3, h_4\}$ is a basis for $\{h_2, h_3, h_4\}$

Solution: Constraints Consensus

processor state: a set of constraints $\mathcal{C}_i$ — initialized $\mathcal{C}_i := \emptyset$

message generation: transmit the set of constraints $\mathcal{C}_i$

state update rule:

1. collect all constraints

$$\mathcal{C}_{\text{tmp}} := \mathcal{C}_i \cup \big(\cup_{\text{for all in-neighbor } j} \mathcal{C}_j \big) \cup \{(a_i, b_i)\}$$

2. solve local LP

$$\begin{aligned} &\text{minimize} \quad c^T x \\ &\text{subject to} \quad a_k^T x \le b_k \qquad \text{for all } (a_k, b_k) \in \mathcal{C}_{\text{tmp}} \end{aligned}$$

3. store

$$\mathcal{C}_i := \begin{cases} \text{active constraints} & \text{if local LP is bounded} \\ \emptyset & \text{if local LP is unbounded} \end{cases}$$

Formal properties of constraints consensus

(Assume one node has bounded solution at initial time)

Monotonicity: the **LP value** at each node is monotonically non-decreasing

Finite time: the **LP value** at each node converges in finite time

Consensus: the **LP values** at all node are equal in finite time

LP solution: after convergence, the **LP constraints set** at each node is an active constraint set for global LP

Uniqueness: if global LP has unique set of active constraints, then the LP constraint set at each node converges that unique set

Time Complexity: unknown, conjectured to be $O(n)$

Formal properties of constraints consensus

(Assume one node has bounded solution at initial time)

Monotonicity: the **LP value** at each node is monotonically non-decreasing

Finite time: the **LP value** at each node converges in finite time

Consensus: the **LP values** at all node are equal in finite time

LP solution: after convergence, the **LP constraints set** at each node is an active constraint set for global LP

Uniqueness: if global LP has unique set of active constraints, then the LP constraint set at each node converges that unique set

Time Complexity: unknown, conjectured to be $O(n)$

Formal properties of constraints consensus

(Assume one node has bounded solution at initial time)

Monotonicity: the **LP value** at each node is monotonically non-decreasing

Finite time: the **LP value** at each node converges in finite time

Consensus: the **LP values** at all node are equal in finite time

LP solution: after convergence, the **LP constraints set** at each node is an active constraint set for global LP

Uniqueness: if global LP has unique set of active constraints, then the LP constraint set at each node converges that unique set

Time Complexity: unknown, conjectured to be $O(n)$

Formal properties of constraints consensus

(Assume one node has bounded solution at initial time)

Monotonicity: the **LP value** at each node is monotonically non-decreasing

Finite time: the **LP value** at each node converges in finite time

Consensus: the **LP values** at all node are equal in finite time

LP solution: after convergence, the **LP constraints set** at each node is an active constraint set for global LP

Uniqueness: if global LP has unique set of active constraints, then the LP constraint set at each node converges that unique set

Time Complexity: unknown, conjectured to be $O(n)$

Formal properties of constraints consensus

(Assume one node has bounded solution at initial time)

Monotonicity: the **LP value** at each node is monotonically non-decreasing

Finite time: the **LP value** at each node converges in finite time

Consensus: the **LP values** at all node are equal in finite time

LP solution: after convergence, the **LP constraints set** at each node is an active constraint set for global LP

Uniqueness: if global LP has unique set of active constraints, then the LP constraint set at each node converges that unique set

Time Complexity: unknown, conjectured to be $O(n)$

Formal properties of constraints consensus

(Assume one node has bounded solution at initial time)

Monotonicity: the **LP value** at each node is monotonically non-decreasing

Finite time: the **LP value** at each node converges in finite time

Consensus: the **LP values** at all node are equal in finite time

LP solution: after convergence, the **LP constraints set** at each node is an active constraint set for global LP

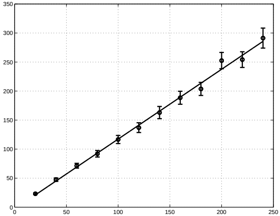
Uniqueness: if global LP has unique set of active constraints, then the LP constraint set at each node converges that unique set

