Social Behavior and Evolutionary Dynamics

Agent-based Modeling: Genetic Algorithm

M. C. Sunny Wong

University of San Francisco

University of Houston, June 18, 2015

Outline

1. Macro-Simulation
2. Background
   - What is Agent-based Modeling?
   - Genetic Algorithm - The Mechanism of Learning
3. Arifovic (1994): Cobweb Model under GA
   - Cobweb Model
   - The GA Learning
   - Conclusions
4. A Simple GA Exercise
   - A Simple Profit Maximization Problem
   - The GA Operators
   - MATLAB Codes
   - Simulations
5. Concluding Remarks

Background

What is Agent-based Modeling?

- ABM has been considered as a bottom-up approach modeling behaviors of a group of agents, rather than a representative agent, in a system.
- The representative-agent hypothesis allows for greater ease in solution procedures.
  - It is easier to find the equilibrium (relatively...).
  - This is usually called the **analytical optimization** .

Background
What is Agent-based Modeling?

- **Examples of the representative-agent models:**
  - Profit maximization, utility maximization, or cost/loss minimization...
- **Methods of optimization:**
  - (1) First-order condition - unconstrained optimization
  - (2) Lagrangian multiplier - constrained optimization
  - (3) Dynamic optimization
    - (a) Bellman equation (over discrete time), and
    - (b) Hamiltonian multiplier (over continuous time).

Background
What is Agent-based Modeling?

- LeBaron and Tesfatsion (2008, 246): "Potentially important real-world factors such as subsistence needs, incomplete markets, imperfect competition, inside money, strategic behavioral interactions, and open-ended learning that tremendously complicate analytical formulations are typically not incorporated"

Background

What is Agent-based Modeling?

- One important element of ABM is that it allows the possibility of agents' interactions in micro levels with the assumption of bounded-rationality or imperfect information.
- Given agents' heterogenous characteristics and their interactions at the micro level, we can simulate the system and observe changes in the macro level over time according to the system-simulated data.

Background
Applications of ABM

- **Poli. Sci.** (Bendor, Diermeier and Ting, APSR 2003; Fowler, JOP 2006)
  - BDT (2003):
    - A computational model by assuming that voters are adaptively rational — voters learn to vote or to stay home in a form of trial-and-error.
    - Voters are reinforced to repeat an action (e.g., vote) in the future given a successful outcome today.
    - The turnout rate is substantially higher than the predictions in rational choice models.
  - Fowler (2006):
    - He revises the BDT model by including habitual voting behavior.
    - Fowler finds his behavioral model is a better fit to the same data BDT use.

Background
Applications of ABM

- Economics
  - **Econ. Growth** - Beckenbach, et al. (JEE, 2012) - Novelty creating behavior and sectoral growth effects.
  - **Market Structure** - Alemdar and Sirakaya (JEDC, 2003) - Computation of Stackelberg Equilibria.
  - **Policy Making** - Arifovic, Bullard and Kostyshyna (EJ, 2013) - The effects of social learning in a monetary policy context.
    - The Taylor Principle is widely regarded as the necessary condition for stable equilibrium.
    - However, they show that it is not necessary for convergence to REE minimum state variable (MSV) equilibrium under genetic algorithm learning.

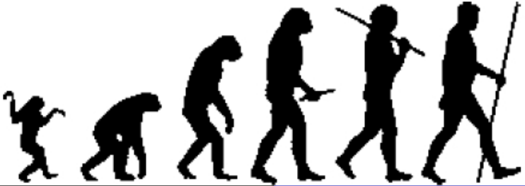
Outline

1. Macro-Simulation
2. Background
   - What is Agent-based Modeling?
   - Genetic Algorithm - The Mechanism of Learning
3. Arifovic (1994): Cobweb Model under GA
   - Cobweb Model
   - The GA Learning
   - Conclusions
4. A Simple GA Exercise
   - A Simple Profit Maximization Problem
   - The GA Operators
   - MATLAB Codes
   - Simulations
5. Concluding Remarks

Background

Genetic Algorithm - The Learning Mechanism

- The genetic algorithm (GA), developed by John Holland (1970), is considered one of the evolutionary algorithms inspired by natural evolution with a core concept of "survival of the fittest".
- The GA describes the evolutionary process of a population of genetic individuals with heterogeneous beliefs in response to the rules of nature.



This Presentation

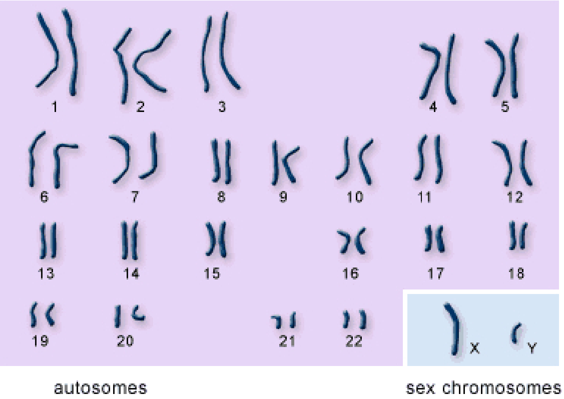
We introduce Arifovic (1994) as an example to investigate if the macro-level stability condition (the cobweb theorem) is necessary for a stable cobweb economy under GA.
We would also like to see how to apply the genetic algorithm on a simple economic model.

Important terms:

- Genes, Chromosomes, and Populations
  - Chromosomes: Genetic individuals making heterogeneous decisions
  - Genes: Elements of a decision that a genetic individual makes
  - Population: A group of genetic individuals with heterogeneous decisions

This Presentation

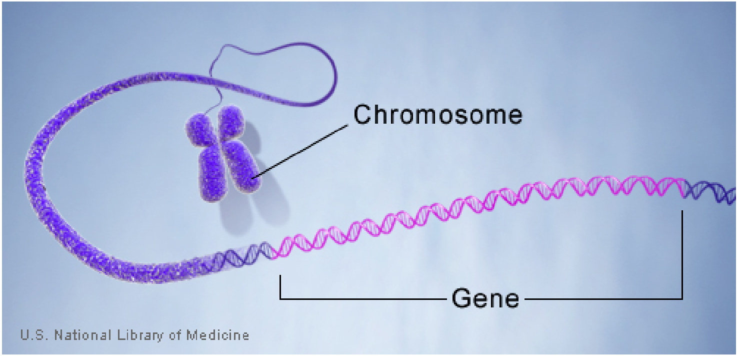
Human Chromosomes - 23 pairs



U.S. National Library of Medicine
Image © Genetics Home Reference. US National Library of Medicine: http://ghr.nlm.nih.gov

This Presentation

$$\sum DNA = Gene, \text{ and } \sum Gene = Chromosome$$



This Presentation

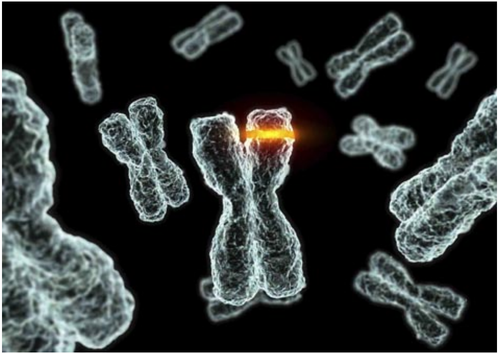
We introduce Arifovic (1994) as an example to investigate if the macro-level stability condition (the cobweb theorem) is necessary for a stable cobweb economy under GA.
We would also like to see how to apply the genetic algorithm on a simple economic model.

