Lydia Teinfalt, 3/04/2025, HW5: Code the bilateral exchange for *A* agents and 2 goods
specs: Python 3x on Google Collab and Spyder

Consumer (agents) class instantiated with Cobb-Douglas preferences with α, β randomly chosen from the list of values = 0.25, 0.33, 0.5. Consumer's initial endowment x_1 is randomly initialized with possible values from 1 to K and x_2 with possible values 1 to L. The parameters, K and L, are global variables representing the dimensions of the Edgeworth box and maximum amounts of goods 1 and goods 2 that can be traded between consumer 1 and consumer 2.

The Population class is instantiated with N agents. The model selects two consumers randomly and determines if a trade is possible. Trade is possible if given the two selected consumers do not have the equal marginal rates of substitutions (MRS) (Foundations of ABM, p. 67)

Cobb-Douglas preferences (see appendix G), i.e.,

$$U^i(x_1^i, x_2^i) = (x_1^i)^{\alpha^i}(x_2^i)^{1-\alpha^i}$$

From this we can compute the marginal rates of substitution:

$$MRS_{12}^i(\boldsymbol{x}^i) = \frac{\dfrac{\partial U^i(\boldsymbol{x}^i)}{\partial x_1^i}}{\dfrac{\partial U^i(\boldsymbol{x}^i)}{\partial x_2^i}} = \frac{\alpha^i x_2^i}{(1-\alpha^i)x_1^i}.$$

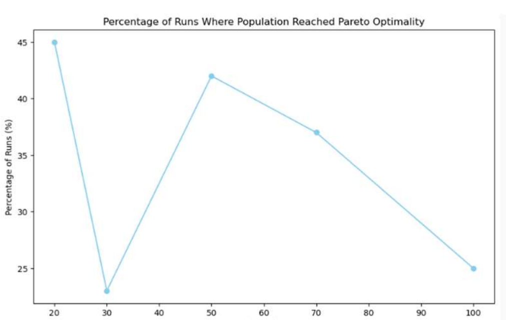
If exchange is possible, the trade is executed by randomly selected a new $x_1$between consumer 1's x_1 initial endowment and consumer 2's x_1 initial endowment. A new x_2 is derived from a contract curve between the two consumers based on the following formula (1) (Foundations of ABM, p. 76)

Solving this for x_2^i as a function of x_1^i gives

$$x_2^i = \frac{\alpha^j(1-\alpha^i)x_1^i x_2^T}{\alpha^i x_1^T(1-\alpha^j) - x_1^i(\alpha^i - \alpha^j)} \qquad (1)$$

Simulation creates a population of N consumers trading two goods and have agents trade until the population reaches Pareto Optimality or the maximum number 1000 trades. Simulation is run 10 times for any given number of N agents.

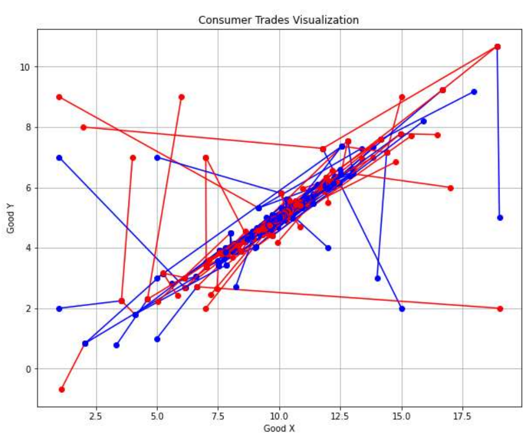
Starting with a population of 10 consumers, I increased the number of runs to determine if it would help the Population reach Pareto Optimality. The number of runs reaches peak around 45 runs and drops significantly.



Then with 45 runs, I increased the number of consumers, and I hypothesized it will be more difficult for the population to reach equilibrium. The following table shows that my initial hypothesis is not true. I ran this multiple times with the following table is a representation of one run and did not detect correlation between number of agents decreasing the likelihood of optimality.

