Learning and Efficiency in Games
with Dynamic Population

Éva Tardos
Cornell

Joint work with Thodoris Lykouris and Vasilis Syrgkanis

Large population games: traffic routing

- Traffic subject to congestion delays
- cars and packets follow shortest path
- Congestion game =cost (delay) depends only on congestion on edges

Example 2: advertising auctions

advertising auctions

- Advertisers leave and join the system
- Changes in system setup
- Advertiser values change

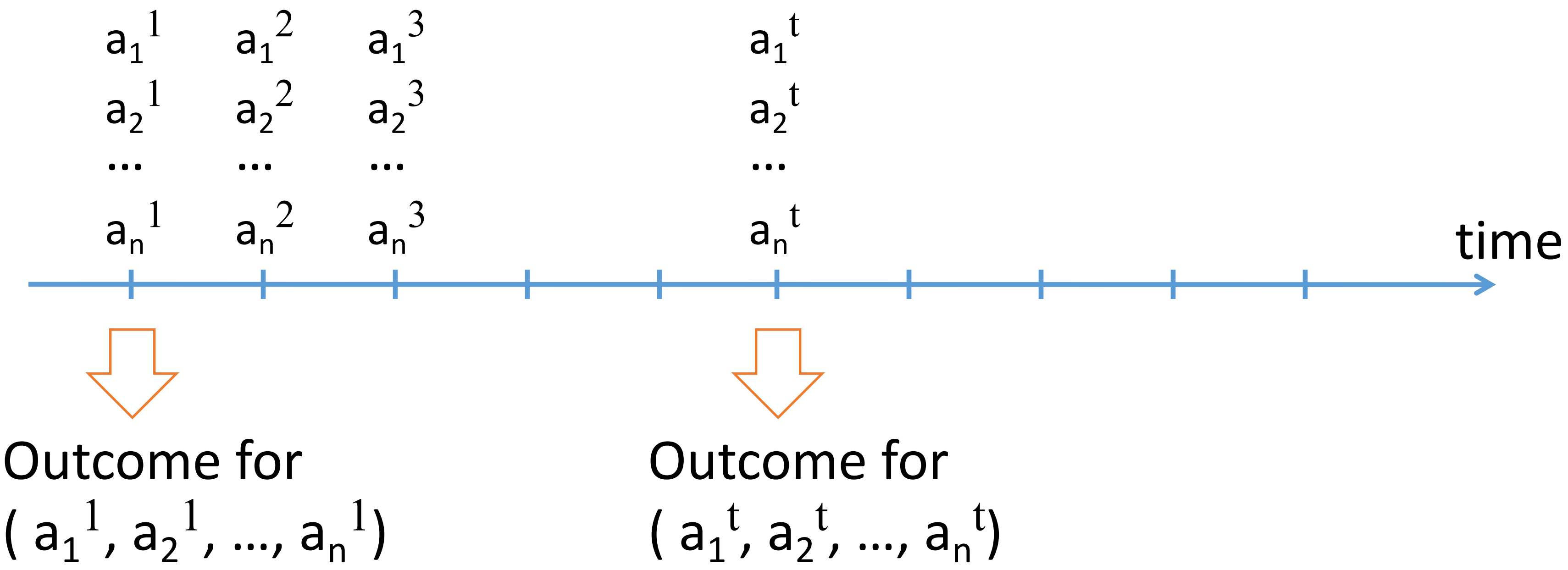
Questions + Motivation

- Repeated play: How do players behave?
  - Nash equilibrium? of one-shot/repeated game?
  - Today: Learning

- With players (or player objectives) changing over time

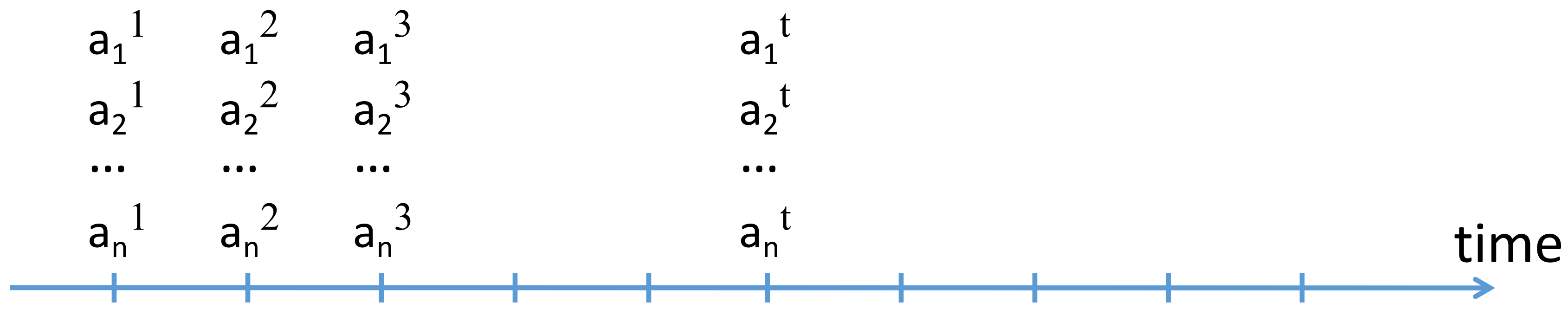
- Efficiency loss due to selfish behavior (Price of Anarchy)

