Genetic Algorithms and Game Theory

I. Imperfect Competition

Xavier Vilà

Universitat Autònoma de Barcelona

UIB. January 14^{th}, 2005

Outline

Outline

Computation *versus* Simulation

Computation

Find solutions to a problem using computer techniques and/or algorithms

- "Black Box"
- The FINAL RESULT is important
- Concern on efficiency, complexity, convergence, ..

Simulation

Study the behavior of a system using computer simulations to emulate its components

- "Crystal Box"
- The WHOLE PROCESS is important
- Concern on complexity, convergence, adequateness

Computation *versus* Simulation

Computation

Find solutions to a problem using computer techniques and/or algorithms

- “Black Box”
- The FINAL RESULT is important
- Concern on efficiency, complexity, convergence, ..

Simulation

Study the behavior of a system using computer simulations to emulate its components

- “Crystal Box”
- The WHOLE PROCESS is important
- Concern on complexity, convergence, adequateness

Computation *versus* Simulation

Computation

Find solutions to a problem using computer techniques and/or algorithms

- "Black Box"
- The FINAL RESULT is important
- Concern on efficiency, complexity, convergence, ..

Simulation

Study the behavior of a system using computer simulations to emulate its components

- "Crystal Box"
- The WHOLE PROCESS is important
- Concern on complexity, convergence, adequateness

Computation *versus* Simulation

Computation

Find solutions to a problem using computer techniques and/or algorithms

- "Black Box"
- The FINAL RESULT is important
- Concern on efficiency, complexity, convergence, ..

Simulation

Study the behavior of a system using computer simulations to emulate its components

- "Crystal Box"
- The WHOLE PROCESS is important
- Concern on complexity, convergence, adequateness

Computation *versus* Simulation

Computation

Find solutions to a problem using computer techniques and/or algorithms

- "Black Box"
- The FINAL RESULT is important
- Concern on efficiency, complexity, convergence, ..

Simulation

Study the behavior of a system using computer simulations to emulate its components

- "Crystal Box"
- The WHOLE PROCESS is important
- Concern on complexity, convergence, adequateness

Computation *versus* Simulation

Computation

Find solutions to a problem using computer techniques and/or algorithms

- "Black Box"
- The FINAL RESULT is important
- Concern on efficiency, complexity, convergence, ..

Simulation

Study the behavior of a system using computer simulations to emulate its components

- "Crystal Box"
- The WHOLE PROCESS is important
- Concern on complexity, convergence, adequateness

Computation *versus* Simulation

Computation

Find solutions to a problem using computer techniques and/or algorithms

- "Black Box"
- The FINAL RESULT is important
- Concern on efficiency, complexity, convergence, ..

Simulation

Study the behavior of a system using computer simulations to emulate its components

- "Crystal Box"
- The WHOLE PROCESS is important
- Concern on complexity, convergence, adequateness

The Role of Agent-Based Computational Economics *(ACE)*

Agent-Based Computational Economics

- Simulation of the *behavior* of agents in an economic environment
- *Bottom-Up* approach
- Focus on *Emergent Properties*

The Role of Agent-Based Computational Economics *(ACE)*

Agent-Based Computational Economics

- Simulation of the *behavior* of agents in an economic environment
- *Bottom-Up* approach
- Focus on *Emergent Properties*

The Role of Agent-Based Computational Economics *(ACE)*

Agent-Based Computational Economics

- Simulation of the *behavior* of agents in an economic environment
- *Bottom-Up* approach
- Focus on *Emergent Properties*

Emergent Properties

Baking a Pie

Mixing Flour, Eggs, and Sugar and baking it you get more than a *heated dough*

The Market

"Mixing" Buyers, Sellers, and Goods and allowing for interrelations (market) you get more than *busy wandering agents*

Emergent Properties

Baking a Pie

Mixing Flour, Eggs, and Sugar and baking it you get more than a *heated dough*

The Market

“Mixing” Buyers, Sellers, and Goods and allowing for interrelations (market) you get more than *busy wandering agents*

Outline

Competition on some *attribute*

All Agents behave strategically

- SELLERS offer homogeneous goods that differ on some *attribute*
- BUYERS search for goods with the best *attribute*
- BUYERS and SELLERS simultaneously learn (evolve) to produce better strategies

Example

- Sellers choose price
- Buyers look for the best price and decide on *loyalty*

Competition on some *attribute*

All Agents behave strategically

- SELLERS offer homogeneous goods that differ on some *attribute*
- BUYERS search for goods with the best *attribute*
- BUYERS and SELLERS simultaneously learn (evolve) to produce better strategies

Example

- Sellers choose price
- Buyers look for the best price and decide on *loyalty*

Competition on some *attribute*

All Agents behave strategically

- SELLERS offer homogeneous goods that differ on some *attribute*
- BUYERS search for goods with the best *attribute*
- BUYERS and SELLERS simultaneously learn (evolve) to produce better strategies

Example

- Sellers choose price
- Buyers look for the best price and decide on *loyalty*

Competition on some *attribute*

All Agents behave strategically

- SELLERS offer homogeneous goods that differ on some *attribute*
- BUYERS search for goods with the best *attribute*
- BUYERS and SELLERS simultaneously learn (evolve) to produce better strategies

Example

- Sellers choose price
- Buyers look for the best price and decide on *loyalty*

Competition on some *attribute*

All Agents behave strategically

- SELLERS offer homogeneous goods that differ on some *attribute*
- BUYERS search for goods with the best *attribute*
- BUYERS and SELLERS simultaneously learn (evolve) to produce better strategies

Example

- Sellers choose price
- Buyers look for the best price and decide on *loyalty*

Outline

Genetic Algorithms: *Concept*

Genetic Algorithms

Class of computer routines developed by Holland for optimization in domains with both complicated search spaces and objective functions with non-linearities, discontinuities and high dimensionality

Genetic Algorithm techniques have been broadly used to simulate the evolution of agent's behavior

Genetic Algorithms: *Concept*

Genetic Algorithms

Class of computer routines developed by Holland for optimization in domains with both complicated search spaces and objective functions with non-linearities, discontinuities and high dimensionality

Genetic Algorithm techniques have been broadly used to simulate the evolution of agent's behavior

Genetic Algorithms: *Schema*

Population at time t

01100110
10011011
00100101
00011011

Population at time $t + 0.5$

00011011
01100110
00100101
10011011

Population at time $t + 1$

00011011
00111011
00101001
01100111

Initial Population

Reorder

According to *performance*

Operators

Replication, Crossover, Mutation

Genetic Algorithms: *Schema*

Population at time t

01100110
10011011
00100101
00011011

Population at time $t + 0.5$

00011011
01100110
00100101
10011011

Population at time $t + 1$

00011011
00111011
00101001
01100111

Initial Population

Reorder

According to *performance*

Operators

Replication, Crossover, Mutation

Genetic Algorithms: *Schema*

Population at time t

01100110
10011011
00100101
00011011

Population at time $t+0.5$

00011011
01100110
00100101
10011011

Population at time $t+1$

00011011
00111011
00101001
01100111

Initial Population

Reorder

According to *performance*

Operators

Replication, Crossover, Mutation

Genetic Algorithms: *Crossover*

Population at time $t + 0.5$

- 00011011
- 01100110
- 00100101
- 10011011

Parents

Offspring

Genetic Algorithms: *Crossover*

Population at time $t + 0.5$

- 00011011
- 01100110
- 00100101
- 10011011

Parents

Offspring

Randomly select two *parents*

Genetic Algorithms: *Crossover*

Population at time $t + 0.5$

- 00011011
- 01100110
- 00100101
- 10011011

Parents

- 00011011
- 00100101

Offspring

Randomly select two *parents*

Genetic Algorithms: *Crossover*

Population at time $t + 0.5$

- 00011011
- 01100110
- 00100101
- 10011011

Parents

- 0001|1011
- 0010|0101

Offspring

Randomly determine a *cut point*

Genetic Algorithms: *Crossover*

Population at time $t + 0.5$

- 00011011
- 01100110
- 00100101
- 10011011

Parents

- 0001|1011
- 0010|0101

Offspring

- 00010101
- 00101011

Create two *children* by crossoving *parents* at the *cut point*

Genetic Algorithms: *Mutation*

For each *string*

00010101

Each bit mutates with small probability

Genetic Algorithms: *Mutation*

For each *string*

00010101

Each bit mutates with small probability ε

00010101

Genetic Algorithms: *Mutation*

For each *string*

00010101

Each bit mutates with small probability ε

00011101

Outline

The Repeated Prisoners' Dilemma

Play the Repeated Prisoners' Dilemma

	C	D
C	3,3	0,5
D	5,0	1,1

Strategies Represented by Finite Automata

Cooperate

$C \longrightarrow 0$

Defect

$D \longrightarrow 1$

The Repeated Prisoners' Dilemma

Play the Repeated Prisoners' Dilemma

	C	D
C	3,3	0,5
D	5,0	1,1

Strategies Represented by Finite Automata

Cooperate

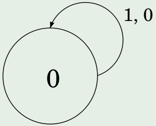
$C \to 0$

Defect

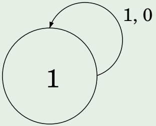
$D \to 1$

Finite Automata

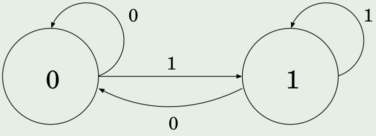
Always Cooperate



Always Defect



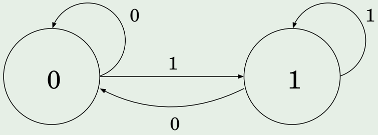
Tit for Tat



Automata *Encoding*

Automata (strategies) need to be encoded as 0's and 1's

Tit for Tat

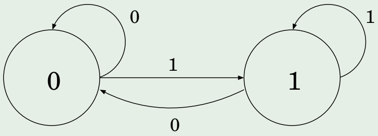


Might be encoded as ...

