Prepared by Julian García, based on material by
Arun Konagurthu

Workshop 24
Equilibrium Selection, Game Theory Applications, Epilogue

FIT 3139 Computational Modelling and Simulation

MONASH University
Information Technology

Equilibrium Selection

- Refinements: Solution concepts that are stricter.
- Dynamics: Which equilibrium is more likely to arise from a simple process carried out by the agents.

How to choose?

	L	C	R
L	5, 5	7,8	2,1
C	8,7	6,6	5,8
R	1, 2	8,5	4,4

$s_1^* = [(1/4, 3/4, 0), (1/4, 3/4, 0)]$ $S_2^* = [(0, 1/3, 2/3), (0, 1/3, 2/3)]$

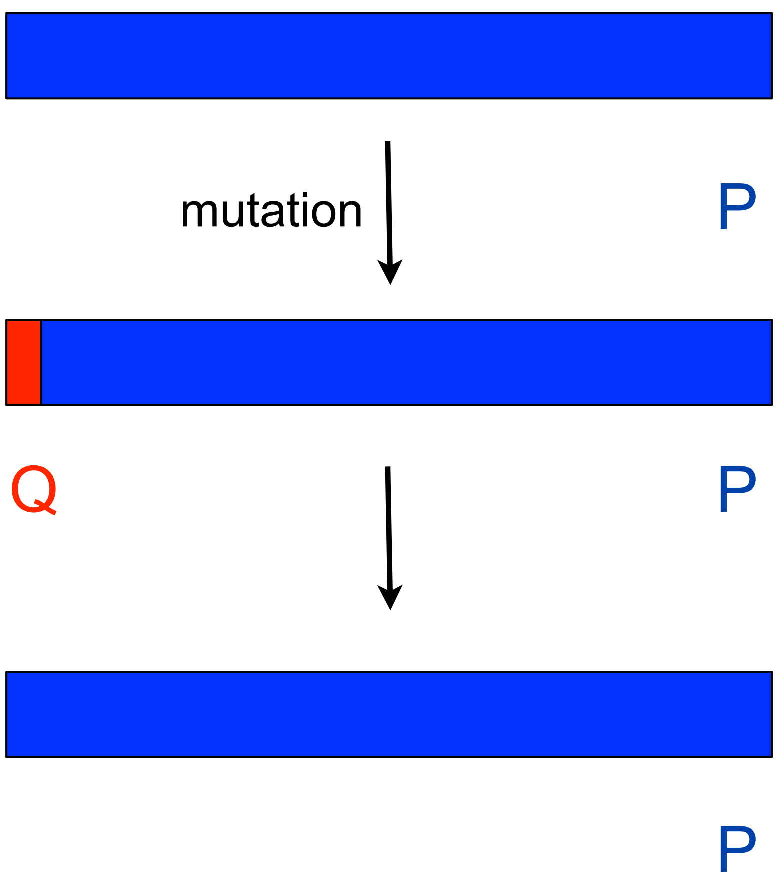
Equilibrium Selection

- **Refinements**: Solution concepts that are stricter.
- **Dynamics**: Which equilibrium is more likely to arise from a simple process carried out by the agents.

ESS: Refinement

Evolutionary Game Theory

- Focus on symmetric equilibria, i.e., profiles are of the form s = (p, p).
- Imagine a <u>large population</u> of players, randomly drawn to play a symmetric game
- The payoffs of the game are assumed to be "fitnesses" in the sense that a process of natural selection favors those earning higher payoffs
- What is the condition for selection to <u>oppose the invasion of a mutant</u>?



P is ESS if it is able to repel any mutant Q that comes in a small quantity

Start with the definition of symmetric N.E.

$$\prod(P,P) \geq \prod(Q,P) \qquad \forall Q \neq P$$

Observe that the payoff is linear:

$$\prod(P,\epsilon Q+(1-\epsilon)P) > \prod(Q,\epsilon Q+(1-\epsilon)P) \qquad \forall Q \neq P, \text{ small } \epsilon$$

P is the incumbent, Q is the mutant

Evolutionarily Stable Strategy

P is ESS if ...

$$\prod(P, \epsilon Q + (1-\epsilon)P) > \prod(Q, \epsilon Q + (1-\epsilon)P) \qquad \forall Q \neq P, \text{ small } \epsilon$$

