Mechanism Design

1. [Sequence announcement] Introduction to Mechanism Design
2. Mechanism Design: Constructing Algorithms for Strategic Agents
3. Incentive compatibility and the Revelation Principle
4. Strategyproof Mechanisms: Impossibilities
5. Strategyproof Mechanisms: Possibilities

[Sequence announcement] Introduction to Mechanism Design

Mechanism design is the theory of how to construct institutions for strategic agents, spanning applications like voting systems, school admissions, regulation of monopolists, and auction design. Think of it as the engineering side of game theory, building algorithms for strategic agents. While it doesn't have much to say about rationality directly, mechanism design provides tools and results for anyone interested in world optimization.

In this sequence, I'll touch on

- The basic mechanism design framework, including the revelation principle and incentive compatibility.
- The Gibbard-Satterthwaite impossibility theorem for strategyproof implementation (a close analogue of Arrow's Theorem), and restricted domains like single-peaked or quasilinear preference where we do have positive results.
- The power and limitations of Vickrey-Clarke-Groves mechanisms for efficiently allocating goods, generalizing Vickrey's second-price auction.
- Characterizations of incentive-compatible mechanisms and the revenue equivalence theorem.
- Profit-maximizing auctions.
- The Myerson-Satterthwaite impossibility for bilateral trade.
- Two-sided matching markets à la Gale and Shapley, school choice, and kidney exchange.

As the list above suggests, this sequence is going to be semi-technical, but my foremost goal is to convey the intuition behind these results. Since mechanism design builds on game theory, take a look at Yvain's Game Theory Intro if you want to brush up.

Various resources:

- For further introduction, you can start with the popular or more scholarly survey of mechanism design from the 2007 Nobel memoriam prize in economics.
- Jeff Ely has lecture notes and short videos to accompany an undergraduate class in microeconomic theory from the perspective of mechanism design.
- The textbook A Toolbox for Economic Design by Dimitrios Diamantaras is very accessible and comprehensive if you can get ahold of a copy.
- Tilman Börgers has a draft textbook intended for graduate students.
- Chapters 9-16 of Algorithmic Game Theory and chapters 10-11 of Multiagent Systems cover various topics in mechanism design from the perspective of computer scientists.
- Video lectures introducing market design and computational aspects of mechanism design.

I plan on following up on this sequence with another focusing on group rationality and information aggregation, surveying scoring rules and prediction markets among other topics.

Suggestions and comments are very welcome.

Mechanism Design: Constructing Algorithms for Strategic Agents

> *tl;dr Mechanism design studies how to design incentives for fun and profit. A puzzle about whether or not to paint a room is posed. A modeling framework is introduced, with lots of corresponding notation.*

Mechanism design is a framework for constructing institutions for group interactions, giving us a language for the design of everything from voting systems to school admissions to auctions to crowdsourcing. Think of it as the engineering side of game theory, building algorithms for strategic agents. In game theory, the primary goal is to answer the question, "Given agents who can take some actions that will lead to some payoffs, what do we expect to happen when the agents strategically interact?" In other words, game theory describes the outcomes of fixed scenarios. In contrast, mechanism design flips the question around and asks, "Given some goals, what payoffs should agents be assigned for the right outcome to occur when agents strategically interact?" The rules of the game are ours to choose, and, within some design constraints, we want to find the best possible ones for a situation.

Although many people, even high-profile theorists, doubt the usefulness of game theory, its application in mechanism design is one of the major success stories of modern economics. Spectrum license auctions designed by economists paved the way for modern cell-phone networks and garnered billions in revenue for the US and European governments. Tech companies like Google and Microsoft employ theorists to improve advertising auctions. Economists like Al Roth and computer scientists like Tuomas Sandholm have been instrumental in establishing kidney exchanges to facilitate organ transplants, while others have been active in the redesign of public school admissions in Boston, Chicago, and New Orleans.

The objective of this post is to introduce all the pieces of a mechanism design problem, providing the setup for actual conclusions later on. I assume you have some basic familiarity with game theory, at the level of understanding the concept of a dominant strategies and Nash equilbria. Take a look at Yvain's Game Theory Intro if you'd like to brush up.

Overly optimizing whether or not to paint an room

Let's start with a concrete example of a group choice: Jack, an economics student, and Jill, a computer science student, are housemates. To procrastinate studying for finals, they are considering whether to repaint their living room. Conveniently, they agree on what color they would choose, but are unsure whether it's worth doing at all. Neither Jack nor Jill would pay the known fixed cost of $300 entirely on their own. They're not even sure the cost is less than their joint value[1], so it's not only a matter of bargaining to split the cost (a non-trivial question on its own). The decision to paint the room depends on information neither person fully knows, since each knows their own value, but not the value to the other person.

The lack of complete information is what makes the problem interesting. If the total value is obviously greater than $300, forcing them to paint the room and split the cost evenly would be utility-maximizing[2]. One person might be worse off, but the other would be correspondingly better off. This solution corresponds to funding a public good (in the technical sense of something non-excludable and non-rivalrous) through taxation. Alternatively, if the total value is obviously less than $300, then the project shouldn't be done, and the question of how to split the cost becomes moot. With some overall uncertainty, we now have to worry that either housemate might misrepresent their value to get a better deal, causing the room to be painted when it shouldn't be or vice versa.

Assuming the two want to repaint the room if and only if their total value is greater than $300, how would you advise they decide whether to do the project and how much each should contribute?

Pause a moment to ponder this puzzle…

.

.

.

Some naive solutions would be to:

- Vote on painting the room, and if both say yes, do the project with each contributing $150.
- Vote, and if either one says yes, do the project. If both say yes, both contribute $150. If only one says yes, that person contributes $225 and the other contributes $75.
- Both simultaneously write down a number. If the total is greater than $300, do the project. Each contributes a share of the $300 proportional their number.
- Flip a coin to decide who makes an initial offer of how to split the cost, i.e. Jack paying $50 and Jill paying $250. The other person can accept the proposal, in which case they do the project with those contributions. Otherwise, that person makes a new proposal. Alternating proposals continue until one is accepted. If 100 rounds pass without an accepted proposal, don't paint the room.

None of these will guarantee the room is painted exactly when it's worth the cost. The first procedure never paints the room when it shouldn't be, but sometimes fails to paint when it would be worthwhile, like when Jack values the renovation at $120 and Jill values it at $200. Jack would vote "no", even though their total value is $320. The second procedure can make mistakes of both kinds and can also result in someone contributing more than their value. The other two are more difficult to analyze, but still turn out to be non-optimal.

These protocols barely scratch the surface of all the possible institutions out there. Maybe we just need to be a little more creative to find something better. To definitively solve this problem, some formalism is in order.

Framework for institutional design

> *Trigger warning: Greek letters, subscripts, sets, and functions.*

Getting a little more abstract, let's specify all the relevant elements in an institutional design setting. First of all, the agents involved in our process need to be identified. Typically, this doesn't need to go beyond labeling each agent. For instance, we might assign each a number from 1 to n, with i representing a generic agent. Above, we have two agents named Jack and Jill.

> *Notes on notation:* A generic agent has the label i. A set or variable Z_i belongs to agent i. Typically, variables are lowercase, and the set a variable lives in is uppercase. Without a subscript, Z refers to the vector $Z = (Z_1, \ldots, Z_i, \ldots, Z_N)$ with one element for each agent. It's often useful to talk about the vector $Z_{-i} = (Z_1, \ldots, Z_{i-1}, Z_{i+1}, \ldots, Z_N)$ for all agents *except* agent i (think of deleting Z_i from the vector Z). This enables us to write Z as (Z_i, Z_{-i}), highlighting agent i's part in the profile.

