Allocating Objects to People

Gabriel Carroll, gdc@stanford.edu
Stanford Math Circle
October 3, 2013

This Circle session is going to talk about a mathematical model for allocating objects, in situations when it is not acceptable or not possible to buy and sell the objects for money. It started out as a fun puzzle, introduced to the world in an article by Shapley and Scarf in 1974, but has recently seen important applications, including allocation of organs for transplantation and assignment of students to public schools.

These notes provide an outline of the essential definitions and results. This brevity might make them rather dry and hard to read. For the full experience, with examples and proofs, hear the lecture!

1 The basics

Suppose there are n *agents*[1], creatively named $1, 2, \ldots, n$. There are also n *objects* — say, rooms in a dormitory, or seats in a classroom, or jobs on a committee. The objects are also creatively named $a_1, a_2, \ldots, a_n$. Sometimes, when n is small, we'll call the objects instead by their nicknames a, b, c, d, etc. The problem is how to give the objects to the agents, with each agent getting one object. That is, we want to choose an *allocation* — a bijection from agents to objects.

Each agent i has a *preference* — a ranking of the objects, with a favorite, a second favorite, and so forth. We'll write $\succ_i$ for i's preference ranking, and also write, for example, $b \succ_i c$ to mean that i prefers b over c. And we'll write $\succeq_i$ for the weak form of the $\succ_i$ relation — that is, $a_j \succeq_i a_k$ when either a_j is preferred over a_k or $a_j = a_k$.

Notice that we've assumed each agent cares only about his own object, and doesn't care what object other people get; and also that he knows his own preferences perfectly. We'll assume also that there are no ties in the preference rankings.

If μ, μ' are two allocations, say that μ (weakly) *dominates* μ' if $\mu(i) \succeq_i \mu'(i)$ for every agent i. We say that an allocation is *efficient* if it is not dominated by any other allocation. So, an efficient allocation is one that gives agents the objects they like, as much as possible.

One way to allocate the objects is what's called *serial dictatorship*: give agent 1 the object he likes most; then give agent 2 the object he likes most out of the remaining objects; and so on. The result of this process is then some allocation. The above describes serial dictatorship with the agents choosing order $1, 2, \ldots, n$; but, of course, we could equally well define a serial dictatorship with the agents choosing in any other order.

Proposition 1. *An allocation is efficient if and only if there is some serial dictatorship that produces it.*

[1] "Agent" is roughly economics-speak for "person." But the word "agent" more generally denotes a decision-making unit — which might be a person, but also might be a family, say, or a company.

2 Introducing mechanisms

In general, when we want to allocate the objects, we don't actually know the agents' preferences. So what we really need is a *rule* for how the objects will be allocated, depending on the agents' preferences. Such a rule is called a *mechanism*. Formally, if we write $\mathcal{P}$ for the set of all $n!$ possible preferences, and $\mathcal{A}$ for the set of all $n!$ possible allocations, then a *mechanism* is a function $f : \mathcal{P}^n \to \mathcal{A}$. We might write $f_i(\succ_1, \ldots, \succ_n)$ for the object given to agent i when the preferences are $\succ_1, \ldots, \succ_n$.

Notice we haven't allowed randomness in the definition of a mechanism, although in practice we might want to use some randomness to make the process fair. More on that later!

Here are some examples of possible mechanisms:

- The mechanism that just always gives each agent i the object a_i, ignoring preferences
- Any serial dictatorship (check that this can be described as a mechanism!)
- The mechanism that gives each agent i the object that is ranked as agent 1's ith favorite
- The *Boston* mechanism, which produces an allocation by the following procedure:
  - First, try to give every agent his favorite object; if several agents want the same object, give it to whoever is lowest-numbered (say).
  - For the agents that don't yet have an object, try to give them their second-favorite object, if it's still available; again, break ties by lowest number.
  - For the agents that still don't have an object, try to give them their third-favorite, if it's still available, and so forth.

