Beyond Consensus and Synchrony in Online Decentralized Optimization using Saddle Point Method

Amrit Singh Bedi [†], Alec Koppel[§], Ketan Rajawat [†]

[†] Dept. of EE, India Institute of Technology Kanpur,

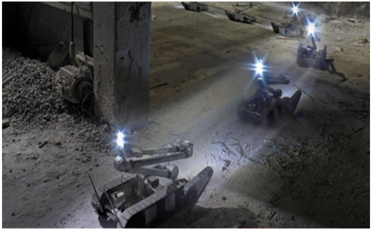
[§] CISD, U.S. Army Research Laboratory, alec.e.koppel.civ@mail.mil

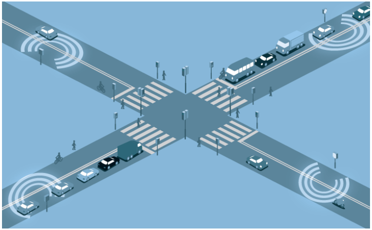
IEEE Asilomar Conference on Signals, Systems, and Computers

Pacific Grove, CA, Oct. 30, 2017

Asynch. Heterogeneous Networked Learning

- Develop framework to learn from **arbitrary streaming data**
  - $\Rightarrow$ in **decentralized**, **heterogeneous**, **asynchronous** systems
  - $\Rightarrow$ e.g., robotic networks or IoT learning systems w/o cloud access



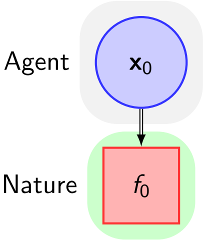


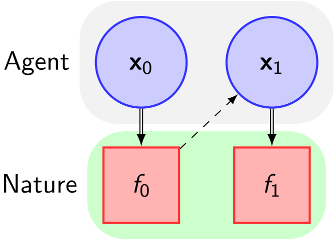
- **Arbitrary streaming data** $\Rightarrow$ addressed via online learning (OL)
- **Decentralized** collaborative systems $\Rightarrow$ distributed opt. algs.
- **Heterogeneity** $\Rightarrow$ no identical data hypothesis (no consensus)
- **Asynchrony** $\Rightarrow$ different node types, avoid synchrony bottleneck

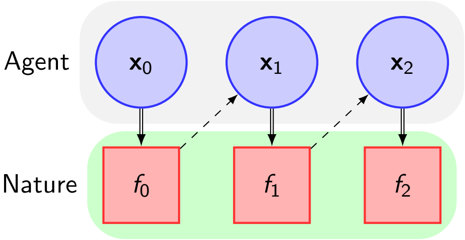
Online Learning

- Online $\Rightarrow$ information received sequentially
  $\Rightarrow$ repeatedly adjust model based on new information
  $\Rightarrow$ no assumptions on data over which learning occurs
- Regret $\Rightarrow$ performance metric for online algorithms
  $\Rightarrow$ Measures no. of mistakes against a fixed optimal offline learner
  $\Rightarrow$ Price learner pays for not being able to see into future

