Online learning in routing games

Walid Krichene Benjamin Drighes Alex Bayen

October 18, 2013

Outline

Outline

Routing game

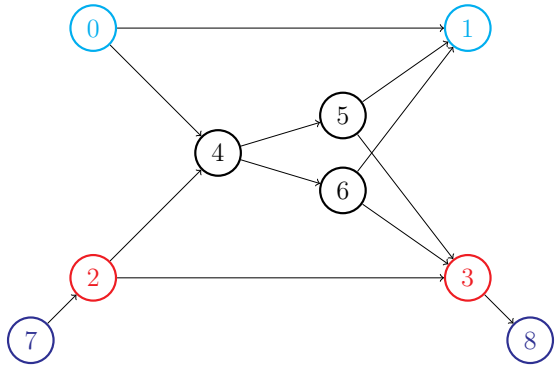


Figure : Example network

- Graph (V, E)
- source-sink pairs, (s_k, t_k): total flow F_k (cars/s, packets/s etc.), paths $\mathcal{P}_k$
- feasible flow: f such that for all k, $\sum_{p \in \mathcal{P}_k} f_p = F_k$

Routing game

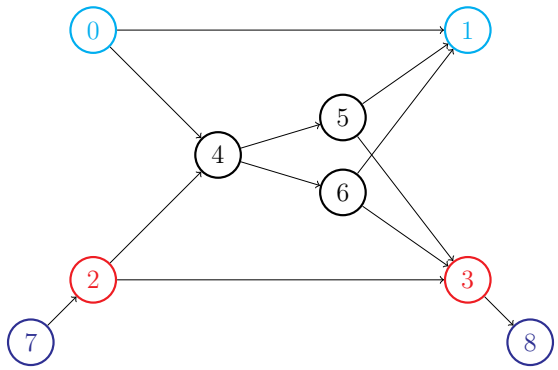


Figure : Example network

- Graph (V, E)
- source-sink pairs, (s_k, t_k): total flow F_k (cars/s, packets/s etc.), paths $\mathcal{P}_k$
- feasible flow: f such that for all k, $\sum_{p \in \mathcal{P}_k} f_p = F_k$
- Latency on edge e: $\ell_e : f_e \mapsto \ell_e(f_e)$, convex increasing

Routing game

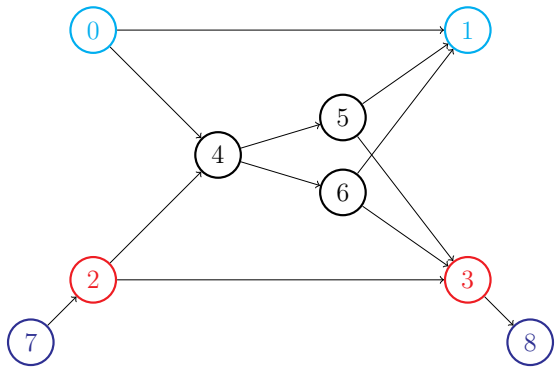


Figure : Example network

- Graph (V, E)
- source-sink pairs, (s_k, t_k): total flow F_k (cars/s, packets/s etc.), paths $\mathcal{P}_k$
- feasible flow: f such that for all k, $\sum_{p \in \mathcal{P}_k} f_p = F_k$
- Latency on edge e: $\ell_e : f_e \mapsto \ell_e(f_e)$, convex increasing
- Players choose a path $p \in \mathcal{P}_k$ selfishly,
  want to minimize personal latency $\ell_p(f) = \sum_{e \in p} \ell_e(f_e)$

player = infinitesimal amount of flow.
f = combined decision of all players.

More precisely

- Measurable set of players $(S_k, \mathcal{S}_k, m_k)$, atomless
- $F_k = m_k(S_k)$

More precisely

- Measurable set of players $(S_k, \mathcal{S}_k, m_k)$, atomless
- $F_k = m_k(S_k)$
- Path choice function

$$C_k : S_k \to \mathcal{P}_k$$
$$x \mapsto C_k(x)$$

$$f_p^k = m_k(C_k^{-1}(\{p\}))$$

Selfish routing game

Nash equilibrium

f is a Nash equilibrium if for all k, for all $p \in \mathcal{P}_k$ with positive flow, $\ell_p(f)$ is minimal on $\mathcal{P}_k$
($\ell_p(f) \leq \ell_{p'}(f)$ for all $p' \in \mathcal{P}_k$).

Selfish routing game

Nash equilibrium

f is a Nash equilibrium if for all k, for all $p \in \mathcal{P}_k$ with positive flow, $\ell_p(f)$ is minimal on $\mathcal{P}_k$
($\ell_p(f) \leq \ell_{p'}(f)$ for all $p' \in \mathcal{P}_k$).

Equivalent to Nash equilibrium for almost every player.

Selfish routing game

Nash equilibrium

f is a Nash equilibrium if for all k, for all $p \in \mathcal{P}_k$ with positive flow, $\ell_p(f)$ is minimal on $\mathcal{P}_k$
($\ell_p(f) \leq \ell_{p'}(f)$ for all $p' \in \mathcal{P}_k$).

Equivalent to Nash equilibrium for almost every player.

- How to compute Nash equilibria?

Selfish routing game

Nash equilibrium

f is a Nash equilibrium if for all k, for all $p \in \mathcal{P}_k$ with positive flow, $\ell_p(f)$ is minimal on $\mathcal{P}_k$
($\ell_p(f) \leq \ell_{p'}(f)$ for all $p' \in \mathcal{P}_k$).

