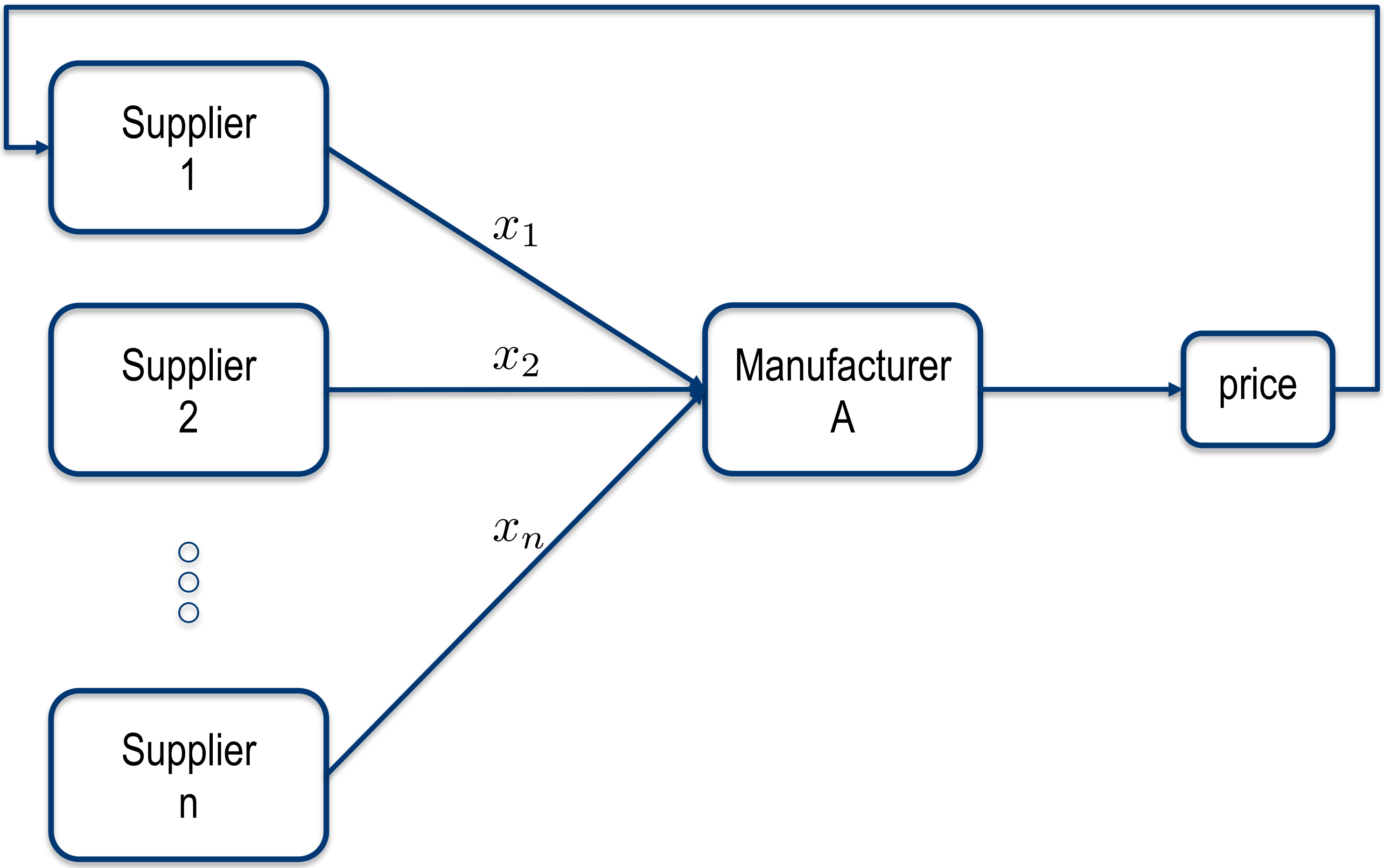
Risk-Averse No-Regret Learning in Online Convex Games