Repeated games



- Assume same game each period
- Player's value/cost additive over periods

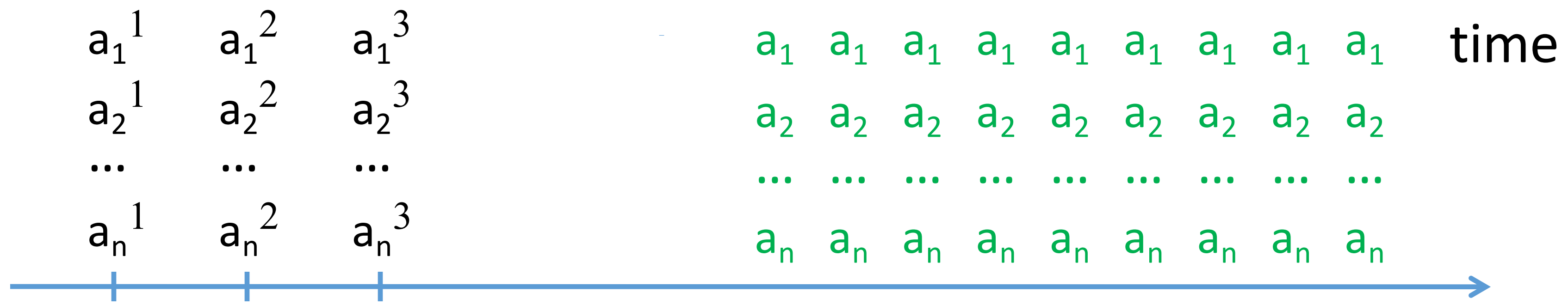
Learning outcome



Maybe **here** they don't know how to play, who are the other players, …

By here they have a better idea…

Nash equilibrium

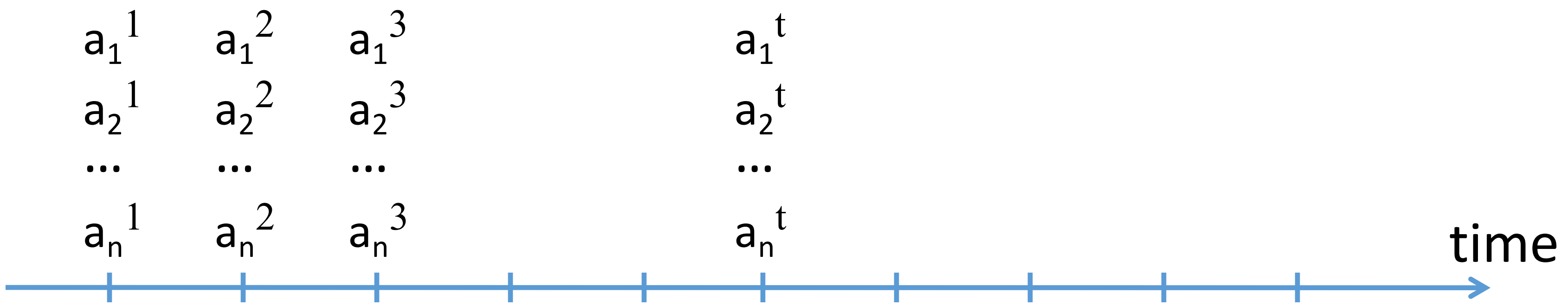


Nash equilibrium (of stage game): Stable actions a with no regret for any alternate strategy x:

$$cost_i(x, a_{-i}) \geq cost_i(a)$$

No regret

No-regret without stability: learning



For any fixed action x :

$$\sum_t cost_i(a^t) \leq \sum_t cost_i(x, a_{-i}^t)$$

No-regret

Regret: $R_i(x,T) = \sum_t cost_i(a^t) - \sum_t cost_i(x, a_{-i}^t) \leq o(T)$

Many simple rules ensure $R_i(x,T)$ approx. $\sim\sqrt{T \log d}$ for all x

MWU (Hedge), Regret Matching, etc.

d = # of strategies

No-regret without stability: learning

a_1^1 a_1^2 a_1^3 a_1^t

a_2^1 a_2^2 a_2^3 a_2^t

… … … …

a_n^1 a_n^2 a_n^3 a_n^t

time

For any fixed action x (with d options) :

$$\sum_t cost_i(a^t) \leq \sum_t cost_i(x, a^t_{-i})$$

Approx. no-regret

Regret: $R_i(x,T) = \sum_t cost_i(a^t) - (1+\epsilon) \sum_t cost_i(x, a^t_{-i}) \leq o(T)$

Many simple rules ensure $R_i(x,T)$ approx. ~$O(\log d/\epsilon)$ for all x

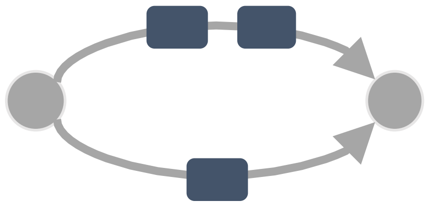
MWU (Hedge), Regret Matching, etc.

Foster, Li, Lykouris, Sridharan, T'16

Main Question

- Efficiency loss due to selfish behavior of players (Price of Anarchy)
- In repeated game settings
- With players (or player objectives) changing over time

Examples



internet routing

- Traffic changes over time

advertising auctions

- Advertisers leave and join the system
- Advertiser values change

Result: routing, limit for very small users

Theorem (Roughgarden-T'02):

In any network with continuous, non-decreasing cost functions and small users

cost of Nash with rates r_i for all i	$\leq$	cost of opt with rates $2r_i$ for all i

Nash equilibrium: stable solution where no player had incentive to deviate.

$$\text{Price of Anarchy} = \frac{\text{cost of worst Nash equilibrium}}{\text{“socially optimum” cost}}$$

Quality of Learning outcomes: Price of Total Anarchy

Bounds average welfare assuming no-regret learners

$$\text{Price of Total Anarchy} = \lim_{T\to\infty} \frac{\frac{1}{T}\sum_{t=1}^{T} cost(a^t)}{\text{“socially optimum” cost}}$$

[Blum, Hajiaghayi, Ligett, Roth, 2008]

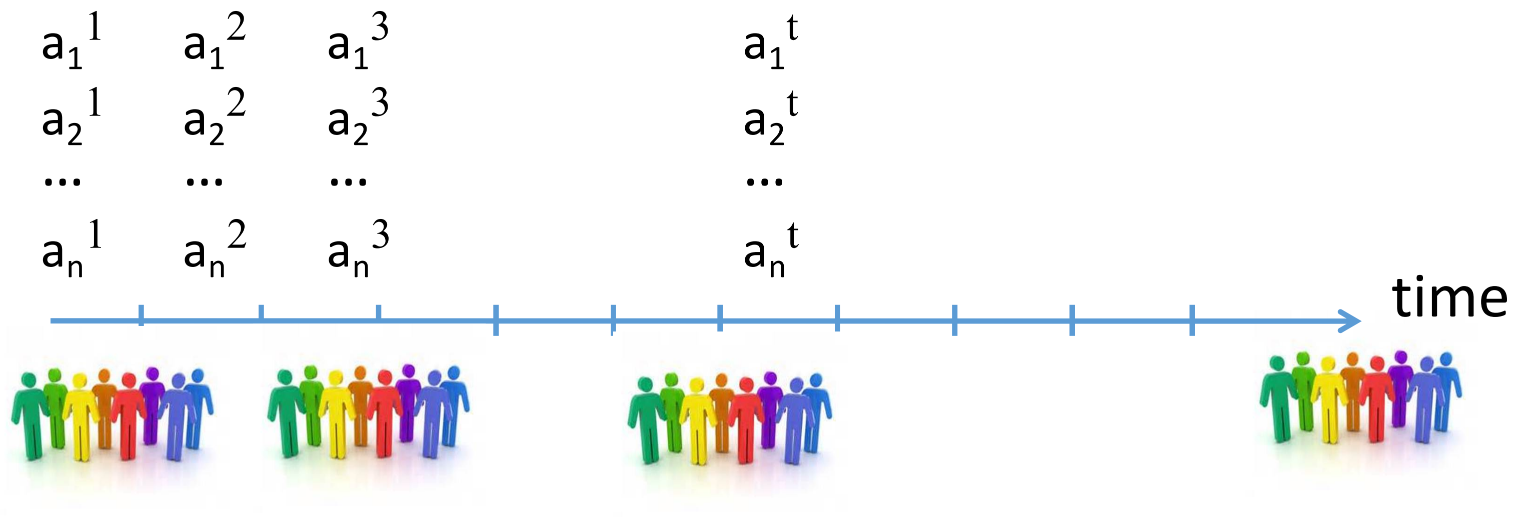
[Roughgarden 2009] 2016 Kalai prize

Most results on Price Anarchy extend to learning outcomes

Result 2: routing with learning players

Theorem (Blum, Even-Dar, Ligett'06; Roughgarden'09):

Price of anarchy bounds developed for Nash equilibria extend to no-regret learning outcomes



Assumes a stable set of participants

Today: Dynamic Population

Classical model:

- Game is repeated identically and nothing changes

Dynamic population model:

At each step t each player i

is replaced with an arbitrary new player with probability p

In a population of N players, each step, Np players replaced in expectation

Learning players can adapt….