Once we've established who's involved, the next step is determining the relevant traits of agents that aren't obviously known. This unknown data is summarized as that agent's *type*. Types can describe an agent's preferences, their capabilities, their beliefs about the state of the world, their beliefs about others' types, their beliefs about others' beliefs, etc. The set of possible types for each agent is their *type space*. A typical notation for the type of agent i is θ_i, an element of the type space Θ_i. If an assumption about an agent's preferences or beliefs seems unreasonable, that's a sign we should enlarge the set of possible types, allowing for more variety in behavior. In our housemate scenario, the type of each agent needs to – at a bare minimum – specify the value each puts on having new paint. If knowing that value fully specifies the person's preferences and beliefs, we're done. Otherwise, we might need to stick more information inside the person's type. Of course, there is a trade-off between realism and tractibility that guides the modeling choice of how to specify types.

Next, we need to consider all possible outcomes that might result from the group interaction. This could be an overarching outcome, like having one candidate elected to office, or a specification of the sub-outcomes for each agent, like the job each one is assigned. Let's call the set of all outcomes X. In the housemate scenario, the outcomes consist of the binary choice of whether the room is painted and the payment each person makes. Throwing some notation on this, each outcome is a triple $(q, P_{\text{Jack}}, P_{\text{Jill}}) \in X = \{0, 1\} \times \mathrm{R} \times \mathrm{R}$.

To talk about the incentives of agents, their preferences over each outcome must be specified as a function of their type. In general, preferences could be any ranking of the outcomes, but we often assume a particular utility function. For instance, we might numerically represent the preferences of Jack or Jill as $u_i(q, P_i, \theta_i) = q\theta_i - P_i$, meaning each gets benefit θ_i if and only if the room is painted, minus their payment, and with no direct preference over the payment of the other person.

After establishing what an agent wants, we need to describe what an agent believes, again conditional on their type. Usually, this is done by assuming agents are Bayesians with a common prior over the state of the world and the types of others, who then update their beliefs after learning their type[3]. For instance, Jack and Jill might both think the value of the other person is distributed uniformly between $0 and $300, independently of their own type.

Based on the agents' preferences and beliefs, we now need to have some theory of how they choose actions. One standard assumption is that everyone is an expected utility maximizer who takes actions in Nash equilibrium with everyone else.

Alternatively, we might consider agents who reason based on the worst case actions of everyone else, who minimize maximum regret, who are boundedly rational, who are willing to tell the truth as long as they only lose a small amount of utility, or who can only be trusted to play dominant strategies rather than Nash equilibrium strategies, etc. What's impossible under one behavioral theory or solution concept can be possible under another.

In summary so far, a design setting consists of:

1. The agents involved.
2. The potential types of each agent, representing all relevant private information the agent has.
3. The potential outcomes available.
4. The agents' preferences over each outcome for each type, possibly expressed as a utility function.
5. The beliefs of each agent as a function of their type.
6. A theory about the behavior of agents.

In our housemate scenario, these could be modeled as following:

1. **Agents involved:** Two people, Jack and Jill.
2. **Potential types:** The maximum dollar amounts, θ_{Jack} and θ_{Jill}, each would be willing to contribute, contained in the type spaces $\Theta_{\text{Jack}} = \Theta_{\text{Jill}} = [0, 300]$.
3. **Potential outcomes:** A binary decision variable $q = 1$ if the room is repainted and $q = 0$ if not, as well as the amounts paid p_{Jack} and p_{Jill}, contained in the outcome space $X = \{0, 1\} \times \mathrm{R} \times \mathrm{R}$.
4. **Preferences over each outcome for each type:** A numerical representation of how much each agent likes each outcome $u_i(q, p_i, \theta_i) = q\theta_i - p_i$.
5. **Beliefs:** Independently of their own valuation, each thinks the valuation of the other is uniformly distributed between $0 and $300.
6. **Behavioral theory:** Each housemate is an expected utility maximizer, and we expect them to play strategies in Nash equilibrium with each other.

Given a design setting, we presumably have some goals about what should happen, once again conditional on the types of each agent. In particular, we might want specific outcomes to occur for each profile of types. A goal like this is called a *social choice function*[4], specifying a mapping from profiles of agent types to outcomes f: $\prod_i \Theta_i \to X$. A social choice function f says, "If the types of agents are θ_1 through θ_N, the outcome $f(\theta_1, \ldots, \theta_N) \in X$ should happen". A social choice fucntion could be defined indirectly as whatever outcome maximizes some objective subject to design constraints. For instance, we could find the social choice function that maximizes agents' utility or the one that maximizes the total payments collected from the agents in an auction, conditional on no agent being worse off for participating.

Putting the mechanism in mechanism design

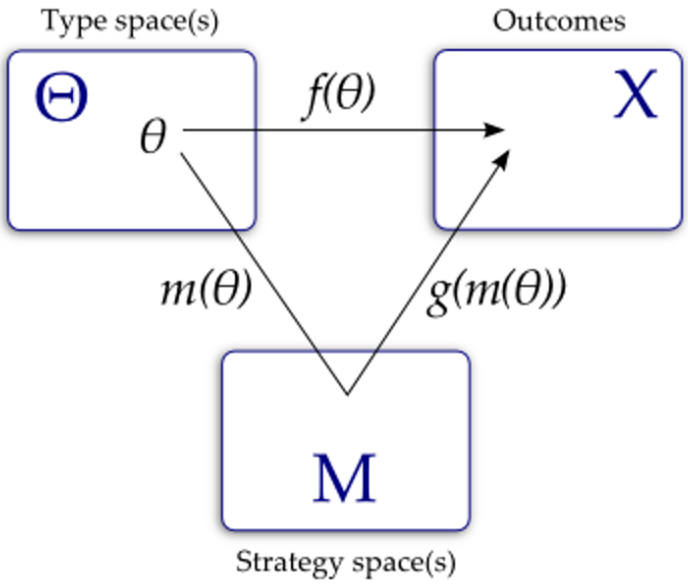
So far, this has been an exercise in precisely specifying the setting we're working in. With all this in hand, we now want to create a protocol/game/institution where agents will interact to produce outcomes according to our favorite social choice function. We'll formalize any possible institution as a *mechanism* (M, g) consisting of a set of messages M_i for each agent and an outcome function g: $\prod_i M_i \to X$ that assigns an outcome for each profile of messages received from agents. The messages could be

very simple, like a binary vote, or very complex, like the source code of a program. We can force any set of rules into this formalism by having agents submit programs to act as a their proxy. If we wanted the housemates to bargain back and forth about how to split the cost, their messages could be parameters for a pre-written bargaining program or full programs that make initial and subsequent offers, depending on how much flexibility we allow the agents. Messages represent strategies we're making available to the agent, which are then translated into outcomes by g.