Yi Shen
Mechanical Engineering & Materials Science
Duke University

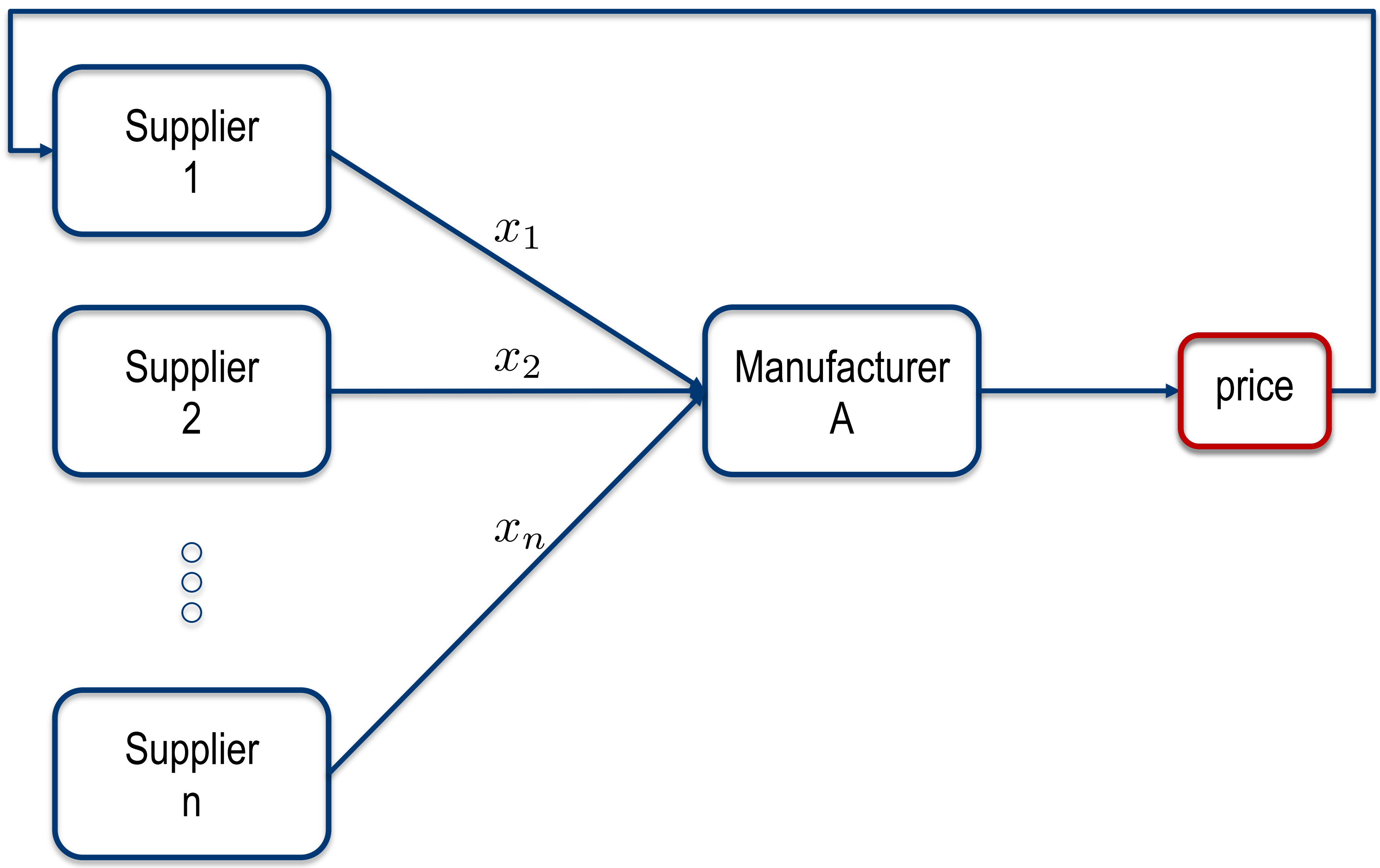
Joint work with: Zifan Wang (KTH), Michael M. Zavlanos (Duke)

