ST5225: Statistical Analysis of Networks
Lecture 5: Matching Markets

WANG Wanjie
staww@nus.edu.sg

Department of Statistics and Applied Probability
National University of Singapore (NUS)

Monday 19th February, 2018

Outline

- Review
- Graph Partition
- Matching Markets

- Eigenvector Centrality
- Cohesion
  - Structure: Cliques, k-cores, coreness
  - Connectivity: k-vertex/edge-connectivity, weakly/strongly connected, bowtie
  - Densities: local density; transitivity (clustering coefficient)

For today:

- Graph partitions: definition and methods

Graph Partition

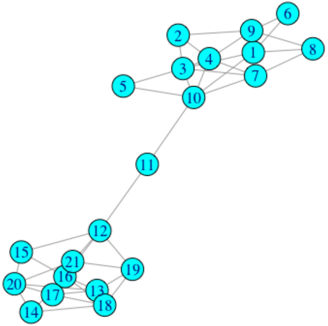
- The graph is not uniformly dense, usually it contains several more cohesive parts

Partition

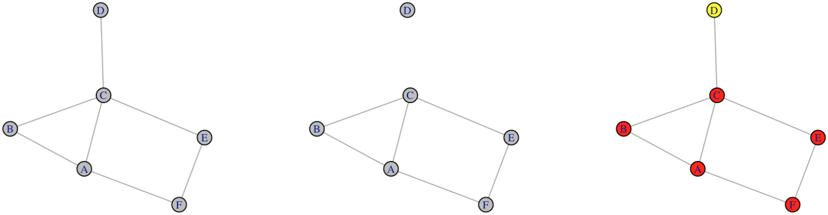
For a set V, a *partition* of V is a set of subsets $\mathfrak{C} = \{C_1, C_2, \cdots, C_K\}$, where $C_1, C_2, \cdots, C_K$ are disjoint, and $\cup_{i=1}^{K} C_i = V$.

- **Goal**: Find a partition of the graph G, so that in each subset C_i, the nodes are *well* connected, and between the subsets, the nodes are *separated*
- Application:
  - Community detection in social networks
  - Identification of possible protein complexes from protein interaction networks
- Motivation for the stochastic block model (introduce in Lecture 9-12)

- Betweenness: How many shortest edges will pass this edge
- If there are two communities, the edges between them should have high betweenness



- If we keep on moving the edges with highest betweenness, we will have more components
- Repeat until we achieve the result we need



Remark.

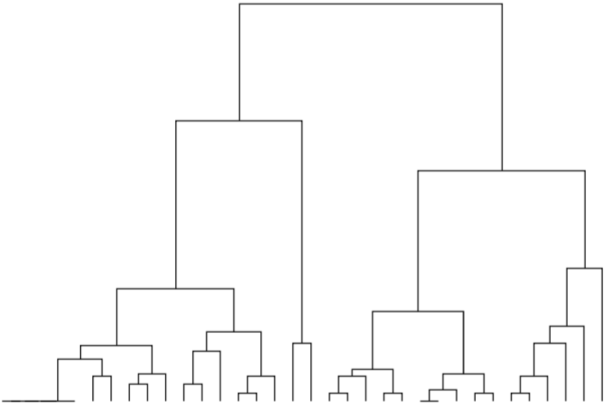
- The process can be repeated until all the nodes are separate (or the result you need according to prior information, say, we already know there are 3 communities)
- However, it is easy to cause the isolation of some nodes.

Questions and Answers:

- How about other measurements, instead of betweenness?
  - Need a *cost function* to *evaluate the quality* of the clustering/partitions
  - Various of cost functions, giving different results
- Dividing nodes, or merging nodes?
  - Both ways work
  - Starting with the original graph, and dividing nodes. We call this as a *divisive* method.
  - Starting with graph $|V|$ clusters (each node is a cluster), and merging nodes. We call this a *agglomerative* method.
- Is this the optimal result?
  - Not necessarily. The optimal result comes from exhaustive search, which is time consuming. This is a greedy search method, which may stuck at local maxima.

Questions and Answers:

- Record the result after each step
  - It will create the entire hierarchy of nested partitions, in the form of a tree. We call it a *dendrogram*.



[1]

[1]SAND, Fig 4.7 Hierarchical clustering of the karate club network.

Cost Functions

Most cost functions is a measure of *(dis)similarity* between sets, based on node-node (dissimilarity).