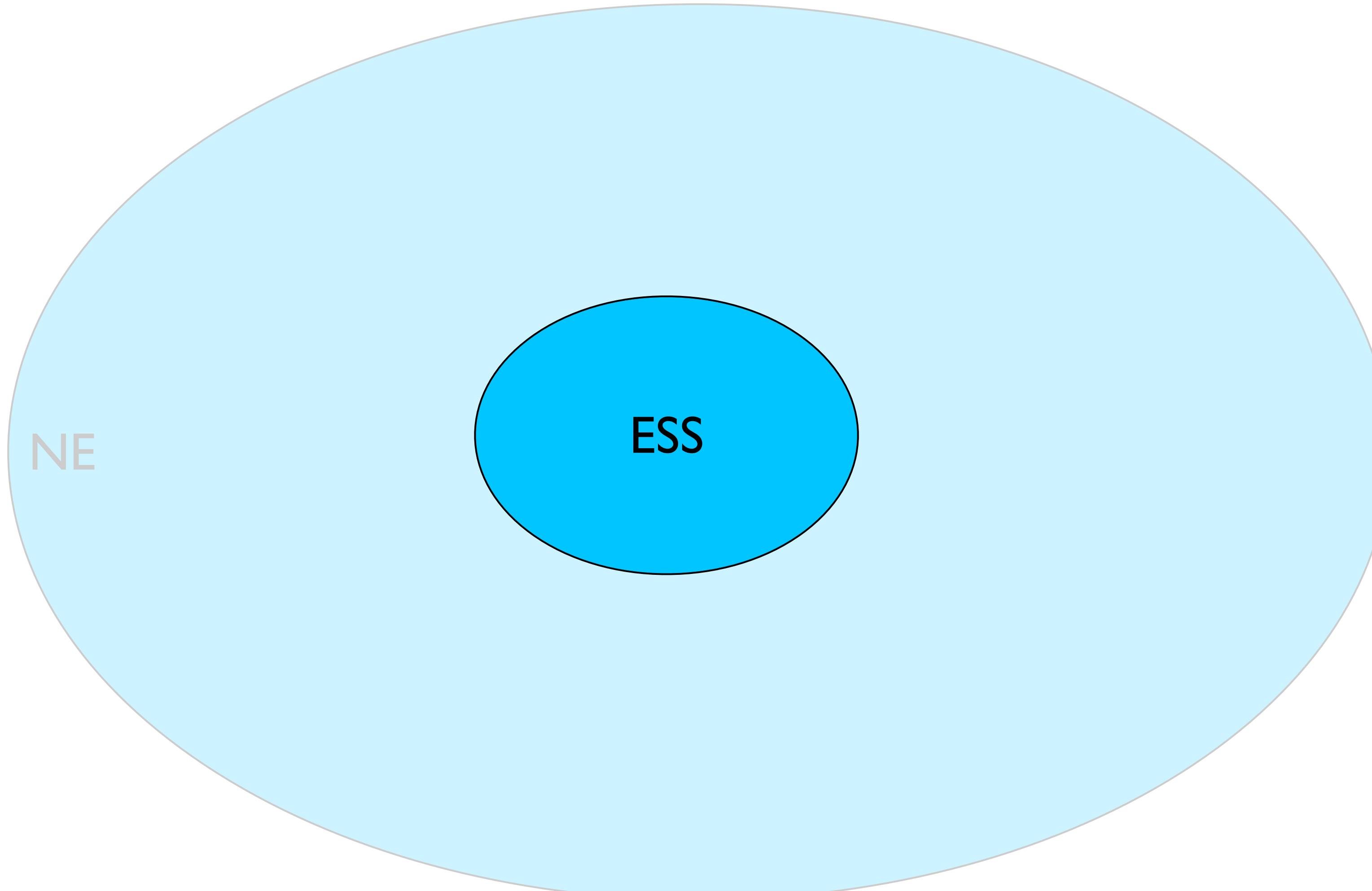
or equivalently...

Nash Equilibrium $\longrightarrow$ $\prod(P,P) \geq \prod(Q,P) \qquad \forall Q$

or

$$\text{if } \prod(P,P) = \prod(Q,P) \rightarrow \prod(P,Q) > \prod(Q,Q) \qquad \forall Q \neq P$$

plus an extra condition



ESS is a refinement of Nash.

Nash refinements

- ESS
- Trembling hand
- Subgame perfection

Replicator Dynamics

Replicator dynamics

$$x = (x_1, \ldots, x_n)$$

$$\sum_{i=1}^{n} x_i = 1$$

$$\dot{x}_i = x_i(f_i(x) - \phi) \qquad i = 1, 2, \ldots n$$

$$A = [a_{ij}]$$

$$f_i(x) = \sum_{j=1}^{n} x_i a_{ij}$$

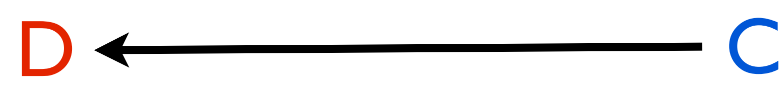
$$\phi = \sum_{i=1}^{n} x_i f_i(x)$$

Taylor and Jonker (1978)

