Distributed Optimization over a Network

Angelia Nedić

angelia@illinois.edu

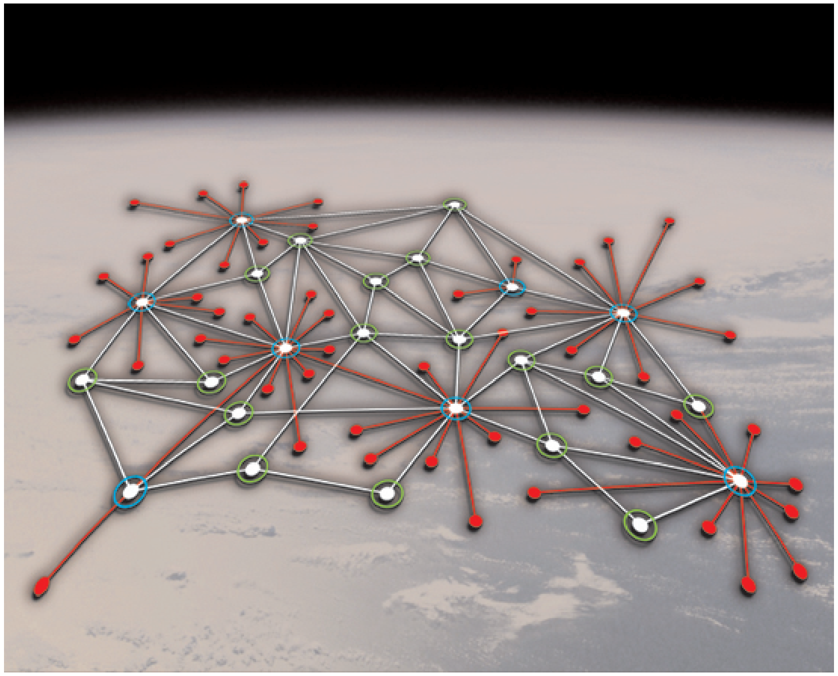
Industrial and Enterprise Systems Engineering Department
and Coordinated Science Laboratory
University of Illinois at Urbana-Champaign

Collaborators A. Ozdaglar, P.A. Parrilo, D. Stipanović, V.V. Veeravali

Ph.D. students S.S. Ram, K. Srivastava, B. Touri

Large Networked Systems

- The recent advances in wired and wireless technology lead to the emergence of large-scale networks
  - Internet
  - Mobile ad-hoc networks
  - Wireless sensor networks
- The advances gave rise to new network applications including
  - Decentralized network operations including resource allocation, coordination, learning, estimation
  - Data-base networks
  - Social and economic networks



- As a result, there is a necessity to develop new models and tools for the design and performance analysis of such large complex dynamics systems

New Application Challenges

- **Lack of central "authority"**
  - Centralized network architecture is **not possible**
    - Size of the network / Proprietary issues
  - Sometimes centralized architecture is **not desirable**
    - Security issues / Robustness to failures

- **Network dynamics**
  - Mobility of the network
    - The agent spatio-temporal dynamics
    - Network connectivity structure is varying in time
  - Time-varying network
    - The network itself is evolving in time

- The challenge is to design algorithms to support efficient operations over such networks.

Trends: Areas of Research

- Network aspects
  - **Consensus, information diffusion, opinion dynamics**
    - □ Stability, characterization of equilibria, bounds on convergence rate/time

- Optimization issues
  - **Distributed algorithms for optimization over networks**
    - □ Different network performance measures
    - □ Synchronous/asynchronous implementations
    - □ Different information diffusion/exchange protocols

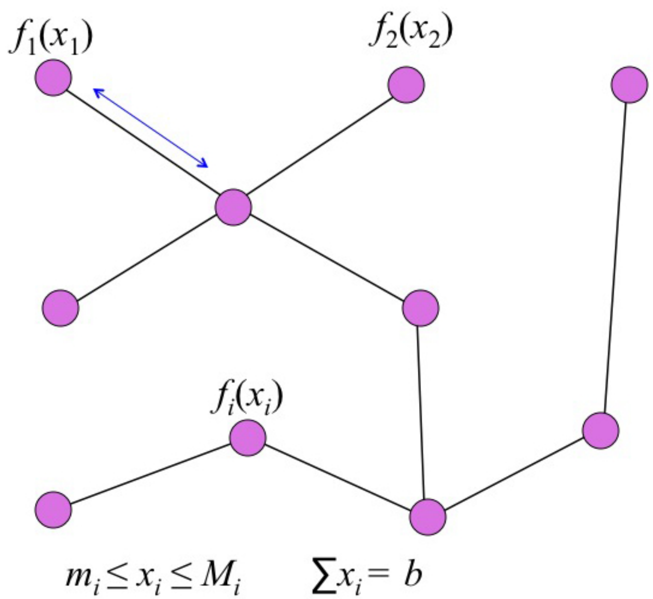
- Uncertainty issues in distributed systems
  - Communication noise and delay (network dependent)
  - Uncertain measurements (application dependent)
  - Computational errors (algorithm dependent)

Distributed Computational Model: Self-organized Agents

The model consists of **a network of computing agents (nodes) that cooperate in order to optimize a network-wide objective function**