- Node-node (dis)similarity examples:
  - Euclidean distance:

$$dist(v_i, v_j) = \sqrt{\sum_{k \neq i,j} (A_{ik} - A_{jk})^2}, \quad A_{ij} : (i,j)th \text{ element of adjacency matrix } A$$

    When v_i and v_j have larger distance, the neighbors of them are less shared, and so v_i and v_j are possibly to be separated.
  - Another dissimilarity measurement:

$$dis(v_i, v_j) = \frac{\#\text{unshared neighbors by } v_i \text{ and } v_j}{\text{largest degree} + \text{second largest degree}}$$

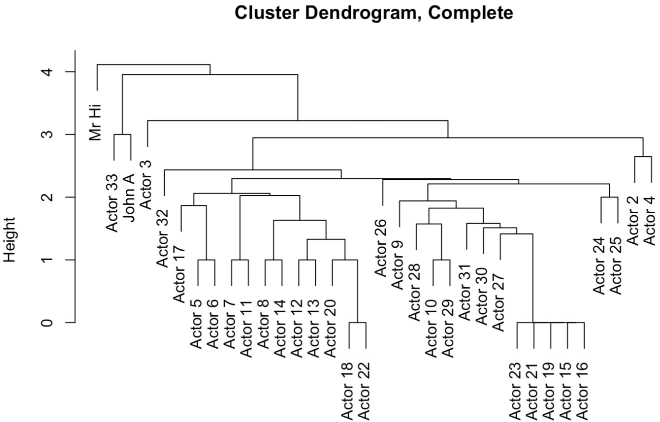
    - Numerator: number of nodes that are either neighbors of v_i only, or neighbors of v_j only
    - Ranges in $(0, 1)$

Cost Functions

Begin with the node-node (dis)similarity, different ways to calculate the (dis)similarity for a partition $\mathfrak{C}$.

- Single Linkage
  For partition $\mathfrak{C} = \{C_1, C_2, \cdots, C_K\}$, the dissimilarity between two elements C_1 and C_2 is defined as
  $$dis(C_1, C_2) = \min_{v_i \in C_1, v_j \in C_2} dis(v_i, v_j),$$
  the *minimal* distance between two nodes in the two subsets.
- Complete Linkage
  The dissimilarity between two subsets C_1 and C_2 is defined as
  $$dis(C_1, C_2) = \max_{v_i \in C_1, v_j \in C_2} dis(v_i, v_j),$$
  the *maximal* distance between two nodes in the two subsets.
- Average Linkage
  The dissimilarity between two subsets C_1 and C_2 is defined as
  $$dis(C_1, C_2) = \frac{1}{|C_1||C_2|} \sum_{v_i \in C_1, v_j \in C_2} dis(v_i, v_j),$$
  the *average* distance between all pairs of nodes in the two subsets.

HC with Complete Linkage: Karate Club



Note: cut off at the number of communities you want

Cost Functions

The cost can be defined on the partition directly, not through the node-node (dis)similarity.
For example,

- *modularity* of a partition $\mathfrak{C} = \{C_1, C_2, \cdots, C_K\}$ of the node set V.
- Define the function $f_{ij}(\mathfrak{C})$, where

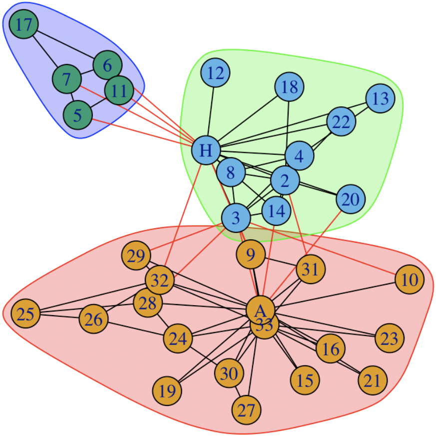
$$f_{ij}(\mathfrak{C}) = \frac{|\{(v_k, v_l) \in E; v_k \in C_i, v_l \in C_j\}|}{|E|},$$

  the fraction of edges that connect a node in C_i and a node in C_j.
- The modularity is

$$mod(\mathfrak{C}) = \sum_{i=1}^{K} [f_{ii}(\mathfrak{C}) - f_{ii}^*], \qquad \text{where } f_{ii}^* = (\sum_{j \in V} f_{ij})(\sum_{j \in V} f_{ji}),$$

- Note:
  - For undirected graphs, $f_{ij} = f_{ji}$

HC with Modularity: Karate Club



Note: 3 communities

Summary: Graph partitions

- A phenomenon in many applications: biology, social network, etc.
- Hierarchical clustering: not sure where to cut off; may stuck at the local minima
- Modularity: computation cost
- Still many other methods:
  - minimal spanning tree
  - k-means
  - Spectral clustering
  - etc.
- The attributes may help

Overview

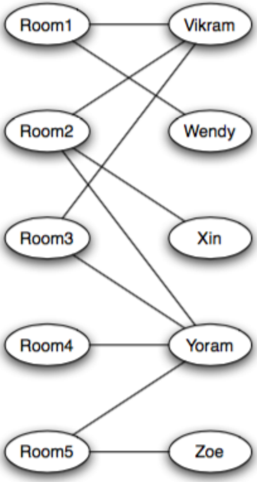
- Bipartite network
- Perfect matching
  - Constricted sets
  - Perfect matching theorem
- Optimal assignments
- Market-clearing prices
  - Properties
  - Construction

