CSC200: Lecture 17

- Today
  - Review discussion of combinatorial auctions and VCG mechanism (not in text)
  - Start of Matching Markets: strategic interaction on networks
- Readings:
  - Ch.10 (*including* advanced material in 10.6)
  - After this, we'll look at *stable* matching problems (material not covered in the text): this make take us to the end of the term
  - After the break, we'll consider internet advertising: combination of matching, auctions, and web integration
  - Then we'll look at various network phenomena at a more aggregate (population) level (Ch.16-18), e.g., information cascades, power laws, tipping points, etc.
  - Lots of topics still to be considered
- Announcements
  - Assignment due Wednesday
  - Term test Friday: Room 1088 will try to start promptly at 3PM; 3:10 in 1069. Scope of test: social-affiliation networks, Nash equilibria, auctions (with CAs)
  - Questions about price of anarchy (POA)?

Clarification of social welfare and POA

- After class, one student indicated that what I was saying was different from what he found on Wikipedia. This is true. There are two ways in which there is a difference.
- One is just a matter of convention (for maximization problems) in terms of whether the OPT value is in the numerator or the denominator and I prefer that it be in the numerator so that the ratio is greater than or equal to 1.
- But the more substantial difference is that say for an auction I am defining the social welfare as a sum of the values for the allocated items whereas Wikipedia is saying that it is the sum of the utilities.
- Given that I define an agents utililty (for a set of allocated items S) as value(S) –price(S), this does seem like a substantial difference. Turn to next slide for explanation.

Clarifying the definition of social welfare

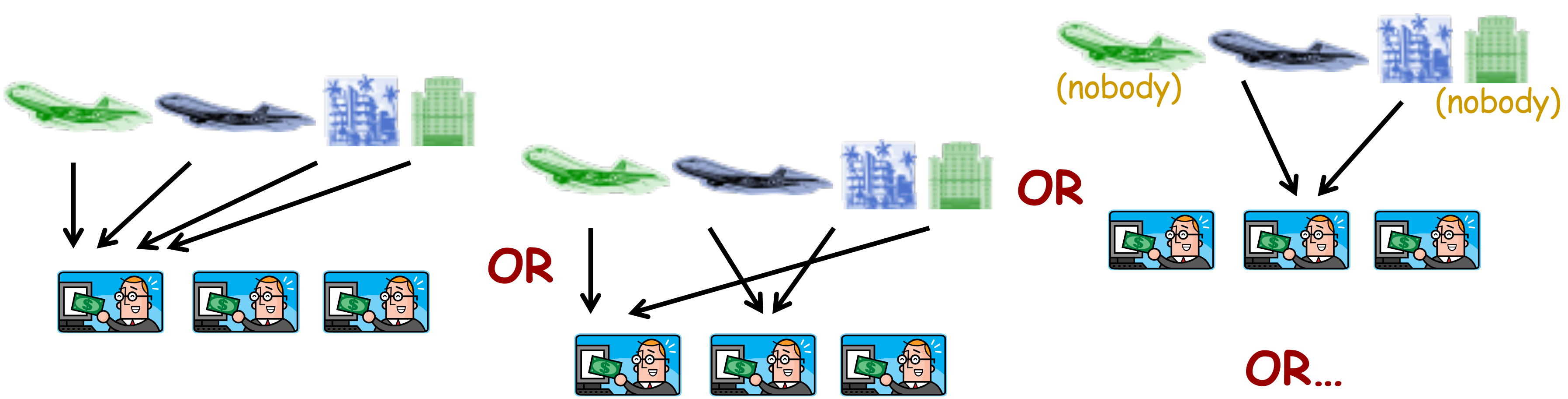
- I define social welfare in terms of the buyers valuations.
- Wikipedia in terms of their utilities.
- I could have used a more neutral term: *payoff*
- The way to reconcile this is to decide whether or not to include the seller (or mechanism) as an agent. I define the payoff of each buyer as their valuation for any set allocated. I was not viewing the seller/mechanism as an agent and not considering the seller in the social welfare.
- But it is also reasonable to think of the seller/mechanism as an agent and if so then the sellers payoff is the sum of prices (for allocated sets) and a buyers payoff is their utilitity. Summing the payoff of all buyers and the seller is the same as summing all buyer values for sets allocated.