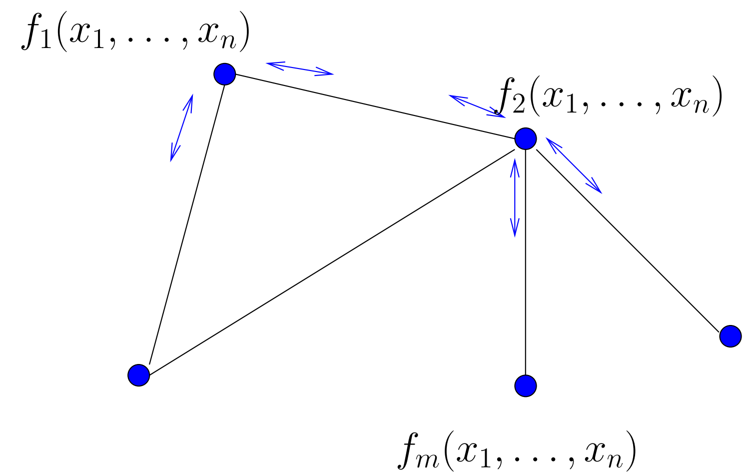
- Agents can communicate only with immediate neighbors in the network
- Agents have individual objective functions that they do not "reveal" to each other
- Agents cooperate (share info.) with their neighbors

Distributed load-balancing in a network



Network objective:
$$\text{minimize}_{\mathbf{1}'x=b} \sum_i f_i(x_i)$$

Uplink power control in mobile cellular net

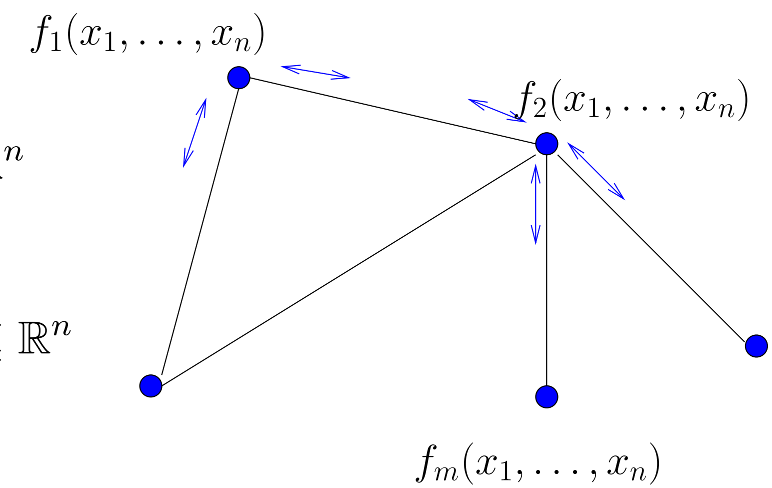


Network objective:
$$\text{minimize}_{x \in X} \max_{1 \leq i \leq m} f_i(x)$$

General Model

- Network of m agents represented by an undirected graph $([m], \mathscr{E})$ where $[m] = \{1, \ldots, m\}$ and $\mathscr{E}$ is the edge set
- Each agent i has an objective function $f_i(x)$ known to that agent only
- Common constraint (closed convex) set X known to all agents

Distributed Self-organized Agent System



The problem can be:

minimize $\sum_{i=1}^{m} f_i(x)$ subject to $x \in X \subseteq \mathbb{R}^n$

or

minimize $\max_{1 \le i \le m} f_i(x)$ subject to $x \in X \subseteq \mathbb{R}^n$

How Agents Manage to Optimize

$$\text{minimize} \sum_{i=1}^{m} f_i(x) \quad \text{subject to } x \in X \subseteq \mathbb{R}^n$$

$$\text{minimize} \max_{1 \le i \le m} f_i(x) \quad \text{subject to } x \in X \subseteq \mathbb{R}^n$$

- Each agent i will generate its own estimate $x^i(t)$ of an optimal solution to the problem

- Each agent will update its estimate $x^i(t)$ by performing two steps:
  - Consensus-like step (mechanism to align agents estimates toward a common point)
  - Local gradient-based step (to minimize its own objective function)

C. Lopes and A. H. Sayed, "Distributed processing over adaptive networks," Proc. Adaptive Sensor Array Processing Workshop, MIT Lincoln Laboratory, MA, June 2006.

A. H. Sayed and C. G. Lopes, "Adaptive processing over distributed networks," IEICE Transactions on Fundamentals of Electronics, Communications and Computer Sciences, vol. E90-A, no. 8, pp. 1504-1510, 2007.

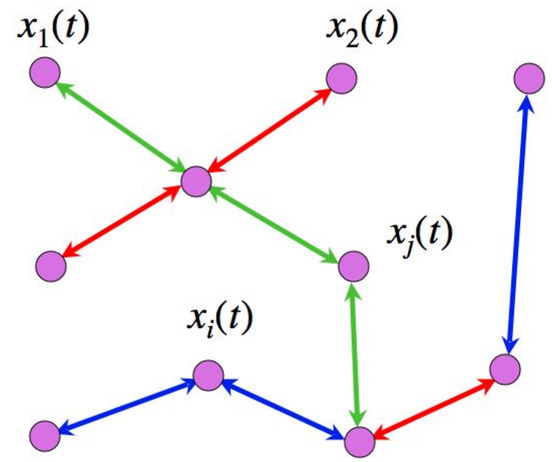
A. Nedić and A. Ozdaglar "On the Rate of Convergence of Distributed Asynchronous Subgradient Methods for Multi-agent Optimization" Proceedings of the 46th IEEE Conference on Decision and Control, New Orleans, USA, 2007, pp. 4711-4716.

A. Nedić and A. Ozdaglar, Distributed Subgradient Methods for Multi-agent Optimization IEEE Transactions on Automatic Control 54 (1) 48-61, 2009.

