Independent Study - Spring 2016
Searching for Equilibria in Graphical Economies

Nehal R. Kamat

February 5, 2017

1 Introduction

A market is a model that represents exchange of goods in an economy on a global scale, a local scale or both. Arrow and Debreu [2] in 1874 helped reach the pinnacle of studies on market economies after they published a paper providing extremely general conditions for the existence of equilibrium in such markets. An equilibrium, in the traditional AD economic setting, is said to exist when the markets clear [1], i.e. supply balances demand, and all individual traders and firms that are a part of that system optimize their utility subject to budget constraints.

Arrow and Debreu's findings established the existence of various notions of equilibria in markets. However, finding or rather computing the equilibria depends on the size and structure of the economy being analyzed and is a rather arduous task. A lot of effort has been put into trying to calculate equilibrium in game-theoretic settings such as trying to calculate Nash-equilibrium in the zero-sum 2 player games, yet not very promising results have been obtained. The best known algorithms also require exponential time in the worst case. [1] Equilibrium in AD economies haven't been studied to that level of depth.

Representations that permit the expression of the common type of structure of economies may help reduce the computation complexity involved in tasks such as calculating points of equilibria in sense economic strcutres such as global markets. [1] One such representation is representing an economic model as a graphical model where the entire economy being analyzed is represented as an undirected graph with nodes acting as traders and edges acting as bidirectional trade links.

2 Graphical Economy Setting

Each vertex i in an undirected graph represents an individual party in a graphical economic setting. If an edge exists between vertices i and j it means that free trade is allowed between the two traders, and the absence of an edge means trade between the 2 traders in question is restricted. Since not all parties may directly engage in trade, the graphical economics model portrays how the price of individual goods may vary across the economy. We can say that the graphical economics model suggests a local notion of clearance, which means that instead of asking that the entire (global) market clear in each good, we can ask for clearing of markets at the local level, i.e., at the level of every node in the graph.

In order to describe the structure of the graphical setting, we must first understand the setting of the Arrow-Debreu economy from which the graphical model is derived. According to theory of AD economies in [1] The classical AD economy consists of n traders who trade k commodities of goods amongst themselves in an unrestricted manner. In this setting, each unit of commodity $h \in \{1, ..., k\}$ can be bought by any trader at prices p_h. The vector of prices is denoted as $p \in R_k^+$ (where $R^+ = x \geq 0$). We assume that each trader i has an initial endowment $e^i \in R_k^+$ of the k commodities, where e_h^i is the amount of commodity h initially held by i. Each trader i purchases a consumption plan $x^i \in R_k^+$, where x_h^i is the amount of commodity h that is purchased by i. These commodities can be sold to other traders and thus provide trader i with wealth to purchase other goods. Hence, if the initial endowment of trader i is completely sold, then the wealth of trader i is $p \cdot e^i$. A consumption plan x^i is budget constrained if $p \cdot x^i \leq p \cdot e^i$, which implicitly assumes the endowment is completely sold and this event can be observed at equilibrium.Every trader i has a utility function $u_i : R_k^+ \to R^+$, where $u_i(x^i)$ describes how much utility trader i receives from consuming the plan x^i. The utility function thus expresses the preferences a trader has for varying bundles of the k goods. [1]

Trade restrictions are also described in the graph. Trade restrictions mean that every trader is not allowed to trade with every other trader. In the undirected graph, a trade restriction between any 2 nodes is represented by an absence of an edge between them. If an edge exists between traders i and j, it means that unrestricted trade is permitted between the 2 of them. More precisely, if we use $N(i)$ to denote the neighbor set of i (which by convention includes i itself), then trader i is free to buy any commodity only from any of the traders in $N(i)$. As deduced in [1], it will naturally turn out that rational

traders only purchase goods from a neighbor with the best available price.

Each trader i has a local price vector $p^i \in R_k^+$ associated with himself/herself, where p_h^i is the price at which commodity h is being sold by i. Let us denote the set of all local price vectors by $P = \{p^i : i = 1, ..., n\}$. Each trader i purchases an amount of commodities $x^{ij} \in R_k^+$, where x_h^{ij} is the amount of commodity h that is purchased from trader j by trader i. The trade restrictions that apply to these systems imply that $x^{ij} = 0$ for $j \notin N(i)$. [1] Let the consumption plan be the set $X^i = \{x^{ij} : j \in N(i)\}$ and an X^i is budget constrained if $\sum_{j \in N(i)} p^j \cdot x^{ij} \leq p^i \cdot e^i$ (it is implicitly assumed that the endowment is completely sold, which holds at a state of equilibrium equilibrium).