Combinatorial Auctions

- Formally:
  - a collection of *goods* G for sale
  - *bids* have the form $\{(S_1, v_1), (S_2, v_2), \ldots, (S_k, v_k)\}$, where:
    - each S_i is a subset of G, v_i is the price bidder will pay for S_i
    - can assign to any bidder *at most one* subset S_i from his bid
- Goal find an *assignment of goods* to bidders that maximizes the sum of the corresponding prices/valuations
  - i.e., if bidder gets the items that correspond to S_{17} in his bid, he will pay v_{17}; if items correspond to no subset in his bid, he pays nothing
  - sometimes "free disposal" assumed…
- But each item can be assigned to at most one bidder, so some hard choices need to be made by the seller

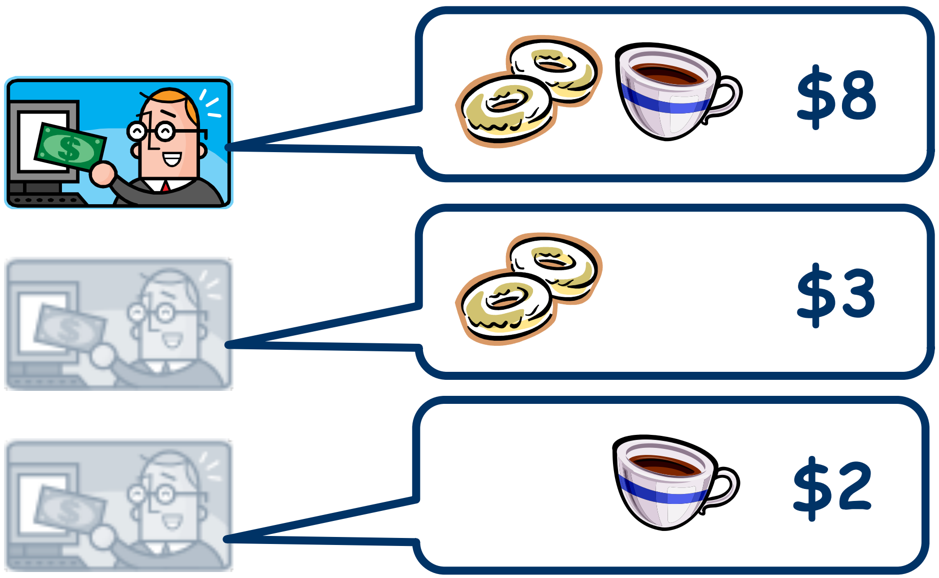
Combinatorial Auctions: Complexity

- Deciding how to allocate goods to bidders to maximize revenue (or social welfare) is computationally difficult (i.e. set packing problem)
  - formally, it is an *NP-complete problem*, which means that it is widely believed to require *exponential time to solve in the worst-case*
  - informally, it is not known whether you can do much better (in the worst case) than exhaustively searching all ways of assigning items to bidders
    - even *approximately* maximizing the social welfare is NP-hard
    - Recall our early discussion of computational difficulty
  - if I have *n* items and *m* bidders, there are $(m+1)^n$ such assignments (allow for the possibility of an item going to no bidder)



Other Issues

- A variety of interesting strategic issues:
  - *envy-free*: find an allocation (and prices to charge) so that no bidder would prefer the bundle of goods allocated to a different bidder
  - *stability*: find payments so no group of bidders could offer to pay more for a set of goods allocated to others, divide it up, and be better off
- *What about pricing:* if people pay what they bid (1^{st}-price), they will obviously hedge their bids; is there an analog of a 2^{nd}-price auction in the combinatorial setting? What would the 2^{nd} highest bid be?



