Using Reinforcement Learning to analyse a simple market model

Carlos Enmanuel Soto López

Jun 20, 2022

Index

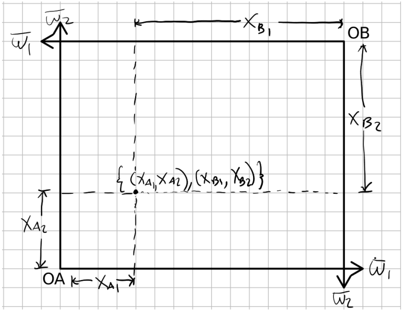
1. Market model
   - Edgeworth box economy
   - Market rules
2. From the Edgeworth box economy to a RL problem
   - Motivation
   - Elements of the RL problem
   - Algorithm
3. Results
   - Results
   - More to study

Index

1. Market model
   - Edgeworth box economy
   - Market rules

Edgeworth box economy

The Edgeworth box economy is an simple economic model with two consumers and two goods. An allocation in this economy can be represented in a graphical way using the Edgeworth box[1].



[1] Mas-Colell et al., 1995

Index

1. Market model
   - Edgeworth box economy
   - Market rules

Market rules

The rules of the market can be summarised as,

- Denoting by $\{A, B\}$ the consumers, x_{il} the total amount of good l that consumer i posses, w_{il} the endowments, the initial amount of good l that consumer i posses and $\bar{w}_l$ the total amount of good l, no production, no consumption and no waste is represented by the equation,

$$x_{Al} + x_{Bl} = w_{Al} + w_{Bl} = \bar{w}_l. \tag{1}$$

Market rules

- All the Allocations that obey equation (1) are called feasible allocations. Given a set of prices (p_1, p_2), each consumer can exchange their initial goods with the budget constrain

$$p_1 x_{i1} + p_2 x_{i2} = p_1 w_{i1} + p_2 w_{i2}. \tag{2}$$

- Each consumer is trying to maximise their utility function $u_i(x_{i1}, x_{i2})$ under the budget constrain. The utility function is a function that describes how happy is the consumer given an endowment.

Index

Motivation

Given a utility function $u_A(x_{A1}, x_{A2})$, a set of prices (p_1, p_2) and an initial endowment (w_{A1}, w_{A2}), there exist a feasible allocation such that the consumer A maximise his utility function subject to his budget constrain. For an arbitrary set of prices, this allocation is different from the allocation that maximises $u_B(x_{B1}, x_{B2})$. A Walraisan or competitive equilibrium for the Edgeworth box economy is a price vector p^* and an Allocation, such that both utility functions are maximised under their respective budget constrains. The problem is that in rare actions, a consumer has actual knowledge of the others utility functions or even about hers own utility function. Is more realistic to think this problem as a reinforcement learning problem, with no knowledge of the world.

Index

2 From the Edgeworth box economy to a RL problem

- Motivation
- Elements of the RL problem
- Algorithm

To translate the Edgeworth box economy to a Reinforcement Learning (RL) problem, it is necessary to identified the elements of the RL problem,

- **World:** The world consist of the utility functions of each consumer $u_A(x_{A1}, x_{A2})$, $u_B(x_{B1}, x_{B2})$, and the set of all possible allocations and prices $\{(x_{A1}, x_{A2}), (x_{B1}, x_{B2}), (p1, p2)\}$ such that equation (1) is obeyed. For this work, a discrete representation of the allocation was used.

- **Actions:** Given a state $s = \{(x_{A1}, x_{A2}), (x_{B1}, x_{B2}), (p_1, p_2)\}$, each consumer can choose between six actions, which are, selling good 2 at price p_2, selling good 2 at price $p_2 + \epsilon$, selling good 2 at price $p_2 - \epsilon$, buying good 2 at price p_2, buying good 2 at price $p_2 + \epsilon$ and buying good 2 at price $p_2 - \epsilon$. ϵ is chosen in such a way that the amount of good 2 sold of bought is an integer number of the discretization chosen.
- **Rewards:** The reward given an action is the difference between the utility function given the previous allocation and the new value of the utility function.

Index

2. From the Edgeworth box economy to a RL problem
    - Motivation
    - Elements of the RL problem
    - Algorithm

Algorithm

The algorithm to be used was an actor-critic algorithm, based on a natural stochastic gradient ascendant method.