The assumption regarding utility in the graphical setting is the utility function only depends on the total amount of each commodity consumed, independent of whom it was purchased from. [1] This expresses the fact that the goods are identical across the economy, and traders seek the best prices available to them. If we take the liberty to define $x^i = \sum_{j \in N(i)} x^{ij}$, which is the total vector amount of goods consumed by i under the plan X^i, then the utility of trader i is given by the function $u_i(x^i)$.

Graphical Equilibria

A market or graphical equilibrium is a set of prices and plans in which all plans are optimal at the current prices and in which the market clears. [1] A trader i uses an optimal plan at prices P if the plan maximizes utility over the set of all plans which are budget constrained under P. For instance, in the graphical setting, a plan X^i for i is optimal at prices P if the plan X^i maximizes the function u_i over all X'^i subject to $\sum_{j \in N(i)} p^j \cdot x'^{ij} \leq p^i \cdot e^i$.

The market clears if the supply equals the demand. In the standard setting, if the total demand vector is defined as $d = \sum_i x^i$ and the total supply vector as $e = \sum_i e^i$. The market clears if $d = e$. In the graphical setting, the concept of clearance is applied to each commodity h sold by i, so we have a local notion of clearance, in which all the goods sold by each trader clear in the neighborhood. The clearance condition is $d^i = e^i$ where the local demand vector d^i is defines as $d^i = \sum_{j \in N(i)} x^{ji}$ and the local endowment vector is e^i.

As stated and proved in [1], the existence of graphical equilibria is based on the following assumptions:

1. u_i is a continuous function
2. u_i is strictly monotonically increasing with each commodity
3. If $u_i(x^{'}) > u_i(x)$ then $u_i(\alpha x^{'} + (1-\alpha)x) > u_i(x)$ for all $0 < \alpha < 1$
4. For each trader i and good h, $e_h^i > 0$

The following conclusions were drawn in [1] from the first 3 assumptions:

1. At equilibrium, the budget constraint inequality for trader i is saturated, a trader using an equilibrium plan x^i spends everything he/she has earned from the sale of their endowment e^i
2. In any graphical equilibrium, a trader only purchases a commodity at the cheapest price among the neighboring traders

3 Using Quadratic Programming to Check for Graphical Equilibria

Although we follow in the footsteps of [1] in defining the terms and structure associated with AD economies and graphical economies, we take a different approach to checking the existence of equilibria in a particular graphical economy setting from the one mentioned in the point of reference [1]. We try to formulate the problem of finding whether or not the entire system clears locally and the optimum consumption plans for each local node by using quadratic programming.

Quadratic programming is a special type of mathematical optimization problem. It is the problem of optimizing (minimizing or maximizing) a quadratic function of several variables subject to linear constraints on these variables. [3]

Since the concept of equilibrium in graphical economies deals is linked to the clearance of markets at a local level, the objective of the project, thus, can be expressed as the following quadratic optimization problem:

$$Min \sum_i \sum_k (e_k^i - \sum_{j \in N(i)} x_k^{ji})^2$$

A quadratic optimization approach to this problem is correct since the objective function as can be seen above is a quadratic program when expanded, but

the constraints for the problem, as we shall see below, are linear. This type of problem fits the description of a quadratic program perfectly.

What the above equation means in words is that for every local economy/trader in the graph we're trying to minimize the difference between the endowment of every good k and the total demand for the good from node i from it's neighbours $j \in N(i)$, expressed as $\sum_{j \in N(i)} x_k^{ji}$. A value of 0 for the above function would mean that given configuration for the system puts it in a state of graphical equilibrium and the plans that the computation outputs would serve as the optimum plans for the traders in the system.Squaring the objective function also allows only for non-negative values to be generated, hence keeping the value of the optimization function ≥ 0.