One desirable property for a mechanism is efficiency. We say a mechanism f is *efficient* if, for all possible preferences $\succ_1, \ldots, \succ_n$, the allocation $f(\succ_1, \ldots, \succ_n)$ is efficient (relative to those preferences).

Another desirable property in a mechanism is that it should be *strategyproof*: no agent can ever get a better outcome by lying about his preference. Why is this important? If the mechanism is efficient, but agents are strategically pretending to have different preferences than they actually have, then the outcome might be efficient with respect to the preferences they pretended to have, but bad with respect to their actual preferences.

To formalize this concept we need a little more notation: If $\succ_1, \succ_2, \ldots, \succ_n$ are preferences for each agent, and $\succ'_i$ is an alternative preference for some agent i, then we write $(\succ'_i, \succ_{-i})$ as shorthand for the list of preferences

$$\succ_1, \succ_2, \ldots, \succ_{i-1}, \succ'_i, \succ_{i+1}, \ldots, \succ_n .$$

Now, we say that mechanism f is *strategyproof* if, for all possible preferences $\succ_1, \ldots, \succ_n$, all agents i, and all possible preferences $\succ'_i$,

$$f_i(\succ_1, \ldots, \succ_n) \;\succeq_i\; f_i(\succ'_i, \succ_i).$$

Of the mechanisms in the list above, which are efficient? Which are strategyproof?

3 Initial allocations and the core

So far we have assumed that the objects initially are not owned by the agents, and are just waiting to be assigned. A variant on the model comes when there is some initial allocation μ_0 of the objects. (For example, agent 1 initially owns a_1, agent 2 initially owns a_2, and so on.) This model turns out to have a number of interesting properties.

In this variant model, we might want our final allocation μ to be not only efficient but also *individually rational*: for each agent i, $\mu(i) \succeq_i \mu_0(i)$. We'll say a mechanism f is *individually rational* if, for all $\succ_1, \ldots, \succ_n$, the allocation $f(\succ_1, \ldots, \succ_n)$ is individually rational (relative to those preferences).

We might think that the allocations that are efficient and individually rational are the ones that might be plausibly reached by starting from μ_0 and then having agents trade with each other.

Individual rationality represents the idea that no agent would prefer to refuse to accept the proposed allocation and instead just keep his original object. Actually, this idea generalizes: we could imagine groups of agents refusing the allocation together, and instead just trading among themselves. Given an allocation μ_0, and a proposed new allocation μ, we say that a nonempty set S of agents *blocks* μ if there exists another allocation μ' such that

(i) for every $i \in S$, there exists some $j \in S$ so that $\mu'(i) = \mu_0(j)$; and

(ii) $\mu'(i) \succeq_i \mu(i)$ for every $i \in S$, with $\mu'(i) \neq \mu(i)$ for at least one such i.

So an allocation is individually rational if there is no single individual agent who can block it; and it is efficient if it cannot be blocked by the set of all agents.

We now introduce a procedure for computing a new allocation, whose importance will be revealed in a moment. This procedure is called *top trading cycles* and works as follows:

- First, have every agent point to his favorite object, and have every object point to its (initial) owner. There must be at least one cycle of agents and objects pointing to each other. In each such cycle, remove the agents and objects, assigning each such agent the object he points to.
- Then, among the remaining agents, each points to his favorite of the remaining objects, and its object points to its owner (who is still among the remaining agents). Remove all cycles, as before.
- Rinse and repeat.

Proposition 2. *(Gale, reported in Shapley & Scarf, 1974) For any μ_0 and any preferences, there is exactly one allocation that is not blocked by any set of agents. It is the allocation computed by top trading cycles.*

The allocation produced by top trading cycles is also called the *core* allocation. The core is a natural choice of "stable prediction" for the result that should happen if the agents start from μ_0 and trade. Indeed, if we predicted any other outcome, then some set of agents could foresee that outcome and then block it.

