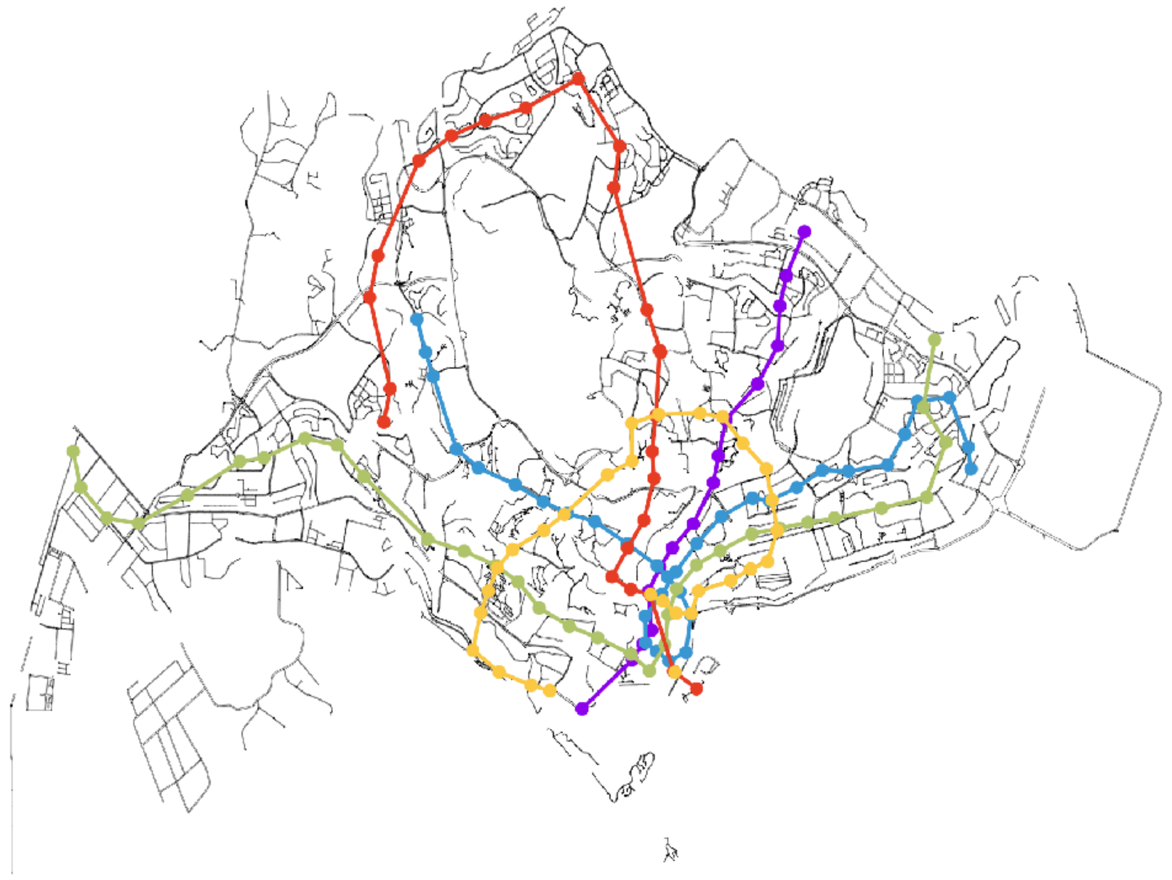
CSCI 357: Algorithmic Game Theory

Lecture 17: Decentralized Markets & Project Discussion

Shikha Singh

Announcements and Logistics

- **HW 7:** Practice question on revenue equivalence for exam (no need to turn it in, solutions posted)
- **Midterm # 2** will be on April 29
  - Similar to Exam 1: closed book but can bring up to 5 pages of notes
  - Exam time: 1.10 - 2.25 pm, **Wachenheim 015** (Arrive 5 mins Early)
  - **Change of Room**: **Wach 015 (Downstairs)**
  - Rohit will be proctoring the exam: he can contact me if needed

Questions?

Exam Resources

- Notes posted on lecture page (they are do not include all topics!)
- Lecture slides
- Homework solutions
    - HW 7 Solution posted
    - Take a look!
- Assignment solutions
- Lecture readings
- I am out of town next week, but feel free to email me if you have a question!

Ascending-Price Algorithm

- Start with prices of all items $p_j = 0$, assume all valuations $v_{ji} \in \mathbb{Z}$
- **Step 1.** Check if there is a **buyer-perfect matching** in preferred item graph at prices $\mathbf{p}$
- **Step 2.** Else, there must a constricted set:
  - There exists $S \subseteq \{1,\ldots,n\}$ such that $|S| > |N(S)|$
  - $N(S)$ are items that are **over-demanded**
  - If there are multiple such sets, choose the **minimal set** $N(S)$
    - Increase $p_j \leftarrow p_j + 1$ for all items in the set $j \in N(S)$
  - Go back to **Step 1.**
- **Question 1.** Does this algorithm eventually terminates?
- **Question 2.** Are the final prices market clearing? Equivalently, is (M, p) a competitive equilibrium?

Ascending-Price Algorithm

How can we do this? Reduce to network flow

- Start with prices of all items $p_j = 0$, assume all valuations $v_{ji} \in \mathbb{Z}$
- **Step 1.** Check if there is a **buyer-perfect matching** in preferred item graph at prices $\mathbf{p}$
- **Step 2.** Else, there must a constricted set:
    - There exists $S \subseteq \{1,\ldots,n\}$ such that $|S| > |N(S)|$
    - $N(S)$ are items that are **over-demanded**
    - If there are multiple such sets, choose the **minimal set** $N(S)$
        - Increase $p_j \leftarrow p_j + 1$ for all items in the set $j \in N(S)$
    - Go back to **Step 1.**
- **Question 1.** Does this algorithm eventually terminates?
- **Question 2.** Are the final prices market clearing? Equivalently, is (M, p) a competitive equilibrium?

Invariant for Analysis

- Let's extend the algorithm to maintain a tentative match M at all times
- **Invariant:** if an item has non-zero cost, that item is tentatively matched to some buyer:
  $p_j > 0 \implies \exists i : (j, i) \in M$

Final Prices are Market Clearing

- **Lemma.** Consider an item whose price increases in step 2, such an item is always tentatively matched to a buyer.
- **Proof.** Consider the items in the minimal constricted set $N(S)$ whose prices increase by 1
    - At the new price, all edges between S to $N(S)$ still exist
        - buyers in S may have more edges to items outside that are now just as good
    - Construct an item-perfect matching for items in $N(S)$
        - Tentatively match each item in $N(S)$ to a buyer in S (if these items were previously matched to other buyers, update their matches)
    - Why is this possible? (Halls' theorem!)