ICML 2022

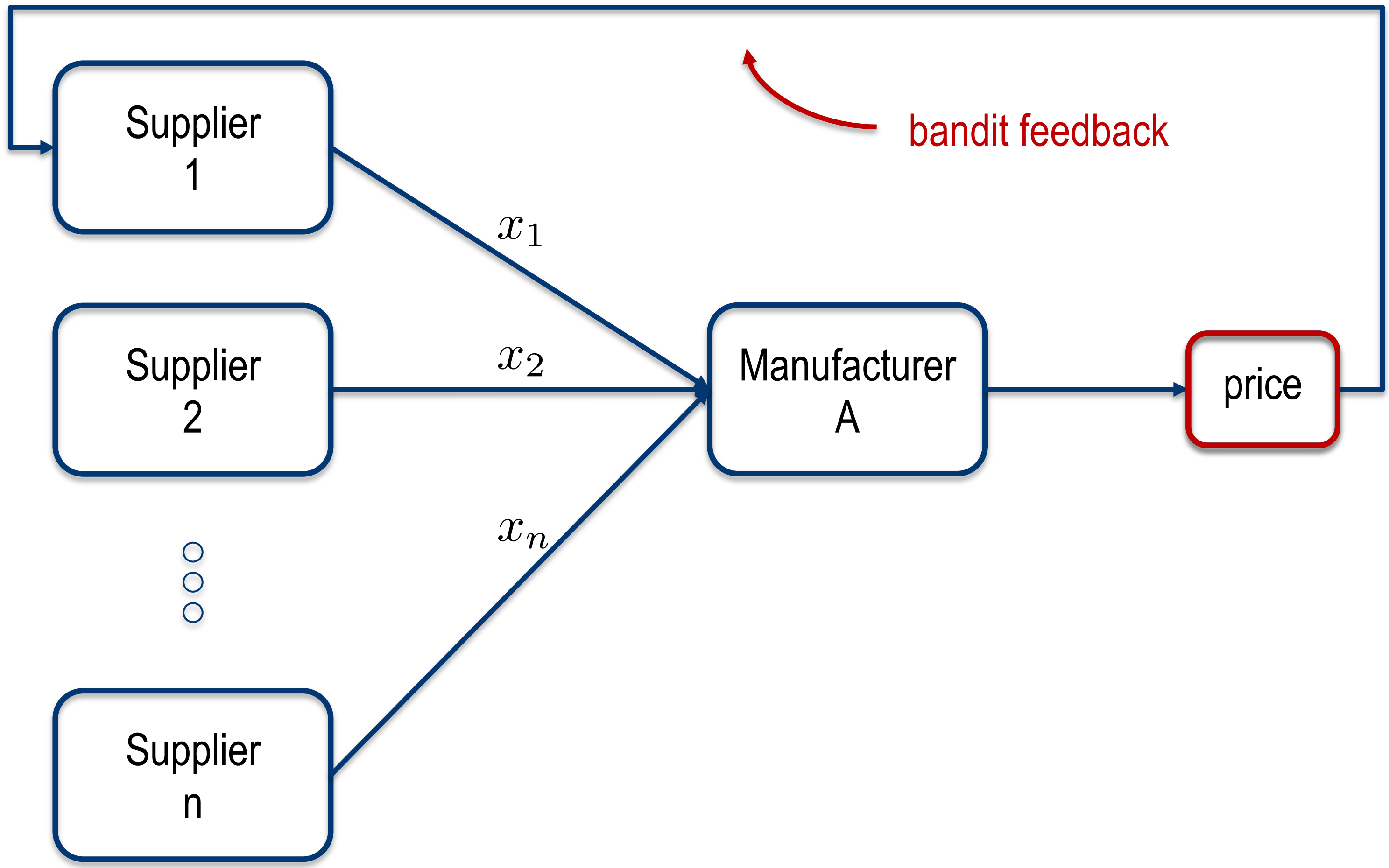
An Online Supply Chain Example



An Online Supply Chain Example



An Online Supply Chain Example



Risk-averse Convex Games

Risk-averse Problem for agent i: $\min_{x_i} C_i(x) := \min_{x_i} \mathrm{CVaR}_{\alpha_i}[f_i(x, \xi_i)]$

where the decision vector $x = [x_1^T, x_2^T, \ldots, x_N^T]^T$ concatenates local decision variables.

