# Lecture 21:
# Solving directional dynamic games for all Markov perfect equilibria

Australian Summer School in Dynamic Structural Econometrics

Fedor Iskhakov, Australian National University
John Rust, Georgetown University
Bertel Schjerning, University of Copenhagen

ANU, Canberra
December 19, 2022

# ROAD MAP

1. Collusion of Australian corrugated fibre packaging (CFP) producers
   - Collusion between Amcor and Visy
   - Bertrand pricing and investment game
   - Solution concept: Markov perfect equilibrium (MPE)
2. Experiment with the model
3. State recursion algorithm
   - Theory of directional dynamic games (DDGs)
4. Recursive lexicographical search (RLS) algorithm
5. Full solution for the leapfrogging game
6. Structural estimation of directional dynamic games with Nested RLS method

# Dynamic Bertrand price competition

**Stochastic dynamic game**

- Two Bertrand competitors, $n = 2$, no entry or exit
- Discrete time, infinite horizon ($t = 1, 2, \ldots, \infty$)
- Each firm maximizes expected discounted profits, common discount factor $\beta \in (0, 1)$
- Each firm has two choices in each period:
  1. Price for the product
  2. Whether or not to buy the state of the art technology

**Static Bertrand price competition in each period**

- Continuum of consumers make static purchase decision
- No switching costs: buy from the lower price supplier

# Cost-reducing investments

**State-of-the-art production cost $c$ process**

- Initial value $c_0$, lowest value 0: $0 \leq c \leq c_0$
- Discretized with $n$ points
- Follows exogenous Markov process and only improves
- Markov transition probability $\pi(c_{t+1}|c_t)$
  $\pi(c_{t+1}|c_t) = 0$ if $c_{t+1} > c_t$

**Investment choice: binary**

- Investment cost of $K(c)$ to obtain marginal cost $c$
- One period construction time: production with technology obtained at $t$ starts at $t+1$

# State space and information structure

**Common knowledge**

- State of the game: cost structure $(c_1, c_2, c)$
- State space is $S = (c_1, c_2, c) \subset R^3$: $c_1 \geq c$, $c_2 \geq c$
- Actions are observable

**Private information**

- In each period each firm incurs additive costs (benefits) from not investing and investing $\eta\epsilon_{i,I}$ and $\eta\epsilon_{i,N}$
- $\epsilon_{i,I}$ and $\epsilon_{i,N}$ are extreme value distributed, independent across choice, time and firms
- $\eta \geq 0$ is a scaling parameter
- Investment choice probabilities have logit form for $\eta > 0$

# Timing of moves

**Pricing decisions are made simultaneously**

Expected one period profit of firm $i$ from Bertrand game ($j \neq i$)

$$r_i(c_1, c_2) = \begin{cases} 0 & \text{if } c_i \geq c_j \\ c_j - c_i & \text{if } c_i < c_j \end{cases}$$

**Two versions regarding investment decisions**

1. Simultaneous moves:
   - Investment decisions are made simultaneously
2. Alternating moves:
   - The "right to move" state variable $m \in \{1, 2\}$,
   - When $m = i$, only firm $i$ can make a cost reducing investment
   - $m$ follows an own Markov process (deterministic alternation as a special case).

# Actions and behavior strategies

**Two choices in each period**

- $p_i(c_1, c_2, c) = \max(c_1, c_2)$ – Bertrand pricing decision
- $P_i^I(c_1, c_2, c)$ – probability of firm $i$ to invest in state-of-the-art production technology
  $P_i^N(c_1, c_2, c) = 1 - P_i^I(c_1, c_2, c)$ – probability not to invest

**Strategy profile**

- $\sigma = (\sigma_1, \sigma_2)$ – pair of Markovian *behavior* strategies
  $\sigma_i = \Big(p_i(c_1, c_2, c), P_i^I(c_1, c_2, c)\Big) \in \mathbb{R}_+ \times [0, 1]$
- Pure strategies included as special case

# Definition of Markov Perfect Equilibium

## Definition (Markov perfect equilibrium (MPE))

MPE of Bertrand investment stochastic game is a pair of

- strategy profile $\sigma^* = (\sigma_1^*, \sigma_2^*)$, and
- pair of *value functions* $V(s) = \Big(V_1(s), V_2(s)\Big)$, $V_i : S \to R$,

such that

1. Bellman equations (below) are satisfied for each firm, and
2. strategies $\sigma_1^*$ and $\sigma_2^*$ constitute mutual best responses, and assign positive probabilities only to the actions in the set of maximizers of the Bellman equations.

# Bellman equations, firm $i = 1$, simultaneous moves

$$V_i(c_1, c_2, c) = \max\left[v_i^I(c_1, c_2, c) + \eta\epsilon_{i,I}, v_i^N(c_1, c_2, c) + \eta\epsilon_{i,N}\right]$$

$$\begin{aligned} v_i^N(c_1, c_2, c) &= r^i(c_1, c_2) + \beta EV_i(c_1, c_2, c|N) \\ v_i^I(c_1, c_2, c) &= r^i(c_1, c_2) - K(c) + \beta EV_i(c_1, c_2, c|I) \end{aligned}$$

With extreme value shocks, the investment probability is

$$P_i^I(c_1, c_2, c) = \frac{\exp\{v_I^I(c_1, c_2, c)/\eta\}}{\exp\{v_i^I(c_1, c_2, c)/\eta\} + \exp\{v_i^N(c_1, c_2, c)/\eta\}}$$

# Bellman equations, firm $i = 1$, simultaneous moves

The expected values are given by

$$
\begin{aligned}
EV_i(c_1, c_2, c|N) &= \int_0^c \left[ P_j^I(c_1, c_2, c) H_i(c_1, c, c') + \right. \\
&\qquad \left. P_j^N(c_1, c_2, c) H_i(c_1, c_2, c') \right] \pi(dc'|c) \\
EV_i(c_1, c_2, c|I) &= \int_0^c \left[ P_j^I(c_1, c_2, c) H_i(c, c, c') + \right. \\
&\qquad \left. P_j^N(c_1, c_2, c) H_i(c, c_2, c') \right] \pi(dc'|c)
\end{aligned}
$$

where

$$H_i(c_1, c_2, c) = \eta \log \left[ \exp\left(v_i^N(c_1, c_2, c)/\eta\right) + \exp\left(v_i^I(c_1, c_2, c)/\eta\right) \right]$$

is the "smoothed max" or logsum function

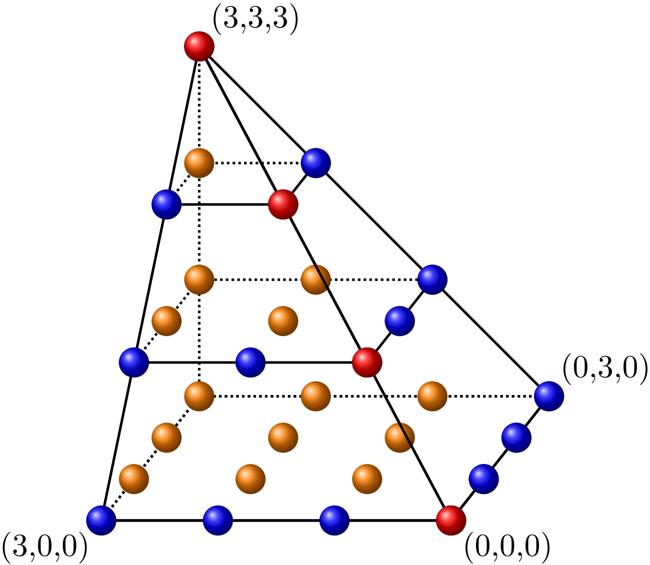
# Solving and estimating directional dynamic games: Bertrand pricing and investment (leapfrogging) game

# ROAD MAP

1. Collusion of Australian corrugated fibre packaging (CFP) producers
   - Collusion between Amcor and Visy
   - Bertrand pricing and investment game
   - Solution concept: Markov perfect equilibrium (MPE)
2. Experiment with the model
3. State recursion algorithm
   - Theory of directional dynamic games (DDGs)
4. Recursive lexicographical search (RLS) algorithm
5. Full solution for the leapfrogging game
6. Structural estimation of directional dynamic games with Nested RLS method

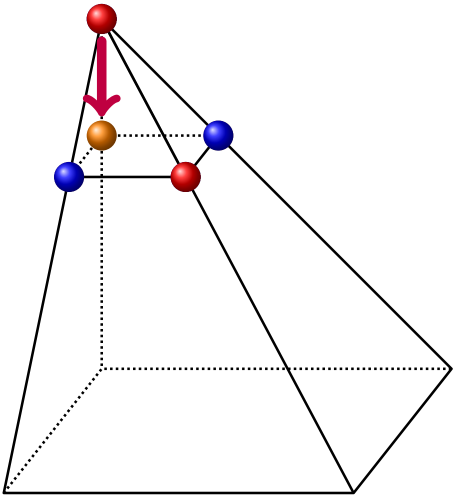
# Discretized state space = a "quarter pyramid"

$S = \{(c_1, c_2, c) | c_1 \geq c,\ c_2 \geq c, c \in [0,3]\},\ n = 4$



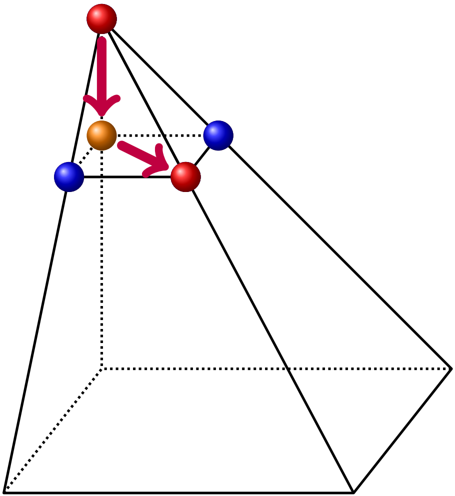
# Game dynamics: example

The game starts at the apex, as some point technology improves



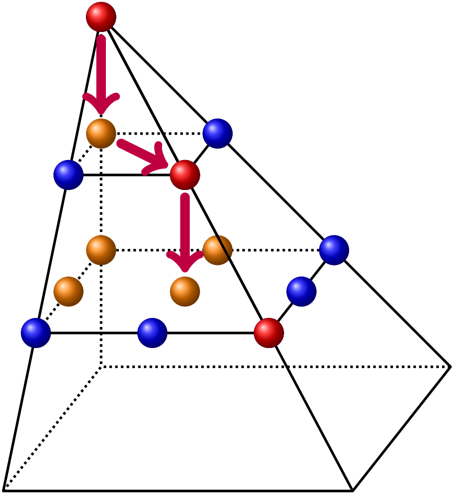
# Game dynamics: example

Both firms buy new technology $c = 2 \rightsquigarrow (c_1, c_2, c) = (2, 2, 2)$



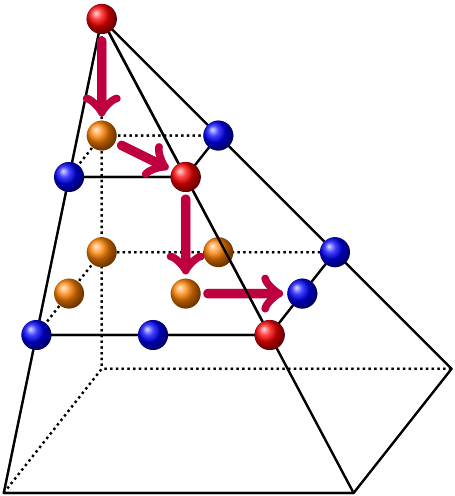
# Game dynamics: example

State-of-the-art technology becomes $c = 1 \rightsquigarrow (c_1, c_2, c) = (2, 2, 1)$