Materials

Networks, Crowds, and Markets: Chapter 10.1-10.4

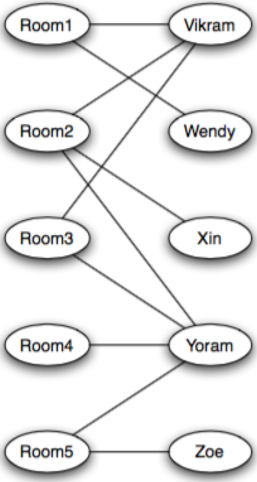
Dorm assignment problem

- 5 dorms available for 5 students; each room for **1** single student
- Each student has a list of acceptable rooms; shown in graph
- Questions:
  - Any assignment so that everyone is happy with the room?
  - Is there always such a "perfect" assignment?
  - How to decide whether this "perfect" assignment exists or not?

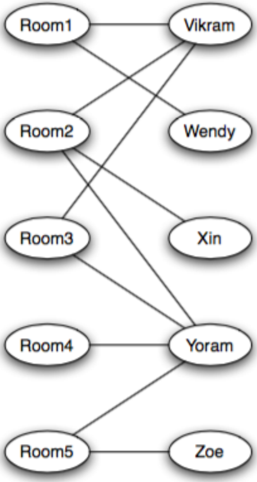


Dorm assignment problem

- 5 dorms available for 5 students; each room for **1** single student
- Each student has a list of acceptable rooms; shown in graph
- Questions:
  - **Any assignment so that everyone is happy with the room?**
  - Is there always such a “perfect” assignment?
  - How to decide whether this “perfect” assignment exists or not?

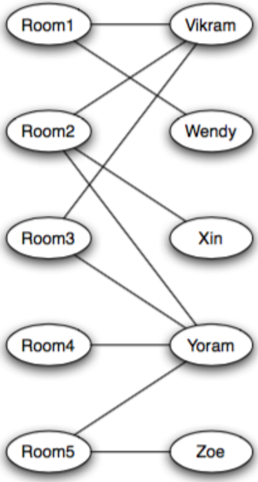


- 5 dorms available for 5 students; each room for **1** single student
- Each student has a list of acceptable rooms; shown in graph
- Questions:
  - **Any assignment so that everyone is happy with the room?**
  - **Is there always such a "perfect" assignment?**
  - **How to decide whether this "perfect" assignment exists or not?**



Dorm assignment problem

- 5 dorms available for 5 students; each room for **1** single student
- Each student has a list of acceptable rooms; shown in graph
- Questions:
  - **Any assignment so that everyone is happy with the room?**
  - **Is there always such a "perfect" assignment?**
  - **How to decide whether this "perfect" assignment exists or not?**



The model we start with is called *bipartite matching problem*.

Bipartite graph

For a graph $G = (V, E)$, if the set of vertices V can be divided into two disjoint and independent sets V_1 and V_2 such that every edge connects a vertex in V_1 to one in V_2, then we call G as a *bipartite graph*.

Remark.

- Bipartite graph usually has two types of nodes (dorms and students).
- It appears frequently in daily life, e.g., network of audience and movies on IMDB, networks of authors and papers, etc.
- Obviously, there won't be self-loops on bipartite graphs. (There might be multi-edges, yet we only consider simple graphs)
- To save space, the adjacency matrix for bipartite graph is usually record as a $|V_1| \times |V_2|$ rectangular matrix, instead of $|V| \times |V|$ matrix.

The model we start with is called *bipartite matching problem*.

Bipartite graph

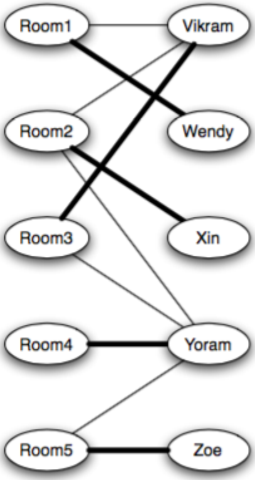
For a graph $G = (V, E)$, if the set of vertices V can be divided into two disjoint and independent sets V_1 and V_2 such that every edge connects a vertex in V_1 to one in V_2, then we call G as a *bipartite graph*.

Remark II.

- V_1 and V_2 may have different number of nodes. If they have the same number of nodes, we call it as *balanced*
- Bipartite matching problem is concerning whether there is a *perfect matching* on balanced bipartite graph.

