Games on Time-varying Networks

Angelia Nedić
Department of Industrial and Enterprise Systems Engineering
University of Illinois at Urbana-Champaign

Tuesday 28th July, 2015

Outline

Duopolistic Cournot Model (1838)

- Predates Nash equilibrium ideas
- Firms 1 and 2 produce items $q_1 \geq 0$ and $q_2 \geq 0$; let $Q = q_1 + q_2$
- Let $p(Q)$ be the price: $p(Q) = a - Q$, with $P(Q) = 0$ when $Q > a$
- Cost of production is $C_i(q_i) = cq_i$ with $c < a$ Normal form:
  - $S_i = [0, \infty)$ - strategy set; decision $q_i \geq 0$
  - Each firm wants to maximize its profit given the decision of the other firm

$$(\text{firm } i) \ : \max_{q_i \geq 0} \quad q_i(a - (q_i + q_{-i})) - cq_i \text{for } q_{-i} \text{ fixed}$$

Solution is a Nash equilibrium point (q_1^*, q_2^*):

$$q_1^* = \arg\max_{q_1 \geq 0} \quad q_1(a - (q_1 + q_2^*)) - cq_1$$
$$q_2^* = \arg\max_{q_2 \geq 0} \quad q_2(a - (q_1^* + q_2)) - cq_2$$

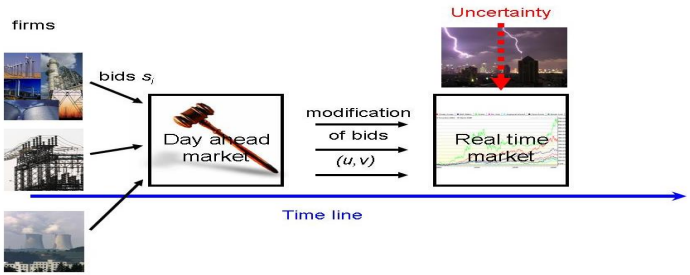
- Nash equilibrium in quantities:

$$q_1^* = \frac{1}{2}(a - q_2^* - c)$$
$$q_2^* = \frac{1}{2}(a - q_1^* - c)$$
$$q_1^* = q_2^* = \frac{a - c}{3}.$$

- Intuition: If $q_2^* = 0$, $q_1^* = \frac{a-c}{2}$, a monopolist level

Electricity markets

A 2-node Market



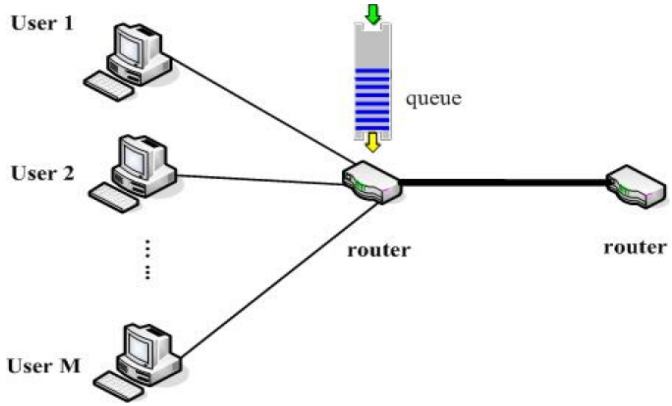
- Consider a 2-node electricity market. Suppose node 1 has a demand center while node 2 houses n electricity producers
- Producer i produces q_i MW (≥ 0) during a specific hour with capacity C_i
- Producer i's costs are $c_i q_i + \frac{1}{2} d_i q_i^2$.
- Transmission capacity of the link $(1, 2)$ is infinite
- Price of power at node 1 is given by $p(Q)$, $Q = \sum_{i=1}^{N} q_i$.
- Producers can affect prices and their payoffs are dependent on the generation levels of the remaining firms
- Specifically, producer i is faced by

$$\mathrm{G}_i(q_{-i}): \quad \begin{array}{ll} \text{maximize} & p(Q)q_i - c_i q_i - \frac{1}{2} d_i q_i^2 \\ \text{subject to} & q_i \leq C_i \\ & q_i \geq 0, \end{array}$$

  where $q_{-i} = (q_j)_{j \neq i}$.
- Nash equilibrium in generation levels: $(q_1^*, \ldots, q_n^*)$ where $q_i^* \in \arg\max_{0 \leq q_i \leq C_i} G_i(q^*{}_{-i})$ for all i.

Flow control games

- Multitude of applications on the internet
- A large number of users - inherently noncooperative in nature



- Demands for bandwidth lead to congestion
- Solution concept - a Nash equilibrium in flow rates, where each agent maximizes its utility less cost of sending flow

- User i has utility given by $U_i(x_i) := \ln(1 + x_i) + d_i,\ x_i \geq 0,$
- Cost of sending flow is $P_i(x_i, x_{-i}) := k_i x_i^2 \left(C - \sum_{i=1}^N x_i\right)^{-1}$
- Finally, flow decisions are required to stay feasible with respect to the capacity constraint $\sum_{i=1}^N x_i \leq C$.

Nash Equilibrium in flow decisions

- Users maximize the benefit of sending flow by solving

$$\begin{array}{lll} \mathrm{D}_i(x_{-i}) & \text{maximize} & U_i(x_i) - P_i(x_i; x_{-i}) \\ & \text{subject to} & x_i \geq 0,\ x_i + \sum_{j \neq i} x_j \leq C \end{array}$$

- Nash equilibrium in flow generation levels: $(x_1^*, \ldots, x_n^*)$ where $x_i^* \in \arg\max D_i(x_{i-}i, *)$, for all i and $\displaystyle\sum_{i=1}^N x_i^* \leq C$.

Continuous Games – Mixed-Strategy Nash Equilibrium

- In a host of practical settings, strategy decisions are continuous variables (as opposed to discrete such as in the bimatrix games).

Definition (Mixed-strategy Nash equilibria)

In the normal form game $G = \{S_1, \ldots, S_N; u_1, \ldots, u_N\}$, let $S_i = \{s_{i1}, \ldots, s_{iK_i}\}$. Then a mixed-strategy for player i is a distribution $p_i = (p_{i1}, \ldots, p_{iK_i})$, where $0 \leq p_{ik} \leq 1$ for all k and $\sum_{k=1}^{K_i} p_{ik} = 1$.