Goal:

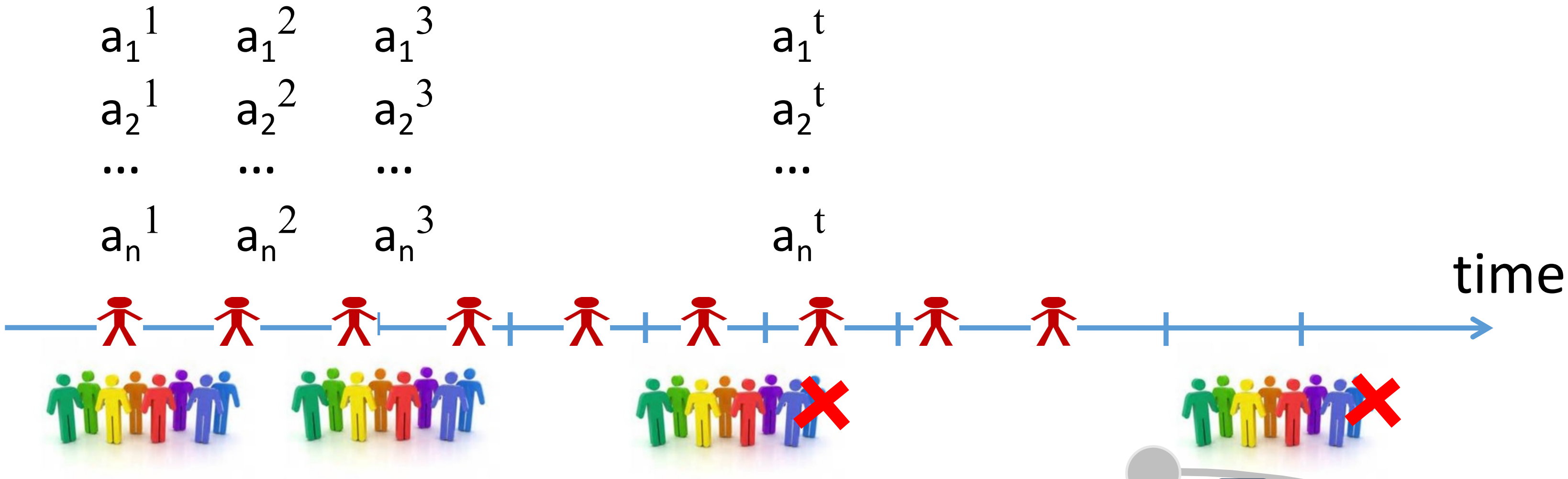
Bound average welfare assuming **adaptive** /shifting expert no-regret learners

$$PoA = \lim_{T\to\infty} \frac{\sum_{t=1}^{T} cost(a^t, v^t)}{\sum_{t=1}^{T} Opt(v^t)}$$

where v^t is the vector of player types at time t

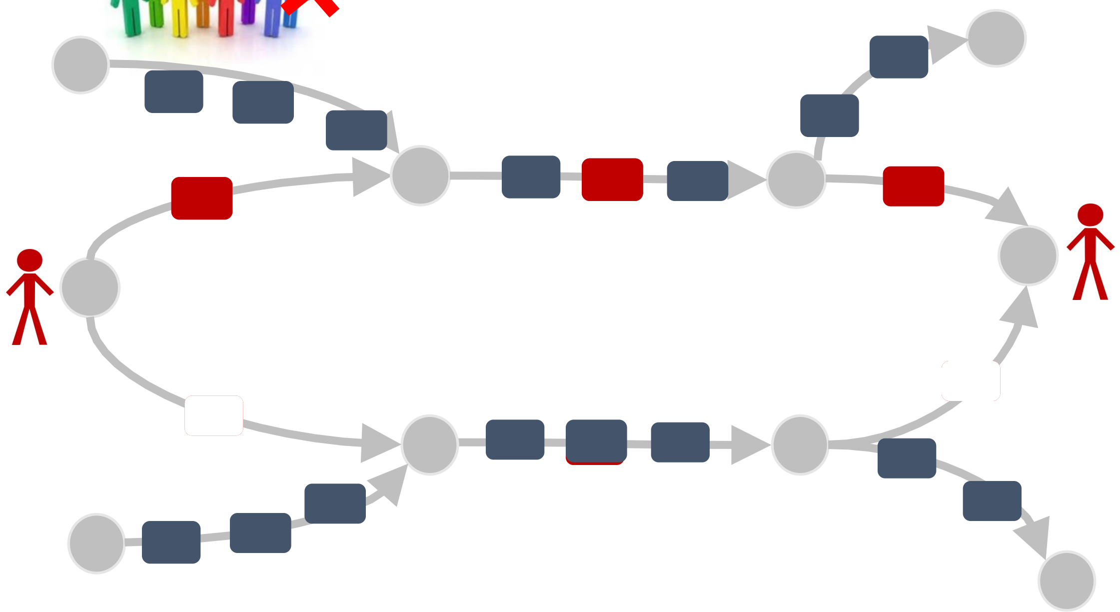
even when the rate of change is high, i.e. a large fraction can turn over at every step.