Consensus Problem

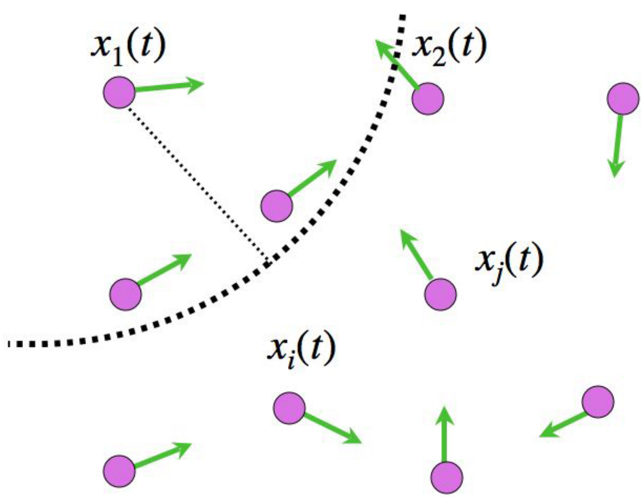
Consider a connected network of m-agent, each knowing its own scalar value $x_i(0)$ at time $t = 0$.
The problem is to design a distributed and local algorithm ensuring that **the agents agree on the same value** x, i.e.,

$$\lim_{t\to\infty} x_i(t) = x \qquad \text{for all } i.$$



Leaderless Heading Alignment

A system of autonomous agents are moving in the plane with the same speed but with different headings [Vicsek 95, Jadbabaie *et al.* 03]
The objective is to design a local protocol that will ensure the alignment of agent headings



Consensus Algorithm

Each agent combines its estimate $x_i(t)$ with the estimates $x_j(t)$ received from its neighbors

$$x_i(t+1) = \sum_{j \in N_i} a_{ij}\, x_j(t) \qquad \text{for all } i.$$

where N_i is the set of neighbors of agent i (including itself)

$$N_i = \{j \in [m] \mid (i,j) \in \mathscr{E}\}$$

and $a_{ij} \geq 0$ is a weight that agent i assigns to the information coming from its neighbor $j \in N_i$.

The weights $\{a_{ij}, j \in N_i\}$ sum to 1, i.e., $\sum_{j \in N_i} a_{ij} = 1$ for all agents i

Introducing the values $a_{ij} = 0$ when $j \notin N_i$, the consensus algorithm can be written as:

$$x_i(t+1) = \sum_{j=1}^{m} a_{ij}\, x_j(t)$$

where

$$a_{ij} \geq 0 \qquad \text{with } a_{ij} = 0 \text{ when } j \notin N_i$$

$$\sum_{j=1}^{m} a_{ij} = 1$$

Distributed Optimization Algorithm

$$\text{minimize } \sum_{i=1}^{m} f_i(x) \quad \text{subject to } x \in X \subseteq \mathbb{R}^n$$

- At time t, each agent i has its own estimate $x^i(t)$ of an optimal solution to the problem

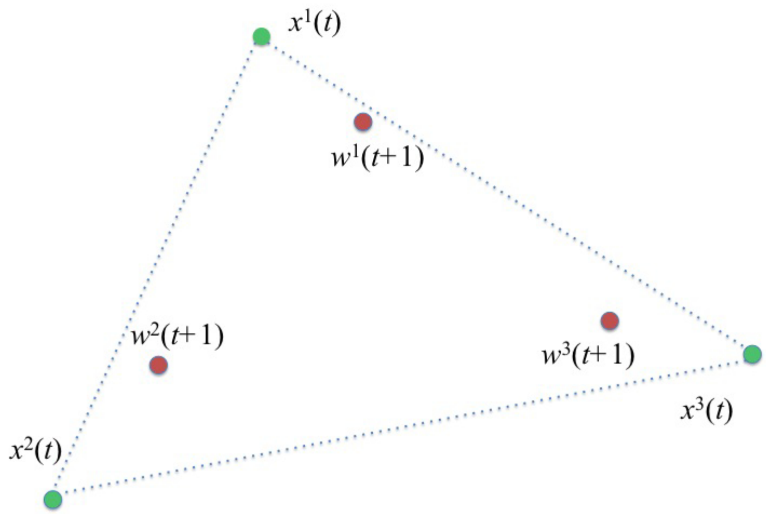
- At time $t+1$, agents communicate their estimates to their neighbors and update by performing two steps:
  - Consensus-like step to mix their own estimate with those received from neighbors

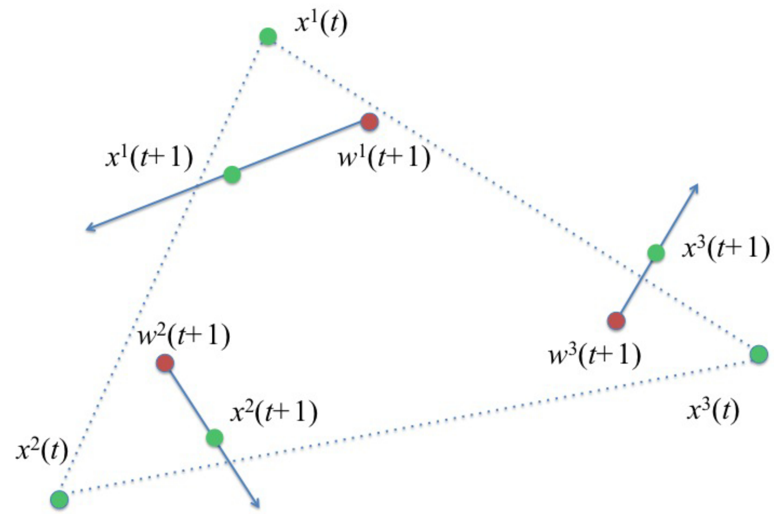
$$w^i(t+1) = \sum_{j=1}^{m} a_{ij} x^j(t) \qquad\qquad (a_{ij} = 0 \text{ when } j \notin N_i)$$

  - Followed by a local gradient-based step

$$x^i(t+1) = \Pi_X[w^i(t+1) - \alpha(t)\nabla f_i(w^i(t+1))]$$

  where $\Pi_X[y]$ is the Euclidean projection of y on X, f_i is the local objective of agent i and $\alpha(t) > 0$ is a stepsize