When agents interact together in the mechanism, each chooses the message $m_i(\theta_i)$ they'll send as a function of their type, which then results in the overall outcome $g(m_1(\theta_1), \ldots, m_n(\theta_n))$. The mechanism (M, g) *implements* a social choice function f if, for all profiles of types θ, the outcome we get under the mechanism equals the outcome we want, i.e.

$$g(m_1(\theta_1), \ldots, m_n(\theta_n)) = f(\theta_1, \ldots, \theta_n), \text{ for all profiles } (\theta_1, \ldots, \theta_n) \in \Theta = \prod_i \Theta_i$$

In other words, we want the strategies (determined by whatever behavioral theory we have for each agent) to compose with the outcome function (which we are free to choose, up to design constraints) to match up with our goal, as shown in the following diagram:



A social choice function f is *implementable* if some mechanism exists that implements it. Whether a social choice function is implementable depends on our behavioral theory. If we think agents choose strategies in Nash equilibrium with each other, we'll have more flexiblity in finding a mechanism than if agents need the stronger incentive of a dominant strategy, since more Nash equilibria exist than dominant-strategy equilibria. Rather than assuming agents choose strategically based on their preferences, perhaps we think agents are naively honest (maybe because they are computer programs we've programmed ourselves). In this case, we can trivially implement a social choice rule by having each agent tell us their full type and simply choosing the corresponding goal by picking $M_i = \Theta_i$ and $g = f$. Here the interesting question is instead which mechanism can implement f with the minimal amount of

communication, either by minimizing the number of dimensions or bits in each message. It's also worth asking whether social choice rules satisfying certain properties can exist at all (much less whether we can implement them), along the lines of Arrow's Impossibility Theorem.

Wrapping up

In summary, agents have *types* that describe all their relevant information unknown to the mechanism designer. Once we have a *social choice function* describing a goal of which outcomes should occur for each realization of types, we can build a *mechanism* consisting of sets of messages or strategies for each agent and a function that assigns outcomes based on the messages received. The hope is that the outcome realized by the agents' choice of messages based on their type matches up with the intended outcome. If so, that mechanism *implements* the social choice function.

How can we actually determine whether a social choice function is implementable though? If we can find a mechanism that implements it, we've answered our question. In the reverse though, how would we go about showing that no such mechanism exists? We're back to the problem of searching over all possible ways the agents could interact and hoping we're creative enough.

In the next post, I'll discuss the Revelation Principle, which allows us to cut through all this complexity via *incentive compatibility*, along with a solution to the painting puzzle.

Next up: Incentive compatibility and the Revelation Principle

---

1. To clarify if necessary, Jack's value for the project is the amount of money that he'd just barely be willing to spend on the project. If his value is $200, then he would be willing to pay $199 since that leaves a dollar of value left over as economic surplus, but he wouldn't pay $201 dollars. If the cost was $200, identical to his value, then he's indifferent between making the purchase and not. The joint value of Jack and Jill is just the sum of their individual valuations.↩

2. Assuming dollars map roughly equally onto utilities for each. In general, maximizing total willingness-to-pay is known as Kaldor-Hicks efficiency.↩

3. This idea is originally due to John Harsanyi. As long as the type spaces are big enough, this can be done without loss of generality as formalized by Mertens and Zamir (1985).↩

4. If multiple outcomes are acceptable for individual profiles, we have a *social choice correspondence*.↩

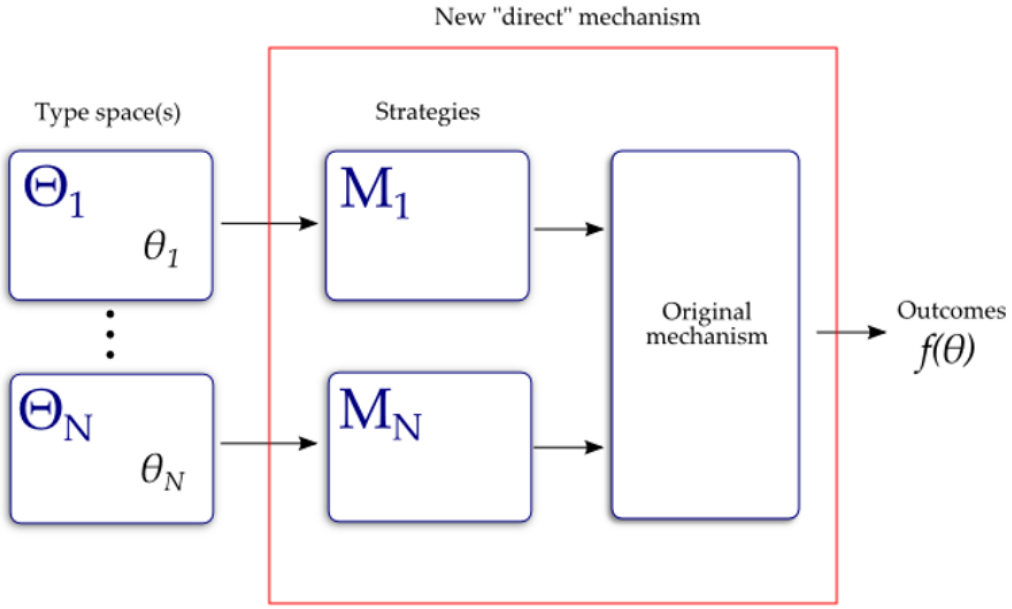
Incentive compatibility and the Revelation Principle

> *In which the Revelation Principle is introduced, showing all mechanisms can be reduced to incentive compatible mechanisms. With this insight, a solution (of sorts) is given to the public good problem in the last post. Limitations of the Revelation Principle are also discussed.*

The formalism I introduced last time will now start paying off. We were left with the question of how to check whether a mechanism exists that satisfies some particular goal, naively requiring us to search over all possible procedures. Luckily though, the space of all possible mechanisms can be reduced to something manageable.

Observe the following: suppose through divine intervention we were granted a mechanism (M, g) that implements a social choice function f. In other words, the outcome when agents of types $\theta_1, ..., \theta_n$ interact with the mechanism is exactly the outcome prescribed by the function f for those types. Since a person's type encodes all relevant variation in their characteristics, preferences, or beliefs, we'd know what that person wants to do if we knew their type. In particular, when agent i is type θ_i, we expect her choice of message to be $m_i(\theta_i)$. Once all the messages are sent, the function g translates the agents' choices into an outcome so that $g(m_1(\theta_1), ..., m_n(\theta_n)) = f(\theta_1, ..., \theta_n)$.

But wait! Since we expect a type θ_i agent to send message $m_i(\theta_i)$, why don't we just package that inside the mechanism? Each agent will tell us their type and the mechanism designer will play in proxy for the agent according to the original mechanism. We'll call this the *direct mechanism* for f since all we need to know is that agents tell us their types and then are assigned outcome $f(\theta)$, no matter what we've blackboxed in the middle.



Why did we expect an agent of type θ_i to send message $m_i(\theta_i)$? Presumably because that message maximized her utility. In particular, it had to be at least as good as the message $m_i(\theta'_i)$ she's send when her type is different, giving us:

$$u_i(g(m_i(\theta_i), m_{-i}), \theta_i) \geq u_i(g(m'_i, m_{-i}), \theta_i), \quad \forall m'_i \in M_i \quad \Rightarrow$$
$$u_i(g(m_i(\theta_i), m_{-i}), \theta_i) \geq u_i(g(m_i(\theta'_i), m_{-i}), \theta_i), \quad \forall \theta'_i \in \Theta_i$$

Since the outcomes of the mechanism coincide with *f*, we can conclude

Even though we started off with some mechanism, this last statement doesn't say anything about the mechanism itself, only the social choice function it implements. Let's call any social choice function *f* that satisfies this constraint for every agent and for all possible types of others *strategy-proof* or *dominant-strategy incentive compatible (DSIC)*. Note that this can add up to a lot of constraints.

These observations lead us to the following powerful tool:

> **The Revelation Principle (for dominant strategies):** A social choice function *f* is implementable in dominant strategies if and only if it is dominant-strategy incentive compatible. Alternatively, every mechanism that implements *f* in dominant strategies is equivalent to a DSIC direct mechanism.