Prisoner's dilemma

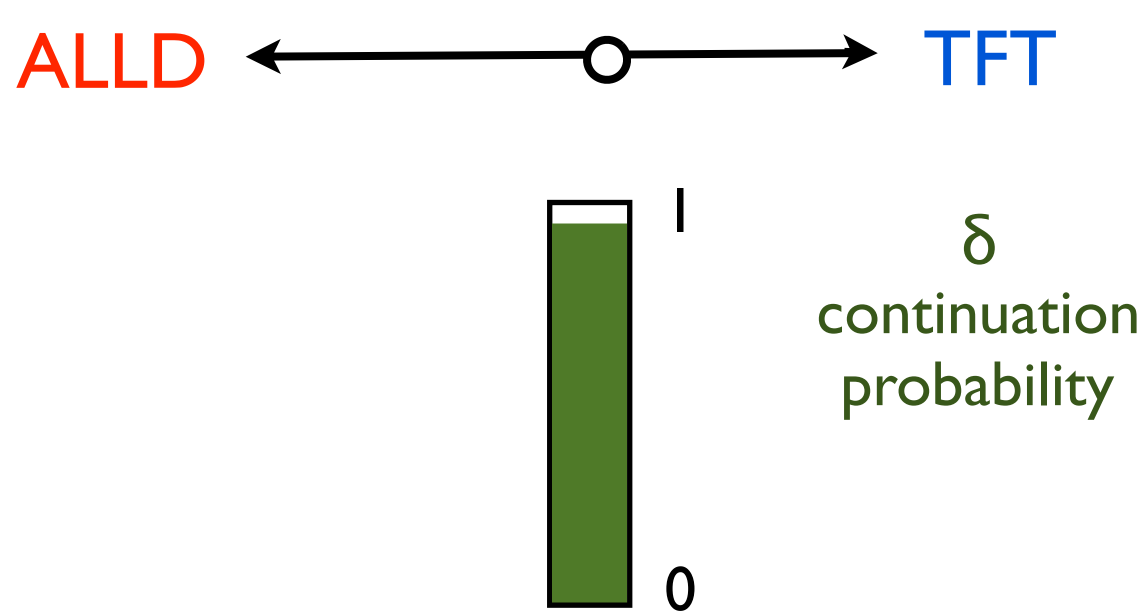
	C	D
C	3	0
D	4	1

One-shot



	TFT	ALLD
TFT	3	δ
ALLD	$4-3\delta$	1

Repeated



$$\dot{x}_i = x_i(f_i(x) - \phi) \qquad i = 1, 2, ...n$$

- If x* is a Nash equilibrium of A, it is a fixed point of the replicator dynamics.

Taylor and Jonker (1978)

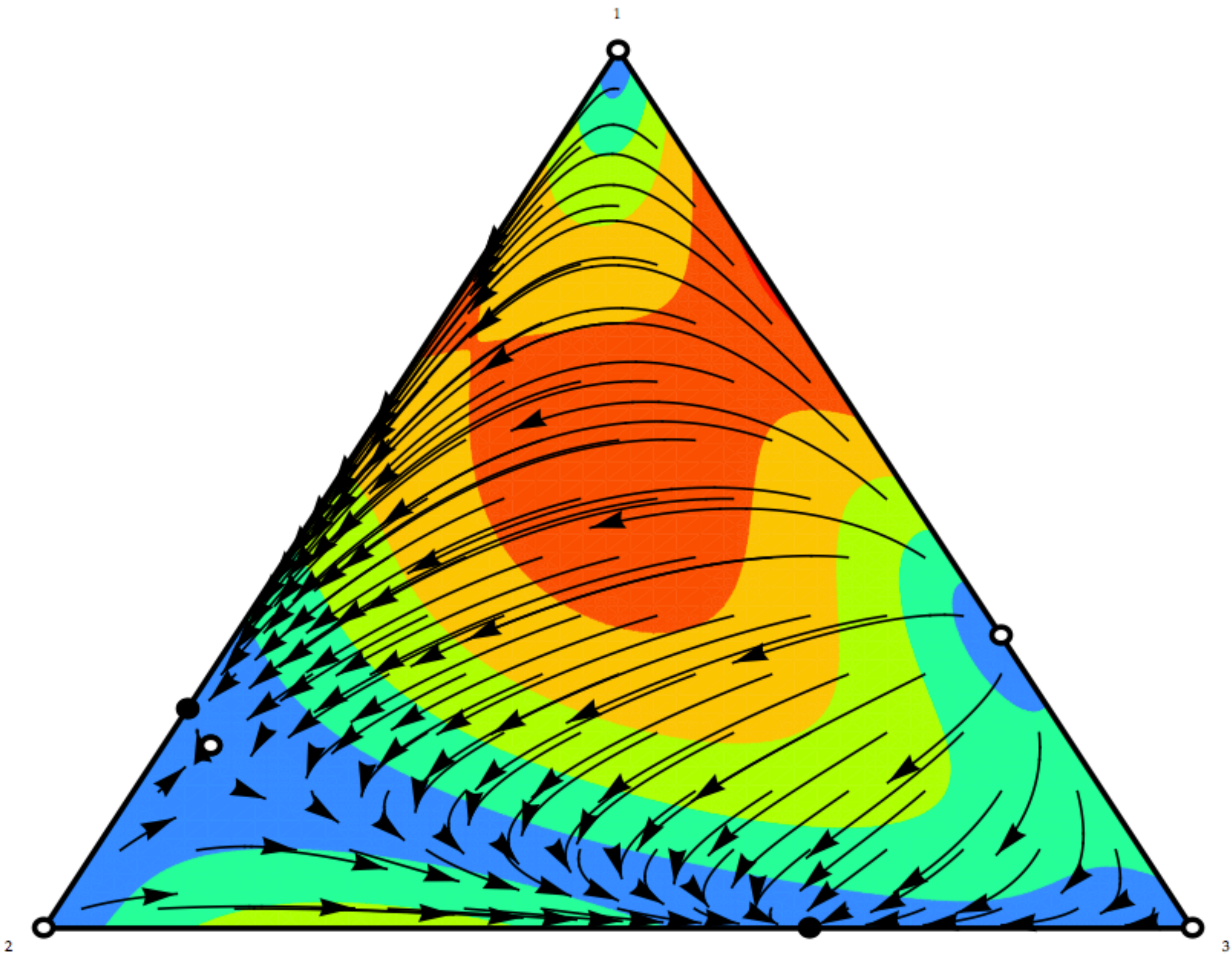
	L	C	R
L	5, 5	7,8	2,1
C	8,7	6,6	5,8
R	1, 2	8,5	4,4

$x^* = (1/4, 3/4, 0)$

$x^* = (0, 1/3, 2/3)$

Two ESS

which one to pick?