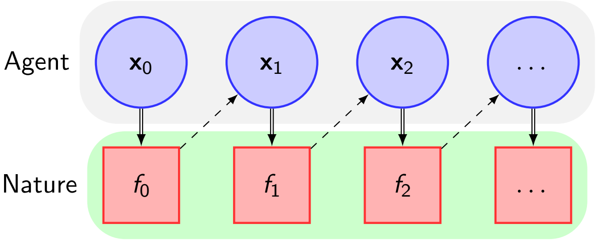
Online Learning



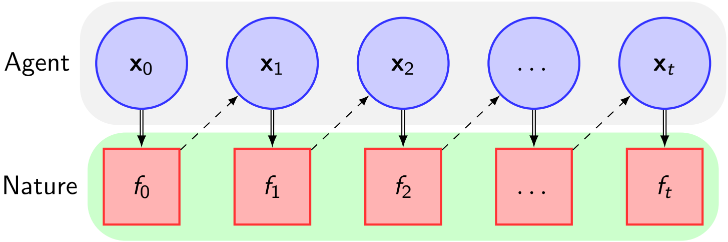

Agent
$\mathbf{x}_0$
$\mathbf{x}_1$
Nature
f_0
f_1


Agent
$\mathbf{x}_0$
$\mathbf{x}_1$
$\mathbf{x}_2$
Nature
f_0
f_1
f_2

Online Learning



Online Learning



- Repeated game over a convex $\mathcal{X} \subset \mathbb{R}^p$
- At the t_{th} round, agent plays $\mathbf{x}_t \in X$, Nature reveals $f_t : \mathcal{X} \to \mathbb{R}$
  - $\Rightarrow$ Suffer arbitrary independent (antagonistic) convex loss $f_t(\mathbf{x}_t)$

Regret

- Regret $\Rightarrow$ performance metric for online learning

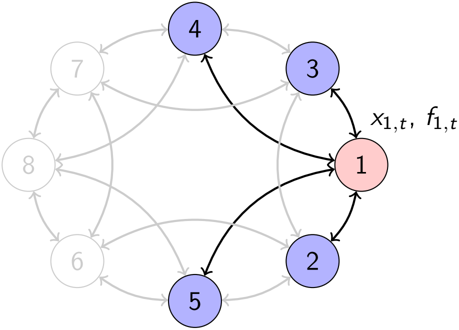
$$\mathbf{Reg}_T := \sum_{t=1}^{T} f_t(\mathbf{x}_t) - \sum_{t=1}^{T} f_t(\mathbf{x}^*)$$

- For fixed T, $x^* = \text{argmin}_{x \in X} \sum_{t=1}^{T} f_t(\mathbf{x})$ is *offline* learner
  - $\Rightarrow$ Price for causal operation
  - $\Rightarrow$ How much we pay for not being clairvoyant

- Goal: $\mathbf{Reg}_T / T \to 0$ as $T \uparrow$, online gradient descent (Zinkevich, '03)

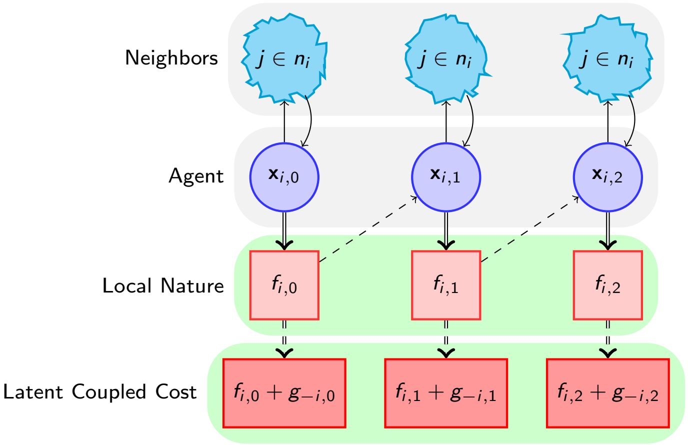
Online Learning in Multi-Agent Networks

- Network $\mathcal{G} = (V, \mathcal{E})$
  - $\Rightarrow |V| = N$, $|\mathcal{E}| = M$
- Neighborhood of agent i
  - $\Rightarrow n_i = \{j : (j, i) \in \mathcal{E}\}$
- N parallel games: agent i, time t
  - $\Rightarrow$ action $\tilde{\mathbf{x}}_{i,t}$ $\Rightarrow$ local loss $f_{i,t}$
  - $\Rightarrow$ minimize local regret?
  - $\Rightarrow \sum_{t=1}^{T} f_{i,t}(\mathbf{x}_{i,t}) - \sum_{t=1}^{T} f_{i,t}(\mathbf{x}_i^*)$



- Hypothesis: smaller local regret using neighborhood info. exchange
  - $\Rightarrow$ If nearby nodes have similar learning goals, decisions correlated

Local and Latent Coupled Cost



- Each node observes local loss and neighbors' information
  - $\Rightarrow$ Better minimize local regret through info. exchange
- Distribution-free: Costs arbitrarily different at distinct nodes