Automata *Encoding*

Automata (strategies) need to be encoded as 0's and 1's

Tit for Tat

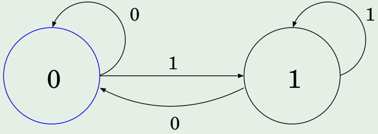


Might be encoded as ...

Automata *Encoding*

Automata (strategies) need to be encoded as 0's and 1's

Tit for Tat



Might be encoded as ...

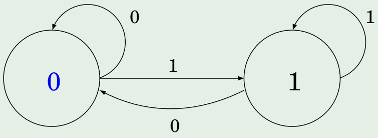
0

- Initial state (state 0)

Automata *Encoding*

Automata (strategies) need to be encoded as 0's and 1's

Tit for Tat



Might be encoded as ...

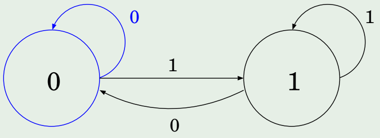
00

- State 0: What to do at this state (COOPERATE)

Automata *Encoding*

Automata (strategies) need to be encoded as 0's and 1's

Tit for Tat



Might be encoded as ...

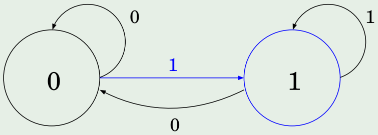
000

- State 0: Where to go if my opponent COOPERATES (state 0)

Automata *Encoding*

Automata (strategies) need to be encoded as 0's and 1's

Tit for Tat



Might be encoded as ...

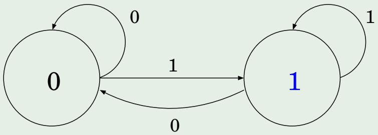
0001

- State 0: Where to go if my opponent DEFECTS (state 1)

Automata *Encoding*

Automata (strategies) need to be encoded as 0's and 1's

Tit for Tat



Might be encoded as ...

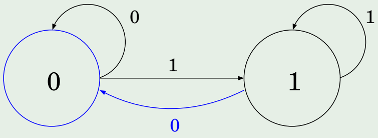
00011

- State 1: What to do at this state (DEFECT)

Automata *Encoding*

Automata (strategies) need to be encoded as 0's and 1's

Tit for Tat



Might be encoded as ...

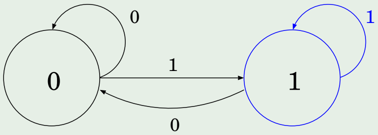
000110

- State 1: Where to go if my opponent COOPERATES (state 0)

Automata *Encoding*

Automata (strategies) need to be encoded as 0's and 1's

Tit for Tat



Might be encoded as ...

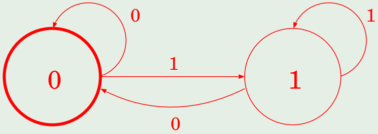
0001101

- State 1: Where to go if my opponent DEFECTS (state 1)

Automata *Encoding*

Automata (strategies) need to be encoded as 0's and 1's

Tit for Tat



Might be encoded as ...

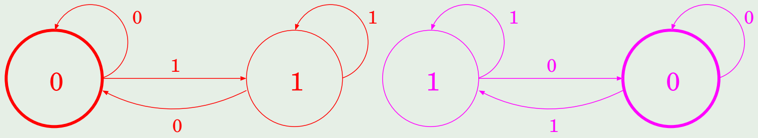
0001101

- Tit for Tat *Encoded*

Automata *Encoding*

Automata (strategies) need to be encoded as 0's and 1's

Tit for Tat



Might be encoded as ...

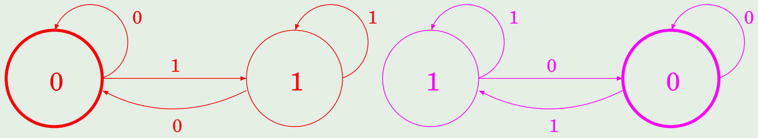
0001101

- Tit for Tat RELABELED

Automata *Encoding*

Automata (strategies) need to be encoded as 0's and 1's

Tit for Tat



Might be encoded as ...

0001101 1110010

- Tit for Tat *Encoded* in two different ways !!

Canonical Crossover *oddities*

With this representation, the standard crossover operator poses two problems:

- What is the interpretation ?
- Missleading behavior:

Canonical Crossover *oddities*

With this representation, the standard crossover operator poses two problems:

- What is the interpretation ?
- Missleading behavior:

Canonical Crossover *oddities*

With this representation, the standard crossover operator poses two problems:

- What is the interpretation ?
- Missleading behavior:

Canonical Crossover *oddities: An Example*

Two “Always Cooperate” strategies are selected for crossover. The “cut point” is right after the first bit

0 | 0 0 0 1 1 1
1 | 1 0 0 0 1 1

The outcome of the crossover will be ...

0 | 1 0 0 0 1 1
1 | 0 0 0 1 1 1

Two “Always Defect” strategies !!!

Canonical Crossover *oddities: An Example*

Two "Always Cooperate" strategies are selected for crossover. The "cut point" is right after the first bit

$$\begin{array}{c|ccc c ccc} 0 & 0 & 0 & 0 & & 1 & 1 & 1 \\ 1 & 1 & 0 & 0 & & 0 & 1 & 1 \end{array}$$

The outcome of the crossover will be ...

$$\begin{array}{c|ccc c ccc} 0 & 1 & 0 & 0 & & 0 & 1 & 1 \\ 1 & 0 & 0 & 0 & & 1 & 1 & 1 \end{array}$$

Two "Always Defect" strategies !!!

Canonical Crossover *oddities: An Example*

Two "Always Cooperate" strategies are selected for crossover.
The "cut point" is right after the first bit

0 | 0 0 0 1 1 1
1 | 1 0 0 0 1 1

The outcome of the crossover will be ...

0 | 1 0 0 0 1 1
1 | 0 0 0 1 1 1

Two "Always Defect" strategies !!!

Modified Crossover: *Partial Imitation*

Partial Imitation Crossover

1. Randomly generate a history of a given length. For instance 0010.
2. Determine the move that each of the two parents would make given the sequence of inputs described by the history generated.
3. Form two new automata that reproduce the two parents but "switching" the move reported in step 2. Hence, the first new automaton will use the move reported by the second automaton if the sequence of inputs it gets is the one described by the history considered and vice versa.

Modified Crossover: *Partial Imitation*

Partial Imitation Crossover

1. Randomly generate a history of a given length. For instance 0010.
2. Determine the move that each of the two parents would make given the sequence of inputs described by the history generated.
3. Form two new automata that reproduce the two parents but "switching" the move reported in step 2. Hence, the first new automaton will use the move reported by the second automaton if the sequence of inputs it gets is the one described by the history considered and vice versa.

Modified Crossover: *Partial Imitation*

Partial Imitation Crossover

1. Randomly generate a history of a given length. For instance 0010.
2. Determine the move that each of the two parents would make given the sequence of inputs described by the history generated.
3. Form two new automata that reproduce the two parents but "switching" the move reported in step 2. Hence, the first new automaton will use the move reported by the second automaton if the sequence of inputs it gets is the one described by the history considered and vice versa.

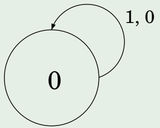
Modified Crossover: *Partial Imitation*

Partial Imitation Crossover

1. Randomly generate a history of a given length. For instance 0010.
2. Determine the move that each of the two parents would make given the sequence of inputs described by the history generated.
3. Form two new automata that reproduce the two parents but "switching" the move reported in step 2. Hence, the first new automaton will use the move reported by the second automaton if the sequence of inputs it gets is the one described by the history considered and vice versa.

Modified Crossover: *An Example*

Always Cooperate



Parents

0 0 0 0 1 1 1
1 1 0 0 0 1 1

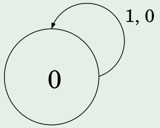
Fictitious History

0010

Children

Modified Crossover: *An Example*

Always Cooperate



Parents

0 0 0 0 1 1 1
1 1 0 0 0 1 1

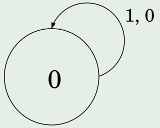
Fictitious History

0010

Children

Modified Crossover: *An Example*

Always Cooperate



Parents

0 0 0 0 1 1 1

1 1 0 0 0 1 1

Fictitious History

0010

Children

0 0 0 0 1 1 1

1 1 0 0 0 1 1

Canonical Mutation *oddities*

Canonical Mutation

In order to complete the population at time $t-1$, each "bit" switches its value according to some probability p. It turns out that this induces a non-uniform distribution across states

$$01 \rightarrow \begin{cases} 00 & (1-p)p \\ 01 & (1-p)^2 \\ 10 & p^2 \\ 11 & (1-p)p \end{cases}$$

Modified Mutation

Modified Mutation

This can be fixed easily making "statewise" mutations

$$01 \longrightarrow \begin{cases} 00 & \frac{p}{3} \\ 01 & (1-p) \\ 10 & \frac{p}{3} \\ 11 & \frac{p}{3} \end{cases}$$

Outline

The Repeated Prisoners' Dilemma

Parameters

- Size of population: 100
- Number of Generations: 5000
- Number of Rounds: *from* 10 *to* 250
- Probability of Mutation: *from* 0.00 *to* 0.02 (step 0.001)
- Four different crossovers and NO Crossover
- Automata of size 4

The Repeated Prisoners' Dilemma

Parameters

- Size of population: 100
- Number of Generations: 5000
- Number of Rounds: *from* 10 *to* 250
- Probability of Mutation: *from* 0.00 *to* 0.02 (step 0.001)
- Four different crossovers and NO Crossover
- Automata of size 4

The Repeated Prisoners' Dilemma

Parameters

- Size of population: 100
- Number of Generations: 5000
- Number of Rounds: *from* 10 *to* 250
- Probability of Mutation: *from* 0.00 *to* 0.02 (step 0.001)
- Four different crossovers and NO Crossover
- Automata of size 4