Large "basin of attraction"

To appear in **"Reviews of Nonlinear Dynamics and Complexity" Vol. II**, Wiley-VCH, 2009, edited by H.-G. Schuster

1

Stochastic evolutionary game dynamics

Arne Traulsen and Christoph Hauert

Learning in a finite population of agents
can also be modelled as an Ergodic Markov Chain

Unique stationary distribution solves Equilibrium Selection

Multiagent Reinforcement Learning: Theoretical Framework and an Algorithm

Junling Hu and Michael P. Wellman
Artificial Intelligence Laboratory
University of Michigan
Ann Arbor, MI 48109-2110, USA
{junling, wellman}@umich.edu
http://ai.eecs.umich.edu/people/{junling,wellman}

Abstract

In this paper, we adopt general-sum stochastic games as a framework for multiagent reinforcement learning. Our work extends previous work by Littman on zero-sum stochastic games to a broader framework. We design a multiagent Q-learning method under this framework, and prove that it converges to a Nash equilibrium under specified conditions. This algorithm is useful for finding the optimal strategy when there exists a unique Nash equilibrium in the game. When there exist multiple Nash equilibria in the game, this algorithm should be combined with other learning techniques to find optimal strategies.

The framework we adopt is ***stochastic games*** (also called *Markov games*) [4, 15], which are the generalization of the Markov decision processes to the case of two or more controllers. Stochastic games are defined as non-cooperative games, where agents pursue their self-interests and choose their actions independently.

Littman [6] has introduced 2-player zero-sum stochastic games for multiagent reinforcement learning. In zero-sum games, one agent's gain is always the other agent's loss, thus agents have strictly opposite interests. In this paper, we adopt the framework of general-sum stochastic games, in which agents need no longer have opposite interests. General-sum games include zero-sum games as special cases. In general-sum games, the notions of "optimality" loses its meaning since each agent's payoff depends on other agents' choices. The solution concept ***Nash equilibrium*** [8] is

Dynamics inspired by classic AI

On Using Game Theory

- Given a game predict the outcome.
- Build or design a game such that the outcome has properties that I want (e.g. Auctions)

Sealed bid auctions

- 1 Auctioneer, 1 Good to sell.
- n bidders who want the good.
- I want to model bidding, so I need to know what bidders want.

$$v = (v_1, v_2, \ldots v_n)$$

PRIVATE valuations

v_i is the maximum price i is willing to pay.

- Utility of bidder i

$$U_i = \begin{cases} 0 & \text{if } i \text{ loses} \\ v_i - p & \text{if } i \text{ wins} \end{cases}$$

- The task of bidder i is to choose a bid b_i

But what are the rules of the game?

Auctioneer

- Gets a vector of bids: $b = (b_1, b_2, \ldots, b_n)$

  Decides:

  A Who wins

  B What price do they pay

Mechanism design: How to design the game so that the outcome has desirable properties

Option 1: Nicest auctioneer

- Give the good for free, to the person that wants it the most.

- Remember, seller cannot see private valuations, only bids.

A Who wins: Highest bidder

B What price do they pay: 0

What should a bidder do? State largest bid. **Lie**.

Winner not necessarily aligned with what the auctioneer wanted

Option 2: First price auction

- Winner is the highest bidder, pays her bid.

A Who wins: $\max_i(b1, \ldots, b_i, \ldots, b_n)$

B What price do they pay: b_i

$$U_i = \begin{cases} 0 & \text{if } i \text{ loses} \\ v_i - p & \text{if } i \text{ wins} \end{cases}$$

What should a bidder do?

- bid less than the valuation $b_i < v_i$
- trade-off: probability of winning, amount paid upon winning

Bids depend non-trivially on n, and what I believe are the valuations of others

Option 2: First price auction

- Winner is the highest bidder, pays her bid.

A Who wins: $\max_i(b1, \ldots, b_i, \ldots, b_n)$

B What price do they pay: b_i

What should a bidder do?

Assume that valuations are uniformly distributed in [0, 1]

For 2 bidders: $s^* = \left(\frac{1}{2}v_1, \frac{1}{2}v_2\right)$ Nash: bid half your valuation

For n bidders, not unique: $s^* = \left(\frac{n-1}{n}v_1, \ldots, \frac{n-1}{n}v_i, \ldots, \frac{n-1}{n}v_n\right)$

Not great, because it encourages underbidding.

Option 3: Second price auction

A Who wins: Highest bidder

B What price do they pay: second largest bid

Example:
Bids are \$42, \$62, \$73 and \$200 → \$200 bidder wins, but pays \$73

What should a bidder do?

One dominant strategy.

Option 3: Second price auction

$$U_i = \begin{cases} 0 & \text{if } i \text{ loses} \\ v_i - p & \text{if } i \text{ wins} \end{cases}$$

What should a bidder do?

- Bidder i *tells the truth*, and bids v_i
- Two possible outcomes, win or lose

lose

Bid <u>less</u> than v_i :

➡Still lose, and get 0 because someone else is highest bidder.

Bid <u>more</u> than v_i :

➡Not enough to win, still lose, and still get 0

➡ Win, but pay more than my valuation, therefore getting a negative payoff.

win

Bid <u>more</u> than v_i :

➡ Still win, pay the same price (second price), same utility

Bid <u>less</u> than v_i :

➡ Still win, pay the same price (second price), same utility

➡Lose and get 0

Option 3: Second price auction

What should a bidder do?