- A mixed strategy specifies the randomization over the set of pure strategies
- Mixed-strategies result in continuous decision variables (the probability masses p_i)

Continuous (Competitive) Game

- A number N of players (agents, users, nodes) indexed by $i = 1, \ldots, N$
- Each player i is controlling a decision variable $x_i \in \mathbb{R}^{n_i}$
- A player i is interested in minimizing its own cost $f_i(x_1, \ldots, x_N)$ in x_i subject to some (continuous) constraint set $K_i \subseteq \mathbb{R}^{n_i}$, given the decisions $\{x_j, j \neq i\}$ of other players:

(Player i problem)

$$\begin{array}{rl} \text{minimize} & f_i(x_1, \ldots, x_N) \\ \text{subject to} & x_i \in K_i \end{array}$$

- Game solutions: Nash equilibrium points $(x_1^*, \ldots, x_N^*)$ characterized by: for all $i = 1, \ldots, N$,

$$x_i^* \in \text{Arg} \min_{x_i \in K_i} f_i(x_1^*, \ldots, x_{i-1}^*, x_i, x_{i+1}^*, \ldots, x_N^*)$$

Outline

Variational Inequality Re-formulation

Let $x = (x_1, \ldots, x_N)$
For each i and a given x, let x_{-i} be the collection $(x_j)_{j \neq i}$
We will focus on convex games, where for every $i = 1, \ldots, N$:

- The functions $f_i(x_i; x_{-i})$ are convex and differentiable in x_i for every fixed x_{-i}
- The sets $K_i \subseteq \mathbb{R}^{n_i}$ are nonempty, closed and convex

Then, by using optimality conditions for Nash-equilibria we see that the Nash-equilibrium points are equivalently characterized by: for all $i = 1, \ldots, N$

$$\underbrace{\langle \nabla_{x_i} f_i(x^*), x_i - x_i^* \rangle \geq 0 \qquad \text{for all } x_i \in K_i,}_{\text{variational inequality}}$$

where $\nabla_{x_i} \phi(x)$ is the partial-gradient of a function ϕ with respect to the block-variable x_i

By stacking all inequalities we define

$$F(x) = \begin{bmatrix} \nabla_{x_1} f_1(x^*) \\ \nabla_{x_2} f_2(x^*) \\ \vdots \\ \nabla_{x_N} f_N(x^*) \end{bmatrix},$$

and the (Cartesian product) set

$$K = K_1 \times K_2 \times \cdots \times K_N.$$

With these, the Nash-equilibrium points x^* are characterized by the following (Cartesian) variational inequality

$$\langle F(x^*), x - x^* \rangle \geq 0 \qquad \text{for all } x \in K.$$

The variational inequality problem of determining a point $x^* \in K$ that satisfies the above relation is denoted by $\mathrm{VI}(K, F)$

Existence of Solutions

Consider VI(K, F) consisting of finding a point $x^* \in K$ such that

$$\langle F(x^*), x - x^* \rangle \geq 0 \qquad \text{for all } x \in K.$$

Such an x^* is termed a (Nash equilibrium) solution of VI(K, F).

Theorem (Solution Existence/Uniqueness [FP03])

Let K be a compact set and let F be a continuous map. Then:

- *VI(K, F) has a solution.*
- *If F is monotone on K, i.e.,*

$$\langle F(x) - F(y), x - y \rangle \geq 0 \qquad \text{for all } x, y \in K.$$

Then, the solution is unique.

Solution Approaches

Consider VI(K, F) and assume that it has a solution $x^* \in K$.
It can be seen that $x^* \in K$ *solves VI*(K, F) *if and only if* x^* *is a fixed point* of a certain map, i.e.,

$$x^* = \Pi_K[x^* - \tau F(x^*)] \qquad \text{for any } \tau > 0,$$

where $\Pi_K[y]$ is the Euclidean projection of y on the set K:

$$\Pi_K[y] = \arg\min_{z \in K} \|z - y\|^2.$$

A class of iterative approaches is based on a fixed-point method for the map $y \mapsto \Pi_K[y - \tau(y)F(y)]$:

$$x^{k+1} = \Pi_K[x^k - \tau_k F(x^k)] \qquad \text{with } \tau_k > 0, \text{ starting with some } x^0 \in K.$$

Computations as seen by players

Our setting $F(x) = [\nabla_{x_1} f_1(x), \nabla_{x_2} f_2(x), \cdots, \nabla_{x_N} f_N(x)]$ and $K = K_1 \times \cdots \times K_N$.
Due to the structure of F and K, we have

$$\begin{aligned} & x^{k+1} = \Pi_K[x^k - \tau_k F(x^k)] \\ \iff \quad & x_i^{k+1} = \Pi_{K_i}[x_i^k - \tau_k \nabla_{x_i} f_i(x^k)] \text{ for } i = 1, \ldots, N. \end{aligned}$$

The later is also known as *gradient play*, as the players adjust their decisions based on gradient information of their private cost functions
Another popular approach is *best response play*, whereby players take turns in updating their decisions and choose the best decision given what the other players have chosen
At time k agent i is updating according to "best response", as follows:

$$x_i^{k+1} \in \text{Arg} \min_{x_i \in K_i} f_i(x_1^k, \ldots, x_{i-1}^k, x_i, x_{i+1}^k, \ldots, x_N^k).$$

The gradient play is natural where the costs are complicated (engineered systems).

Gradient Play: Computations distributed across players?

$$x_i^{k+1} = \Pi_{K_i}[x_i^k - \tau_k \nabla_{x_i} f_i(x^k)] \text{ for } i = 1, \ldots, N$$

Computations are "coupled" as each evaluation of $\nabla_{x_i} f_i(x^k)$ requires access to all players decisions $x^k = (x_1^k, \ldots, x_N^k)$.