Perfect matching

- Problem: Assignment so that each student is happy with the room
- Graph: Find a sub-graph, so that in this subgraph, each student is assigned a distinct room
- Solution: The darkened edges is such a sub-graph
- We call this solution as a *perfect matching*



Perfect matching

Perfect Matching

When there are an equal number of nodes on each side of a bipartite graph, a *perfect matching* is an assignment of nodes on the left to nodes on the right, in such a way that

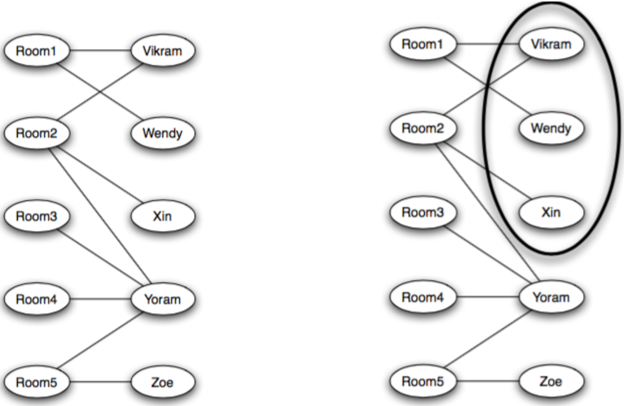
- (Connection) each node is connected by an edge to the node it is assigned to, and
- (distinction) no two nodes on the left are assigned to the same node on the right.

Remark.

- The dorm-student problem is asking for a perfect matching
- For a perfect matching, each node in V_1 is connected to one and only one "friend" in V_2, and vice versa.
- If the perfect matching subgraph is denoted as $H = (V_H, E_H)$, then $V_H = V$, $|E_H| = |V_H|/2$.

Constricted Sets

Are there always be such an assignment?



- New graph: Delete an edge between Vikram and Room 3, so Vikram only has two acceptable rooms
- No proper assignment! (3 students prefer only 2 rooms)

Constricted Sets

- For a set of nodes S, let $N(S)$ denote the collection of neighbors of the nodes in S. For a bipartite graph, note that $N(S)$ is concerning the other group of nodes.

Constricted sets

A set S is *constricted* if $|S| > |N(S)|$.

- *If there's a constricted set S, there is no perfect matching*
- If there is a perfect matching, then with the perfect matching, the neighbors of S equals to $|S|$, which is impossible.

How can we claim whether there is perfect matching or not?

Matching Theorem

If a balance bipartite graph has no perfect matching, then it must contain a constricted set.

Remark.

- Equivalence between *perfect matching* and *no constricted set*

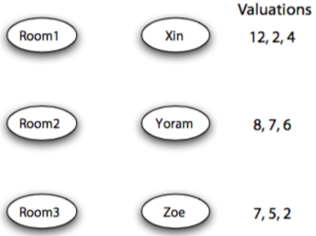
$$\text{Perfect matching} \Longleftrightarrow \text{No constricted sets}$$

- We do not need considering all the possible matchings. The problem reduced to find a set and check whether it is constricted or not

Bipartite Graph with Valuations

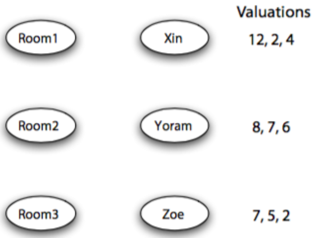
Now, each student can also set a score to show *how much* they like the room

- 3 dorms available for 3 students; each room for **1** single student
- The three numbers show the evaluation for the three rooms.
- What is the “perfect” assignment here?

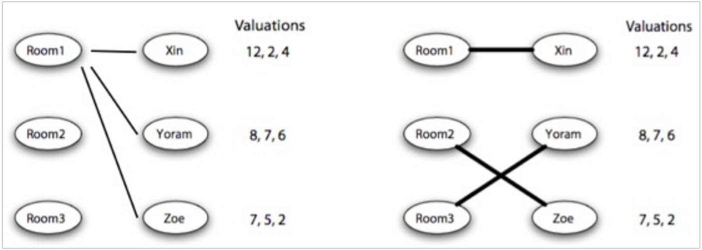


Now, each student can also set a score to show *how much* they like the room

- 3 dorms available for 3 students; each room for **1** single student
- The three numbers show the evaluation for the three rooms.
- **What is the "perfect" assignment here?**

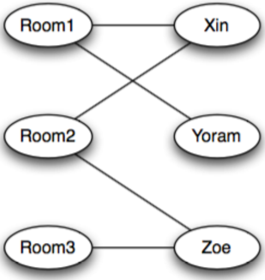


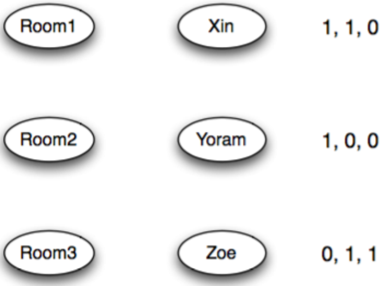
Bipartite Graph with Valuations



- Left: "Favorite graph": each student is connected with the room with largest evaluation score – No perfect matching!
- Right: Not everyone gets the optimal result, but generally everyone is happy
- Right: The overall valuation is: $12 + 6 + 5 = 23$!
- We call the assignment that can maximize the overall evaluation score as an *optimal assignment*

Bipartite Graph with Valuations, Examples





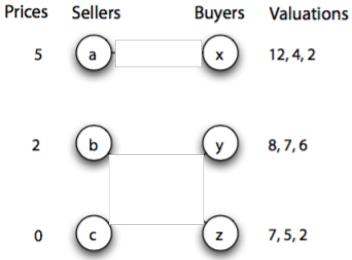
- The "acceptable or not" options can also be seen as a graph with valuations.
- The "perfect matching" is to figure out whether the optimal assignment could reach valuation $|V|$ (=3) or not.