Bid v_i is a dominant strategy

your bid has no effect on the price you pay, therefore you tell the truth and
the good goes to the bidder that wants it the most.

lose

Bid less than v_i:

➡ Still lose, and get 0 because someone else is highest bidder.

Bid more than v_i:

➡ Not enough to win, still lose, and still get 0

➡ Win, but pay more than my valuation, therefore getting a negative payoff.

win

Bid more than v_i:

➡ Still win, pay the same price (second price), same utility

Bid less than v_i:

➡ Still win, pay the same price (second price), same utility

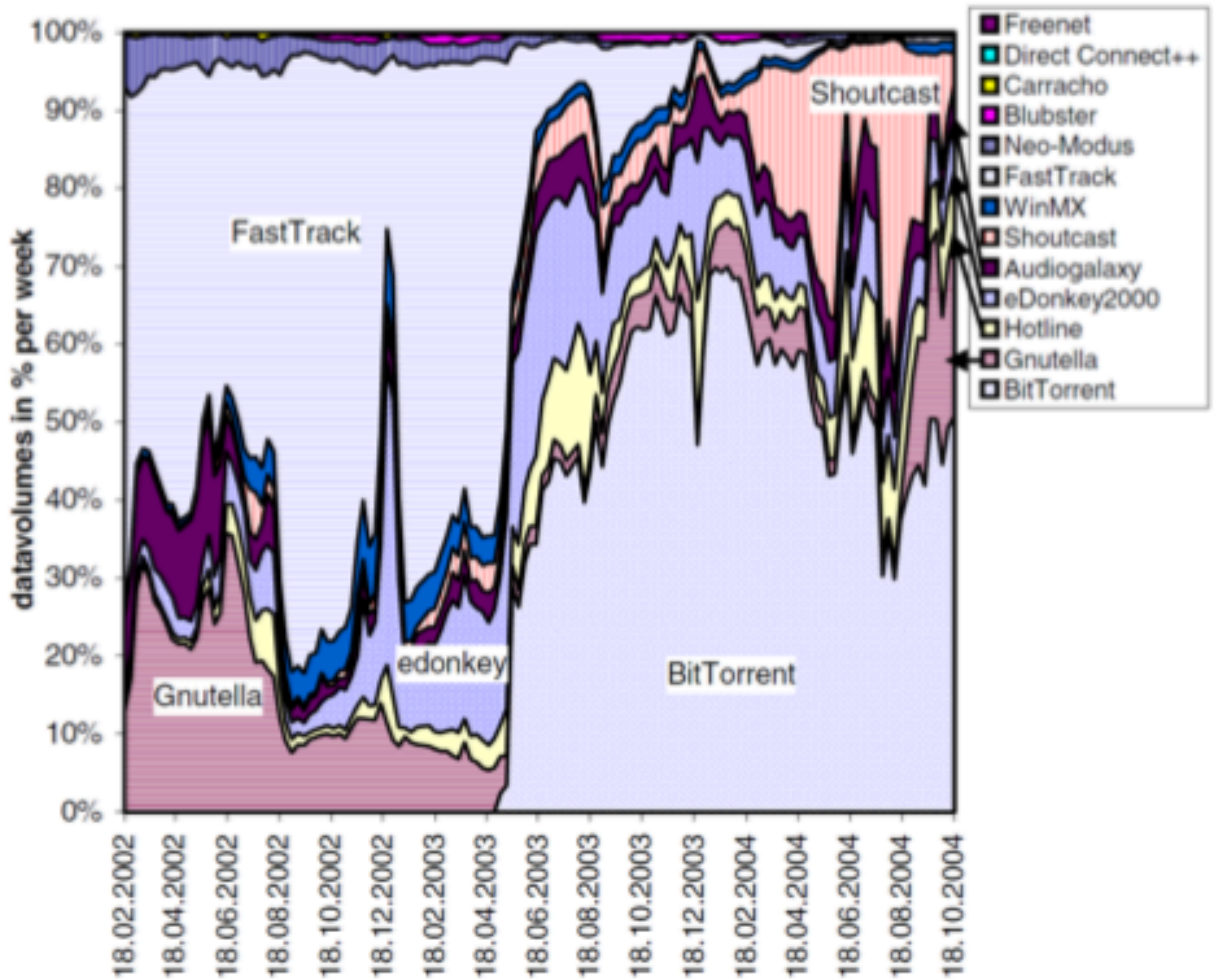
➡ Lose and get 0

eBay Inc.

Type	Public
Traded as	NASDAQ: EBAY NASDAQ-100 Component S&P 500 Component
Industry	Internet
Founded	September 3, 1995; 23 years ago (as AuctionWeb)
Founder	Pierre Omidyar
Headquarters	San Jose, California, U.S.
Key people	Thomas J. Tierney (Chairman) Devin Wenig (CEO)
Services	Online shopping
Revenue	▲ US$9.567 billion (2017)[1]
Operating income	▼ US$2.265 billion (2017)[1]
Net income	▼ US$-1.016 billion (2017)[1]
Total assets	▲ US$25.981 billion (2017)[1]
Total equity	▼ US$8.063 billion (2017)[1]
Number of employees	~14,100 (December 2017)[1]
Subsidiaries	eBayClassifieds, Kijiji, iBazar, GittiGidiyor, Gumtree, G-Market, Stubhub, Half.com, Marktplaats.nl
Website	ebay.com

https://youtu.be/SZV_J92fY_I

Incentives at play



Traffic proportions due to different P2P protocols in the US between 2002 and 2004 (Eberspächer and Schollmeier, 2005).