Need for adaptive learning

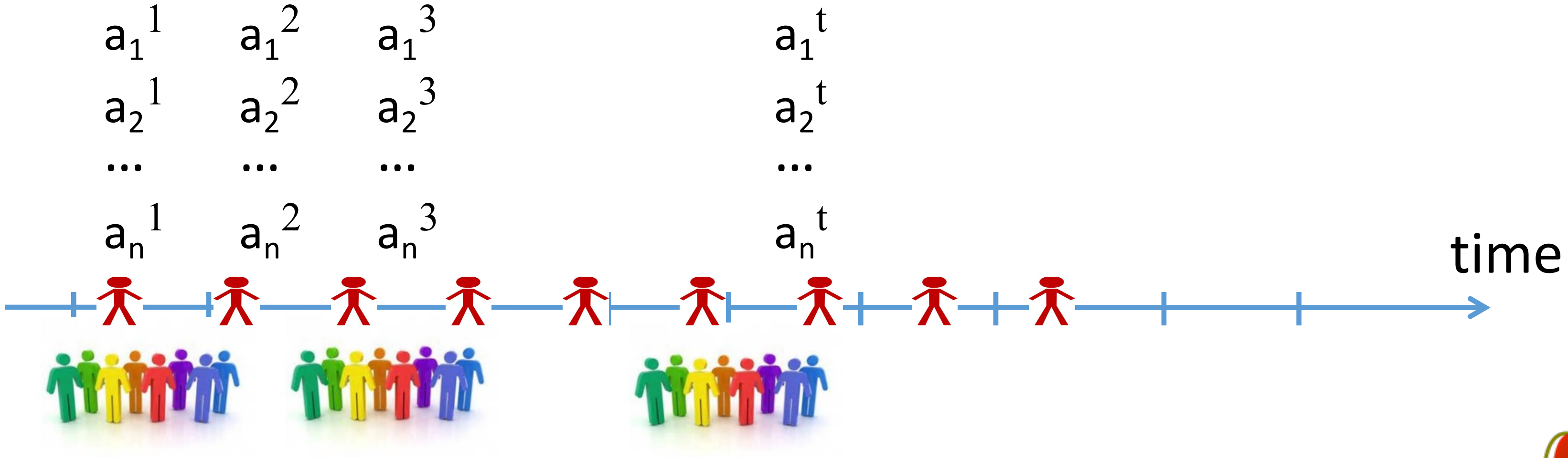


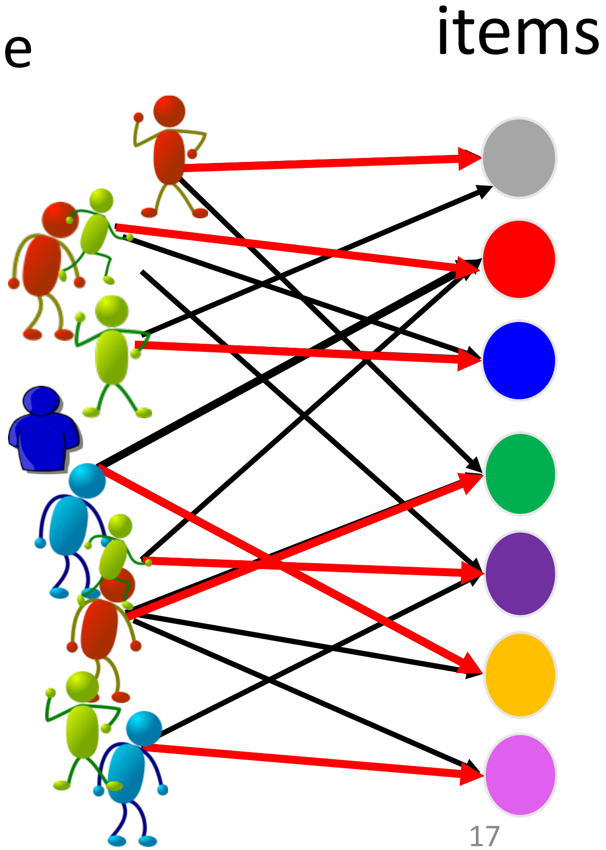
Example routing

- Strategy = path
- Best "fixed" strategy in hindsight very weak in changing environment
- Learners can adapt to the changing environment



Need for adaptive learning

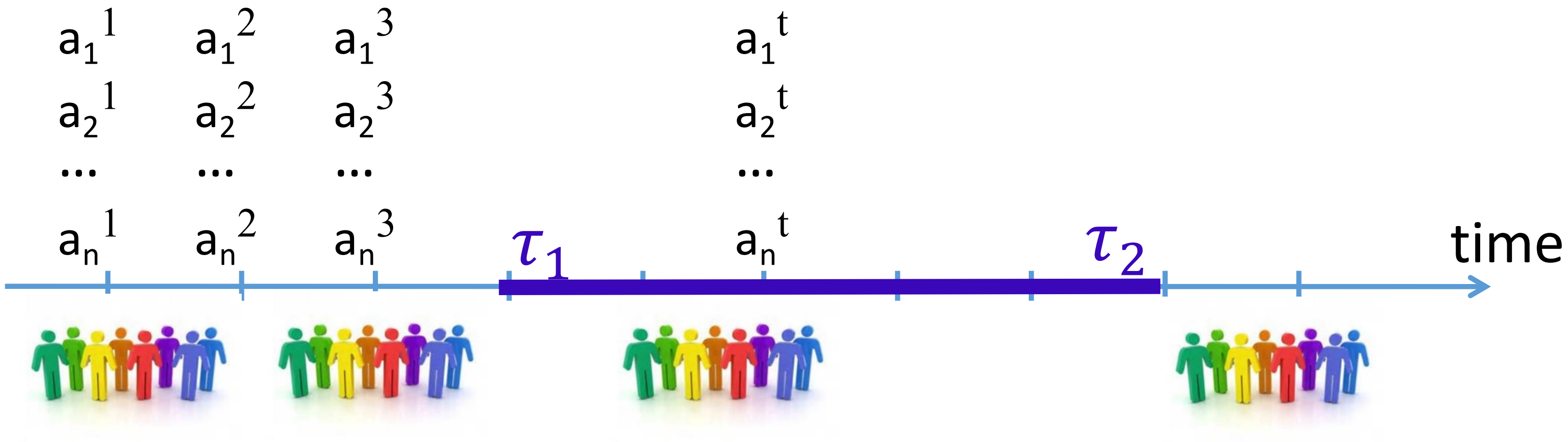




Example 2: matching (item auction)

- Strategy ≈ choose an item
- Best "fixed" strategy in hindsight very weak in changing environment
- Learners can adapt to the changing environment

Adaptive Learning



- Adaptive regret [Hazan-Seshadiri'07, Luo-Schapire'15, Blum-Mansour'07, Lehrer'03]

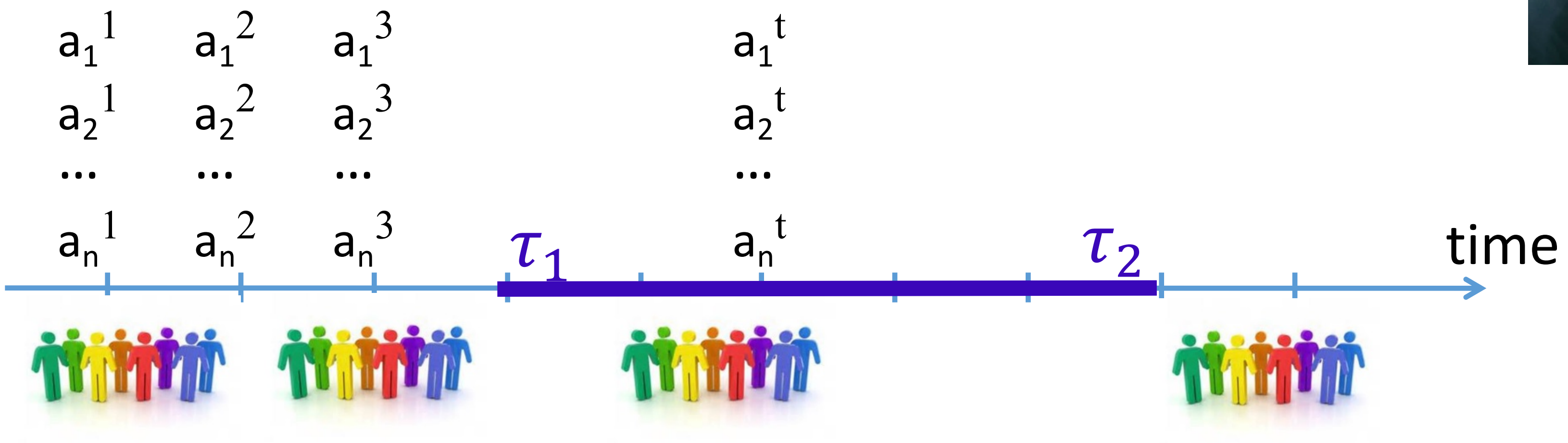
  for all player i, strategy x and interval $[\tau_1, \tau_2]$

$$R_i(x, \tau_1, \tau_2) = \sum_{t=\tau_1}^{\tau_2} cost_i(a^t; v^t) - cost_i\left(x, a_{-i}^t; v^t\right) \leq o(\tau_2 - \tau_1)$$

rates of $\sim\sqrt{\tau_2 - \tau_1}$

$\Rightarrow$ Regret with respect to a strategy that changes k times $\leq \sim\sqrt{kT}$