- Input $\Pi_1(a|s)$, $\Pi_2(a|s)$, $\hat{V}_1(s, w_1)$, $\hat{V}_2(s, w_2)$, α_θ and α_w.
- Initialize θ_1, θ_2, w_1 and w_2,$\gamma_t = 1$.
- Loop over episodes:
  - Initialize a state s.
  - Loop over time:
    - Pick $A = (a_1, a_2)$ according to the policy $\Pi(A|s) = \Pi_1(a_1|s)\Pi_2(a_2|s)$.
    - Observe $\hat{s}, u_1(\hat{s}), u_2(\hat{s})$.
    - Compute the temporal difference error for each player $\delta_i = u_i(\hat{s}) - u_i(s) + \gamma\hat{V}_i(\hat{s}, w) - \hat{V}_i(s, w)$.
    - Modify the parameters, $w_i = w_i + \alpha_w \delta_i s$, $\theta_i = \theta_i + \alpha_\theta \delta_i \mathbf{I}(a, s)/\Pi_i(a|s)$,$\gamma_t = \gamma_t\gamma$, $s = \hat{s}$.

Index

3 Results

- Results
- More to study

Results

The algorithm was implemented, at https://github.com/carlossoto362/QLS2021-2022Diploma/blob/main/RL_proyect_git/proyect.py

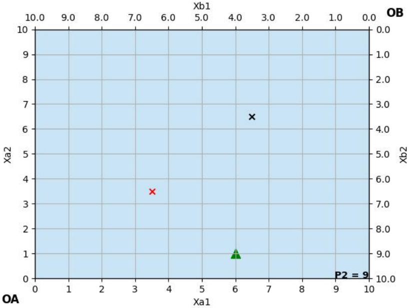


Figure: Representation of one state in the Edgeworth box. The green triangle is the Allocation, the red mark is the point where consumer A maximises his utility function given his budget constrain, and the black mark for the consumer B.

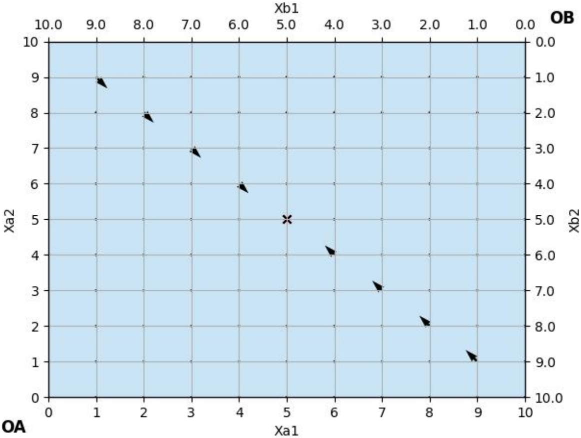
Results

Because of the size of the states space, it was decided to work with a simplified subset of the problem, with a price fixed and a symmetric initial endowment, in such a way that, if the two consumers have the same utility function, the market equilibrium is realised in the center of the edgeworth box, and the dynamics occur only on the diagonal opposed to the origins. In this simplified case, the optimal policy would be the one that exchange always in the direction of the equilibrium, and don't exchange when equilibrium is reached.

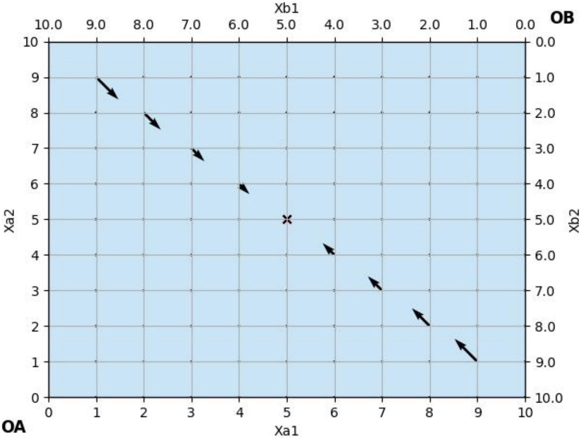
Results

Using learning rates of $\alpha_\theta = 0.0009$ adn $\alpha_w = 0.001$, and episodes of 10000 steps...

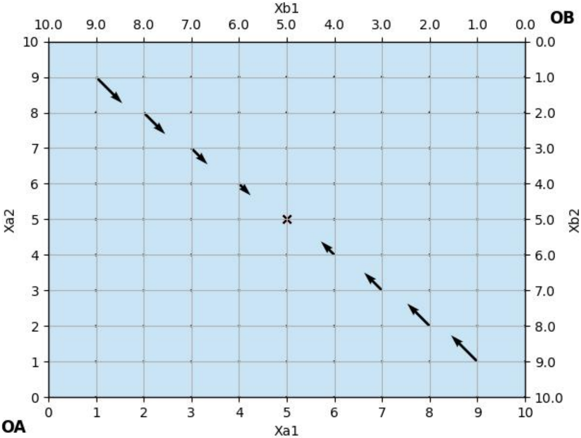
Results


Xb1
10.0
9.0
8.0
7.0
6.0
5.0
4.0
3.0
2.0
1.0
0.0
OB
Xa2
10
9
8
7
6
5
4
3
2
1
0
Xb2
0.0
1.0
2.0
3.0
4.0
5.0
6.0
7.0
8.0
9.0
10.0
0
1
2
3
4
5
6
7
8
9
10
Xa1
OA