Learning in Heterogeneous Networks

- Learning in networks $\Rightarrow$ regret comparable to $\mathbf{Reg}_T = \mathcal{O}(\sqrt{T})$
- Many previous works on OL in homogeneous synchronized networks
  $\Rightarrow$ primal (DOGD), primal-dual (SP, ADMM), dual averaging, etc.
- Very little that applies to heterogeneous networks (Mahdavi 2012)
- Audience question: others methods for async. heterogeneous OL?
- Contribution: async. heterogeneous OL
  $\Rightarrow$ by using async. online saddle point alg.
  $\Rightarrow$ seek saddle point of online *dual-augmented* Lagrangian
- Proposed method achieves networked online learning goal:
  $\Rightarrow$ $\mathbf{Reg}_T/T \rightarrow 0$ and incentives coordination among similar agents

Coordinated Learning among Multiple Agents

- Global Regret $\Rightarrow$ stacks all local regrets

$$\mathbf{Reg}_T = \sum_{t=1}^{T}\sum_{i=1}^{N} f_{i,t}(\tilde{\mathbf{x}}_{i,t}) - \sum_{t=1}^{T}\sum_{i=1}^{N} f_{i,t}(\mathbf{x}^*).$$

$\Rightarrow$ how well agent i learns in terms of its local cost sequence $f_{i,t}$

- $\mathbf{Reg}_T$ fails to incentivize collaboration

Coordinated Learning among Multiple Agents

- Global Regret $\Rightarrow$ stacks all local regrets

$$\mathbf{Reg}_T = \sum_{t=1}^{T}\sum_{i=1}^{N} f_{i,t}(\tilde{\mathbf{x}}_{i,t}) - \sum_{t=1}^{T}\sum_{i=1}^{N} f_{i,t}(\mathbf{x}^*).$$

$\Rightarrow$ how well agent i learns in terms of its local cost sequence $f_{i,t}$

- $\mathbf{Reg}_T$ fails to incentivize collaboration

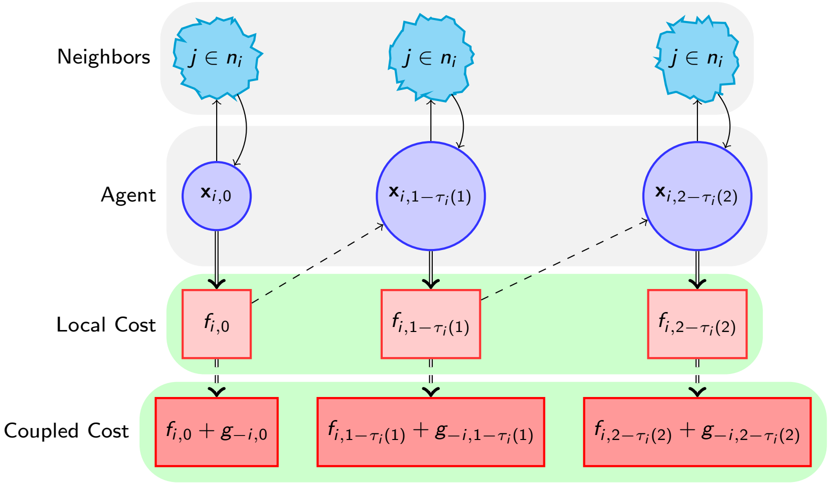
- How should agents address the black-box latent coupling term $g_{-i,t}$?
  - $\Rightarrow$ hypothesis: nearby agents have similar goals
  - $\Rightarrow$ approximate $g_{-i,t}$ by convex proximity functions $h_{ij}(\mathbf{x}_i, \mathbf{x}_j)$
- Define the Network Discrepancy

$$\mathbf{ND}_T := \sum_{(i,\ j)\in\mathcal{E}} \left[\sum_{t=1}^{T} h_{ij}(\mathbf{x}_{i,t}, \mathbf{x}_{j,t}) - \gamma_{ij}\right]_+$$

$\Rightarrow$ Measures agents' coordination with those nearby on network

- $\mathbf{x}^* = \text{argmin}_{\mathbf{x}\in\mathcal{X}} \sum_{t=1}^{T}\sum_{i=1}^{N} f_{i,t}(\mathbf{x})$ s. t. $h_{ij}(\mathbf{x}_{i,t}, \mathbf{x}_{j,t}) \leq \gamma_{ij}, (i,j) \in \mathcal{E}.$