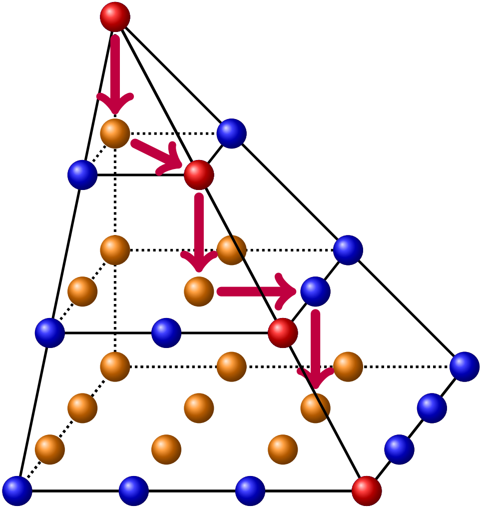
# Game dynamics: example

Firm 1 invests and becomes cost leader $\rightsquigarrow (c_1, c_2, c) = (1, 2, 1)$



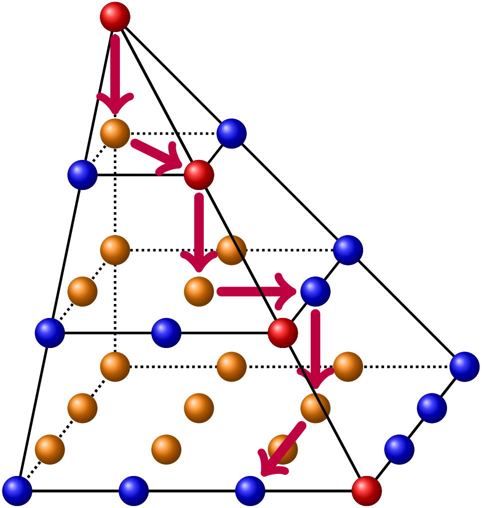
# Game dynamics: example

State-of-the-art technology becomes $c = 0 \rightsquigarrow (c_1, c_2, c) = (1, 2, 0)$



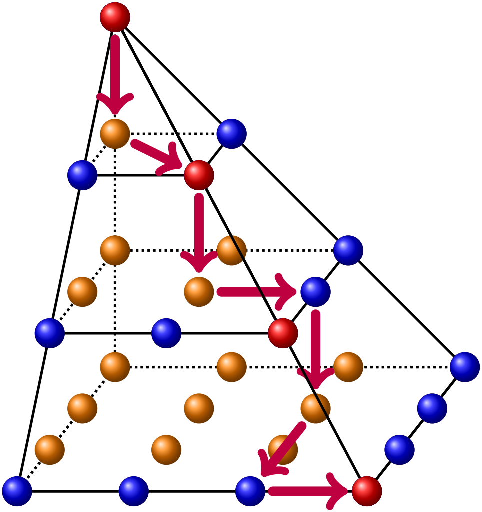
# Game dynamics: example

Firm 2 leapfrogs firm 1 to become new cost leader $\rightsquigarrow (c_1, c_2, c) = (1, 0, 0)$



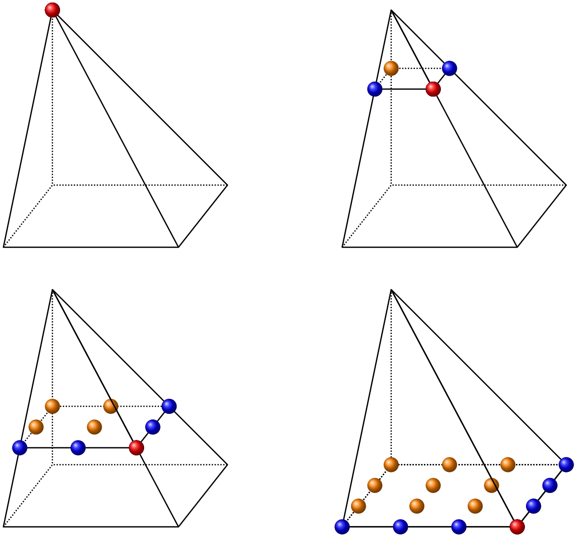
# Game dynamics: example

Firm 1 invests, and the game reaches terminal state $\rightsquigarrow (c_1, c_2, c) = (0, 0, 0)$



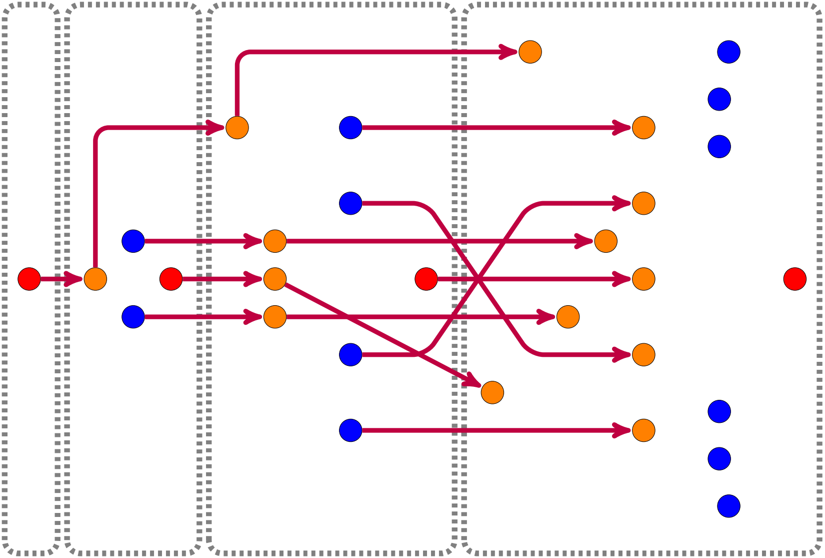
# Transitions due to technological progress

As $c$ decreases, the game falls through the layers of the pyramid



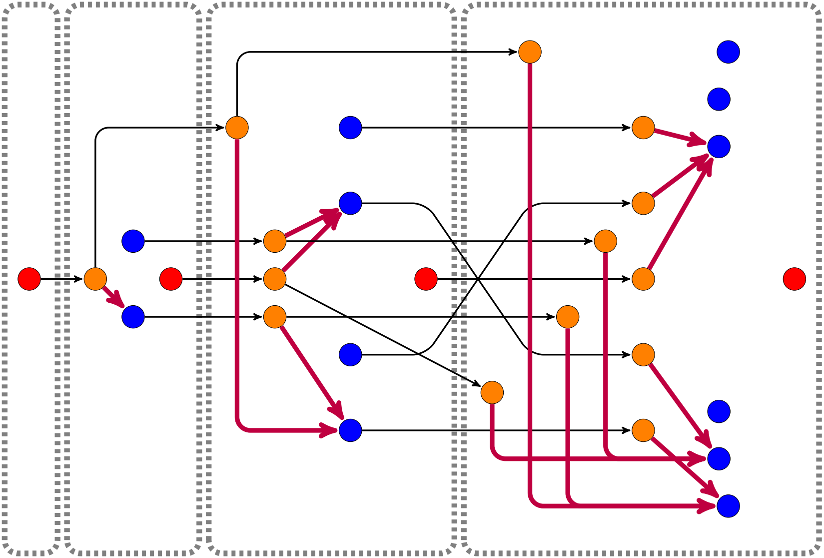
# Transitions due to technological progress

## As $c$ decreases, the game falls through the layers of the pyramid



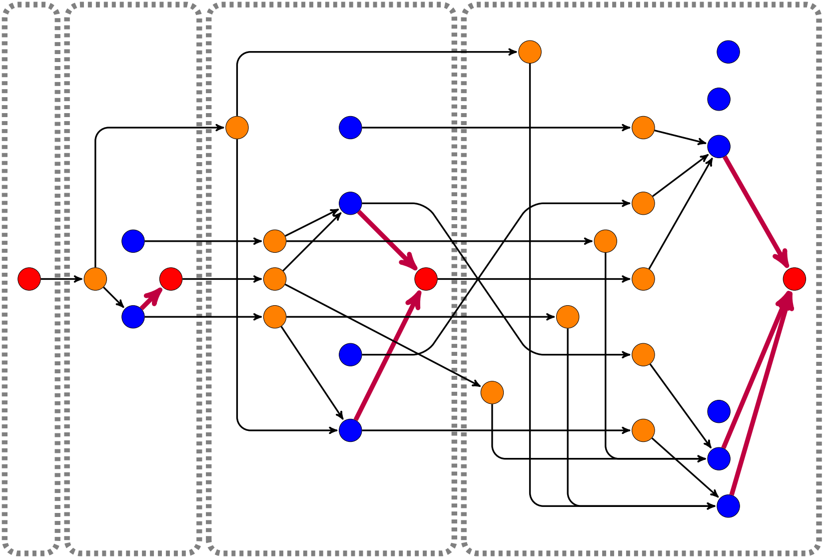
# Strategy-specific partial order on $S$