The Repeated Prisoners' Dilemma

Parameters

- Size of population: 100
- Number of Generations: 5000
- Number of Rounds: $from$ 10 to 250
- Probability of Mutation: $from$ 0.00 to 0.02 (step 0.001)
- Four different crossovers and NO Crossover
- Automata of size 4

The Repeated Prisoners' Dilemma

Parameters

- Size of population: 100
- Number of Generations: 5000
- Number of Rounds: $from$ 10 to 250
- Probability of Mutation: $from$ 0.00 to 0.02 (step 0.001)
- Four different crossovers and NO Crossover
- Automata of size 4

The Repeated Prisoners' Dilemma

Parameters

- Size of population: 100
- Number of Generations: 5000
- Number of Rounds: $from$ 10 to 250
- Probability of Mutation: $from$ 0.00 to 0.02 (step 0.001)
- Four different crossovers and NO Crossover
- Automata of size 4

Different Crossovers

Fifty-Fifty Crossover

Randomly select two parents (as in the "canonical crossover")
Then each "locus" of the new children will have 50% probability of coming from each of the parents

Replica Crossover

Randomly select two parents (as in the "canonical crossover")
Then each children will be a exact replica of each parent

Different Crossovers

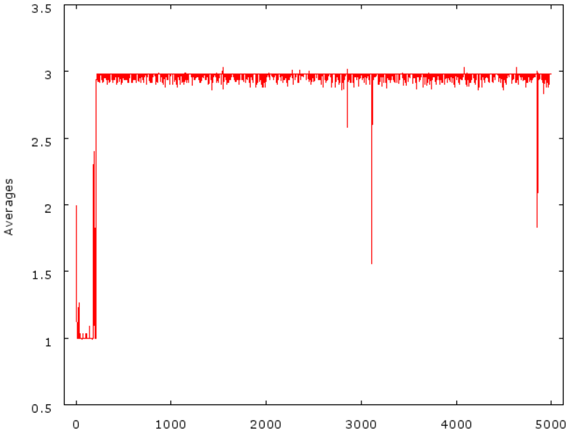
Fifty-Fifty Crossover

Randomly select two parents (as in the “canonical crossover”)
Then each “locus” of the new children will have 50% probability of coming from each of the parents

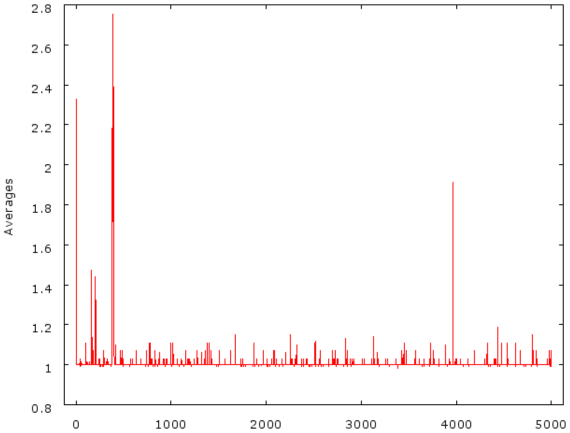
Replica Crossover

Randomly select two parents (as in the “canonical crossover”)
Then each children will be a exact replica of each parent

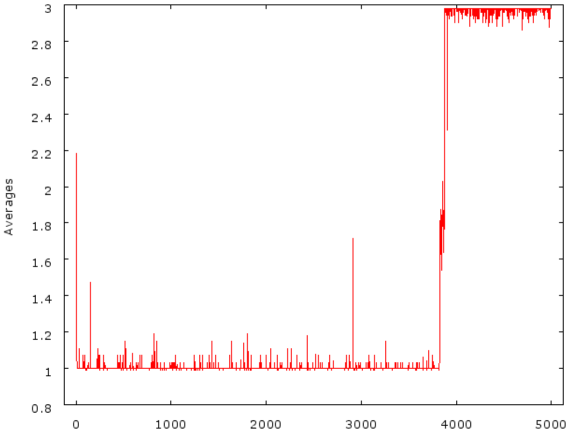
The Repeated Prisoners' Dilemma: Typical Output



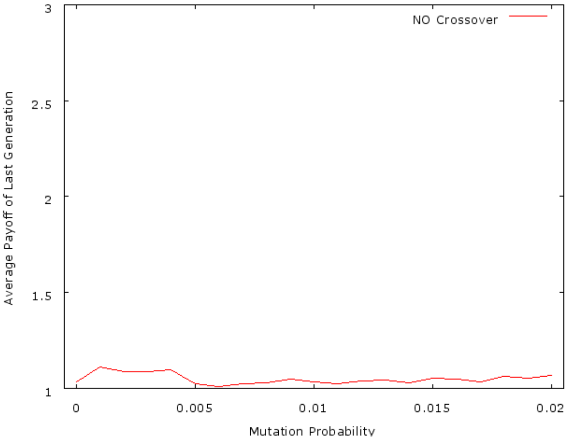
The Repeated Prisoners' Dilemma: Alternative Output



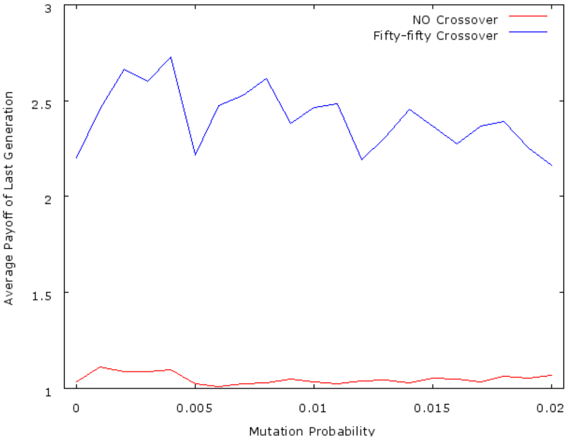
The Repeated Prisoners' Dilemma: Occasional Output



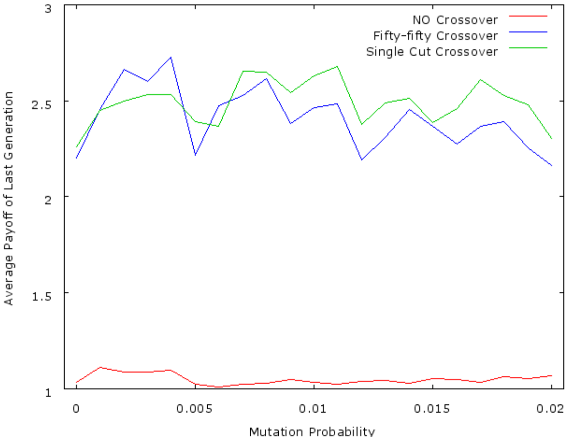
Different Crossovers (*changing Mutations*)



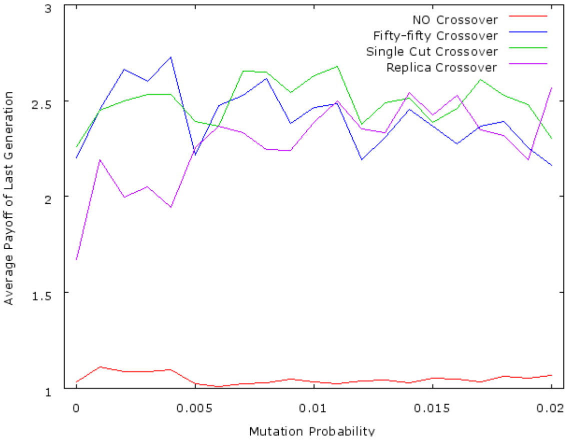
Different Crossovers (*changing Mutations*)



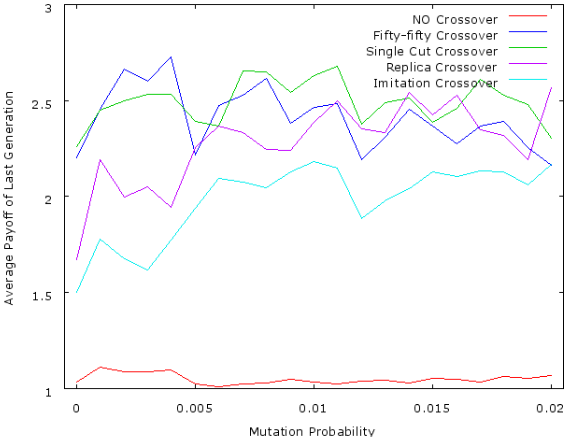
Different Crossovers (*changing Mutations*)



Different Crossovers *(changing Mutations)*



Different Crossovers (*changing Mutations*)



Outline

1. Motivation
   - Methodological Motivation
   - Economical Motivation
2. Genetic Algorithms
   - Canonical Genetic Algorithms
   - Modified Genetic Algorithms
   - An Example: The Repeated Prisoners' Dilemma
   - Interpretation: What are we simulating ?
3. Application: Imperfect Competition
   - The Model
   - The Deterministic Approach
   - The Simulation Approach

Interpretation

The Repeated Prisoners' Dilemma

	C	D
C	3,3	0,5
D	5,0	1,1

Interpretation

The Repeated Prisoners' Dilemma

	C	D
C	3,3	0,5
D	5,0	1,1

Consider only tree possible strategies

- Always Coperate: C
- Always Defect: D
- Tit for Tat: T

Interpretation

The Repeated Prisoners' Dilemma

	C	D
C	3,3	0,5
D	5,0	1,1

Consider only tree possible strategies

- Always Coperate: C
- Always Defect: D
- Tit for Tat: T

Interpretation

The Repeated Prisoners' Dilemma

	C	D
C	3,3	0,5
D	5,0	1,1

Consider only tree possible strategies

- Always Coperate: C
- Always Defect: D
- Tit for Tat: T

Interpretation

The Repeated Prisoners' Dilemma

	C	D
C	3,3	0,5
D	5,0	1,1

Consider only tree possible strategies

- Always Coperate: C
- Always Defect: D
- Tit for Tat: T

Interpretation

Play the Prisoners' Dilemma for R rounds. The results will be as in the table below