Number of Agents	Number of Runs	% Runs with Pareto Optimal Solution	Avg Number of Trades
10	45	31%	691.02
20	45	27%	737.18
50	45	29%	721.55
70	45	24%	768.0
100	45	40%	629.36

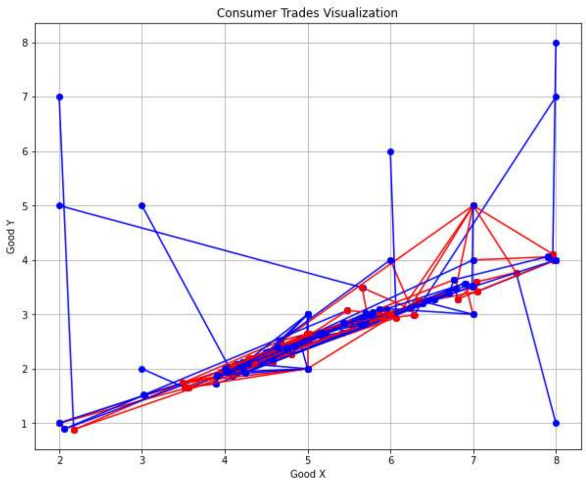
I experimented with the initial values of goods 1 and goods 2 of the Edgeworth box. I initially had initial values of [10,10]. I increased these values to [15,15], [20, 20], and [25,25] the population could always reach equilibrium solution. However, if I modified where the amount of goods 1 was not equal to goods 2, it was a challenge for the population to reach Pareto Optimal solution. There must be a bug in the logic or code. One example of Edgeworth box [20,15] population of 20 consumers finding Pareto optimal solution in 419 trades. Once I creased the number of goods to [25,15], the same 20 consumers were unable to reach equilibrium.



Visualization of Trades for Selected Runs

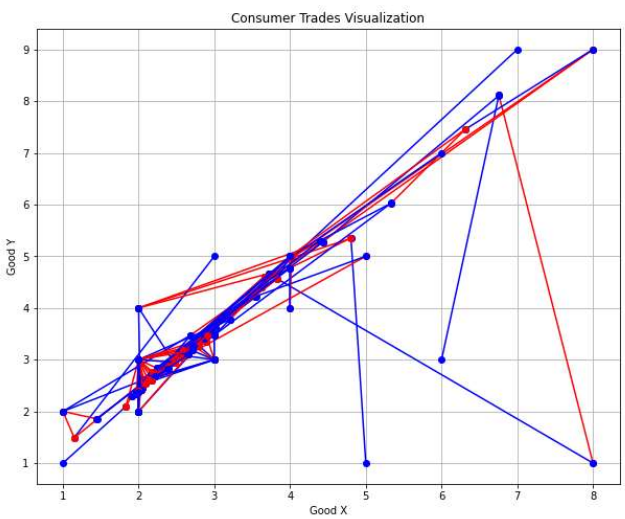
Number of Agents = 10

Run #3 reached Pareto Optimality
Pareto Optimality achieved after trade
Pareto Optimality reached at iteration 107
Number of trades = 107

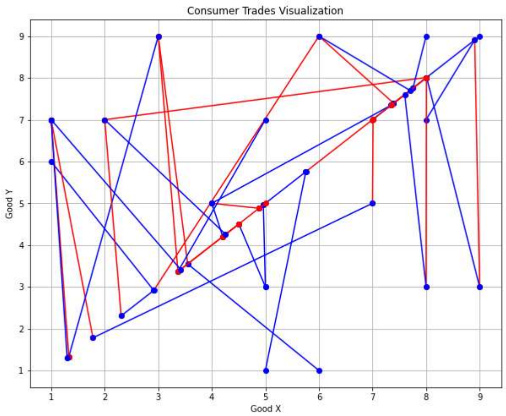


****

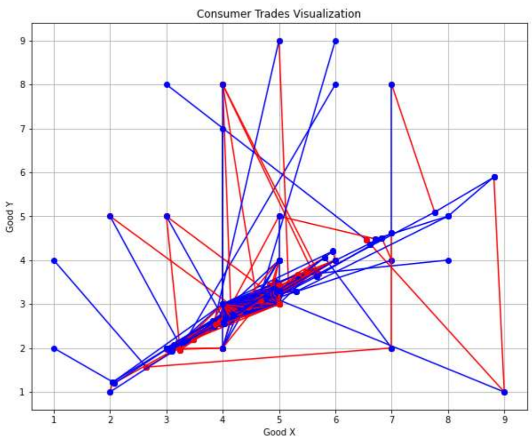
Run #4 did not find pareto optimal solution
Number of agents 10 and max number of iterations 1000
Number of trades = 1000