Intuition Behind the Algorithm: It can be viewed as a consensus steered by a "force":

$$
\begin{aligned}
x^i(t+1) &= w^i(t+1) + (\Pi_X[w^i(t+1) - \alpha(t)\nabla f_i(w^i(t+1))] - {\color{blue}w^i(t+1)}) \\
&= w^i(t+1) + \underbrace{(\Pi_X[w^i(t+1) - \alpha(t)\nabla f_i(w^i(t+1))] - {\color{blue}\Pi_X[w^i(t+1)]})}_{\text{small stepsize } \alpha(t)} \\
&\approx w^i(t+1) - \alpha(t)\nabla f_i(w^i(t+1)) \\
&= {\color{red}\sum_{j=1}^m a_{ij}x^j(t)} - \alpha(t)\nabla f_i\left({\color{red}\sum_{j=1}^m a_{ij}x^j(t)}\right)
\end{aligned}
$$

Matrices A that lead to consensus, also yield convergence of an optimization algorithm

Convergence Result for Static Network

Convex Problem: Let X be closed and convex, and each $f_i : \mathbb{R}^n \to \mathbb{R}$ be convex with bounded (sub)gradients over X. Assume the problem $\min_{x \in X} \sum_{i=1}^m f_i(x)$ has a solution.

Stepsize Rule: Let the stepsize $\alpha(t)$ be such that $\sum_{t=0}^\infty \alpha(t) = \infty$ and $\sum_{t=0}^\infty \alpha^2(t) < \infty$.

Network: Let the graph $([m], \mathscr{E})$ be directed and strongly connected. Let the matrix $A = [a_{ij}]$ of agents' weights be doubly stochastic. Then,

$$\lim_{t\to\infty} x^i(t) = x^* \quad \text{for all } i,$$

where x^* is a solution of the problem.

<u>*Proof Outline*:</u>

Use $\sum_{i=1}^m \|x^i(t) - x^*\|^2$ as a Lyapunov function, where x^* is a solution to the problem

Due to convexity and (sub)gradient boundedness, we have

$$\sum_{i=1}^m \|x^i(t+1) - x^*\|^2 \le \sum_{i=1}^m \|w^i(t+1) - x^*\|^2 - 2\alpha(t) \sum_{i=1}^m \big(f_i(w^i(t+1)) - f_i(x^*)\big) + \alpha^2(t) C^2$$

By $w^i(t+1) = \sum_{j=1}^m a_{ij}\, x^j(t)$ and the doubly stochasticity of A, we have

$$\sum_{i=1}^m \|x^i(t+1)-x^*\|^2 \leq \sum_{j=1}^m \|x^j(t)-x^*\|^2 - 2\alpha(t)\sum_{i=1}^m \left(f_i(w^i(t+1)) - f_i(x^*)\right) + \alpha^2(t)C^2$$

Thus, letting $s(t+1) = \frac{1}{m}\sum_{i=1}^m x^i(t+1)$ we see

$$\sum_{i=1}^m \|x^i(t+1)-x^*\|^2 \leq \sum_{j=1}^m \|x^j(t)-x^*\|^2 - 2\alpha(t)\sum_{i=1}^m (f_i(s(t+1)) - f_i(x^*))$$

$$+\, 2\alpha(t)\sum_{i=1}^m \left(f_i(s(t+1)) - f_i(w^i(t+1))\right) + \alpha^2(t)C^2$$

Letting $F(x) = \sum_{i=1}^m f_i(x)$ and using (sub)gradient boundedness, we find

$$\underbrace{\sum_{i=1}^m \|x^i(t+1)-x^*\|^2}_{V(t+1)} \leq \underbrace{\sum_{j=1}^m \|x^j(t)-x^*\|^2}_{V(t)} - 2\alpha(t)\underbrace{(F(s(t+1)) - F(x^*))}_{\geq 0}$$

$$+\, 2\alpha(t)C\sum_{i=1}^m \|s(t+1) - w^i(t+1)\| + \alpha^2(t)C^2$$

We can see $\sum_{t=0}^\infty \alpha(t)C\sum_{i=1}^m \|s(t+1) - w^i(t+1)\| < \infty$

The result would hold if we can show $\|s(t+1) - w^i(t+1)\| \to 0$ as $t \to \infty$ for all i

The trouble is in showing $\|s(t+1) - w^i(t+1)\| \to 0$ as $t \to \infty$ for all i, which is exactly where the **network impact is** – rate of convergence of A^t to its limit is needed

When the network is connected, the matrices A^t converge to the matrix $\frac{1}{m}\mathbf{1}\mathbf{1}'$, as $t \to \infty$

The convergence rate is

$$\left|[A^t]_{ij} - \frac{1}{m}\right| \le q^t, \qquad \text{where } q \in (0,1)$$

We have for arbitrary $0 \le \tau < t$

$$x^i(t+1) = w^i(t+1) + (\underbrace{\Pi_X[w^i(t+1) - \alpha(t)\nabla f_i(w^i(t+1))] - w^i(t+1)}_{e^i(t)})$$

$$= \sum_{j=1}^m a_{ij}\, x^j(t) + e^i(t) = \cdots$$

$$= \sum_{j=1}^m [A^{t+1-\tau}]_{ij}\, x^j(\tau) + \sum_{k=\tau+1}^{t} \sum_{j=1}^m [A^k]_{ij} e_j(t-k) + e^i(t)$$

Similarly, for $s(t+1) = \frac{1}{m}\sum_{i=1}^m x^i(t+1)$ we have

$$s(t+1) = s(t) + \frac{1}{m}\sum_{j=1}^m e^j(t) = \cdots = \sum_{j=1}^m \frac{1}{m} x^j(\tau) + \sum_{k=\tau+1}^{t}\sum_{j=1}^m \frac{1}{m} e_j(t-k) + \sum_{j=1}^m \frac{1}{m} e^j(t)$$