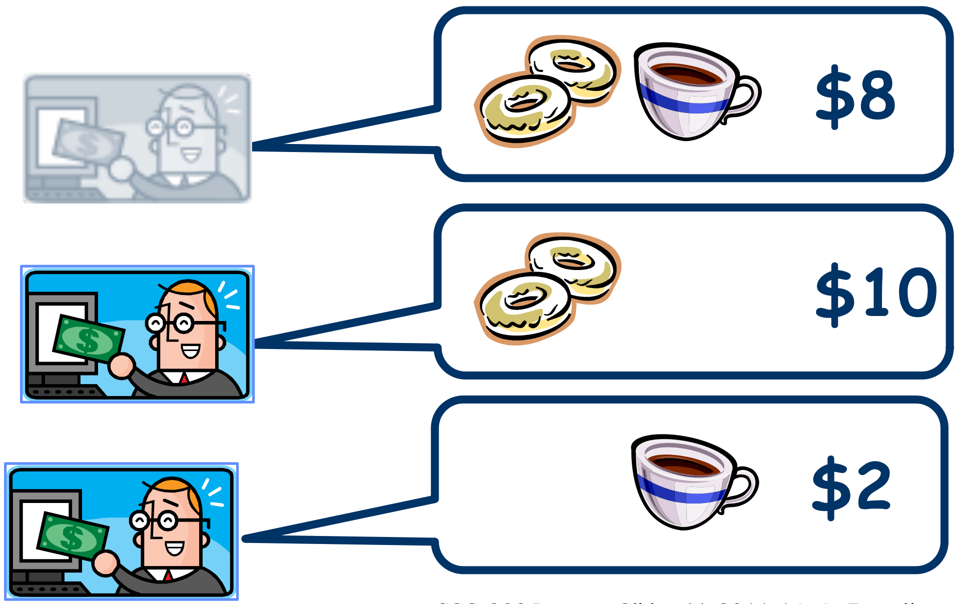
Winner gets bundle of both items. But no other bidder offered a bid on the same bundle: so what price should we charge using "2^{nd} price"?

The VCG Mechanism

- There is a generalization of the 2nd-price (or Vickrey) auction to CAs
  - The *Vickrey-Clarke-Groves (VCG)* mechanism
- Lets assume (and this is a big assumption) that we know how to allocate items optimally. Now how to charge?
- Roughly speaking, you charge someone based on the "externality" they inflicted on other players by their presence
  - e.g., bidder *X* gets bundle *B*, this (potentially) prevented other bidders *Y, Z, …* from getting (some parts of) bundle *B*
  - figure out social welfare that *Y, Z, …* got in the actual auction
  - then pretend *X* didn't exist and figure *SW* that *Y, Z, … would have attained* if X hadn't bid/didn't exist (can't be any worse, could be higher)
  - charge X the difference of the two: what he cost *Y, Z,…* by his presence
- In previous slide winner pays $5 for coffee and donuts.
- Notice that 2nd-price auction is a special case of VCG with one item
- We'll see VCG in more detail in Ch.15 (advertising auctions)

Another combinatorial auction

- Agents can have very different valuations:
  - *What about pricing:* if people pay what they bid (1st-price), they will obviously hedge their bids; is there an analog of a 2nd-price auction in the combinatorial setting?



To maximize social welfare we have two winners if agents bid truthfully: second agent gets donuts while third agent gets coffee. How would VCG price the sets?