Since we are using a quadratic solver, we must also provide linear constraints. The constraints are derived from the concepts of budget constraint and consumer rationality. Budget constraint refers to the traders having a restriction on the amount of money that they can spend on buying items from their neighbours, basically their expenditure can be limited to the wealth they earn selling their endowments. Consumer rationality is expressed by consumers choosing the consumption plan that maximizes their respective utilities.

The linear constraints for the objective can be expressed as:

$$\forall i \rightarrow \sum_{j \in N(i)} x_k^{ji} \cdot p^j \leq p^i \cdot e^i$$

$$\forall i \rightarrow u_i \cdot \sum_{j \in N(i)} x_k^{ji} = u_i^{OPT}$$

$$\forall i \forall j \forall k \rightarrow x_k^{ji} \geq 0$$

The final constraint implies that only a non-negative amount of every good can be purchased by any trader i from any of its neighbours $N(i)$. u_i^{OPT} refers to the maximum utility that a trader can get from a consumption plan x^i. u_i^{OPT} is obtained in the following manner:

$$\forall i \rightarrow u_i^{OPT} = max(u_i \cdot \sum_{j \in N(i)} x_k^{ij})$$

A major point to note is that this method can be used only when the utilities are linear in nature. If we have non-linear utilities, then the constraints will

not remain linear and will not adhere to the rules of a quadratic program structure.

4 Tests and Results

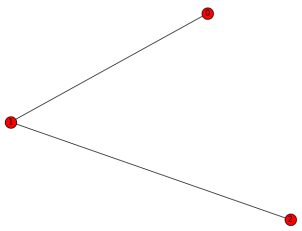

Figure 1: Test 1 Graph Structure

```
# modify values according to your preferred {node-label : endowment-per-good vector} configuration
endowment = {0:np.array([1,2]), 1:np.array([1,1]), 2:np.array([2,1])}

# modify values according to your preferred {node-label : utility-per-good vector} configuration
utility = {0:LinearUtility([1,0]), 1:LinearUtility([1,1]), 2:LinearUtility([0,1])}

# modify values according to your preferred {node-label : price-per-good vector} configuration
prices = {0:np.array([2,1]), 1:np.array([2,2]), 2:np.array([1,2])}
```

Figure 2: Test 1 Graph Configuration

```
Market Cleared!
trader: 0 | neighbor: 0 | good: 0 | Optimum purchase : 1.0 units
trader: 0 | neighbor: 0 | good: 1 | Optimum purchase : 0.0 units
trader: 0 | neighbor: 1 | good: 0 | Optimum purchase : 1.0 units
trader: 0 | neighbor: 1 | good: 1 | Optimum purchase : 0.0 units

trader: 1 | neighbor: 0 | good: 0 | Optimum purchase : 0.0 units
trader: 1 | neighbor: 0 | good: 1 | Optimum purchase : 2.0 units
trader: 1 | neighbor: 1 | good: 0 | Optimum purchase : 0.0 units
trader: 1 | neighbor: 1 | good: 1 | Optimum purchase : 0.0 units
trader: 1 | neighbor: 2 | good: 0 | Optimum purchase : 2.0 units
trader: 1 | neighbor: 2 | good: 1 | Optimum purchase : 0.0 units

trader: 2 | neighbor: 1 | good: 0 | Optimum purchase : 0.0 units
trader: 2 | neighbor: 1 | good: 1 | Optimum purchase : 1.0 units
trader: 2 | neighbor: 2 | good: 0 | Optimum purchase : 0.0 units
trader: 2 | neighbor: 2 | good: 1 | Optimum purchase : 1.0 units
```

Figure 3: Test 1 Market Status

As can be observed from the output for the first test configuration, the given configuration of endowments, utilities and price vectors clears the market and is in equilibrium. This is the only configuration of price vectors for this setting that results in a equilibrium.