Thus,

$$\|x^i(t+1)-s(t+1)\| \leq q^{t+1-\tau}\sum_{j=1}^{m}\|x^j(\tau)\|+\sum_{k=\tau+1}^{t}\sum_{j=1}^{m}q^k\|e_j(t-k)\|+\sum_{j=1}^{m}\frac{1}{m}\|e^j(t)\|+\|e^i(t)\|$$

Since

$$e^i(t) = \Pi_X[w^i(t+1) - \alpha(t)\nabla f_i(w^i(t+1))] - w^i(t+1)$$

we have

$$\|e^i(t)\| \leq \alpha(t)C$$

Hence

$$\|x^i(t+1) - s(t+1)\| \leq q^{t+1-\tau}\sum_{j=1}^{m}\|x^j(\tau)\| + mC\sum_{k=\tau+1}^{t}q^k\alpha(t-k) + 2C\alpha(t)$$

By choosing τ such that $\|e(t)\| \leq \epsilon$ for all $t \geq \tau$ and then, using some properties of the sequences involved in the above relation, we show

$$\|x^i(t+1) - s(t+1)\| \to 0$$

which in view of $x^i(t+1) = w^i(t+1) + e^i(t)$ and $e^i(t) \to 0$ implies

$$\|w^i(t+1) - s(t+1)\| \to 0$$

Extension to Time-varying Networks

- Consensus-like step to mix their own estimate with those received from neighbors

$$w^i(t+1) = \sum_{j=1}^{m} a_{ij}(t)x^j(t) \qquad (a_{ij}(t) = 0 \text{ when } j \notin N_i(t))$$

- Followed by a local gradient-projection step

$$x^i(t+1) = \Pi_X[w^i(t+1) - \alpha(t)\nabla f_i(w^i(t+1))]$$

For convergence, some conditions on the weight matrices $A(t) = [a_{ij}(t)]$ are needed.

Convergence Result for Time-varying Network:*

Let the problem be convex, f_i have bounded (sub)gradients on X, and $\sum_{t=0}^{\infty} \alpha(t) = \infty$ and $\sum_{t=0}^{\infty} \alpha^2(t) < \infty$. Let the graphs $G(t) = ([m], \mathscr{E}(t))$ be directed and strongly connected, and the matrices $A(t)$ be such that $a_{ij}(t) = 0$ if $j \notin N_i(t)$, while $a_{ij}(t) \geq \gamma$ whenever $a_{ij}(t) > 0$, where $\gamma > 0$. Also assume that $A(t)$ are doubly stochastic†. Then,

$$\lim_{t\to\infty} x^i(t) = x^* \quad \text{for all } i,$$

where x^* is a solution of the problem.

---

*AN and A. Ozdaglar "Distributed Subgradient Methods for Multi-agent Optimization" *IEEE Transactions on Automatic Control* 54 (1) 48-61, 2009.

†J. N. Tsitsiklis, "Problems in Decentralized Decision Making and Computation," Ph.D. Thesis, Department of EECS, MIT, November 1984; technical report LIDS-TH-1424, Laboratory for Information and Decision Systems, MIT

Other Extensions

$$w^i(t+1) = \sum_{j=1}^{m} a_{ij}(t)x^j(t) \qquad (a_{ij}(t) = 0 \text{ when } j \notin N_i(t))$$

$$x^i(t+1) = \Pi_X[w^i(t+1) - \alpha(t)\nabla f_i(w^i(t+1))]$$

Extensions include

- Gradient directions $\nabla f_i(w^i(t+1))$ can be erroneous
  Ram, Nedić, Veeravali 2009, 2010, Srivastava and Nedić 2011

$$x^i(t+1) = \Pi_X[w^i(t+1) - \alpha(t)(\nabla f_i(w^i(t+1)) + \varphi_i(t+1))]$$

- The set X can be $X = \cap_{i=1}^m X_i$ where each X_i is a private information of agent i
  Nedić, Ozdaglar, and Parrilo 2010, Srivastava‡ and Nedić 2011

$$x^i(t+1) = \Pi_{X_i}[w^i(t+1) - \alpha(t)\nabla f_i(w^i(t+1))]$$

- The links can be noisy i.e., $x^j(t)$ is sent to agent i, but the agent receives $x^j(t) + \epsilon^{ij}(t)$
  Srivastava and Nedić 2011
- The updates can be asynchronous - edge-set $\mathscr{E}(t)$ is random
  Ram, Nedić, and Veeravalli, Nedić 2011
- Different sum-based functional structures [Ram, Nedić, and Veeravalli 2012]

‡Uses different weights

Application to Data Classification

Given a set of data points $\{(z_j, y_j), j = 1, \ldots, p\}$, find a vector (x, u) that

$$\text{minimizes} \quad \frac{\lambda}{2}\|x\|^2 + \sum_{j=1}^{p} \max\{0, 1 - y_j(x'z_j + u)\}$$

Suppose that the data is distributed at m locations, with each location having data points $\{(z_\ell, y_\ell), \ell \in S_i\}$, with S_i being the index set

The problem can be written as:

$$\text{minimize} \sum_{i=1}^{m} \underbrace{\left(\frac{\lambda}{2m}\|x\|^2 + \sum_{\ell \in S_i} \max\{0, 1 - y_\ell(x'z_\ell + u)\}\right)}_{f_i(\mathbf{x})} \quad \text{over } \mathbf{x} = (x, u) \in \mathbb{R}^n \times \mathbb{R}$$

The algorithm has the form:

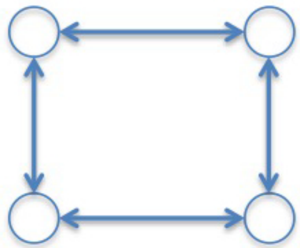
$$w^i(t+1) = \mathbf{x}^i(t) - \eta(t) \sum_{j=1}^{m} r_{ij}\mathbf{x}^j(t) \qquad (r_{ij} = 0 \text{ when } j \notin N_i)$$

$$\mathbf{x}^i(t+1) = w^i(t+1) - \alpha(t) \underbrace{g_i(w^i(t+1))}_{\textbf{subgradient}}$$

Algorithm is discussed in K. Srivastava and AN "Distributed Asynchronous Constrained Stochastic Optimization" *IEEE Journal of Selected Topics in Signal Processing* 5 (4) 772-790, 2011.

Case with perfect communications

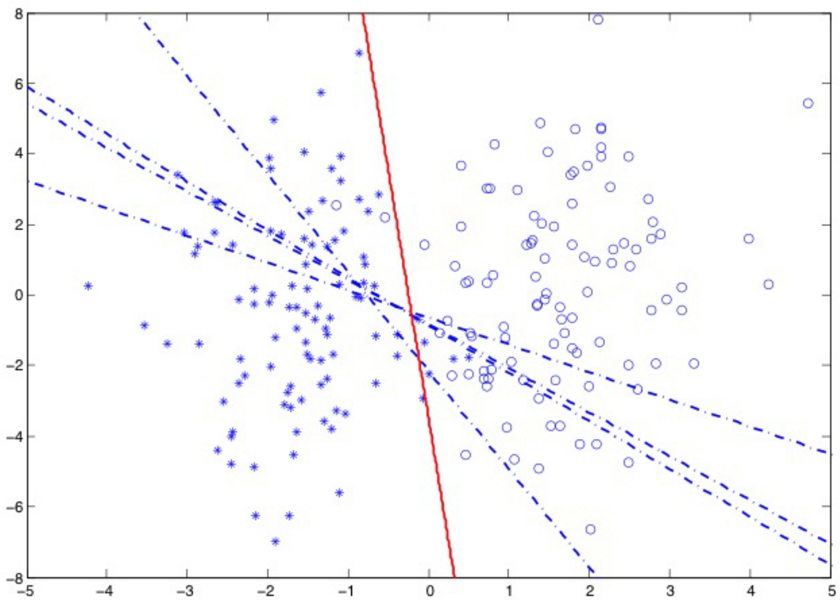
Illustration uses a simple graph of 4 nodes organized in a ring-network



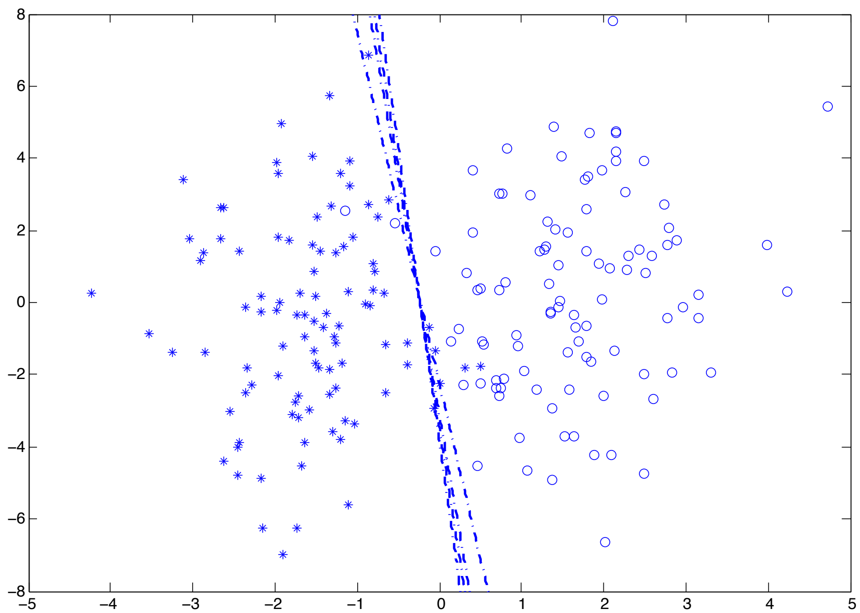
$\lambda = 6$

$\alpha(t) = \frac{1}{t}$

$\eta(t) = 0.8$



After 20 iterations



After 500 iterations

Case with imperfect communications

$$\text{minimize} \sum_{i=1}^{m} \underbrace{\left(\frac{\lambda}{2m} \|x\|^2 + \sum_{\ell \in S_i} \max\{0, 1 - y_\ell(x' z_\ell + u)\} \right)}_{f_i(\mathbf{x})} \quad \text{over } \mathbf{x} = (x, u) \in \mathbb{R}^n \times \mathbb{R}$$

$$w^i(t+1) = \mathbf{x}(t) - \eta(t) \sum_{j=1}^{m} r_{ij}(\mathbf{x}^j(t) + \underbrace{\xi_{ij}(t)}_{\textbf{noise}})$$

with $w_{ij} = 0$ when $j \notin N_i$, $\eta(t) > 0$ is a noise-damping stepsize

$$\mathbf{x}^i(t+1) = w^i(t+1) - \alpha(t) g_i(w^i(t+1))$$

Noise-damping stepsize $\eta(t)$ has to be coordinated with sub-gradient related stepsize $\alpha(t)$