Matching Markets

- Auctions: how to sell (or buy) things
  - So far we did not consider *multiple buyers and sellers* simultaneously
  - looked briefly (but not in depth) at *multiple items* (CAs)
  - did not consider potential restrictions on which buyers could interact with which sellers
  - we'll begin to look at these additional issues that arise by first using a very stylized model: *bipartite matching*
- Many settings where matching is critical (and difficult)
  - prime example is a stock market: buy/sell orders must be matched
  - combinatorial auctions: matching buyers to (bundles of) items
  - advertising auction: matching bidders to "user eyeballs" satisfying certain characteristics (see Ch.15: January)
  - in many cases, we need to find matchings that make people happy even though no money changes hands…

Future topic: Ad Allocation as a Matching Market

- Tie back to internet (the "ultimate network"):
  - Advertising auctions: special case of a *one-sided matching market*
  - single seller (Google, MSFT, …) with a set of items (ad slots)
  - a lot of agents interested in buying these slots
  - trying to find an allocation of items to potential buyers that is "good" (maximize social welfare, revenue to seller, …)
  - we'll look at this in depth in January: immediately after the break

Google™

query: loan consolidation

Position 1

Position 2

Position 3

Position 4

Matching Markets

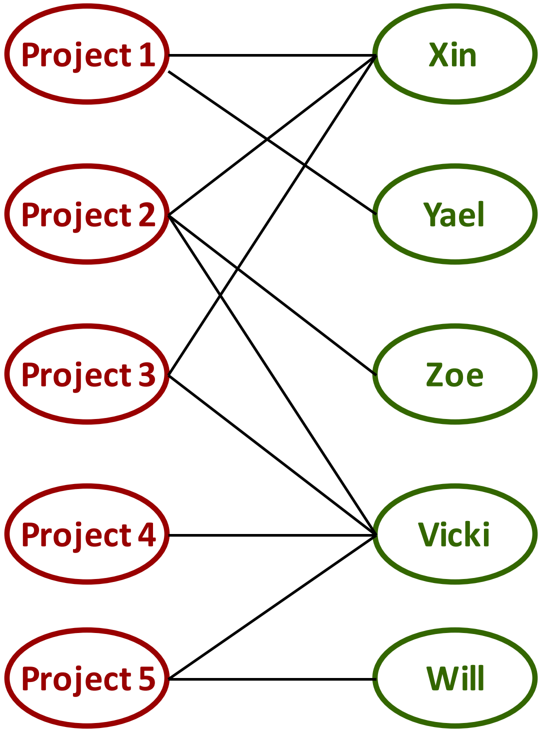
- *One-sided matching markets:*
  - finite set of items *Y* (goods, tasks, services, etc.)
    - single-seller or multiple sellers
      - *"seller" used loosely: doesn't mean money necessarily changes hands*
    - perhaps "multi-unit" or with capacities
  - set of agents (individuals) *X* with preferences/values for different items
  - goal: assign items *Y* to agents *X* to satisfy some objective
    - e.g., maximize social welfare, be in equilibrium, …
    - we'll assume each agent wants at most one item; i.e. unit demand
- Examples:
  - online advertising slots as we will see in Ch.15
  - assigning students to (unshared) dorm rooms, or to schools
  - assigning tasks to employees
  - assigning conference papers to reviewers
  - awarding government contracts to contractors
  - and on and on…

Matching Markets

- Limitation of one-sided: only agents have preferences
  - even in examples above, the sellers might have preferences
  - e.g., company wants to assign tasks to *most* qualified employee (but not possible, since best employees would do all the work!)
- *Two-sided matching markets:*
  - two sets of agents, *X* and *Y*
    - *X* and *Y* could be the same set in some instances
  - any *x* in *X* has preferences over *Y*, any *y* in *Y* has preferences over *X*
  - goal: match agents in *X* to agents in *Y* to satisfy some objective
- Examples:
  - hiring workers/interns (companies *X* and interviewees *Y*)
  - assign tasks *X* to employees *Y* (tasks prefer better qualified employees)
  - matching men *X* and women *Y* (classic: stable marriage problem)
  - assign pairs of students to *shared* dorm rooms
    - basically, matching students with each other ($X=Y$)