Important terms:

- Reproduction, Mutation, and Crossover
  - Reproduction: An individual chromosome is copied from the previous population to a new population.
  - Mutation: One or more gene within an individual chromosome changes value randomly.
  - Crossover: Two randomly drawn chromosomes exchange parts of their genes.

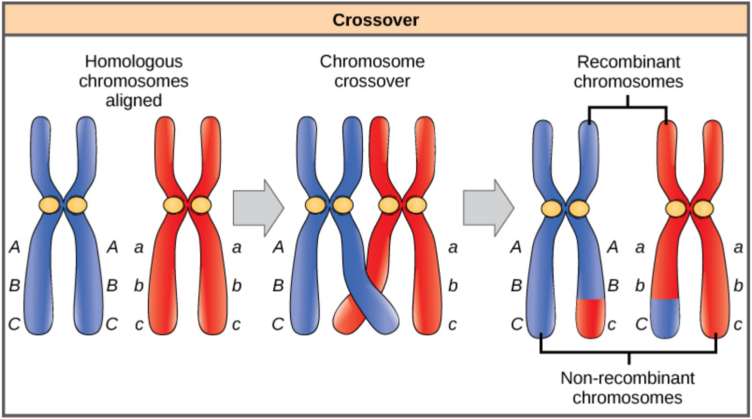
This Presentation

Genetic Mutation



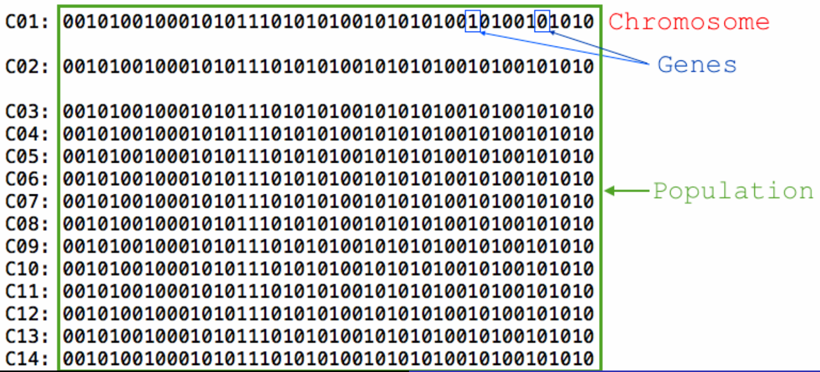
This Presentation

Genetic Crossover



Computational GA - Genes, Chromosomes, Population

The computational GA Environment can be presented as follows:



Computational GA - Mutation

The mutation which occurs when one or more gene within an individual chromosome changes value randomly: Agents may change their strategies suddenly through innovations.

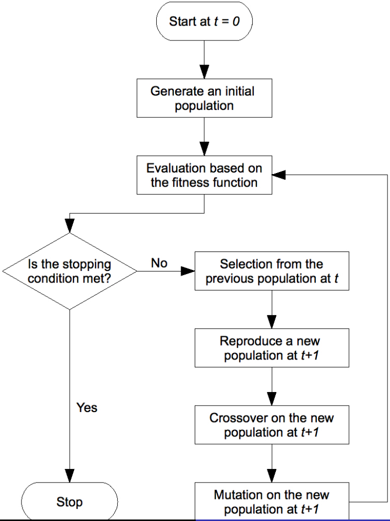
```
C01: 0010100100010101110101010010101010010100101010
C01: 001010010 0 01010111010 1 0100101010100101001010 10
C01: 001010010 1 01010111010 0 0100101010100101001010 10
```

Computational GA - Crossover

The crossover which occurs when two randomly drawn chromosomes exchange parts of their genes: *Agents work with others to innovate or develop a new strategy.*

```
C01: 0010100100010101110101010010101010010100101010

C02: 1010010101001010101001010001000101011110101000


C01: 0010100100010101110101010010 101010010100101010

C02: 1010010101001010101001010001 000101011110101000


C01: 0010100100010101110101010010 000101011110101000

C02: 1010010101001010101001010001 101010010100101010
```

Computational GA - Operational Flowchart



Outline

1 Macro-Simulation
2 Background
- What is Agent-based Modeling?
- Genetic Algorithm - The Mechanism of Learning

3 Arifovic (1994): Cobweb Model under GA
- Cobweb Model
- The GA Learning
- Conclusions

4 A Simple GA Exercise
- A Simple Profit Maximization Problem
- The GA Operators
- MATLAB Codes
- Simulations

5 Concluding Remarks

The cobweb model- An Introduction

- It is a classic model which illustrates the dynamic process of prices in **agricultural** markets (Kaldor, 1934).
- Due to a lag between planting and harvesting, farmers cannot adjust the amount of agricultural output immediately to fulfill the demand in the market.
- As a result, farmers make their planting decisions today based on the predicted (or forecasted) price of the agricultural product in the next period.
- If farmers expect the price is high in the next period, they would like to plant more today to make more money tomorrow, and vice versa. (The Law of Supply.)