Equivalent to Nash equilibrium for almost every player.

- How to compute Nash equilibria?
  Convex formulation

Rosenthal potential function

f is a Nash equilibrium iff it minimizes a potential function

$$\begin{aligned} \text{minimize}_{f \geq 0,\, \phi} \quad & \sum_e \int_0^{\phi_e} \ell_e(u)du \\ \text{subject to} \quad & \forall e,\ \sum_{p \ni e} f_p = \phi_e \qquad \sum_p f_p = F \end{aligned}$$

Motivation for a learning model

- How do players find a Nash equilibrium?

Motivation for a learning model

- How do players find a Nash equilibrium?
  Ideally: distributed, and has minimal information requirements.
  - observed value of the latency on the player's path
  - observed value of latency on commodity path
  - all edge flows
  - all latency functions

Motivation for a learning model

- How do players find a Nash equilibrium?
  Ideally: distributed, and has minimal information requirements.
  - observed value of the latency on the player's path
  - observed value of latency on commodity path
  - all edge flows
  - all latency functions
- Need a model of dynamics to apply control

Outline

The hedge algorithm

Fix one player.
Player maintains a probability distribution $\mu(t)$ over paths, draw path according to $\mu(t)$.

Multiplicative Weights

- distribution $\mu(t)$ over paths p on day t
- update the distribution according to observed loss

$$\mu_p(t+1) \propto \mu_p(t) e^{-\gamma \ell_p(t)}$$

Regret Bound

- Assume losses are in $[0, \rho]$.
- Expected loss is $\ell_{alg}(t) = \sum_p \mu_p(t)\ell_p(t)$

$$R(T) = \sum_{t=1}^{T} \ell_{alg}(t) - \min_p \sum_{t=1}^{T} \ell_p(t)$$

Regret Bound

- Assume losses are in $[0, \rho]$.
- Expected loss is $\ell_{alg}(t) = \sum_p \mu_p(t)\ell_p(t)$

$$R(T) = \sum_{t=1}^{T} \ell_{alg}(t) - \min_p \sum_{t=1}^{T} \ell_p(t)$$

Regret of the expected loss

$$\frac{R(T)}{T} \leq \frac{\rho \ln |\mathcal{P}|}{T\gamma} + \rho\gamma$$

No-regret routing at the population level

- Assume all players apply the same learning algorithm.

No-regret routing at the population level

- Assume all players apply the same learning algorithm.
- For any player x, $C(x,t)$ is a random variable with $P(C(x,t)=p)=\mu_p(t)$

No-regret routing at the population level

- Assume all players apply the same learning algorithm.
- For any player x, $C(x,t)$ is a random variable with $P(C(x,t)=p)=\mu_p(t)$
- The flow $f_p(t)=m(C(\cdot,t)^{-1}(\{p\}))$ is a random variable

No-regret routing at the population level

- Assume all players apply the same learning algorithm.
- For any player x, $C(x,t)$ is a random variable with $P(C(x,t)=p)=\mu_p(t)$
- The flow $f_p(t)=m(C(\cdot,t)^{-1}(\{p\}))$ is a random variable

$$\mathrm{E}\, f_p(t)=\mu_p(t)$$
$$\mathrm{var}\, f_p(t)=0 \qquad (\textit{Fubini's theorem})$$

Outline

Cesàro convergence

Cesàro convergence of no-regret routing

If an update rule satisfies the regret bound, then for all $\epsilon > 0$, for γ small enough, no-regret learning with rate γ converges in the sense

$$\lim_{T\to\infty} \frac{1}{T}\sum_{t=1}^{T} \mu(t) \in \mathcal{N}_\epsilon$$

$\mathcal{N}_\epsilon = \{\mu : V(\mu) < V_{\mathcal{N}} + \epsilon\}$: ϵ-approximate Nash equilibrium.

Recall the regret bound

$$\frac{R(T)}{T} \leq \frac{\rho \ln |\mathcal{P}|}{T\gamma} + \rho\gamma$$

Outline

Hedge as a regularized greedy algorithm

Can show the hedge update rule is solution to

Greedy algorithm, regularized by the K-L divergence

$$\begin{aligned} \text{minimize}_{\mu \geq 0} \quad & \sum_p \mu_p \ell_p(t-1) + \frac{1}{\gamma} D(\mu || \mu(t-1)) \\ \text{subject to} \quad & \sum_p \mu_p = 1 \end{aligned}$$

Hedge as a regularized greedy algorithm

Can show the hedge update rule is solution to

Greedy algorithm, regularized by the K-L divergence

$$\begin{aligned} \text{minimize}_{\mu \geq 0} \quad & \sum_p \mu_p \ell_p(t-1) + \frac{1}{\gamma} D(\mu || \mu(t-1)) \\ \text{subject to} \quad & \sum_p \mu_p = 1 \end{aligned}$$

- $D(\mu||\mu(t-1)) = \sum_p \mu_p \ln \frac{\mu_p}{\mu_p(t-1)}$
- $\ell_p(t-1)$ loss on the previous day.