Adaptive Learning



- Adaptive regret [Foster,Li,Lykouris,Sridharan,T'16]

  for all player i, strategy x and interval $[\tau_1, \tau_2]$

$$R_i(x, \tau_1, \tau_2) = \sum_{t=\tau_1}^{\tau_2} cost_i(a^t; v^t) - (1+\epsilon)\, cost_i\left(x, a_{-i}^t; v^t\right) \leq O(\mathrm{k} \log d/\epsilon)$$

Regret with respect to a strategy that changes k times

Using any of MWU (Hedge), Regret Matching, etc. mixed with a bit of "forgetting"

Result (Lykouris, Syrgkanis, T'16) :

Bound average welfare close to Price of Anarchy for Nash

even when the rate of change is high, $\boldsymbol{p \approx \frac{1}{\log n}}$ with n players

assuming **adaptive** no-regret learners, and stable enough optimum solution

- Worst case change of player type $\Rightarrow$ need for adapting to changing environment
- Sudden large change is unlikely

No-regret and Price of Anarchy

Low regret:

$$R_i(x) = \sum_{t=1}^{T} cost_i(a^t; v^t) - cost_i\left(x, a_{-i}^t; v^t\right) \le o(T)$$

Best action varies with choices of others…

Idea: Consider Optimal Solution

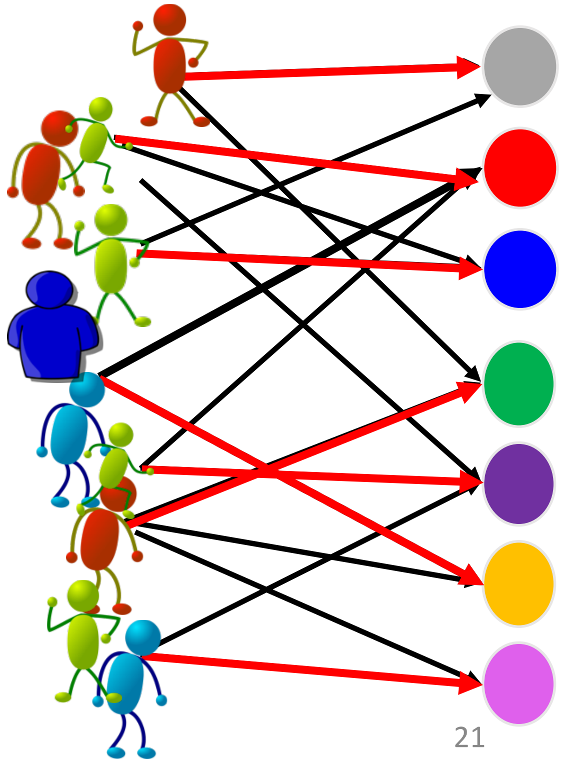
Let $x = a_i^*$ the choice in OPT

No regret for all players i:

$$\sum_t cost_i(a^t) \le \sum_t cost_i(\boldsymbol{a_i^*}, a_{-i})$$

Players don't have to know $\boldsymbol{a_i^*}$

items



Proof Technique: Smoothness (Roughgarden'09)

Consider optimal solution: player i does action a_i^* in optimum

Nash equilibrium a has

$$\sum_i cost_i(a) \leq \sum_i cost_i(a_i^*, a_{-i}) \quad \leq \lambda\, OPT + \mu\, cost(a)$$

A game is (λ,μ)-smooth (λ > 0; μ< 1):

if for all strategy vectors a

$$\sum_i cost_i(a_i^*, a_{-i}) \leq \lambda OPT + \mu\, cost(a)$$

A Nash equilibrium a has cost(a) $\leq \frac{\lambda}{1-\mu}$OPT

Smoothness and no-regret learning

Consider optimal solution: player i does action a_i^* in optimum

No regret: $\sum_t cost_i(a^t) \leq \sum_t cost_i(a_i^*, a_{-i}^t)$ (doesn't need to know a_i^*)

A cost minimization game is (λ,μ)-smooth (λ > 0; μ< 1):

if for all strategy vectors a

$$\frac{1}{T}\sum_t \sum_i cost_i(a^t) \leq \frac{1}{T}\sum_t \sum_i cost_i(a_i^*, a_i^t) \leq \lambda\, OPT + \mu \frac{1}{T}\sum_t cost(a^t)$$

A no-regret sequence a^t has

$$\frac{1}{T}\sum_t cost(a^t) \leq \frac{\lambda}{1-\mu}\text{Opt}$$

Smoothness Example:

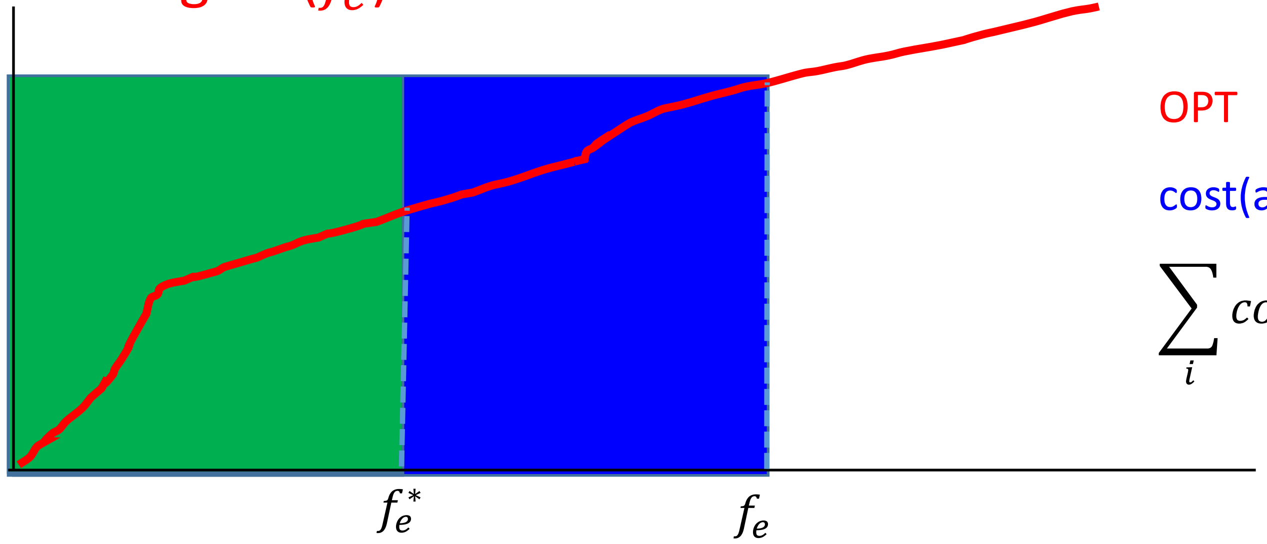
(Roughgarden-Tardos'02):