- Most of the existing literature on VI's is not concerned with "distributed" aspect
- In some special instances, such as Electricity Markets and Network Flow Games [Sri04, SS07, KNS09, KNS11], an entity collects decisions x_i^k from all players and returns "prices" as a mechanism to guide players actions toward a Nash equilibrium
- Another special class of "aggregative games" is discussed next

Outline

An Aggregative Game

We have a system with N agents, where each agent i solves the following problem:

$$\begin{aligned} \text{minimize} \quad & f_i(x_i, \bar{x}), \quad \bar{x} = \sum_{j=1}^{N} x_j \\ \text{subject to} \quad & x_i \in K_i \subseteq \mathbb{R}^n. \end{aligned} \tag{AggGame}$$

- $f_i(\cdot,\cdot)$ and K_i are known only to agent i
- Agent i decides on x_i, and does not have access to the decisions x_j of the other agents
- Unlike [Jensen06, MartimortStole2010], the emphasis here is on computation of an equilibrium in the absence of instantaneous access to the aggregate value $\bar{x}$
- This will be achieved by allowing agents to communicate locally over a connected network

Motivating Example

Example (Networked Nash-Cournot Oligopoly)

- N firms competing at M locations
- Optimization problem for firm i

$$\begin{aligned} \text{minimize} \quad & \sum_{\ell=1}^{M} \left(c_{i\ell}(x_{i\ell}) - p_\ell(\bar{s}_\ell)s_{i\ell}\right) + t_i(x_{i1},\ldots,x_{iM}) \\ \text{subject to} \quad & \sum_{\ell=1}^{M} x_{i\ell} = \sum_{\ell=1}^{M} s_{i\ell}, \\ & x_{i\ell}, s_{i\ell} \geq 0, \quad x_{i\ell} \leq \text{cap}_{i\ell}, \qquad \ell = 1,\ldots,M. \end{aligned}$$

- $x_{i\ell}$ and $s_{i\ell}$ are production and sales for firm i at location ℓ
- p_ℓ denotes the price function and $\bar{s}_\ell = \sum_{j=1}^{N} s_{j\ell}$
- $p_\ell(\bar{s}_\ell)s_{i\ell}$ is the revenue at location ℓ for firm i
- t_i is the transportation cost

Computational algorithm

$$\begin{aligned} \text{minimize} \quad & f_i(x_i, \bar{x}), \quad \bar{x} = \sum_{j=1}^{N} x_j \\ \text{subject to} \quad & x_i \in K_i \subseteq \mathbb{R}^n. \end{aligned} \tag{AggGame}$$

- Are there distributed algorithms ?? [1]
- Under ideal conditions and suitable assumption

$$x_i^{k+1} = \Pi_{K_i}[x_i^k - \alpha \nabla_{x_i} f_i(x_i^k, \ \bar{x}^k \ )]$$

- What is the challenge then?

[1] Projection based [Facchinei-Pang03, Alpcan03] and their regularized variants [Kannan-Shanbhag10]

Computational algorithm

$$\begin{aligned} \text{minimize} \quad & f_i(x_i, \bar{x}), \quad \bar{x} = \sum_{j=1}^{N} x_j \\ \text{subject to} \quad & x_i \in K_i \subseteq \mathbb{R}^n. \end{aligned} \tag{AggGame}$$

- Are there distributed algorithms ?? [1]
- Under ideal conditions and suitable assumption

$$x_i^{k+1} = \Pi_{K_i}[x_i^k - \alpha \nabla_{x_i} f_i(x_i^k, \ \bar{x}^k \ )]$$

- What is the challenge then?
  - No agent/entity knows what is $\bar{x}^k$

[1] Projection based [Facchinei-Pang03, Alpcan03] and their regularized variants [Kannan-Shanbhag10]

Computational algorithm

$$\begin{aligned} \text{minimize} \quad & f_i(x_i, \bar{x}), \quad \bar{x} = \sum_{j=1}^{N} x_j \\ \text{subject to} \quad & x_i \in K_i \subseteq \mathbb{R}^n. \end{aligned} \tag{AggGame}$$

- Are there distributed algorithms ?? [1]
- Under ideal conditions and suitable assumption

$$x_i^{k+1} = \Pi_{K_i}[x_i^k - \alpha \nabla_{x_i} f_i(x_i^k, \ \bar{x}^k \ )]$$

- What is the challenge then?
  - No agent/entity knows what is $\bar{x}^k$
  - No central entity exists to provide it.

[1] Projection based [Facchinei-Pang03, Alpcan03] and their regularized variants [Kannan-Shanbhag10]

Computational algorithm

$$\begin{aligned} \text{minimize} \quad & f_i(x_i, \bar{x}), \quad \bar{x} = \sum_{j=1}^{N} x_j \\ \text{subject to} \quad & x_i \in K_i \subseteq \mathbb{R}^n. \end{aligned} \tag{AggGame}$$

- Are there distributed algorithms ?? [1]
- Under ideal conditions and suitable assumption

$$x_i^{k+1} = \Pi_{K_i}[x_i^k - \alpha \nabla_{x_i} f_i(x_i^k, \ \bar{x}^k \ )]$$

- What is the challenge then?
  - No agent/entity knows what is $\bar{x}^k$
  - No central entity exists to provide it.

↘ [1] Projection based [Facchinei-Pang03, Alpcan03] and their regularized variants [Kannan-Shanbhag10]

Computational algorithm

$$\begin{aligned} \text{minimize} \quad & f_i(x_i, \bar{x}), \quad \bar{x} = \sum_{j=1}^{N} x_j \\ \text{subject to} \quad & x_i \in K_i \subseteq \mathbb{R}^n. \end{aligned} \tag{AggGame}$$

- Are there distributed algorithms ?? [1]
- Under ideal conditions and suitable assumption

$$x_i^{k+1} = \Pi_{K_i}[x_i^k - \alpha \nabla_{x_i} f_i(x_i^k, \bar{x}^k)]$$

- What is the challenge then?
  - No agent/entity knows what is $\bar{x}^k$
  - No central entity exists to provide it.

↘ [1] Projection based [Facchinei-Pang03, Alpcan03] and their regularized variants [Kannan-Shanbhag10]

The central theme

Goal :

- Develop distributed algorithms that can aid agents to learn the aggregate and lead to an equilibrium point
- And still satisfy the informational constraints

How?

The central theme

Goal :

- Develop distributed algorithms that can aid agents to learn the aggregate and lead to an equilibrium point
- And still satisfy the informational constraints

How?

- By allowing agents to communicate their beliefs of the aggregate: $(\bar{x} = \sum_{i=1}^{N} x_i)$

The central theme

Goal :

- Develop distributed algorithms that can aid agents to learn the aggregate and lead to an equilibrium point
- And still satisfy the informational constraints

How?

- By allowing agents to communicate their beliefs of the aggregate: $(\bar{x} = \sum_{i=1}^{N} x_i)$
- But not their decisions: (x_i)
- This is in sharp contrast with "distributed optimization" setting (since here agents are competing)

Assumptions for a Unique Equilibrium

Basic Assumption 1

For each $i = 1, \ldots, N$,

- The set $K_i \subset \mathbb{R}^n$ is compact and convex
- Function $f_i(x_i, y)$ is continuously differentiable in (x_i, y) over some open set containing the set $K_i \times \bar{K}$, where $\bar{K} = \sum_{i=1}^{N} K_i$
- The function $x_i \mapsto f_i(x_i, s(x))$ is convex over the set K_i.