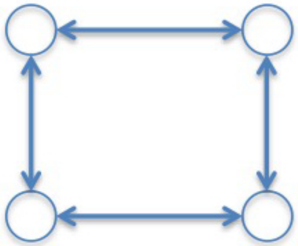
$$\sum_t \alpha(t) = \infty, \qquad \sum_t \alpha^2(t) < \infty$$

$$\sum_t \eta(t) = \infty, \qquad \sum_t \eta^2(t) < \infty$$

$$\sum_t \alpha(t)\eta(t) < \infty, \qquad \sum_t \frac{\alpha^2(t)}{\eta(t)} < \infty$$

Case with imperfect communications

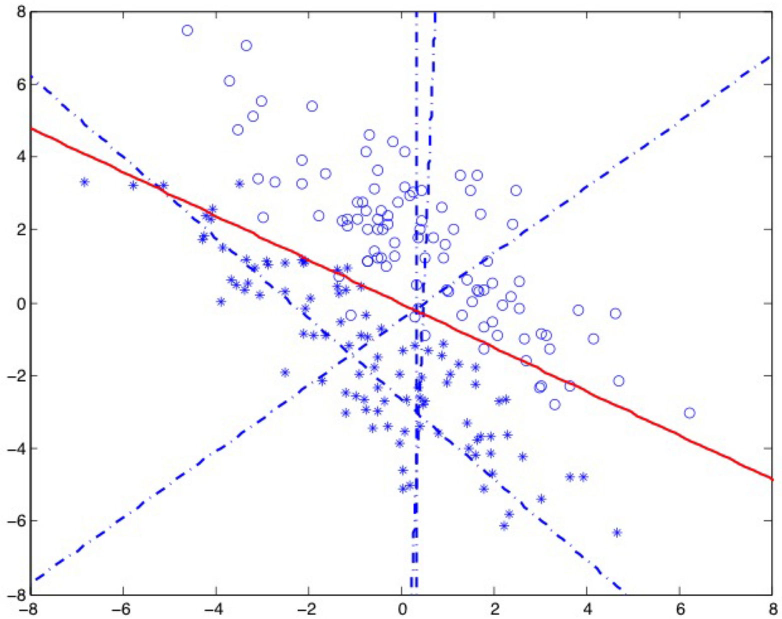
Illustration uses a simple graph of 4 nodes organized in a ring-network



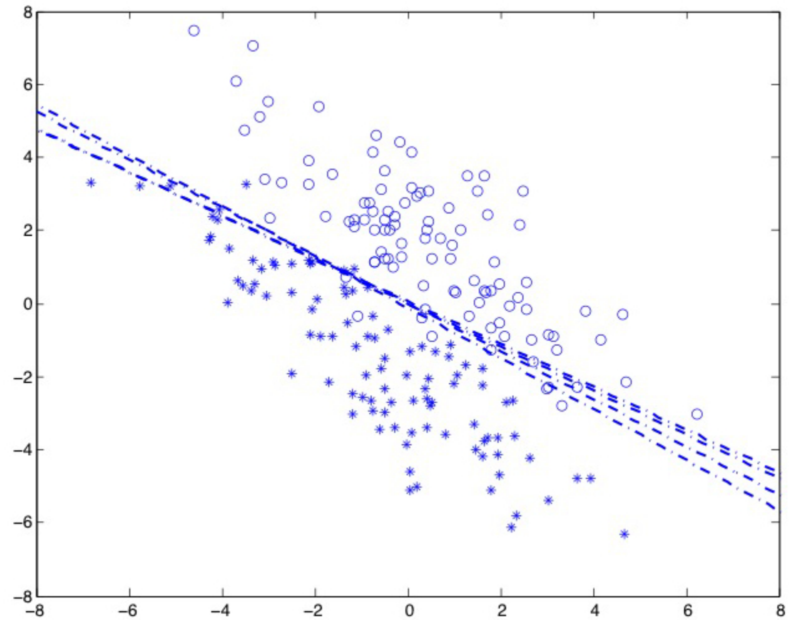
$\lambda = 6$

$\alpha(t) = \frac{1}{t}$

$\eta(t) = \frac{1}{t^{0.55}}$



After 1 iteration



After 500 iterations

Minimizing Max of Agents' Objectives

Network objective: minimize $\max_i f_i(x)$ over $x \in X \subseteq \mathbb{R}^n$

Makes sense when "fair network resource-utilization" is desired

Use epi-graph reformulation of the problem

$$\begin{array}{ll} \text{minimize} & \eta \\ \text{subject to} & f_i(x) \leq \eta \text{ for all } i = 1, \ldots, m, \;\; x \in X, \;\; \eta \in \mathbb{R} \end{array} \tag{1}$$

Under Slater condition (satisfied here) the strong duality holds for problem (1) and its dual

$$\text{maximize} \quad q(\mu) \quad \text{over } \mu \geq 0, \mu \in \mathbb{R}^m,$$

$$\text{where} \quad q(\mu) = \min_{x \in X, \eta \in \mathbb{R}} \left(\eta + \sum_{i=1}^{m} \mu_i (f_i(x) - \eta) \right)$$

So the problem can be solved by using a **primal-dual algorithm** or a **penalty approach**

Consider penalty approach: problem (1) is replaced with an equivalent "penalized" problem

$$\begin{array}{ll} \text{minimize} & F(x, \eta) = \eta + \sum_{i=1}^{m} \mathbf{c}_i (f_i(x) - \eta) \\ \text{subject to} & (x, \eta) \in X \times \mathbb{R} \end{array}$$

where $c_i > 1$ for all i.