Hedge as a regularized greedy algorithm

Can show the hedge update rule is solution to

Greedy algorithm, regularized by the K-L divergence

$$\begin{aligned} \text{minimize}_{\mu \geq 0} \quad & \sum_p \mu_p \ell_p(t-1) + \frac{1}{\gamma} D(\mu || \mu(t-1)) \\ \text{subject to} \quad & \sum_p \mu_p = 1 \end{aligned}$$

- $D(\mu||\mu(t-1)) = \sum_p \mu_p \ln \frac{\mu_p}{\mu_p(t-1)}$
- $\ell_p(t-1)$ loss on the previous day.

Limit cases:

- $\gamma \to \infty$, greedy algorithm
- $\gamma \to 0+$, static distribution

Simulations

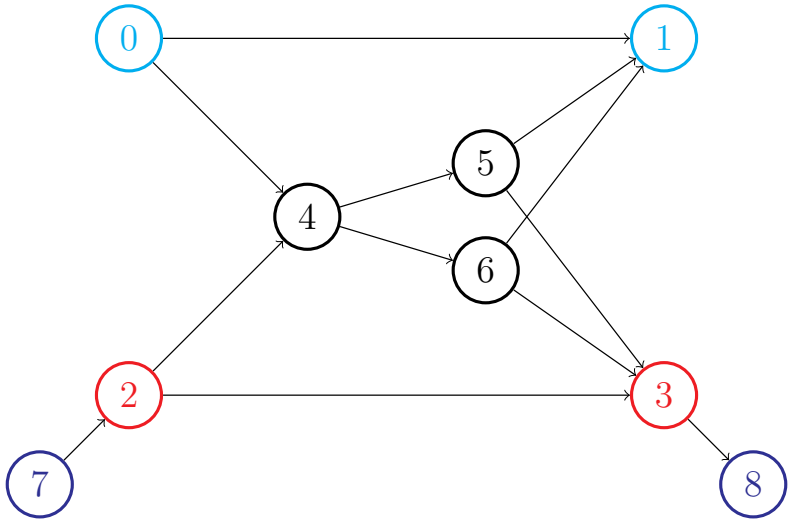
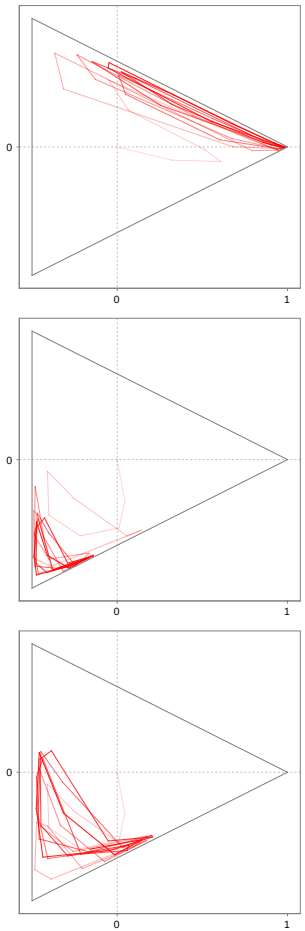
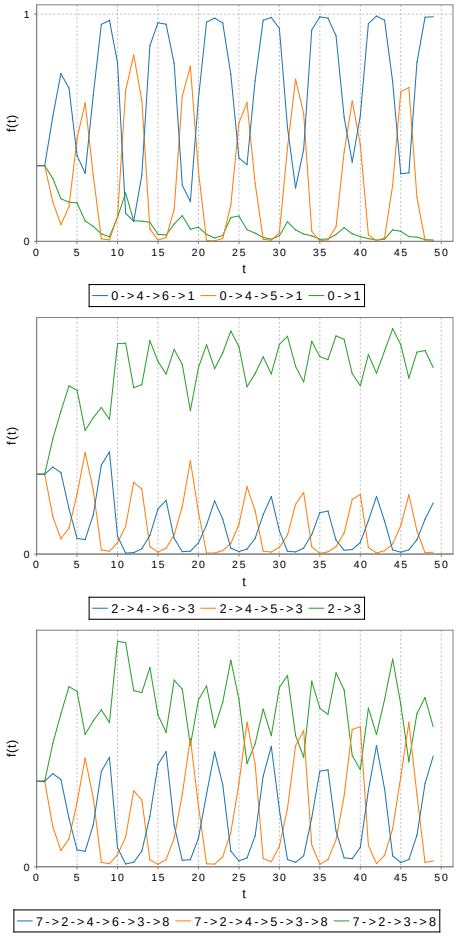


Figure : Example network

Simulations



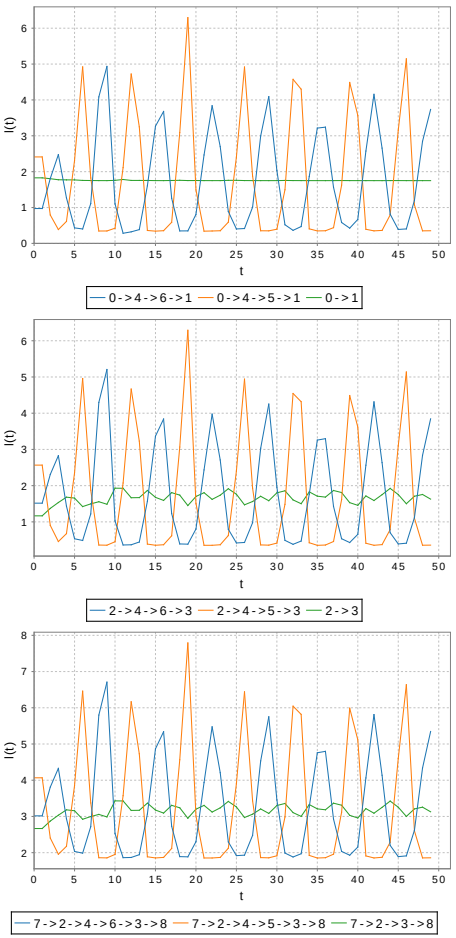
(a) Trajectories $(\mu^k(\tau))_\tau$.