- These assumptions are sufficient for the existence of an equilibrium.

Additional Assumption: Uniqueness

- Define mapping $\Phi(x)$ by

$$\Phi(x) \triangleq \begin{pmatrix} F_1(x_1, s(x)) \\ \vdots \\ F_N(x_N, s(x)) \end{pmatrix}$$

$$F_i(x_i, s(x)) = \nabla_{x_i} f_i(x_i, s(x)), \qquad s(x) = \sum_{j=1}^{N} x_j$$

Basic Assumption 2 (Strict Monotonicity)

The mapping $\Phi(x)$ is strictly monotone over $K_1 \times \cdots \times K_N$, i.e.,

$$(\Phi(x) - \Phi(x'))^T (x - x') > 0, \quad \forall x, x' \in K_1 \times \cdots \times K_N.$$

Equilibrium conditions

Unique Equilibrium

Consider the aggregative Nash game defined in (AggGame). Suppose Basic Assumptions hold. Then, the game admits a unique Nash equilibrium.

Further Assumptions

Basic Assumption 3 [Computational]

The mapping $F_i(x_i, u)$ is uniformly Lipschitz continuous in u over $\bar{K} = K_1 + \cdots + K_N$, for every fixed $x_i \in K_i$ i.e., for some $L_{-i} > 0$

$$\|F_i(x_i, u) - F_i(x_i, z)\| \leq L_{-i}\|u - z\|.$$

Outline

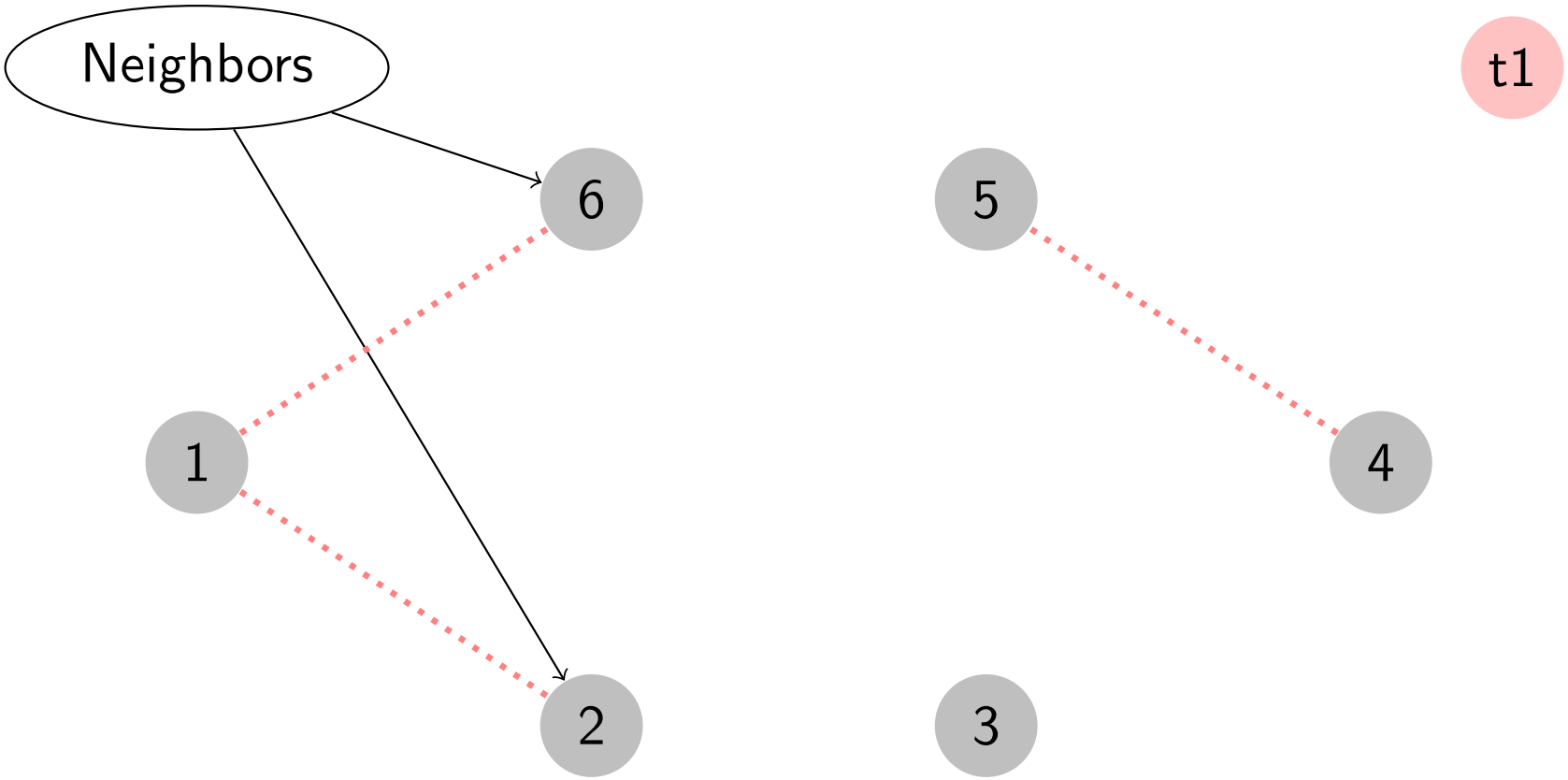
Synchronous setting

- A time varying undirected connectivity graph $\mathcal{G}_k = (\mathcal{N}, \mathcal{E}_k)$
- $\mathcal{N} = \{1, \ldots, N\}$ be a set of N agents
- $\mathcal{E}_k$ is the set of edges at time k
- $\mathcal{N}_i(k)$: immediate neighbors of agent i at time k