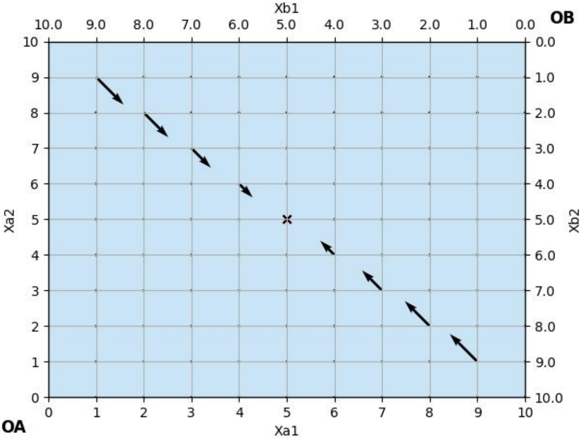
Results


Xb1
10.0
9.0
8.0
7.0
6.0
5.0
4.0
3.0
2.0
1.0
0.0
OB
Xa2
10
9
8
7
6
5
4
3
2
1
0
Xb2
0.0
1.0
2.0
3.0
4.0
5.0
6.0
7.0
8.0
9.0
10.0
0
1
2
3
4
5
6
7
8
9
10
Xa1
OA

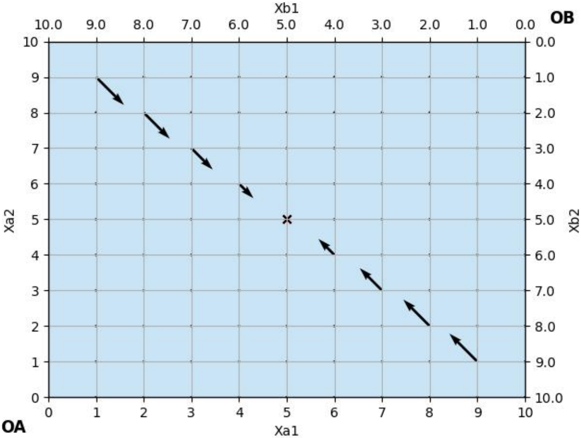
Results


Xb1
10.0
9.0
8.0
7.0
6.0
5.0
4.0
3.0
2.0
1.0
0.0
OB
Xa2
10
9
8
7
6
5
4
3
2
1
0
Xb2
0.0
1.0
2.0
3.0
4.0
5.0
6.0
7.0
8.0
9.0
10.0
OA
0
1
2
3
4
5
6
7
8
9
10
Xa1

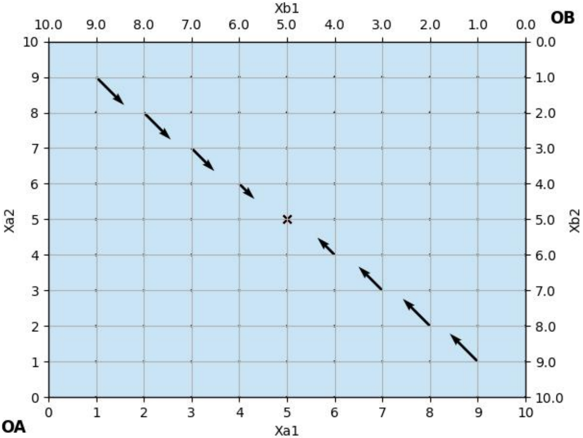
Results


Xb1
10.0
9.0
8.0
7.0
6.0
5.0
4.0
3.0
2.0
1.0
0.0
OB
Xa2
10
9
8
7
6
5
4
3
2
1
0
Xb2
0.0
1.0
2.0
3.0
4.0
5.0
6.0
7.0
8.0
9.0
10.0
0
1
2
3
4
5
6
7
8
9
10
Xa1
OA