## Strategy $\sigma_1$ of firm 1: invest at all interior points



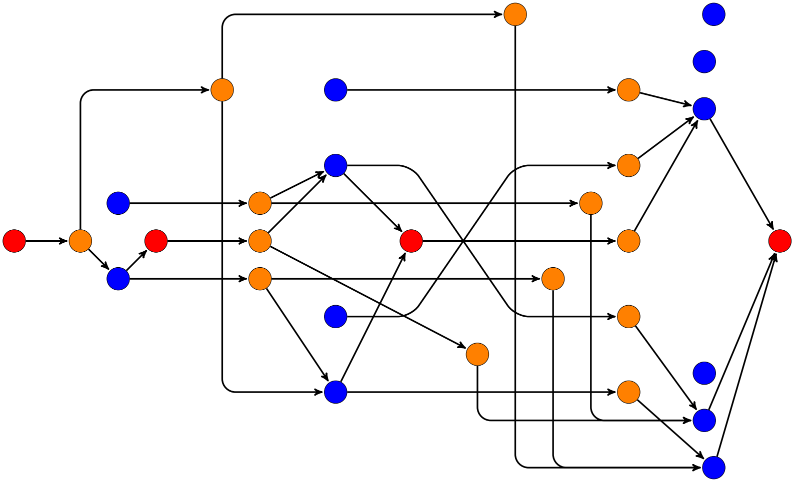
# Strategy-specific partial order on $S$

## Strategy $\sigma_2$ of firm 2: invest at all edge points



# Strategy-specific partial order on $S$

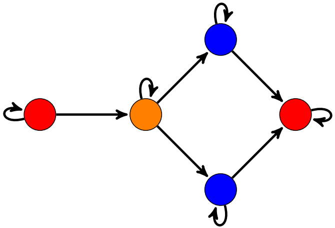
Strategy $\sigma = (\sigma_1, \sigma_2)$ of both firms

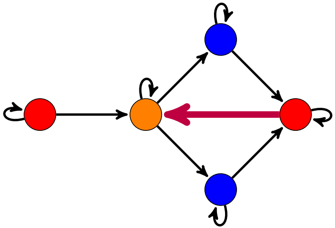


# No loop (anti-cycling) condition

## Hypothetical strategy profile inducing cycles

Self-loops appear when the game remains in the same state for two or more consecutive periods of time



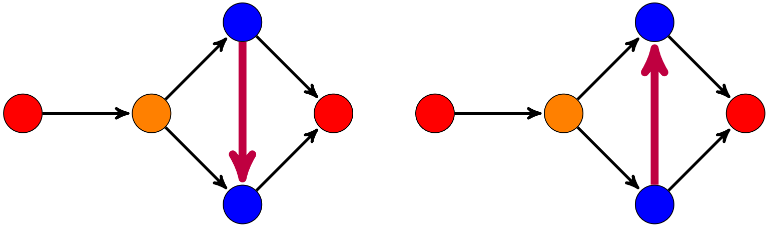


But loops between different states are not allowed

# Consistency of strategy specific partial orders

## Two hypothetical inconsistent strategies

Two strategies that induce opposite transitions are inconsistent



Note that in both cases the no-loop condition is satisfied

# Definition of the Dynamic Directional Games

## Definition (Dynamic Directional Games, DDG)

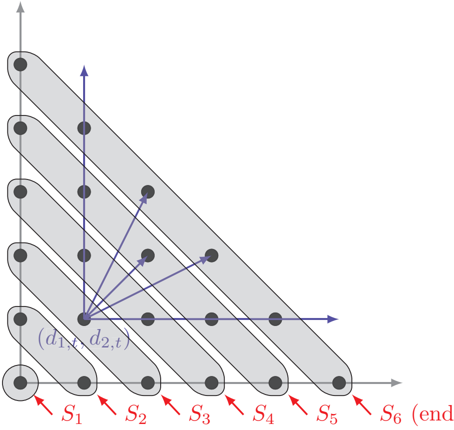
Finite state Markovian stochastic game is a DDG if it holds:

1. Every feasible Markovian strategy $\sigma$ satisfies the no loop condition.
2. Every pair of feasible Markovian strategies $\sigma$ and $\sigma'$ induce consistent partial orders on the state space.

# Example of DDG: Martket tipping game

Dubè, Hitsch and Chintagunta, 2010 *Marketing Science*
Tipping and Concentration in Markets with Indirect Network Effects

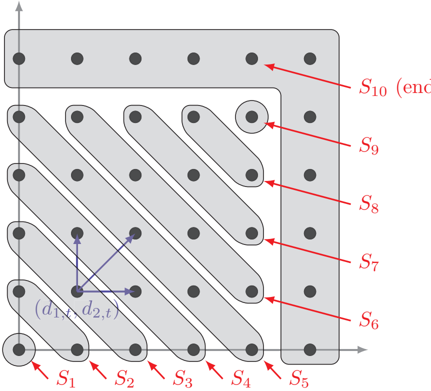
- Two competing gaming **platforms** (Xbox vs. Playstation)
- Number of games for each console depends on market share
- Consumers choose product they believe will win the war of the standards, or delay purchase
- Brand choices are absorbing: adoption base can only increase



# Example of DDG: Patent race game

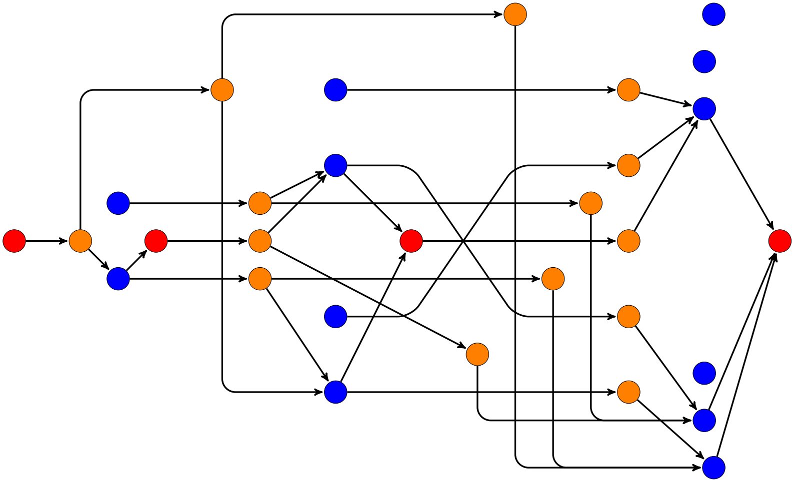
Judd, Schmedders and Yeltekin, 2012 *IER*
Optimal Rules for Patent Races

- Two firms are engaged in the $N$-steps race to acquire a patent
- Probability a step depends on the chosen R&D investment
- Only forward steps are possible, no "forgetting"
- Used state dependence structure to speed up computation of MPE by solving non-linear systems



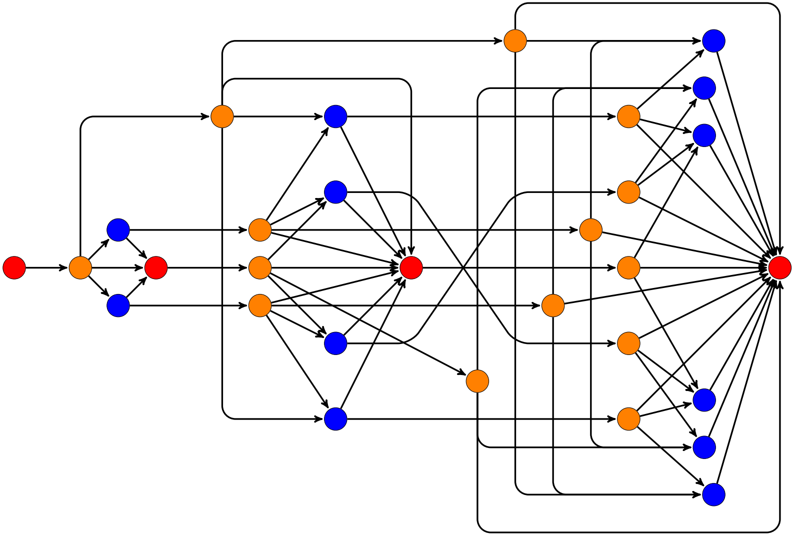
# Strategy-specific partial order on $S$