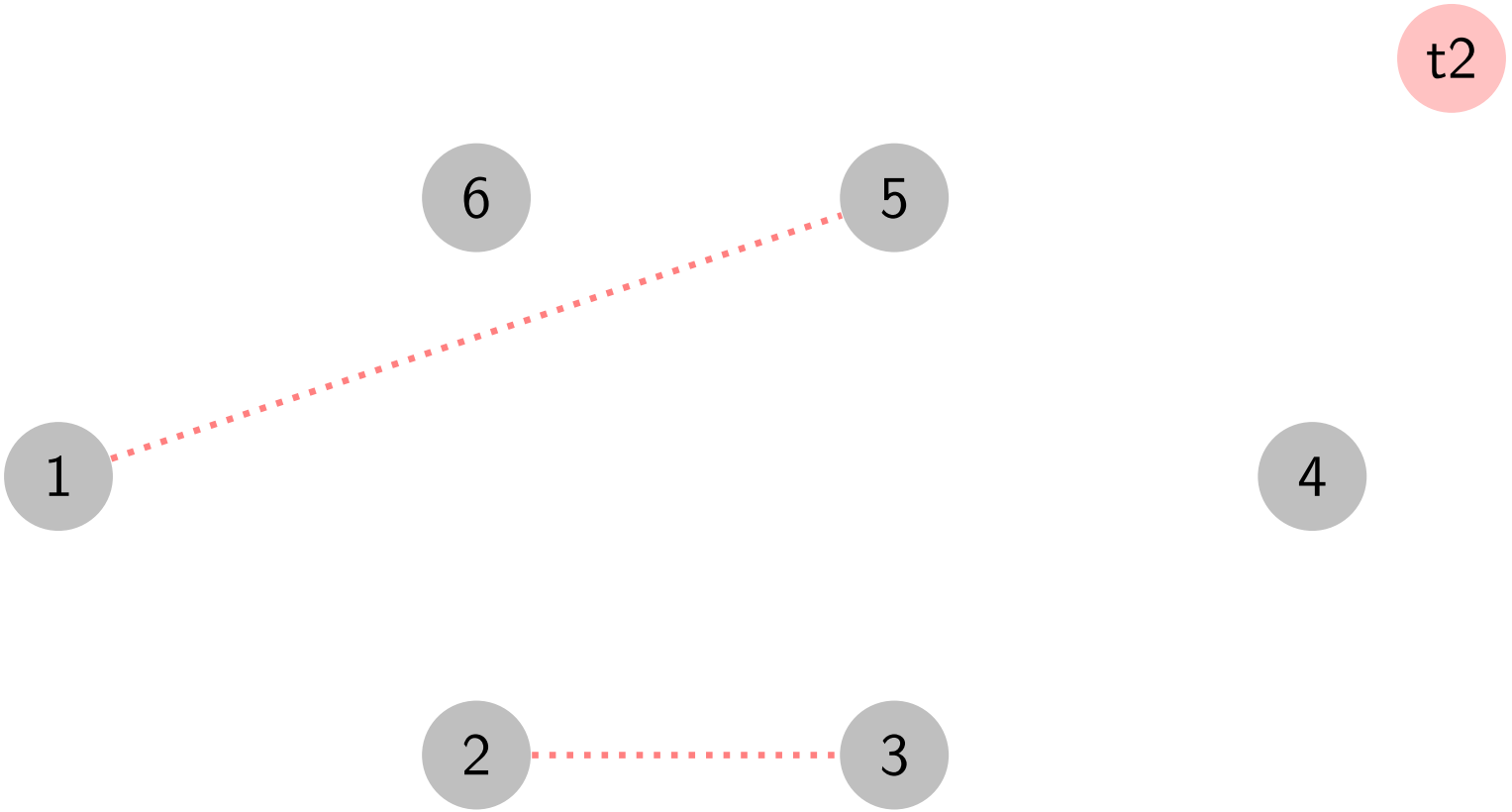
Connectivity Assumption

There exists an integer $Q \geq 1$ such that the graph $\bigcup_{l=1}^{Q} \mathcal{G}_{l+k}$ is connected for all k.

