Design of a Combinatorial Algorithm for Computing A-D Market Equilibria

Piyush Ahuja

Pradeep George Mathias

Supervisor: Prof. Naveen Garg

Contents

- Mathematical Model
  - Non-zero liking graph
- Non convex program & implications
- Equivalent convex program and combinatorial characterization
- Removal of negative cycles
- Open Questions

Mathematical Model

- N people in the system, with initial endowment of divisible goods
- Without loss of generality (in the linear utility case):
  - Each person has a single good
  - Quantity of each good is 1
- Further,
  - Each good has atleast someone interested in it
  - Each person is interested in atleast one good
- With a little loss of generality
  - Each subset of persons has atleast one person outside the set who is interested in a good possessed by a person in the subset.

Characterization – a non-convex program

$$
\begin{aligned}
&\forall j: \ \sum_i x_{ij} = 1 \\
&\forall i,j: \ x_{ij} \geq 0 \\
&\forall i,j: \ \frac{u_{ij}}{p_j} \leq \frac{\sum_k u_{ik} x_{ik}}{p_i} \\
&\forall i: \ p_i > 0
\end{aligned} \tag{1}
$$

Non-zero liking graph

- Directed graph having n nodes, each node representing a person/good
- Directed edge from node i to node j iff $u_{ij} > 0$.

- Our assumptions ensure that the Non-zero liking graph is a single SCC.
  - If there exists set of goods in whom others are not interested, then it will not be a single SCC
  - Solve the problem for each SCC
  - In the acyclic components of the SCCs, scale the prices appropriately so that people will buy only goods from their own SCC
  - Prices can be set to 0 only for 'initial' SCCs.

Implications

- *The feasible region of non-convex program has all and only general market equilibria.*

- Prices are market-clearing:

$$\forall i: \; p_i = \sum_j x_{ij} p_j$$

- Money is spent optimally

Convex Program

- Modifying equation (3) gives :

$$u_{ij} > 0 : p_i/p_j \leq \sum_k u_{ik}x_{ik}/u_{ij}$$

$$w(ij) = \frac{\sum_k u_{ik}x_{ik}}{u_{ij}}$$

$$\forall j : \sum_i x_{ij} = 1$$

$$\forall i, j : x_{ij} \geq 0$$

$$\text{For every cycle, } C \text{, of } G : \prod_{ij \in C} w(ij) \geq 1$$

- Non-Convex program feasible if and only of No Negative Cycle in the non-zero liking graph (on logarithmic scale)

Toward Combinatorial Algorithms

- Combinatorial characterization - No Negative Cycle
- Passive Characterization – doesn't tell us how to fix it
- Active characterization in Fisher's model through Eisenberg-Gale's LP

Detection of Negative Cycles

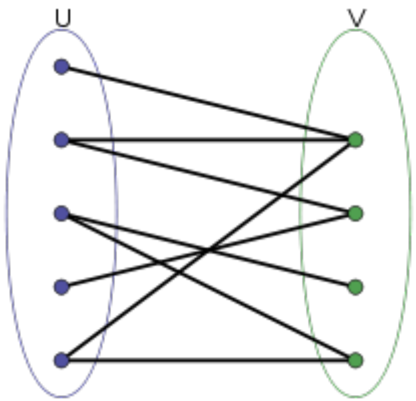
- Value of most negative cycle should be made >= 0. Iteratively increase the weight of this most negative cycle.
- Control over only x_{ij}. Increase in x_{ij} results in increase of weights of all edges from node i.
- If we have two negative cycles having different values, we can transfer some amount of good from one to the other so that the most negative cycle increases.

Floyd Warshall

- Use Floyd-Warshall algorithm with edge weights to find most negative cycle containing a given vertex.
- Gives value of the most negative cycle containing a vertex, not all the cycles that achieve this value.

Allocation Graph

- Bipartite graph (U, V): U are goods, V are users, edges demarcate allocation.
- Useful in visualizing (re-)allocation of goods.



Open Questions

- Can we reallocate goods within a cycle so as to ensure its weight increases?
- Can we always find cycles with differing negativity? -Not necessarily. Eg. A single cycle graph.
- FW cannot give us all cycles that achieve the maximum negativity. There can be exponential number of cycles. Need to correct all of them.
- Cycles interleave at nodes. Hence trying to correct one cycle, affects all cycles that the node is part of.

References

K.Jain. A Polynomial Time Algorithm for Computing an Arrow-Debreu Market Equilibrium for Linear Utilities [2004]

Q/A ,Feedback?