Monotone $cost_i$ and small players: game is (1,1)-smooth:

a_i^* (Opt) with ∀ action vector a

$$\sum_i cost_i(a_i^* , a_{-i}) \leq OPT + cost(a)$$

Cost of an edge: c(f_e)



OPT = f_e^* players cost c(f_e^*)

cost(a) = f_e players cost c(f_e)

$\sum_i cost_i(a_i^*, a_{-i}) =$ f_e^* players cost c(f_e)

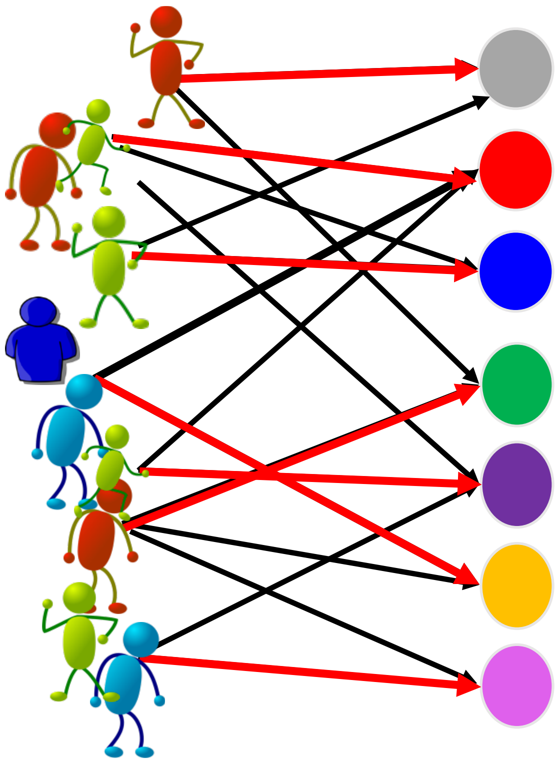
Smoothness Example:

Unit demand auction first price: (λ,1)-smooth (where $\lambda = 1\text{−}1/e$)

Monotone utility u_i =expected value – price

Smooth mechanism (Syrgkanis-T'13): $\exists\, a_i^*$ (Opt) with $\forall$ action vector a

$$\sum_i u_i({a_i}^*, a_{-i}) \geq \lambda\, OPT - Revenue(a)$$



a_i^* = bit on item allocated in optimum

Smooth inequality true item by item:

$$u_i(a_i^*, a_i) \geq \lambda v_{ij^*} - price_{j^*}$$

- If price high, then right hand side is non-positive
- Else: players benefit $\geq \lambda$ OPT from bidding on opt item

Smoothness in utility games

- Vetta utility games are (1,1)-smooth Vetta FOCS'02
- First price is (1-1/e)-smooth (we have seen ½, see also Hassidim, Kaplan, Mansour, Nisan EC'11)
- All pay auction ½-smooth
- First position auction (GFP) is ½-smooth
- Variants with second price (see also Christodoulou, Kovacs, Schapira ICALP'08)

Other applications include:

- public goods
- Fair sharing (Kelly, Johari-Tsitsiklis)
- Walrasian Mechanism (Babaioff, Lucier, Nisan, and Paes Leme EC'13)

Examples of “smoothness bounds”

- Monotone increasing congestion costs (1,1) smooth

  $\Rightarrow$ Nash cost ≤ opt of double traffic rate (Roughgarden-T’02)
- affine congestion cost are (1, ¼) smooth (Roughgarden-T’02)

  $\Rightarrow$ 4/3 price of anarchy

- Atomic game (players with >0 traffic) with linear delay (5/3,1/3)-smooth (Awerbuch-Azar-Epstein & Christodoulou-Koutsoupias’05)

  $\Rightarrow$ 2.5 price of anarchy

Resulting bounds are tight

Adapting smoothness to dynamic populations

Inequality we "wish to have"

$$\sum_t cost_i(a^t; v^t) \leq \sum_t cost_i(a_i^{*t}, a_{-i}^t; v^t)$$

where a_i^{*t} is the optimum strategy for the players at time t.

with stable population = no regret for a_i^*

Too much to hope for in dynamic case:

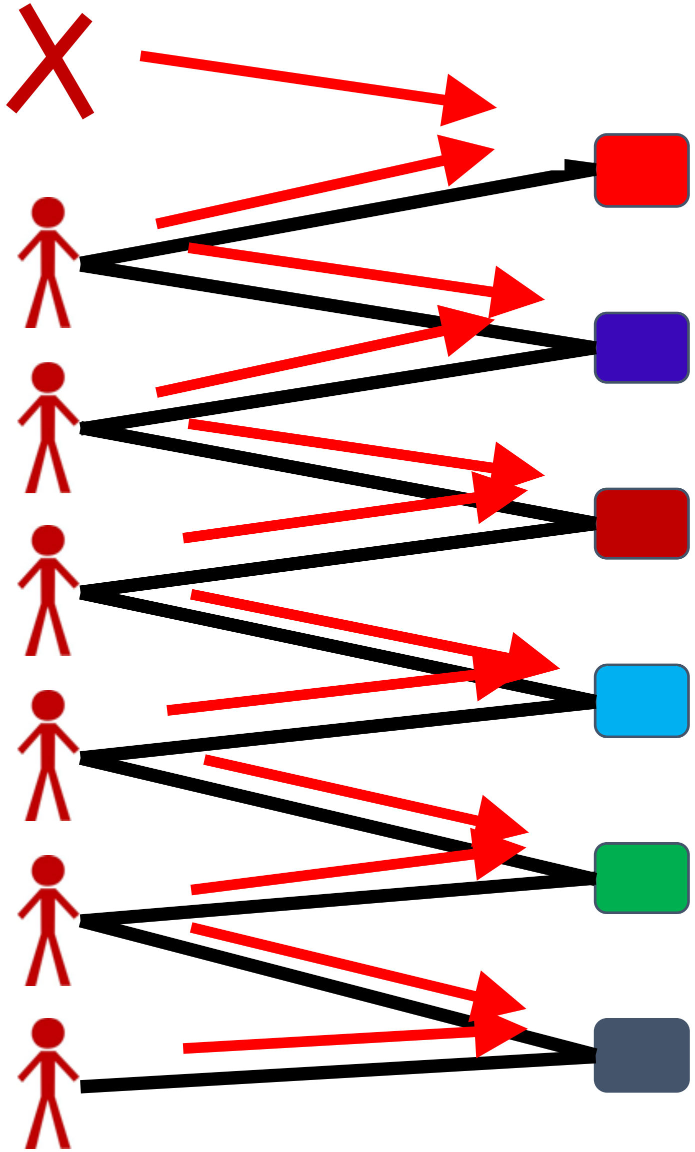
- sequence a^{*t} of optimal solutions changes too much.
- No hope of learners not to regret this!