Proving Our Algorithm Terminates

- **Theorem.** The ascending price auction terminates.
- **Proof.** Show that algorithm starts with a certain amount of **"potential energy"** which goes down by at least 1 in each iteration
- Let the potential of any round be defined as:

$$E = \sum_{\text{items } j} p_j + \sum_{\text{buyers } i} u_i^*$$

- where p_j is the price of item j in that round and u_i^* is the maximum utility i can obtain given prices $\mathbf{p}$ in that round

Proving Our Algorithm Terminates

$$E = \sum_{\text{items } j} p_j + \sum_{\text{buyers } i} u_i^*$$

- **Theorem.** The ascending price auction terminates.
- **Proof.**
- At the the beginning, all prices are zero and $u_i^* = \max_j v_{ij}$
- Thus, before the auctions starts $E_0 = \sum_i \max_j v_{ij}$
- To wrap up proof, we show
  - Potential can never be negative $E \geq 0$
  - Potential at each step goes down by at least 1
- Thus, in E_0 steps the algorithm terminates. ∎

Proving Our Algorithm Terminates

$$E = \sum_{\text{items } j} p_j + \sum_{\text{buyers } i} u_j^*$$

- **Lemma:** Potential energy E is always non-negative.
- **Proof.**
- If there is at least one item with price 0 then each $u_i^* \geq 0$
  - Why is this true? Use our invariant!
  - Every non-zero priced item is matched, thus when $n-1$ items are matched, no need to raise the price of nth item
- Since prices are always are always nonnegative $E \geq 0$

Proving Our Algorithm Terminates

$$E = \sum_{\text{items } j} p_j + \sum_{\text{buyers } i} u_j^*$$

- **Claim.** Potential E goes down by at least one each step.
- **Proof**. At each step, we raise the price of all items in $N(S)$, how much does it increase the first term in E ?
  - $|N(S)|$
- However, the value of u_i^* goes down by one for each node in S, how much does this decrease the second term in E?
  - $|S|$
- Since $|N(S)| < |S|$, then potential decreases by at least 1
- Thus, the algorithm must terminate in E_0 steps ∎

Final Prices are Market Clearing

- We know the algorithm eventually terminates at some price vector $\mathbf{p}$
- To show $\mathbf{p}$ is market clearing, we need to show the following two conditions holds:
  - **Condition 1.** there exists a buyer-perfect matching M in the (final) preferred item graph at prices $\mathbf{p}$
  - **Condition 2.** If an item j is not matched to any buyer, then its price $p_j = 0$
- What is the termination condition for the algorithm?
  - Existence of buyer-prefect matching (condition 1)
- Condition 2 follows from our invariant

Analysis Summary

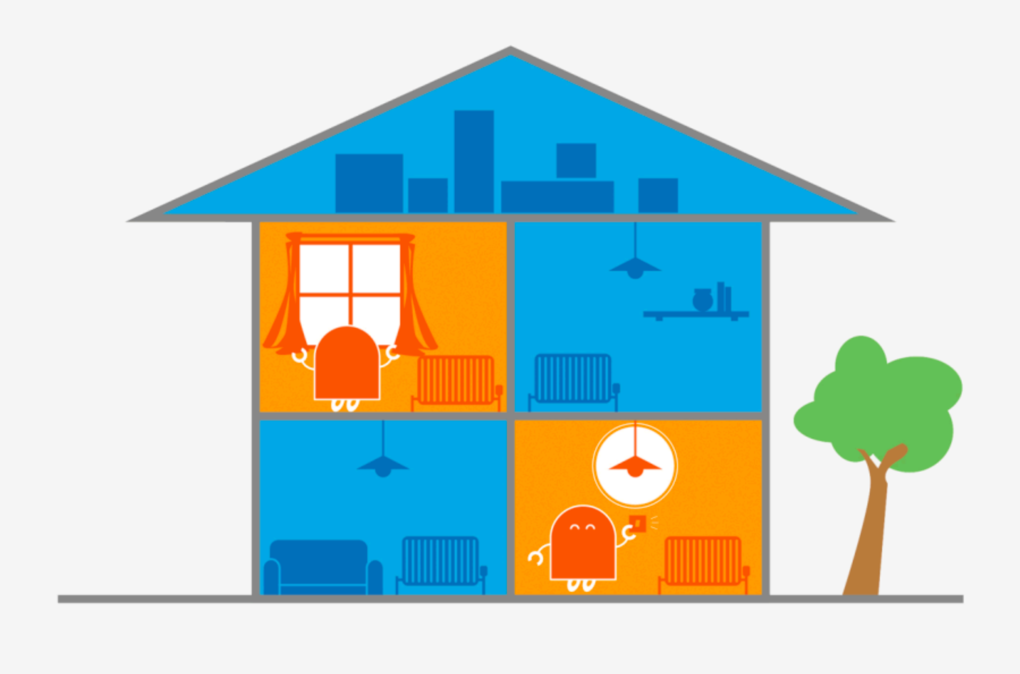
- Notice that the ascending price algorithm implicitly also maintains a matching
    - A deferred acceptance variant has buyers propose "prices" to items and items upgrade
- By the definition, the final matching is the max-weight bipartite matching
    - We learnt an alternate way to find the max-weight bipartite matching in a graph
    - (Algorithm in 256 uses network flows)
    - How efficient is this algorithm?

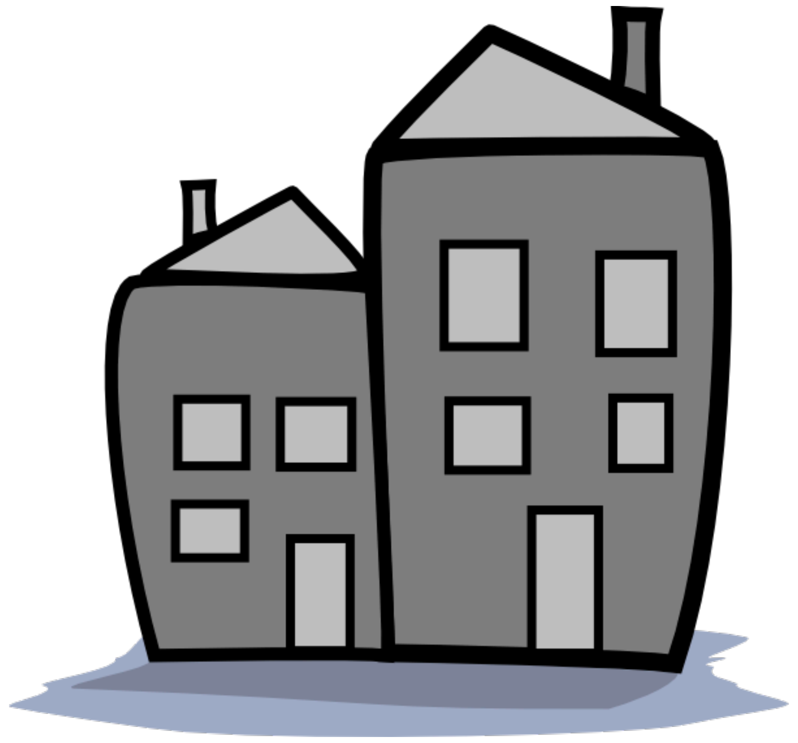
Remember VCG?