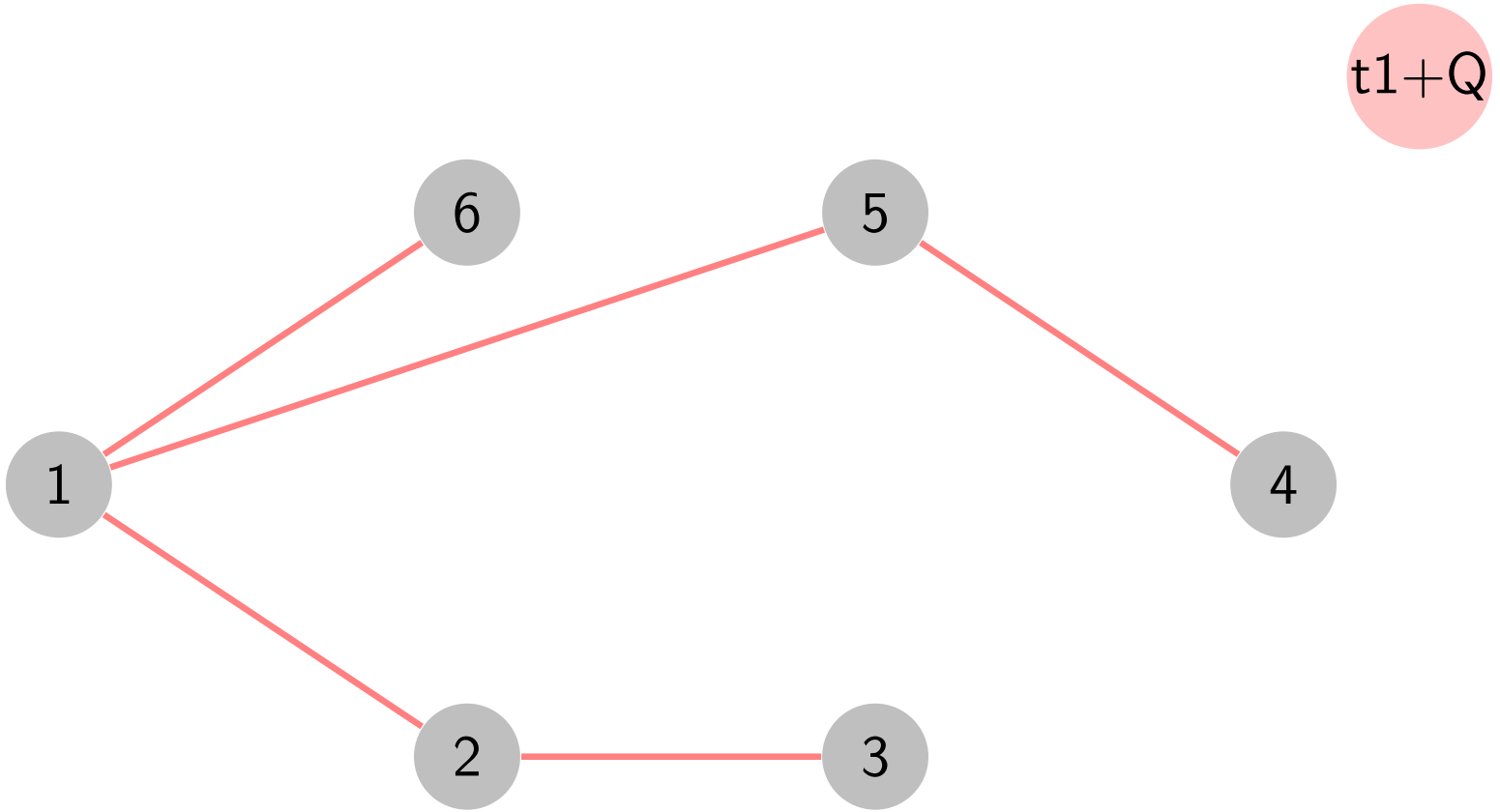
Limited connectivity of agents



Limited connectivity of agents



Limited connectivity of agents



Variational Inequality Formulation

We want to determine the point $x^* = (x_1^*, \ldots, x_N^*) \in K$ such that

$$\langle \Phi(x^*), x - x^* \rangle \geq 0 \qquad \text{for all } x \in K,$$

where

$$\Phi(x) \triangleq \begin{pmatrix} F_1(x_1, s(x)) \\ \vdots \\ F_N(x_N, s(x)) \end{pmatrix}$$

$$F_i(x_i, s(x)) = \nabla_{x_i} f_i(x_i, s(x)), \qquad s(x) = \sum_{j=1}^{N} x_j$$

We use a suitable analog of

$$x_i^{k+1} = \Pi_{K_i}[x_i^k - \alpha \nabla_{x_i} f_i(x_i^k, s(x^k))$$

where we distribute learning of $s(x^k)$ in the network

Outline of the synchronous algorithm: the three steps

1. Learn aggregate:[2]
   - Each player maintains and updates an estimate v_i^k of $s(x^k) = \sum_{i=1}^N x_i^k$ (tricky since x_j^k's are players' private variables & are changing in time)
   - Every player communicates its estimate v_i^k to the neighbors
2. Update decision:
   - Using aligned estimate agents update their decision (x_i^k)
3. Update estimate of aggregate:
   - Agents update their estimate to reflect their current decision.

[2] Inspiration: Consensus based algorithm [Tsi84, SRNV12], Distributed optimization problem [NO09, SNS13]

Synchronous algorithm

1. Learn aggregate:

$$\hat{v}_i^k = \sum_{j \in \mathcal{N}_i(k)} w_{ij}(k) v_j^k, \quad v_i^0 = x_i^0,$$

$w_{ij}(k)$: agent i's weight to agent j's information at step k

2. Update decision:

$$x_i^{k+1} = \Pi_{K_i}[x_i^k - \alpha_k F_i(x_i^k, N\hat{v}_i^k)].$$

where agent i uses its estimate $N\hat{v}_i^k$ instead of $\sum_{j=1}^N x_j^k$

3. Update estimate of aggregate:

$$v_i^{k+1} = \hat{v}_i^k + x_i^{k+1} - x_i^k$$

Step suggested by Ram in [Ram-Nedić-Veeravali 2012]

Weight assumptions

Synchronous Weight

For all $i \in \mathcal{N}$ and all $k \geq 0$, the following hold:

(i) $w_{ij}(k) \geq \delta$ for all $j \in \mathcal{N}_i(k)$ and $w_{ij}(k) = 0$ for $j \notin \mathcal{N}_i(k)$;

(ii) $\sum_{j=1}^{N} w_{ij}(k) = 1$ for all i;

(iii) $\sum_{i=1}^{N} w_{ij}(k) = 1$ for all j.

Stepsize assumptions

Synchronous Stepsize

The stepsize α_k is chosen such that the following hold:

(i) The sequence $\{\alpha_k\}$ is monotonically non-increasing i.e., $\alpha_{k+1} \leq \alpha_k$ for all k;

(ii) $\sum_{k=0}^{\infty} \alpha_k = \infty$;

(iii) $\sum_{k=0}^{\infty} \alpha_k^2 < \infty$.

Convergence of synchronous algorithm

Proposition ([KNS12a])

Let Basic Assumptions hold. Further let the Synchronous Weight and Synchronous Stepsize assumption hold. Then, the sequence $\{x^k\}$ generated by the Synchronous Algorithm converges to the (unique) solution x^* of the game.

Proof sketch

1. Gradient projection algorithm technique ($\|x^k - x^*\|$)
   - Utilizing the Lipschitz continuity of the map leads to a bound on $\|x_i^{k+1} - x_i^*\|$ in terms of $\|Nv_i^k - \bar{x}^*\|$
2. Estimate the errors for the beliefs ($\|Nv_i^k - \bar{x}^*\|$)
   - Introduce $y^k = \sum_{\ell=1} v_\ell^k / N$, the average estimate under perfect information
   - y^k also tracks the average of the aggregate i.e., $y^k = \sum_{\ell=1}^N x_\ell^k / N$
   - Construct bounds for $\|v_i^k - y^k\|$ and $\|Ny^k - \bar{x}^*\|$

Moving ahead

Convergence established for synchronous regime but
Key Drawback:

- Synchronization is a challenge in large networks
- Requires coordination in terms of stepsize

Remedy: Desynchronization

- No explicit dependency
- Allows independent solution of stepsizes
- Cannot accommodate time-varying graphs

Outline

Asynchronous setting

Connected Graph

The undirected graph $\mathcal{G} = (\mathcal{N}, \mathcal{E})$ is connected.