Summary

- Bipartite graph: two sets of nodes V_1 and V_2; edges between V_1 and V_2; no associated values to the nodes/edges
  - Perfect matching
  - Subgraph of the original graph, where each node is associated to one and only one edge
- Bipartite graph with valuations: each $v_j \in V_2$ has valuations for each node in V_1
  - Optimal assignment
  - The assignment could maximize the valuations

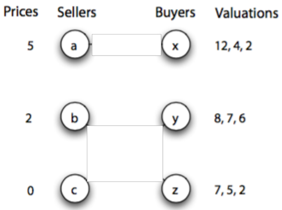
Prices

- A university may not have the same charge on the rooms, yet the market won't
- On a market, there are a set of sellers and a set of buyers. Each buyer has valuation on the product of the sellers, and each seller ask for a price.



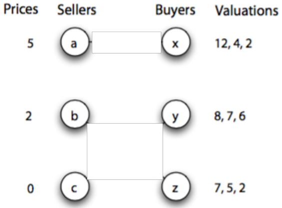
Valuations and prices

- 3 houses available for 3 buyers
- Each buyer has his/her valuations for each house. Each seller has a price for his/her house
- Questions
  - Can we still find an “optimal” assignment here?
  - If we can change the price, can we find an even better assignment?



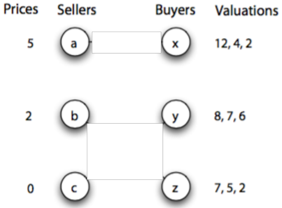
Valuations and prices

- 3 houses available for 3 buyers
- Each buyer has his/her valuations for each house. Each seller has a price for his/her house
- Questions
  - **Can we still find an "optimal" assignment here?**
  - If we can change the price, can we find an even better assignment?



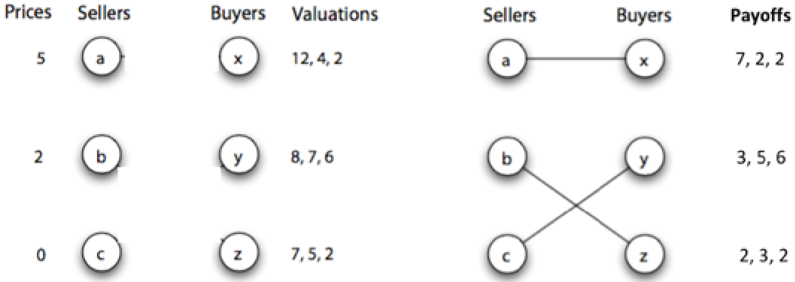
Valuations and prices

- 3 houses available for 3 buyers
- Each buyer has his/her valuations for each house. Each seller has a price for his/her house
- Questions
  - **Can we still find an "optimal" assignment here?**
  - **If we can change the price, can we find an even better assignment?**

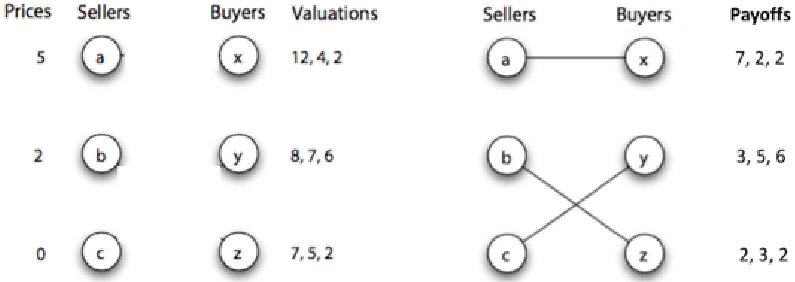


Prices and Payoffs

- *Payoffs*: For a buyer and a house, the buyer's payoff is valuation – price of this house
- Therefore, we have a graph with *payoffs* as the previous *valuations*
- Given the prices, we can find an *optimal assignment* with respect to the payoffs
- The optimal assignment could maximize the payoffs

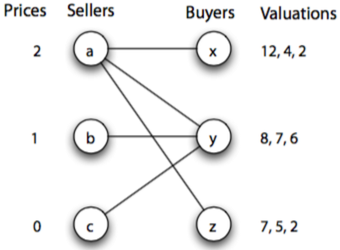


- For the result, note that each buyer is connected to the seller with largest *payoff*
- *Preferred-seller graph*: Connect each buyer with the sellers that maximizes the payoff of this buyer, and the corresponding graph is called preferred-seller graph, since the seller is the preferred seller of the connected buyers.
- If there is a *perfect matching* in the preferred-seller graph, then this perfect matching is the optimal assignment.