- VCG prices set **centrally**: ask each buyer to report their valuation and charge each buyer a **"personalized price"** for their allocation
- VCG prices are only set after a matching has been determined (the matching that maximizes total valuation of the buyers)
  - Not just about the item itself, but who gets the item
- Market-clearing prices are **"posted prices"** at which buyers are free to pick whatever item they like
  - Prices are chosen first and posted on the item
  - Prices cause certain buyers to select certain items leading to a matching

Applying VCG

Prices

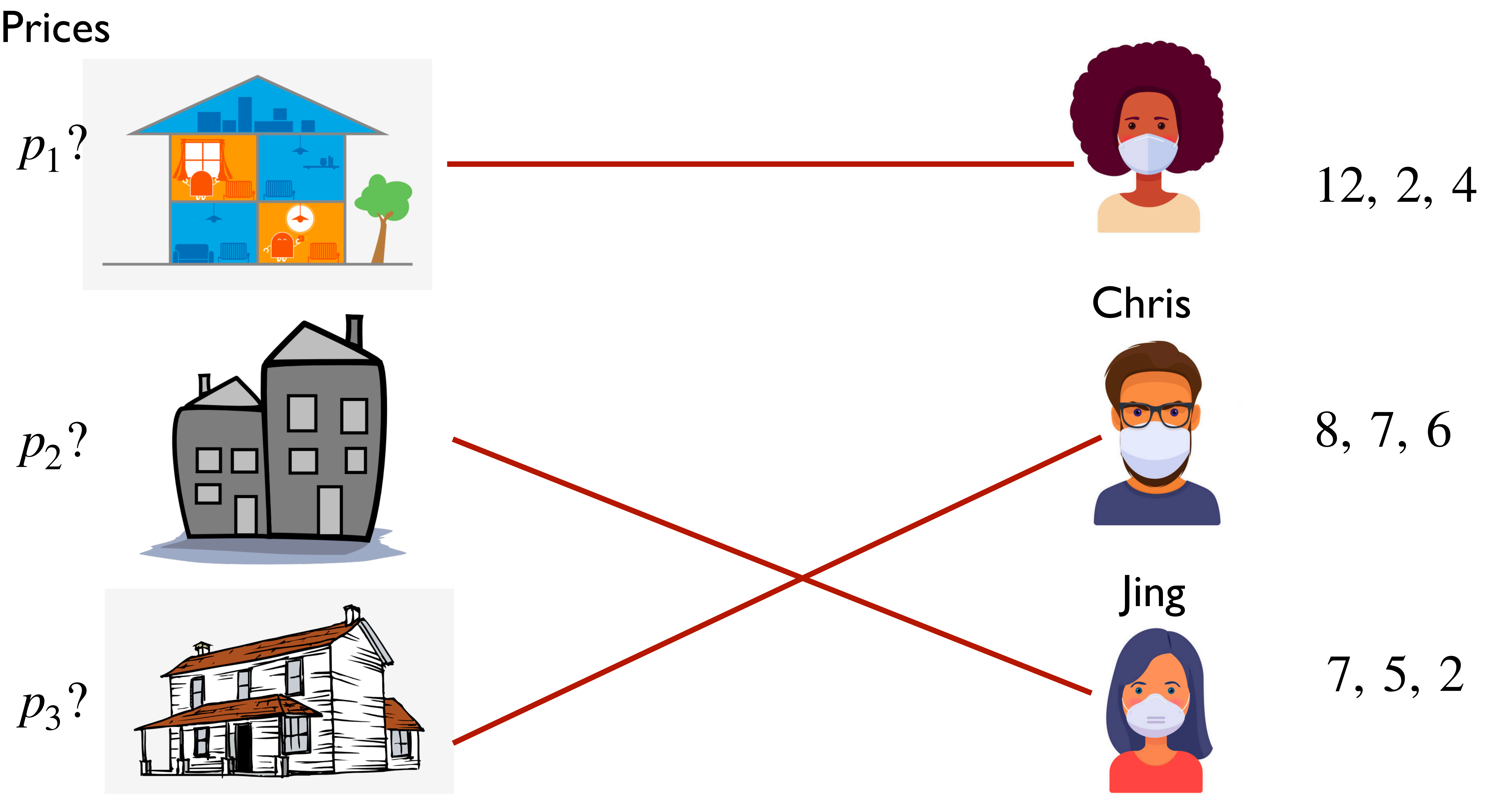




VCG. Need to find surplus maximizing allocation first

	Valuations
Zoe 	12, 2, 4
Chris 	8, 7, 6
Jing 	7, 5, 2

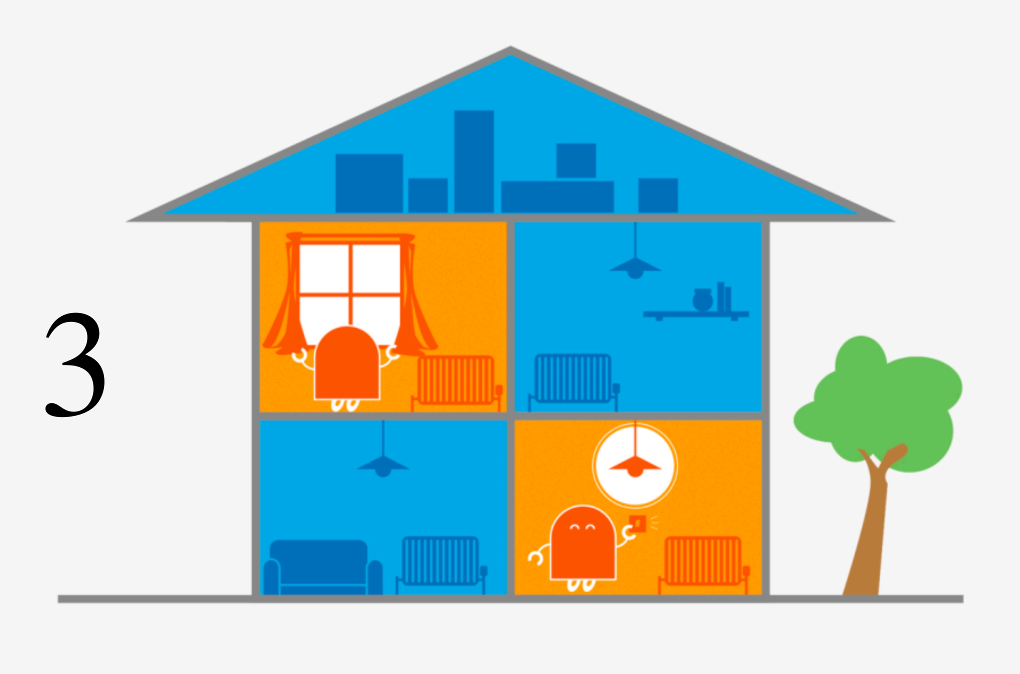
Applying VCG

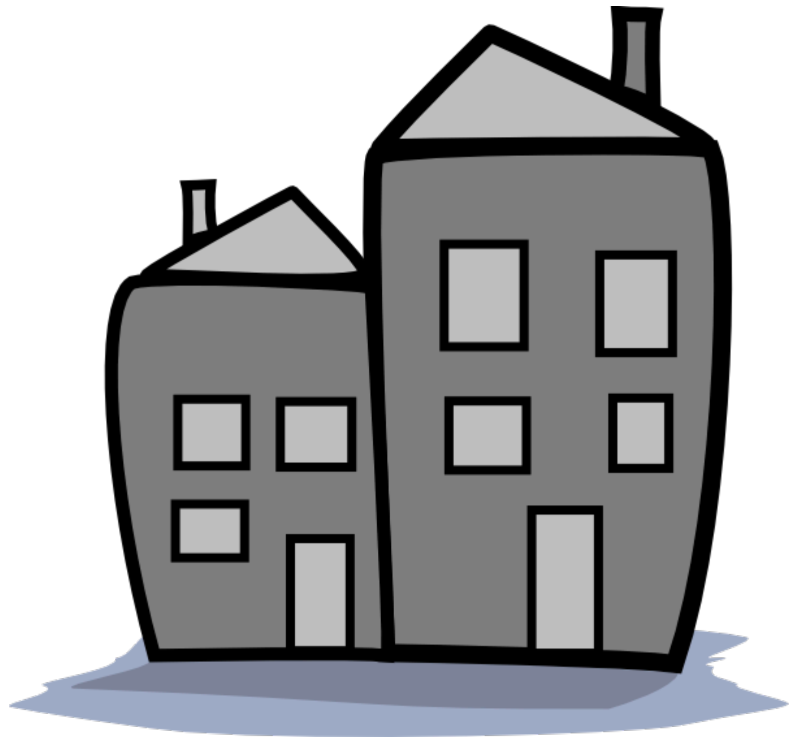


Applying VCG

Prices

$p_1 = 3$





Surplus without Zoe: **7+7 = 14**
Surplus by others when Zoe is present:
6 + 5 = 11