The Asynchronous Setting



- At time $t - \tau_i(t)$, agent plays $\mathbf{x}_{t-\tau_i(t)} \in \mathcal{X}$
  - $\Rightarrow$ receive cost $f_{i,t-\tau_i(t)} : \mathcal{X} \to \mathbb{R}$
  - $\Rightarrow$ $\tau_i(t)$ is random delay from unequal computing power of nodes

Asynchrony and Delayed Regret

- Asynchrony $\Rightarrow$ agent selects $\mathbf{x}_{i,t-\tau_i(t)}$, observes delayed loss $t-\tau_i(t)$
  $\Rightarrow$ faced with learning at random times $t-\tau_i(t)$ with delay $\tau_i(t)$
- Motivates modified regret def: **Network Delayed Regret**

$$\mathbf{Reg}_T := \sum_{t=1}^{T}\sum_{i=1}^{N} f_{i,t-\tau_i(t)}(\mathbf{x}_{i,t-\tau_i(t)}) - \sum_{t=1}^{T}\sum_{i=1}^{N} f_{i,t-\tau_i(t)}(\bar{\mathbf{x}}_T^\star)$$

- Network heterogeneity $\Rightarrow$ coordination via **Network Discrepancy**

$$\mathbf{ND}_T := \sum_{(i,j)\in\mathcal{E}} \left[\sum_{t=1}^{T} h_{ij}(\mathbf{x}_{i,t-\tau_i(t)}, \mathbf{x}_{j,t-\tau_j(t)}) - \gamma_{ij}\right]_+$$

  $\Rightarrow$ Measures agents' coordination with those nearby on network
- $\bar{\mathbf{x}}_T^\star = \text{argmin}_{\mathbf{x}\in\mathcal{X}} \sum_{t=1}^{T}\sum_{i=1}^{N} f_{i,t-\tau_i(t)}(\mathbf{x})$ s.t. $h_{ij}(\mathbf{x}_{i,t}, \mathbf{x}_{j,t}) \le \gamma_{ij}$ for all t

The Online Dual-Augmented Lagrangian

- We develop a primal-dual or saddle point method for this problem.

  $\Rightarrow$ To this end, define **online augmented Lagrangian** at time t as

$$\mathcal{L}_t(\mathbf{x}, \boldsymbol{\lambda}) = \sum_{i=1}^{N} \left[f_{i,t}(\mathbf{x}_i) + \frac{1}{2} \sum_{j \in n_i} \left(\lambda_{ij} \left(h_{ij}(\mathbf{x}_i, \mathbf{x}_j) - \gamma_{ij} \right) - \frac{\delta \epsilon_t}{2} \lambda_{ij}^2 \right) \right].$$

- Saddle pt method: alternate primal/dual gradient descent/ascent

$$\mathbf{x}_{t+1} = \mathcal{P}_{\mathcal{X}} \left[\mathbf{x}_t - \epsilon_t \nabla_{\mathbf{x}} \mathcal{L}_t(\mathbf{x}_t, \boldsymbol{\lambda}_t) \right],$$

$$\boldsymbol{\lambda}_{t+1} = \left[\boldsymbol{\lambda}_t + \epsilon_t \nabla_{\boldsymbol{\lambda}} \mathcal{L}_t(\mathbf{x}_t, \boldsymbol{\lambda}_t) \right]_+ ,$$

- Augmented Lagrangian node-separable $\Rightarrow$ decentralized processing

The Online Dual-Augmented Lagrangian

- We develop a primal-dual or saddle point method for this problem.
  - $\Rightarrow$ To this end, define **online augmented Lagrangian** at time t as

$$\mathcal{L}_t(\mathbf{x}, \boldsymbol{\lambda}) = \sum_{i=1}^{N} \left[f_{i,t}(\mathbf{x}_i) + \frac{1}{2} \sum_{j \in n_i} \left(\lambda_{ij} \left(h_{ij}(\mathbf{x}_i, \mathbf{x}_j) - \gamma_{ij} \right) - \frac{\delta \epsilon_t}{2} \lambda_{ij}^2 \right) \right].$$

- Asynchronous variant of saddle point method: process delayed info.