Strategy $\sigma = (\sigma_1, \sigma_2)$ of both firms

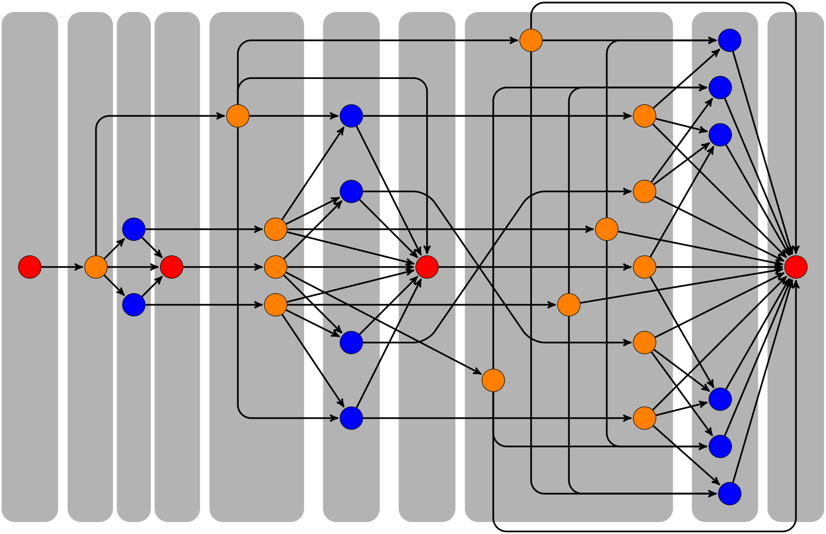


# Strategy independent partial order on $S$

## Coarsest common refinement of partial orders induced by all strategies

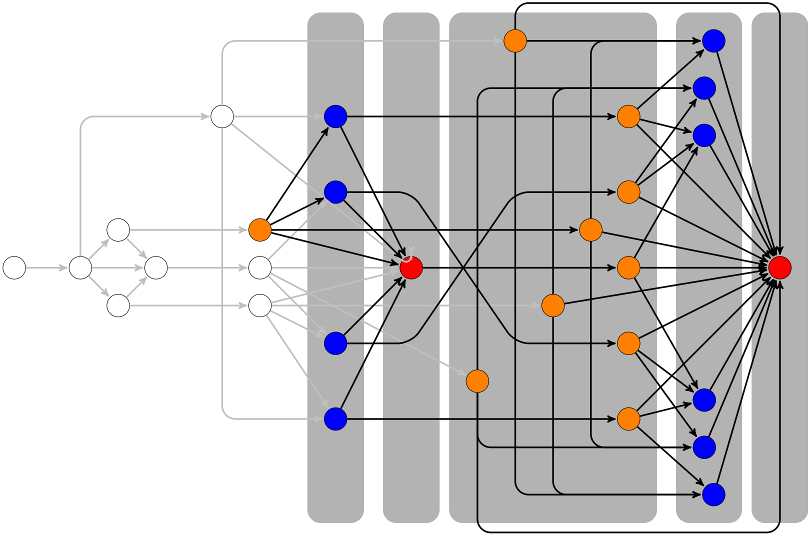


# Total order on the set of stages



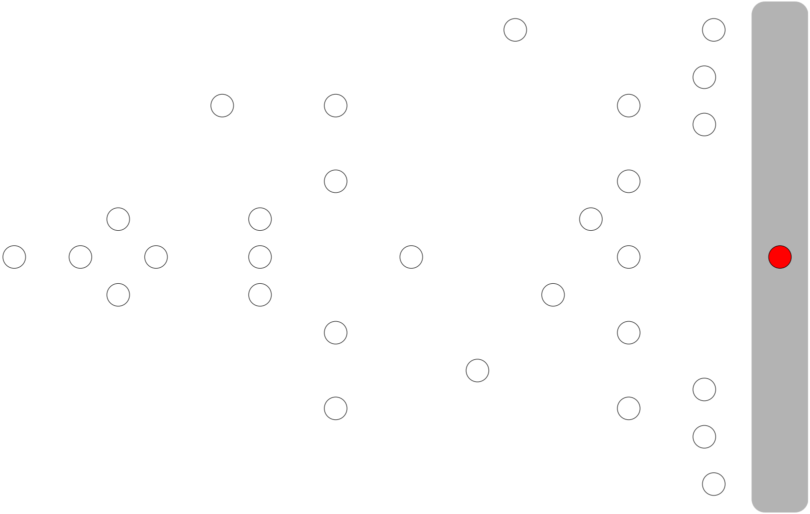
# Subgames of DDG and continuation strategies