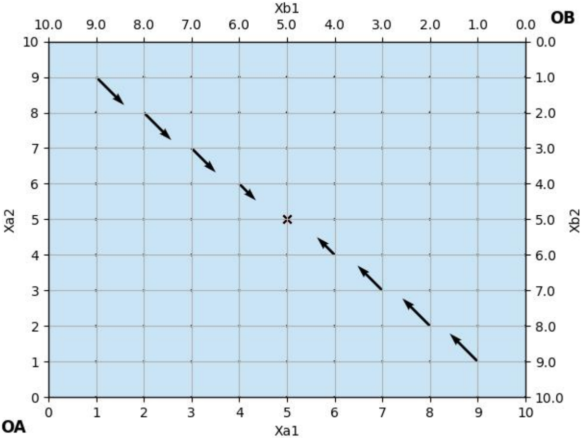
Results


Xb1
10.0
9.0
8.0
7.0
6.0
5.0
4.0
3.0
2.0
1.0
0.0
OB
Xa2
10
9
8
7
6
5
4
3
2
1
0
Xb2
0.0
1.0
2.0
3.0
4.0
5.0
6.0
7.0
8.0
9.0
10.0
0
1
2
3
4
5
6
7
8
9
10
Xa1
OA

Results


Xb1
10.0
9.0
8.0
7.0
6.0
5.0
4.0
3.0
2.0
1.0
0.0
OB
Xa2
10
9
8
7
6
5
4
3
2
1
0
0.0
1.0
2.0
3.0
4.0
5.0
6.0
7.0
8.0
9.0
10.0
Xb2
0
1
2
3
4
5
6
7
8
9
10
OA
Xa1

Results

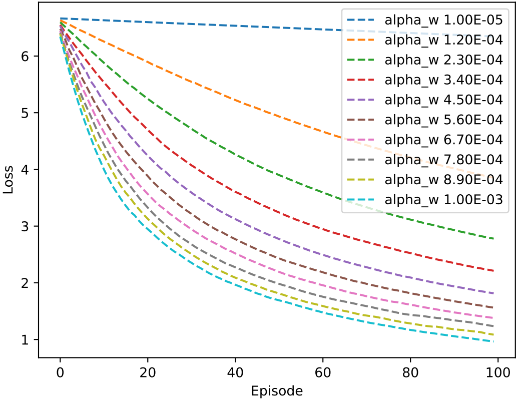

Xb1
10.0
9.0
8.0
7.0
6.0
5.0
4.0
3.0
2.0
1.0
0.0
OB
Xa2
10
9
8
7
6
5
4
3
2
1
0
Xb2
0.0
1.0
2.0
3.0
4.0
5.0
6.0
7.0
8.0
9.0
10.0
OA
0
1
2
3
4
5
6
7
8
9
10
Xa1

Results

In order to know which value to use for the learning rates, the loss function was defined, as the difference between the value of the vest policy and the value of the present policy, as a function of the number of episodes used to learn. If the actor-critic algorithm would be used in this market, it was found:

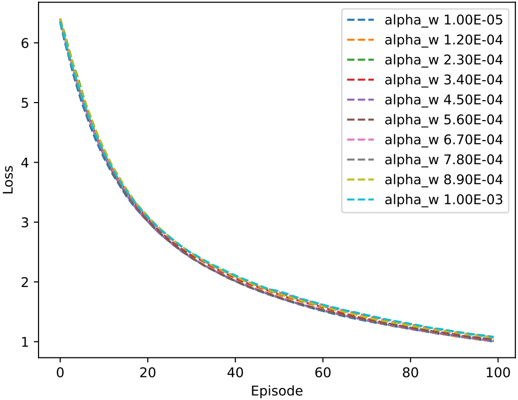
Results

Existence of an optimal learning rate α_θ...



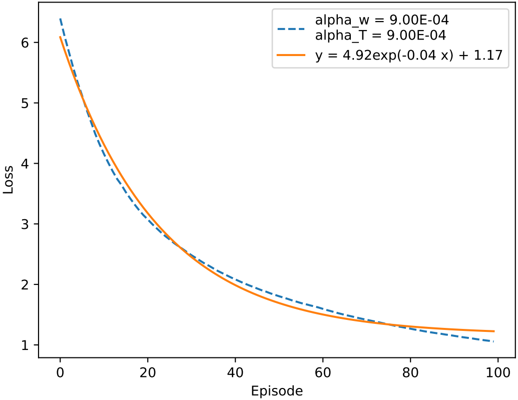
Results

Independence of learning rate α_w...



Results

and Bias in the policy...



Index

3 Results

- Results
- More to study

More to study

- Come back to the general case. Does the customers learn an optimal policy? Does an optimal policy exist?
- Customers with noisy utility functions.
- It is possible to include production in this frame?
- Much more...

bibliography

Mas-Colell, A., Whinston, M. D., & Green, J. R. (1995). *Microeconomic theory*. Oxford University Press.