Distributed Algorithm for Min-Max Optimization

$$\text{minimize} \qquad F(x,\eta) = \sum_{i=1}^{m} \underbrace{\left(\frac{\eta}{m} + c_i(f_i(x) - \eta)\right)}_{F_i(x,\eta)}$$

$$\text{subject to} \qquad (x,\eta) \in X \times \mathbb{R}$$

where $c_i > 1$ for all i.

$$F_i(x,\eta) = \left(\frac{1}{m} - 1\right)\eta + c_i f_i(x)$$

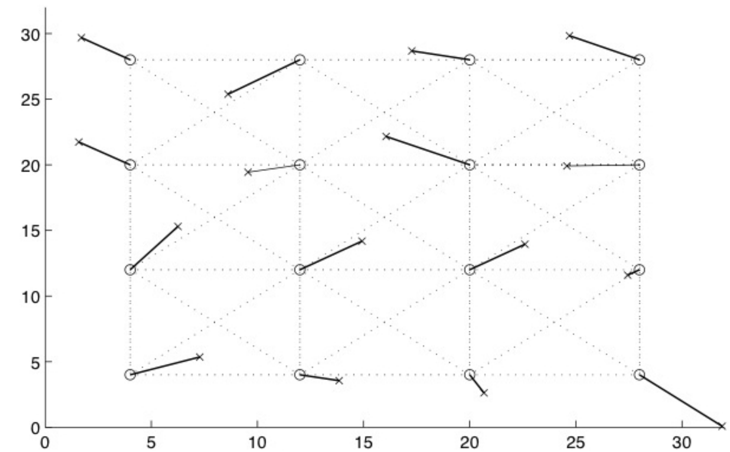
Observations:

- Each agent can choose its own c_i, as long as $c_i > 1$
- Every agent has to know m
- $\nabla_x F_i(x,\eta) = c_i \nabla f_i(x)$ and $\nabla_\eta F_i(x,\eta) = \frac{1}{m} - 1$

Example: Uplink Power Control

- m mobile users (MU) communicate with respective base stations (BS)
- p_i - power used by i^{th} MU to communicate with i^{th} BS
- $\mathbf{p} = (p_1, \ldots, p_m)$ power-allocation vec.
- $h_{i,j}$ - channel coefficient for MU j and BS i
- $\mathbf{h}_i = (h_{i1}, \ldots, h_{im})$ channel coef.vec. for i
- σ_i^2 - receiver noise variance
- SINR at BS i is given by

$$\gamma_i(\mathbf{p}, \mathbf{h}_i) = \frac{p_i h_{i,i}^2}{\sigma_i^2 + \sum_{j \neq i} p_j h_{i,j}^2},$$



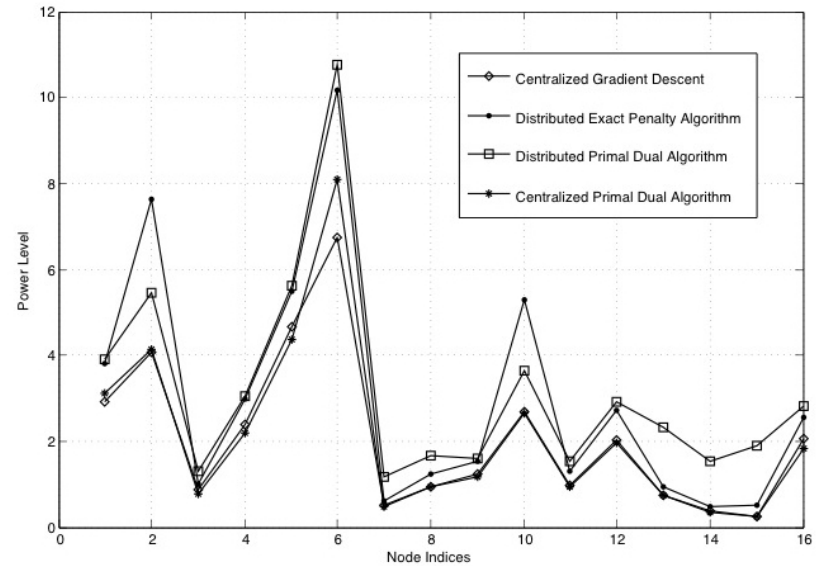
The circles denote the base stations. The dotted lines denote the communication links between adjacent BSs. The cross denotes the MUs. The bold lines connect each MU to its respective BS.

- $U_i(\gamma_i(\mathbf{p}, \mathbf{h}_i))$ is the utility for BS i based on SINR
- $V(p_i)$ is a cost function penalizing excessive power
- We are interested in computing the min-max fair allocation

$$\min_{\mathbf{p} \in \Pi} \max_{i \in V} \left[V(p_i) - U_i(\gamma_i(\mathbf{p}, \mathbf{h}_i)) \right], \ \Pi = \{\mathbf{p} \in \mathbb{R}^m \mid 0 \leq p_i \leq p_{\max} \text{ for all } i\}$$

Example: Uplink Power Control

- Generally a non-convex problem
- Logarithmic utility $U_i(y) = \ln(y)$
- Linear power cost $V(p_i) = ap_i$, $a > 0$
- The coordinate transformation $p_i = e^{x_i}$ makes the problem a convex optimization problem (in x)



The final iterate values after 2000 iterations of the algorithm for step size $\alpha_k = \frac{50}{k^{0.65}}$.

- The resulting convex problem is

$$\min_{x \in X} \max_{i \in V} f_i(x),$$

$$f_i(x) = \ln\left(\sigma_i^2 h_{i,i}^{-2} e^{-x_i} + \sum_{j \neq i} h_{i,i}^{-2} h_{j,i}^2 e^{x_j - x_i}\right) + V(e^{x_i})$$

and $X = \{x \mid x_i \leq \ln(p_{\max}) \text{ for all } i\}$.

Conclusions

- Considered algorithms for distributed optimization over network
- Illustrated them on data classification problem
- Considered dynamic TU games over networks
  - Dynamic in the game and in the player's network connectivity
- Discussed distributed allocation algorithms that converge to an allocation in the core of the limiting game