Rather than requiring an agent to always (weakly) prefer their type's outcome, it might suffice that agents get greater utility on average for their true type:

$$\mathrm{E}_{\theta_{-i}}\left[\, u_i(f(\theta_i, \theta_{-i}), \theta_i) \mid \theta_i \,\right] \geq \mathrm{E}_{\theta_{-i}}\left[\, u_i(f(\theta_i', \theta_{-i}), \theta_i) \mid \theta_i \,\right], \quad \forall \theta_i, \theta_i' \in \Theta_i$$

Social choice functions *f* that satisfy this constraint for each agent are *Bayesian incentive compatible (BIC)*. Just as we had a Revelation Principle for dominant strategies, we have an analogous one for Bayes-Nash equilibrium:

> **The Revelation Principle (for Bayes-Nash equilibrium):** A social choice function *f* is implementable in Bayes-Nash equilibrium if and only if it is Bayes-Nash incentive compatible.

As a check whether you're following, consider these questions:

1. Suppose you have a social choice function *f* that is (dominant-strategy or Bayesian) incentive compatible for the type spaces $\Theta_1, \ldots, \Theta_n$. Now, throw out some types for each agent. Is the restriction of *f* to these smaller type spaces still incentive compatible?
2. Is it easier for a social choice function to be DSIC or BIC?

Answers in the footnotes.[1]

With the reduction provided by the various versions of the Revelation Principle, we can get away with analyzing only incentive compatible direct mechanisms rather than all possible ones. If a social choice function isn't incentive compatible, then it can't be implemented, no matter how fancy we get.

There are many reasons why we wouldn't want to use a direct mechanism in practice: full types might be too complex to communicate easily, agents might be worried about privacy, or agents might not trust the mechanism operator to use the information like promised (changing the rules of the game after the fact, despite a claimed commitment to a particular outcome)[2], [3]. Still, direct mechanisms are very straightforward for people to use. Rather than investing time to figure out how to play the game, the participants only have to consider what their true preferences are.

In any case, direct mechanisms are very handy theoretically. From this perspective, all incentive problems boil down to encouraging agents to be honest about their private information. The only leeway we have is whether being honest should be a dominant strategy, a best response to the honesty of others, minimize the agent's maximum regret, etc.

Returning to the painting puzzle

In our housemate story, we don't need to consider all possible mechanisms; we only need to consider procedures where they write down their valuation and both have an incentive to be honest about their true preferences. This is still a big space of possible mechanisms, but is much more manageable. Think for a moment about what direct mechanism you'd recommend to the housemates.

Here is one procedure that always makes the efficient decision, though it's less than ideal: Collect valuations simultaneously. If the sum of the two reports $\theta_{Jack} + \theta_{Jill}$ is at least \$300, the room will be painted. Jack's payment will be $p_{Jack} = 300 - \theta_{Jill}$ if the room is painted and zero otherwise. Similarly, Jill's payment will be $p_{Jill} = 300 - \theta_{Jack}$ when the room is painted and zero otherwise.

Take a moment to convince yourself that honesty is (weakly) dominant for each person under this mechanism. For instance, suppose Jack's value is \$120. What would happen if he reports above this, say at \$150? If Jill reports something less than \$150, then the room isn't painted and Jack's utility is zero, the same as when he is honest. If Jill reports between \$150 and \$180, the room is painted and Jack gets a payoff of $120 - (300 - \theta_{Jill}) = \theta_{Jill} - 180 < 0$, less than his payoff from honesty where the room isn't painted. If Jill reports above \$180, the room is painted whether or not Jack reports honestly at \$120 or dishonestly at \$150, so the payoff is the same. Similar reasoning goes through for any report Jack considers below \$120.

So what's the issue here? Look at the sum of the payments. When the room is painted, Jack and Jill have a \$300 bill, but the mechanism only collects $300 - \theta_{Jack} + 300 - \theta_{Jill} < 300$. While the mechanism is efficient, it assumes extra money is coming from somewhere! If the individual values could be anywhere between \$0 and \$300, then the deficit could be up to \$300.

In an attempt to solve this budget issue, we could tweak the payments to the following: $p_{Jack} = 450 - \theta_{Jill}$ and $p_{Jill} = 450 - \theta_{Jack}$ when the room is painted and both 150 when it isn't. Honest reporting is still a dominant strategy and there is never a budget shortfall, but now the two are paying out \$300 whether or not the room is painted. Hard to imagine them agreeing to use a procedure that could leave them worse off than never having considered the renovation.

Is there a mechanism that makes the efficient decision but avoids both these pitfalls, never forcing someone to pay more than their valuation while still covering the bill? Alas, this turns out to be impossible.

Second-best mechanisms and the Myerson-Satterthwaite impossibility

Myerson and Satterthwaite (1983) prove no possible mechanism exists that paints the room exactly when efficient without requiring an outside subsidy or making one agent worse off for having participated. The original paper was framed in terms of a single buyer and seller considering whether to trade an item, showing perfectly efficient trade is impossible in the presence of incomplete information when types are independently distributed and trade isn't a foregone conclusion[4].

Rather than looking for the "first-best" mechanism that always makes the efficient decision, we're going to be forced to find a "second-best" mechanism that maximizes welfare while still satisfying our constraints. Surprisingly, the naive "vote and split the cost" mechanism is the best feasible procedure in dominant strategies. Restated in direct mechanism terms, if both submit a valuation greater than \$150, the room is painted and they split the cost equally. Jeff Ely provides a very nice graphical proof of this fact here.

Honesty as a dominant strategy is a compelling feature of a mechanism—agents don't have to put any thought into what others might do. Requiring honesty to be dominant might be overly strict though. Not all games have dominant strategies. On the other hand, we expect all (well-behaved) games to have a Nash equilibrium. If we weaken dominant-strategy IC to Bayes-Nash IC, we can do slightly better, but full efficiency is still impossible. The best feasible direct mechanism—assuming values are uniform between \$0 and \$300—is to paint the room if and only if the values sum to more than \$375, with each paying \$150 and the person with the high valuation giving the other one-third of the difference in their valuations, i.e.

$$p_{Jack} = \begin{cases} 150 + (\theta_{Jack} - \theta_{Jill})/3, & \text{if } \theta_{Jack} + \theta_{Jill} \geq 375 \\ 0, & \text{if } \theta_{Jack} + \theta_{Jill} < 375 \end{cases}$$

and similarly for Jill. This direct mechanism corresponds to a Bayes-Nash equilibrium of the following, more intuitive mechanism: each writes down a bid and the room is painted if the bids sum to more than $300, with excess total payments over $300 split equally between them (so if Jack bids $200 and Jill bids $150, Jack pays $175 and Jill pays $125).

Conclusion

By reducing the design problem to encouraging the roommates to be honest via the Revelation Principle, we can identify the best feasible mechanism and end up uncovering a general impossibility about bargaining along the way. In the next post, I'll delve further into what is possible under dominant-strategy incentive compatibility.

