Probabilistic assignments

Haihan Yu

Multi-agent Lab, Kyushu University

August 31, 2019

Introduction

Assignment (or matching) algorithms are rarely **deterministic** (re-running the algorithm with the same preferences the outcome does not change).
There are often some randomness. For instance:

- Objects' priorities can be coarse: two or more individuals are equally ranked.
  We **need** to introduce randomness to break ties (e.g., if the law prohibits to sort according to the name or other factors)
- The random order in the serial dictatorship.
  Randomness is **not needed**: the algorithm does nor require the order of dictators to be random.

Randomness can also be used to introduce some **fairness**.
2 objects to be assigned between two people.

P_{Alice}	P_{Bob}
A	A
B	B

- A deterministic assignment will favor one of the individuals at the expense of the other.
  For instance: Alice always gets A, her top choice.
- A **random assignment** can make the allocation **fair**: each individual has a probability of 50% to get her top choice.

Random assignments

We study here **random assignment** problems, which consist of:

- A finite set of individuals $I = \{i_1, i_2, \ldots, i_n\}$; and
- A finite set of objects $I = \{k_1, k_2, \ldots, k_n\}$.

For simplicity, we assume:

- each individual has a strict preference ordering over **all** objects (i.e., all objects are **acceptable**);
- There are as many objects as there are individuals.

Definition

A **random assignment** is a collection of $n \times n$ probabilities β, such that

(a) For each agent–object pair (i, k) there is a probability $\beta_{i,k}$ that indicates the probability that individual i is assigned object k;

(b) For each individual i, $\sum_{k \in K} = \beta_{i,k_1} + \beta_{i,k_2} + \cdots + \beta_{i,k_n} = 1$. *each individual has a probability 1 to get an object.*

(c) For each object k, $\sum_{i \in I} = \beta_{i_1,k} + \beta_{i_2,k} + \cdots + \beta_{i_n,k} = 1$. *each object has a probability 1 to be assigned to some individual.*

We could dispense with *(b)* and *(c)*. But it's easier to do like this.

Example

A random assignment can be represented by a matrix:

$$\begin{array}{cc} & \begin{array}{ccc} \text{apple} & \text{orange} & \text{pear} \\ \downarrow & \downarrow & \downarrow \end{array} \\ \begin{array}{c} \text{Alice} \rightarrow \\ \text{Bob} \rightarrow \\ \text{Carol} \rightarrow \end{array} & \begin{pmatrix} 0.3 & 0.7 & 0 \\ 0.5 & 0.3 & 0.2 \\ 0.2 & 0 & 0.8 \end{pmatrix} \end{array}$$

Alice obtains:

- an apple with a probability of 0.3: $\beta_{\text{Alice,apple}} = 0.3$.
- an orange with a probability of 0.7: $\beta_{\text{Alice,orange}} = 0.7$.
- a pear with a probability of 0: $\beta_{\text{Alice,pear}} = 0$.

A **deterministic assignment** is also a probabilistic assignment. The assignment

$$\mu(\text{Alice}) = \text{orange} \qquad \mu(\text{Bob}) = \text{apple} \qquad \mu(\text{Carol}) = \text{pear}$$

has this following matrix representation:

$$\begin{array}{cc} & \begin{array}{ccc} \text{apple} & \text{orange} & \text{pear} \\ \downarrow & \downarrow & \downarrow \end{array} \\ \begin{array}{c} \text{Alice} \rightarrow \\ \text{Bob} \rightarrow \\ \text{Carol} \rightarrow \end{array} & \begin{pmatrix} 0 & 1 & 0 \\ 1 & 0 & 0 \\ 0 & 0 & 1 \end{pmatrix} \end{array}$$

The Birkhoff–von Neumann theorem

A random assignment assigns to each **individual-object pair** a probability.
How do we ensure that no two individuals eventually end up with the same object?
The way we described the probabilities implicitly assumed that they are independent. So both Alice and Bob could get **at the same time** the apple.
There a in fact two ways to describe a random assignment, and they are (somehow) equivalent.

Two possible approaches:

- A random assignment assigns to each individual-objet pair a probability (our definition).
- A random assignment assigns probabilities to deterministic assignments.
  Example: We have with probability 0.4 the assignment

  $\mu(\text{Alice}) = \text{orange} \qquad \mu(\text{Bob}) = \text{apple} \qquad \mu(\text{Carol}) = \text{pear}$

  and with probablity 0.6 the assignment

  $\mu'(\text{Alice}) = \text{pear} \qquad \mu'(\text{Bob}) = \text{orange} \qquad \mu'(\text{Carol}) = \text{apple}$

Theorem (Birkhoff–von Neumann)

Any random assignment can be decomposed into a lottery over deterministic assignments.

Remark

The decomposition is not necessarily unique.