There is actually another way to think of the core allocation in connection with trading. What if we allowed money, so the agents could buy and sell the objects, but in just such a way that no money actually ends up changing hands? Imagine that the objects may have some prices $p_1, p_2, \ldots, p_n$, which may be any nonnegative numbers. Some prices may be equal to each other. An allocation μ is a *market equilibrium* at these prices if, for each agent i, $\mu(i)$ is the best object that doesn't cost any more than $\mu_0(i)$ (according to $\succ_i$).

Proposition 3. *Given any μ_0, and given any preferences, there is exactly one allocation that is a market equilibrium at some prices — namely, the core allocation.*

We define the *core* mechanism (from a given initial allocation μ_0) to be the mechanism that, for any preferences $\succ_1, \succ_2, \ldots, \succ_n$, chooses the corresponding core allocation. We have already seen that this mechanism is individually rational and efficient. In fact, it is also strategyproof!

Theorem 4. *(Roth, 1982) The core mechanism is strategyproof.*

At this point we have a lot of reasons to like the core mechanism. But it turns out that there is another result that is even more awesome than Roth's theorem. It highlights just how special the core mechanism actually is.

Theorem 5. *(Ma, 1994) Fix an initial allocation μ_0. Then, there is only one mechanism that is efficient, individually rational, and strategyproof: namely, the core mechanism.*

4 Random allocations

Now let's go back to our original allocation problem. The n objects are just sitting in space; there's no initial ownership. We knew from the beginning that serial dictatorship — say, letting agent 1 choose, then 2, and so forth — is efficient, and strategyproof. But it's not very fair.

We now know another efficient and strategyproof mechanism. Namely, allocate the objects initially in some arbitrary way (say, allocating object a_1 to agent 1, a_2 to agent 2, and so on), and then find the core allocation. This, however, might still not be fair — for example if a_1 is a very popular object, agent 1 is favored.

But both of these mechanisms can be made fair, in an intuitive way, simply by randomizing the positions of the agents first. So, allowing for randomization, we have two natural ways to allocate the objects:

(a) arrange the agents in a random order, with all orders equally likely, and then apply serial dictatorship;

(b) choose an initial allocation of the objects at random, with all allocations equally likely, and then find the core.

Both of these procedures are efficient and strategyproof. Is one of them better than the other?

Remarkably, the answer is no, because of the following theorem:

Theorem 6. *(Abdulkadiroğlu & Sönmez, 1998) The two randomized mechanisms (a) and (b) above are identical. That is, each allocation μ has the same probability of being produced under (a) as under (b).*

This randomized mechanism often goes by the name of *random serial dictatorship* (even though the above theorem shows that we would be equally justified in calling it the "random core").

Random serial dictatorship inherits the strategyproofness property of serial dictatorship (we'll make this precise a bit later). And serial dictatorship is efficient — there's no alternative way to allocate the objects that would make all the agents happier. Combining these with the fairness property, it seems like random serial dictatorship has everything we should want. And the Abdulkadiroğlu-Sönmez theorem seems to show even more how central it is. So random serial dictatorship seems like the perfect way to allocate objects. Right?

Well, actually the efficiency bit is more complicated. Consider the following example. There are 4 agents and 4 objects, and the preferences are

$$\begin{array}{ccccccccc} 1 & : & a & \succ_1 & b & \succ_1 & c & \succ_1 & d \\ 2 & : & a & \succ_2 & b & \succ_2 & c & \succ_2 & d \\ 3 & : & b & \succ_3 & a & \succ_3 & c & \succ_3 & d \\ 4 & : & b & \succ_4 & a & \succ_4 & c & \succ_4 & d \end{array}$$

A quick computation finds that, under random serial dictatorship, each agent has a 5/12 chance of getting his first choice, a 1/12 chance of getting his second choice, and a 1/4 chance each for his third and fourth choices.