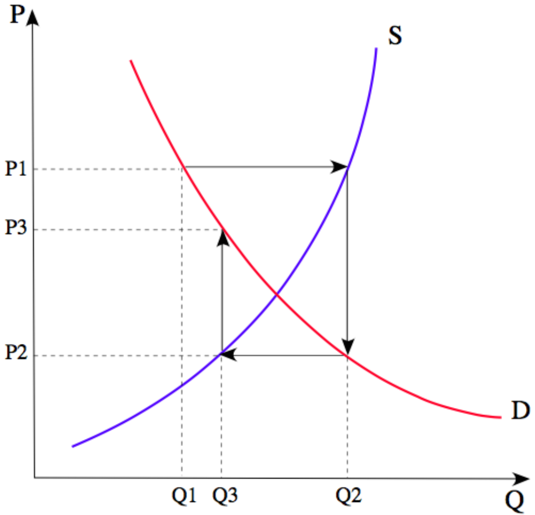
The cobweb model- An Introduction

- Assuming that farmers "forecast" the price in the next period based on the price they observe today, that is, $P^e_{t+1} = P_t$.
- If the current price level P_t is high (and is higher than the equilibrium price P^*, which is assumed to be unknown for the farmers). It can be written as: $P_{t=1} > P^*$.
    - **At time $t = 1$,** farmers would be very happy to plant more today so that they will have more output ($Q_{t=2}$) which can be sold at the high price they expect in the next period.
    - **At time $t = 2$,** since all farmers did the same in period 1, there are too much output available, which creates a "surplus" in the market, the price drops sharply at $t = 2$ due to the excess supply, and it goes below the equilibrium: $P_{t=2} < P^* < P_{t=1}$.

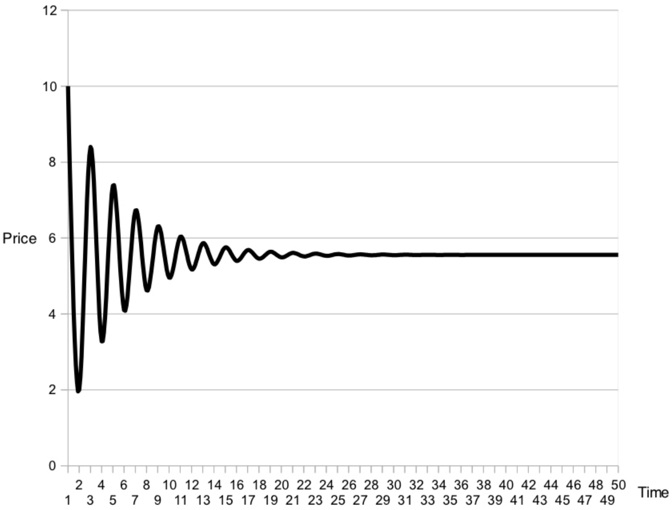
The cobweb model- An Introduction

- **What would be the planting decision for the farmers at $t = 2$?**
  - **At time** $t = 2$, since they observe the today's price is low, they would expect the price will also be low in the next period ($t = 3$). Therefore, they decide to plant less today...
  - **At time** $t = 3$, since all farmers again are doing the exact same thing, the total output level turns out to be very low this time. $Q_{s,t=3} < Q_{d,t=3}$ **(shortage!)**. Therefore, the price jumps up!
- **What would be the planting decision for the farmers at $t = 3$ now?**
  - **At time** $t = 3$, since they observe the today's price is now high again, they would expect the price will also be high in the next period ($t = 4$). Therefore, they decide to plant more today...
- This story keeps going...

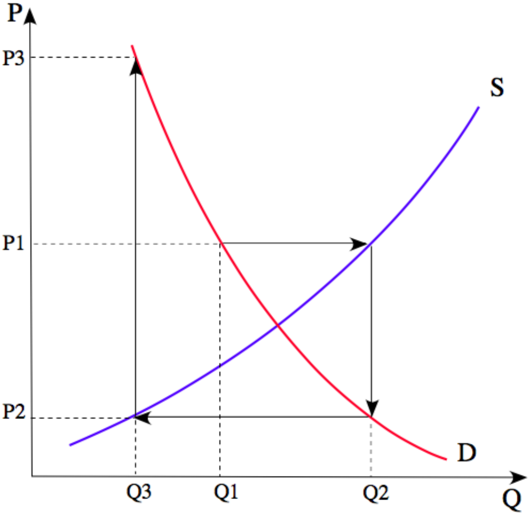
The cobweb model- An Introduction



The cobweb model- An Introduction



The cobweb model- An Introduction



The cobweb model - an mathematical illustration

- Arifovic (1994) assumes each firm i chooses a production level q_{it} to maximize its expected profit π_{it}^{e}.
- The cost function for firm i is:

$$C_{it} = aq_{it} + \frac{1}{2}bmq_{it}^{2}, \text{ where } a, b > 0.$$

- Given the expected price of the good P_t^e at time t, firm i is maximizing the following profit function:

$$\pi_{it}^{e} = P_t^e q_{it} - C_{it}\left(q_{it}\right) = P_t^e q_{it} - aq_{it} - \frac{1}{2}bmq_{it}^{2}.$$

- The first order condition for each firm i is:

$$P_t^e - a - bmq_{it} = 0 \Rightarrow q_{it} = \frac{P_t^e - a}{bm}.$$

The cobweb model - an mathematical illustration

- Assuming all firms are identical so that $q_{it} = q_t \ \forall i$, the aggregate supply in the market is:

$$Q_t = \sum_{i=1}^{m} q_{it} = mq_t = \frac{P_t^e - a}{b}, \tag{1}$$

where $m =$ number of firms in the market.

- Assuming that the market demand is a linear function:

$$P_t = \gamma - \theta Q_t, \tag{2}$$

where $Q_t = \sum q_{it}$.

- In equilibrium where (1)=(2), we can derive the following law of motion for the price level:

$$\frac{\gamma - P_t}{\theta} = \frac{P^e - a}{b} \Rightarrow P_t = \frac{\gamma b + a\theta}{b} - \frac{\theta}{b} P_t^e.$$

The Cobweb Theorem and Other Expectations Formations

- The dynamics of the price level:

$$P_t = \frac{\gamma b + a\theta}{b} - \frac{\theta}{b} P_t^e.$$

  - According to Cobweb Theorem, the model is stable if $\theta/b < 1$, that is, $\theta < b$. However, the model is unstable if $\theta/b > 1$, that is, $\theta > b$.

- Arifovic discusses three types of expectations formations:
  1. Static expectations (i.e., $P_t^e = P_{t-1}$):
     - The model is stable only if $\theta/b < 1$.
  2. Simple adaptive expectations ($P_t^e = \frac{1}{t}\sum_{s=0}^{t-1} P_s$):
     - The model is stable in both cases (Carlson, 1968).
  3. Least squares learning ($P_t^e = \beta_t P_{t-1}$, β_t = OLS coefficient):
     - The model is stable only if $\theta/b < 1$ (Bray and Savin, 1986).