We consider Conditional Value at Risk (CVaR) as the risk measure for our problem.

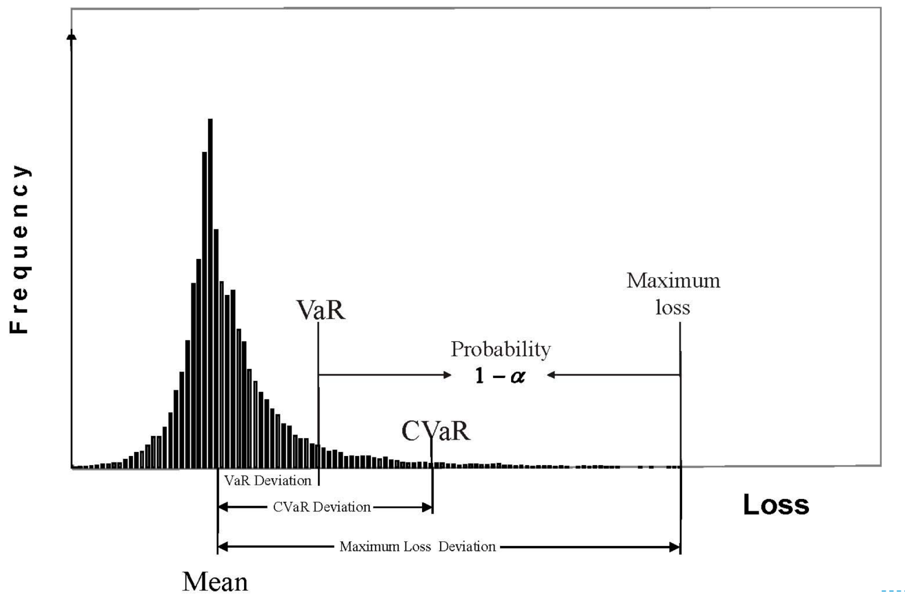


Figure source: Dr. Uryasev's 2000 CVaR tutorial

Risk-averse Convex Games

Risk-averse Problem for agent i: $\min_{x_i} C_i(x) := \min_{x_i} \mathrm{CVaR}_{\alpha_i}[f_i(x, \xi_i)]$

where the decision vector $x = [x_1^T, x_2^T, \ldots, x_N^T]^T$ concatenates local decision variables.

We consider Conditional Value at Risk (CVaR) as the risk measure for our problem.

Figure source: Dr. Uryasev's 2000 CVaR tutorial

Risk-averse Convex Games

Risk-averse Problem for agent i: $\min_{x_i} C_i(x) := \min_{x_i} \mathrm{CVaR}_{\alpha_i}[f_i(x, \xi_i)]$

where the decision vector $x = [x_1^T, x_2^T, \ldots, x_N^T]^T$ concatenates local decision variables.

We consider Conditional Value at Risk (CVaR) as the risk measure for our problem.

Goal: to achieve sublinear regret:

Figure source: Dr. Uryasev's 2000 CVaR tutorial

Risk-averse Convex Games

Risk-averse Problem for agent i: $\min_{x_i} C_i(x) := \min_{x_i} \mathrm{CVaR}_{\alpha_i}[f_i(x, \xi_i)]$

where the decision vector $x = [x_1^T, x_2^T, \ldots, x_N^T]^T$ concatenates local decision variables.

We consider Conditional Value at Risk (CVaR) as the risk measure for our problem.