On the other hand, if we knew these were the preferences, we could instead propose the following randomized allocation: give a to 1 or 2 (with equal probability); give b to 3 or 4 (with equal probability); and split c and d between whichever two agents are not yet assigned. This gives each agent a 1/2 chance of his favorite object, and 1/4 each for his third and fourth choices. So, compared to random serial dictatorship, each agent is better off: a 1/12 probability share has moved from his second-choice object to his first choice. So, random serial dictatorship wasn't so efficient after all!

What's going on here? It's true that *after* the objects are allocated, we have an efficient assignment, and there's no way to make everyone happier by trading. But before the allocation happens, when agents are facing uncertainty about what objects they'll get, there is room for improvement.

Let's see if we can describe the issue — and seek a solution — in more precise terms. First, we need to talk about *random allocations*. One natural way to define a random allocation is as a probability distribution over the set $\mathcal{A}$. However, $\mathcal{A}$ has $n!$ elements. This is a lot, and so these probability distributions might be cumbersome to write out. It turns out we can economize, for the following reason. The only thing each agent cares about is his probability of receiving each object. That is: define an *object lottery* as a vector $p = (p_1, p_2, \ldots, p_n)$, consisting of n nonnegative numbers summing to 1; each p_j represents the probability of receiving object j. Any probability distribution on $\mathcal{A}$ then naturally gives rise to n object lotteries, one for each agent. These object lotteries are the only thing the agents care about, and so we can encapsulate all the relevant information in these n^2 numbers.

Can every vector of n object lotteries $p^1, p^2, \ldots, p^n$ (one per agent) be obtained from a randomization over allocations? Not quite: each object a_j has to go to exactly one agent, so we must have $p^1_j + p^2_j + \cdots + p^n_j = 1$. It turns out that this is the only constraint; this is a restatement of a famous theorem of combinatorics.

Theorem 7. *(Birkhoff–von Neumann Theorem) If $p^1, p^2, \ldots, p^n$ are object lotteries such that $p^1_j + p^2_j + \cdots + p^n_j = 1$ for each object a_j, then there exists a probability distribution over allocations μ such that, for all agents i and objects a_j, the probability that $\mu(i) = a_j$ is exactly p^i_j.*

So with this in mind, we can forget about the complete description in terms of allocations, and just define a *lottery allocation* as a collection of n object lotteries satisfying the constraints $p^1_j + \cdots + p^n_j = 1$ for each j. If $\widehat{\mathcal{A}}$ is the set of all lottery allocations, then we can define a *lottery mechanism* as a function $f : \mathcal{P}^n \to \widehat{\mathcal{A}}$.

What do agents' preferences over object lotteries look like? Suppose, for instance, we know agent i's preference satisfies $a \succ_i b \succ_i c$. We might not be able to conclude whether a fifty-fifty chance of a versus c is or isn't better than getting b for sure. But at least some lotteries can be compared. For example, a fifty-fifty chance of a versus b is surely better than getting c for sure. More generally, suppose i's preference over objects is $\succ_i$. We can then compare two object lotteries p and p' by saying that $p \mathrel{\widehat{\succeq}_i} p'$ if any of the following three (equivalent) conditions is satisfied:

(a) For every $k = 1, 2, \ldots, n$, the total probability of getting one of the k best objects (as rated by $\succ_i$) under p is at least as high as the total probability of these objects under p'.

(b) It is possible to get from p' to p by a sequence of "improving moves," where an improving move consists of reducing the probability of some object by some amount $\epsilon \geq 0$ and increasing the probability of a more-preferred object by ϵ.

(c) For any scores $x_1, x_2, \ldots, x_n$ that respect the ranking $\succ_i$, in the sense that $x_j \geq x_k$ whenever $a_j \succ_i a_k$, then the expected score under p is at least as high as under p':

$$p_1 x_1 + \cdots + p_n x_n \geq p'_1 x_1 + \cdots + p'_n x_n.$$

Now we can redefine efficiency in this world of lotteries. If we have two lottery allocations, $p = (p^1, \ldots, p^n)$ and $p' = (p'^1, \ldots, p'^n)$, we say that p' *dominates* p' (with respect to given preferences) if $p^i \mathrel{\widehat{\succeq}_i} p'^i$ for each i. We then say that a lottery allocation is *ordinally efficient* if it is not dominated by any other lottery allocation. A lottery mechanism is *ordinally efficient* if, for any preferences, it always chooses an ordinally efficient lottery allocation.

