A New Solution To The Random Assignment Problem

By Anna Bogomolnaia, Herve Moulin

Presented By Zach Jablons, Bharath Santosh

The Assignment Problem

- How to best assign *n* objects to *n* agents
- Lotteries
  - Random assignments of objects to agents
- Random Priority mechanism
  - AKA Random Serial Dictatorship
  - Draw a random ordering of agents, then let them pick objects in that order

Properties

- Random Priority is fair
- Incentive compatible
  - Agents have no reason to lie about their preference
- Inefficient in a certain setting
  - When agents have Von Neumann-Morgenstern (VNM) preferences over lotteries
  - VNM preferences are characterized by VNM utility function
    - Simply the expected value over the lotteries

The Assignment Problem

- CEEI
  - View VNM utility function as utility over shares
  - Shares are the probability of receiving
- Properties
  - Not strategyproof
    - In fact no such mechanism can be strategyproof
  - Efficient for VNM utilities

Different types of Efficiencies

- Ex-Post Efficiency
  - All possible assignments are Pareto optimal
- Ex-Ante Efficiency
  - Efficient in terms of the profile of VNM utilities
- New! Ordinal Efficiency
  - In terms of distributions over assignments
  - Most probable and most valuable in terms of utilities
  - Will get into more detail later

Notation

- N is the set of n agents, A is the set of n objects
- Π is some bistochastic matrix of 1s and 0s
  - Deterministic assignment
- $\underline{D}$ is the set of all Π
- P is some bistochastic matrix
  - Random assignment
  - Weighted sum of all $\Pi \in \underline{D}$
- $\underline{R}$ is the set of all P
- > is all agents strict preference orders over A
- $\underline{A}$ is the domain of A

More notation

- A random allocation to an agent is a probability distribution over A
- L(A) is the set of all such allocations
- u_i is a mapping of A -> $\mathbf{R}^n$, the VNM utility
    - u is the profile over all of these
- Compatibility: $>_i$ is compatible with u_i means that for any $a, b \in A$,
    - $a > b$ in $>_i$ iff $u_i(a) > u_i(b)$

Even more notation

- σ is an ordering of agents
- θ is the set of all such orderings
- Prio(σ, >) is a function mapping the orderings and the set of preferences to a deterministic assignment
- Prio creates an assignment by going through the ordering σ and giving each agent their top-ranked available item by >

Efficiencies

- Given some random assignment matrix P and a profile of utilities u compatible with a profile of preferences >
  - Ex-ante efficiency comes from:
    - Pareto optimality at u
  - Ex-post efficiency
    - If P can be represented as a sum over a distribution of Prio(σ,>) from all possible orderings σ with some weights

Random Priority

- In this notation, easy to define random priority assignment
- P is the average over all Prio(σ,>)
    - All weights are 1/n!
    - That is, average over all serial dictatorships

Stochastic Dominance

- A strict ordering $>_i$ implies a partial ordering on $\underline{L}(A)$
- This is called the stochastic dominance relation, $sd(>_i)$
- Formally, given some P_i and Q_i from $\underline{L}(A)$
    - P_i $sd(>_i)$ Q_i iff for all t in [1,n], the sum over the row P_i from 1 to t is greater than or equal to Q_i's sum
    - Example

Stochastic Dominance

- Given some preference $>_i$, P_i *sd*($>_i$) Q_i is equivalent to $u_i P_i >= u_i Q_i$ for all compatible utilities u_i
- Definition: If some random assignment P dominates some other random assignment Q for all agents, then Q is stochastically dominated by P

Ordinal Efficiency (O-efficiency)

- A random assignment P is O-efficient if it is not stochastically dominated by any other random assignment
- Some corollaries
  - If P is ex-ante efficient for u, then it is O-efficient at $>$
  - If P is ex-post efficient for $>$, then it is O-efficient at $>$
  - Extra conditions when $n <= 4$

Simultaneous Eating Algorithm

- Each object is an infinitely divisible commodity
- Each agent has an eating speed function $\omega_i(t)$
  - Each agent is allowed to consume an object with speed $\omega_i(t)$ at time t
  - $\omega_i(t)$ is non-negative and integrates to 1 over the interval [0,1]