Goal: to achieve sublinear regret:

$$\mathrm{R}_{C_i}(T) = \sum_{t=1}^{T} C_i(\hat{x}_{i,t}, \hat{x}_{-i,t}) - \min_{\tilde{x}_i \in \mathcal{X}_i} \sum_{t=1}^{T} C_i(\tilde{x}_i, \hat{x}_{-i,t}).$$

KTH VETENSKAP OCH KONST

Figure source: Dr. Uryasev's 2000 CVaR tutorial

Duke UNIVERSITY

Challenges in Risk-averse Convex Games

Risk-Averse Online Convex Game:

1. Each agent takes an action and receives a cost (sample) that depend on all agents' actions
   I. estimating the distribution of the cost function
   II. calculating the CVaR values using the estimated distribution
2. Using Zeroth-Order method to estimate gradient and run gradient-descent
3. To achieve sublinear regret, i.e., R(T)/T →0

Challenges:

1. One sample is not enough to estimate the distribution. → taking more samples
   However, samples are not free. We want to design a sampling strategy that finally converges to 1.

$$n_t = \lceil bU^2(T-t+1)^a \rceil$$

2. Compared to risk-neutral ZO gradient descent, we have extra errors in CVaR estimates. → decompose the whole errors into CVaR error and ZO error.

Challenges in Risk-averse Convex Games

Algorithm 1 Risk-averse learning

Require: Initial value x_0, step size η, parameters a, b, δ, T, risk level α_i, $i = 1, \cdots, N$.

```
for episode t = 1, ..., T do
    Select n_t = ⌈bU^2(T − t + 1)^a⌉
    Each agent samples u_{i,t} ∈ S^{d_i}, i = 1, ..., N
    Each agent play x̂_{i,t} = x_{i,t} + δu_{i,t}, i = 1, ..., N
    for j = 1, ..., n_t do
        Let all agents play x̂_{i,t}
        Obtain J_i(x̂_{i,t}, x̂_{−i,t}, ξ_i^j)
    end for
    for agent i = 1, ..., N do
        Build EDF F̂_{i,t}(y)
        Calculate CVaR estimate: CVaR_{α_i}[F̂_{i,t}]
        Construct gradient estimate
        ĝ_{i,t} = (d_i/δ) CVaR_{α_i}[F̂_{i,t}] u_{i,t}
        Update x: x_{i,t+1} ← P_{X_i^δ}(x_{i,t} − ηĝ_{i,t})
    end for
end for
```

- **for** *episode* $t = 1, \ldots, T$ **do**
    - Select $n_t = \lceil bU^2(T-t+1)^a \rceil$
    - Each agent samples $u_{i,t} \in \mathbb{S}^{d_i}$, $i = 1, \ldots, N$
    - Each agent play $\hat{x}_{i,t} = x_{i,t} + \delta u_{i,t}$, $i = 1, \ldots, N$
    - **for** $j = 1, \ldots, n_t$ **do**
        - Let all agents play $\hat{x}_{i,t}$
        - Obtain $J_i(\hat{x}_{i,t}, \hat{x}_{-i,t}, \xi_i^j)$
    - **end for**
    - **for** agent $i = 1, \ldots, N$ **do**
        - Build EDF $\hat{F}_{i,t}(y)$
        - Calculate CVaR estimate: $\mathrm{CVaR}_{\alpha_i}[\hat{F}_{i,t}]$
        - Construct gradient estimate $\hat{g}_{i,t} = \frac{d_i}{\delta}\mathrm{CVaR}_{\alpha_i}[\hat{F}_{i,t}]u_{i,t}$
        - Update x: $x_{i,t+1} \leftarrow \mathcal{P}_{\mathcal{X}_i^\delta}(x_{i,t} - \eta\hat{g}_{i,t})$
    - **end for**
- **end for**