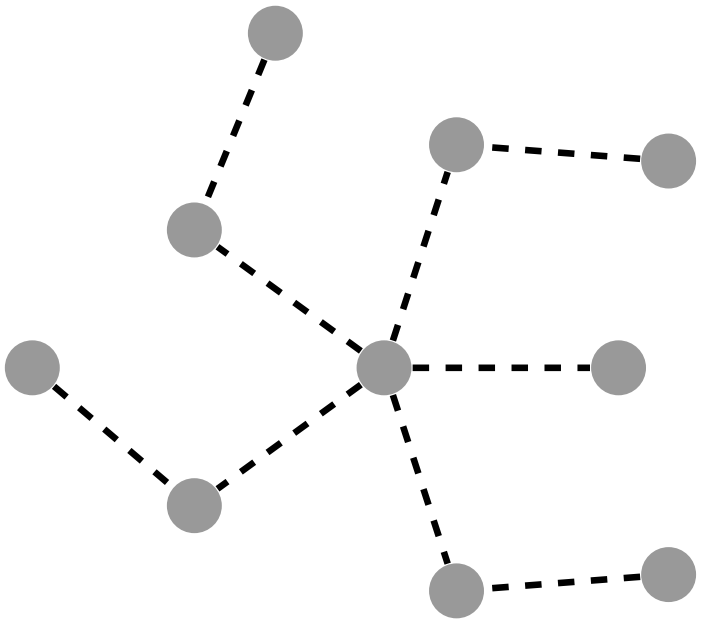


Figure: A depiction of a gossip communication.

Asynchronous setting

Connected Graph

The undirected graph $\mathcal{G} = (\mathcal{N}, \mathcal{E})$ is connected.

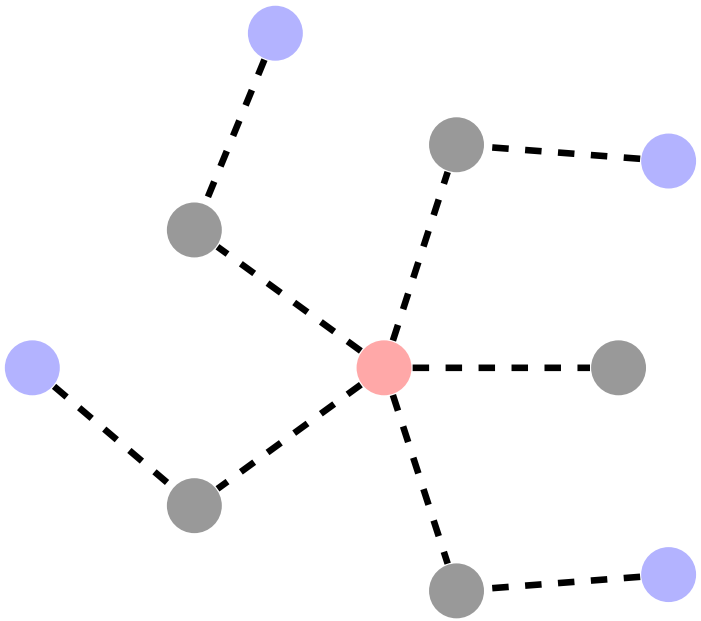


Figure: A depiction of a gossip communication.

Asynchronous setting

Connected Graph

The undirected graph $\mathcal{G} = (\mathcal{N}, \mathcal{E})$ is connected.

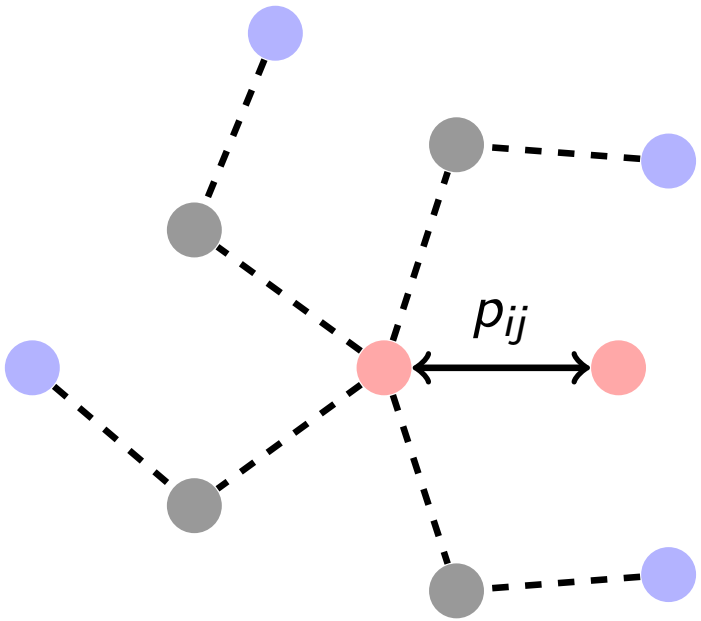


Figure: A depiction of a gossip communication.

Outline of the asynchronous algorithm: the four steps

1. Agents have local clocks ticking at rate 1
2. Wake Up:
   - Agent I^k wakes up at time k and then contacts J^k
3. Learn aggregate:
   - Only agents $\{I^k, J^k\}$ communicate their estimate (v_i^k) and perform mixing
4. Update decision:
   - Using aligned estimate agents update their decision (x_i^k)
5. Update estimate of aggregate:
   - Agents update their estimate to reflect their current decision.

Asynchronous algorithm

- Wake Up:
- Agent I^k wakes up at time k and then contacts J^k

1. Learn aggregate:

$$\hat{v}_i^k = \frac{v_{I^k}^k + v_{J^k}^k}{2} \quad \text{for } i \in \{I^k, J^k\}, \quad v_i^0 = x_i^0,$$

2. Update decision:

$$x_i^{k+1} = \left(\Pi_{K_i}[x_i^k - \alpha_{k,i} F_i(x_i^k, N\hat{v}_i^k)] - x_i^k\right) \mathbb{1}_{\{i \in \{I^k, J^k\}\}} + x_i^k$$

3. Update estimate of aggregate:

$$v_i^{k+1} = \hat{v}_i^k + x_i^{k+1} - x_i^k.$$

Stepsize Assumptions

Asynchronous Stepsize

The stepsize $\alpha_{k,i}$ is updated as follows:

$$\alpha_{k,i} = \frac{1}{\Gamma_k(i)}, \quad \text{where}$$

- $\Gamma_k(i)$ denotes the number of updates that agent i has executed up to time k inclusively.

Convergence of asynchronous algorithm

Proposition ([KNS12b])

Let Basic Assumption, Connected Graph and the Asynchronous Stepsize assumption hold. Then, the sequence $\{x^k\}$ generated by the Asynchronous Algorithm converges to the (unique) solution x^* of the game.

