Designing Random Allocation Mechanisms: Theory and Applications

Eric Budish, Yeon-Koo Che, Fuhito Kojima, Paul Milgrom[1]

November 22, 2010

[1] University of Chicago, Columbia University, Stanford University

Overview

- Lotteries are common in resource allocation
  - School choice. (Abdulkadiroglu et al, 2005a, b)
  - House allocation. (Chen and Sonmez, 2002)
  - Organ transplantation. (Roth, Sonmez and Unver, 2004)
  - Office assignment. (Baccara et al, 2009)
  - Course allocation. (Budish and Cantillon, 2009)
- Deterministic allocations are unfair, when
  - goods are indivisible and
  - monetary transfers are limited.
- Randomizing allocations is necessary to restore ex-ante fairness

Overview

- Lotteries are common in resource allocation
    - School choice. (Abdulkadiroglu et al, 2005a, b)
    - House allocation. (Chen and Sonmez, 2002)
    - Organ transplantation. (Roth, Sonmez and Unver, 2004)
    - Office assignment. (Baccara et al, 2009)
    - Course allocation. (Budish and Cantillon, 2009)
- Deterministic allocations are unfair, when
    - goods are indivisible and
    - monetary transfers are limited.
- Randomizing allocations is necessary to restore ex-ante fairness

Overview

- Lotteries are common in resource allocation
  - School choice. (Abdulkadiroglu et al, 2005a, b)
  - House allocation. (Chen and Sonmez, 2002)
  - Organ transplantation. (Roth, Sonmez and Unver, 2004)
  - Office assignment. (Baccara et al, 2009)
  - Course allocation. (Budish and Cantillon, 2009)
- Deterministic allocations are unfair, when
  - goods are indivisible and
  - monetary transfers are limited.
- Randomizing allocations is necessary to restore ex-ante fairness

Overview

- Lotteries are common in resource allocation
  - School choice. (Abdulkadiroglu et al, 2005a, b)
  - House allocation. (Chen and Sonmez, 2002)
  - Organ transplantation. (Roth, Sonmez and Unver, 2004)
  - Office assignment. (Baccara et al, 2009)
  - Course allocation. (Budish and Cantillon, 2009)
- Deterministic allocations are unfair, when
  - goods are indivisible and
  - monetary transfers are limited.
- Randomizing allocations is necessary to restore ex-ante fairness

Designing random allocation mechanisms

- **A typical method:** (i) Select a set of ex post desirable allocations, and (ii) "randomize" among them: (e.g., Random serial dictatorship, Gale-Shapley DA, Top trading cycles with ties)
  $\Rightarrow$ entails ex ante inefficiencies.
- **Alternative method:** Choose directly "lotteries of goods" for the agents, called **random assignment**.
  - The Walrasian "pseudo-market" mechanism (Hylland and Zeckhauser 1979),
  - The probabilistic serial mechanism (Bogomolnaia and Moulin 2001).
- An issue: What random assignments are implementable? I.e., given a random assignment, is there always a lottery over sure outcomes that realizes it?

Designing random allocation mechanisms

- **A typical method:** (i) Select a set of ex post desirable allocations, and (ii) "randomize" among them: (e.g., Random serial dictatorship, Gale-Shapley DA, Top trading cycles with ties)

  $\Rightarrow$ entails ex ante inefficiencies.
- **Alternative method:** Choose directly "lotteries of goods" for the agents, called **random assignment**.
  - The Walrasian "pseudo-market" mechanism (Hylland and Zeckhauser 1979),
  - The probabilistic serial mechanism (Bogomolnaia and Moulin 2001).
- An issue: What random assignments are implementable? I.e., given a random assignment, is there always a lottery over sure outcomes that realizes it?

Designing random allocation mechanisms

- **A typical method:** (i) Select a set of ex post desirable allocations, and (ii) "randomize" among them: (e.g., Random serial dictatorship, Gale-Shapley DA, Top trading cycles with ties)
  $\Rightarrow$ entails ex ante inefficiencies.
- **Alternative method:** Choose directly "lotteries of goods" for the agents, called **random assignment**.
  - The Walrasian "pseudo-market" mechanism (Hylland and Zeckhauser 1979),
  - The probabilistic serial mechanism (Bogomolnaia and Moulin 2001).
- An issue: What random assignments are implementable? I.e., given a random assignment, is there always a lottery over sure outcomes that realizes it?

Designing random allocation mechanisms

- **A typical method:** (i) Select a set of ex post desirable allocations, and (ii) "randomize" among them: (e.g., Random serial dictatorship, Gale-Shapley DA, Top trading cycles with ties)
  $\Rightarrow$ entails ex ante inefficiencies.
- **Alternative method:** Choose directly "lotteries of goods" for the agents, called **random assignment**.
  - The Walrasian "pseudo-market" mechanism (Hylland and Zeckhauser 1979),
  - The probabilistic serial mechanism (Bogomolnaia and Moulin 2001).
- An issue: What random assignments are implementable? I.e., given a random assignment, is there always a lottery over sure outcomes that realizes it?

