Distributed Software Development
Self-interested MAS

Chris Brooks

Department of Computer Science
University of San Francisco

Engineering systems vs Engineering agents

- Recall that at the end of Thursday's class, we were talking about ant algorithms.
  - By specifying a simple set of rules, we can achieve interesting large-scale behavior.
- Ant-type approaches lead us to think about how we can build systems that produce the effects we want.
- "Given that agents will act in a particular way, how can we constrain the environment to achieve a desirable outcome?"
- This method of problem solving is best applied to problems involving self-interested agents.

Preferences and Ut

- Agents will typically have preferences over outcomes
  - This is declarative knowledge about the relative value of different states of the world.
  - "I prefer ice cream to spinach"
- Often, the value of an outcome can be quantified (perhaps in monetary terms.)
- This allows the agent to compare the utility (or expected utility) of different actions.
- A rational agent is one that maximizes expected utility
- Self-interested agents each have their own utility function.

Rationality and protocol design

- By treating participants as rational agents, we can exploit techniques from game theory and economics.
- Assume everyone will act to maximize their own payoff
- How do we structure the rules of the game so that this behavior leads to a desired outcome?
- This approach is called *mechanism design*.

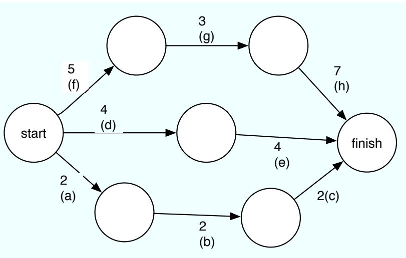
Example: Clarke tax

- Assume that we want to find the shortest path through a graph.
- Each edge is associated with an agent.
- Each edge has a privately known transmission cost.
  - Agents might choose to lie about their transmission cost.
- How can we find the shortest path?

Clarke

- Rule:
  - Accept each agent's bid.
  - If they are not on the shortest path, they get 0.
  - If they are on the shortest path, they get:
    - Cost of next shortest path - (cost of shortest path without their contribution).

Example



- Assume each agent bids truthfully.
- Agents A, B, and C are each paid 8 - (6 - 2) = 4
  - This is their contribution to the 'best solution'
- Other agents are paid nothing.

Example

- Why is truth-telling a dominant strategy?
  - What if a underbids?
    - A bids 1: paid 8 - (5 - 1) = 4. No benefit.
  - What if A overbids?
    - A bids 3: paid 8 - (7 - 3) = 4. No benefit.
    - A bids 5. No longer on the shortest path, so A gets 0.
  - What if d underbids?
    - D bids 3: no change.
    - D bids 1: paid 6 - (5 - 1) = 2. But his cost is 4.
  - D overbids: no change.

Solution conce

- So how do we evaluate an algorithm or protocol involving self-interested agents?
- Some solutions may be better for some agents and worse for others.
  - Example: cake-cutting problem
- We know that each agent will try to maximize its own welfare
- What about the system as a whole?

Solution concepts

- There are a number of potential solution concepts we can use:
  - Social welfare - sum of all agent utility.
  - Pareto efficiency
    - Is there a solution that makes one agent better off without making anyone worse off?
  - Individual rationality
    - An agent who participates in the solution should be better off than if it hadn't participated.
  - Stability
    - The mechanism should not be able to be manipulated by one or more agents.
- It's not usually possible to optimize all of these at the same time.

Stability

- Ideally, we can design mechanisms with *dominant strategies*
  - A dominant strategy is the best thing to do no matter what any other agent does.
  - In the previous example, truth-telling was a dominant strategy.
  - We would say that the mechanism is non-manipulable. (lying can't break it.)
- Unfortunately, many problems don't have a dominant strategy.
- Instead, the best thing for agent 1 to do depends on what agents 2,3,4,... do.

Nash equilibr

- This leads to the concept of a *Nash equilibrium*
- A set of actions is a Nash equilibrium if, for every agent, given that the other agents are playing those actions, it has no incentive to change.
- Example: big monkey and little monkey
  - Monkeys usually eat ground-level fruit
  - Occasionally they climb a tree to get a coconut (1 per tree)
  - A Coconut yields 10 Calories
  - Big Monkey expends 2 Calories climbing the tree.(net 8 calories)
  - Little Monkey expends 0 Calories climbing the tree. (net 10 calories)

Nash equilibrium

- If BM climbs the tree
  - BM gets 6 C, LM gets 4 C
  - LM eats some before BM gets down
- If LM climbs the tree
  - BM gets 9 C, LM gets 1 C
  - BM eats almost all before LM gets down
- If both climb the tree
  - BM gets 7 C, LM gets 3 C
  - BM hogs coconut
- How should the monkeys each act so as to maximize their own calorie gain?