Convergence analysis of asynchronous algorithm

Key property: of some k large enough almost surely we have

$$\alpha_{k,i} \leq \frac{2}{kp_i}, \qquad \left|\alpha_{k,i} - \frac{1}{kp_i}\right| \leq \frac{2}{k^{3/2-q}p_{\min}^2}.$$

1. Stochastic gradient technique $\left(\mathbb{E}\left[\|x^k - x^*\|^2 \mid \mathcal{F}_{k-1}\right]\right)$
2. Estimate the expected errors for the beliefs $\left(\mathbb{E}\left[\|Nv_i^k - \bar{x}^*\| \mid \mathcal{F}_{k-1}\right]\right)$

Constant stepsize asynchronous algorithm

Agents employ a constant deterministic yet uncoordinated stepsize

- Wake Up:
- Agent I^k wakes up at time k and then contacts J^k

1. Learn aggregate:

$$\hat{v}_i^k = \frac{v_{I^k}^k + v_{J^k}^k}{2} \quad \text{for } i \in \{I^k, J^k\}, \quad v_i^0 = x_i^0,$$

2. Update decision:

$$x_i^{k+1} = \left(\Pi_{K_i}[x_i^k - \alpha F_i(x_i^k, N\hat{v}_i^k)] - x_i^k\right) \mathbb{1}_{\{i \in \{I^k, J^k\}\}} + x_i^k$$

3. Update estimate of aggregate:

$$v_i^{k+1} = \hat{v}_i^k + x_i^{k+1} - x_i^k.$$

Basic Assumption 4

The mapping $F_i(x_i, u)$ is uniformly Lipschitz continuous in x_i over K_i, for every fixed $u \in \bar{K}$ i.e., for some $L_i > 0$

$$\|F_i(x_i, u) - F_i(y_i, u)\| \leq L_i \|x_i - y_i\|;$$

Error bound: constant stepsize asynchronous algorithm

Proposition

Let Basic Assumptions hold with the mapping Φ be strongly monotone over the set K. Further let Connected Graph assumption hold. Then, the following holds for the sequence $\{x^k\}$ generated by the Asynchronous Algorithm with stepsize $\alpha_{k,i} = \alpha_i$

$$\limsup_{k\to\infty} \mathbb{E}[\|x^{k+1} - x^*\|^2] \leq Err(N, n, \mathcal{G}),$$

where x^* is the unique equilibrium of the game.

Expression for error

$$Err(N,n,\mathcal{G}) = \frac{4p_{\max}\alpha_{\max}^2 C^2 N + 2p_{\max}\alpha_{\max}^2 B\frac{\sqrt{2nN}\,C}{1-\sqrt{\lambda}}}{2\mu p_{\min}\alpha_{\min} - 2p_{\max}(\max_i L_i)(\alpha_{\max}-\alpha_{\min})}$$

- where $\alpha_{\max} = \max_{i=1,\ldots,N}\{\alpha_i\}$, $\alpha_{\min} = \min_{i=1,\ldots,N}\{\alpha_i\}$,
- $p_{\max} = \max_{i=1,\ldots,N}\{p_i\}$ and $p_{\min} = \min_{i=1,\ldots,N}\{p_i\}$.
- p_i is the probability of agent i updating
- $B = (\max_i L_{-i})NM$
- $M \geq \max_{x_i,z_i\in K_i}\|x_i - z_i\|$ for all i

Expression for error - equal step sizes

$$Err(N,n,\mathcal{G}) = \frac{\alpha p_{\max}}{\mu p_{\min}}\left(2C^2N + B\frac{\sqrt{2nN}\,C}{1-\sqrt{\lambda}}\right)$$

- where $\alpha_{\max} = \max_{i=1,\ldots,N}\{\alpha_i\}$, $\alpha_{\min} = \min_{i=1,\ldots,N}\{\alpha_i\}$,
- $p_{\max} = \max_{i=1,\ldots,N}\{p_i\}$ and $p_{\min} = \min_{i=1,\ldots,N}\{p_i\}$.
- p_i is the probability of agent i updating
- $B = (\max_i L_{-i})NM$
- $M \geq \max_{x_i,z_i\in K_i}\|x_i - z_i\|$ for all i

References I

F. Facchinei and J.-S. Pang.

Finite-Dimensional Variational Inequalities and Complementarity Problems, volume 1 and 2.
Springer-Verlag Inc., New York, 2003.

J . Koshal, A. Nedić, and U.V. Shanbhag.

Distributed multiuser optimization: Algorithms and error analysis.
In *Proceedings of the 48th IEEE Conference on Decision and Control, jointly with the 28th Chinese Control Conference (CDC/CCC)*, pages 4372 –4377, 2009.

J . Koshal, A. Nedić, and U.V. Shanbhag.

Multiuser optimization: Distributed algorithms and error analysis.
SIAM Journal on Optimization, 21(3):1046–1081, 2011.

J. Koshal, A. Nedić, and U.V. Shanbhag.

Distributed algorithms for aggregative games on graphs.
under review, 2012.

J. Koshal, A. Nedić, and U.V. Shanbhag.

A gossip algorithm for aggregative games on graphs.
In *Proceedings of the 51st IEEE Conference on Decision and Control (CDC), Maui, Hawaii, December 9-13*, pages 4840– 4845, 2012.

A. Nedić and A. E. Ozdaglar.

Distributed subgradient methods for multi-agent optimization.
IEEE Trans. Automat. Contr., 54(1):48–61, 2009.

References II

K. Srivastava, A. Nedić, and D. Stipanović.

Distributed Bregman-distance algorithms for min-max optimization.
in the book Agent-Based Optimization I. Czarnowski, P. Jedrzejowicz and J. Kacprzyk (Eds.), Springer Studies in Computational Intelligence (SCI), 2013.

R. Srikant.

Mathematics of Internet Congestion Control.
Birkhauser, 2004.

S. S. Ram, A. Nedić, and V. V. Veeravalli.

A new class of distributed optimization algorithms: application to regression of distributed data.
Optimization Methods and Software, 27(1):71–88, 2012.

S. Shakkottai and R. Srikant.

Network optimization and control.
Foundations and Trends in Networking, 2:271–379, 2007.

J.N. Tsitsiklis.

Problems in decentralized decision making and computation.
PhD thesis, Massachusetts Institute of Technology, 1984.

Further extensions

- More general case can be solved

$$\begin{aligned} \text{minimize} \quad & f_i(x_i, \bar{x})), \quad \bar{x} = \sum_{j=1}^{N} x_j \\ \text{subject to} \quad & x_i \in K_i \subseteq \mathbb{R}^n, \\ & g(x)_1 \leq 0, \ldots, g_m(x) \leq 0 \end{aligned} \qquad \text{(AggGame)}$$

  under suitable assumptions on g_i.

- Push-sum can be used for mixing (instead of convex combinations)