Example

The following deterministic assignments can decompose the probabilistic assignment of the previous example:

$$\mu = \begin{pmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{pmatrix} \qquad \mu' = \begin{pmatrix} 0 & 1 & 0 \\ 1 & 0 & 0 \\ 0 & 0 & 1 \end{pmatrix} \qquad \mu'' = \begin{pmatrix} 0 & 1 & 0 \\ 0 & 0 & 1 \\ 1 & 0 & 0 \end{pmatrix}$$

with the probabilities

$$Prob(\mu) = 0.3 \qquad Prob(\mu') = 0.5 \qquad Prob(\mu'') = 0.2$$

Alice gets the orange with μ' and μ''. This occurs with probability

$$0.5 + 0.2 = 0.7$$

which is precisely $\beta_{\text{Alice,orange}}$.

Evaluating random assignments

When Alice compares two deterministic assignments she only needs to look at the object she gets: she prefers μ to μ' if she prefers her object with μ to the one with μ'.
With probabilistic assignment we cannot do that.
Worse, we cannot calculate **expected payoffs**: the only information we have is an ordering.

Stochastic dominance

To compare vector of probabilites (one probability for each object) we proceed the following way:

- Compare the probabilities to have the most preferred object.
- Compare the probabilities to have the **two most** preferred objects.
- Compare the probabilities to have the **three most** preferred objects.
- etc.

If the comparison always go in the same direction then we can say that one probabilistic assignment **dominates** the other.

Definition

For an individual i, a probabilistic assignnment β **stochastically dominates** a probabilistic assignnment β' if, for each $h = 1, \dots, n-1$,

$$\sum_{\ell \le h} \beta_{i,k_\ell} \ge \sum_{\ell \le h} \beta'_{i,k_\ell} \, .$$

Remark

We don't need to calculate up to $h = n$: Part (b) of the definition of a probabilistic assignment states that

$$\sum_{h=1}^{n} \beta_{i,h} \; = \; 1 \; = \; \sum_{h=1}^{n} \beta'_{i,h}$$

Remark

β not dominating $\beta' \not\Rightarrow \beta'$ dominates β'.

Example

Suppose Alice's preferences are: apple, orange, pear, cherry.

	β	β'	β''
apple	0.2	0.3	0.3
orange	0.3	0.3	0.6
pear	0.1	0.2	0.1
cherry	0.4	0.2	0

Probabilities to get something at least as good as:

	β	β'	β''
the apple			
the orange	0.5	0.6	0.9
the pear	0.6	0.8	1

So β'' dominates β', which and they both dominate β.

Random serial dictatorship

Obvious critique for serial dictatorship: individuals high in the queue are favored.

Random serial dictatorship brings fairness: everyone has the same chances to be first, last, or at any other position.

Nothing much to mention (for now), except that is has an interesting connection with TTC.

The TTC with mixed endowments had a trick: assign randomly the public endowments to new applicants.

We can generalize this idea and obtain a **TTC with random endowments**.

Example

P_{Alice}	P_{Bob}	P_{Carol}
k_1	k_1	k_3
k_2	k_2	k_2
k_3	k_3	k_1

Order for serial dictatorship: Alice, Bob, Carol. Assignment is:

$$\mu(\text{Alice}) = k_1 \qquad \mu(\text{Bob}) = k_2 \qquad \mu(\text{Carol}) = k_3$$

For TTC, endow with the following objects:

$$\text{Alice} \to k_1 \qquad \text{Bob} \to k_3 \qquad \text{Carol} \to k_2$$

First cycle: Alice points to herself. Second cycle: Bob $\to$ Carol $\to$ Bob Assignment: μ again!

Theorem (Abdulkadiroğlu and Sönmez)

For any assignment problem with the same number of individuals and objects:
The random assignments generated by the Random Serial Dictatorship algorithm are the same (with the same probabilities) as the random assignments generated by the Top Trading Cycle algorithm with random endowments.

Ex-ante v. ex-post

When introducing randomness in an assignment there are two ways to study the assignments:

- **Ex-ante**:
  We look at the assignments that can be obtained **before** the random draw is realized.
  We use stochastic dominance to compare assignments.
- **Ex-post**:
  We look at the assignments that are obtained **after** the random draw is realized.
  To compare assignments we compare the objects the individuals receive.

Definition

A random assignment β is **ex-ante** efficient (or **ordinally efficient** if there is no other random assignment β' such that, for each individual, β' stochastically dominate β.

The problem is that ex-ante efficiency is **not equivalent** to **ex-ante efficiency**.
Serial dictatorship is **ex-post efficient**. But it is **not** ex-ante efficient...

Example

P_{Alice}	P_{Bob}	P_{Carol}	P_{Denis}
apple	apple	orange	orange
orange	orange	apple	apple
pear	pear	strawberry	strawberry
strawberry	strawberry	pear	pear

For serial dictatorship there are 24! different orders.