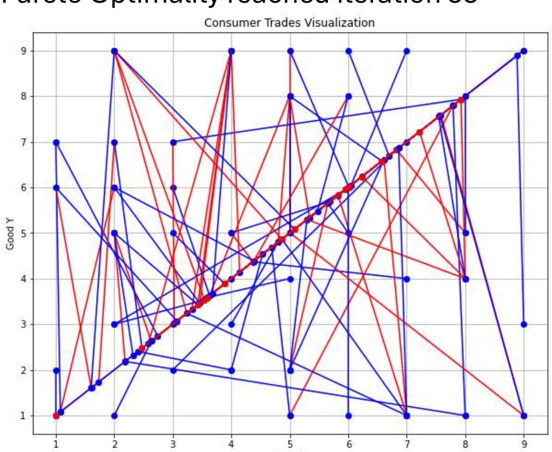
20 Agents
Run #1 Reached Pareto Optimality
Pareto Optimality reached at iteration 23
Number of trades = 23



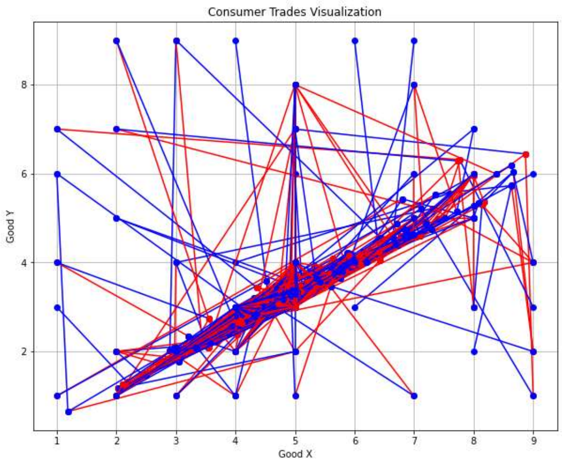
Run #2 did not reach Pareto Optimality after 1000 Trades



50 Agents
Run #10 Reached Pareto Optimality
Pareto Optimality reached iteration 58

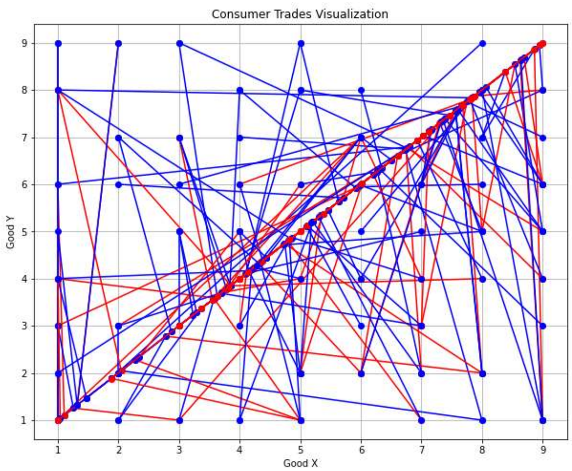


Run #9 did not reach Pareto Optimality after 1000 Trades

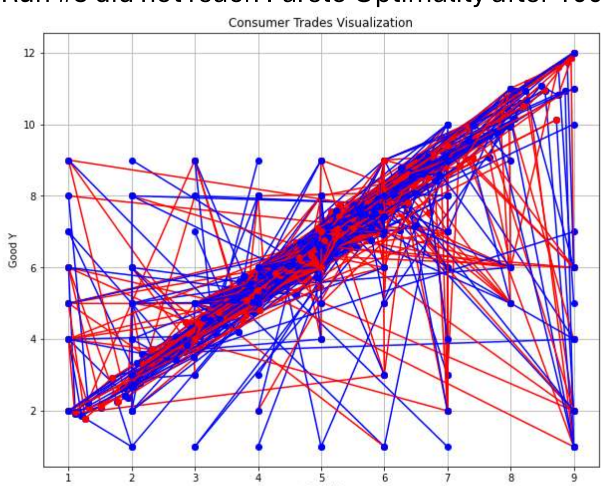


100 Agents

Run #2 Pareto Optimality achieved at 113 Trades



Run #3 did not reach Pareto Optimality after 1000 Trades



Code repository

https://github.com/lydiateinfalt/CSS610-AgentBasedModelingSimulation-Spring2025/blob/main/EdgeworthBox.py