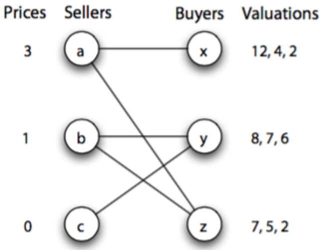


Market-Clearing Prices

- Note: every buyer gets the house that maximizes the payoffs
- We call such a set of prices *market-clearing*, since the prices setting cause each house to get bought by a different buyer–the house market is clear!
- It is possible there is no perfect matching



(c) *Prices that Don't Clear the Market*



(d) *Market-Clearing Prices (Tie-Breaking Required)*

- *Existence: for any set of buyer valuations, there exists a set of market-clearing prices*
  - For any set of buyers, the sellers can set the prices carefully, so that each buyer gets distinct house
- *Optimality: For any set of market-clearing prices, the optimal assigment has the maximum total valuation of any assignment of sellers to buyers*
  - For an assignment M

    Total payoff of M = Total valuation of M - Sum of all prices
  - Note that sum of all prices don't change. The optimal assignment maximizes the total payoff, so that maximizes the total valuation.

- *Existence: for any set of buyer valuations, there exists a set of market-clearing prices*
  - For any set of buyers, the sellers can set the prices carefully, so that each buyer gets distinct house

- *Optimality: For any set of market-clearing prices, the optimal assigment has the maximum total valuation of any assignment of sellers to buyers*
  - For an assignment M

    Total payoff of M = Total valuation of M - Sum of all prices
  - Note that sum of all prices don't change. The optimal assignment maximizes the total payoff, so that maximizes the total valuation.

Market-Clearing Prices: Construction

How to construct a set of market-clearing prices?

- Idea: if one house is in high demand (more than 1 buyers have maximum payoff on this house), then the price increases by 1
- Note: we always set the house with smallest price to be with price 0. It does not affect the result, and it helps to scale the result
- In a real market, the "hot" house usually has a higher price, so that the final buyer is the buyer who likes it most

Market-Clearing Prices: Construction

How to construct a set of market-clearing prices?

- Idea: if one house is in high demand (more than 1 buyers have maximum payoff on this house), then the price increases by 1
- Note: we always set the house with smallest price to be with price 0. It does not affect the result, and it helps to scale the result
- In a real market, the "hot" house usually has a higher price, so that the final buyer is the buyer who likes it most

Market-Clearing Prices: Construction

How to construct a set of market-clearing prices?

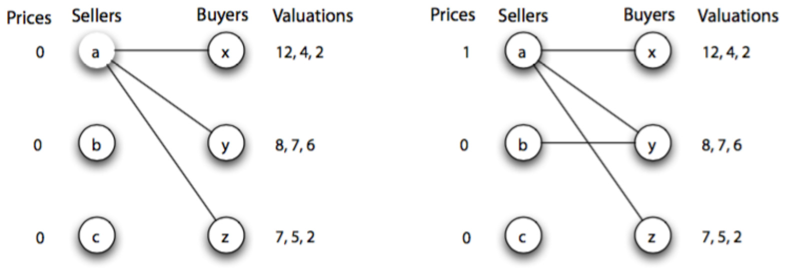
- Idea: if one house is in high demand (more than 1 buyers have maximum payoff on this house), then the price increases by 1
- Note: we always set the house with smallest price to be with price 0. It does not affect the result, and it helps to scale the result
- In a real market, the “hot” house usually has a higher price, so that the final buyer is the buyer who likes it most

Market-Clearing Prices: Construction

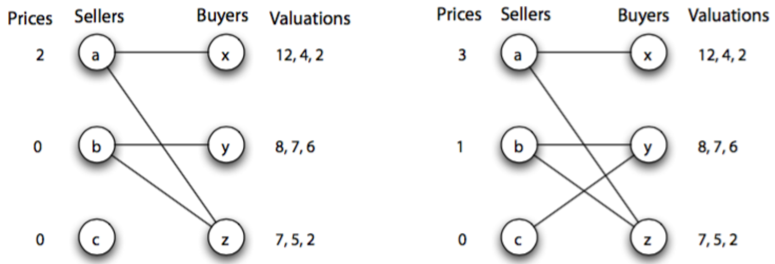
Say there are n sellers (items) and buyers

Start: each item has price 0; each buyer assigns a value to each item

1. Assume smallest value is 0; if not, scale the price so that the smallest is 0
2. Construct the preferred-seller graph and check if there is a perfect matching
3. If yes, done
4. If not, find a constricted set of buyers S
5. Each seller in $N(S)$ increases the price by 1
6. check if the smallest price is 0, if not, subtract the same amount of each price so that the smallest is 0
7. go back to Step 1.