But

$$\begin{array}{cc} & \begin{array}{cccc} \text{apple} & \text{orange} & \text{pear} & \text{strawberry} \\ \downarrow & \downarrow & \downarrow & \downarrow \end{array} \\ \begin{array}{c} \text{Alice} \rightarrow \\ \text{Bob} \rightarrow \\ \text{Carol} \rightarrow \\ \text{Denis} \rightarrow \end{array} & \begin{pmatrix} 5/12 & 1/12 & 5/12 & 1/12 \\ 5/12 & 1/12 & 5/12 & 1/12 \\ 1/12 & 5/12 & 1/12 & 5/12 \\ 1/12 & 5/12 & 1/12 & 5/12 \end{pmatrix} \end{array}$$

is stochastically dominated by

$$\begin{array}{cc} & \begin{array}{cccc} \text{apple} & \text{orange} & \text{pear} & \text{strawberry} \\ \downarrow & \downarrow & \downarrow & \downarrow \end{array} \\ \begin{array}{c} \text{Alice} \rightarrow \\ \text{Bob} \rightarrow \\ \text{Carol} \rightarrow \\ \text{Denis} \rightarrow \end{array} & \begin{pmatrix} 1/2 & 0 & 1/2 & 0 \\ 1/2 & 0 & 1/2 & 0 \\ 0 & 1/2 & 0 & 1/12 \\ 0 & 1/2 & 0 & 1/12 \end{pmatrix} \end{array}$$

One way to obtain ex-ante efficient assignment is to use an **eating algorithm**.
Such algorithms imagine that individuals "eat" the objects, and the share of each object they ate give the probabilities.
Probabilistic Serial algorithm:

- When the algorithm starts each individual starts eating her most preferred object.
- When the object an individual is eating is entirely eaten then
  - The fraction of the object that has been eaten will be the probability that the individual is assigned that object.
  - The individual starts eating the most preferred object among the objects that are not entirely eaten yet.

Example I

$$\begin{array}{c|cccc} & \text{apple} & \text{orange} & \text{pear} & \text{strawberry} \\ & \downarrow & \downarrow & \downarrow & \downarrow \\ \hline \text{Alice} \rightarrow & 1/2 & 0 & 1/2 & 0 \\ \text{Bob} \rightarrow & 1/2 & 0 & 1/2 & 0 \\ \text{Carol} \rightarrow & 0 & 1/2 & 0 & 1/12 \\ \text{Denis} \rightarrow & 0 & 1/2 & 0 & 1/12 \end{array}$$

- Alice and Bob start eating the apple
  $\Rightarrow$ they both eat one half, so they both get a probability of $\frac{1}{2}$ to have it.
  Same for Carol and Denis: $\frac{1}{2}$ probability to have the orange.
- Next Alice and Bob eat the pear (orange already eaten by Carol and Denis).
  $\Rightarrow$ they both eat one half, so they both get a probability of $\frac{1}{2}$ to have it. Same witht the strawberry for Carol and Denis.

Example II

P_{Alice}	P_{Bob}	P_{Carol}
apple	apple	orange
pear	orange	pear
orange	pear	apple

- Alice and Bob eat the apple, get $\frac{1}{2}$ of it.
  When done, Carol ate $\frac{1}{2}$ of the orange.
- Next Bob joins Carol on the orange.
  Both Bob and Carol eat $\frac{1}{2}$ of what's left of it.
  Bob ate $\frac{1}{4}$ and Carol $\frac{3}{4}$ of the orange.
- In the meantime, Alice ate $\frac{1}{4}$ of the pear.

- Then they all eat what's left of the pear ($\frac{3}{4}$).
  So they all eat $\frac{1}{4}$ of it.
- So we obtain

apple orange pear

Theorem (Bogomolnaia and Moulin)

For any problem and for any eating speed functions the random assignment calculated with the probabilistic serial algorithm is ex-ante efficient.
Conversely, for any ex-ante efficient random assignment there exists an eating speed function for each individual such that the probabilistic serial algorithm yields this random assignment.

Randomness introduced to introduce some fairness. The formal notion generally used is **equal treatment of equals**: two individuals with the same preferences should obtain the same probabilities.

Theorem (Bogomolnaia and Moulin)

Whenever there at least four individuals, there is no random assignment mechanism that is always ***ex-ante efficient****,* ***strategyproof*** *and that satisfies the* ***equal treatment of equals*** *property.*

Take-away

- Some “randomness” (or constructed probabilities) are often used in assignment mechanisms. They permit to address fairness issues.
- Probabilistic assignments are best evaluated using **stochastic dominance** (but we may want to use other methods).
- **Random Serial Dictatorship** is a common and simple probabilistic assignment mechanism. It is equivalent to **Top Tradind Cycle with random endowments**.
- With probabilistic assignments there are two way to gauge an assignments: **ex-ante** and **ex-post**
- Ex-ante efficiency and ex-post efficiency are not equivalent.

- Random Serial Dictatorship is ex-post efficient but it is **not ex-ante efficient**.
- Any ex-ante efficiency can be obtained using an eating algorithm like **probabilistic serial** (using appropriate eating function speeds).
- With 4 or more individuals, **ex-ante efficiency**, **strategyproofness** and **equal treatment of equals** are incompatible.