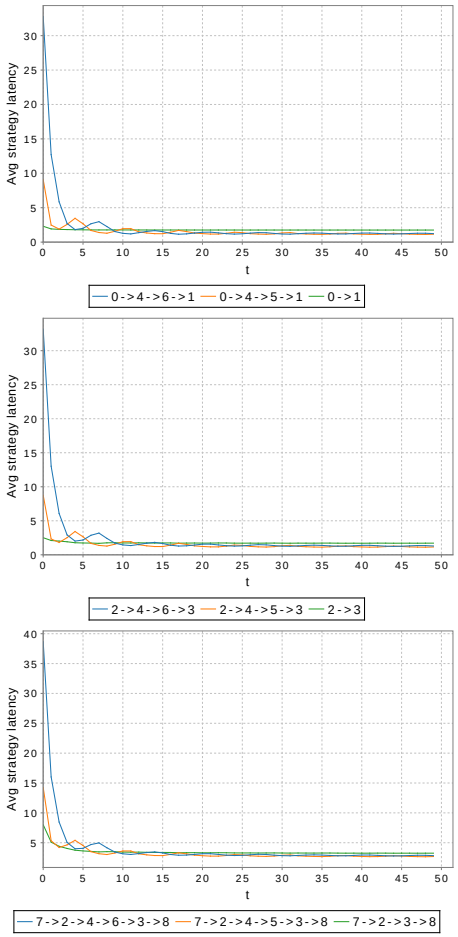
(b) Path flows $f_p^k, p \in \mathcal{P}_k$

Figure : Constant learning rate $\gamma = 0.7$. The trajectories do not converge.

Simulations



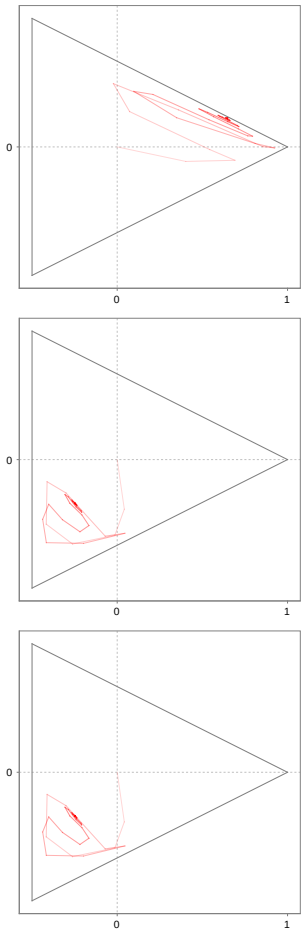
(a) Path latencies $\ell_p(\mu(\tau))$



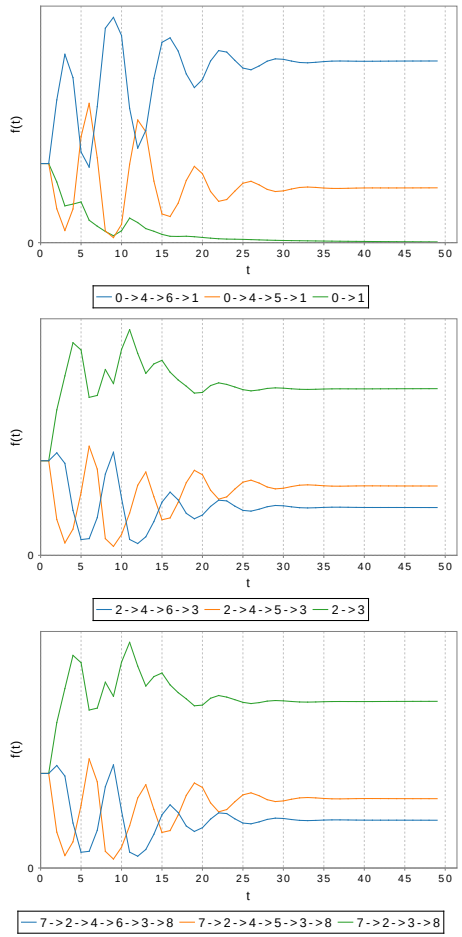
(b) Path latencies for the Cesàro means, $\ell_p(\frac{1}{\tau}\sum_{u\leq\tau}\mu(u))$

Figure : Path latencies

Simulations



(a) Trajectories $(\mu^k(\tau))_\tau$.