Challenges in Risk-averse Convex Games

Algorithm 1 Risk-averse learning

Require: Initial value x_0, step size η, parameters a, b, δ, T, risk level α_i, $i = 1, \cdots, N$.

for *episode* $t = 1, \ldots, T$ **do**
    Select $n_t = \lceil bU^2(T - t + 1)^a \rceil$
    Each agent samples $u_{i,t} \in \mathbb{S}^{d_i}$, $i = 1, \ldots, N$
    Each agent play $\hat{x}_{i,t} = x_{i,t} + \delta u_{i,t}$, $i = 1, \ldots, N$
    for $j = 1, \ldots, n_t$ **do**
        Let all agents play $\hat{x}_{i,t}$
        Obtain $J_i(\hat{x}_{i,t}, \hat{x}_{-i,t}, \xi_i^j)$
    end for
    for agent $i = 1, \ldots, N$ **do**
        Build EDF $\hat{F}_{i,t}(y)$
        Calculate CVaR estimate: $\mathrm{CVaR}_{\alpha_i}[\hat{F}_{i,t}]$
        Construct gradient estimate
        $\hat{g}_{i,t} = \frac{d_i}{\delta}\mathrm{CVaR}_{\alpha_i}[\hat{F}_{i,t}]u_{i,t}$
        Update x: $x_{i,t+1} \leftarrow \mathcal{P}_{\mathcal{X}_i^\delta}(x_{i,t} - \eta \hat{g}_{i,t})$
    end for
end for

Challenges in Risk-averse Convex Games

Algorithm 1 Risk-averse learning

Require: Initial value x_0, step size η, parameters a, b, δ, T, risk level α_i, $i = 1, \cdots, N$.

for *episode* $t = 1, \ldots, T$ **do**
  Select $n_t = \lceil bU^2(T - t + 1)^a \rceil$
  Each agent samples $u_{i,t} \in \mathbb{S}^{d_i}$, $i = 1, \ldots, N$
  Each agent play $\hat{x}_{i,t} = x_{i,t} + \delta u_{i,t}$, $i = 1, \ldots, N$
  for $j = 1, \ldots, n_t$ **do**
    Let all agents play $\hat{x}_{i,t}$
    Obtain $J_i(\hat{x}_{i,t}, \hat{x}_{-i,t}, \xi_i^j)$
  end for
  for agent $i = 1, \ldots, N$ **do**
    Build EDF $\hat{F}_{i,t}(y)$
    Calculate CVaR estimate: $\mathrm{CVaR}_{\alpha_i}[\hat{F}_{i,t}]$
    Construct gradient estimate $\hat{g}_{i,t} = \frac{d_i}{\delta}\mathrm{CVaR}_{\alpha_i}[\hat{F}_{i,t}]u_{i,t}$
    Update x: $x_{i,t+1} \leftarrow \mathcal{P}_{\mathcal{X}_i^\delta}(x_{i,t} - \eta \hat{g}_{i,t})$
  end for
end for

Convergence Analysis

Theorem Algorithm 1 achieves regret $\mathrm{R}_{C_i}(T) = \tilde{\mathcal{O}}(T^{1-\frac{a}{4}})$ with probability at least $1-\gamma$.

$$n_t = \lceil bU^2(T-t+1)^a \rceil$$

Two Variants with Better Regret

1. Sample Reuse

$$\tilde{F}_{i,t}(y) = \frac{n_t}{N_t}\hat{F}_{i,t} + \frac{n_{t-1}}{N_t}\hat{F}_{i,t-1}$$

$$\tilde{g}_{i,t} = \frac{d_i}{\delta}\mathrm{CVaR}_{\alpha_i}[\tilde{F}_{i,t}]u_{i,t}$$

2. Residual Feedback

$$\bar{g}_{i,t} = \frac{d_i}{\delta}(\mathrm{CVaR}_{\alpha_i}[\hat{F}_{i,t}] - \mathrm{CVaR}_{\alpha_i}[\hat{F}_{i,t-1}])u_{i,t}$$