$$\mathbf{x}_{t+1} = \mathcal{P}_{\mathcal{X}} \left[\mathbf{x}_t - \epsilon_t \nabla_{\mathbf{x}} \mathcal{L}_{t-\tau(t)}(\mathbf{x}_{t-\boldsymbol{\tau}(t)}, \boldsymbol{\lambda}_t) \right],$$

$$\boldsymbol{\lambda}_{t+1} = \left[\boldsymbol{\lambda}_t + \epsilon_t \nabla_{\boldsymbol{\lambda}} \mathcal{L}_{t-\tau(t)}(\mathbf{x}_{t-\boldsymbol{\tau}(t)}, \boldsymbol{\lambda}_t) \right]_+ ,$$

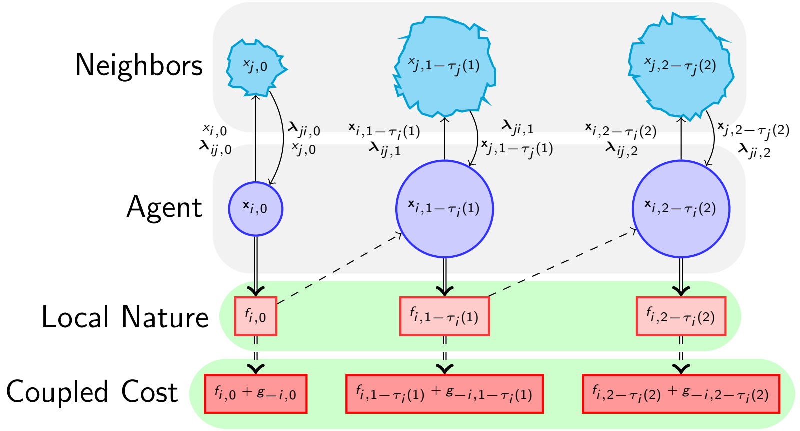
Decentralized implementation

- Send primal $\mathbf{x}_{i,t-\tau_i(t)}$, dual $\boldsymbol{\lambda}_{ij,t}$ to $j \in n_i$, receive $\mathbf{x}_{j,t-\tau_j(t)}, \boldsymbol{\lambda}_{ji,t}$
- Then node i executes the update

$$\mathbf{x}_{i,t+1} = \mathcal{P}_{\mathcal{X}}\Big[\mathbf{x}_{i,t} - \epsilon_t\Big(\nabla_{\mathbf{x}_i} f_{i,t-\tau_i(t)}(\mathbf{x}_{i,t-\tau_i(t)}) + \frac{1}{2}\sum_{j\in n_i}(\lambda_{ij,t}+\lambda_{ji,t})\nabla_{\mathbf{x}_i} h_{ij}(\mathbf{x}_{i,t-\tau_i(t)}, \mathbf{x}_{j,t-\tau_j(t)})\Big)\Big] .$$

- At links of network, update dual var. $\lambda_{ij,t}$ at edge $(i,j) \in \mathcal{E}$

$$\lambda_{ij,t+1} = \Big[(1-\epsilon_t^2\delta)\lambda_{ij,t} + \epsilon_t h_{ij}(\mathbf{x}_{i,t-\tau_i(t)}, \mathbf{x}_{j,t-\tau_j(t)})\Big]_+ .$$



Neighbors
Agent
Local Nature
Coupled Cost
$x_{j,0}$
$x_{j,1-\tau_j(1)}$
$x_{j,2-\tau_j(2)}$
$x_{i,0}$ $\boldsymbol{\lambda}_{ij,0}$
$\boldsymbol{\lambda}_{ji,0}$ $x_{j,0}$
$\mathbf{x}_{i,1-\tau_i(1)}$ $\boldsymbol{\lambda}_{ij,1}$
$\boldsymbol{\lambda}_{ji,1}$ $\mathbf{x}_{j,1-\tau_j(1)}$
$\mathbf{x}_{i,2-\tau_i(2)}$ $\boldsymbol{\lambda}_{ij,2}$
$\mathbf{x}_{j,2-\tau_j(2)}$ $\boldsymbol{\lambda}_{ji,2}$
$\mathbf{x}_{i,0}$
$\mathbf{x}_{i,1-\tau_i(1)}$
$\mathbf{x}_{i,2-\tau_i(2)}$
$f_{i,0}$
$f_{i,1-\tau_i(1)}$
$f_{i,2-\tau_i(2)}$
$f_{i,0} + g_{-i,0}$
$f_{i,1-\tau_i(1)} + g_{-i,1-\tau_i(1)}$
$f_{i,2-\tau_i(2)} + g_{-i,2-\tau_i(2)}$