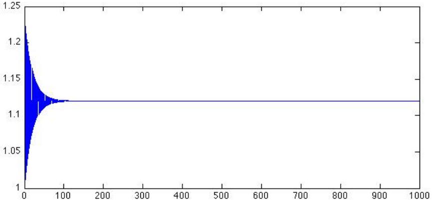
The Cobweb Theorem and Simulation

Parameters	Stable Case $\left(\frac{\theta}{b} < 1\right)$	Unstable Case $\left(\frac{\theta}{b} > 1\right)$
γ	2.184	2.296
θ	0.0152	0.0168
a	0	0
b	0.016	0.016
m	6	6
P^*	1.12	1.12
$Q^*=mq^*$	70	70

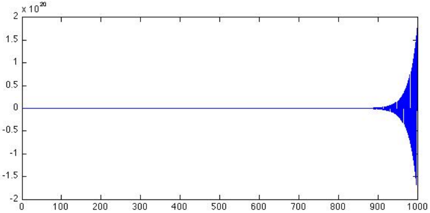
Table 12.1: Cobweb Model Parameters

The Cobweb Theorem and Simulation - Static

Static expectations (i.e., $P_t^e = P_{t-1}$):



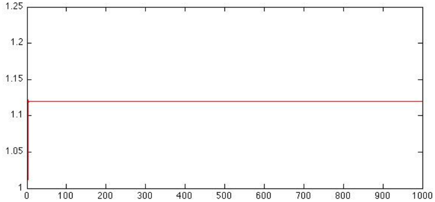
(Stable Case: $\frac{\theta}{b} < 1$)



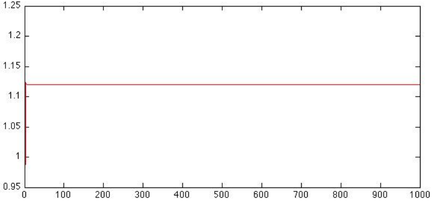
(Unstable Case: $\frac{\theta}{b} > 1$)

The Cobweb Theorem and Simulation - Adaptive

Simple adaptive expectations ($P_t^e = \frac{1}{t}\sum_{s=0}^{t-1} P_s$):



(Stable Case: $\frac{\theta}{b} < 1$)

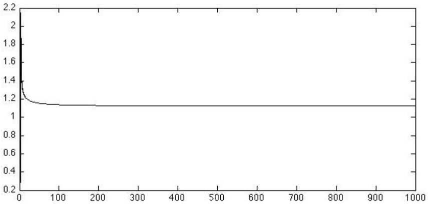


(Stable Case: $\frac{\theta}{b} > 1$)

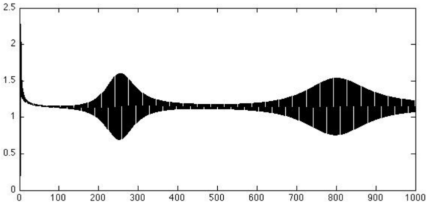
The Cobweb Theorem and Simulation - Least Squares

Least squares learning ($P_t^e = \beta_t P_{t-1}$):

(Stable Case: $\frac{\theta}{b} < 1$)



(Unstable Case: $\frac{\theta}{b} > 1$)



The Cobweb Theorem and GA

WHAT ABOUT THE GA LEARNING?

DOES THE COBWEB THEOREM HOLD UNDER THE GA?

Outline

1. Macro-Simulation
2. Background
   - What is Agent-based Modeling?
   - Genetic Algorithm - The Mechanism of Learning
3. Arifovic (1994): Cobweb Model under GA
   - Cobweb Model
   - The GA Learning
   - Conclusions
4. A Simple GA Exercise
   - A Simple Profit Maximization Problem
   - The GA Operators
   - MATLAB Codes
   - Simulations
5. Concluding Remarks

The Basic GA and Arifovic's New GA Operator

- Arifovic (1994) simulates the cobweb model based on three basic genetic operators in the GA simulations:
  - (1) *reproduction*, (2) *mutation*, and (3) *crossover*.
- She also introduces a new operator, called *election*, in the simulations.
- Election is an operator to "examine" the fitness of newly generated (or offspring) chromosomes and then compare them with their parent chromosomes.

New GA Operator - Arifovic (1991, 1994)

- The Rules of Election:
  - Both offspring chromosomes *are elected* to be in the new population at time $t+1$ if $E_t\left(V\left(C_{it+1}^{offspring}\right)\right) > V\left(C_{it}^{Parent}\right)$.
  - However, if only one new chromosome has a higher fitness value than their parents, the one with lower value will not enter the new population, but one of the parents with a higher values stays in the new population.
  - If both new chromosomes have lower values than their parents $E_t\left(V\left(C_{it+1}^{offspring}\right)\right) < V\left(C_{it}^{Parent}\right)$, they cannot enter but their parents stay in the new population.

GA Learning Parameters

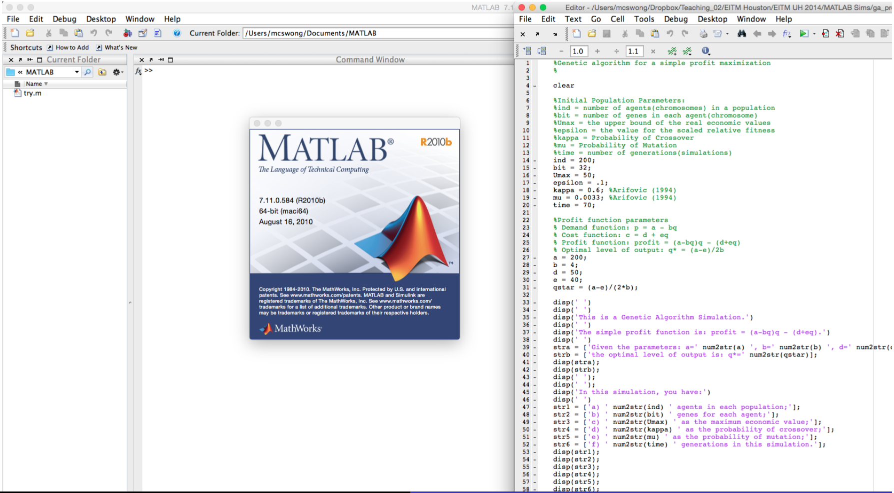
Parameters	Stable Case $(\frac{\theta}{b} < 1)$	Unstable Case $(\frac{\theta}{b} > 1)$
γ	2.184	2.296
θ	0.0152	0.0168
a	0	0
b	0.016	0.016
m	6	6
P^*	1.12	1.12
$Q^*=mq^*$	70	70