	C	D	T
C	$3R$	0	$3R$
D	$5R$	R	$5+(R-1)$
T	$3R$	$0+(R-1)$	$3R$

Represented by the Matrix

$$A = \begin{pmatrix} 3R & 0 & 3R \\ 5R & R & 5+(R-1) \\ 3R & 0+(R-1) & 3R \end{pmatrix}$$

Interpretation

Play the Prisoners' Dilemma for R rounds. The results will be as in the table below

	C	D	T
C	$3R$	0	$3R$
D	$5R$	R	$5+(R-1)$
T	$3R$	$0+(R-1)$	$3R$

Represented by the Matrix

$$A = \begin{pmatrix} 3R & 0 & 3R \\ 5R & R & 5+(R-1) \\ 3R & 0+(R-1) & 3R \end{pmatrix}$$

Interpretation

Let p_t the vector consisting of the proportions of each strategy in the population at time t

$$p_t = \begin{pmatrix} p_t(C) \\ p_t(D) \\ p_t(T) \end{pmatrix}$$

Then, the expected payoff of strategy $s \in \{C, D, T\}$ at time t is the product of the $s - th$ row of A and p_t:

$$E_t(s) = A^s.p_t$$

Interpretation

Let p_t the vector consisting of the proportions of each strategy in the population at time t

$$p_t = \begin{pmatrix} p_t(C) \\ p_t(D) \\ p_t(T) \end{pmatrix}$$

Then, the expected payoff of strategy $s \in \{C, D, T\}$ at time t is the product of the $s - th$ row of A and p_t:

$$E_t(s) = A^s . p_t$$

Interpretation

Let p_t the vector consisting of the proportions of each strategy in the population at time t

$$p_t = \begin{pmatrix} p_t(C) \\ p_t(D) \\ p_t(T) \end{pmatrix}$$

Then, the expected payoff of strategy $s \in \{C, D, T\}$ at time t is the product of the $s-th$ row of A and p_t:

$$E_t(s) = A^s . p_t$$

Interpretation

Let p_t the vector consisting of the proportions of each strategy in the population at time t

$$p_t = \begin{pmatrix} p_t(C) \\ p_t(D) \\ p_t(T) \end{pmatrix}$$

Then, the expected payoff of strategy $s \in \{C, D, T\}$ at time t is the product of the $s - th$ row of A and p_t:

$$E_t(s) = A^s . p_t$$

Interpretation

Consider the replicator dynamics to represent the evolution of strategies:

$$p_{t+1}(s) = p_t(s)\frac{A^s \cdot p_t}{p_t^T \cdot A \cdot p_t} \quad s \in \{C, D, T\}$$

The change in the proportion of each strategy is proportional to its performance (numerator) relative to the average performance (denominator)

Interpretation

Consider the replicator dynamics to represent the evolution of strategies:

$$p_{t+1}(s) = p_t(s)\frac{A^s \cdot p_t}{p_t^T \cdot A \cdot p_t} \quad s \in \{C, D, T\}$$

The change in the proportion of each strategy is proportional to its performance (numerator) relative to the average performance (denominator)

Interpretation

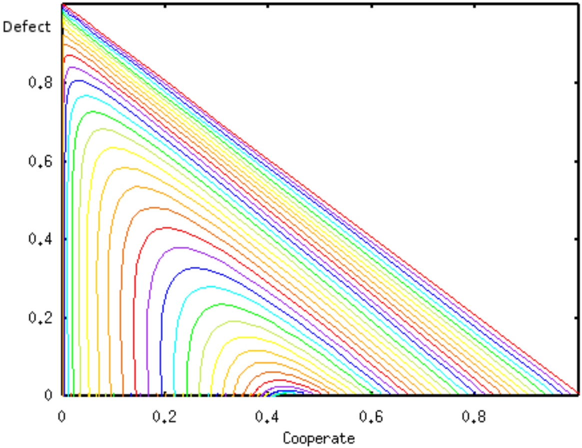
Consider the replicator dynamics to represent the evolution of strategies:

$$p_{t+1}(s) = p_t(s)\frac{A^s \cdot p_t}{p_t^T \cdot A \cdot p_t} \quad s \in \{C, D, T\}$$

The change in the proportion of each strategy is proportional to its performance (numerator) relative to the average performance (denominator)

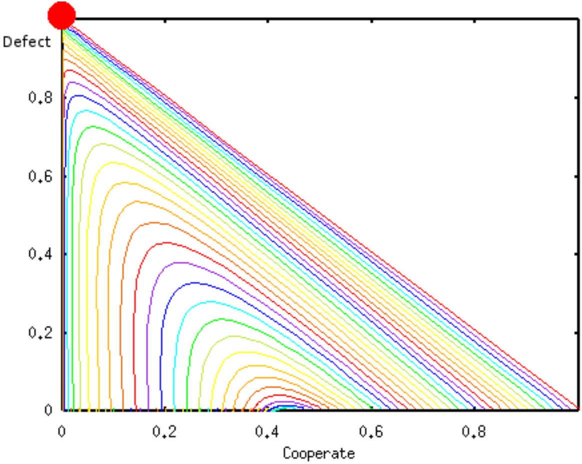
Interpretation

Behavior of the system:



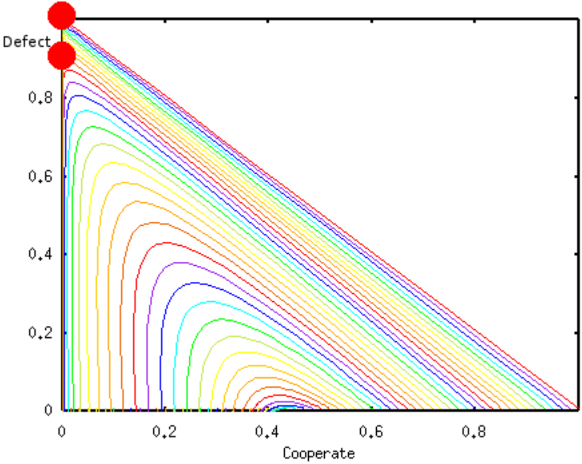
Interpretation

Behavior of the system:



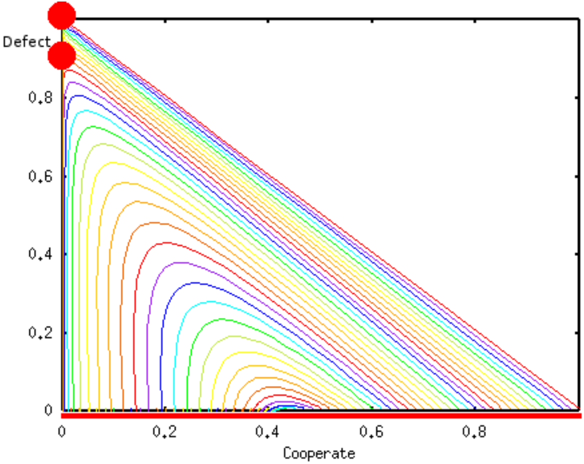
Interpretation

Behavior of the system:



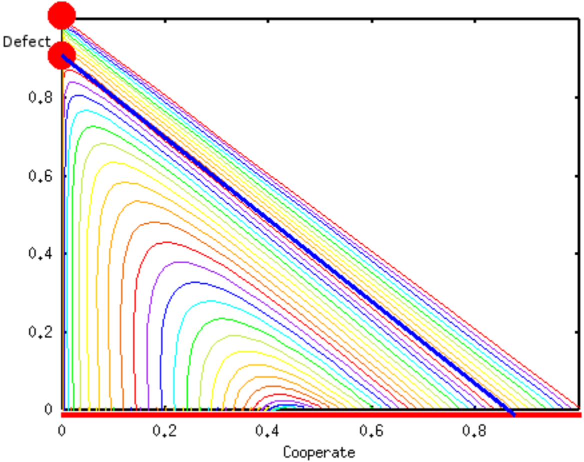
Interpretation

Behavior of the system:



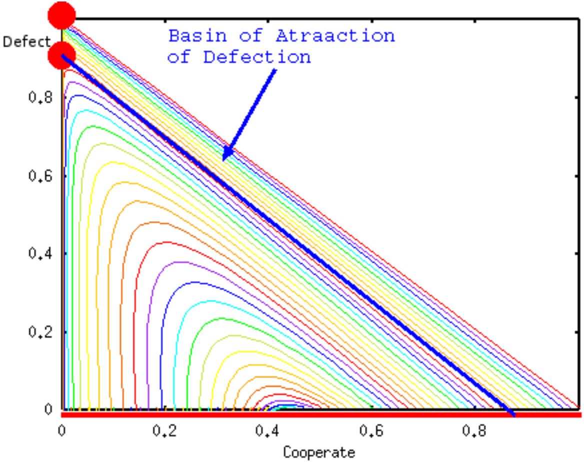
Interpretation

Behavior of the system:



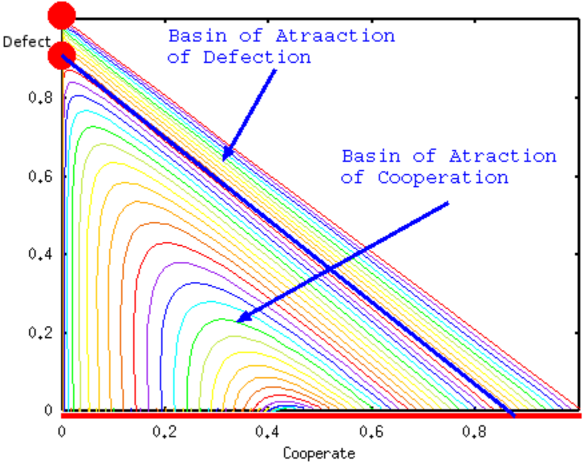
Interpretation

Behavior of the system:

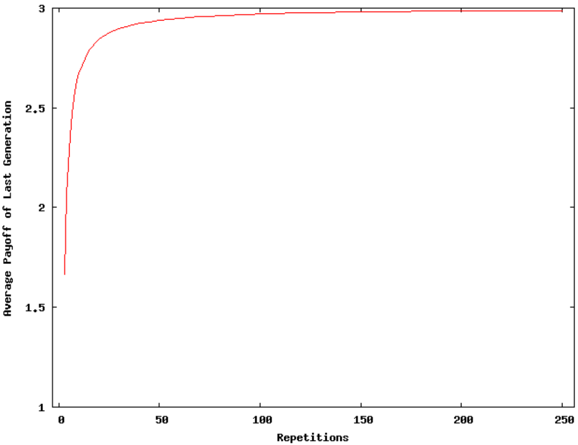


Interpretation

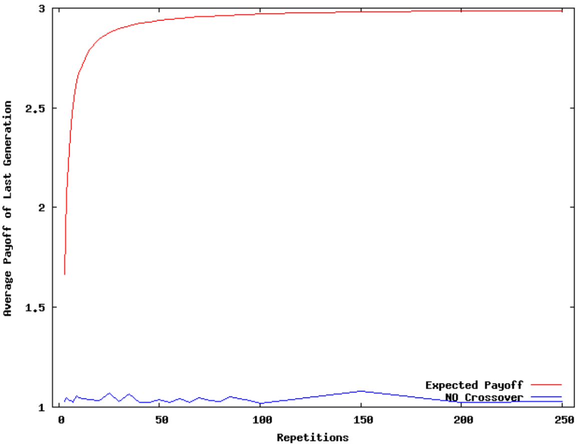
Behavior of the system:



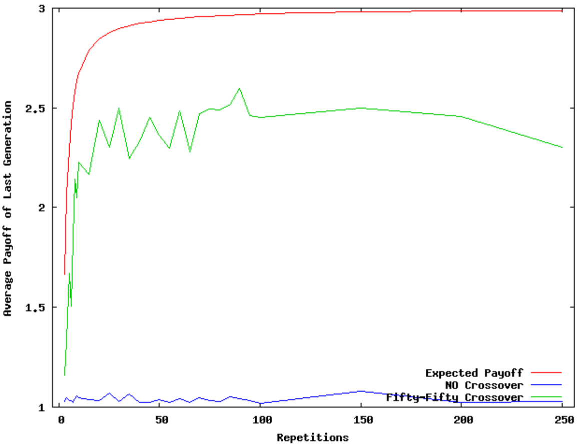
Different Crossovers (*changing Rounds*)



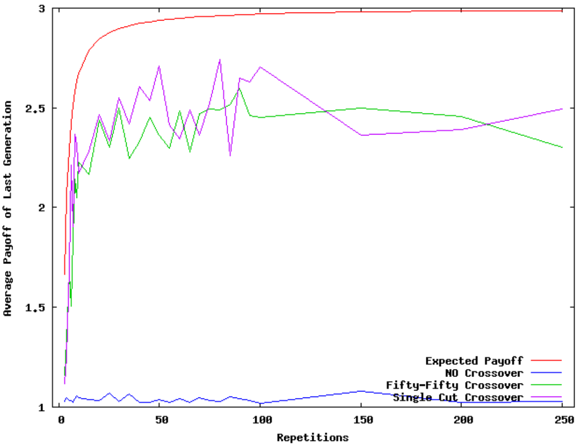
Different Crossovers (*changing Rounds*)



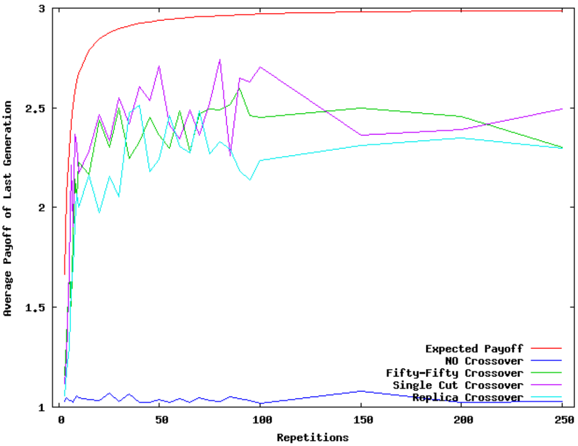
Different Crossovers *(changing Rounds)*



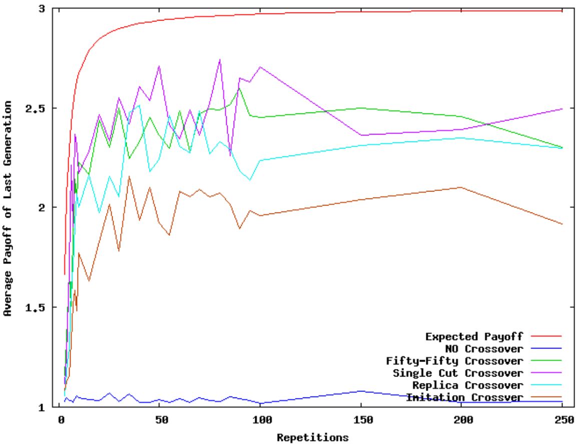
Different Crossovers (*changing Rounds*)



Different Crossovers (*changing Rounds*)



Different Crossovers *(changing Rounds)*



Outline

Buyers and Sellers

Sellers

Two SELLERS offering the same good compete repeatedly in some *attribute* (price in this example)

Buyers

m BUYERS Look for the best *attribute* and shop from the corresponding SELLER. In the case the two SELLERS offer the same attribute, BUYERS can decide to shop from the same SELLER as before (loyalty) or decide randomly

Buyers and Sellers

Sellers

Two SELLERS offering the same good compete repeatedly in some *attribute* (price in this example)

Buyers

m BUYERS Look for the best *attribute* and shop from the corresponding SELLER. In the case the two SELLERS offer the same attribute, BUYERS can decide to shop from the same SELLER as before (loyalty) or decide randomly

The Model

- 2 sellers
- m buyers
- T periods, R rounds per period (Example: 54 weeks, 5 rounds per week)
- Sellers' strategies per period
  - Set a price for Round = 1
  - For Round > 1, set a price based on previous rounds of the period
  - price $\in \{\bar{p}, \underline{p}\}$
- Buyers' strategies per period
  - Always go to the cheapest seller
  - If prices are equal:
    - Randomly split
    - Be "Loyal"

The Model

- 2 sellers
- m buyers
- T periods, R rounds per period (Example: 54 weeks, 5 rounds per week)
- Sellers' strategies per period
  - Set a price for Round = 1
  - For Round > 1, set a price based on previous rounds of the period
  - price $\in \{\bar{p}, \underline{p}\}$
- Buyers' strategies per period
  - Always go to the cheapest seller
  - If prices are equal:
    - Randomly split
    - Be "Loyal"

The Model

- 2 sellers
- m buyers
- T periods, R rounds per period (Example: 54 weeks, 5 rounds per week)
- Sellers' strategies per period
  - Set a price for Round = 1
  - For Round > 1, set a price based on previous rounds of the period
  - price $\in \{\bar{p}, \underline{p}\}$
- Buyers' strategies per period
  - Always go to the cheapest seller
  - If prices are equal:
    - Randomly split
    - Be "Loyal"

The Model

- 2 sellers
- m buyers
- T periods, R rounds per period (Example: 54 weeks, 5 rounds per week)
- Sellers' strategies per period
  - Set a price for Round = 1
  - For Round > 1, set a price based on previous rounds of the period
  - price $\in \{\bar{p}, \underline{p}\}$
- Buyers' strategies per period
  - Always go to the cheapest seller
  - If prices are equal:
    - Randomly split
    - Be "Loyal"

The Model

- 2 sellers
- m buyers
- T periods, R rounds per period (Example: 54 weeks, 5 rounds per week)
- Sellers' strategies per period
  - Set a price for Round = 1
  - For Round > 1, set a price based on previous rounds of the period
  - price $\in \{\bar{p}, \underline{p}\}$
- Buyers' strategies per period
  - Always go to the cheapest seller
  - If prices are equal:

The Model

- 2 sellers
- m buyers
- T periods, R rounds per period (Example: 54 weeks, 5 rounds per week)
- Sellers' strategies per period
  - Set a price for Round = 1
  - For Round > 1, set a price based on previous rounds of the period
  - price $\in \{\bar{p}, \underline{p}\}$
- Buyers' strategies per period
  - Always go to the cheapest seller
  - If prices are equal:
    - Randomly split

The Model

- 2 sellers
- m buyers
- T periods, R rounds per period (Example: 54 weeks, 5 rounds per week)
- Sellers' strategies per period
  - Set a price for Round = 1
  - For Round > 1, set a price based on previous rounds of the period
  - price $\in \{\bar{p}, \underline{p}\}$
- Buyers' strategies per period
  - Always go to the cheapest seller
  - If prices are equal:
    - Randomly split
    - Be "Loyal"

The Model

- 2 sellers
- m buyers
- T periods, R rounds per period (Example: 54 weeks, 5 rounds per week)
- Sellers' strategies per period
  - Set a price for Round = 1
  - For Round > 1, set a price based on previous rounds of the period
  - price $\in \{\bar{p}, \underline{p}\}$
- Buyers' strategies per period
  - Always go to the cheapest seller
  - If prices are equal:
    - Randomly split
    - Be "Loyal"

The Model

- 2 sellers
- m buyers
- T periods, R rounds per period (Example: 54 weeks, 5 rounds per week)
- Sellers' strategies per period
  - Set a price for Round = 1
  - For Round > 1, set a price based on previous rounds of the period
  - price $\in \{\bar{p}, \underline{p}\}$
- Buyers' strategies per period
  - Always go to the cheapest seller
  - If prices are equal:
    - Randomly split
    - Be "Loyal"