Nash equilibrium

- Assume BM decides first
  - Two choices: wait or climb
- LM has four choices:
  - Always wait, always climb, same as BM, opposite of BM.

Nash equilibr

- What should Big Monkey do?
- If BM waits, LM will climb (1 is better than 0): BM gets 9
- If BM climbs, LM will wait :BM gets 4
- BM should wait.
- What about LM?
  - LM should do the opposite of BM.
- This is a Nash equilibrium. For each monkey, given the other's choice, it doesn't want to change.
- Each monkey is playing a *best response*.

Nash equilibrium

- Nash eqilibria are nice in systems with rational agents.
- If I assume other agents are rational, then I can assume they'll play a best response.
- I only need to consider Nash equilibria.
- They are *efficient* (in the Pareto sense).
- Problems:
  - There can be many Nash equilibria. (the cake-cutting problem has an infinite number of Nash equilibria)
  - Some games have no Nash equilibrium.
  - There may be ways for groups of agents to cheat.

Selecting between equilibria

- Given that there are lots of possible Nash equilibria in a problem, how does an agent choose a strategy?
- In some cases, external forces are used to make one equilibrium more attractive.
  - Government regulation, taxes or penalties
- In other cases a natural *focal point* exists.
  - There is a solution that is attractive or sensible *outside the scope of the game*.

Bilateral and multilateral negotia

- There are two different ways that we can think about agents negotiating or bargaining with each other.
- Bilateral: negotiation happens one-on-one.
  - Game theory is applicable here.
- Multilateral: Many agents negotiate simultaneously.
  - Markets and auctions are appropriate here.

Auctions

- An auction is a negotiation mechanism where:
  - The mechanism is well-specified (it runs according to explicit rules)
  - The negotiation is mediated by an intermediary
  - Exchanges are market/currency-based
- Agents place bids on items or collections of items.
- An auctioneer determines how goods are allocated.
- Requirements: the auction should be fair, efficient, easy to use, and computationally efficient.
- We'll need to trade these against each other.

Auctions

- Private-value auctions are easier to think about at first.
- In this case, the value agent A places on a job has nothing to do with the value that agent B places on the object.
  - For example, an hour of computing time.
- In common-value auctions, the value an agent places on an item depends on how much others value it.
  - Example: art, collectibles, precious metals.

English aucti

- An English (or first-price) auction is the kind we're most familiar with.
- Bids start low and rise. All agents see all bids.
- May be a reserve price involved.
- Dominant strategy: bid ϵ more than the highest price, until your threshold is reached.
- Problems: requires multiple rounds, not efficient for the seller, requires agents to reveal their valuations to each other.
- There may be technical problems to solve with making sure all agents see all bids within a limited period of time.

First-price sealed-bid auction

- Each agent submits a single sealed bid. Highest wins and pays what they bid.
  - This is how you buy a house.
- Single round of bidding. All preferences remain private.
- Problems: No Nash equilibrium - agents need to counterspeculate. Item may not go to the agent who valued it most. (inefficient).

Dutch auction

- Prices start high and decline.
- First agent to bid wins.
- Strategically equivalent to first-price sealed-bid.
- In practice, closes quickly.

Vickrey auc

- The Vickrey, or second-price, auction, has a number of appealing aspects from a computational point of view.
- Single round of bidding.
- Efficient allocation of goods.
- Truth-telling is the dominant strategy.
- Rule: each agent bids. Highest bid wins, but pays the second price.
  - (the example we used earlier is isomorphic to the Vickrey auction).

Example

- Angel, Buffy and Cordelia are bidding on a sandwich.
  - Angel is willing to pay $5, Buffy $3, and Cordelia $2.
- Each participant bids the amount they're willing to pay.
- Angel gets the sandwich and pays $3.

Proof

- Let's prove that truth-telling is a dominant strategy.
- Angel:
  - If he overbids, he still pays $3. No advantage.
  - If he bids between $3 and $5, he still pays $3. No advantage.
  - If he bids less than $3, then he doesn't get the sandwich - but he was willing to pay $5, so this is a loss.