(b) Path flows $f_p^k, p \in \mathcal{P}_k$

Figure : harmonic sequence of learning rates $\gamma(\tau) = \frac{1}{1+\tau/10}$

Simulations

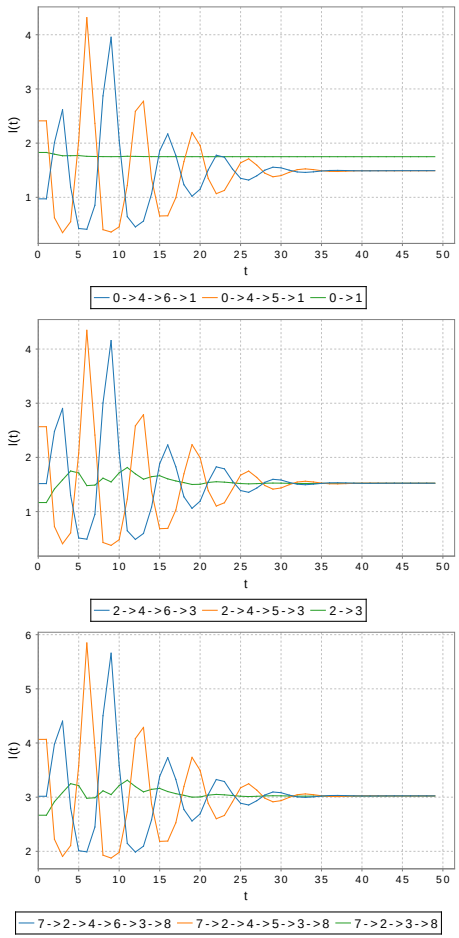


Figure : Path latencies $\ell_{\boldsymbol{p}}(\mu(\tau))$

Discounting the losses

Put weights on time $(\gamma(t))_{t\in\mathbb{N}}$: players care more about present than future.

Discounting the losses

Put weights on time $(\gamma(t))_{t\in\mathbb{N}}$: players care more about present than future.
The sequence of discounting factors γ is universal.

Discounting the losses

Put weights on time $(\gamma(t))_{t\in\mathbb{N}}$: players care more about present than future.
The sequence of discounting factors γ is universal.

Assumptions

- $\gamma(t) > 0$
- $\gamma(t) \downarrow 0$
- $\sum_t \gamma(t) = \infty$

Discounting the losses

Put weights on time $(\gamma(t))_{t\in\mathbb{N}}$: players care more about present than future.
The sequence of discounting factors γ is universal.

Assumptions

- $\gamma(t) > 0$
- $\gamma(t) \downarrow 0$
- $\sum_t \gamma(t) = \infty$

Definition (Discounted regret)

$$R(T) = \sum_{t=0}^{T} \gamma(t) \sum_{p} \mu_p(t)\ell_p(\mu(t)) - \min_{p} \sum_{t=0}^{T} \gamma(t)\ell_p(\mu(t))$$

No-regret if

$$\frac{1}{\sum_{t=0}^{T}\gamma(t)} R(T) \xrightarrow{T\to\infty} 0$$

Discounted hedge algorithm

Regret bound

$$\frac{R(T)}{\sum_{t\leq T}\gamma(t)} \leq \frac{\rho \log|\mathcal{P}|}{\sum_{t\leq T}\gamma(t)} + \frac{\rho}{8}\frac{\sum_{t\leq T}\gamma(t)^2}{\sum_{t\leq T}\gamma(t)}$$

Discounted hedge algorithm

Regret bound

$$\frac{R(T)}{\sum_{t\leq T}\gamma(t)} \leq \frac{\rho \log|\mathcal{P}|}{\sum_{t\leq T}\gamma(t)} + \frac{\rho}{8}\frac{\sum_{t\leq T}\gamma(t)^2}{\sum_{t\leq T}\gamma(t)}$$

Consequence: if γ is square-summable, discounted Hedge achieves no-regret.

Outline

Convergence of a dense subsequence

Theorem

Under a discounted no-regret routing algorithm, $(\mu(t))_t$ converges to Nash equilibria on a subset of days of density one.

- subsequence $(\mu_{t_k})_k$ converges
- $\lim_{T\to\infty} \frac{\sum_{t_k \leq T} \gamma(t_k)}{\sum_{t \leq T} \gamma(t)} = 1$

Convergence of a dense subsequence

Theorem

Under a discounted no-regret routing algorithm, $(\mu(t))_t$ converges to Nash equilibria on a subset of days of density one.

- subsequence $(\mu_{t_k})_k$ converges
- $\lim_{T\to\infty} \frac{\sum_{t_k \le T} \gamma(t_k)}{\sum_{t \le T} \gamma(t)} = 1$

Proof.

- Convexity:

$$V(\mu(t)) - V(\mu) \le \nabla V(\mu(t))^T (\mu(t) - \mu) = \sum_{k=1}^{K} F_k \sum_{p \in \mathcal{P}_k} \ell_p(\mu(t))(\mu_p(t) - \mu_p)$$

Convergence of a dense subsequence

Theorem

Under a discounted no-regret routing algorithm, $(\mu(t))_t$ converges to Nash equilibria on a subset of days of density one.

- subsequence $(\mu_{t_k})_k$ converges
- $\lim_{T\to\infty} \frac{\sum_{t_k \le T} \gamma(t_k)}{\sum_{t \le T} \gamma(t)} = 1$

Proof.

- Convexity:

$$V(\mu(t)) - V(\mu) \le \nabla V(\mu(t))^T (\mu(t) - \mu) = \sum_{k=1}^{K} F_k \sum_{p \in \mathcal{P}_k} \ell_p(\mu(t))(\mu_p(t) - \mu_p)$$

- Discounted no-regret:

$$\sum_{t \le T} \gamma(t)(V(\mu(t)) - V(\mu)) \le \sum_{k=1}^{K} F_k R_k(T)$$

Convergence of a dense subsequence

Theorem

Under a discounted no-regret routing algorithm, $(\mu(t))_t$ converges to Nash equilibria on a subset of days of density one.