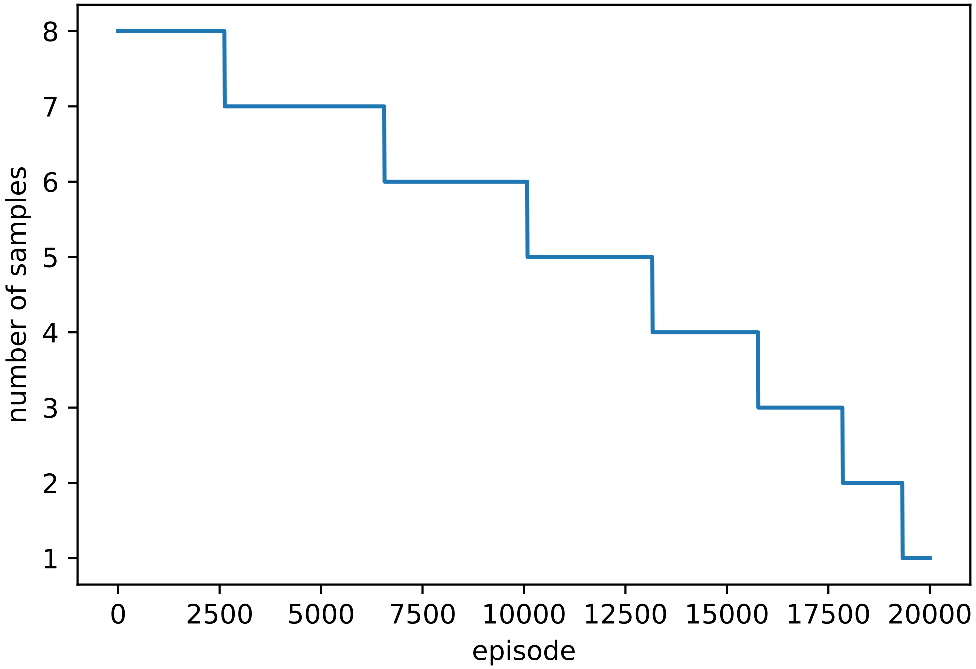
Duke UNIVERSITY

Numerical Results

$$J_i = -(2 - \sum_j x_j)x_i + 0.1x_i + \xi_i x_i + 1$$

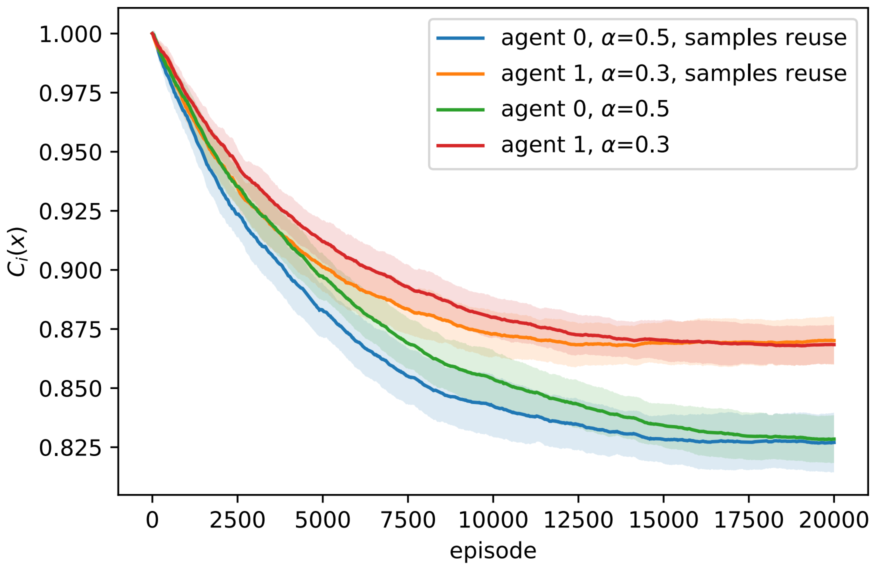
Numerical Results

$$J_i = -(2 - \sum_j x_j)x_i + 0.1x_i + \xi_i x_i + 1$$



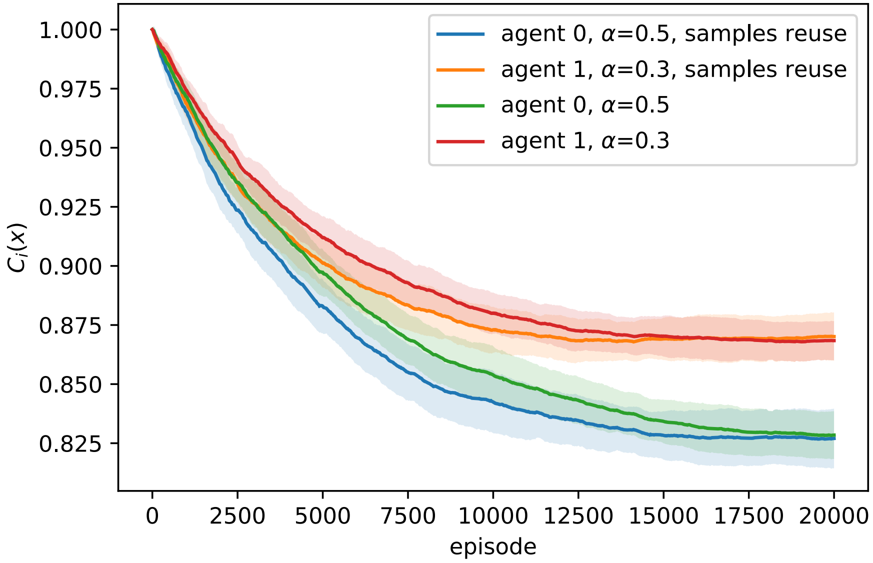
Numerical Results