P

- Buffy (the same reasoning will hold for Cordelia)
  - If she bids less than $3, she still doesn't get the sandwich. (notice that we assume she doesn't care how much Angel pays.)
  - If she bids between $3 and $5, she still doesn't get the sandwich. No benefit.
  - If she bids more than $5, she gets the sandwich and pays $5. But she was only willing to pay $3, so this is a loss.

Vickrey Auctions in real life

- Because of these properties, Vickrey auctions have been adopted for:
  - Placement of Google ads
  - Allocation of computer resources
  - Distribution of electrical power
  - Bandwidth allocation
  - Scheduling problems.
- Interestingly, they are not widely used in human auctions.
  - Perhaps people are not rational ...

Advantages and disadvantages

- Advantages of the Vickrey auction/Clarke tax:
  - Truth-telling as a dominant strategy
    - Easy for participants, no need for multiple rounds of bidding.
    - Most efficient solution is always discovered.
  - Disadvantages:
    - Leaves money 'on the table' (payments are more than cost of job)
    - Payments are a function of the quality of the second-best solution.
    - Not intuitive for humans.

Common and correlated-va auct

- Everything we've said so far applies only to private value auctions.
- Common or correlated-value auctions are much less predictable.
- In particular, common-value auctions are subject to the *winner's curse*
  - As soon as you win a common-value auction, you know you've paid too much.

Winner's curse

- Example: Oil drilling
  - Suppose that four firms are bidding on drilling rights. Each has an estimate of how much oil is available in that plot.
    - A thinks \$5M, B thinks \$10M, C thinks \$12M, and D thinks \$20M.
    - Let's say it's really \$10 M, but the firms don't know this.
  - In an English auction, D will win for \$12M+1
  - They lose \$2M on this deal.
  - Problem: The winner is the firm who tended to overestimate by the most.
  - (Assumption: all firms have access to the same information.)

Winner's curse

- This also explains why sports free agents seem to underperform their contracts.
  - They're not underperforming, they're overpaid.
- How to avoid the winner's curse:
  - Better information gathering
  - Caution in bidding

Combinatorial aucti

- Often, goods that are being sold in an auction have complementarities.
  - Owning one good makes a second good more valuable.
- For example, let's say supercomputer access is sold in 1-hour increments.
- Lab 1 needs three hours before 5 pm - less time is worthless.
- Lab 2 needs two hours before noon.

Combinatorial auctions

- How to approach this:
  - 1. Separate auctions for each hour.
    - Complicated rules for backing out and reallocating needed.
  - 2. Auction combinations (or bundles) of goods.

Winner-determination problem

- Finding the winner for a single-item Vickrey auction is easy.
- Finding the winner for a combinatorial auction is (computationally) hard.
- Formulation:
  - Given: n bidders, m items
  - Let a bundle S be a subset of the m items.
  - A bid b is a pair (v,s), where v is the amount an agent will pay for s.
  - An allocation $x_i(S)$is described by a mapping from (i,s) into {0,1}. ((i,s) = 0 if i does not get s, and 1 if he does.)

Winner-determination prob

- We can then write the winner-determination problem as an optimization problem:
  - Find the set of allocations that maximizes: $\sum_{i \in N}(v,s)x_i(s)$
  - This problem can be solved in a number of ways; integer linear programming or backtracking search are the most common.

Winner-determination problem

- Problem: The size of the WDP is exponential in the number of items that can be sold.
  - Every possible bundle must be considered.
- Formulating the problem as ILP helps some
  - This problem has been studied since the 50s, so good heuristic techniques exist.

Winner-determination problem

- Other solutions:
  - Limiting the sorts of bundles allowed.
  - OR bids and XOR bids.
    - This transforms the problem into the knapsack problem.
    - Still NP-hard, but good heuristics exist
- Limiting size of bundles.
- Approximation algorithms

Current research iss

- Auctions are a particularly hot area of research.
- Topics include:
  - Information revelation - how can we preserve the truth-telling strategy of Vickrey without agents revealing their preferences to each other?
  - Winner determination.
  - Languages for expressing more complex constraints.
  - Preventing collusion and false-name bids.
  - ’online’ auctions
    - Not “on the Internet” - meaning agents continuously arrive and leave.

Summary

- Vickrey auctions are particularly appealing from a computational standpoint.
  - Easy for participants to decide how to act.
  - Hard to manipulate.
- Resources always allocated to the agent that values them the most.
- Challenges:
  - Dealing with imperfect information
  - Combinatorial auctions run us up against NP-completeness (again).