Change in Optimum Solution

True optimum is too sensitive

- Example using matching
- The optimum solution
- One person leaving
- Can change the solution for everyone

- Np changes each step → No time to learn!! (we have p>>1/N)



Theorem (high level)

If a game satisfies a “smoothness property” [Roughgarden’09]

The welfare optimization problem admits an approximation algorithm whose outcome $\widetilde{a}^*$ is stable to changes in one player’s type

Then any adaptive learning outcome is approximately efficient even when the rate of change is high.

Proof idea: use this approximate solution as $\widetilde{\boldsymbol{a}}^*$ in Price of Anarchy proof

With $\widetilde{\boldsymbol{a}}^*$ not changing much, learners have time to learn not to regret following $\widetilde{\boldsymbol{a}}^*$

Note: learner doesn’t have to know $\widetilde{\boldsymbol{a}}^*$!!

Do Stable Solutions Exist?

- How close can we remain to the optimum, while being stable?

- How much change can we manage, while being stable?

Recall: Regret of adaptive learning is bounded by $\leq \sqrt{kT \log d}$

Approximate shifting regret bounded by $\leq \frac{k \log d}{\epsilon}$

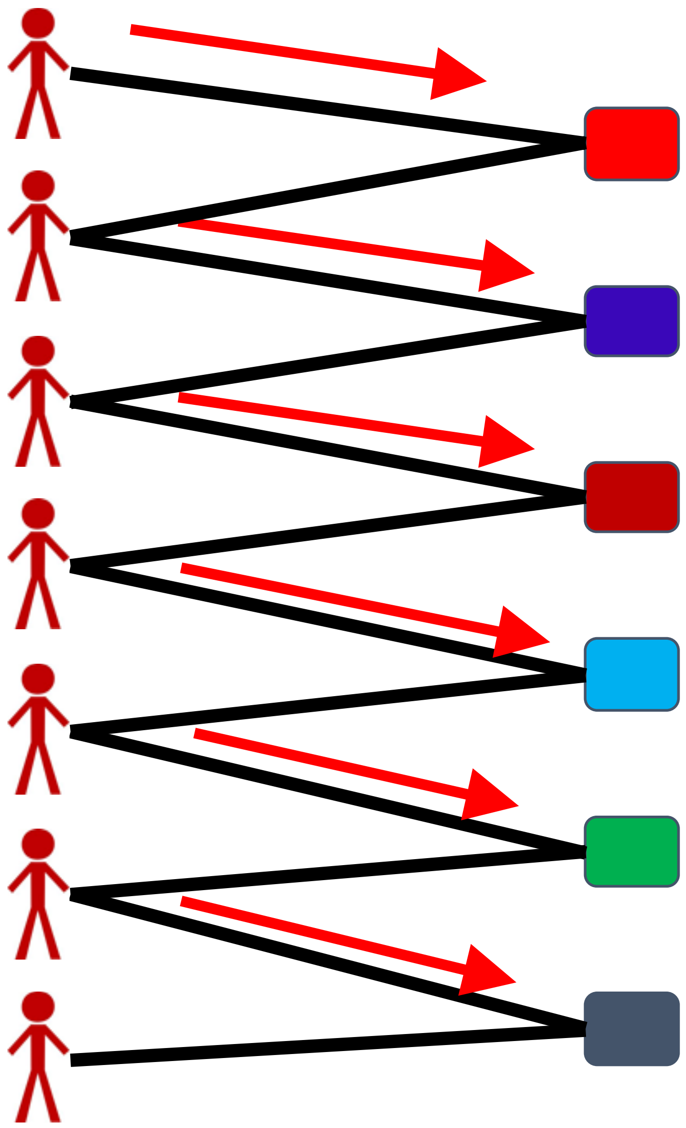
with respect to any strategy that changes k times

Stable ≈ Optimum in Matching

True optimum is too sensitive

- Use greedy allocation: assign large values first (loss of factor of 2)
- Use coarse approximation of value, e.g., power of 2 only
- Potential function argument:

  increase in log value of allocation only m $\log v_{max}$, decrease due to departures



Use Differential Privacy $\rightarrow$ Stable Solutions

Joint privacy [Kearns et al. ’14, Dwork et al. ‘06]

A randomized algorithm is jointly differentially private if

- when input from player **i** changes
- the probability of change in solution of players other than **i** is smaller than ϵ

- Turn a sequence of randomized solutions to a randomized sequence with small number of changes using Coupling Lemma
- and handling “failure probabilities” of private algorithms

Application 1: Large Congestion Games

- Using joint differentially private algorithm of Rogers et al EC'15,
- the (5/3,1/3)-smoothness congestion with affine cost:

Theorem. Atomic congestion game with m edges, and affine and increasing costs:

$$\frac{1}{T}\sum_{t} Cost(a^t; v^t) \leq 2.5(1+\epsilon)\frac{1}{T}\sum_{t} \mathrm{OPT}(v^t)$$

with $p = O\left(\frac{poly(\epsilon)}{poly(m)\ polylog(n)}\right)$ if each player controls only a 1/n fraction of the total flow.

Almost a constant fraction of change each step: dependence on number of players only polylog

Other Applications

Using joint differentially private algorithm of Hsu et al ’14

Theorem 2. Matching markets if values are $[\rho,1]$

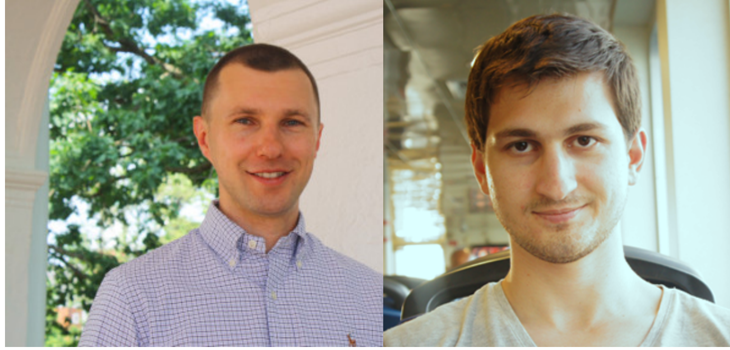
$$\frac{1}{T}\sum_t W(a^t; v^t) \geq \frac{1}{4(1+\epsilon)}\frac{1}{T}\sum_t \mathrm{OPT}(v^t) \text{ with } p = O\left(\frac{\rho^2\epsilon^2}{polylog(m,1/\rho,1/\epsilon)}\right)$$

Theorem 3. Large Combinatorial Markets with Gross-Substitutes

$$\frac{1}{T}\sum_t W(a^t; v^t) \geq \frac{1}{2(1+\epsilon)}\frac{1}{T}\sum_t \mathrm{OPT}(v^t) \text{ with } p = O\left(\frac{\rho^5\epsilon^5}{m\ polylog(n)}\right)$$

Each item in large supply $\Omega\left(polylog(n)\log(\frac{1}{\epsilon},\frac{1}{\rho})\right)$ and $\Theta(n)$ items

Do players really learn?