Technical Assumptions

- **Assumption 1:** Cost $f_{i,t}(\mathbf{x})$ for each node i is Lipschitz for all t

$$\|f_{i,t}(\mathbf{x}) - f_{i,t}(\tilde{\mathbf{x}})\| \leq L_f \|\mathbf{x} - \tilde{\mathbf{x}}\| \; .$$

  Likewise, the constraint function $h_{ij}(\mathbf{x})$ for each edge (i, j) satisfies

$$\|h_{ij}(\mathbf{x}) - h_{ij}(\tilde{\mathbf{x}})\| \leq L_h \|\mathbf{x} - \tilde{\mathbf{x}}\|.$$

- **Assumption 2:** Set $\mathcal{X}$ contains constrained optimizer (Slater's).
- **Assumption 3:** The constraint function is bounded

$$D := \max_i \max_{\mathbf{x} \in \mathcal{X}} h_{ij}(\mathbf{x}, \mathbf{x}_j) \leq L_g R \quad \textit{for all} \quad j \in n_i.$$

- **Assumption 4:** The delay at each node i is bounded $\tau_i(t) \leq \tau$.

Delayed Regret and Constraint Violation Bounds

Theorem

The asynchronous saddle point method $(\mathbf{x}_t, \lambda_t)$ *with step-size* $\epsilon = T^{-1/2}$ *attains sublinear regret in* T*:*

$$\mathbf{Reg}_T \leq \mathcal{O}(\sqrt{\mathbf{T}}).$$

The network discrepancy of the algorithm grows sublinearly in time T *as*

$$\mathbf{ND}_T \leq \mathcal{O}(\mathbf{T^{3/4}}).$$

Delayed Regret and Constraint Violation Bounds

Theorem

The asynchronous saddle point method $(\mathbf{x}_t, \lambda_t)$ *with step-size* $\epsilon = T^{-1/2}$ *attains sublinear regret in* T*:*

$$\mathbf{Reg}_T \leq \mathcal{O}(\sqrt{\mathbf{T}}).$$

The network discrepancy of the algorithm grows sublinearly in time T *as*

$$\mathbf{ND}_T \leq \mathcal{O}(\mathbf{T}^{3/4}).$$

Discussion and Implications

- This result is for regret evaluated at delayed variables
  - ⇒ impossible to obtain sync. regret bounds with async. updates
  - ⇒ *unless the difference of functions across time grows sublinearly*
- Mean convergence of async. stoch. approx. w/o absurd assumptions
  - ⇒ Expectation is computed with respect to *static distribution*

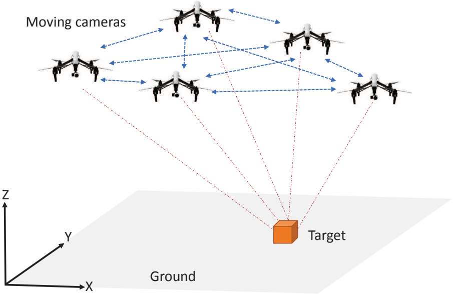
Mobile Source Localization with UAVs

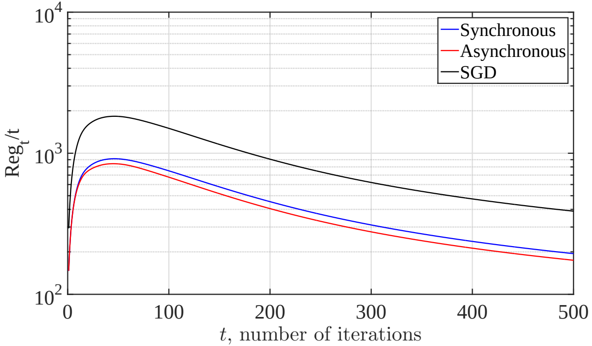
- Localize target via Angle of Arrival (AOA) measurements on UAVs
- $\mathbf{c}_i(t) \in \mathbb{R}^3 \Rightarrow$ camera location; $\mathbf{z} \Rightarrow$ target loc.; captures dir. $\mathbf{s}_i(t)$
- Line perpendicular to target from location $\mathbf{s}_i(t)$ denoted as