Table 12.1: Cobweb Model Parameters

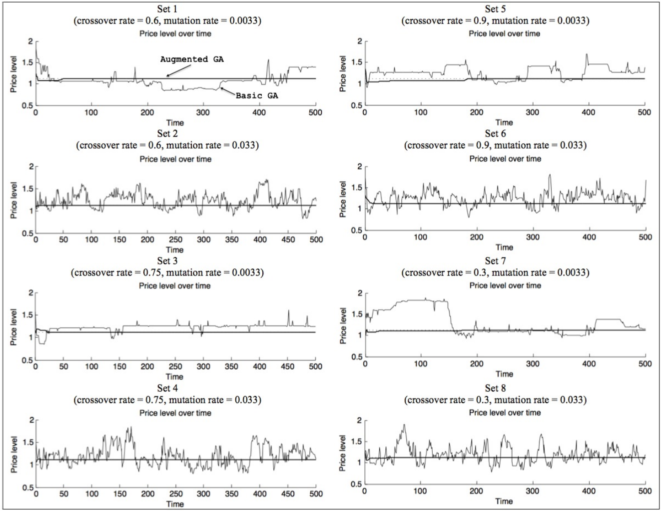
Set	1	2	3	4	5	6	7	8
Crossover rate: κ	0.6	0.6	0.75	0.75	0.9	0.9	0.3	0.3
Mutation rate: μ	0.0033	0.033	0.0033	0.033	0.0033	0.033	0.0033	0.033

Table 12.2: Crossover and Mutation Rates

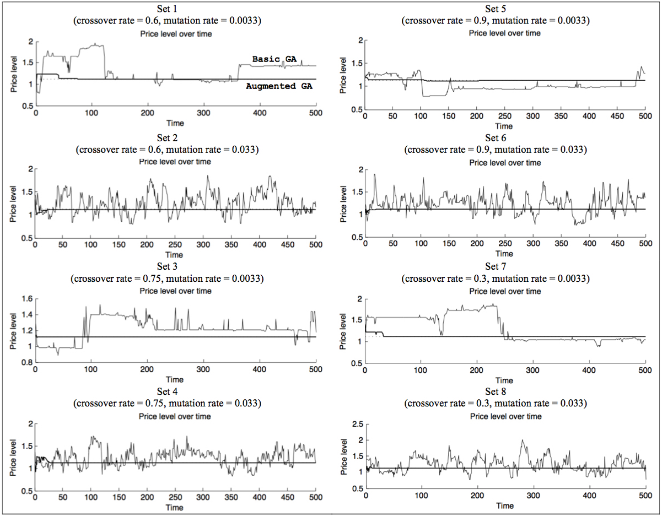
Running Simulations - MATLAB



The GA Simulations - Stable Case ($\theta/b < 1$)



The GA Simulations - Unstable Case ($\theta/b > 1$)



Outline

1. Macro-Simulation
2. Background
   - What is Agent-based Modeling?
   - Genetic Algorithm - The Mechanism of Learning
3. Arifovic (1994): Cobweb Model under GA
   - Cobweb Model
   - The GA Learning
   - Conclusions
4. A Simple GA Exercise
   - A Simple Profit Maximization Problem
   - The GA Operators
   - MATLAB Codes
   - Simulations
5. Concluding Remarks

Conclusions

- Arifovic (1994) introduces the GA procedure as an alternative learning mechanism.
- This alternative learning mechanism mimics social behavior:
  - imitation, communication, experiment, and examination.
- Arifovic uses the GA simulated data to compare with the data generated in human-subject experiments (Wellford, 1989).
  - In an unstable case of the cobweb model, the divergent patterns *do not* happen under both GA learning and human-subject experiments.
  - Price and quantity fluctuate around the equilibrium in *basic* GA learning and human-subject experiments.

Outline

1. Macro-Simulation
2. Background
   - What is Agent-based Modeling?
   - Genetic Algorithm - The Mechanism of Learning
3. Arifovic (1994): Cobweb Model under GA
   - Cobweb Model
   - The GA Learning
   - Conclusions
4. A Simple GA Exercise
   - A Simple Profit Maximization Problem
   - The GA Operators
   - MATLAB Codes
   - Simulations
5. Concluding Remarks

Profit Maximization

1. Profit function: $\pi = p \times q - c(q)$.
2. Demand: $p = a - bq$.
3. Supply (cost function): $c = d + eq$.
4. Maximizing profit: $\max_q \pi = (a - bq)\, q - (d + eq)$.
5. Optimal level of output: $q^* = (a - e)/2b$.

Outline

1. Macro-Simulation
2. Background
   - What is Agent-based Modeling?
   - Genetic Algorithm - The Mechanism of Learning
3. Arifovic (1994): Cobweb Model under GA
   - Cobweb Model
   - The GA Learning
   - Conclusions
4. A Simple GA Exercise
   - A Simple Profit Maximization Problem
   - The GA Operators
   - MATLAB Codes
   - Simulations
5. Concluding Remarks

Notations under the GA

- Chromosome C_i consists of a set of 0 and 1, where L is the length of a chromosome (the number of genes).
- $B^{max}(C_i) = 2^L - 1$ represents the maximum numerical value of a chromosome with the length L.
  - For example, if $L = 10$, the maximum value of a chromosome:

$$B(1111111111) = 2^{10} - 1 = 1023.$$

- We can use the B operator to compute a numerical value of a chromosome (e.g., $C_i = 0100101110$) :

$$\begin{aligned} B(0100101110) &= 0 \times 2^9 + 1 \times 2^8 + 0 \times 2^7 + 0 \times 2^6 + \\ & 1 \times 2^5 + 0 \times 2^4 + 1 \times 2^3 + 1 \times 2^2 + \\ & 1 \times 2^1 + 0 \times 2^0 = 302. \end{aligned}$$

Notations under the GA

- Assume that there are $M = 8$ genetic individuals. For $L = 10$, we can generate an initial genetic population P_0 in an $M \times L$ matrix (that is, 8×10 matrix):
- For example:

$$P_0 = \begin{matrix} 0100101110 \\ 1110101010 \\ 0101110100 \\ 0100001010 \\ 1110101000 \\ 0101101101 \\ 1100101010 \\ 0100011100 \end{matrix}$$

Notations under the GA

- According to the problem of profit maximization, if $a = 200$, $b = 4$, and $e = 40$, then $q^* = 20$.
- In this case, the maximum value of a chromosome can be too large for this problem $(B^{max} = 1023)$.
- We can define a maximum economic value for a chromosomes $V(C_i)$ based on the following value function:

$$V(C_i) = \frac{U^{max}}{B^{max}} \times B(C_i),$$

where $V(C_i) \in [0, U^{max}]$ for $B(C_i) \in [0, B^{max}]$, and U^{max} is the maximum economic value in the problem.