- subsequence $(\mu_{t_k})_k$ converges
- $\lim_{T\to\infty} \frac{\sum_{t_k \leq T} \gamma(t_k)}{\sum_{t \leq T} \gamma(t)} = 1$

Proof.

- Convexity:

$$V(\mu(t)) - V(\mu) \leq \nabla V(\mu(t))^T (\mu(t) - \mu) = \sum_{k=1}^{K} F_k \sum_{p \in \mathcal{P}_k} \ell_p(\mu(t))(\mu_p(t) - \mu_p)$$

- Discounted no-regret:

$$\sum_{t \leq T} \gamma(t)(V(\mu(t)) - V(\mu)) \leq \sum_{k=1}^{K} F_k R_k(T)$$

- Absolute Cesaro convergence implies convergence on a subset of density one.

Outline

Replicator dynamics

Imagine an underlying continuous time. Updates happen at $\gamma_1, \gamma_1 + \gamma_2, \ldots$

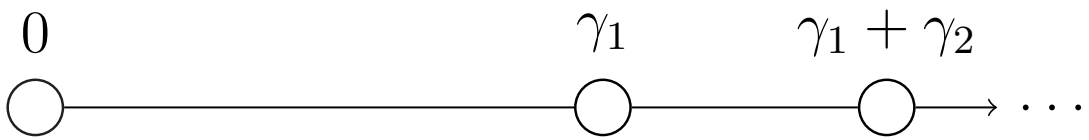


Figure : Underlying continuous time

Replicator dynamics

Imagine an underlying continuous time. Updates happen at $\gamma_1, \gamma_1 + \gamma_2, \dots$

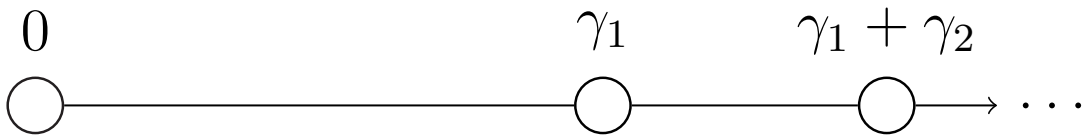


Figure : Underlying continuous time

In the update equation $\mu_p(t+1) \propto \mu_p(t) e^{-\gamma_t \ell_p(t)}$, take $\gamma_t \to 0$
We obtain the autonomous ODE:

Replicator equation

$$\begin{cases} \mu(0) \in \mathring{\Delta} \\ \forall p \in \mathcal{P}_k, \frac{d\mu_p}{dt} = \mu_p(\bar{\ell}^k(\mu) - \ell_p(\mu))/\rho \end{cases} \tag{1}$$

Also in evolutionary game theory.

Replicator dynamics

Replicator equation

$$\begin{cases} \mu(0) \in \mathring{\Delta} \\ \forall p \in \mathcal{P}_k, \frac{d\mu_p}{dt} = \mu_p(\bar{\ell}^k(\mu) - \ell_p(\mu))/\rho \end{cases} \tag{2}$$

Also in evolutionary game theory.

Replicator dynamics

Replicator equation

$$\begin{cases} \mu(0) \in \mathring{\Delta} \\ \forall p \in \mathcal{P}_k, \frac{d\mu_p}{dt} = \mu_p(\bar{\ell}^k(\mu) - \ell_p(\mu))/\rho \end{cases} \tag{2}$$

Also in evolutionary game theory.

Restricted Nash equilibria are stationary points, partitioned into:

- Nash equilibria
- non-Nash equilibria

Replicator dynamics

Replicator equation

$$\begin{cases} \mu(0) \in \mathring{\Delta} \\ \forall p \in \mathcal{P}_k, \frac{d\mu_p}{dt} = \mu_p(\bar{\ell}^k(\mu) - \ell_p(\mu))/\rho \end{cases} \tag{2}$$

Also in evolutionary game theory.

Restricted Nash equilibria are stationary points, partitioned into:

- Nash equilibria
- non-Nash equilibria unstable

Replicator dynamics

Replicator equation

$$\begin{cases} \mu(0) \in \mathring{\Delta} \\ \forall p \in \mathcal{P}_k, \frac{d\mu_p}{dt} = \mu_p(\bar{\ell}^k(\mu) - \ell_p(\mu))/\rho \end{cases} \tag{2}$$

Also in evolutionary game theory.

Restricted Nash equilibria are stationary points, partitioned into:

- Nash equilibria
- non-Nash equilibria unstable

Theorem

Every solution of the ODE (1) converges to the set of restricted Nash equilibria of the routing game.

Replicator dynamics

Replicator equation

$$\begin{cases} \mu(0) \in \mathring{\Delta} \\ \forall p \in \mathcal{P}_k, \frac{d\mu_p}{dt} = \mu_p(\bar{\ell}^k(\mu) - \ell_p(\mu))/\rho \end{cases} \tag{2}$$

Also in evolutionary game theory.

Restricted Nash equilibria are stationary points, partitioned into:

- Nash equilibria
- non-Nash equilibria unstable

Theorem

Every solution of the ODE (1) converges to the set of restricted Nash equilibria of the routing game.