(a) *Start of first round* (b) *Start of second round*



Market-Clearing Prices: Example II

Prices	Sellers	Buyers	Valuations
0	a	x	5, 7, 1
0	b	y	2, 3, 1
0	c	z	5, 4, 4

Summary

- Bipartite graphs with two sets of nodes V_1 and V_2; each $v_i \in V_1$ has a price; each $v_j \in V_2$ has a valuation for every $v_i \in V_1$
  - Find payoffs of each $v_j \in V_2$ assigned to each $v_i \in V_1$
  - With the payoffs, there is *optimal assignment*
  - For each $v_j \in V_2$, there is a preferred seller in V_1, which maximizes the payoff of v_j. We call this seller as *preferred seller*
  - If the preferred-seller graph has a *perfect matching*, then we call the price as market-clearing price
- Market-clearing prices must exist, and maximizes the total valuations
- Construction of market-clearing prices: begin with 0, and increase the prices of constricted set by 1.

- Power on networks
- An experiment and the results
- Nash bargaining solution
- Ultimatum game
- Stable outcomes
- Find natural stable outcomes

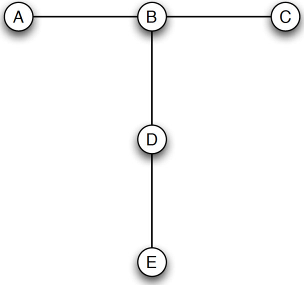
- In the buyer-seller network, the buyer has payoff as valuation minus price.
- The seller may want to increase the price, so that the payoff splits between the seller and the buyer
- If the seller increases the price only a little bit, then the buyer still transacts with this seller with a fewer payoff, and the seller gets some profit.
- If the seller increases the price too much, the buyer may give up, and no transaction is done.
- *Bargaining on network*
- Bargaining is related to *power*, the popularity of the seller/buyer in the network

Power on Networks

We talk about the general cases, not restricted to buyer-seller networks.

- Intuitionally, we may say these two nodes have the same power (related to this graph), since they are symmetric

- Node B in this graph has a larger power, since nodes A and C requires the connection with B to get in touch with D and E



- Power of distributing resources

Bargaining on networks: experiment

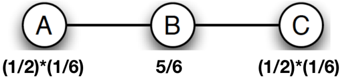
Set up an experiment to see people's bargaining behavior on networks.

1. Fix a small network. Let each node represent each player. Players that are connected with an edge can exchange information/make offers.
2. Each edge carries a resource (e.g., $100). This resource is split among adjacent nodes if there is a deal between nodes.
3. Each node can only be part of at most one deal (possible that no deal is done).
4. Players can freely negotiate for a fixed amount of time how the resource is split.
5. The experiment is repeated multiple times to get the final result

Bargaining on networks: outcomes

- For a 2-node network, end up with a fair split. Since A and B has the same power on the transaction, they agree to split the money equally.

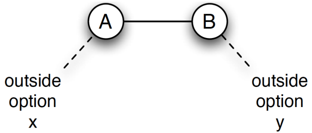
- For a 3-node network, B has higher power, where it can exclude A or C in the transaction. The result shows that A and C are symmetric, while B receives the majority of the money in the transaction.



- In the textbook, more complicated experiments, say, 4-nodes, 5-nodes are also discussed.

Nash bargaining solution

To discuss the problem, we think of each edge. The resources outside this edge is summarized as “outside options”, as the following figure



- Assume the total resource to split on this edge is 1.
- Since A can get x from outside if not split with B. So the expectation of A's payoff is at least x
- Similarly, the expectation of B's payoff is at least y
- If $x + y > 1$, impossible to satisfy both A and B
- If $x + y \leq 1$, then there is surplus $s = 1 - x - y$
- *Nash bargaining solution*: both A and B will get half of the surplus, $(1 - x - y)/2$, which means the outcome is

$$x + \frac{1-x-y}{2} = \frac{1+x-y}{2} \text{ to A;} \quad y + \frac{1-x-y}{2} = \frac{1-x+y}{2} \text{ to B.}$$

Nash bargaining solution

Interesting remarks:

- The resource is not only money, it can be friendship, collaboration, etc.
- When bargaining, people tries to communicate about the outside option, so that gets a reasonable split
- In real life, for the two endpoints, usually one has a "higher-status" and the other has a "lower-status". For example, a sophomore and a graduate student with high grades.
- It has been found that, people with high-status tend to inflate the size of their outside option; people with lower-status tend to reduce the size of their outside option.
- What's more, the reduced size of outside option will be further underestimated by the higher-status people.
- Do you have the same problem in your job search?