## Subgames and continuation strategies



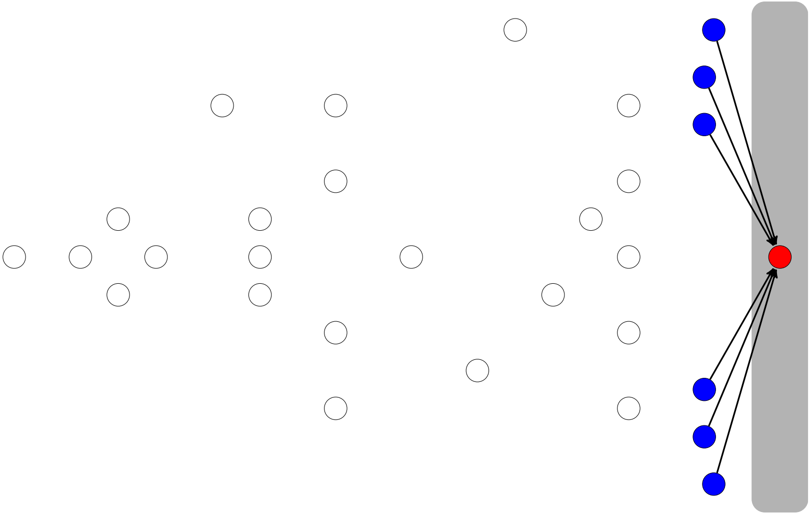
# State recursion algorithm

## Backward induction on stages of DDG



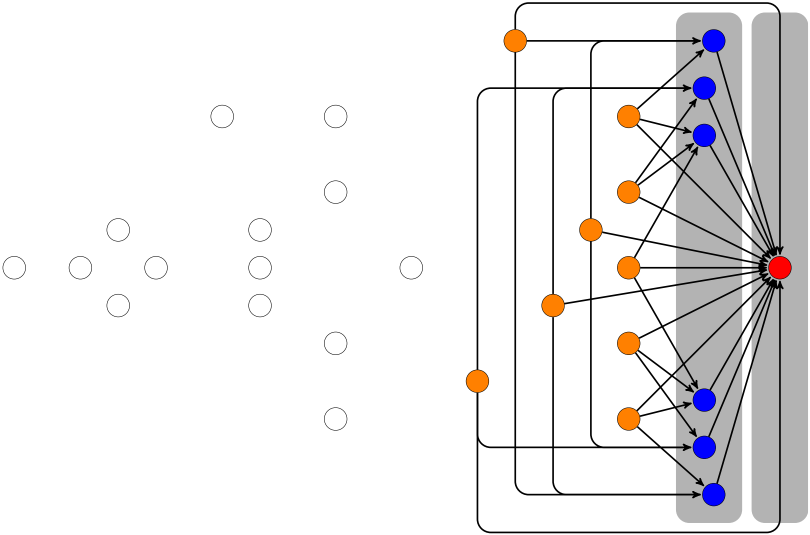
# State recursion algorithm

## Backward induction on stages of DDG



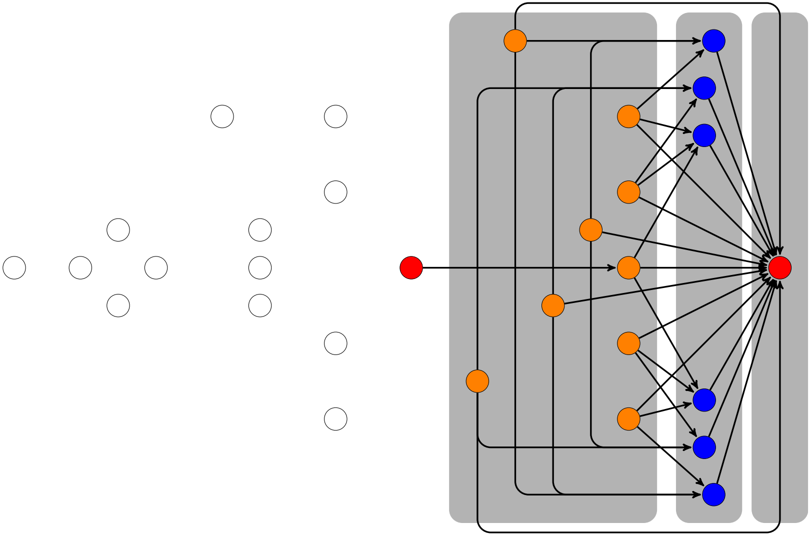
# State recursion algorithm

## Backward induction on stages of DDG

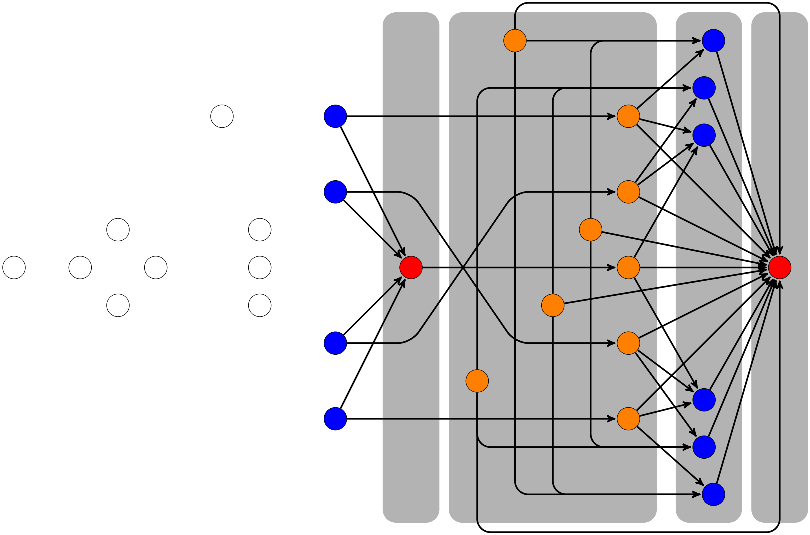


# State recursion algorithm

## Backward induction on stages of DDG

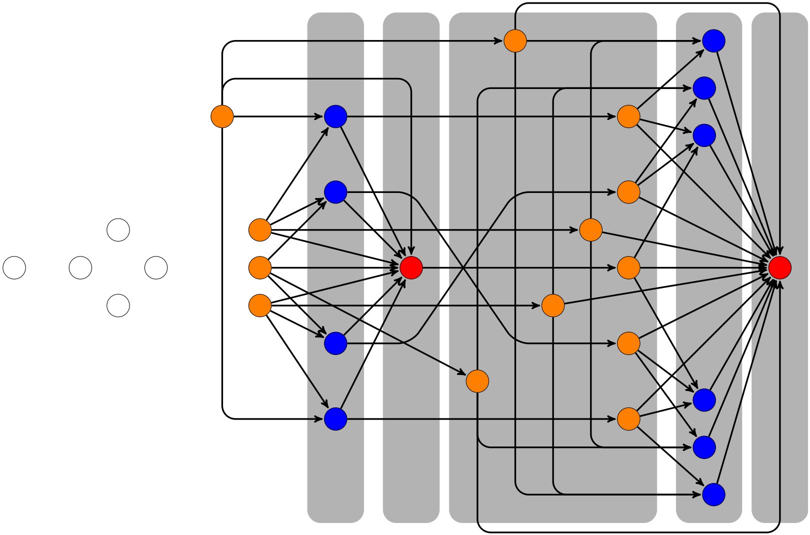


# State recursion algorithm
## Backward induction on stages of DDG



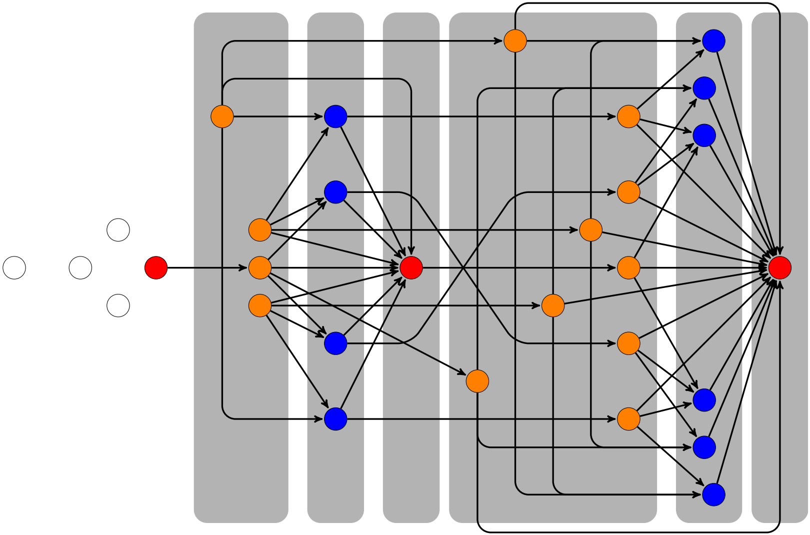
# State recursion algorithm

## Backward induction on stages of DDG



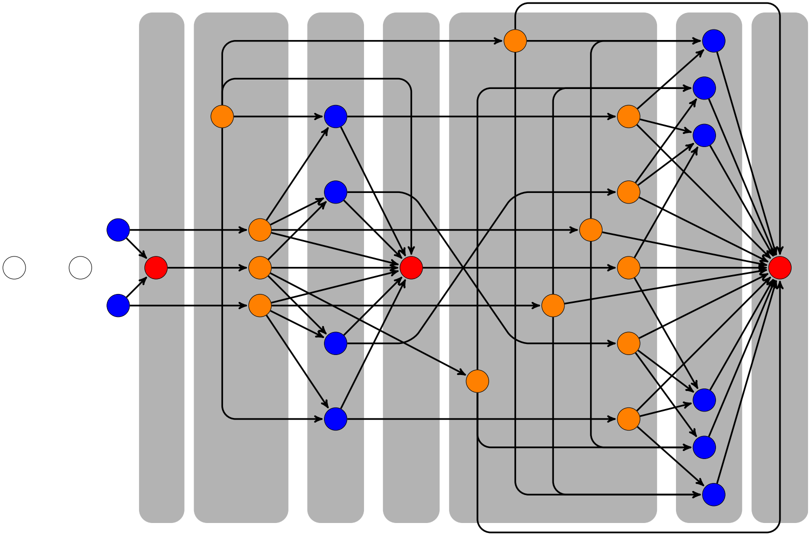
# State recursion algorithm

## Backward induction on stages of DDG



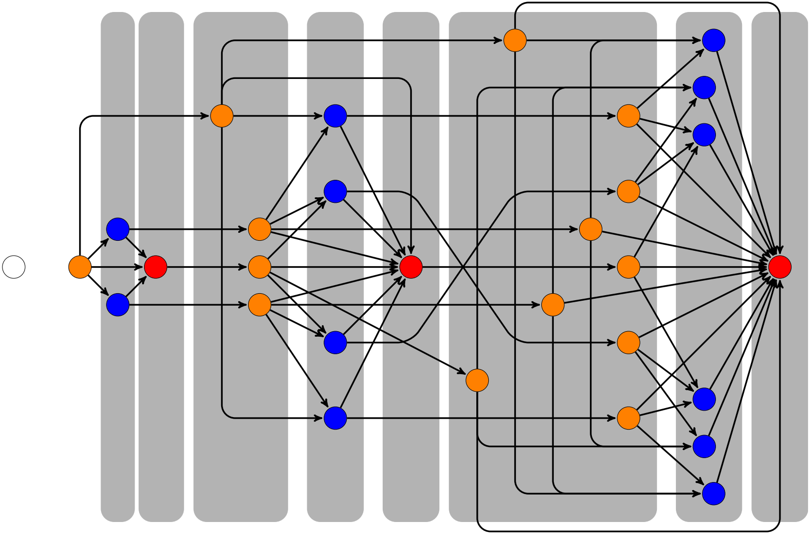
# State recursion algorithm

## Backward induction on stages of DDG



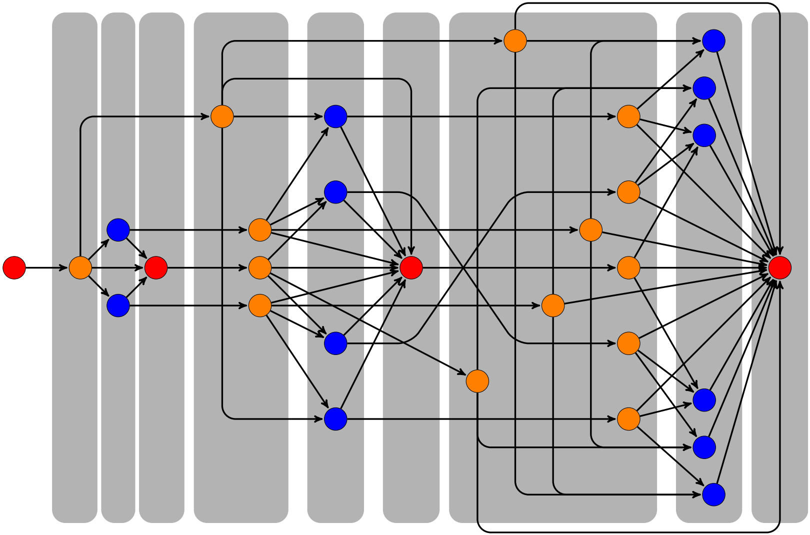
# State recursion algorithm

## Backward induction on stages of DDG



# State recursion algorithm

## Backward induction on stages of DDG



# State Recursion versus Backward Induction

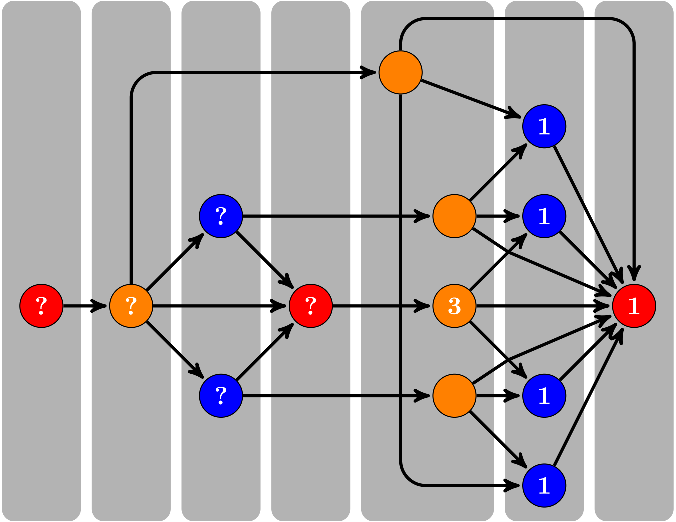
- State recursion – generalization of backward induction
- Runs on state space instead of time periods
- Time ($t$) evolves as $t \to t + 1$ with probability 1
- For stages of state space ($\tau$) transitions are stochastic and not necessarily sequential
- Yet, probability of going $\tau \to \tau'$ is zero when $\tau' < \tau$
- With multiplicity, state recursion is performed conditional of a particular equilibrium selection rule (ESR)

# ROAD MAP

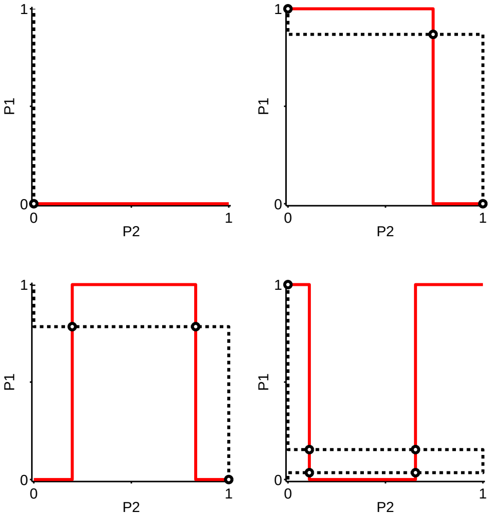
1. Collusion of Australian corrugated fibre packaging (CFP) producers
    - Collusion between Amcor and Visy
    - Bertrand pricing and investment game
    - Solution concept: Markov perfect equilibrium (MPE)
2. Experiment with the model
3. State recursion algorithm
    - Theory of directional dynamic games (DDGs)
4. Recursive lexicographical search (RLS) algorithm
5. Full solution for the leapfrogging game
6. Structural estimation of directional dynamic games with Nested RLS method

# Multiplicity of stage equibiria

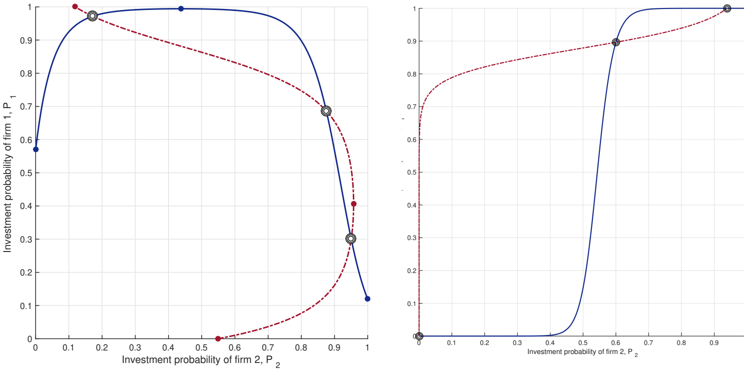
## Number of equilibria in the higher stages depends on the selected equilibria



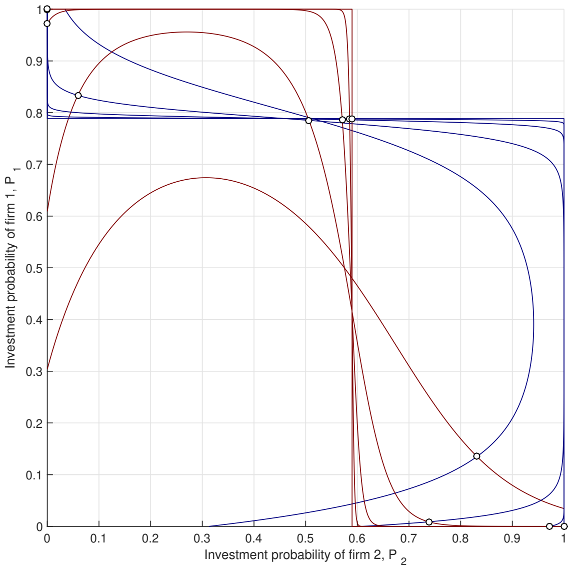
# Best response correspondences for $\eta = 0$



# Best response functions for $\eta > 0$



# Best response functions as $\eta \to 0$



# Recursive Lexicographic Search Algorithm

Building blocks of RLS algorithm:

1. State recursion algorithm solves the game conditional on equilibrium selection rule (ESR)
2. RLS algorithm efficiently cycles through all feasible ESRs

**Challenge:**

- Choice of a particular MPE for any stage game at any stage
- may alter the set and even the number of stage equilibria at earlier stages

Need to find feasible ESRs

- ESR = string of digits that index the selected stage equilibrium in each point

# All possible ESR strings in lexicographic order

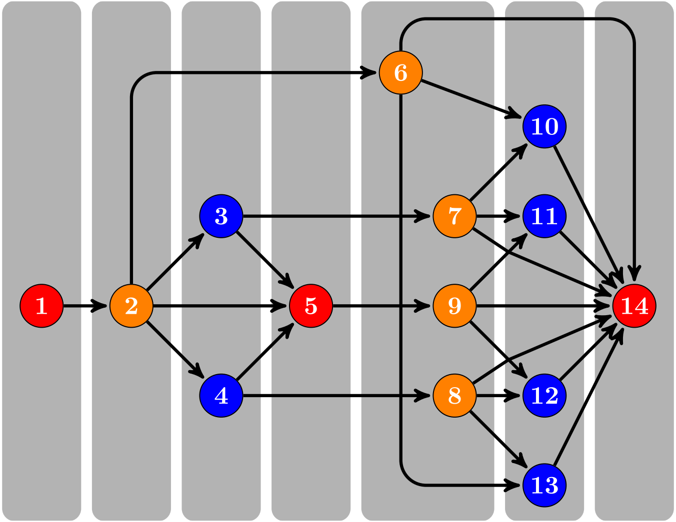
- Bound the maximum number of equilibria in each stage by $K = 3$

| | c | e | e | e | e | i | i | i | i | c | e | e | i | c |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| ESR string | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 |
| Lexicograph | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 |
| | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 2 |
| | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 0 |
| | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 1 |
| | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 2 |
| | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 2 | 0 |
| | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 2 | 1 |
| | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 2 | 2 |
| | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 0 |
| | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 1 |
| | | | | | | | | | | | | | | ... |
| | 2 | 2 | 2 | 2 | 2 | 2 | 2 | 2 | 2 | 2 | 2 | 2 | 2 | 0 |
| | 2 | 2 | 2 | 2 | 2 | 2 | 2 | 2 | 2 | 2 | 2 | 2 | 2 | 1 |
| | 2 | 2 | 2 | 2 | 2 | 2 | 2 | 2 | 2 | 2 | 2 | 2 | 2 | 2 |

4,782,969

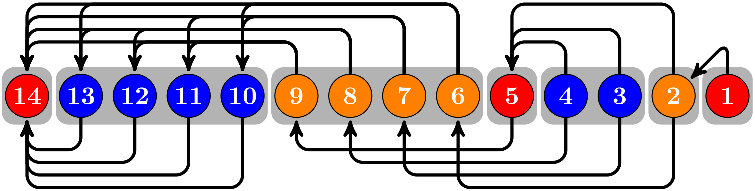
# Indexing of points in the state space

## Lower index for dependent points, highest for terminal stage

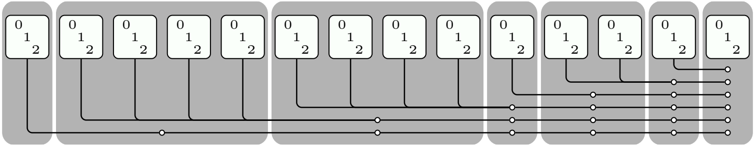


# Preserving stage order in ESR strings

## Formalization of the ESR as stings of digits




- Digits arranged to preserve the dependence structure



# Recalculation of feasibility condition for new ESR

## Avoid recalculation of subgames

| | | | | | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | c | e | e | e | e | i | i | i | i | c | e | e | i | c | |
| ESR string | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | |
| | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | always admissible |
| Nr of eqb | 1 | 1 | 1 | 1 | 1 | 3 | 3 | 3 | 3 | 1 | 1 | 1 | **3** | 1 | |
| | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | **1** | |
| | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | **2** | |
| | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | **1** | 0 | admissible, solve |
| | 1 | 1 | 1 | 1 | 1 | 3 | 3 | 3 | 3 | 1 | 1 | 1 | 3 | * | |

No changes in the solution of the game including the number of stage equilibria

Might have change

# Jumping over blocks of infeasibles ESRs

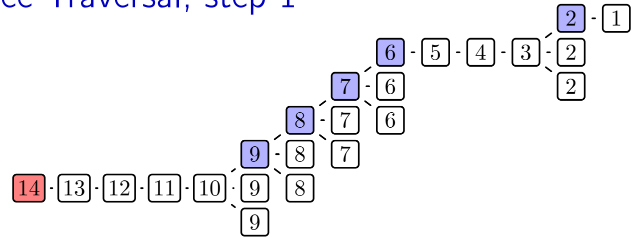
| | c | e | e | e | e | i | i | i | i | c | e | e | i | c | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| ESR string | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | Iteration: |
| | **0** | **0** | **0** | **0** | **0** | **0** | **0** | **0** | **0** | **0** | **0** | **0** | **0** | **0** | 1 |
| | 1 | 1 | 1 | 1 | 1 | 3 | 3 | 3 | 3 | 1 | 1 | 1 | 3 | 1 | 1a |
| | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | |
| | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 2 | |
| | **0** | **0** | **0** | **0** | **0** | **0** | **0** | **0** | **0** | **0** | **0** | **0** | **1** | **0** | 2 |
| | 1 | 1 | 1 | 1 | 1 | 3 | 3 | 3 | 3 | 1 | 1 | 1 | 3 | 1 | 2a |
| | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 1 | |
| | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 2 | |
| | **0** | **0** | **0** | **0** | **0** | **0** | **0** | **0** | **0** | **0** | **0** | **0** | **2** | **0** | 3 |
| | 1 | 1 | 1 | 1 | 1 | 3 | 3 | 3 | 3 | 1 | 1 | 1 | 3 | 1 | |
| | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 2 | 1 | 3a |
| | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 2 | 2 | |
| | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 3b |
| | | | | | | | | | | | | | | … | |
| | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 3c |
| | | | | | | | | | | | | | | … | |
| | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 3d |
| | | | | | | | | | | | | | | … | |
| | **0** | **0** | **0** | **0** | **0** | **0** | **0** | **0** | **1** | **0** | **0** | **0** | **0** | **0** | 4 |
| | | | | | | | | | | | | | | … | |

# Recursive Lexicographic Search (RLS) Algorithm

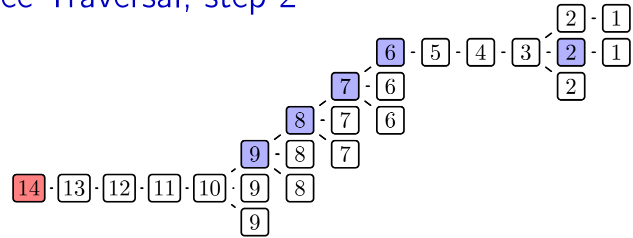
1. Set ESR $= (0, \ldots, 0)$
2. Run State Recursion using the current ESR
3. Save the number of equilibria in every stage game as *ne*(ESR)
4. Add 1 to the ESR in bases *ne*(ESR) to obtain new feasible ESR
5. Stopping rule: successor function exceeds the maximum number with given number of digits
6. Return to step 2