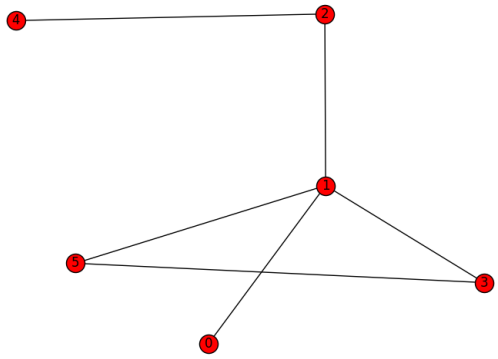

Figure 4: Test 2 Graph Structure

```
# modify values according to your preferred {node-label : endowment-per-good vector} configuration
endowment = {0:np.array([1,2]), 1:np.array([1,1]), 2:np.array([2,1]), 3:np.array([2,1]),
4:np.array([2,1]), 5:np.array([2,1])}

# modify values according to your preferred {node-label : utility-per-good vector} configuration
utility = {0:LinearUtility([1,0]), 1:LinearUtility([1,1]), 2:LinearUtility([0,1]),
3:LinearUtility([0,1]), 4:LinearUtility([0,1]), 5:LinearUtility([0,1])}

# modify values according to your preferred {node-label : price-per-good vector} configuration
prices = {0:np.array([2,1]), 1:np.array([2,2]), 2:np.array([1,2]), 3:np.array([1,2]),
4:np.array([1,2]), 5:np.array([1,2])}
```

Figure 5: Test 2 Graph Configuration

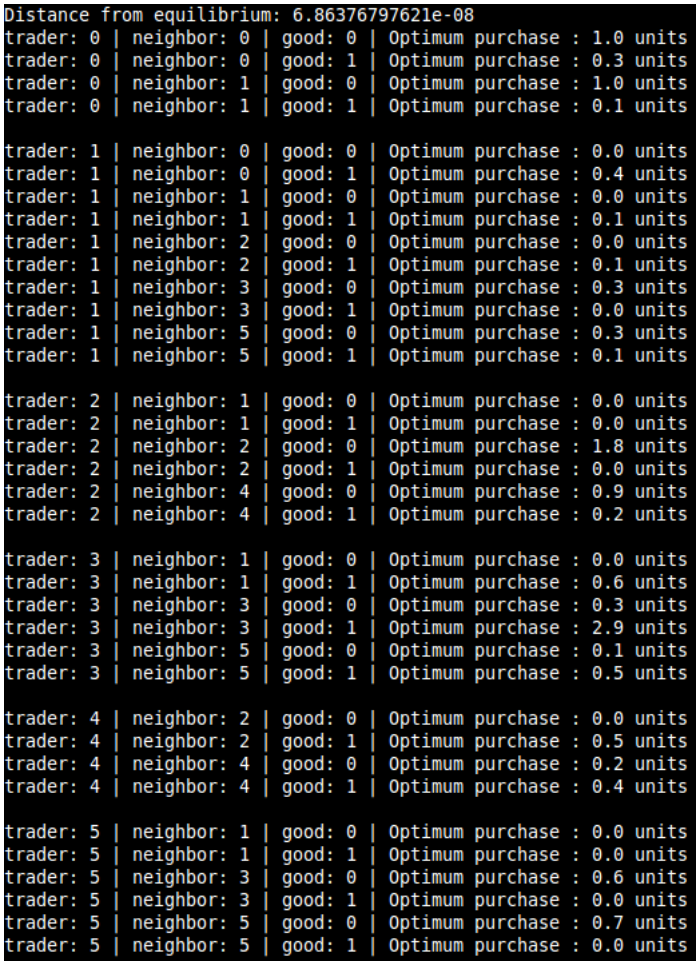

Figure 6: Test 2 Market Status

The output of the second test shows that the given pair of structure and configuration for this particular system does not result in the market clearing and hence no equilibrium. Since the distance from equilibrium is extremely tiny, this can be considered almost in equilibrium and if a perfect equilibrium is to be achieved, the demand and supply of the commodities can be balanced out by tweaking their respective local prices accordingly.

5 Future Work

Implementing a general-purpose solver to solve for configurations having non-linear utilities

References

[1] Kakade, Sham M., Michael Kearns, and Luis E. Ortiz. "Graphical economics." Learning Theory. Springer Berlin Heidelberg, 2004. 17-32.

[2] Kenneth J. Arrow and Gerard Debreu. Existence of an equilibrium for a competitive economy. Econometrica, 22(3):265290, July 1954.

[3] https://en.wikipedia.org/wiki/Quadratic_programming