Notations under the GA

- An economic value for a chromosomes $V(C_i)$ based on the following value function:

$$V(C_i) = \frac{U^{max}}{B^{max}} \times B(C_i).$$

- For example, given the maximum output level is $U^{max} = 100$, and $C_i = 0100101110$ (i.e., $B(C_i) = 302$), we can calculate the output level for firm i:

$$q_i = V(C_i) = \frac{100}{1023} \times 302 = 29.52 \approx 30.$$

Notations under the GA

- Is firm i doing a good job? We need to evaluate firm i using a fitness function $F(C_i)$.
- The profit function is used as the fitness function in this case:

$$\begin{aligned} F(C_i) &= \pi(V(C_i)) \\ &= \pi(q_i) = (a - bq_i)\,q_i - (d + eq_i)\,. \end{aligned}$$

- In this case,

$$\begin{aligned} F(C_i) &= \pi(V(C_i)) \\ &= \pi(29.52) = (200 - 4(29.52))(29.52) - (50 + 40(29.52)) \\ &= 1187.48. \end{aligned}$$

- The maximum profit is (for $q* = 20$):

$$F^{max} = \pi(q^*) = \pi(20) = 1550.$$

The GA Operators

Reproduction ⇒ Evolutionary Dynamics

- Reproduction is a genetic operator where an individual chromosome is copied from the previous population to a new population.
- The probability of being drawn for each chromosome is calculated based on the fitness value.
  - Higher fitness value ⇒ higher probability of being drawn to the new population.
- The relative fitness function is:

$$R(C_{i,t}) = \frac{F(C_{i,t})}{\sum_{m=1}^{M} F(C_{m,t})},$$

  where $\sum_{i \in M} R(C_{i,t}) = 1$.
- The relative fitness value $R(C_{i,t})$ gives us the probability chromosome i is copied to the new population at time $t+1$.

The GA Operators
Reproduction

- What if $F(C_{i,t})$ is negative for some firm i? (a negative profit?)
- Goldberg (1989) proposes a scaled relative fitness function:

$$S(C_{i,t}) = \frac{F(C_{i,t}) + A}{\sum_{m=1}^{M}[F(C_{m,t}) + A]} = \frac{F(C_{i,t}) + A}{\sum_{m=1}^{M} F(C_{m,t}) + MA},$$

where A is a constant such that $A > -\min_{C_i \in P_t} F(C_{i,t})$.

The GA Operators

Crossover

- A crossover point will be randomly chosen to separate each chromosome into two sub-strings.
- Two "offspring" chromosomes will be formed by swapping the right-sided parents' substrings with probability κ.

```
C01: 00101001000101011101010100101010100101001010
C02: 10100101010010101010010100010001010111101010000
```

```
C01: 0010100100010101110101010010 101010010100101010
C02: 1010010101001010101001010001 0001010111110101000

C01: 0010100100010101110101010010 0001010111110101000
C02: 1010010101001010101001010001 101010010100101010
```

The GA Operators

Crossover

Assuming that there are $M = 6$ individuals in the population (each chromosome has 20 genes) :

```
[6x20] matrix
C01: 10010100100110101010
C02: 10101010010001101100
C03: 01101100101000110110
C04: 11011001010001110100
C05: 10110010111101100101
C06: 10110101111011001010
```

The GA Operators
Crossover

Therefore, there are $20 - 1 = 19$ possible positions for crossover. We randomly pick a position for each pair of chromosomes.

```
Break the population into 3 groups.
Randomly pick a position between Position 1 and Position 19

C01: 10010100100110101010
C02: 10101010010001101100

C03: 01101100101000110110
C04: 11011001010001110100

C05: 10110010111101100101
C06: 10110101111011001010

_______________________________________
```

The GA Operators

Crossover

Given $\kappa = 0.3$, the position for the 1st pair is 8, the 2nd pair is 3, and the 3rd is 0.

```
C01: 100101001001_10101010 [Position 8]
C02: 101010100100_01101100

C03: 01101100101000110_110 [Position 3]
C04: 11011001010001110_100

C05: 10110010111101100101  [Position 0]
C06: 10110101111011001010
______________________________________
```

The GA Operators

Crossover

This is a new population after crossover.

```
C01: 100101001001_01101100 [Position 8]
C02: 101010100100_10101010

C03: 0110110010100011O_100 [Position 3]
C04: 11011001010001110_110

C05: 101100101111011001O1_ [Position 0] – NO CROSSOVER
C06: 10110101111011001010_
_______________________________________
```

The GA Operators

Mutation

- Every gene within a chromosome has a small probability, μ, changing in value, independent of other positions.

```
C01: 0010100100010101110101010010101010010100101010

C01: 001010010 0 01010111010 1 010010101010010100101010

C01: 001010010 1 01010111010 0 010010101010010100101010
```

Outline

1. Macro-Simulation
2. Background
   - What is Agent-based Modeling?
   - Genetic Algorithm - The Mechanism of Learning
3. Arifovic (1994): Cobweb Model under GA
   - Cobweb Model
   - The GA Learning
   - Conclusions
4. A Simple GA Exercise
   - A Simple Profit Maximization Problem
   - The GA Operators
   - MATLAB Codes
   - Simulations
5. Concluding Remarks

Defining Parameter Values

```
%Initial Population Parameters:
%ind = number of agents(chromosomes) in a population
%bit = number of genes in each agent(chromosome)
%Umax = the upper bound of the real economic values
%epsilon = the value for the scaled relative fitness
%kappa = Probability of Crossover
%mu = Probability of Mutation
%time = number of generations(simulations)
ind = 200;
bit = 32;
Umax = 50;
epsilon = .1;
kappa = 0.6; %Arifovic (1994)
mu = 0.0033; %Arifovic (1994)
time = 500;

%Profit function parameters
% Demand function: p = a - bq
% Cost function: c = d + eq
% Profit function: profit = (a-bq)q - (d+eq)
% Optimal level of output: q* = (a-e)/2b
a = 200;
b = 4;
d = 50;
e = 40;
qstar = (a-e)/(2*b);
```

Creating an Initial Population

```
%Value Function and Definitions
Bmax = (2 .^ bit) - 1;
m = ind;
n = bit;

%Generate the Initial Population: "gen"
gen = rand(m,n);
for i=1:m
    for j=1:n
        if gen(i,j)<.5;
            gen(i,j)=0;
        else
            gen(i,j)=1;
    end
end
end
```