**RLS = tree traversal!**

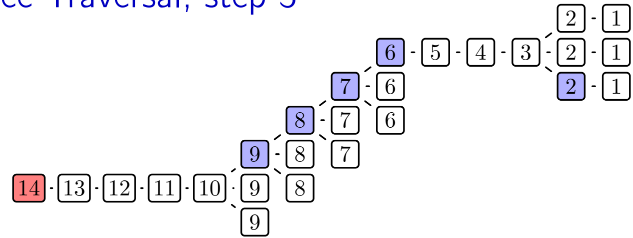
# RLS Tree Traversal, step 1



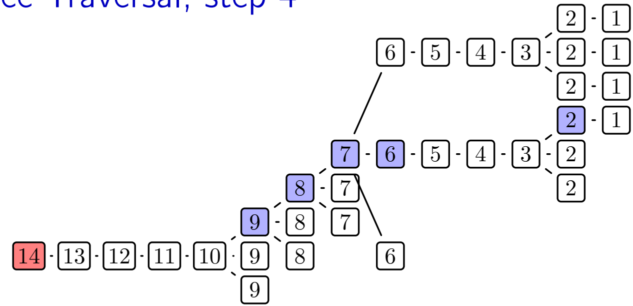
# RLS Tree Traversal, step 2



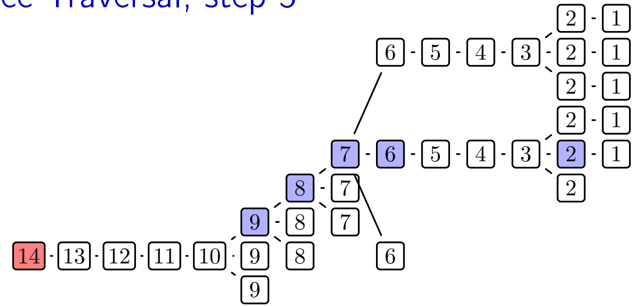
# RLS Tree Traversal, step 3



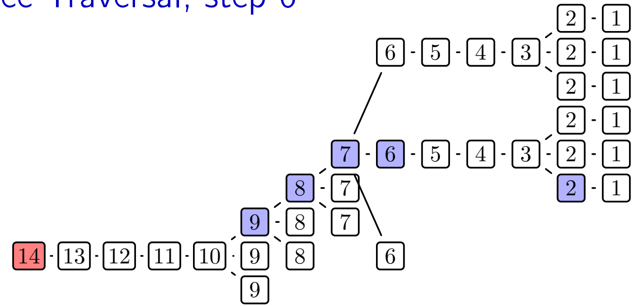
# RLS Tree Traversal, step 4



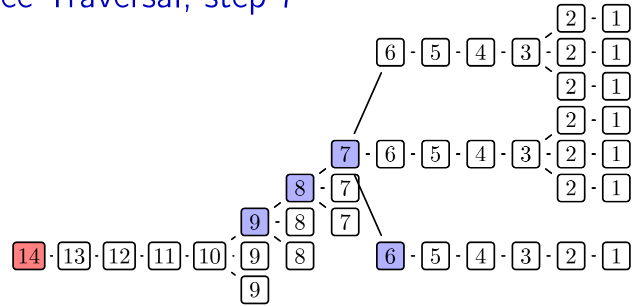
# RLS Tree Traversal, step 5



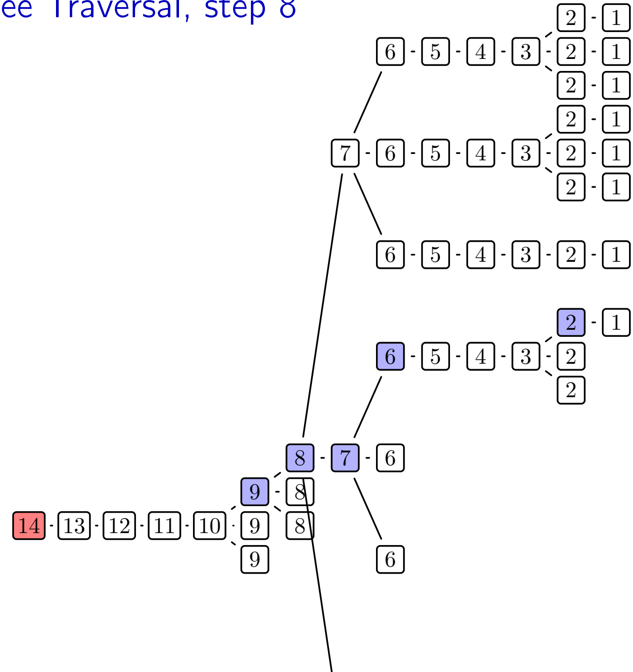
# RLS Tree Traversal, step 6



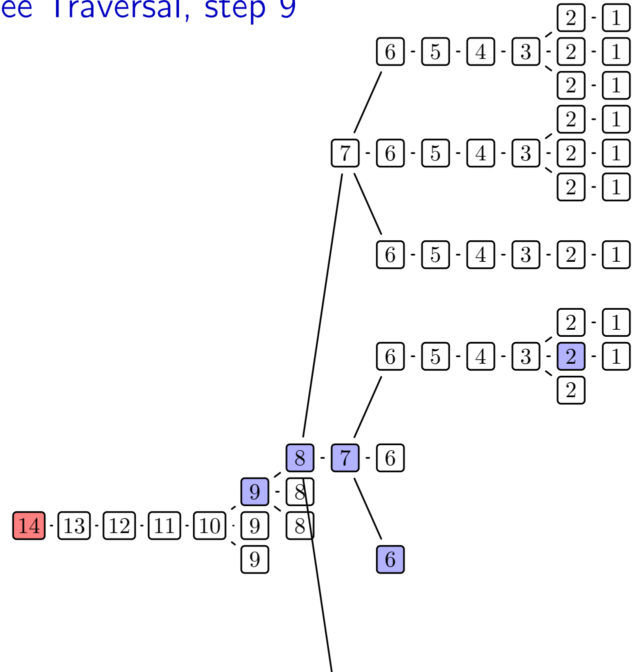
# RLS Tree Traversal, step 7



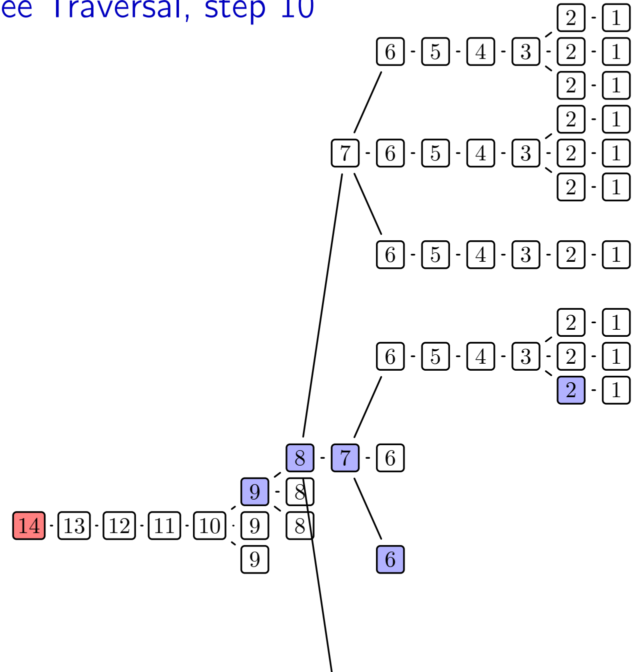
# RLS Tree Traversal, step 8



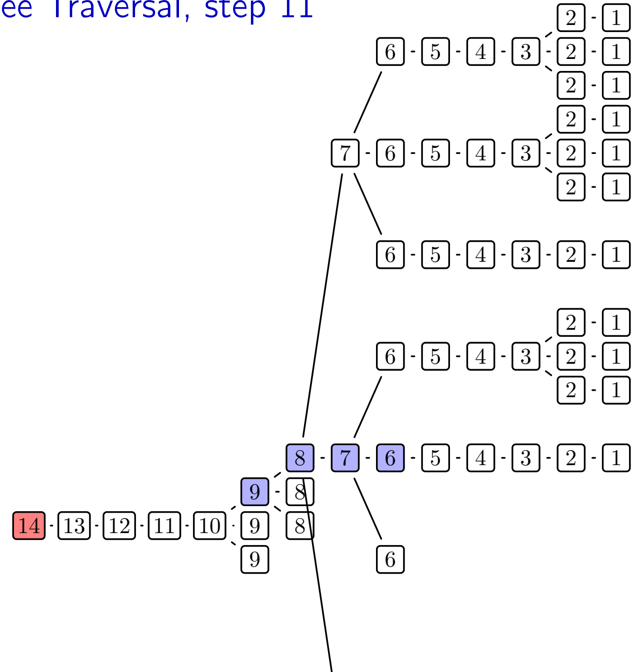
# RLS Tree Traversal, step 9



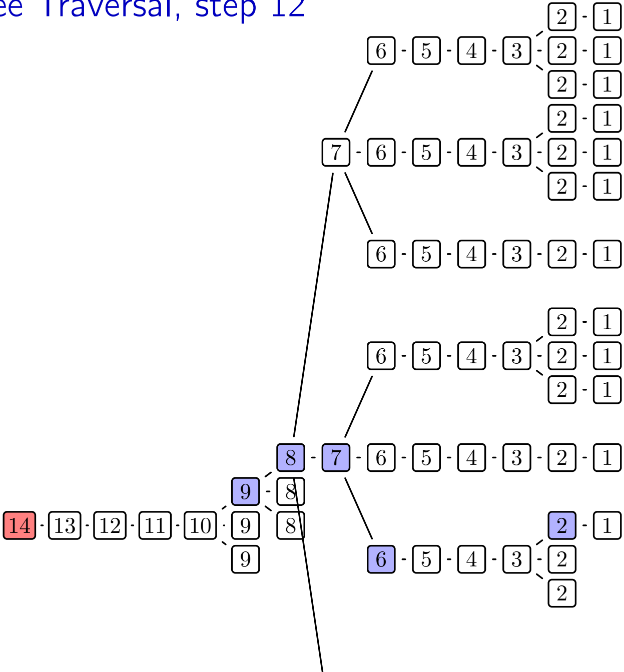
# RLS Tree Traversal, step 10



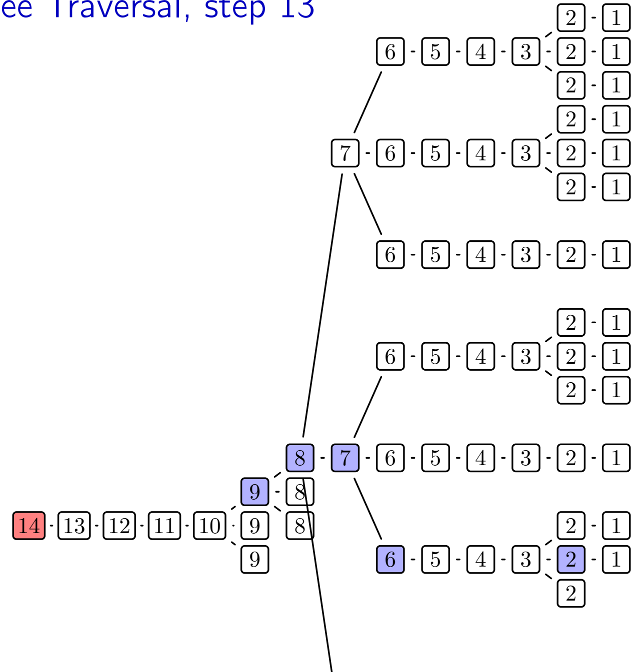
# RLS Tree Traversal, step 11



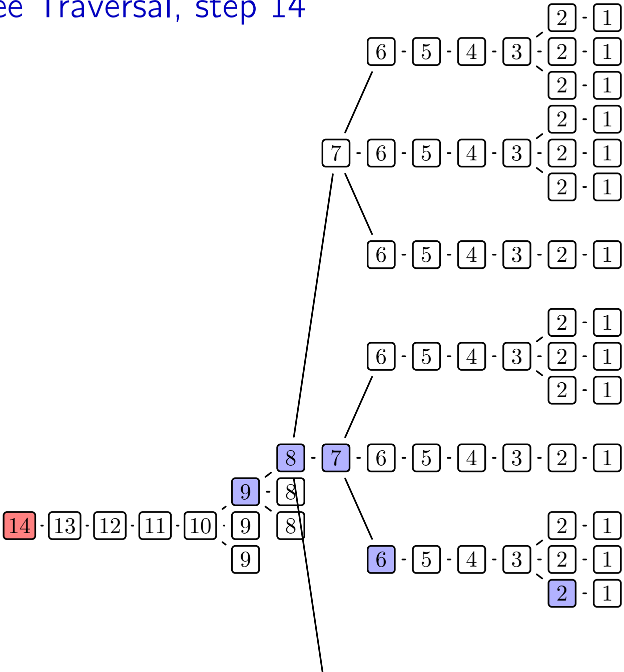
# RLS Tree Traversal, step 12



# RLS Tree Traversal, step 13



# RLS Tree Traversal, step 14



# Main result of the RLS Algorithm

## Theorem (Decomposition theorem, strong)

*Assume there exists an algorithm that can find all MPE of every stage game of the DDG, and that the number of these equilibria is finite in every stage game.*
*Then the RLS algorithm finds all MPE of the DDG in a finite number of steps, which equals the total number of MPE.*

Iskhakov, Rust and Schjerning, 2016, ReStud
"Recursive lexicographical search: Finding all markov perfect equilibria of finite state directional dynamic games."