Zoe	Valuations
	12, 2, 4

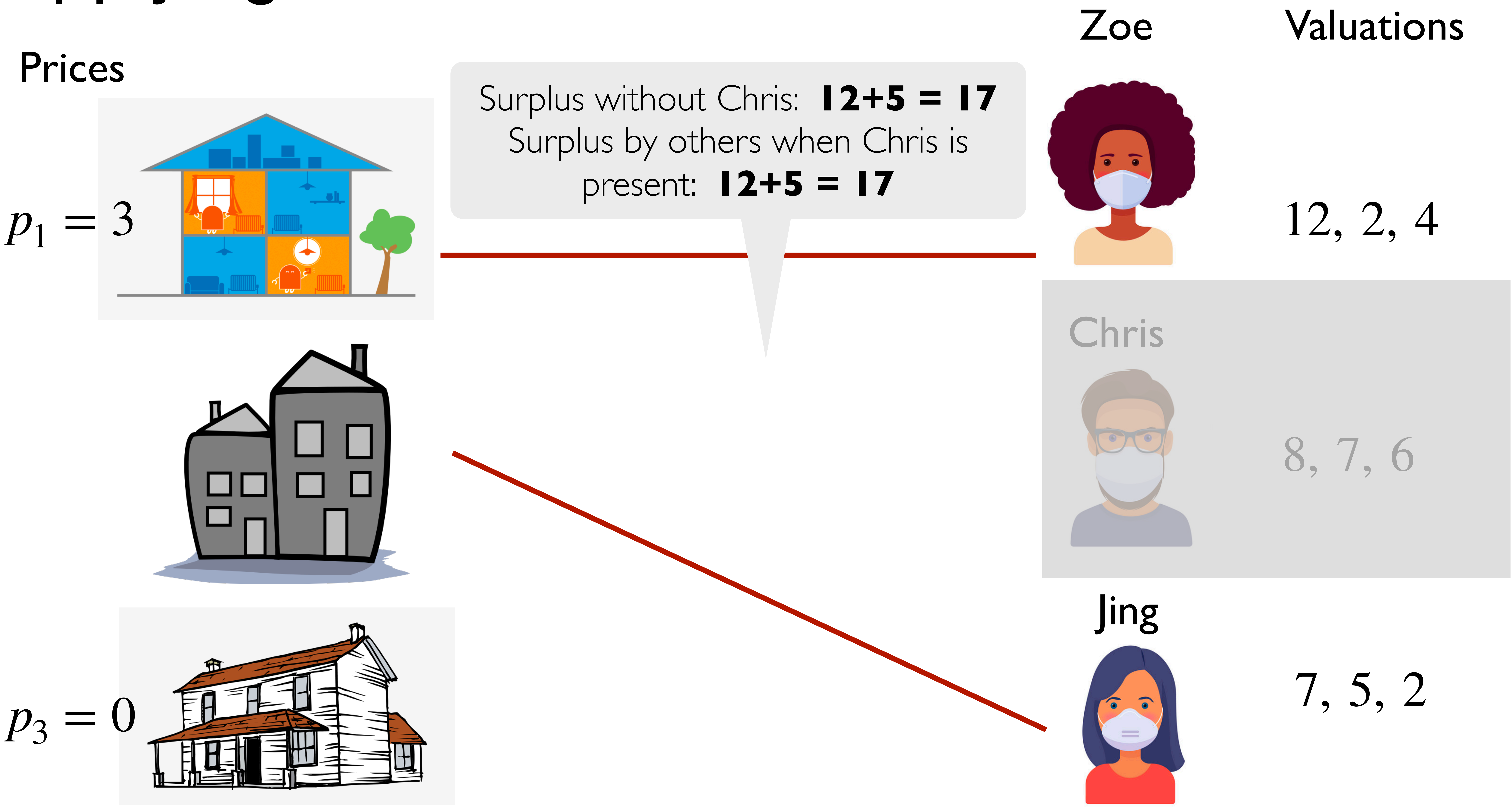
Chris

8, 7, 6

Jing

7, 5, 2

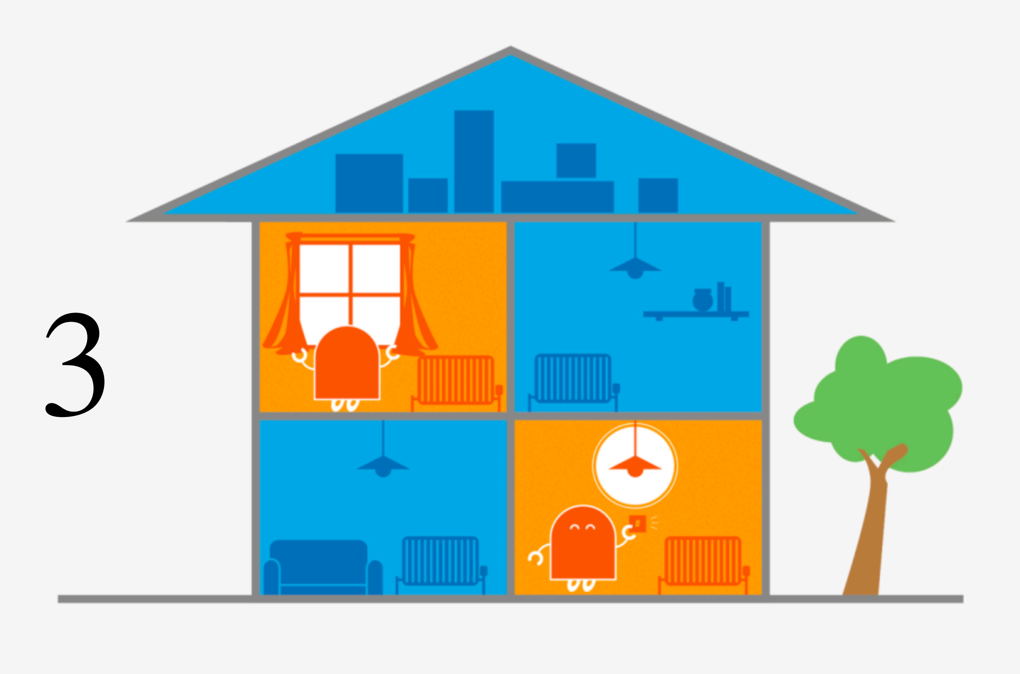
Applying VCG



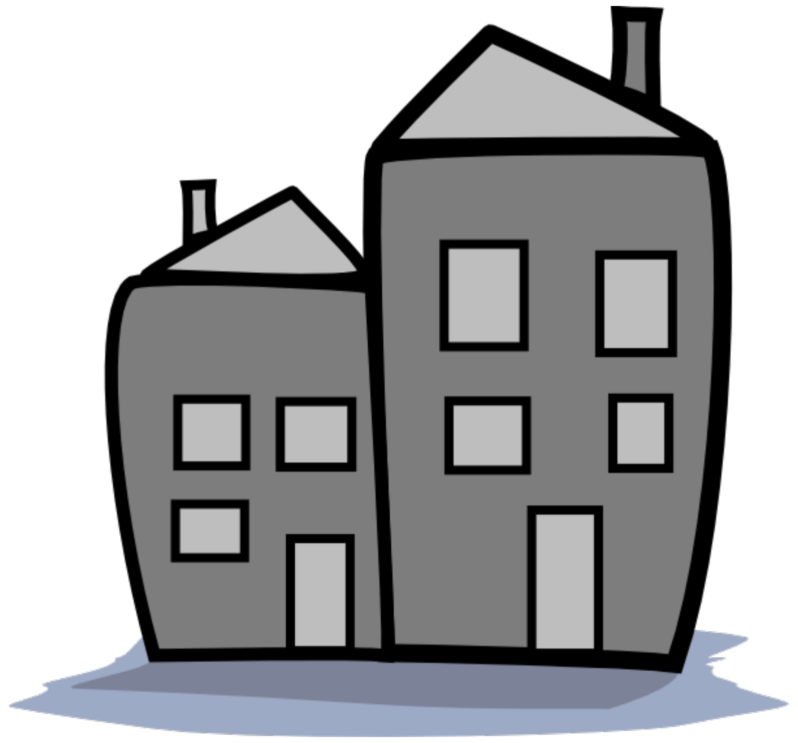
Applying VCG

Prices

$p_1 = 3$



$p_2 = 1$



$p_3 = 0$

Surplus without Jing: **12+7 = 19**
Surplus by others when Jing is present:
12+6 = 18

Zoe

Chris

Jing

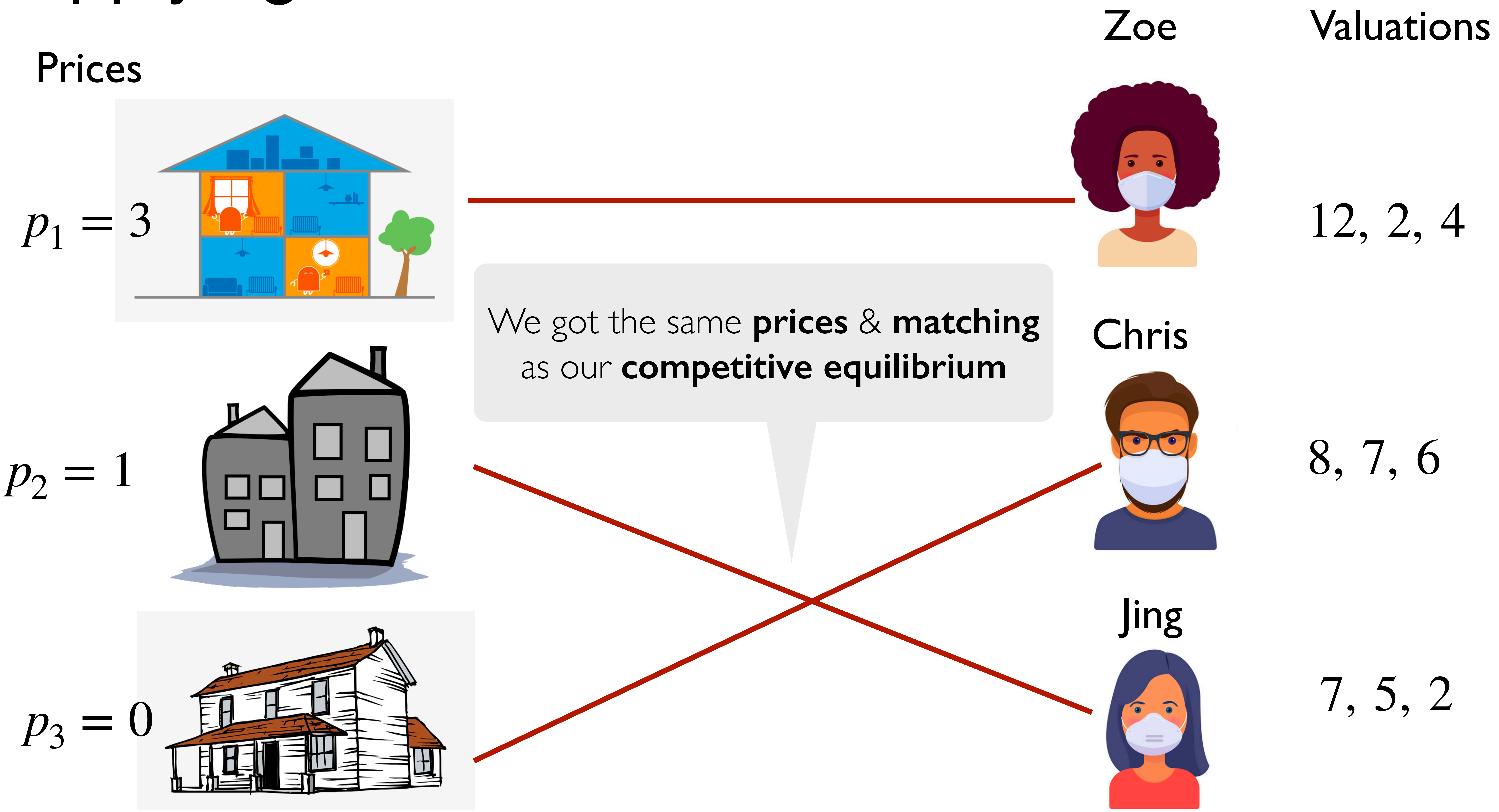
Valuations

12, 2, 4

8, 7, 6

7, 5, 2

Applying VCG



VCG Prices are Market Clearing

- Despite their definition as personalized prices, VCG prices are always market clearing (for the case when each buyer wants a single item)
- Suppose we computed VCG prices for a given matching market
- Then, instead of assigning the VCG allocation and charging the price, we post the prices publicly
    - Without requiring buyers to follow the VCG match
- Despite this freedom, each buyer will in fact achieve the highest utility by selecting the item that was allocated by the VCG mechanism!
- **Theorem.** In any matching market (where each buyer can receive a single item) the VCG prices form the unique set of **market clearing prices of minimum total sum**.