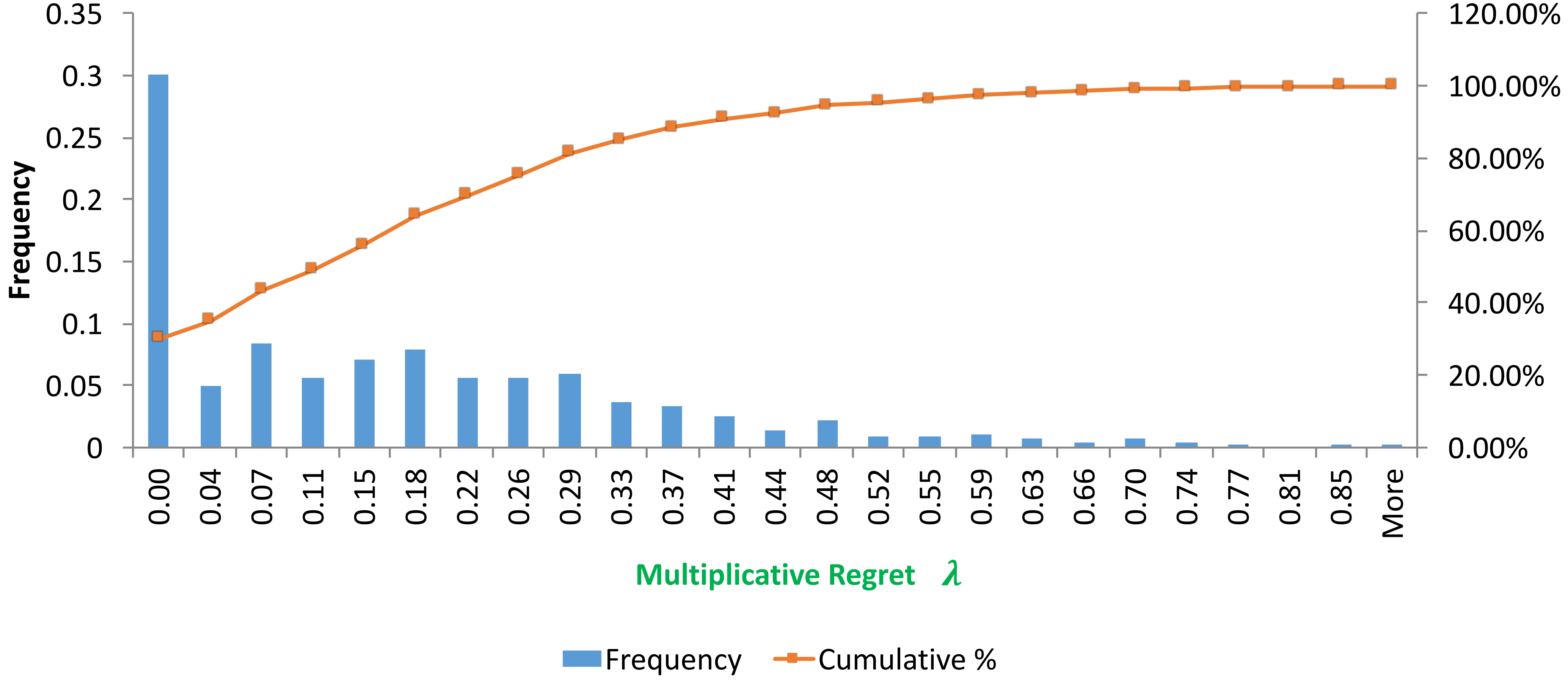


- Data from Microsoft: 9 frequent bid changing advertisers

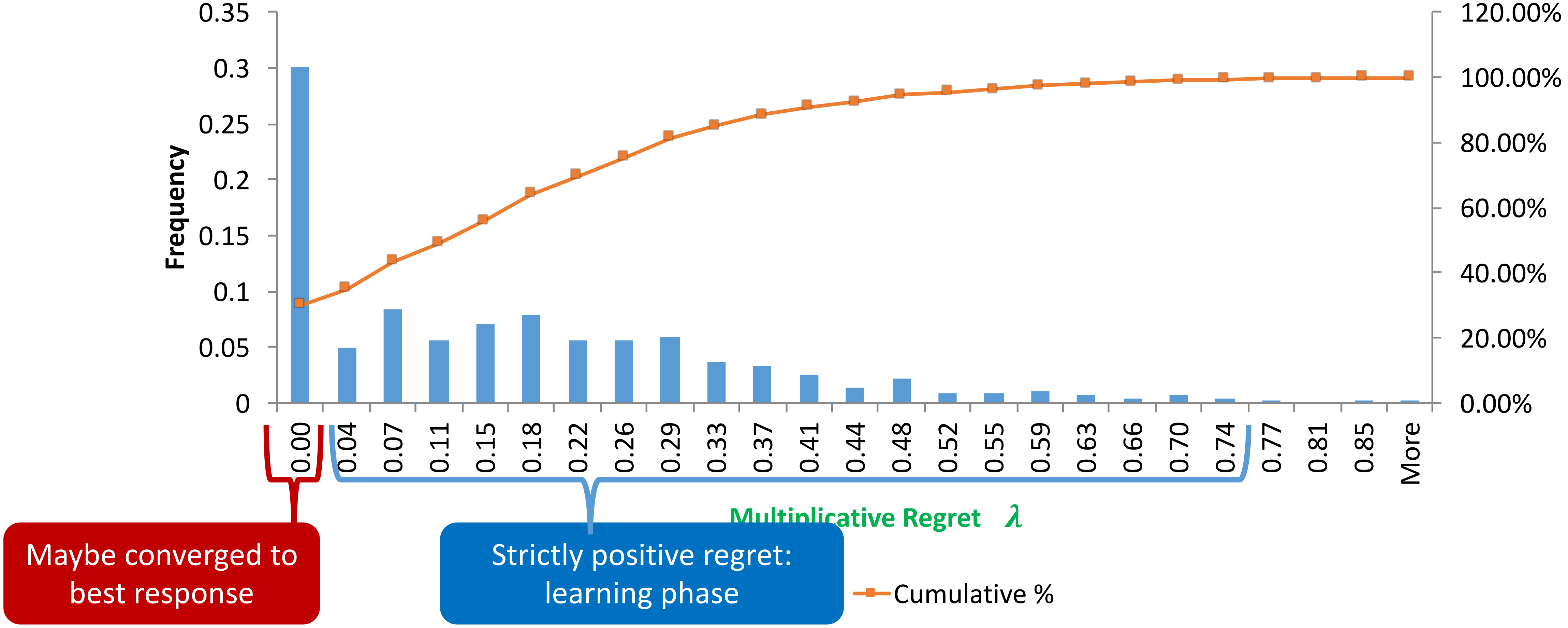
Value of advertiser?

- Nekipelov, Syrgkanis, T'15: infer the value smallest multiplicative regret

Distribution of smallest rationalizable multiplicative regret



Distribution of smallest rationalizable multiplicative regret



Conclusions

Learning in games:

- Good way to adapt to opponents
- No need for common prior
- Takes advantage of opponent playing badly.

Learning players do well even in dynamic environments

- Stable approx. solution + good PoA bound $\Rightarrow$ good efficiency with dynamic population
- Strong connection of stable solutions with differential privacy