Previously on: Mechanism Design: Constructing Algorithms for Strategic Agents

Next up: Strategyproof Mechanisms: Impossibilities

---

1. *Question 1:* Yes, it is still incentive compatible since the constraint still holds for all types kept, with the constraints involving all types thrown out being no longer relevant. *Question 2:* Bayesian incentive compatibility is the weaker condition, implied by dominant-strategy incentive compatibility, since if the constraint holds for every possible realization, then it must hold on average for any distribution across possible types.↩

2. Although cryptography could help solve the privacy and commitment problems of direct mechanisms. See Naor et al 1999, "Privacy preserving auctions and mechanism design".↩

3. I've also glossed over the issue of *full* vs *partial* implementation. The revelation principle guarantees some equilibrium of the direct mechanism coincides with our social choice function. However, there might be other "bad" equilibria in the direct mechanism, whil an indirect mechanism might have a unique good equilibrium.↩

4. Luckily, the welfare loss disappears quickly as the size of the market grows. Once there are at least six people on each side of the market, the overall welfare loss due to incomplete info is less than 1% (Satterthwaite et al 2014, "Optimality vs Practicality in Market Design)↩

Strategyproof Mechanisms: Impossibilities

> *In which the limits of dominant-strategy implementation are explored. The Gibbard-Satterthwaite dictatorship theorem for unrestricted preference domains is stated, showing no universal strategyproof mechanisms exist, along with a proof for a special case.*

Due to the Revelation Principle, most design questions can be answered by studying incentive compatible mechanisms, as discussed in the last post. Incentive compatibility comes in many different flavors corresponding to different solution concepts—dominant-strategy IC and Bayes-Nash IC being the two most common. In this post, I'll delve into what's possible under dominant strategy incentive compatibility.

Recall that a strategy is *dominant* if playing it always leads to (weakly) higher payoffs for an agent than other strategy would, no matter what strategies other agents play. A social choice function is *dominant-strategy incentive compatible* if honest revelation is a dominant strategy in the direct mechanism for that SCF[1]. The appeal of implementation in dominant strategies is that an agent doesn't need to think about what other agents will do. Even in the worst case, a dominant-strategy IC social choice function leaves an agent better off for being honest. Since the need for strategic thinking is eliminated, dominant-strategy IC is also referred to as *strategyproofness*.

Gibbard-Satterthwaite: universal strategyproof mechanisms are out of reach

Arrow's theorem is well-known for showing dictatorships are the only aggregators of ordinal rankings that satisfy a set of particular criteria. The result is commonly interpreted as saying there is no perfect voting system for more than two candidates. However, since Arrow deals with *social welfare functions* which take a profile of preferences as input and outputs a full preference ranking, it really says something about aggregating a set of preferences into a single group preference. Most of the time though, a full ranking of candidates will be superfluous—all we really need to know is who wins the election. Although Arrow doesn't give social welfare functions much room to maneuver, maybe there are still some nice social choice functions out there.

Alas, it's not to be. Alan Gibbard and Mark Satterthwaite have shown that, in general, the only strategyproof choice from three or more alternatives that is even slightly responsive to preferences is a dictatorship by one agent.

Stated formally:

> **Gibbard-Satterthwaite dictatorship theorem:** Suppose $f: L(A)^n \to A$ is a social choice function for n agents from profiles of ordinal rankings $L(A)^n$ to a set of outcomes A with at least three elements. If f is strategyproof and onto[2], then f is dictatorial.

Being strategyproof seems intuitively more desirable to me than the properties in Arrow's theorem (especially the much critiqued *independence of irrelevant alternatives* criterion). As it turns out though, Gibbard-Satterthwaite and Arrow are equivalent! Weakening our ambition from social welfare functions to social choice functions didn't give us anything.

Gibbard-Satterthwaite seems a great blow to mechanism design at first glance. On reflection, perhaps we were asking for too much. A completely generic strategyproof mechanism that can be used in any situation does sound too good to be true.

There are three ways to proceed from this point:

1. Building strategyproof mechanisms for specialized applications,
2. Weakening strategyproofness to another form of incentive compatibility, or
3. Stepping sideways from ordinal preferences and enriching the model while adding further assumptions.

On that last path, note that switching from ordinal to cardinal preferences can't save us unless we offset that generalization with other assumptions—cardinal information expands the ordinal type space, which can only make implementation more difficult.

Since dictatorships are the only universal strategyproof mechanisms for choosing between three options, it's worth investigating why life is pleasant with only two options. In this case, lots of non-trivial strategyproof mechanisms exist. For example, suppose a committee is restricted to painting a bike shed either red or blue due to zoning restrictions. The social choice function that paints the shed red if and only if a majority have red as their first choice is strategyproof, as is the social choice to paint the shed red if and only if the committee unanimously lists red as their top choice.

Of course, not every voting rule for binary outcomes will be strategyproof, like the rule that paints the shed red if and only if an odd number of committee members top-rank red. While I can't imagine anyone arguing in favor of this rule, what's going wrong here? The problem is this odd rule isn't *monotonic*—raising red in someone's preference ranking can cause the outcome to switch away from red.

Here are the formal definitions of strategyproofness and monotonicity:

A social choice function $f: L(A)^n \to A$ is *strategyproof* if for each agent i, the social choice satisfies

$$f(\succ_i, \succ_{-i}) \succeq_i f(\succ'_i, \succ_{-i})$$

for all rankings

$$\succ_i, \succ'_i \in L(A)$$

and

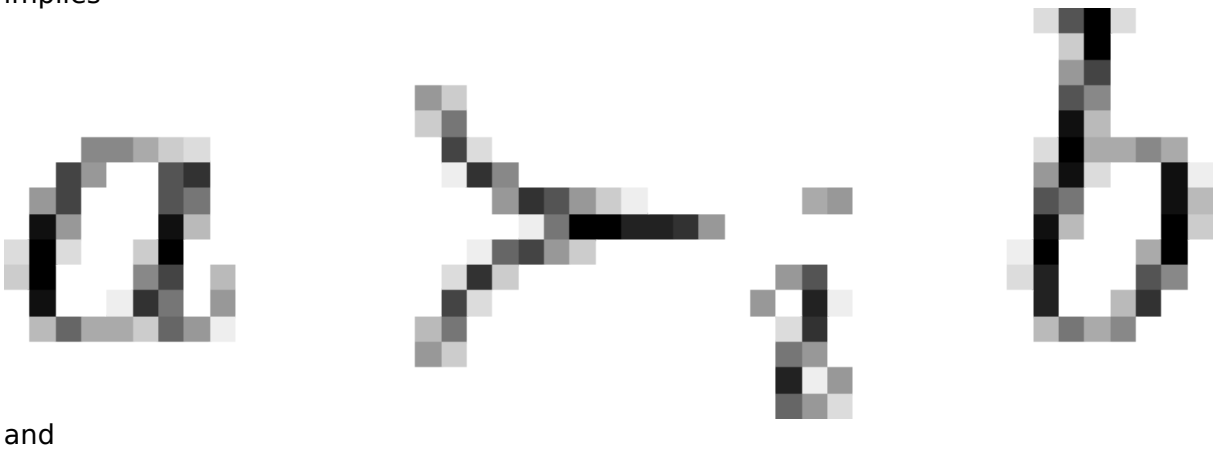
$$\succ_{-i} \in L(A)^{n-1}$$

.