The Model

- 2 sellers
- m buyers
- T periods, R rounds per period (Example: 54 weeks, 5 rounds per week)
- Sellers' strategies per period
  - Set a price for Round = 1
  - For Round > 1, set a price based on previous rounds of the period
  - price $\in \{\bar{p}, \underline{p}\}$
- Buyers' strategies per period
  - Always go to the cheapest seller
  - If prices are equal:
    - Randomly split
    - Be "Loyal"

The Model

- 2 sellers
- m buyers
- T periods, R rounds per period (Example: 54 weeks, 5 rounds per week)
- Sellers' strategies per period
  - Set a price for Round = 1
  - For Round > 1, set a price based on previous rounds of the period
  - price $\in \{\bar{p}, \underline{p}\}$
- Buyers' strategies per period
  - Always go to the cheapest seller
  - If prices are equal:
    - Randomly split
    - Be "Loyal"

The Model

- 2 sellers
- m buyers
- T periods, R rounds per period (Example: 54 weeks, 5 rounds per week)
- Sellers' strategies per period
  - Set a price for Round = 1
  - For Round > 1, set a price based on previous rounds of the period
  - price $\in \{\bar{p}, \underline{p}\}$
- Buyers' strategies per period
  - Always go to the cheapest seller
  - If prices are equal:
    - Randomly split
    - Be "Loyal"

Outline

Strategies

Strategies for the Sellers

$$s \in \{C, D, T\}$$

- C: Always set a high price (always cooperate)
- D: Always set a low price (always defect)
- T: Tit for Tat (start with a high price and then imitate your opponent)

Strategies for the Buyers

- Go for cheap
- If indifferent
  - Randomize
  - Be “Loyal”

Strategies

Strategies for the Sellers

$$s \in \{C, D, T\}$$

- C: Always set a high price (always cooperate)
- D: Always set a low price (always defect)
- T: Tit for Tat (start with a high price and then imitate your opponent)

Strategies for the Buyers

- Go for cheap
- If indifferent
  - Randomize
  - Be “Loyal”

System

Parameters

- $\bar{p} = 3$ (5)
- $\underline{p} = 2$
- $m = 2$

Payoff Matrix

$$A = \begin{pmatrix} 3R & 0 & 3R \\ 4R & 2R & 4 + 2(R-1) \\ 3R & 2(R-1) & 3R \end{pmatrix}$$

Dynamics

$$p_{t+1}(s) = p_t(s) \frac{A^s \cdot p_t}{p_t^T \cdot A \cdot p_t}$$

System

Parameters

- $\bar{p}$ = 3 (5)
- $\underline{p}$ = 2
- m = 2

Payoff Matrix

$$A = \begin{pmatrix} 3R & 0 & 3R \\ 4R & 2R & 4 + 2(R-1) \\ 3R & 2(R-1) & 3R \end{pmatrix}$$

Dynamics

$$p_{t+1}(s) = p_t(s)\frac{A^s \cdot p_t}{p_t^T \cdot A \cdot p_t}$$

System

Parameters

- $\bar{p}$ = 3 (5)
- $\underline{p}$ = 2
- m = 2

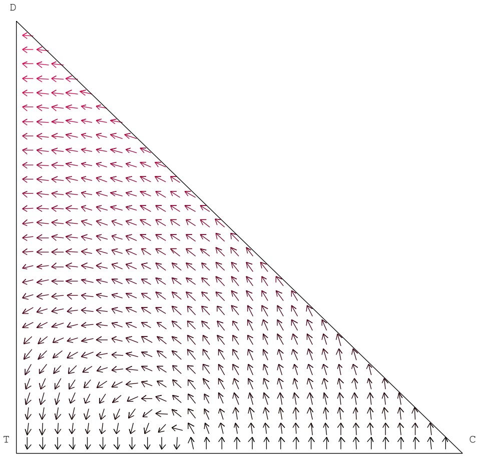
Payoff Matrix

$$A = \begin{pmatrix} 3R & 0 & 3R \\ 4R & 2R & 4+2(R-1) \\ 3R & 2(R-1) & 3R \end{pmatrix}$$

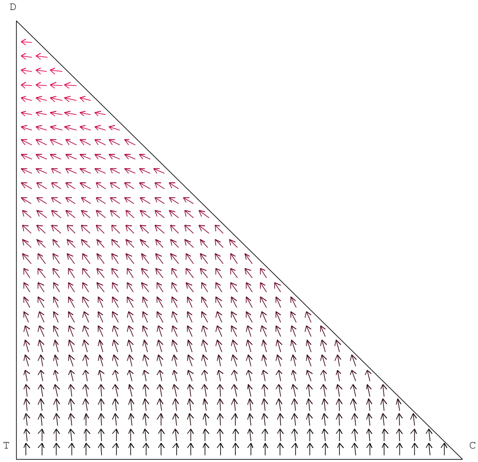
Dynamics

$$p_{t+1}(s) = p_t(s)\frac{A^s \cdot p_t}{p_t^T \cdot A \cdot p_t}$$

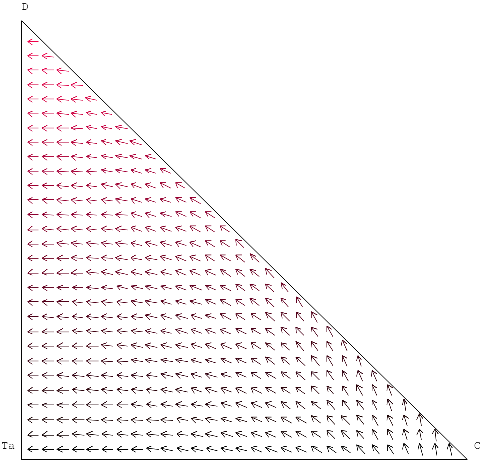
Results: Non-Loyal Buyers ($\bar{p} = 3$)



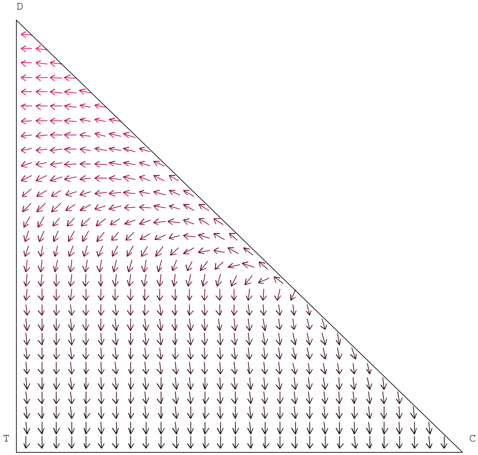
Results: Loyal Buyers ($\bar{p} = 3$)



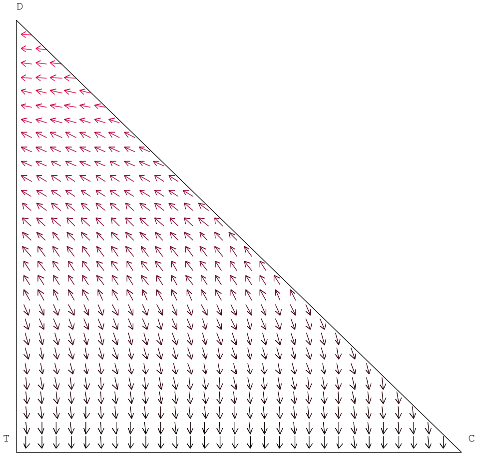
Results: Loyal Buyers and Tat for Tit Sellers ($\bar{p} = 3$)



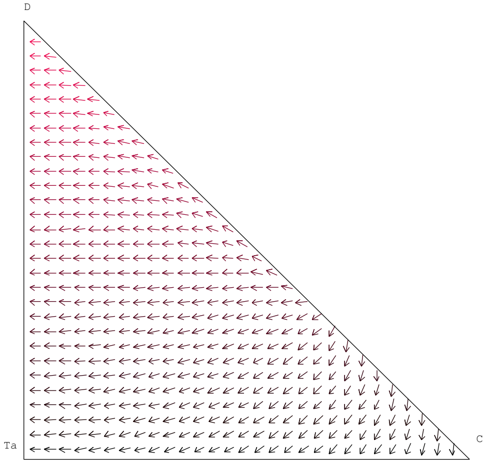
Results: Non-Loyal Buyers ($\bar{p} = 5$)



Results: Loyal Buyers ($\bar{p} = 5$)



Results: Loyal Buyers and Tat for Tit Sellers ($\bar{p} = 5$)



Results

Main conclusion

It makes sense for the consumers to be loyal for that makes the sellers behave in such a way that competition works in the best interest of the consumers, that is, they get low prices.

Comments

Results

Main conclusion

It makes sense for the consumers to be loyal for that makes the sellers behave in such a way that competition works in the best interest of the consumers, that is, they get low prices.

Comments

-
-

Results

Main conclusion

It makes sense for the consumers to be loyal for that makes the sellers behave in such a way that competition works in the best interest of the consumers, that is, they get low prices.

Comments

- Only 3 strategies for the seller at a time
-

Results

Main conclusion

It makes sense for the consumers to be loyal for that makes the sellers behave in such a way that competition works in the best interest of the consumers, that is, they get low prices.

Comments

- Only 3 strategies for the seller at a time
- Extremely rational buyers

Results

On the other hand ...

Very difficult to generalize

- What other strategies ?
- How to deal with these dynamics
- What other prices ?
- What about buyers' "rationality" ?

Results

On the other hand ...

Very difficult to generalize

- What other strategies ?
- How to deal with these dynamics
- What other prices ?
- What about buyers' “rationality” ?

Results

On the other hand ...

Very difficult to generalize

- What other strategies ?
- How to deal with these dynamics
- What other prices ?
- What about buyers' “rationality” ?