Time Complexity: unknown, conjectured to be $O(n)$

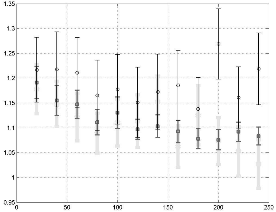
Linear time complexity via Monte Carlo analysis

Nominal problem: $d = 4$, graph $=$ line graph, random LP $=$ hyperplanes with normal vectors uniformly distributed on the unit sphere, and at unit distance from the origin.

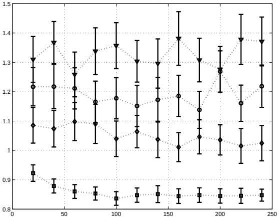
Monte Carlo probability estimation: With 99% confidence, there is 99% probability that a nominal problem with $n \in \{40, 60, 80\}$ is solved via constraints consensus in time bounded by $4(n-1)$.



nominal problems



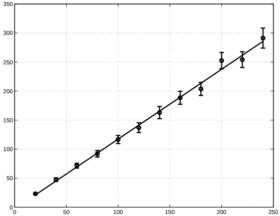
line/Erdős-Rènyi/random-geom graphs



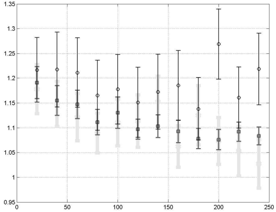
(time/diameter for increasing d)

Nominal problem: $d = 4$, graph $=$ line graph, random LP $=$ hyperplanes with normal vectors uniformly distributed on the unit sphere, and at unit distance from the origin.

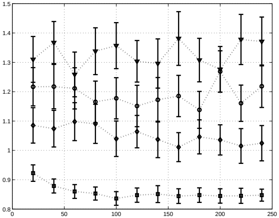
Monte Carlo probability estimation: With 99% confidence, there is 99% probability that a nominal problem with $n \in \{40, 60, 80\}$ is solved via constraints consensus in time bounded by $4(n-1)$.



nominal problems



line/Erdős-Rènyi/random-geom graphs



(time/diameter for increasing d)

End of story ... almost

1. we only considered distributed LPs!
2. what about more general optimization problems?
3. how to generalize constraints consensus?
4. what about formation control?

End of story ... almost

1. we only considered distributed LPs!
2. what about more general optimization problems?
3. how to generalize constraints consensus?
4. what about formation control?

Abstract Optimization

Abstract optimization problem is (H, ω)

- H is a finite set of constraints,
- $\omega(G)$ is the *value function*
  (minimum value attainable by cost function subject to $G \subset H$)

Axioms

Monotonicity: For any F, G, with $F \subset G \subset H$

$$\omega(F) \leq \omega(G)$$

Locality: For any $F \subset G \subset H$ with $\omega(F) = \omega(G)$ and any $h \in H$, then

$$\omega(G) < \omega(G \cup \{h\}) \quad \Longrightarrow \quad \omega(F) < \omega(F \cup \{h\})$$

abstract framework that captures the main features of LP

rich lit: Matousek, Sharir, Welzl, Gärtner, Agarwal, ...

Abstract Optimization

Abstract optimization problem is (H, ω)

- H is a finite set of constraints,
- $\omega(G)$ is the *value function*
  (minimum value attainable by cost function subject to $G \subset H$)

Axioms

Monotonicity: For any F, G, with $F \subset G \subset H$

$$\omega(F) \leq \omega(G)$$

Locality: For any $F \subset G \subset H$ with $\omega(F) = \omega(G)$ and any $h \in H$, then

$$\omega(G) < \omega(G \cup \{h\}) \implies \omega(F) < \omega(F \cup \{h\})$$

abstract framework that captures the main features of LP
rich lit: Matousek, Sharir, Welzl, Gärtner, Agarwal, ...