This is a generalization of the VCG/GSP result (where valuations are constrained). The general proof is beyond the scope of this course.

General Demand

- Market clearing prices **may not exist in combinatorial markets**
- **Example**, suppose our market has two items $\{L, R\}$
- Two buyers Alice and Maya
- Alice wants both $v_a(\{L, R\}) = 5$, $v_a(\{L\}) = v_s(\{R\}) = 0$
- Maya wants either, $v_p(\{L\}) = v_p(\{R\}) = v_p(\{L, R\}) = 3$
- What's the welfare-maximizing allocation?
    - Give both to Alice
- What must the price of each be so that Maya doesn't want it?
    - $p(\{L\}) \geq 3, p(\{R\}) \geq 3$
- At a price of ≥ 6 does Alice want it?

Summary

- Ascending price auction is also called Hungarian algorithm in matching literature
- Hungarian algorithm is used to find max-weight bipartite matching
  - Prices are just a conceptual interpretation of "dual" variables
- Caveats:
  - No sales occur until prices have settled at their equilibrium point
  - Coordination required for tie breaks
  - Running time to convergence can be very slow

Competitive Equilibrium Research

- [Left] 2016 Article argues that competitive equilibrium's tie breaking requirement can be fairly strong
  - Use **learning theory** to predict buyer's behavior and demand and show convergence under such some mild assumptions
- [Right 2021]. Algorithms with predictions paper predicts "prices" for faster runtime

Do Prices Coordinate Markets?

Justin Hsu*
Department of Computer and Information Science
University of Pennsylvania
USA
justhsu@cis.upenn.edu

Jamie Morgenstern†
Departments of Computer and Information Science and Economics
University of Pennsylvania
USA
jamiemmt@cs.cmu.edu

Ryan Rogers‡
Department of Applied Mathematics and Computational Science
University of Pennsylvania
USA
ryrogers@sas.upenn.edu

Aaron Roth§
Department of Computer and Information Sciences
University of Pennsylvania
USA
aaroth@cis.upenn.edu

Rakesh Vohra
Economics Department
University of Pennsylvania
USA
rvohra@seas.upenn.edu

Faster Matchings via Learned Duals

Michael Dinitz
Johns Hopkins University
mdinitz@cs.jhu.edu

Sungjin Im
UC Merced
sim3@ucmerced.edu

Thomas Lavastida
Carnegie Mellon University
tlavasti@andrew.cmu.edu

Benjamin Moseley
Carnegie Mellon University
moseleyb@andrew.cmu.edu

Sergei Vassilvitskii
Google
sergeiv@google.com

Abstract

A recent line of research investigates how algorithms can be augmented with machine-learned predictions to overcome worst case lower bounds. This area has revealed interesting algorithmic insights into problems, with particular success in the design of competitive online algorithms. However, the question of improving algorithm running times with predictions has largely been unexplored.

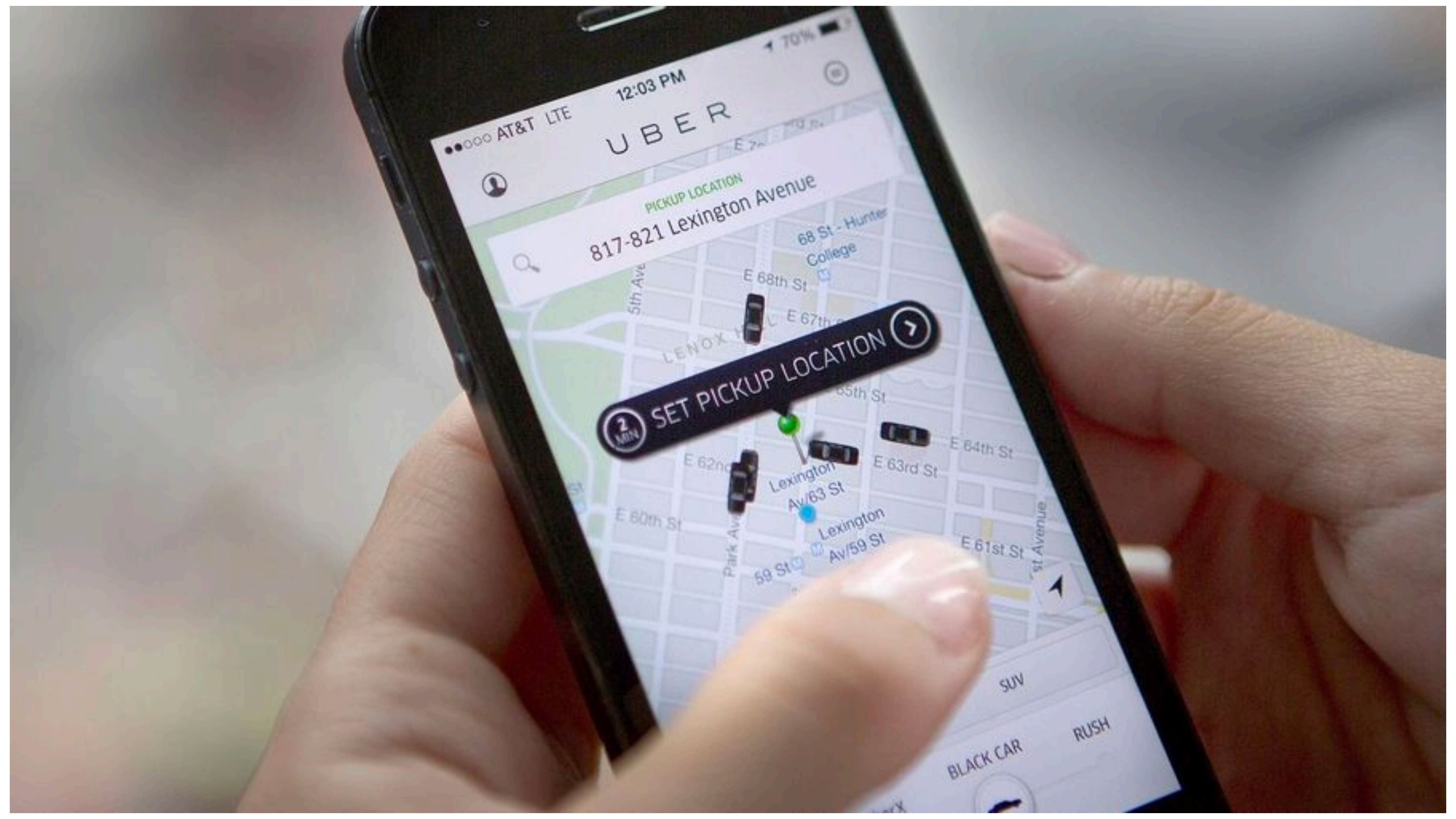
We take a first step in this direction by combining the idea of machine-learned predictions with the idea of "warm-starting" primal-dual algorithms. We consider one of the most important primitives in combinatorial optimization: weighted bipartite matching and its generalization to b-matching. We identify three key challenges when using learned dual variables in a primal-dual algorithm. First, predicted duals may be infeasible, so we give an algorithm that efficiently maps predicted infeasible duals to nearby feasible solutions. Second, once the duals are feasible, they may not be optimal, so we show that they can be used to quickly find an optimal solution. Finally, such predictions are useful only if they can be learned, so we show that the problem of learning duals for matching has low sample complexity. We validate our theoretical findings through experiments on both real and synthetic data. As a result we give a rigorous, practical, and empirically effective method to compute bipartite matchings.