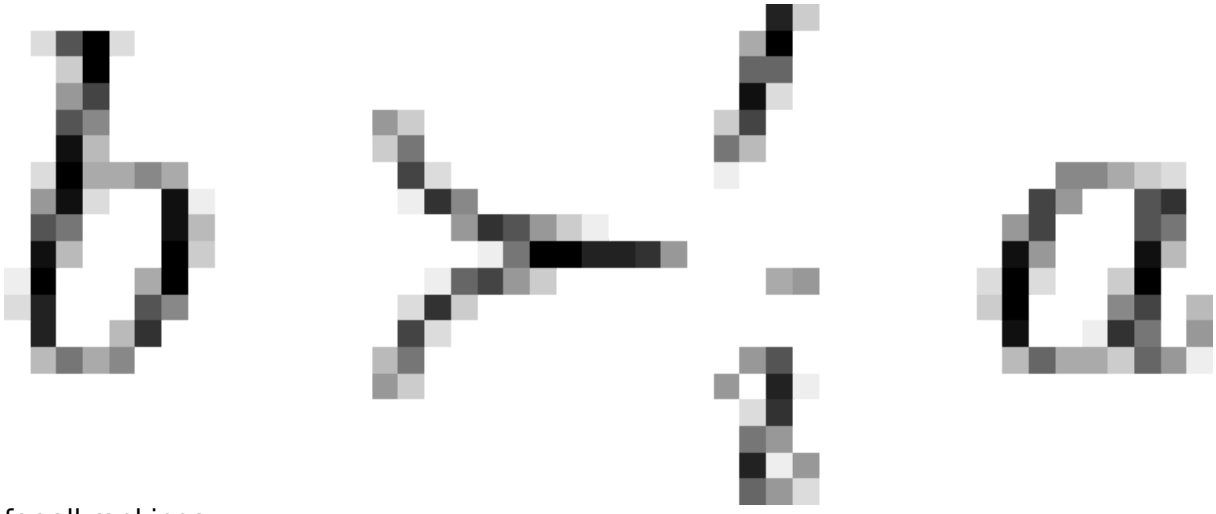
A social choice function $f: L(A)^n \to A$ is *monotonic* if for each agent i, we have that

$$f(\succ_i, \succ_{-i}) = a \neq b = f(\succ'_i, \succ_{-i})$$

implies



and



for all rankings

$$\succ_i, \succ'_i \in L(A)$$

and

$$\succ_{-i} \in L(A)^{n-1}$$

.

Take a moment to convince yourself that monotonicity is just a slight rephrasing of strategyproofness and hence equivalent. Though they are identical at heart,

monotonicity carries two useful interpretation as an invariance property. First, if an agent submits a new ranking where the previous outcome goes up, the outcome can't change. Second, submitting a new ranking where the rank of the previous outcome relative any other option is unchanged has to leave the outcome unchanged as well.

When there are only two alternatives, we can think of the implicit outcome space as one dimensional, with one outcome on the left and the other on the right. Going towards one outcome corresponds exactly with going away from the other. With three or more alternatives, we don't have the same nice structure, leading to the impossibility of a non-trivial monotonic rule.

In the next post, I'll describe domains where we can enrich the outcome space beyond a binary choice and still get strategyproofness. Since the outcome space won't naturally have a nice order structure, we'll have to ensure it does by restricting the preferences agents can have over it. Even though we don't have universal strategyproof mechanisms other than dictatorships, we can uncover strategyproof mechanisms for specific applications. In the meantime, here's a proof of Gibbard-Satterthwaite for the curious.

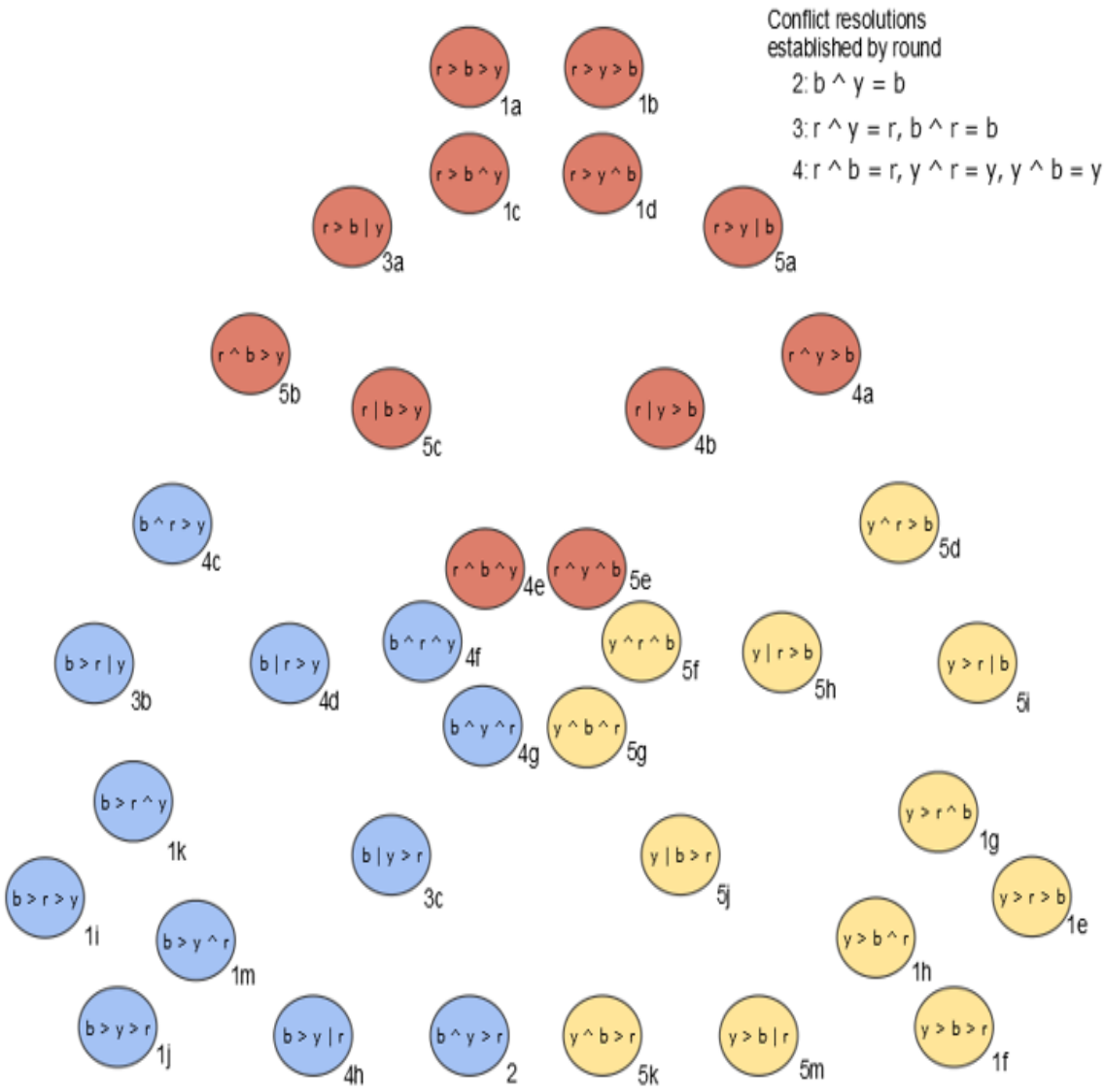
Proof of Gibbard-Satterthwaite (in a special case)

Suppose the committee consists of two people, Betty and Bill. In addition to red and blue, the city recently approved yellow as an option to paint bike sheds. Each person has six possible rankings over the three colors, and there are 36 preference profiles containing one ranking from each. The 36 will fall into six relevant classes. Here is an example of each class along with some shorthand to describe it:

1. $r > b > y$: Both agents agree on the ranking of red over blue over yellow.
2. $r > b \hat{} y$: Both agents agree red is the best. Betty puts blue as her second choice, while Bill has yellow as his second choice.
3. $b \hat{} y > r$: Both agree red is worst. Betty thinks blue is better than yellow, while Bill thinks yellow is better.
4. $r \mid (b > y)$: Both agents agree blue is better than yellow. Betty thinks red is better than them both, while Bill thinks red is worse than both.
5. $(b > y) \mid r$: Both agree blue is better than yellow. Betty thinks red is worst, while Bill thinks red is best.
6. $r \hat{} b \hat{} y$: Betty has the ranking red over blue over yellow, while Bill has the reverse.