Bipartite Graph Model

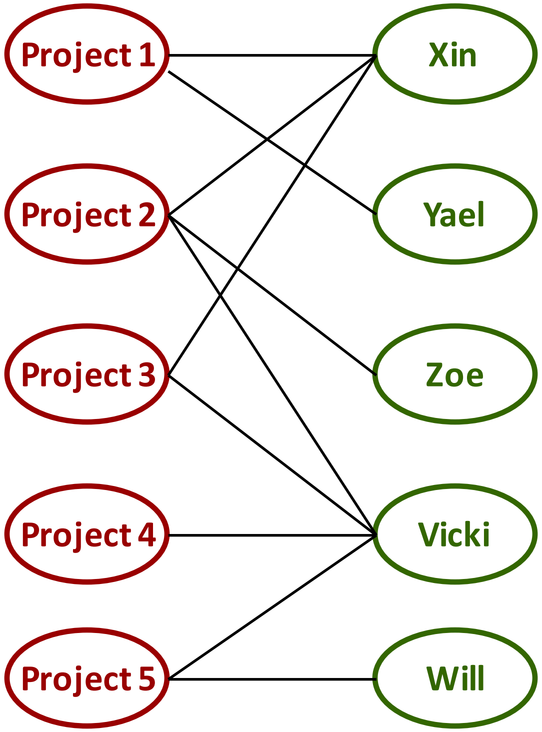
- Consider simple one-sided model:
  - "items" *Y*, agents *X*
  - *simple preferences:* for any *x, y,* either *x* satisfied with *y*, or not satisfied with *y*
- Easy to visualize as a *bipartite graph*
  - nodes *Y*, nodes *X*, edge between *x* and *y* iff *x* is satisfied with *y*
- Example: five students assigned to five summer research projects/faculty advisers
- *One can model two-sided simple preferences this way too: how?*

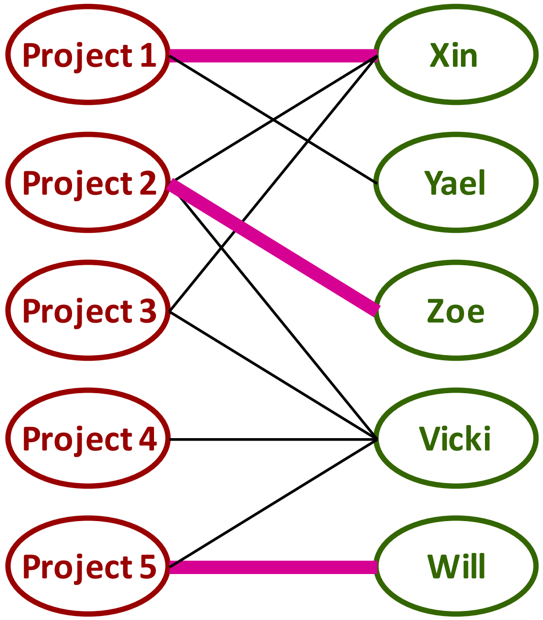


e.g., Vicki would be happy with four different projects (P2, P3, P4, P5); Will only wants one (P5)