The ultimatum game

Setup: Two players A and B; no communication allowed; 1-time game

1. Person A is given \$100 and told to propose a division among A and B
2. If B accepts the \$100 are split accordingly; if B rejects, both A and B get nothing

If both A and B are rational,

- B should accept any positive offer, since that's better than nothing
- Since B will accept any positive offer, A, as the person who propose the division, can keep more money
- Solution: A offers \$99 to A, and \$1 to B
- The difference comes from the power of division

The ultimatum game

Setup: Two players A and B; no communication allowed; 1-time game

1. Person A is given \$100 and told to propose a division among A and B
2. If B accepts the \$100 are split accordingly; if B rejects, both A and B get nothing

If both A and B are rational,

- B should accept any positive offer, since that's better than nothing
- Since B will accept any positive offer, A, as the person who propose the division, can keep more money
- Solution: A offers \$99 to A, and \$1 to B
- The difference comes from the power of division

The ultimatum game

Setup: Two players A and B; no communication allowed; 1-time game

1. Person A is given \$100 and told to propose a division among A and B
2. If B accepts the \$100 are split accordingly; if B rejects, both A and B get nothing

Reality:

- In 1/3 of the experiments, A offers 1/2-1/2 split to B
- Moreover, when the offer is very unfair, B rejects to accept the offer – even though he will get nothing
- When the amount of money changes, the result is still the same
- Guess: It hurts B's feelings when offered very small proportion, which behaves as a negative payoff
- Solution: Give B more time to consider, then more people would accept the "unfair" offer

Stable outcomes

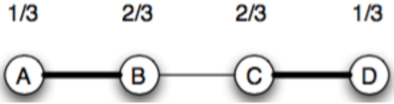
- We need some mathematically defined terms to describe the exchanges on graphs

Outcomes

An *outcome* of a network exchange on a given graph consists of:

- A matching on the set of nodes, specifying who exchanges with whom.
- A number associated with each node, called its *value*, indicating how much this node gets from its exchange. The sum of matched nodes should be 1.

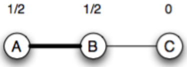
Example.



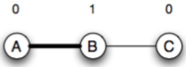
- All discussions are based on the outcomes
- Is the outcome stable?

Stable outcome

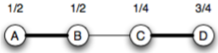
Outcome where no player can make an offer to another player such that both are *better* off, is called a *stable outcome*.



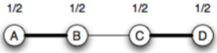
(a) *Not a stable outcome*



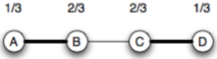
(b) *A stable outcome*



(c) *Not a stable outcome*



(d) *A stable outcome*



(e) *A stable outcome*

Stable outcomes

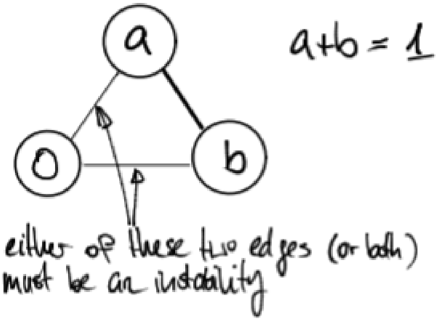
- *Instability*: if there is an edge not part of the matching, such that the sum of the endpoint values is < 1, then we call this outcome as *an unstable outcome*.
  - Say the endpoints of this edge is A and B, then obviously A and B would prefer to exchange, so that they are better off. The original outcome is not stable.
- If there is no such instabilities, then an outcome is stable.
  - If an outcome is unstable, then there always exists such an edge.
  - It helps us to figure out the stability of an outcome. Just check all the edges and the values of their endpoints, and we know whether it is stable or not

Stable outcomes

- *Instability*: if there is an edge not part of the matching, such that the sum of the endpoint values is < 1, then we call this outcome as *an unstable outcome*.
  - Say the endpoints of this edge is A and B, then obviously A and B would prefer to exchange, so that they are better off. The original outcome is not stable.

- If there is no such instabilities, then an outcome is stable.
  - If an outcome is unstable, then there always exists such an edge.
  - It helps us to figure out the stability of an outcome. Just check all the edges and the values of their endpoints, and we know whether it is stable or not

Example (restrictions of outcomes):
For a triangle, there are no stable outcomes!



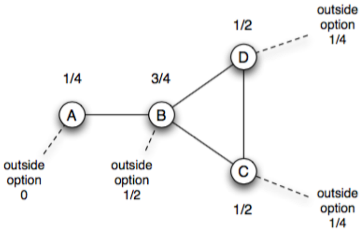
Balanced stable outcomes

- If we have many stable outcomes, then we prefer some "natural" set of outcomes, for which we call *balanced*

Balanced outcomes

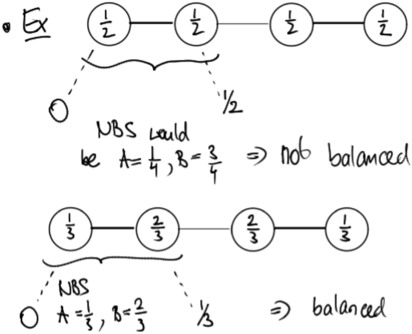
An outcome is called a *balanced outcome* if, for each edge in the matching, the split of money represents the Nash bargaining outcome for the two nodes involved, given the best outside option.

Example: a balanced outcome



Balanced stable outcomes

- It is closer to what happens in reality



- Power on networks with experiments
- Nash bargaining solution for two nodes with outside options
- Ultimatum game and the results
- Stable outcomes and natural stable outcomes (Nash bargaining solution required here)