Simultaneous Eating Algorithm

- Simply allow agents to ‘eat’ from their best available objects at the specified eating speeds
- Example

Simultaneous Eating Algorithm

- Getting P_ω can be done with an iterative algorithm
- M(a,A) is the set of agents who prefer a to all other objects in A.
- Initialize: $A^0 = A$, $y^0 = 0$, P^0 = zeros(n,n)
- Basically this formalizes having each agent eat from their best available object, and the algorithm finds best times to allow

Simultaneous Eating Algorithm

- Let $y^s(a)$ be the minimum y such that the
  - sum over all agents i in $M(a,A^{s-1})$ of the integral from y^{s-1} to y of $\omega_i(t)$
  - plus the sum over all agents of the probability of that agent getting a in P^{s-1}
  - is equal to 1.
  - With the condition that $y^s(a)$ be ∞ if there are no agents that prefer a to all other objects in A^{s-1}

Simultaneous Eating Algorithm

- At each step s, let
  - y^s be the minimum $y^s(a)$ over all objects in A^{s-1}
  - A^s be A^{s-1} without the object that minimized y^s
  - P^s be the following
    - Update each cell $P^s[i,a]$ by using the previous if i is not in the set of agents that prefer a to any other object
    - Otherwise add the eating speed $\omega_i(t)$ integrated from y^{s-1} to y^s to $P^{s-1}[i,a]$

Simultaneous Eating Algorithm

- Since at each step we remove an object, at A^n there will be no objects, so P^n is the final random assignment
- Theorem:
  - P_ω is ordinally efficient for all profiles of eating functions.
  - Conversely, there exists a profile of eating functions for any ordinally efficient P

Probabilistic Serial Assignment

- Apply Simultaneous Eating Algorithm to profile of uniform eating speeds
  - All $\omega_i(t) = 1$ for all t in [0,1] and all agents i in N
- This makes $y^s(a)$ easy to compute at any step
- Has some nice properties

Probabilistic Serial Assignment

- Anonymous
- Only equitable mechanism
  - In order to construct an anonymous assignment, we will always end up with the Probabilistic Serial assignment

Fairness and Incentives of PS vs RP

- Random Priority may generate envy
- Probabilistic Serial may be manipulated
- Both only happen under limited conditions
- For small n:
  - n = 2, trivially RP and PS give the same results
  - n = 3, RP may generate envy and PS may be manipulated
  - n >= 4?

For n = 3

- RP
  - O-efficient
  - Strategy-proof
  - Treats equal utilities with equal random allocations
- PS
  - O-efficient
  - No envy
  - Weakly strategy-proof

For n >= 3

- Proposition:
- PS
    - Envy free
    - Weakly strategy-proof
- RP
    - Weakly envy free
    - Strategy-proof

Impossibility Result

- For n >= 4, there is no possible mechanism such that
  - It is O-efficient
  - It is strategyproof
  - Treats equal preferences equally
  - Proof is very long

Further caveats

- Note some assumptions
  - Same number of agents and objects
    - Models can be easily adjusted for either more agents than objects or more objects than agents
  - Objective Indifferences
    - Some pair of objects are the same to all agents
  - Subjective Indifferences
    - Some pair of objects are the same to some agents

n agents and m objects

- Both RP and PS still work
    - If there are more objects than agents, everything still holds if the bistochastic matrices loosen to allow the columns to sum to less than one
    - If there are more agents than objects, then rows sum to m/n and if the eating functions integrate to m/n instead of 1.
    - Can instead add the remainder of null objects, which are the same to all agents

Objective Indifferences

- The simultaneous eating theorem still holds since the choice is inconsequential
- This provides no issue with the current results

Subjective Indifferences

- Since the difference could be unimportant to some agent but not to others, an agent can’t be allowed to choose arbitrarily
- Best option seems to be eliciting more preferences from those agents
- Could be a subject of further research

Discussion Considerations

- Other caveats?
- How computable is
  - Probabilistic Serial
  - Random Priority