For the notation summarizing the profile, $r > b$ indicated both agree that red is better than blue. $r \hat{} b$ says there is a conflict between the two with Betty preferring red. $r \mid (b > y)$ says there are two preference conflicts, with Betty preferring red over the other two options, so we alternatively think of this profile as $r \hat{} y$, $r \hat{} b$, and $y > b$.

Now, we'll assign the 36 profiles a color, using each at least once, in a way that is monotonic. This will happen in six steps as depicted in the following diagram.



1. At least one profile must be red by assumption. Starting from that profile (whatever it is), we could move red to the top of both rankings and the outcome would still be red by monotonicity. Then all other profiles with red top-ranked by both also must be red since any swaps of blue and yellow can't change the outcome since red is still relatively above both. This gives us 1a,b,c,d. Blue and yellow go through similarly.
2. Consider the profile *b^y > r* at the bottom of the diagram. The profile *y > b > r* got yellow in 1f, so if Betty starts liking blue better from 2, the outcome has to stay yellow or switch to blue. Since monotonicity can't tell us more than that, we have to make a choice. Let's decide in favor of Betty and pick blue.
3. Since we chose to resolve the conflict *b^y* as *b*, three more colorings follow since we must resolve this particular conflict in the same way everywhere. Keep in mind that *b^y* is a different conflict than *y^b* since it might matter who prefers which color. Consider 3b. This can't be yellow since yellow increases between this profile and 2, but 2 isn't yellow. It also can't be red since 1k isn't red, so we conclude 3b must be blue. Now consider 3a. Even though the conflict *b^y* resolves in favor of

blue, the outcome can't be blue since 1c isn't blue. Hence 3a must be red. From 3a, we conclude that $r \wedge y$ was resolved as r, so this new rule must apply everywhere. From 3c, we get a third rule that $b \wedge r = b$.
4. With two new rules in the third step in addition to the rule from the second, more colorings follow. These colorings then give us the rules $r \wedge b = r$, $y \wedge b = y$, and $y \wedge r = y$.
5. With all possible pairwise resolutions settled, all profiles can be colored.

We've found a monotonic, onto coloring! Notice that this is a dictatorship by Betty, choosing her top-ranked color for each profile. Everything comes down to favoring Betty over Bill in step two. Since Betty was pivotal once, she ends up having complete control. Of course, we could have resolved $b \wedge y$ as y instead, which would have given us a dictatorship by Bill. That choice in step two was the only degree of latitude we had, so these are the only two monotonic, onto colorings.

This special-case proof of Gibbard-Satterthwaite was inspired by Hansen (2002), "Another graphical proof of Arrow's impossibility theorem". Full proofs of the theorems, done simultaneously side-by-side, are given in Reny (2001), "Arrow's theorem and the Gibbard-Satterthwaite theorem: a unified approach".

Previously on: Incentive-Compatibility and the Revelation Principle

Next up: Strategyproof Mechanisms: Possibilities

---

1. Recall that the direct mechanism for an SCF is where we simply ask the agents what type they are and then assign the outcome prescribed by the SCF for that type profile.↩

2. The function f is *onto* if every outcome in A occurs for at least one input. The main role of this assumption to prevent f from covertly restricting its image to two elements, so it's almost without loss of generality.↩

Strategyproof Mechanisms: Possibilities

> *Despite dictatorships being the only strategyproof mechanisms in general, more interesting strategyproof mechanisms exist for specialized settings. I introduce single-peaked preferences and discrete exchange as two fruitful domains.*

Strategyproofness is a very appealing property. When interacting with a strategyproof mechanism, a person is never worse off for being honest (at least in a causal decision-theoretic sense), so there is no need to make conjectures about the actions of others. However, the Gibbard-Satterthwaite theorem showed that dictatorships are the only universal strategyproof mechanisms for choosing from three or more outcomes. If we want to avoid dictatorships while keeping strategyproofness, we'll have to narrow our attention to specific applications with more structure. In this post, I'll introduce two restricted domains with more interesting strategyproof mechanisms.

Before jumping into those, I should mention another potential escape route from Gibbard-Satterthwaite: randomization. So far, I've only considered deterministic social choice rules, where a type profile corresponds to a particular outcome. Considering randomized social choice rules—mapping a type profile onto a lottery over outcomes—will widen the scope of possibility just slightly. Instead of one person being the dictator, we can flip a coin to decided who is in charge. In particular, any strategyproof mechanism that chooses an outcome with certainty when all agents rank it as their first choice must be a *random dictatorship*, selecting among agents with some fixed probability and then choosing that agent's favorite. Unlike a deterministic dictatorship, a random dictatorship with equal weights on agents seems palatable enough to use in practice.

Single-peaked preferences

Our first refuge from dictatorship theorems is the land of single-peaked preferences on a line, a tree, or more generally a median graph. These settings have enough structure on preferences that a mechanism doesn't have to waste power eliciting rankings through dictatorship. Instead, we can deduce enough information from an agent's ideal outcome that we have power left over to do something more interesting with a mechanism.

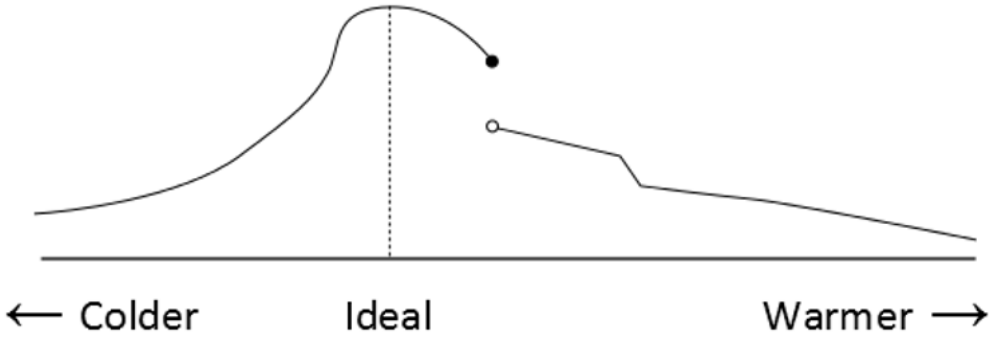
In the previous post, I discussed how strategyproofness comes down to the *monotonicity* of a social choice rule. Monotonicity says that if the outcome chosen by the rule is your favorite, the outcome can't change if you report a different ranking with the same top element but with other options swapped around. This essentially restricts a strategyproof mechanism to using agents' favorite alternatives as the only inputs.

With two alternatives to choose from, knowing an agent's favorite immediately tells us their full ranking, so we have room to build non-dictatorial mechanisms like majority rule. With three alternatives and unrestricted preferences, knowing an agent's favorite doesn't tell us anything about the ranking of the other two. For example, suppose we have three options: red, yellow, and blue. If two agents agree that red is best, we don't have enough information to unambiguously say whether switching from blue to yellow with make those agents better off or not.