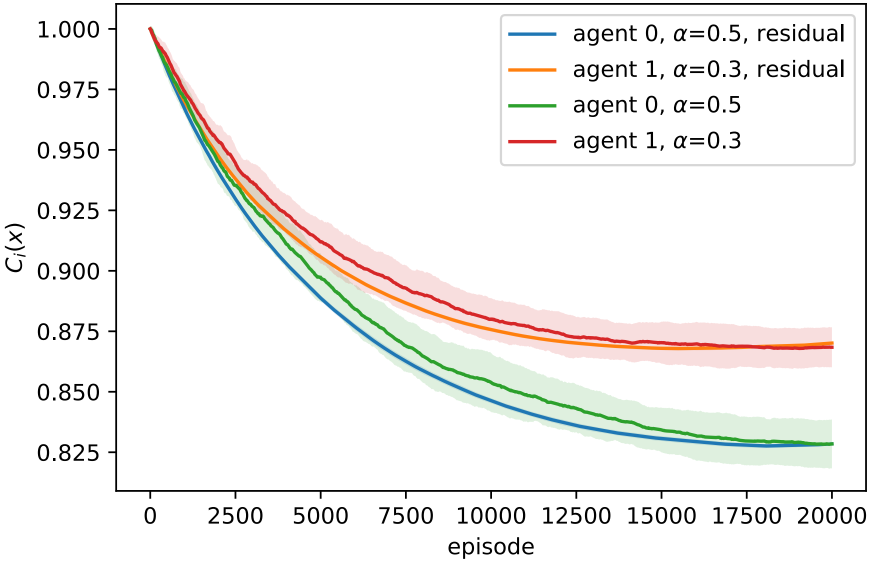
$$J_i = -(2 - \sum_j x_j)x_i + 0.1x_i + \xi_i x_i + 1$$



Numerical Results

$$J_i = -(2 - \sum_j x_j)x_i + 0.1x_i + \xi_i x_i + 1$$





Summary

We proposed a zeroth-order method and two improved variants for online convex games with risk-averse agents and validated them by a Cournot game example.

Support