Ex #1: Distributed training of Support Vector Machines

Max Margin Problem

Separable training set = a separable set $\{(x_i, \ell_i)\}$ of n examples $x_i \in \mathbb{R}^k$ and labels $\ell_i \in \{-1, +1\}$. Find $(t_+, t_-) \in \mathbb{R}^2$ and $w \in \mathbb{R}^k$

$$\begin{aligned} \text{minimize} \quad & \frac{1}{2}\|w\|^2 - (t_+ - t_-) \\ \text{subject to} \quad & w \cdot x_i \geq t_+ \quad \text{if } \ell_i = +1 \\ & w \cdot x_i \leq t_- \quad \text{if } \ell_i = -1 \end{aligned}$$

Balcázar et al, TCS '08: Max Margin satisfies axioms

Distributed Max Margin Problem

1. a separable training set $\{(x_i, \ell_i)\}$
2. G is directed graph with n nodes, strongly connected
3. memory of node i contains the example-label pair (x_i, ℓ_i)

Constraints Consensus solves the Distributed Max Margin

Ex #1: Distributed training of Support Vector Machines

Max Margin Problem

Separable training set = a separable set $\{(x_i, \ell_i)\}$ of n examples $x_i \in \mathbb{R}^k$ and labels $\ell_i \in \{-1, +1\}$. Find $(t_+, t_-) \in \mathbb{R}^2$ and $w \in \mathbb{R}^k$

$$\begin{aligned} \text{minimize} \quad & \frac{1}{2}\|w\|^2 - (t_+ - t_-) \\ \text{subject to} \quad & w \cdot x_i \geq t_+ \quad \text{if } \ell_i = +1 \\ & w \cdot x_i \leq t_- \quad \text{if } \ell_i = -1 \end{aligned}$$

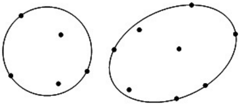
Balcázar et al, TCS '08: Max Margin satisfies axioms

Distributed Max Margin Problem

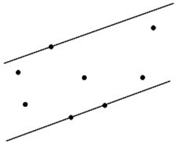
1. a separable training set $\{(x_i, \ell_i)\}$
2. G is directed graph with n nodes, strongly connected
3. memory of node i contains the example-label pair (x_i, ℓ_i)

Constraints Consensus solves the Distributed Max Margin

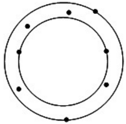
- Smallest enclosing ball, ellipsoid and axis-aligned bounding box



- Smallest enclosing stripe (generic points)



- Smallest enclosing annulus

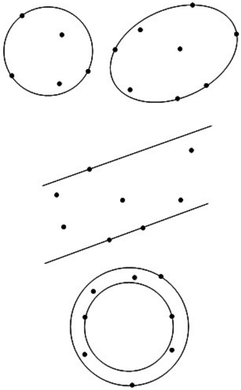


Application to motion coordination in robotic networks

1. computing optimal shapes in distributed fashion
2. from distributed shape consensus, easy to design formation control

Ex #2: Distributed geometric optim & formation control

- Smallest enclosing ball, ellipsoid and axis-aligned bounding box
- Smallest enclosing stripe (generic points)
- Smallest enclosing annulus



Application to motion coordination in robotic networks

1. computing optimal shapes in distributed fashion
2. from distributed shape consensus, easy to design formation control

End of story

1. distributed abstract optimization
2. consensus constraints: correctness and time complexity
3. applications to target tracking & formation control

References:

- B. Gärtner. A subexponential algorithm for abstract optimization problems. *SIAM J Computing*, 24(5):1018–1035, 1995
- P. K. Agarwal and M. Sharir. Efficient algorithms for geometric optimization. *ACM Computing Surveys*, 30(4):412–458, 1998
- G. Notarstefano and F. Bullo. Network abstract linear programming with application to minimum-time formation control. In *Proc CDC*, pages 927–932, New Orleans, LA, December 2007
- G. Notarstefano and F. Bullo. Distributed abstract optimization via constraints consensus: theory and applications, October 2009. Available at `http://arxiv.org/abs/0910.5816`