Generalized Bilateral Exchange

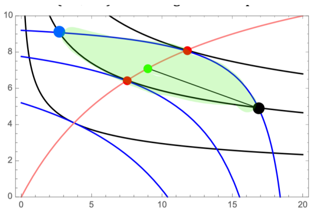
John Wong

Agenda

1. Motivation
2. Demonstration
3. Psuedo-code
    3.1 Uniform activation of pairwise combinations
4. Shuffling
5. Extensions
6. More visualizations

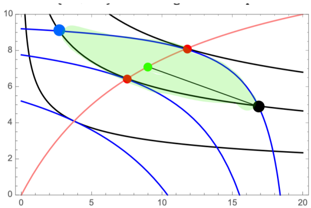
Motivation

- Two agents trading two goods is already pretty gnarly to solve.



Motivation

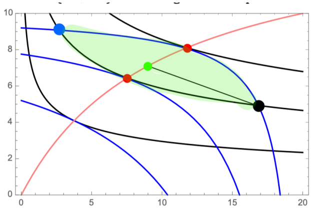
- Two agents trading two goods is already pretty gnarly to solve.



- What if we increase the number of agents to three? Or four? Or any arbitrary A? And what if we increase the number of goods to N?

Motivation

- Two agents trading two goods is already pretty gnarly to solve.



- What if we increase the number of agents to three? Or four? Or any arbitrary A? And what if we increase the number of goods to N?
  - You would need a numeraire (i.e., reference good), e.g., money.
  - Each agent would need to know their demand curve at every price (i.e., for every unit of the numeraire).
  - Then each agent would need to submit this to an auctioneer, who centrally sets price so that quantity demanded is equal to the fixed supply.

Demonstration

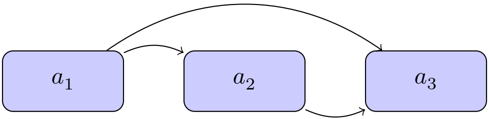
Pseudo-code

```
initialize Market(agents = A, goods = N):
  generate goods list
  for 1:A:
    initialize agent
    generate N random elasticities that sum to 1
    generate N random inventories
---------------------------------------------
```

Pseudo-code

```
initialize Market(agents = A, goods = N):
  generate goods list
  for 1:A:
    initialize agent
    generate N random elasticities that sum to 1
    generate N random inventories
-----------------------------------------------
execute exchange(days = 1):
  shuffle agents list and goods list
  FOR a in 1:(A-1):
    FOR p in 1:(A-a):
      pair agent a with agent a+p
      FOR n in 1:(N-1):
        FOR q in 1:(N-n):
          agent with higher MRS(n, n+q) gets good n
          other agent gets good n+q
          WHILE trade increases both agents' utilities:
            trade one good n for one good n+q
```

Uniform activation of pairwise combinations (A=3)



There are total $\frac{A^2-A}{2}$ combinations—or the upper triangle of an $A \times A$ matrix.

Shuffling

1. Agents are stored in a list of agents that is never shuffled.

$$\begin{bmatrix} a_1 & a_2 & a_3 & a_4 & a_5 & a_6 & a_7 & a_8 & a_9 & a_{10} \end{bmatrix}$$

Shuffling

1. Agents are stored in a list of agents that is never shuffled.

$$[a_1 \quad a_2 \quad a_3 \quad a_4 \quad a_5 \quad a_6 \quad a_7 \quad a_8 \quad a_9 \quad a_{10}]$$

2. We create a duplicate list *of indices*.

$$[1 \quad 2 \quad 3 \quad 4 \quad 5 \quad 6 \quad 7 \quad 8 \quad 9 \quad 10]$$

3. Shuffle list of indices:

$$[4 \quad 9 \quad 6 \quad 1 \quad 10 \quad 2 \quad 8 \quad 5 \quad 3 \quad 7]$$

Shuffling

1. Agents are stored in a list of agents that is never shuffled.

$$[a_1 \quad a_2 \quad a_3 \quad a_4 \quad a_5 \quad a_6 \quad a_7 \quad a_8 \quad a_9 \quad a_{10}]$$

2. We create a duplicate list *of indices*.

$$[1 \quad 2 \quad 3 \quad 4 \quad 5 \quad 6 \quad 7 \quad 8 \quad 9 \quad 10]$$

3. Shuffle list of indices:

$$[4 \quad 9 \quad 6 \quad 1 \quad 10 \quad 2 \quad 8 \quad 5 \quad 3 \quad 7]$$

4. Retrieve agent (a_4, a_9), then (a_4, a_6), and so forth.

Extensions: random strategies

```
execute exchange(days = 1, threshold = 0.5):
  shuffle agents list and goods list
  FOR a in 1:(A-1):
    FOR p in 1:(A-a):
      pair agent a with agent a+p
      FOR n in 1:(N-1):
        FOR q in 1:(N-n):
          agent with higher MRS(n, n+q) gets good n
          other agent gets good n+q
          WHILE trade increases both agents' utilities:
            draw u from U[0,1]
            IF u < threshold:
              break
            trade one good n for one good n+q
```

Extensions: networks

```
initialize Market(..., friends = 3):
  ...
  create RelationshipChart
  for 1:A:
    sample agents list for 3 friends
    store friends' indices in dictionary
    append dictionary to RelationshipChart
execute exchange(days = 1):
  shuffle agents list and goods list
  FOR a in 1:A:
    FOR p in RelationshipChart[a]:
      FOR n in 1:N
        FOR q in 1:(N-n):
          ...
```

More visualizations!

1. Inventory over time
2. Utilities over time
3. Multilateral Edgeworth box