Designing random allocation mechanisms

- **A typical method:** (i) Select a set of ex post desirable allocations, and (ii) "randomize" among them: (e.g., Random serial dictatorship, Gale-Shapley DA, Top trading cycles with ties)
  $\Rightarrow$ entails ex ante inefficiencies.
- **Alternative method:** Choose directly "lotteries of goods" for the agents, called **random assignment**.
  - The Walrasian "pseudo-market" mechanism (Hylland and Zeckhauser 1979),
  - The probabilistic serial mechanism (Bogomolnaia and Moulin 2001).
- An issue: What random assignments are implementable? I.e., given a random assignment, is there always a lottery over sure outcomes that realizes it?

Designing random allocation mechanisms

- **A typical method:** (i) Select a set of ex post desirable allocations, and (ii) "randomize" among them: (e.g., Random serial dictatorship, Gale-Shapley DA, Top trading cycles with ties)
  $\Rightarrow$ entails ex ante inefficiencies.
- **Alternative method:** Choose directly "lotteries of goods" for the agents, called **random assignment**.
  - The Walrasian "pseudo-market" mechanism (Hylland and Zeckhauser 1979),
  - The probabilistic serial mechanism (Bogomolnaia and Moulin 2001).
- An issue: What random assignments are implementable? I.e., given a random assignment, is there always a lottery over sure outcomes that realizes it?

Designing random allocation mechanisms

- **A typical method:** (i) Select a set of ex post desirable allocations, and (ii) "randomize" among them: (e.g., Random serial dictatorship, Gale-Shapley DA, Top trading cycles with ties)
  $\Rightarrow$ entails ex ante inefficiencies.
- **Alternative method:** Choose directly "lotteries of goods" for the agents, called **random assignment**.
  - The Walrasian "pseudo-market" mechanism (Hylland and Zeckhauser 1979),
  - The probabilistic serial mechanism (Bogomolnaia and Moulin 2001).
- An issue: What random assignments are implementable? I.e., given a random assignment, is there always a lottery over sure outcomes that realizes it?

Designing random allocation mechanisms

- **A typical method:** (i) Select a set of ex post desirable allocations, and (ii) "randomize" among them: (e.g., Random serial dictatorship, Gale-Shapley DA, Top trading cycles with ties)
  $\Rightarrow$ entails ex ante inefficiencies.
- **Alternative method:** Choose directly "lotteries of goods" for the agents, called **random assignment**.
  - The Walrasian "pseudo-market" mechanism (Hylland and Zeckhauser 1979),
  - The probabilistic serial mechanism (Bogomolnaia and Moulin 2001).
- An issue: What random assignments are implementable? I.e., given a random assignment, is there always a lottery over sure outcomes that realizes it?

Illustration

- Implementing random assignments is nontrivial since assignments need to be "correlated." Consider assigning 3 goods a, b, c to 3 agents $1, 2, 3$, one for each. Can express an arbitrary random assignment in a matrix form:

$$\mathbf{P} = \begin{pmatrix} 0.5 & 0.5 & 0 \\ 0 & 0.5 & 0.5 \\ 0.5 & 0 & 0.5 \end{pmatrix} = \begin{pmatrix} & & \\ & & \\ & & \end{pmatrix}$$

- Birkhoff-von Neumann Theorem shows: For the one-to-one assignment problem, any random assignment can be implemented as a lottery over deterministic assignments. (More formally, any bistochastic matrix can be written as a convex combination of permutation matrices.)

Illustration

- Implementing random assignments is nontrivial since assignments need to be "correlated." Consider assigning 3 goods a, b, c to 3 agents $1, 2, 3$, one for each. Can express an arbitrary random assignment in a matrix form:

$$\mathbf{P} = \begin{pmatrix} 0.5 & 0.5 & 0 \\ 0 & 0.5 & 0.5 \\ 0.5 & 0 & 0.5 \end{pmatrix} = \begin{pmatrix} & & \\ & & \\ & & \end{pmatrix}$$

- Birkhoff-von Neumann Theorem shows: For the one-to-one assignment problem, any random assignment can be implemented as a lottery over deterministic assignments. (More formally, any bistochastic matrix can be written as a convex combination of permutation matrices.)

Illustration

- Implementing random assignments is nontrivial since assignments need to be "correlated." Consider assigning 3 goods a, b, c to 3 agents $1, 2, 3$, one for each. Can express an arbitrary random assignment in a matrix form:

$$\mathbf{P} = \begin{pmatrix} 0.5 & 0.5 & 0 \\ 0 & 0.5 & 0.5 \\ 0.5 & 0 & 0.5 \end{pmatrix} = \begin{pmatrix} 1 & & \\ & & \\ & & \end{pmatrix}$$

- Birkhoff-von Neumann Theorem shows: For the one-to-one assignment problem, any random assignment can be implemented as a lottery over deterministic assignments. (More formally, any bistochastic matrix can be written as a convex combination of permutation matrices.)