The Myth of the Invisible Auctioneer

- One fundamental assumption when we executed the ascending price mechanism to compute market-clearing prices is:
    - The market does not actually clear until prices have settled at their equilibrium point
- As if an invisible auctioneer is coordinating the prices and lets the market know when the prices have converged and trade can actually take place
- In practice, one might imagine that sales are actually happening concurrently with price adjustment

Fluctuations in Practice: Research

- In practice, one might imagine that sales are actually happening concurrently with price adjustment
- It turns out, the way buyers and sellers respond to prices in the short-run can dramatically influence prices
- **Example**. Surge pricing on ride-sharing platforms can be viewed as an attempt to find market-clearing prices
- However, if passengers and drivers respond to prices myopically, the resulting behavior can be erratic
- Recent research in AGT studies **dynamic (online) resource allocation problems** that take these factors into account



Project Overview

Final Project

Overview In the final project, you must analyze the **role of incentives and strategic behavior** in a particular **application or class of algorithms**. The domain you choose can be similar to ones we studied in class, for example:

- Markets with money: Auctions, matching markets with money, resource allocation etc.
- Markets without money: school choice, kidney exchange, one or two-sided matching, voting, fair division, incentives in tournaments, etc.
- Decentralized systems: Incentives in P2P systems, network routing etc.
- Game theory: Any problem that can modeled as game and can be analyzed using equilibrium concepts.

Learning Goals There are several learning goals of the final project:

- To dig deeper in one of the areas of algorithmic game theory.
- To read and understand research papers in the field.
- To apply the framework and concepts developed in class to a new regime.

Choosing a Topic As you choosing a project topic, keep the following in mind:

- Pick a topic you will enjoy working on (something fun!)
- Pick a topic that you will learn a lot from (something useful!)
- Pick a topic that is relevant to the course (something relevant!)
- Pick a topic with the scope in mind (~3 weeks)
- Pick a topic that is technical interesting: relevant research is published in CS/Econ conferences.

Theory vs Implementation Your project may be purely implementation or purely theoretical (or anywhere in between). If you are choosing an implementation project, keep in mind:

- Scope and effort should match the timeline (3x a regular assignment)
- Even when implementing an algorithm, it is important to understand the theoretical foundations behind the work and put the implementation in context with the theory in your write up.

If you choose to work on a purely theoretical topic, keep in mind that you are expected to go above and beyond summarizing existing results. You do not need to solve an open problem to do this, your contribution can take many forms:

- Really understanding the literature in a way that you can recreate the results
- Filling in details and providing context and examples that the papers may have overlooked
- Finding something new to say about the existence work
- Identifying interesting directions to take the literature forward (even if there is not enough time to pursue them)

It is important to note that purely theoretical projects can often be more challenging and open-ended, with a final product that may feel less tangible. As a result, most projects in the past were somewhere in the middle: understanding the theory behind a market or algorithms and evaluating its properties through a series of simulations.

Sample Projects and Timeline See timeline and rubric for the checkpoints and grading rubric. See Project Ideas and Sample Student Projects.

Project Ideas and Timeline

- Project Ideas: https://docs.google.com/document/d/1gJhycwdkcLXsFyO-Mfr9wamcguipXYF7T29MStwIx84/edit?tab=t.0
- Rubric and timeline: https://docs.google.com/document/d/1FS8HjeGNSDKFKpDzzSrJyEAjB0M9_pr9cHyPM0eGpX4/edit?tab=t.0
- Sample Student Projects:

Project Ideas and Timeline

- Project Ideas:
    - https://docs.google.com/document/d/1gJhycwdkcLXsFyO-Mfr9wamcguipXYF7T29MStwIx84/edit?tab=t.0
- Rubric and timeline:
    - https://docs.google.com/document/d/1FS8HjeGNSDKFKpDzzSrJyEAjB0M9_pr9cHyPM0eGpX4/edit?tab=t.0
- Sample Student Projects on GLOW:
    - https://glow.williams.edu/courses/4311932/files/folder/Project%20Files

Decentralized Markets without Money

Motivation: Incentives in P2P

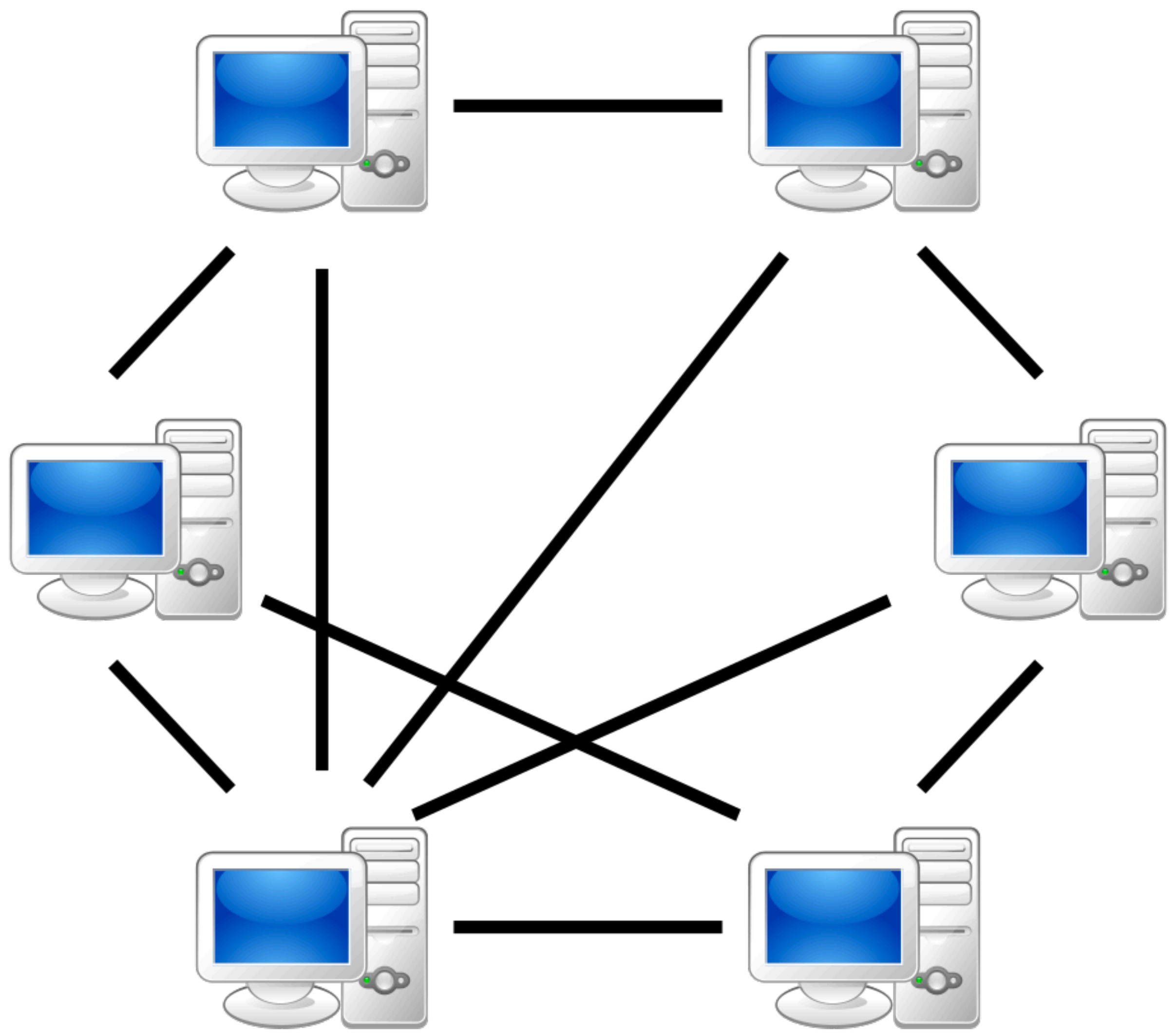
- Peer-to-peer (P2P) systems provide a case study of how a system evolves in response to decentralized incentive issues
- Peer-to-peer file sharing:
    - Distribute a file between users where they upload and download from each other in a distributed network
- P2P is now fundamental to blockchain platforms, such as Bitcoin and Ethereum
- **AGT view**: **do peers in a P2P system to have an incentive to cooperate**?