Converting Binary Value into Numerical Value

```
%Calculate the real value of each chromosome: "BC"
m2 = 2 * ones(n,1);
for i=1:n
    m2(i,1)=m2(i,1).^(n-i);
end

BC = ones(m,1);
for i=1:m
    BC(i,1)=gen(i,:) * m2; %Converting Binary # to Decimal # for each i
end

```

- For example,

$$\begin{aligned} B(0100101110) = \; & 0\times 2^9 + 1\times 2^8 + 0\times 2^7 + 0\times 2^6 + \\ & 1\times 2^5 + 0\times 2^4 + 1\times 2^3 + 1\times 2^2 + \\ & 1\times 2^1 + 0\times 2^0 = 302. \end{aligned}$$

Notations under the GA

- An economic value for a chromosomes $V(C_i)$ based on the following value function:

$$V(C_i) = \frac{U^{max}}{B^{max}} \times B(C_i).$$

- For example, given the maximum output level is $U^{max} = 100$, and $C_i = 0100101110$ (i.e., $B(C_i) = 302$), we can calculate the output level for firm i:

$$q_i = V(C_i) = \frac{100}{1023} \times 302 = 29.52 \approx 30.$$

Notations under the GA

- Is firm i doing a good job? We need to evaluate firm i using a fitness function $F(C_i)$.
- The profit function is used as the fitness function in this case:

$$\begin{aligned} F(C_i) &= \pi(V(C_i)) \\ &= \pi(q_i) = (a - bq_i)\, q_i - (d + eq_i)\,. \end{aligned}$$

- In this case,

$$\begin{aligned} F(C_i) &= \pi(V(C_i)) \\ &= \pi(29.52) = (200 - 4(29.52))(29.52) - (50 + 40(29.52)) \\ &= 1187.48. \end{aligned}$$

- The maximum profit is (for $q* = 20$):

$$F^{max} = \pi(q^*) = \pi(20) = 1550.$$

The GA Operators

Reproduction ⇒ Evolutionary Dynamics

- Reproduction is a genetic operator where an individual chromosome is copied from the previous population to a new population.
- The probability of being drawn for each chromosome is calculated based on the fitness value.
  - Higher fitness value ⇒ higher probability of being drawn to the new population.
- The relative fitness function is:

$$R(C_{i,t}) = \frac{F(C_{i,t})}{\sum_{m=1}^{M} F(C_{m,t})},$$

where $\sum_{i \in M} R(C_{i,t}) = 1$.
- The relative fitness value $R(C_{i,t})$ gives us the probability chromosome i is copied to the new population at time $t+1$.

The GA Operators

Reproduction

```
%This is the code for Reproduction
norm_fit = SC
selected = rand(size(SC))
sum_fit = 0;
for i=1:length(SC)
    sum_fit = sum_fit + norm_fit(i)
    index = find(selected<sum_fit)
    selected(index) = i*ones(size(index))
end
gen = gen(selected,:)
```

Goldberg (1989) proposes a scaled relative fitness function:

$$S(C_{i,t}) = \frac{F(C_{i,t}) + A}{\sum_{m=1}^{M} [F(C_{m,t}) + A]} = \frac{F(C_{i,t}) + A}{\sum_{m=1}^{M} F(C_{m,t}) + MA},$$

where A is a constant such that $A > -\min_{C_i \in P_t} F(C_{i,t})$.

The GA Operators

Reproduction

```
>> norm_fit = SC

norm_fit =

    0.1283
    0.1230
    0.1182
    0.0000
    0.1276
    0.0785
    0.0780
    0.1271
    0.0927
    0.1266

>>   selected

selected =

    10
     1
     3
     5
     1
     1
     1
    10
    10
     5
```

The GA Operators

Crossover

```
%This is the code for Crossover (Point & Pairwise)
%size(gen,1) = ind = number of individual
%size(gen,2) = bit = number of genes
sites = ceil(rand(size(gen,1)/2,1)*(size(gen,2)-1))
sites = sites.*(rand(size(sites))<kappa)
for i = 1:length(sites)
    newgen([2*i-1 2*i],:) = [gen([2*i-1 2*i],1:sites(i)) ...
                                 gen([2*i 2*i-1],sites(i)+1:size(gen,2))]
end
gen=newgen
```

The GA Operators

Crossover

```
>> rand(size(gen,1)/2,1)

ans =

    0.6378
    0.3878
    0.8372
    0.7663
    0.1256

>> size(gen,2)-1

ans =

    31

>> ceil(rand(size(gen,1)/2,1)*(size(gen,2)-1))

ans =

     3
    21
    12
     4
    20
```

The GA Operators

Mutation

```
%This is the code for Mutation
mutated = find(rand(size(gen))<mu)
newgen = gen
newgen(mutated) = 1-gen(mutated)
gen=newgen;
ngen=newgen;
```

The GA Operators

Mutation

```
mutated =

     3
    43


newgen =

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


newgen =

     1     0     1     1     1     1     1     0
     1     1     0     0     1     0     0     1
     1     1     1     1     1     0     1     1
     0     1     1     1     0     0     1     1
     0     1     0     0     0     1     0     1
     0     0     0     1     1     1     0     1
     0     1     1     1     0     0     1     1
     0     1     1     1     0     0     1     1
     0     1     0     0     0     1     0     1
     1     0     0     0     0     0     0     0
```

Outline

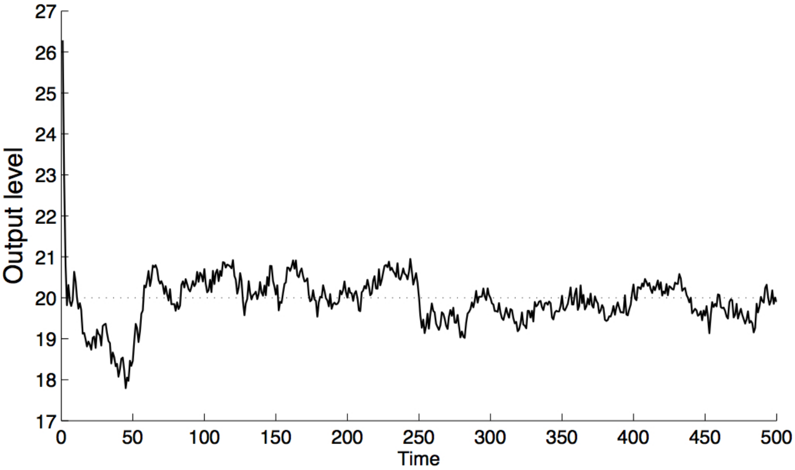
1. Macro-Simulation
2. Background
   - What is Agent-based Modeling?
   - Genetic Algorithm - The Mechanism of Learning
3. Arifovic (1994): Cobweb Model under GA
   - Cobweb Model
   - The GA Learning
   - Conclusions
4. A Simple GA Exercise
   - A Simple Profit Maximization Problem
   - The GA Operators
   - MATLAB Codes
   - Simulations
5. Concluding Remarks