$$\mathbf{o}_i(t) = \frac{\mathbf{s}_i(t)^T(\mathbf{z} - \mathbf{c}_i(t))}{\mathbf{s}_i(t)^T\mathbf{s}_i(t)}\mathbf{s}_i(t) + \mathbf{c}_i(t)$$

- Cost: distance to target along perpendicular $f_{i,t}(\mathbf{z}) := \|\mathbf{o}_i(t) - \mathbf{z}\|^2$,

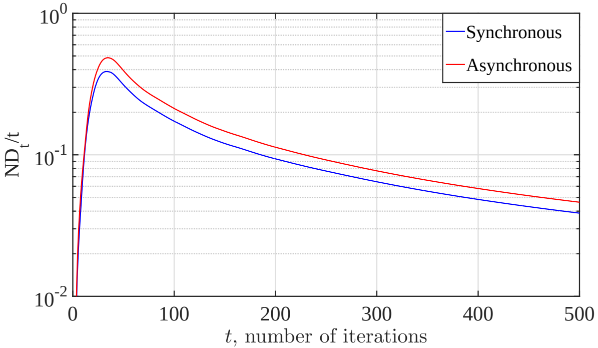
  $\Rightarrow$ randomness $\Rightarrow$ no pose info ; asynchrony $\Rightarrow$ wireless comms.
- Node i's estimate improved by others' $\Rightarrow$ constraint $\|\mathbf{z}^i - \mathbf{z}^j\|^2 \leq \gamma$





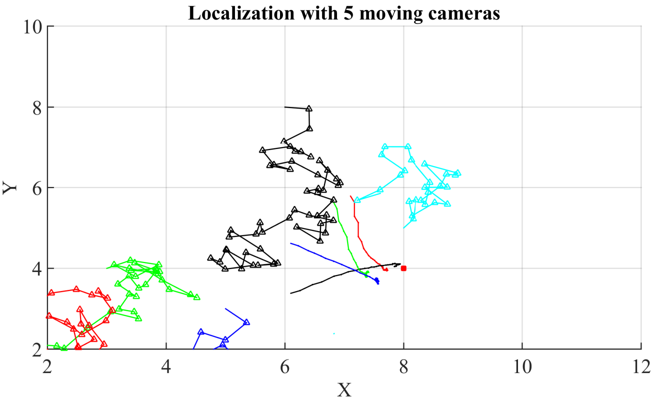
- Comparison with synchronous SP and online gradient descent
  - $\Rightarrow$ **collaboration** translates to **smaller local regret**
  - $\Rightarrow$ distance between agents suffices for unknown exogenous $g_{-i,t}$
- Only small price to pay for asynchronous info. processing

Mobile Source Localization with UAVs



- Network discrepancy (constraint violation) also grows sublinearly
  $\Rightarrow$ synchronous and asynchronous saddle point perform comparably

Mobile Source Localization with UAVs



- Top down view: triangled lines are locations of UAVs
  - ⇒ solid lines are target estimates; red dot is target location
- Agents *learn* the location of the target they are seeking
  - ⇒ via asynchronous online decentralized processing of AOA data

Conclusion

- Addressed on online learning problems in multi-agent networks
  - ⇒ focused on the case where agents' losses are not the same
  - ⇒ how to balance local regret with coordination incentives
  - ⇒ when nodes do not operate on common synchronized clock
- Proposed a new asynchronous online saddle point algorithm
  - ⇒ **sublinear growth** of Delayed regret, Network Discrepancy
- Online asynchronous vision-based localization with moving cameras
  - ⇒ Obtain stable learning in practice, outperform local-only learning
- Future: beyond vector-valued decisions (nonlinear statistical models)
  - ⇒ bridge gap between repeated network games & dist. opt. algs.