What about strategyproofness? We want to be sure that each agent is always better off reporting the truth than reporting a false preference. So, we say f is *strategyproof* if, for all preferences $\succ_1, \ldots, \succ_n$, all agents i, and all other preferences $\succ'_i$, we have

$$f^i(\succ_1, \ldots, \succ_n) \mathrel{\widehat{\succeq}_i} f^i(\succ'_i, \succ_{-i}).$$

In this extended setting, are there mechanisms that are ordinally efficient and strategyproof? Sure: every non-random allocation can be thought of as a lottery allocation (where all the probabilities are 0 or 1); we can check that efficient allocations are now ordinally efficient, and strategyproofness carries over too. So, for example, a serial dictatorship (with the agents choosing, say, in order $1, 2, \ldots, n$) is still efficient and strategyproof. But it's not very fair.

Now that we are looking at random mechanisms, we can articulate a specific criterion for fairness. One of the least demanding criteria is called *equal treatment of equals*: For all preferences $\succ_1, \succ_2, \ldots, \succ_n$ satisfying $\succ_i = \succ_j$ for some i and j, we have

$$f^i(\succ_1, \succ_2, \ldots, \succ_n) = f^j(\succ_1, \succ_2, \ldots, \succ_n).$$

We can check, for example, that random serial dictatorship satisfies this.

Can we find a lottery mechanism that is efficient, strategyproof, and fair? With the criteria we have defined, this turns out to be impossible:

Theorem 8. *(Bogomolnaia & Moulin, 2001) If $n \geq 4$, there is no lottery mechanism that is ordinally efficient, strategyproof, and satisfies equal treatment of equals.*

5 Problems!

1. For what preferences is there just one efficient allocation? For what preferences is every allocation efficient?

2. Consider the version of the model with an initial allocation. Imagine that two agents can meet and trade the objects they are currently holding, as long as the trade makes each of them better off. Is it always possible, for any initial allocation and any preferences, to eventually get to the core allocation by a sequence of such trades?

3. Suppose that the agents' preferences have the following property: there is a set S of $n - k$ agents and T of k objects, for some $1 \leq k < n$, such that every agent in S prefers agent 1's bottom-ranked object above any object in T. Prove that in any efficient allocation, agent 1 cannot get stuck with his bottom-ranked object.

4. A mechanism is *group-strategyproof* if there is no way any group of agents can jointly lie about their preferences and end up better off.

   Let's give a more precise definition. For any preferences $\succ_1, \ldots, \succ_n$, any set S of agents, and alternative preferences $\succ'_i$ for each $i \in S$, write $(\succ'_S, \succ_{-S})$ for the new list of preferences obtained by replacing $\succ_i$ with $\succ'_i$ for each $i \in S$. Now, when the true preferences are $\succ_1, \ldots, \succ_n$, we say the nonempty set S of agents has a *group deviation* if there are false preferences $\succ'_i$ for each $i \in S$ such that

   $$f_i(\succ'_S, \succ_S) \ \succeq_i \ f_i(\succ_1, \ldots, \succ_n) \qquad \text{for all } i \in S,$$

   and $f_i(\succ'_S, \succ_S) \neq f_i(\succ_1, \ldots, \succ_n)$ for at least one such i. The mechanism f is *group-strategyproof* if there is never any group deviation, for any preferences and any nonempty set of agents.

   (a) Give a simple example of a mechanism that is group-strategyproof.

   (b) Give another example of a mechanism that is strategyproof but not group-strategyproof.

5. We've been assuming that agents' preferences have no ties: for any two different objects, one is better than the other. Suppose now that ties are allowed.

