Economics through TCS lens

How Can We Apply our TCS Knowledge to Economics??

- Supply Chain Logistics and Operations Research
  - Linear Programming, Integer Programming, Convex Optimization
- Computing Market Equilibrium:
  - Hardness of finding Nash equilibrium
- Mechanism Design
  - Polynomial Time Algorithms that are incentive compatible
- Combinatorial Auctions
- Fair allocation of goods
- Learning and Regret Minimization

Nobel Prize Winners:

- LINEAR PROGRAMMING
  - Leonid Kantorovich (1975)
- STABLE MATCHING
  - David Gale, Lloyd Shapley, Alvin Roth (2012)

Ordinal vs Cardinal Preferences

Ordinal Preferences:

- Each agent has a preference list over goods

	First Choice	Second Choice	Third Choice
John	House 1	House 2	House 3
Jane	House 3	House 1	House 2
Henry	House 2	House 1	House 3

Ordinal vs Cardinal Preferences

Cardinal Preferences:

- Each agent has some utility over each good

	House 1	House 2	House 3
John	55	46	85
Jane	66	73	0
Henry	50	50	50

One Sided Matching markets

  

- Agent Cares about which house it gets

- House doesn't care at all

Two Sided Matching Markets

- Hospital and Residents
- Students and Schools
- Kidney Exchange
- Uber
- Speed Dating

Both parties care about whom they're matched too!!!

Our Example: Investor Game (Shark Tank)

- N investors
- N innovators
- Each investor has a strict preference list over all innovators
- Each innovator has a strict preface list over all investors

GOAL: Come up with an assignment of investors to innovators that is good (??)

Kevin	John	Jane	Henry
Mark	John	Henry	Jane
Lori	Jane	John	Henry

Jane	Kevin	Mark	Lori
John	Kevin	Mark	Lori
Henry	Kevin	Lori	Mark

Is this a good Matching????

Kevin	John	Jane	Henry
Mark	John	Henry	Jane
Lori	Jane	John	Henry

Jane	Kevin	Mark	Lori
John	Kevin	Mark	Lori
Henry	Kevin	Lori	Mark

Is this a good Matching????

Kevin	John	Jane	Henry
Mark	John	Henry	Jane
Lori	Jane	John	Henry

Jane	Kevin	Mark	Lori
John	Kevin	Mark	Lori
Henry	Kevin	Lori	Mark

- Kevin and John form a blocking pair

Stable Matching

- Stable Matching: An assignment with no blocking pair (GOOD!!!)

- Formalize in TCS terms
  - Given a complete bipartite graph
  - Output a perfect matching with no blocking pairs

Is this Stable??

Kevin	John	Jane	Henry
Mark	John	Henry	Jane
Lori	Jane	John	Henry

Jane	Kevin	Mark	Lori
John	Kevin	Mark	Lori
Henry	Kevin	Lori	Mark

Natural Questions

- Is there always a stable assignment ??
- Is it easy (polynomial time) to find such an assignment ??
- What if there are multiple different stable assignments how do I compare them ??

Gale-Shapley Deferred Accept Algorithm (DA)

- Initially we start with an empty matching
- While (exists an innovator who is unmatched):
  - Each unmatched innovator m proposes to the highest ranked investor i on their list they haven’t proposed to yet
  - If i is not matched we tentatively match i to m
  - If i is currently matched to some m’ we have two cases:
    - i prefers m’ over m in which case she rejects m
    - i prefers m over m’ in which case we unmatch m’ and i is now matched to m
- Return the matching

Deferred Accept

Kevin	John	Jane	Henry
Mark	John	Henry	Jane
Lori	Jane	John	Henry

Jane	Kevin	Mark	Lori
John	Kevin	Mark	Lori
Henry	Kevin	Lori	Mark

Kevin	Jane	John	Henry
Mark			
Lori			

Deferred Accept

Kevin	John	Jane	Henry
Mark	John	Henry	Jane
Lori	Jane	John	Henry

Jane	Kevin	Mark	Lori
John	Kevin	Mark	Lori
Henry	Kevin	Lori	Mark

Kevin		John	
Mark	Kevin		
Lori	Henry		

DA runs in polynomial time

- At every step atleast one innovator is crossed of the table or there is a perfect matching
- Atmost n^2 entries in the table, so need atmost n^2 steps

DA returns a perfect matching

- Let m be unmatched at the end of DA, i.e. he was rejected by all
- Then there must be a i that is also unmatched
- But by DA m must have proposed to i at which time she was matched to someone she preferred
- But by DA once an investor i is matched, they are never unmatched
- CONTRADICTION!!!!!

DA returns a Stable Matching

- Let M be the matching from DA, and (m, i) be a blocking pair for M
- M must contain the following pairs (m, i’) and (m’, i)
  - m prefers i to i’
  - i prefers m to m’
- By DA m must have proposed to i and got rejected
- But if i rejected m then she must been matched to someone better, since i only improve the quality of their match
- i prefers m’ over m, (m,i) is not a blocking pair, CONTRADICTION!!!!

Properties of DA

- Innovator Optimal: Each innovator gets the best possible match they could have possibly gotten over all stable matchings
- Investor Pessimal: Each investor gets the worst possible match they could have gotten over all stable matchings
- DA is DSIC (Dominant Strategy Incentive Compatible), i.e. lying about your preferences will not help you improve your partner

DA is Innovator-Optimal

- An investor is a valid partner of an innovator if there exists a stable matching where they are matched to each other.
- Let us assume that DA is not Innovator optimal, i.e. there exists a innovator who was rejected by a valid partner in M.
- Let m be the first innovator who gets rejected by a valid partner say i.
- Consider the matching M' where m is matched to i, it exists since they are valid partners.

DA is Innovator-Optimal

- i must prefer its partner m’ in M over M’ since it rejected m by DA.
- Let m’ be matched to i’ in M’.
- Note that since m was the first person to be rejected m’ was never rejected by a valid partner and so must prefer i to i’
- But that means that (m’, i) is a blocking pair in M’
- CONTRADICTION

DA is Investor Pessimal

- Let there be a pair (m,i) in M from DA where m is not the worst valid partner for i
- Then there exists a stable matching M' where i is matched to m' whom she prefers less than m
- Let m be matched to some i' in M'
- Then (m,i) form a blocking pair in M'
- CONTRADICTION!!!!

Extending our simple case

- For the case of Hospital/Resident, Students/Schools the problem is now many-to-one instead of one-one, but same algorithm works too
- Stable Roomates, considers the problem where instead of a bipartite graph you are now given an arbitrary graph
  - Stable solution doesn't always exist
  - Polynomial algorithm to find one if it exists, more advanced
- Incomplete lists
- Partial Ordered Preference Lists
- Adding weights on graph: Min/Max Wt Stable Matching