Idea: V is a Lyapunov function for $\mathcal{RN}$.

Outline

REP algorithms

Discretization of the replicator dynamics (REP)

$$\begin{cases} \mu(0) \in \mathring{\Delta} \\ \mu_{\boldsymbol{p}}(t+1) - \mu_{\boldsymbol{p}}(t) = \gamma(t)\mu_{\boldsymbol{p}}(t)\left(\frac{\bar{\ell}^{\boldsymbol{k}}(\mu(t)) - \ell_{\boldsymbol{p}}(\mu(t))}{\rho}\right) + \gamma(t)U_{\boldsymbol{p}}(t+1) \end{cases}$$

$(U(t))_{t\geq 1}$ deterministic or stochastic perturbations that satisfy for all $T > 0$,

$$\lim_{\tau\to\infty} \max\left\{ \left\| \sum_{t=\tau}^{\tau'-1} \gamma(t)U(t+1) \right\| : \tau' = \{\tau+1, \cdots, \sup\{t \geq 0 : t \geq T_t + T\}\} \right\} = 0$$

REP algorithms

In particular for $U = 0$, we obtain a new update rule

$$\begin{cases} \mu(0) \in \mathring{\Delta} \\ \mu_{\boldsymbol{p}}(t+1) - \mu_{\boldsymbol{p}}(t) = \gamma(t)\mu_{\boldsymbol{p}}(t)\left(\frac{\bar{\ell}^{\boldsymbol{k}}(\mu(t)) - \ell_{\boldsymbol{p}}(\mu(t))}{\rho}\right) \end{cases}$$

REP algorithms

In particular for $U = 0$, we obtain a new update rule

$$\begin{cases} \mu(0) \in \mathring{\Delta} \\ \mu_{\boldsymbol{p}}(t+1) - \mu_{\boldsymbol{p}}(t) = \gamma(t)\mu_{\boldsymbol{p}}(t)\left(\frac{\bar{\ell}^{\boldsymbol{k}}(\mu(t)) - \ell_{\boldsymbol{p}}(\mu(t))}{\rho}\right) \end{cases}$$

- Discounted no-regret algorithm.

REP algorithms

In particular for $U = 0$, we obtain a new update rule

$$\begin{cases} \mu(0) \in \mathring{\Delta} \\ \mu_{\boldsymbol{p}}(t+1) - \mu_{\boldsymbol{p}}(t) = \gamma(t)\mu_{\boldsymbol{p}}(t)\left(\frac{\bar{\ell}^{\boldsymbol{k}}(\mu(t)) - \ell_{\boldsymbol{p}}(\mu(t))}{\rho}\right) \end{cases}$$

- Discounted no-regret algorithm.
- Solution to regularized optimization:

$$\mu(t) \in \arg\min_{\mu\in\Delta} \sup_{p} \mu_p \frac{\ell_p(\mu(t-1))}{\rho} + \frac{1}{\gamma(t)} R(\mu \| \mu(t-1))$$

where $R(x||y) = \frac{1}{2}\sum_p y_p \left(\frac{x_{\boldsymbol{p}}}{y_{\boldsymbol{p}}} - 1\right)^2$

REP algorithms

In particular for $U = 0$, we obtain a new update rule

$$\begin{cases} \mu(0) \in \mathring{\Delta} \\ \mu_p(t+1) - \mu_p(t) = \gamma(t)\mu_p(t)\left(\frac{\bar{\ell}^{\boldsymbol{k}}(\mu(t)) - \ell_{\boldsymbol{p}}(\mu(t))}{\rho}\right) \end{cases}$$

- Discounted no-regret algorithm.
- Solution to regularized optimization:

$$\mu(t) \in \arg\min_{\mu\in\Delta} \sup_{p} \mu_p \frac{\ell_p(\mu(t-1))}{\rho} + \frac{1}{\gamma(t)} R(\mu\|\mu(t-1))$$

where $R(x||y) = \frac{1}{2}\sum_p y_p \left(\frac{x_{\boldsymbol{p}}}{y_{\boldsymbol{p}}} - 1\right)^2$

The discounted hedge algorithm is also a REP algorithm.

Convergence to Nash equilibria

Theorem

Under any discounted no-regret REP algorithm, the sequence $\mu(t)$ converges to the set of Nash equilibria.

Convergence to Nash equilibria

Theorem

Under any discounted no-regret REP algorithm, the sequence $\mu(t)$ converges to the set of Nash equilibria.

Proof:

- Let X be the affine interpolation of the sequence $\mu(t)$. X is an Asymptotic Pseudo Trajectory for the ODE.

Convergence to Nash equilibria

Theorem

Under any discounted no-regret REP algorithm, the sequence $\mu(t)$ converges to the set of Nash equilibria.

Proof:

- Let X be the affine interpolation of the sequence $\mu(t)$. X is an Asymptotic Pseudo Trajectory for the ODE.
- Let $L(X)$ be the set limit points of X.

Convergence to Nash equilibria

Theorem

Under any discounted no-regret REP algorithm, the sequence $\mu(t)$ converges to the set of Nash equilibria.

Proof:

- Let X be the affine interpolation of the sequence $\mu(t)$. X is an Asymptotic Pseudo Trajectory for the ODE.
- Let $L(X)$ be the set limit points of X.
- V is constant over $L(X)$.