Results

On the other hand ...

Very difficult to generalize

- What other strategies ?
- How to deal with these dynamics
- What other prices ?
- What about buyers' "rationality" ?

Results

On the other hand ...

Very difficult to generalize

- What other strategies ?
- How to deal with these dynamics
- What other prices ?
- What about buyers' "rationality" ?

Outline

Strategies

Strategies represented by FINITE AUTOMATA

Sellers

Set a high price $(\bar{p})$ or a low price $(\underline{p})$ depending on the behavior of your competitor in the past

Buyers

Remain loyal to your current Seller or switch to its competitor depending on whether the price set by your current Seller is lower, equal, or higher than that of its competitor

Strategies

Strategies represented by FINITE AUTOMATA

Sellers

Set a high price $(\bar{p})$ or a low price $(\underline{p})$ depending on the behavior of your competitor in the past

Buyers

Remain loyal to your current Seller or switch to its competitor depending on whether the price set by your current Seller is lower, equal, or higher than that of its competitor

Strategies

Strategies represented by FINITE AUTOMATA

Sellers

Set a high price $(\bar{p})$ or a low price $(\underline{p})$ depending on the behavior of your competitor in the past

Buyers

Remain loyal to your current Seller or switch to its competitor depending on whether the price set by your current Seller is lower, equal, or higher than that of its competitor

Sketch of the Algorithm

Two initial populations (Sellers and buyers)

1. Randomly match two strategies in the sellers population
2. Bring each of these pairs to a simulated market
    1. The market opens R times
    2. At each round, the sellers set prices according to strategies and buyers decide seller
    3. Trade takes place and players get they payoff for the period
3. Form new populations
    1. Include the 50% top performers
    2. Create the remaining 50%
        1. Randomly select two “parents” based on performance
        2. Form two new strategies by CROSSOVER
        3. Apply MUTATION
        4. Repeat the steps above to fill the new population

Parameters

Case 3A:	$\overline{p} = 3$	$\underline{p} = 2$	$c = 0$
Case 3B:	$\overline{p} = 3$	$\underline{p} = 2$	$c = 1$
Case 3C:	$\overline{p} = 3$	$\underline{p} = 2$	$c = 1.5$
Case 4A:	$\overline{p} = 4$	$\underline{p} = 2$	$c = 0$
Case 4B:	$\overline{p} = 4$	$\underline{p} = 2$	$c = 2$
Case 4C:	$\overline{p} = 4$	$\underline{p} = 2$	$c = 3$
Case 5A:	$\overline{p} = 5$	$\underline{p} = 2$	$c = 0$
Case 5B:	$\overline{p} = 5$	$\underline{p} = 2$	$c = 3$
Case 5C:	$\overline{p} = 5$	$\underline{p} = 2$	$c = 4.5$

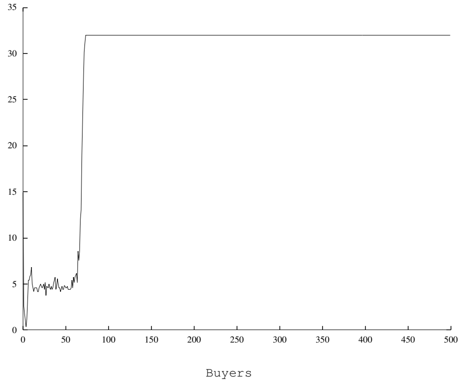
Parameters

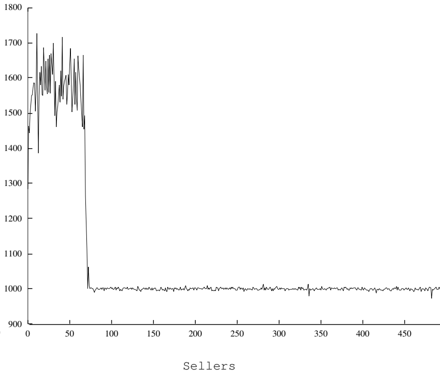
Case 3A:	$\overline{p} = 3$	$\underline{p} = 2$	$c = 0$
Case 3B:	$\overline{p} = 3$	$\underline{p} = 2$	$c = 1$
Case 3C:	$\overline{p} = 3$	$\underline{p} = 2$	$c = 1.5$
Case 4A:	$\overline{p} = 4$	$\underline{p} = 2$	$c = 0$
Case 4B:	$\overline{p} = 4$	$\underline{p} = 2$	$c = 2$
Case 4C:	$\overline{p} = 4$	$\underline{p} = 2$	$c = 3$
Case 5A:	$\overline{p} = 5$	$\underline{p} = 2$	$c = 0$
Case 5B:	$\overline{p} = 5$	$\underline{p} = 2$	$c = 3$
Case 5C:	$\overline{p} = 5$	$\underline{p} = 2$	$c = 4.5$

Parameters

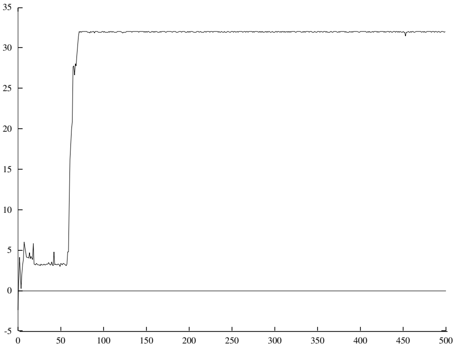
Case 3A:	$\overline{p} = 3$	$\underline{p} = 2$	$c = 0$
Case 3B:	$\overline{p} = 3$	$\underline{p} = 2$	$c = 1$
Case 3C:	$\overline{p} = 3$	$\underline{p} = 2$	$c = 1.5$
Case 4A:	$\overline{p} = 4$	$\underline{p} = 2$	$c = 0$
Case 4B:	$\overline{p} = 4$	$\underline{p} = 2$	$c = 2$
Case 4C:	$\overline{p} = 4$	$\underline{p} = 2$	$c = 3$
Case 5A:	$\overline{p} = 5$	$\underline{p} = 2$	$c = 0$
Case 5B:	$\overline{p} = 5$	$\underline{p} = 2$	$c = 3$
Case 5C:	$\overline{p} = 5$	$\underline{p} = 2$	$c = 4.5$

Results Case 3A $(\overline{p} = 3 \quad \underline{p} = 2 \quad c = 0)$

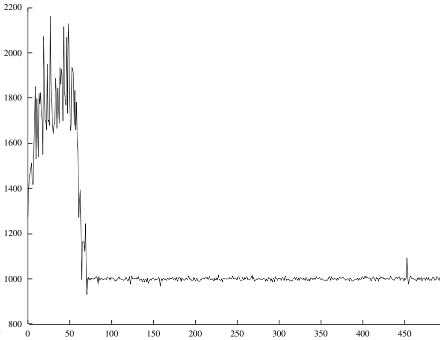




Results Case 3B $(\bar{p} = 3 \quad \underline{p} = 2 \quad c = 1)$

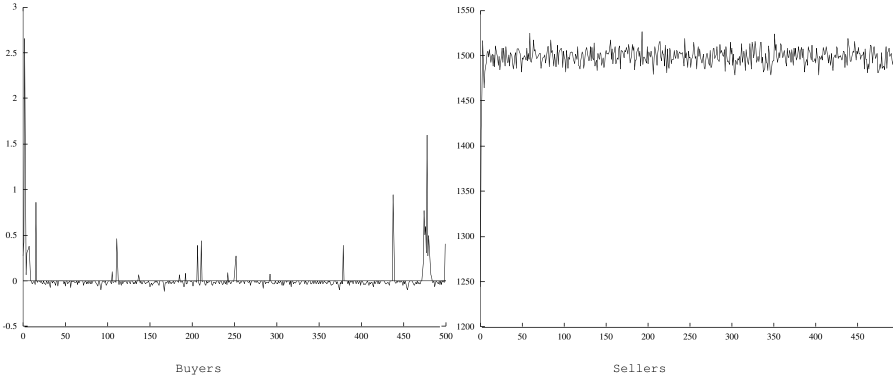


Buyers

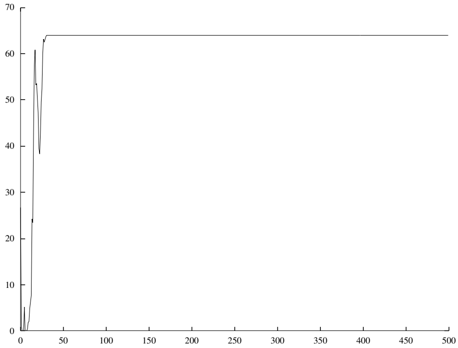


Sellers

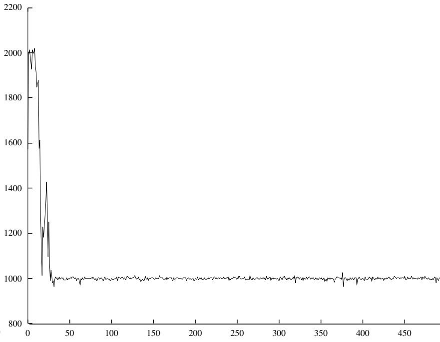
Results Case 3C $(\overline{p} = 3 \quad \underline{p} = 2 \quad c = 1.5)$