Incentives at play

The Miner's Dilemma

Ittay Eyal
Cornell University

***Abstract*—An open distributed system can be secured by requiring participants to present proof of work and rewarding them for participation. The Bitcoin digital currency introduced this mechanism, which is adopted by almost all contemporary digital currencies and related services.**

A natural process leads participants of such systems to form pools, where members aggregate their power and share the rewards. Experience with Bitcoin shows that the largest pools are often open, allowing anyone to join. It has long been known that a member can sabotage an open pool by seemingly joining it but never sharing its proofs of work. The pool shares its revenue with the attacker, and so each of its participants earns less.

We define and analyze a game where pools use some of their participants to infiltrate other pools and perform such an attack. With any number of pools, no-pool-attacks is not a Nash equilibrium. With two pools, or any number of identical pools, there exists an equilibrium that constitutes a tragedy of the commons where the pools attack one another and all earn less than they would have if none had attacked.

For two pools, the decision whether or not to attack is the miner's dilemma, an instance of the iterative prisoner's dilemma. The game is played daily by the active Bitcoin pools, which apparently choose not to attack. If this balance breaks, the revenue of open pools might diminish, making them unattractive to participants.

In order to win the reward, man
blocks. The system automatically adj
generation, such that one block is a
the blockchain. This means that eac
a block. Although its revenue may b
a miner may have to wait for an e
a block and earn the actual Bitcoins
mining pools, where all members mi
share their revenue whenever one of

Pools are typically implemented
cohort of miners. The pool manager
as a single miner. Instead of gene
outsources the work to the miners.
miners' efforts, the pool manager acc
and estimates each miner's *power* a
which it submits such partial proof
generates a full proof of work, it sen
which publishes this proof of work t
pool manager thus receives the full
distributes it fairly according to its r
the pools are open — they allow any
a public Internet interface.

Incentive Compatibility of Pay Per Last N Shares in Bitcoin Mining Pools

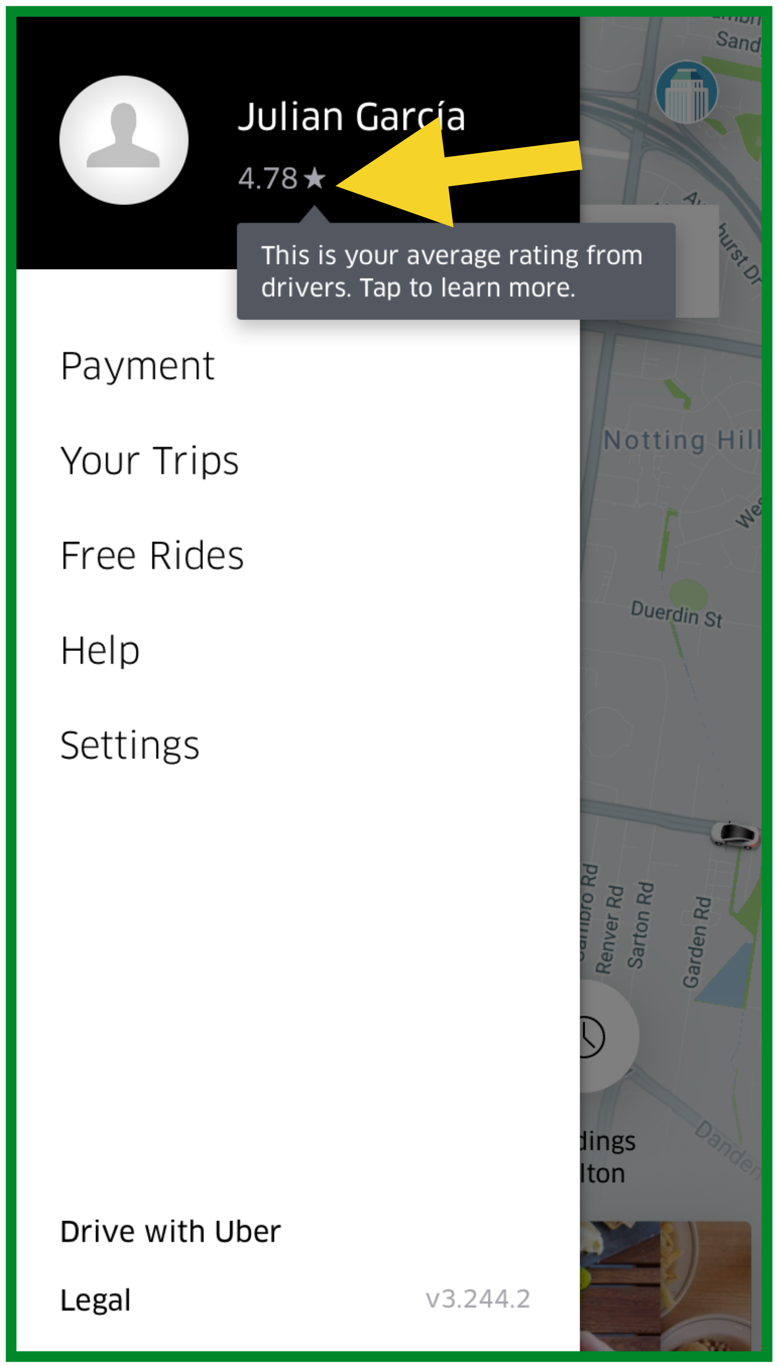
Yevhen Zolotavkin, Julian García(✉), and Carsten Rudolph