Convergence to Nash equilibria

Theorem

Under any discounted no-regret REP algorithm, the sequence $\mu(t)$ converges to the set of Nash equilibria.

Proof:

- Let X be the affine interpolation of the sequence $\mu(t)$. X is an Asymptotic Pseudo Trajectory for the ODE.
- Let $L(X)$ be the set limit points of X.
- V is constant over $L(X)$.
- Use Cesàro convergence to conclude that constant value is minimum of V.

Stochastic optimization interpretation

- Setting: minimize V, know an unbiased estimate of the gradient at a given μ, $G(\mu)$

$$\mathrm{E}\, G(\mu) = \nabla V(\mu)$$

- stochastic gradient descent $\mu(t+1) = \pi_\Delta(\mu(t) - \gamma_t G(\mu(t)))$ Euclidean projection on Δ.

Stochastic optimization interpretation

- Setting: minimize V, know an unbiased estimate of the gradient at a given μ, $G(\mu)$

$$\mathrm{E}\, G(\mu) = \nabla V(\mu)$$

- stochastic gradient descent $\mu(t+1) = \pi_\Delta(\mu(t) - \gamma_t G(\mu(t)))$ Euclidean projection on Δ.

Idea: generalized projection

- Use a prox-function (Bregman divergence)

$$D(\nu, \mu) = \omega(\mu) - \omega(\nu) - \langle \nabla\omega(\nu), \mu - \nu \rangle$$

- Define prox-mapping

$$\Pi_\nu(g) = \arg\min_{\mu \in \Delta} \langle g, \mu - \nu \rangle + D(\nu, \mu)$$

Stochastic optimization interpretation

Algorithm: Stochastic Mirror Descent

$$\mu(t+1) = \Pi_{\mu(t)}(\gamma_t G(\mu(t)))$$

where $\mathsf{E}\, G(\mu(t)) = \nabla V(\mu(t))$

Stochastic optimization interpretation

Algorithm: Stochastic Mirror Descent

$$\mu(t+1) = \Pi_{\mu(t)}(\gamma_t G(\mu(t)))$$

where $\mathsf{E}\, G(\mu(t)) = \nabla V(\mu(t))$

- Example 1: $D(\nu, \mu) = \frac{1}{2}\|\nu - \mu\|_2^2$.

Stochastic optimization interpretation

Algorithm: Stochastic Mirror Descent

$$\mu(t+1) = \Pi_{\mu(t)}(\gamma_t G(\mu(t)))$$

where $\mathrm{E}\, G(\mu(t)) = \nabla V(\mu(t))$

- Example 1: $D(\nu, \mu) = \frac{1}{2}\|\nu - \mu\|_2^2$.
  This is Euclidean projection

Stochastic optimization interpretation

Algorithm: Stochastic Mirror Descent

$$\mu(t+1) = \Pi_{\mu(t)}(\gamma_t G(\mu(t)))$$

where $\mathrm{E}\, G(\mu(t)) = \nabla V(\mu(t))$

- Example 1: $D(\nu,\mu) = \frac{1}{2}\|\nu-\mu\|_2^2$.
  This is Euclidean projection
- Example 2: $D(\nu,\mu) = D_{KL}(\nu||\mu)$ (equiv. $\omega(\mu) = \sum_p \mu_p \log \mu_p$).

Stochastic optimization interpretation

Algorithm: Stochastic Mirror Descent

$$\mu(t+1) = \Pi_{\mu(t)}(\gamma_t G(\mu(t)))$$

where $\mathrm{E}\, G(\mu(t)) = \nabla V(\mu(t))$

- Example 1: $D(\nu, \mu) = \frac{1}{2}\|\nu - \mu\|_2^2$.
  This is Euclidean projection
- Example 2: $D(\nu, \mu) = D_{KL}(\nu||\mu)$ (equiv. $\omega(\mu) = \sum_p \mu_p \log \mu_p$).
  This is Hedge algorithm on the gradient of V

Stochastic optimization interpretation

Algorithm: Stochastic Mirror Descent

$$\mu(t+1) = \Pi_{\mu(t)}(\gamma_t G(\mu(t)))$$

where $\mathsf{E}\, G(\mu(t)) = \nabla V(\mu(t))$

- Example 1: $D(\nu,\mu) = \frac{1}{2}\|\nu-\mu\|_2^2$.
  This is Euclidean projection
- Example 2: $D(\nu,\mu) = D_{KL}(\nu||\mu)$ (equiv. $\omega(\mu) = \sum_p \mu_p \log \mu_p$).
  This is Hedge algorithm on the gradient of V

Hedge algorithm is stochastic mirror descent on V (Rosenthal potential function)

Open problems and extensions

- Conjecture: if $\mu(0) \in \mathring{\Delta}$, the replicator dynamics converge to $\mathcal{N}$

Open problems and extensions

- Conjecture: if $\mu(0) \in \mathring{\Delta}$, the replicator dynamics converge to $\mathcal{N}$
- Relax assumption that all players "learn in the same way" (universal discount sequence $\gamma(t)$, universal initial distribution $\mu(0)$).

Open problems and extensions

- Conjecture: if $\mu(0) \in \mathring{\Delta}$, the replicator dynamics converge to $\mathcal{N}$
- Relax assumption that all players "learn in the same way" (universal discount sequence $\gamma(t)$, universal initial distribution $\mu(0)$).
- Apply control to the system: e.g. tolling

Thank you.