   We can again define *blocking* of an allocation. The definition is the same as before, except that now condition (ii) is changed to say that every $i \in S$ should like $\mu'(i)$ at least as much as $\mu(i)$, and at least one $i \in S$ should like $\mu'(i)$ strictly more than $\mu(i)$.

   Give an example to show that there may be no core allocation — that is, it may happen that no matter what allocation is proposed, some set of agents blocks it.

6. Consider the model with an initial allocation. Suppose all n agents negotiate over a new allocation, as follows. Whenever an allocation μ is proposed, if some set S of agents blocks μ, via some allocation μ' that requires only trading among agents in S (i.e. $\mu'(i) = \mu_0(i)$ for each $i \notin S$), then they can propose μ' instead. Then some other set of agents that blocks μ' may propose another allocation μ'', and so on.

   Is it always possible, starting from any proposed allocation μ, to get to the core allocation by a sequence of such blocking proposals?

7. Prove that the three given definitions of $\widehat{\succeq}_i$ (preference over object lotteries) really are equivalent.

8. A mechanism is called *nonbossy* if, whenever an agent changes his reported preference, he cannot change anyone else's assigned object unless he changes his own too. That is, for all preferences $\succ_1, \ldots, \succ_n$, all agents i and preferences $\succ'_i$,

   $$\text{if} \qquad f(\succ_1, \ldots, \succ_n) \neq f(\succ'_i, \succ_{-i}), \qquad \text{then} \qquad f_i(\succ_1, \ldots, \succ_n) \neq f_i(\succ'_i, \succ_{-i}).$$

(a) Prove that a mechanism is group-strategyproof if and only if it is both strategyproof and nonbossy.

(b) Conclude that the core mechanism (from any initial allocation μ_0) is group-strategyproof.

9. We saw that every lottery allocation can be obtained by a randomization over pure (non-lottery) allocations. Now let's constrain this by looking at efficiency.

   (a) True or false: Every ordinally efficient lottery allocation can be obtained by a randomization over efficient pure allocations. (Prove or give a counterexample.)

   (b) True or false: Every ordinally *in*efficient lottery allocation can be obtained by a randomization over pure allocations that are not all efficient.

10. For lottery mechanisms, we introduced equal treatment of equals as one possible criterion for fairness. Another that is often talked about is *envy-freeness*. Given preferences $\succ_1, \ldots, \succ_n$, a lottery allocation $\widehat{\mu}$ is *envy-free* if, for all agents i and j, $\widehat{\mu}(i) \mathrel{\widehat{\succeq}_i} \widehat{\mu}(j)$. That is, each agent would rather have his own object lottery than anyone else's.

    Is the lottery allocation that results from random serial dictatorship always envy-free?

11. Bogomolnaia and Moulin (2001) proposed to solve the problem of ordinal inefficiency by proposing a new lottery mechanism, called *probabilistic serial*. It chooses the lottery allocation defined as follows.

    Imagine representing each object by a 1-pound pile of ice cream. Each agent now simultaneously "eats" from his favorite object, at a constant speed of 1 pound per hour. Whenever an object gets fully eaten up, each agent who was eating it now switches to his favorite of the remaining objects. Since there are n agents eating n pounds of ice cream, after 1 hour, all the ice cream will be finished and each agent will have eaten exactly 1 pound, in some combination of flavors. Now turn this dessert feast back into a lottery allocation for the original objects, as follows: if p_j^i represents the amount of a_j-flavored ice cream that i ate, then we now have i receive object a_j with exactly that probability p_j^i.

    (a) Prove that the resulting lottery allocation is envy-free, and also that the mechanism satisfies equal treatment of equals.

    (b) Prove that the resulting lottery allocation is ordinally efficient.

    (By the Bogomolnaia-Moulin theorem, then, we know that this mechanism cannot be strategyproof in general.)

12. We may or may not have proved most of the theorems in this handout during the lecture, depending how far we got. But we definitely didn't prove the Bogomolnaia-Moulin theorem. For a serious challenge, try your hand at proving it.