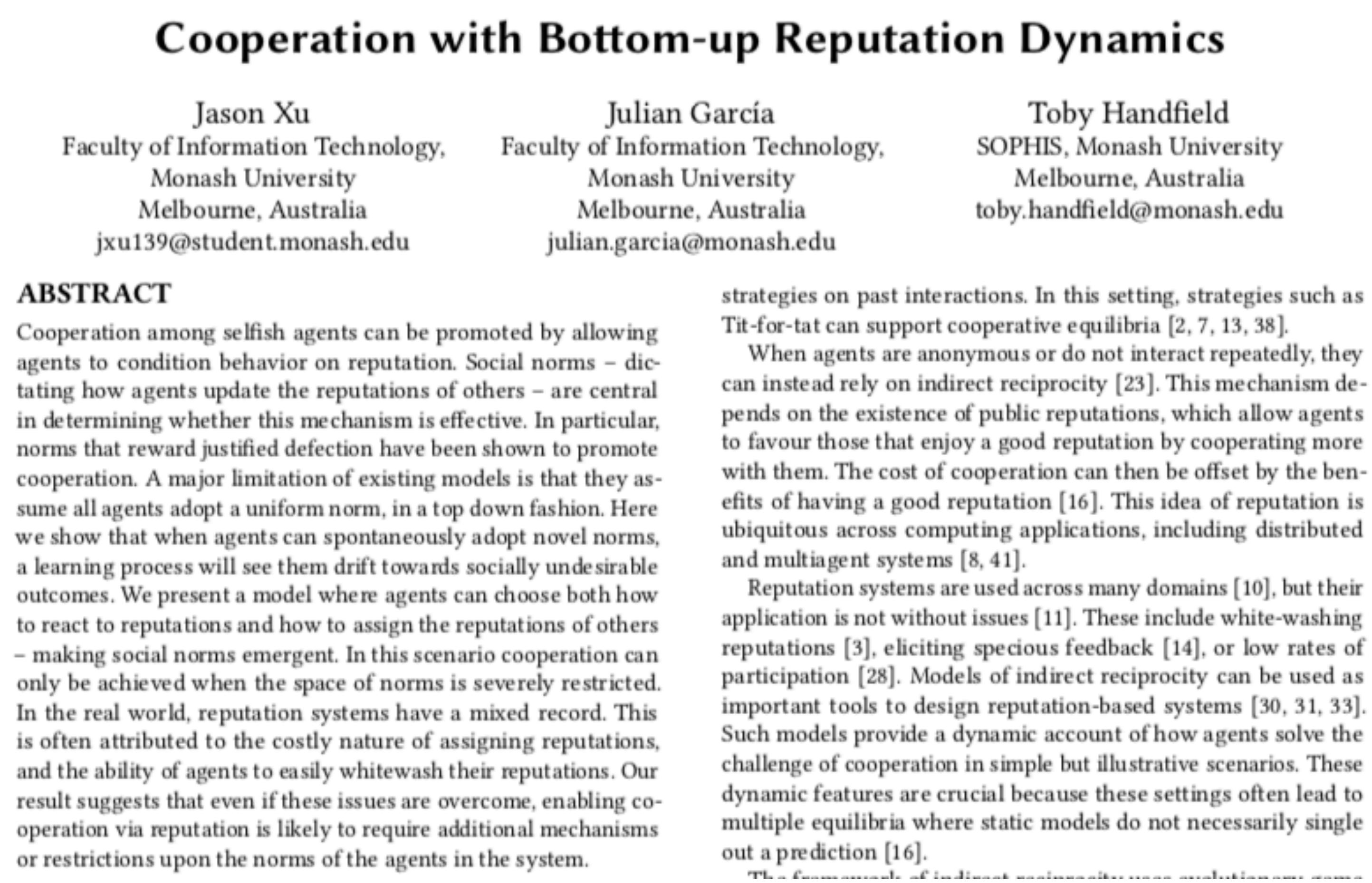
Faculty of IT, Monash University, Clayton, Australia
{yevhen.zolotavkin,julian.garcia,carsten.rudolph}@monash.edu

Abstract. Pay per last N shares (PPLNS) is a popular pool mining reward mechanism on a number of cryptocurrencies, including Bitcoin. In PPLNS pools, miners may stand to benefit by delaying reports of found shares. This attack may entail unfair or inefficient outcomes. We propose a simple but general game theoretical model of delays in PPLNS. We derive conditions for incentive compatible rewards, showing that the power of the most powerful miner determines whether incentives are compatible or not. An efficient algorithm to find Nash equilibria is put forward, and used to show how fairness and efficiency deteriorate with inside-pool inequality. In pools where all players have comparable computational power incentives to deviate from protocol are minor, but gains may be considerable in pools where miner's resources are unequal. We explore how our findings can be applied to ameliorate delay attacks by fitting real-world parameters to our model.