Failure of Centralization

- In the days of early internet, file sharing was done in an ad hoc way
- **Napster (1999):** provided a centralized, searchable directory listing which users have copies of various files (e.g. mp3s)
    - Matchmaker (matched up people who want file to people who have the file)
    - File transfer was then done directly between users
- Lawsuits against Napster for copyright infringement (2000s)
    - By RIAA, Metallica, etc
- After Napster failed to comply, it was shut down in 2001
- Napster's rise (25 million users) pointed to the demand for such systems but its failure motivated **decentralized designs**

Benefits of P2P

- Client-server model: server provider is associated with the server machines, users device is a client machine
    - These platforms need to make use of millions of distributed servers in order to cache content on machines close to users to provide low latency and maintaining this infrastructure
- In contrast, P2P systems there is no distinction between client and servers: each computer acts as both and is called a peer
- Main advantage: can scale well to large numbers of users while keeping the costs low for the initial uploader of the content
    - Provide robustness by avoiding a single point of failure
- Disadvantages: no control over content and who will download it, for how long the files will be available, etc



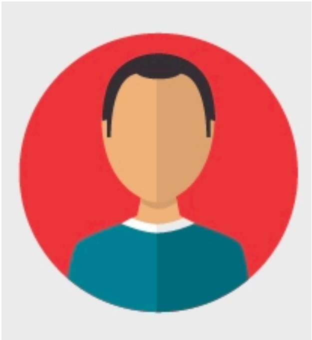
Decentralized: Gnutella

- First decentralized P2P network of its kind
- Design highlighted various incentive issues inherent in P2P networks
- Functionality rested on users conforming to the reference behavior
- Users were not given any incentive to actually behave in this way
- **Free-riding in Gnutella**:
    - A user is a free-rider who downloads but never uploads
  - A study by researchers showed that free-riding was the dominant behavior in Gnutella: 2/3rd of the users were free riders
  - In follow up study in 2005, free riding **had climbed to 85%** leading to the extinction of the system

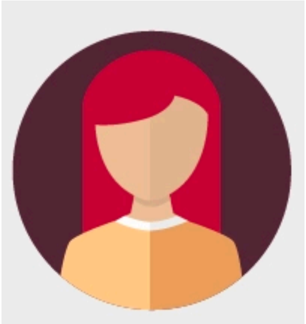
File Sharing Game

- Consider two players: Aamir and Beth
- Aamir has a file that Beth wants and vice versa
- They simultaneously and independently decide whether or not to upload the requested file
- For each player, the benefit of receiving the file is 3 and the cost of uploading is 1 (bandwidth charges, opportunity costs, etc)

Aamir



Beth



	Upload	Don't Upload
Upload	$2, 2$	$-1, 3$
Don't Upload	$3, -1$	$0, 0$

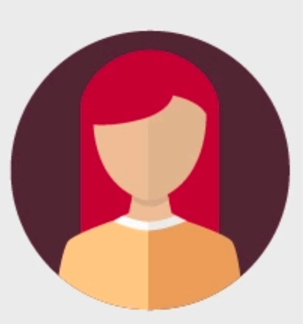
Prisoner’s Dilemma

- Our payoff matrix is just a variant of the prisoner’s dilemma game from Lecture 2
- Each player has a strictly dominant strategy to defect
    - In this case, to not upload
- When Aamir and Beth play their dominant strategy neither uploads and each gets a payoff of zero
- Prisoner’s dilemma summarizes the essential conflict between individual good and the collective good

	Upload	Don’t Upload
Upload	$2, 2$	$-1, 3$
Don’t Upload	$3, -1$	$0, 0$

Aamir

Beth



	C	D
C	a, a	b, c
D	c, b	d, d

$$c > a > d > b$$

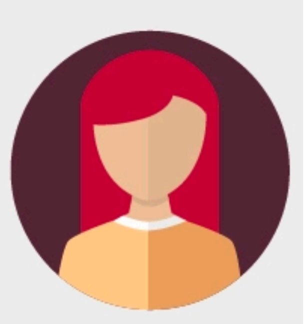
Repeated Prisoner's Dilemma

- In real life examples of Prisoner's dilemma players do seem to cooperate: how can we explain this?
- **Intuition**: game is not played once but repeatedly
- Need to analyze equilibrium in sequential games

	Upload	Don't Upload
Upload	$2, 2$	$-1, 3$
Don't Upload	$3, -1$	$0, 0$

Aamir

Beth



	C	D
C	a, a	b, c
D	c, b	d, d

$$c > a > d > b$$

Split or Steal

- Nash equilibrium no longer a good equilibrium if players act in rounds
- Players can choose split or steal the prize money
  - If both steal, no one gets any money
  - If one splits, other steals: the thief gets all the money
  - If both split: they share the only in half
- Weakly dominant action?
  - Steal weakly dominates Split for both players
- In both the video game and game show, the game
  is multi-stage and current decisions have future consequences
- https://www.youtube.com/watch?v=S0qjK3TWZE8&ab_channel=spinout3

	Split	Steal
Split	1/2, 1/2	0, 1
Steal	1, 0	0, 0