Results Case 4A $(\bar{p} = 4 \quad \underline{p} = 2 \quad c = 0)$

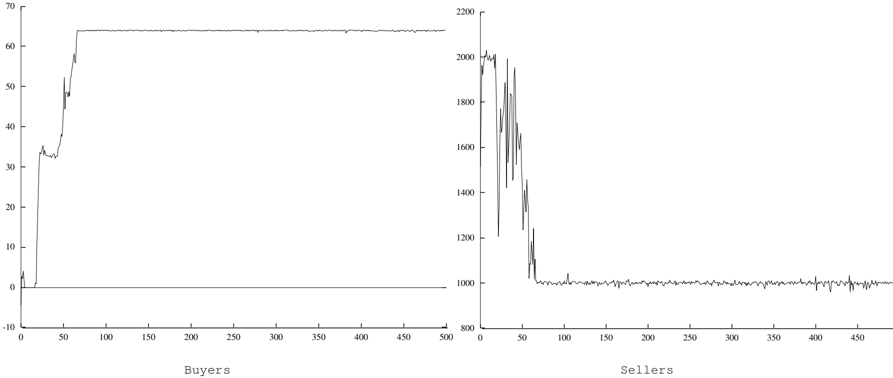


Buyers

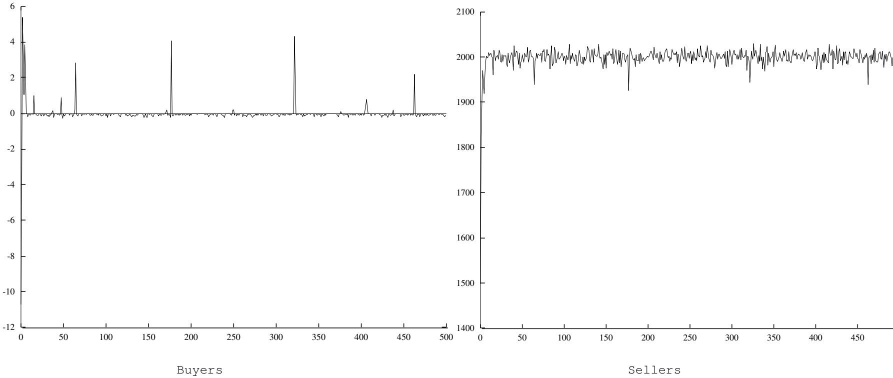


Sellers

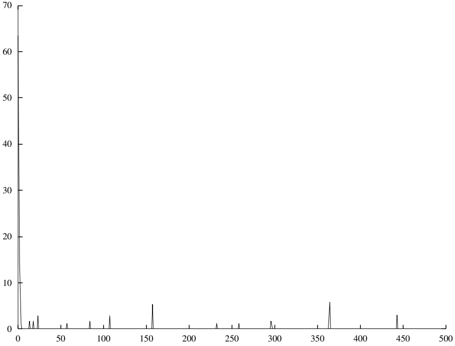
Results Case 4B $(\overline{p} = 4 \quad \underline{p} = 2 \quad c = 2)$



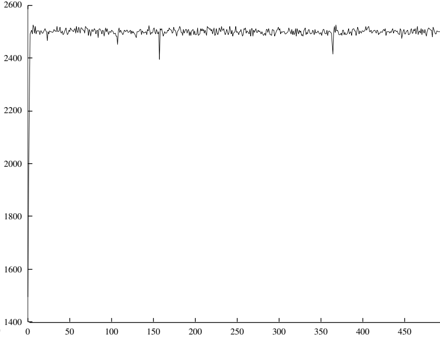
Results Case 4C $(\overline{p} = 4 \quad \underline{p} = 2 \quad c = 3)$



Results Case 5A $(\bar{p} = 5 \quad \underline{p} = 2 \quad c = 0)$

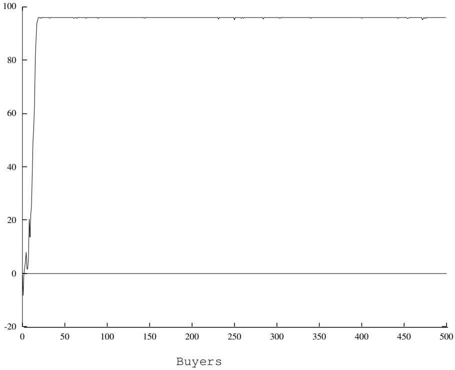


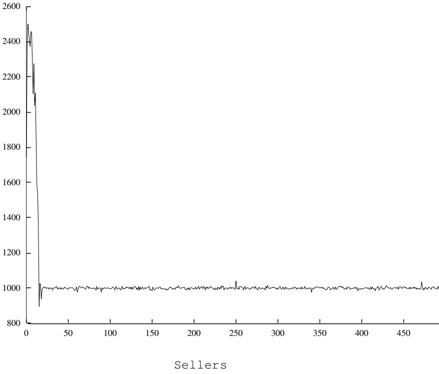
Buyers



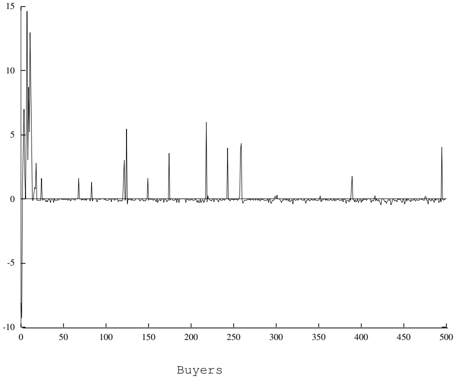
Sellers

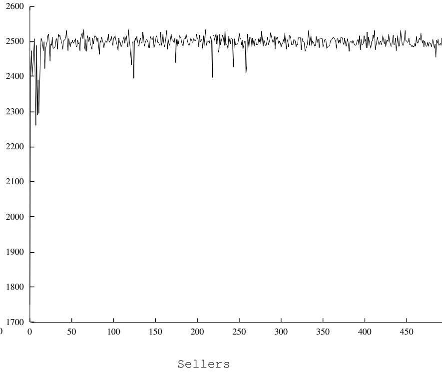
Results Case 5B $(\bar{p} = 5 \quad \underline{p} = 2 \quad c = 3)$





Results Case 5C $(\bar{p} = 5 \quad \underline{p} = 2 \quad c = 4.5)$





Results Explained (Cases *A)

The buyers will buy from the cheapest seller and will stay with their current ones if prices are equal (loyalty strategy). The sellers will set a low price regardless the price set by the opponent in the previous round (always defect strategy, D). The only difference is when the low price is too low compared to the high price that the sellers are better off by charging always the high price. (Case 5A).

Results Explained (Cases *B)

The buyers will buy from the cheapest seller and will stay with their current ones if prices are equal (loyalty strategy). The sellers will set a low price regardless the price set by the opponent in the previous round (always defect strategy, D). In this case, the fact that consumers have a switching cost equal to the difference between the two prices , does not make them “more loyal” in the sense of make them wiling to stick to a seller given that there is no possible gain by switching to another. They do not want to do that because then the sellers would have some sort of monopolistic power as the next case indicates.

Results Explained (Cases *C)

The switching cost is so high that the buyers will not switch, not even in the case that someone else offers a better price. Then, sellers take advantage of this monopolistic power and set a high price (always cooperate strategy, C).

Results

Main conclusion

It makes sense for the consumers to be loyal for that makes the sellers behave in such a way that competition works in the best interest of the consumers, that is, they get low prices.

Results

Main conclusion

Same conclusion as before in the analogous case (Case 3A). Furthermore, other cases are studied and the model can be handled with much more flexibility

Results

Comments

- Only 3 strategies for the seller at a time
- Extremely rational buyers

Results

Comments

- Many (26) strategies considered (could be more easily, nothing changes)
- Buyers and Sellers coevolve simultaneously

Summary

Summary

- Simulation techniques (Agent-Based Computational Economics) are increasingly used, but some care should be taken into account to really understand what we are simulating
- Some Agent-Based Computational Economics techniques (GA in this case) seem to perform similarly to analytical techniques yet provide ways to overcome their limitations
- Furher work should produce ways to “understand” the output of simulations

Summary

Summary

- Simulation techniques (Agent-Based Computational Economics) are increasingly used, but some care should be taken into account to really understand what we are simulating
- Some Agent-Based Computational Economics techniques (GA in this case) seem to perform similarly to analytical techniques yet provide ways to overcome their limitations
- Furher work should produce ways to "understand" the output of simulations

Summary

Summary

- Simulation techniques (Agent-Based Computational Economics) are increasingly used, but some care should be taken into account to really understand what we are simulating
- Some Agent-Based Computational Economics techniques (GA in this case) seem to perform similarly to analytical techniques yet provide ways to overcome their limitations
- Furher work should produce ways to “understand” the output of simulations

Summary

Summary

- Simulation techniques (Agent-Based Computational Economics) are increasingly used, but some care should be taken into account to really understand what we are simulating
- Some Agent-Based Computational Economics techniques (GA in this case) seem to perform similarly to analytical techniques yet provide ways to overcome their limitations
- Furher work should produce ways to "understand" the output of simulations

For Further Reading I

Goldberg, D.E.
Genetic Algorithms in Search, Optimization & Machine Learning
Addison–Wesley.(1989)

Vilà, X.
Artificial Adaptive Agents play a finitely repeated discrete Principal-Agent game.
In R. Conte, R. Hegselmann and P. Terna, eds: *Simulating Social Phenomena*. Lecture Notes in Economics and Mathematical Systems. Springer-Verlag. Berlin. (1997).

For Further Reading II

- Ferdinandova, I.
  On the Influence of Consumers' Habits on Market Structure.
  Mimeo. Universitat Autònoma de Barcelona. (2003)

- Foster, D. and H. Peyton Young.
  Cooperation in the short and the long run.
  Games and economic behavior. No. 3. (1991)

- Harrington, J. E. and Chang, Myong-Hun.
  Co-Evolution of Firms and Consumers and the Implications for Market Dominance.
  Journal of Economic Dynamics and Control (2005)

For Further Reading III

- Hehenkamp, B.
  Sluggish Consumers: An Evolutionary Solution to the Bertrand Paradox.
  Games and Economic Behavior, 40, 44-76, (2002).
- Miller, J.
  The evolution of automata in the repeated prisoner's dilemma.
  Journal of Economic Behavior and Organization. (1996).
- Rubinstein, A.
  Finite automata play the repeated prisoner's dilemma.
  Journal of Economic Theory, No. 39, (1986).