What if instead the three alternatives had some natural ordering? For instance, some housemates have a terrible heating system with three settings: freezing, temperate, and burning. It's reasonable to assume that anyone who loves it hot or cold has "temperate" as their second choice, with no one preferring both extremes to something in the middle. Moving the thermostat from "freezing" to "temperate" makes everyone who has "temperate" or "burning" as their ideal better off and those with "freezing" as their ideal worse off, allowing us to order the alternatives:

This ordering allows us to reduce the decision between three alternatives into a multiple binary decisions that will be consistent with one another. One strategyproof, non-dictatorial mechanism would be to start with the thermostat at “temperate” and hold a vote whether or not to increase the temperature. If a majority support a temperature increase, the thermostat is bumped up and we’re done. If that vote fails, another one is held for a temperature decrease, with the majority winner of that vote being the final outcome. Another mechanism that gives the same results would be for everyone to submit their ideal temperature and choose the median (with a dummy vote of “temperate” to break ties if there are an even number of housemates). Take a moment to convince yourself this mechanism is strategyproof.

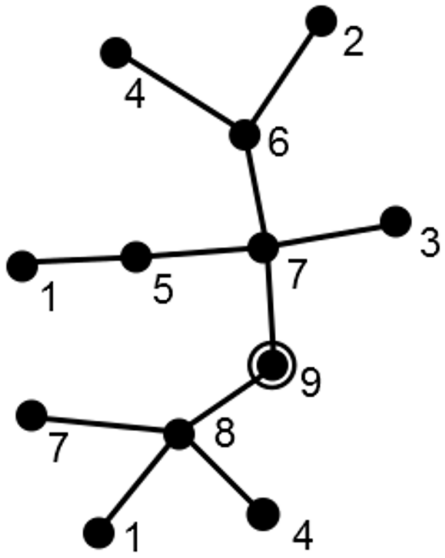
The median vote is well-defined even if the thermostat has more than three temperatures. We can push this all the way and consider every possible temperature. With a continuum of outcomes, it’ll be easier to think in terms of a utility function rather than a ranking, though we’re still only concerned with ordinal comparisons. Choosing the median will be strategyproof as long as preferences are *single-peaked*, with a single local maximum and utility falling off as we go further in either direction away from it. Here is one example of a single-peaked preference, depicting the utility for each temperature:



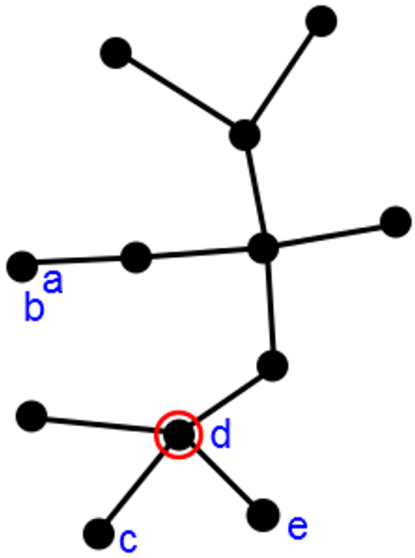
Since this agent has a complicated ranking over temperatures, asking agents to report only their peaks rather than their full utility function comes off as a practical advantage rather than a severe constraint.

Consider the thought process of someone deciding which temperature to report, knowing the median report will be chosen as the final outcome. If her ideal would be the median report (whether those are being made honestly or not), she has direct control over the outcome as long as she stays the median, so reporting something different makes her worse off. If her ideal is above the median, reporting something higher leaves the outcome unchanged. Reporting something lower either leaves the outcome unchanged or it makes the outcome drop even further away from her ideal than it already was. Similar reasoning goes through when her ideal is below the median. Hence she is never worse off for reporting her peak honestly. Contrast this with a mechanism that averages the peaks to get the outcome, which is not strategyproof.

Choosing an outcome on the real line has many natural applications like setting the temperature of a thermostat or the sales tax rate of a city. We can go further than lines though. For instance, single-peaked preferences make sense on a tree since we can talk unambiguously about going closer or further from some point as we move along the edges. Here is one tree, with an example of single-peaked preferences for it described by utilities for each node and the peak circled:



As on a line, we'll ask agents to report their peaks and choose the median node, i.e. the node that minimizes the distance to each agent's report. Suppose there are five agents, labeled *a* through *e*, and they report the following peaks:



The outcome will be the node circled in red, resulting in agent *d* getting his first choice since he is the "median" of the five.

Going even further, there are non-dictatorial strategyproof mechanisms on any graph where medians are unique and well-defined. We can also tweak the median rule, throwing in some dummy voters or weighting agents differently. However, it turns out that the median is essentially the only strategyproof mechanism that treats agents and outcomes symmetrically.

Discrete exchange

In the previous examples, the mechanism chose a single outcome for all agents. Consider instead a situation where each agent owns one object and agents might want to swap things

around. Rather than a single outcome, it makes more sense to think of sub-outcomes describing who gets which object, especially if each cares only about what he ends up with and not what the others get. Indifferences over parts of the allocation not involving that agent is another preference restriction that allows us to evade dictatorship theorems.

For example, three Roman soldiers have currently assigned duties which their eccentric new centurion will allow them to trade around. At the moment, Antonius is a standard-bearer, Brutus is a trumpeter, and Cato is an artilleryman. Even though the full outcome will be a list of who gets what, we consider only the preference of each over the three jobs. Perhaps Antonius has preferences *trumpeter* $\succ$ *standard-bearer* $\succ$ *artilleryman*, Brutus has preferences *standard-bearer* $\succ$ *artilleryman* $\succ$ *trumpeter*, and Cato has preferences *trumpeter* $\succ$ *standard-bearer* $\succ$ *artilleryman*. Given this, it's natural to say Antonius and Brutus should switch jobs, with Cato stuck as an artilleryman.

We can formalize this inclination using David Gale's *Top Trading Cycle* algorithm, which operates as follows:

1. Each agent starts as active. Each object starts as active and pointing at the agent that owns it.
2. Active agents point at their favorite active object.
3. At least one cycle must exist from agent to object to agent and etc. Deactivate agents and objects in a cycle.
4. Iterate steps 2 and 3 until all objects are deactivated. The final allocation is each object going to the agent pointing at it.

In the example, Antonius and Cato point at *trumpeter*, with Brutus pointing at *standard-bearer*. Deactivating the Antonius → *trumpeter* → Brutus → *standard-bearer* → Antonius cycle, only Cato is left active, so he points at his own job, and the mechanism is done.

As I've told the story, the soldiers already have control over a particular job. For instance, if Antonius liked being a standard-bearer best, he could guarantee he keeps the job. What would we do if the soldiers were new and didn't have a pre-existing job? One option is to randomly assign tasks and run the algorithm from there. This is actually how New Orleans matches K-12 students with public schools as of 2012. Since the mechanism is strategyproof, parents don't have to worry about gaming the system when they rank their top choices from the 67 schools in the district[1]. Unlike many of the toy examples I've given, here is a real-world case of improvements made by mechanism design.

Conclusion

By restricting preferences to be single-peaked or indifferent over the allocations of others, we can find useful strategyproof mechanisms. In the next post, I'll continue exploring strategyproof mechanisms in the very fruitful special case when we can make transfers between agents, introducing the famed Vickrey-Clarke-Groves mechanisms.

The idea of single-peaked preferences on a line is quite old, dating back to Duncan Black in 1948. For a full characterization of strategyproof mechanisms on a line, see Moulin (1980). For a characterization of which single-peaked preference domains admit non-dictatorial, strategyproof mechanisms, see Nehring and Puppe (2007).

For an overview of mechanism design aspects of school choice, see Abdulkadiroglu and Sonmez (2003).

Previously on: Strategyproof Mechanisms: Impossibilities

Next up: Mechanism Design with Money

---

1. Though technically there might be a tiny reason to misrepresent preferences since the parents since the ranking is truncated to the top eight schools rather than all 67.↩