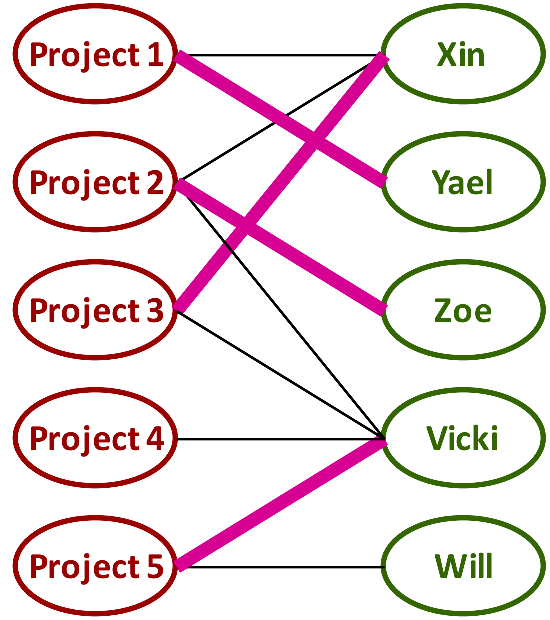
Bipartite Matchings

- A *matching* in a graph *G=(V,E)* is subset $E' \subseteq E$ of the edges with no node participating in more than one edge.
- In a bipartite graph, a matching assigns:
  - no more than one agent to an item
  - no more than one item to an agent
- A matching is *maximal* if there is no way to add another edge to it (and still be a matching)
- Can also define max size, or max weight matchings (if edges have weights)
- Maximum implies maximal
- A bipartite matching is *perfect* iff each node is matched
  - can only exist if graph has equal number of nodes on both sides

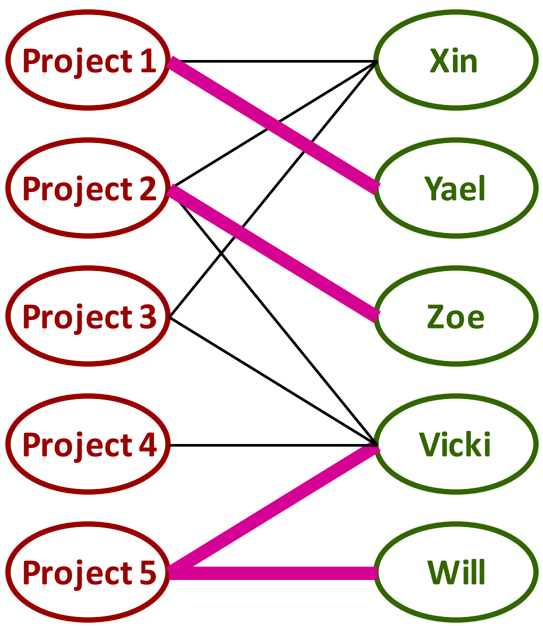




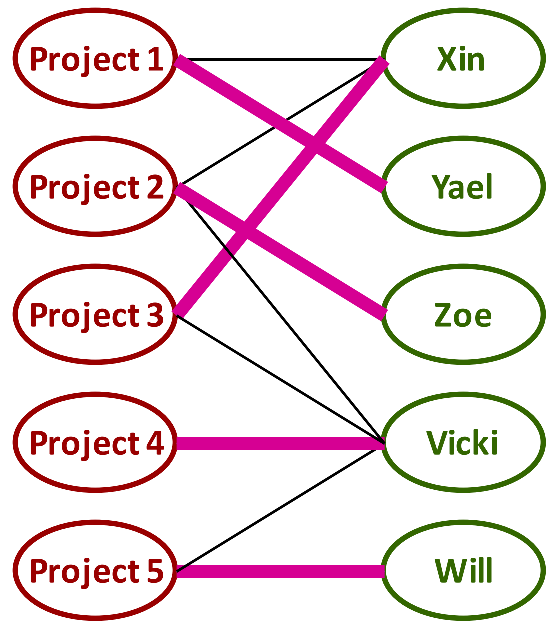
Matching: yes
Maximal: no
Maximum: no
Perfect: no