The Basic GA Simulations

- Market Parameters:
  - Demand: $a = 200$, and $b = 400$.
  - Supply: $d = 50$, and $e = 40$.
  - Optimal output: $q^* = 20$.
- GA Parameters:
  - $M = 200$ (200 genetic agents)
  - $L = 16$, therefore $B^{max} = 65535$.
  - $U^{max} = 50$ (maximum output $q^{max} = 50$)
  - $\kappa = 0.3$ (probability of crossover)
  - $\mu = 0.0033$ (probability of mutation)
  - $t = 500$ (500 generations)

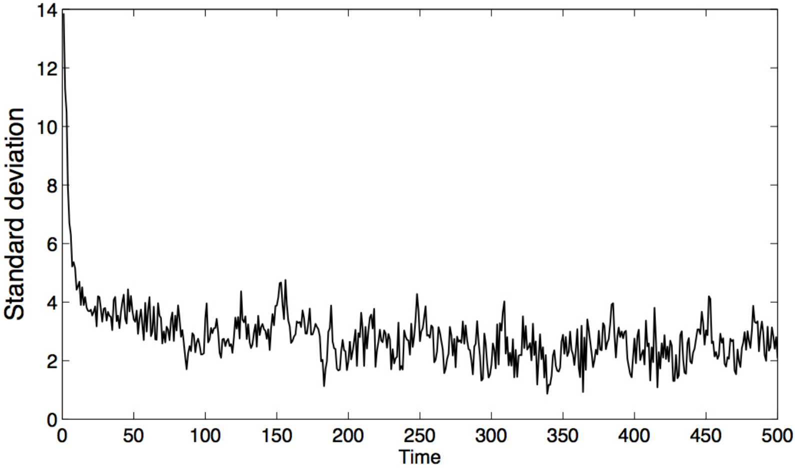
The Basic GA Simulations

The Output Level over time



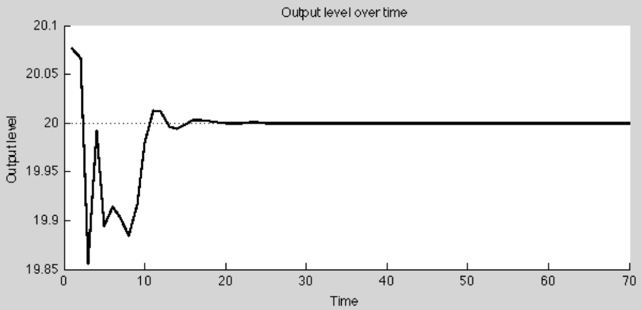
The Basic GA Simulations

The Standard Deviation of Output Level over time

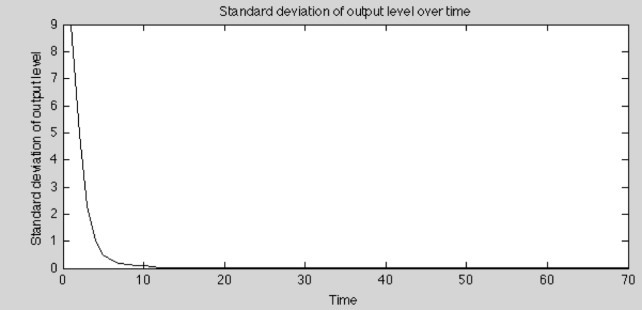


The Augmented GA Simulations

The Output Level over time



The Augmented GA Simulations
The Standard Deviation of Output Level over time

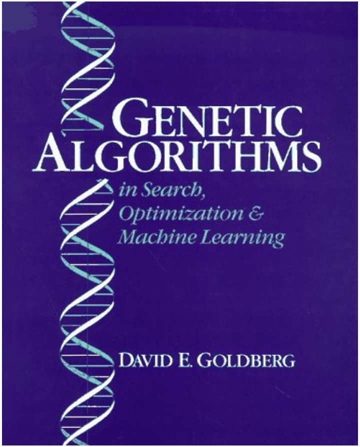


Concluding Remarks

- Why do we use the GA (or ABM in general) for political science / economics research??
  - Some models are mathematically intractable (we cannot find a closed-form equilibrium).
  - No strong assumptions imposed (such as, efficient markets, rational agents, representative agent hypothesis).
  - It allows non-linearity in a theoretical model.
  - It is relatively easier to capture equilibrium (equilibria) in a multi-national, multi-sector model.

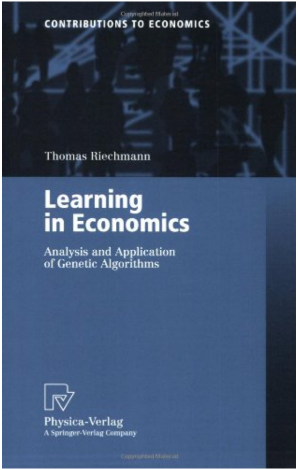
Learn GA Learning?

Genetic Algorithms in Search, Optimization, and Machine Learning (David E. Goldberg, 1989)



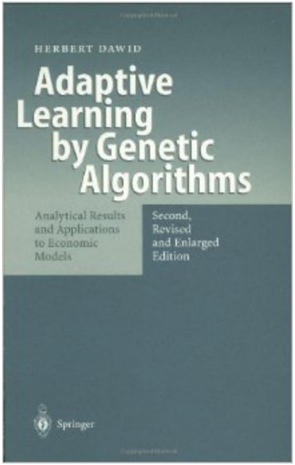
Learn GA Learning?

Learning in Economics: Analysis and Application of Genetic Algorithms (Thomas Riechmann, 2001)



Learn GA Learning?

Adaptive Learning by Genetic Algorithms: Analytical Results and Applications to Economic Models (Herbert Dawid, 2012)



Concluding Remarks

Thank You.

Questions?

Sources of Figures

- Evolutionary figure: http://mme.uwaterloo.ca/~fslien/ga/ga.html
- Human chromosome: http://ghr.nlm.nih.gov/handbook/illustrations/chromosomes.jpg
- Genetic mutation: http://farm3.static.flickr.com/2350/1583336323_33661151a2_o.jpg
- Genetic crossover: http://cnx.org/content/m45471/latest/Figure_08_03_06.jpg