# Main result of the RLS Algorithm

## Theorem (Decomposition theorem, weak)

*Assume there exists an algorithm that can find at least one MPE of every stage game of the DDG, and that the number of these equilibria is finite in every stage game.*
*Then the RLS algorithm finds some (at least one) MPE of the DDG in a finite number of steps, which does not exceed the total number of MPE.*

# RLS algorithm: running times
## $K = 3$

| Simultaneous moves | $n = 3$ | $n = 4$ |
|---|---|---|
| Upper bound on number of MPE | 4,782,969 | 3,948,865,611 |
| Actual number of equilibria | 127 | 46,707 |
| Time used | 0.008 sec. | 0.334 sec. |

| Simultaneous moves | $n = 5$ |
|---|---|
| Upper bound on number of MPE | 174,449,211,009,120,166,087,753,728 |
| Actual number of equilibria | 192,736,405 |
| Time used | 45 min. |

| Alternating moves | $n = 5$ |
|---|---|
| Upper bound on number of MPE | 174,449,211,009,120,166,087,753,728 |
| Actual number of equilibria | 1 |
| Time used | 0.006 sec. |

# Resolution to the Bertrand investment paradox

### Theorem (Solution to Bertrand investment paradox)

*If investment is socially optimal at a state point $(c_1, c_2, c) \in S$, then*

- *no investment by both firms cannot be an MPE outcome in the subgame starting from $(c_1, c_2, c)$ in either the simultaneous or alternating move versions of the dynamic game.*

**We show:**

1. Many types of endog. coordination is possible in equilibrium
   - Leapfrogging (alternating investments)
   - Preemption (investment by cost leader)
   - Duplicative (simultaneous investments)
2. The equilibria are generally inefficient due to over-investment
   - Duplicative or excessively frequent investments

# Multiplicity of equilibria

### Theorem (Sufficient conditions for uniqueness)

*In the dynamic Bertrand investment and pricing game a sufficient condition for the MPE to be unique is that*

1. *firms move in alternating fashion (i.e. $m \neq 0$), and,*
2. *for each $c > 0$ in the support of $\pi$ we have $\pi(c|c) = 0$.*

1. If firms move simultaneously,
   equilibrium is generally not unique.
2. If technological change is stochastic,
   equilibrium is generally not unique.

Iskhakov, Rust and Schjerning, 2018, IER
"The dynamics of Bertrand price competition with cost-reducing investments."

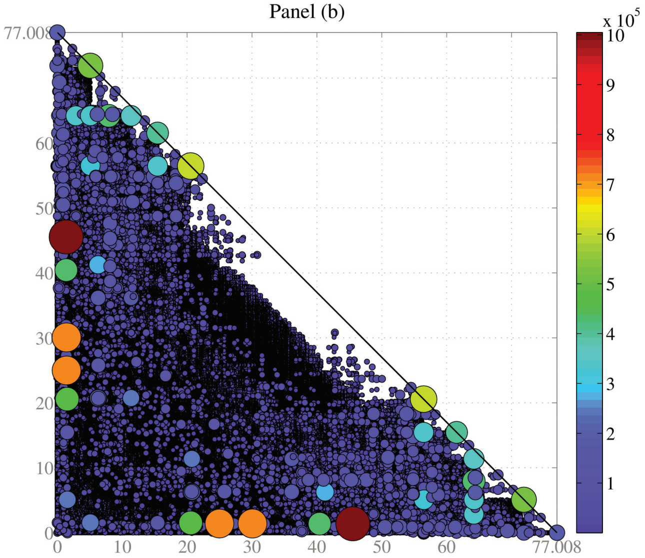
# Pay-offs in the simultaneous move game

## Theorem (Triangular payoffs in the simultaneous move game)

*Suppose that the* $\{c_t\}$ *process has finite support, that there are no idiosyncratic shocks to investment (i.e.* $\eta = 0$*) and that firms move simultaneously*

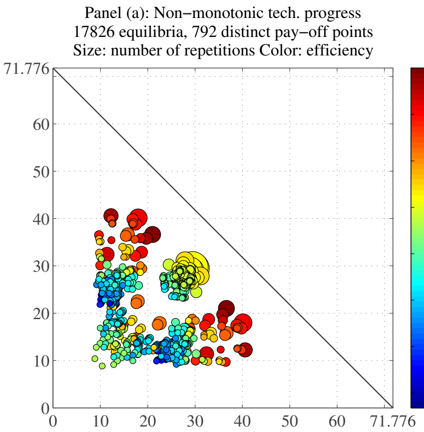
- *The (convex hull of the) set of the expected discounted equilibrium payoffs at the apex state* $(c_0, c_0, c_0) \in S$ *is a triangle*
- *The vertices of this triangle are at the points* $(0, 0)$*,* $(0, V_M)$ *and* $(V_M, 0)$ *where* $V_M = v_{N,i}(c_0, c_0, c_0)$ *is the expected discounted payoff to firm* $i$ *in the monopoly equilibrium where firm* $i$ *is the monopolist investor.*

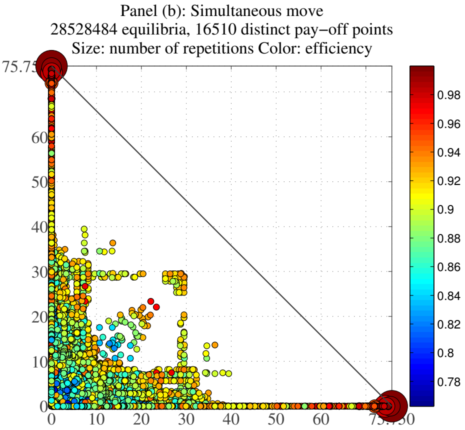
# Pay-off map



Panel (b)
77.008
60
50
40
30
20
10
0
0
10
20
30
40
50
60
77.008
x 10^5
10
9
8
7
6
5
4
3
2
1

# Pay-offs: alternating vs simultaneous move games



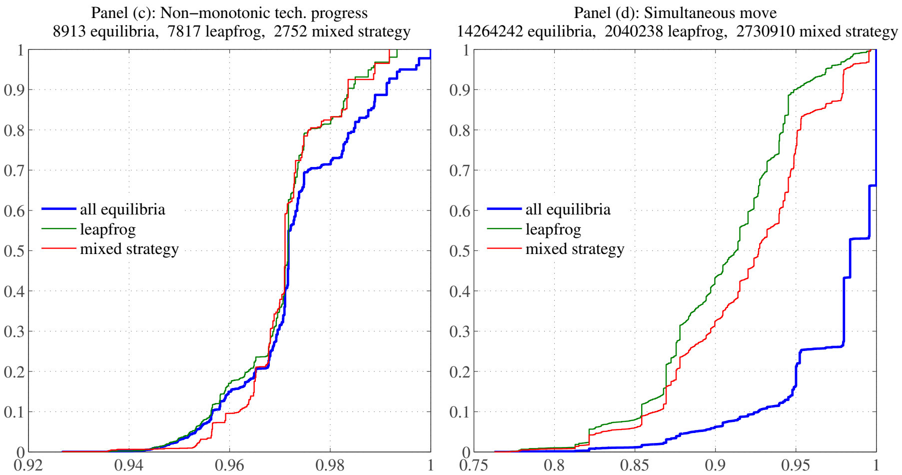
# Efficiency of equilibria

## Simultaneous move game

### Theorem (Inefficiency of mixed strategy equilibria)

*A necessary condition for efficiency in the dynamic Bertrand investment and pricing game is that along MPE path only pure strategy stage equilibria are played.*

# Efficiency: alternating vs simultaneous move games



# Riordan and Salant: Full Preemption

### Theorem (**Riordan and Salant, 1994**)

*The continuous time investment game where*

1. *right to move alternates deterministically.*
2. $K(c) = K$ *and is not prohibitively high.*
3. *technological progress is deterministic:* $c(t)$ *is a continuous, decreasing function*

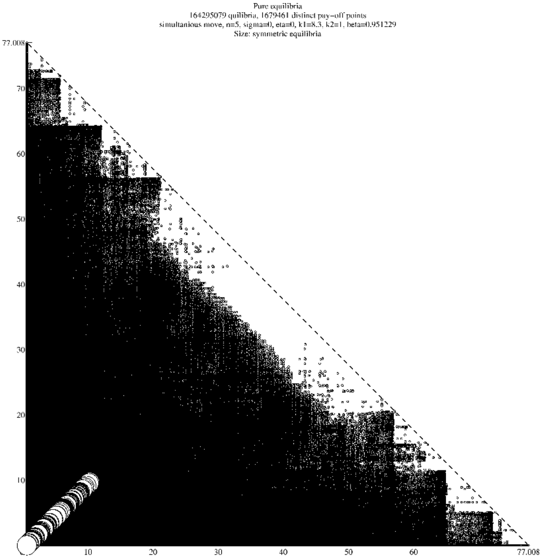
*has a unique MPE with*

- *preemptive investments: by only one firm and no investment in equilibrium by its opponent.*
- *rent dissipation: discounted payoffs of both firms in equilibrium is* $0$*, so the entire surplus is wasted on excessively frequent investments by the preempting firm.*

# We show by computing examples and counterexamples

1. Confirm R&S the result with high $K$and small $dt$
2. Underinvestment: Rent dissapation is not a general outcome - disappears when $K$ is low relative $dt$
3. Leapfrogging: Preemption is not the general outcome - disappears when $K$ is even lower
4. Random move alternation $\rightarrow$ Leapfrogging
5. Random onestep technology $\rightarrow$ Leapfrogging
6. Random multistep technology $\rightarrow$ Leapfrogging
7. Simultaneous moves $\rightarrow$ Leapfrogging

# Symmetric equilibria: $V_1(c_1, c_2, c) = V_2(c_2, c_1, c)$



# Failure of homotopy approach

**Homotopy parameter:** $\eta$

- In each period each firm incurs additive random costs/benefit from not investing and investing
- $\eta$ is a scaling parameter that index variance of idiosyncratic shocks to investment
- High $\eta \to$ unique equilibrium $\eta \to 0 \to$ multiple equilibria

**Problems:**

- Multiplicity of equilibria $\to$ too many bifurcations along the path
- Equilibrium correspondence is not lower hemi-continuous

# Failure of homotopy approach

Equilibrium correspondance, alternating move game: $V_{N,1}(c_0, c_0, c_0)$ vs. $\eta$