Matching: yes
Maximal: yes
Maximum: no
Perfect: no



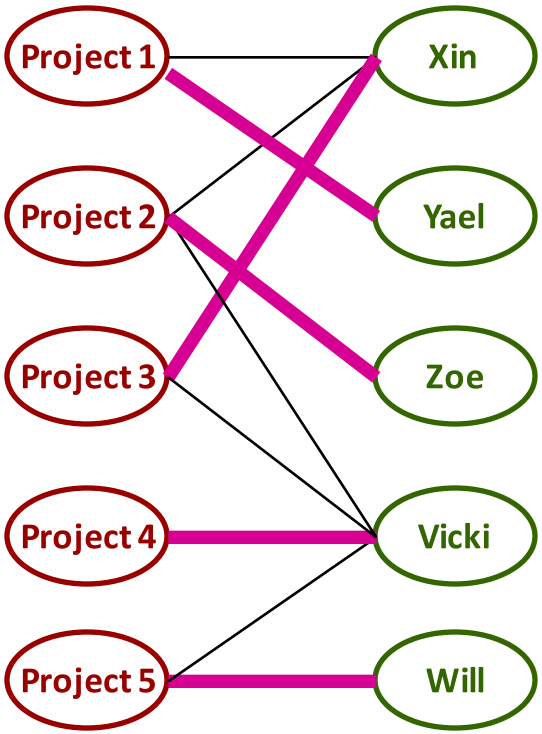
Matching: no



Matching: yes
Maximal: yes
Maximum: yes
Perfect: yes

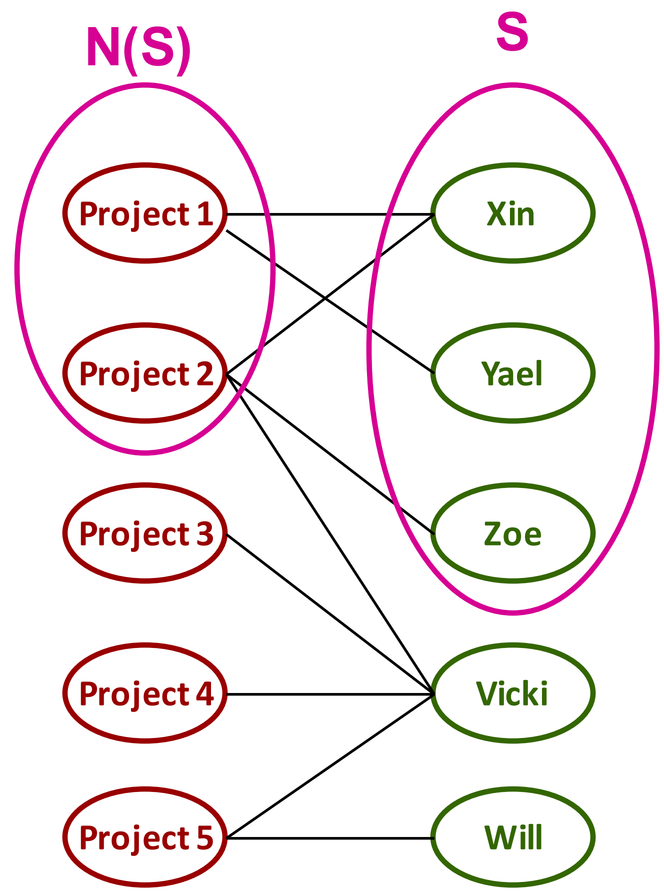
Perfect Matchings and Social Welfare

- Assume equal number of agents, items
- A maximum matching gives us an assignment of items to agents that maximizes total preference (social welfare) in this simple model of preferences
- If perfect matching exists, then everyone is satisfied; if not someone will be left unsatisfied
- If one doesn't exist, how can you know?
  - *Don't want to enumerate all matchings!*
- A simple "test" can verify non-existence



Constricted Sets

- This graph has no perfect matching
- Why?
  - *X, Y, Z* only collectively satisfied with *P1, P2*: not enough projects for the three of them
- If *S* is a subset of agents, let *neighbor set N(S)* be the set of all items they are connected to
- *Constricted set:* any set *S* whose neighbor set *N(S)* is smaller than *S* itself
- If *G* has a constricted set, then obviously there is no perfect matching
- **Matching Theorem:** If *G* has no perfect matching, then it must have a constricted set



- We'll look briefly at an algorithm for constructing matchings next time
  - if it does not find a perfect matching, it will identify a constricted set