Incentives at play

4.78





Cooperation with Bottom-up Reputation Dynamics

Jason Xu
Faculty of Information Technology,
Monash University
Melbourne, Australia
jxu139@student.monash.edu

Julian García
Faculty of Information Technology,
Monash University
Melbourne, Australia
julian.garcia@monash.edu

Toby Handfield
SOPHIS, Monash University
Melbourne, Australia
toby.handfield@monash.edu

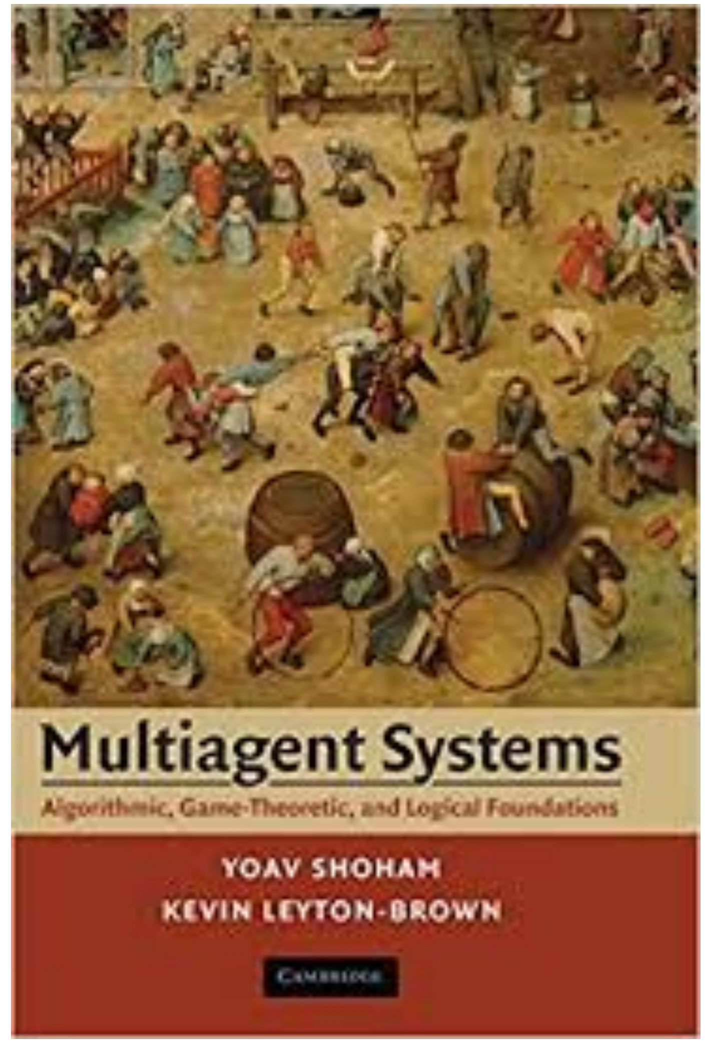
ABSTRACT

Cooperation among selfish agents can be promoted by allowing agents to condition behavior on reputation. Social norms – dictating how agents update the reputations of others – are central in determining whether this mechanism is effective. In particular, norms that reward justified defection have been shown to promote cooperation. A major limitation of existing models is that they assume all agents adopt a uniform norm, in a top down fashion. Here we show that when agents can spontaneously adopt novel norms, a learning process will see them drift towards socially undesirable outcomes. We present a model where agents can choose both how to react to reputations and how to assign the reputations of others – making social norms emergent. In this scenario cooperation can only be achieved when the space of norms is severely restricted. In the real world, reputation systems have a mixed record. This is often attributed to the costly nature of assigning reputations, and the ability of agents to easily whitewash their reputations. Our result suggests that even if these issues are overcome, enabling co-operation via reputation is likely to require additional mechanisms or restrictions upon the norms of the agents in the system.

strategies on past interactions. In this setting, strategies such as Tit-for-tat can support cooperative equilibria [2, 7, 13, 38].

When agents are anonymous or do not interact repeatedly, they can instead rely on indirect reciprocity [23]. This mechanism depends on the existence of public reputations, which allow agents to favour those that enjoy a good reputation by cooperating more with them. The cost of cooperation can then be offset by the benefits of having a good reputation [16]. This idea of reputation is ubiquitous across computing applications, including distributed and multiagent systems [8, 41].

Reputation systems are used across many domains [10], but their application is not without issues [11]. These include white-washing reputations [3], eliciting specious feedback [14], or low rates of participation [28]. Models of indirect reciprocity can be used as important tools to design reputation-based systems [30, 31, 33]. Such models provide a dynamic account of how agents solve the challenge of cooperation in simple but illustrative scenarios. These dynamic features are crucial because these settings often lead to multiple equilibria where static models do not necessarily single out a prediction [16].



Shoham, Yoav, and Kevin Leyton-Brown. **Multiagent systems: Algorithmic, Game-theoretic, and Logical foundations**. Cambridge University Press, 2008.

Is this the end?

Epilogue

- A new perspective on computational modelling and simulation.
- Applications of computing across disciplines.
- <u>Introductory</u> course: so focus is on breadth, not depth.
- We have seen some classic algorithms in the process.

Exam!

- E-exam: be familiar with what will happen. https://www.monash.edu/exams/electronic-exams/about
- Pen and paper. Do not "solve things on the platform." Do it on paper, then transcribe.
- 2 hours long.
- Prepare ahead of time: Re-doing tutes, reviewing labs.
- Ask for help: Forums, Consultations.
- We know it has been a *special* semester.

Thank you.

[Good luck with your exam]