Illustration

- Implementing random assignments is nontrivial since assignments need to be "correlated." Consider assigning 3 goods a, b, c to 3 agents $1, 2, 3$, one for each. Can express an arbitrary random assignment in a matrix form:

$$\mathbf{P} = \begin{pmatrix} 0.5 & 0.5 & 0 \\ 0 & 0.5 & 0.5 \\ 0.5 & 0 & 0.5 \end{pmatrix} = \begin{pmatrix} 1 & 0 & \\ & & \\ & & \end{pmatrix}$$

- Birkhoff-von Neumann Theorem shows: For the one-to-one assignment problem, any random assignment can be implemented as a lottery over deterministic assignments. (More formally, any bistochastic matrix can be written as a convex combination of permutation matrices.)

Illustration

- Implementing random assignments is nontrivial since assignments need to be "correlated." Consider assigning 3 goods a, b, c to 3 agents $1, 2, 3$, one for each. Can express an arbitrary random assignment in a matrix form:

$$\mathbf{P} = \begin{pmatrix} 0.5 & 0.5 & 0 \\ 0 & 0.5 & 0.5 \\ 0.5 & 0 & 0.5 \end{pmatrix} = \begin{pmatrix} 1 & 0 & \\ & 1 & \\ & & \end{pmatrix}$$

- Birkhoff-von Neumann Theorem shows: For the one-to-one assignment problem, any random assignment can be implemented as a lottery over deterministic assignments. (More formally, any bistochastic matrix can be written as a convex combination of permutation matrices.)

Illustration

- Implementing random assignments is nontrivial since assignments need to be "correlated." Consider assigning 3 goods a, b, c to 3 agents $1, 2, 3$, one for each. Can express an arbitrary random assignment in a matrix form:

$$\mathbf{P} = \begin{pmatrix} 0.5 & 0.5 & 0 \\ 0 & 0.5 & 0.5 \\ 0.5 & 0 & 0.5 \end{pmatrix} = \begin{pmatrix} 1 & 0 & \\ & 1 & 0 \\ & & \end{pmatrix}$$

- Birkhoff-von Neumann Theorem shows: For the one-to-one assignment problem, any random assignment can be implemented as a lottery over deterministic assignments. (More formally, any bistochastic matrix can be written as a convex combination of permutation matrices.)

Illustration

- Implementing random assignments is nontrivial since assignments need to be "correlated." Consider assigning 3 goods a, b, c to 3 agents $1, 2, 3$, one for each. Can express an arbitrary random assignment in a matrix form:

$$\mathbf{P} = \begin{pmatrix} 0.5 & 0.5 & 0 \\ 0 & 0.5 & 0.5 \\ 0.5 & 0 & 0.5 \end{pmatrix} = \begin{pmatrix} 1 & 0 & \\ & 1 & 0 \\ & & 1 \end{pmatrix}$$

- Birkhoff-von Neumann Theorem shows: For the one-to-one assignment problem, any random assignment can be implemented as a lottery over deterministic assignments. (More formally, any bistochastic matrix can be written as a convex combination of permutation matrices.)

Illustration

- Implementing random assignments is nontrivial since assignments need to be "correlated." Consider assigning 3 goods a, b, c to 3 agents $1, 2, 3$, one for each. Can express an arbitrary random assignment in a matrix form:

$$\mathbf{P} = \begin{pmatrix} 0.5 & 0.5 & 0 \\ 0 & 0.5 & 0.5 \\ 0.5 & 0 & 0.5 \end{pmatrix} = \frac{1}{2}\begin{pmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{pmatrix}$$

- Birkhoff-von Neumann Theorem shows: For the one-to-one assignment problem, any random assignment can be implemented as a lottery over deterministic assignments. (More formally, any bistochastic matrix can be written as a convex combination of permutation matrices.)

Illustration

- Implementing random assignments is nontrivial since assignments need to be "correlated." Consider assigning 3 goods a, b, c to 3 agents $1, 2, 3$, one for each. Can express an arbitrary random assignment in a matrix form:

$$\mathbf{P} = \begin{pmatrix} 0.5 & 0.5 & 0 \\ 0 & 0.5 & 0.5 \\ 0.5 & 0 & 0.5 \end{pmatrix} = \frac{1}{2}\begin{pmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{pmatrix} + \frac{1}{2}\begin{pmatrix} 0 & 1 & 0 \\ 0 & 0 & 1 \\ 1 & 0 & 0 \end{pmatrix}.$$

- Birkhoff-von Neumann Theorem shows: For the one-to-one assignment problem, any random assignment can be implemented as a lottery over deterministic assignments. (More formally, any bistochastic matrix can be written as a convex combination of permutation matrices.)

Illustration

- Implementing random assignments is nontrivial since assignments need to be "correlated." Consider assigning 3 goods a, b, c to 3 agents $1, 2, 3$, one for each. Can express an arbitrary random assignment in a matrix form:

$$\mathbf{P} = \begin{pmatrix} 0.5 & 0.5 & 0 \\ 0 & 0.5 & 0.5 \\ 0.5 & 0 & 0.5 \end{pmatrix} = \frac{1}{2}\begin{pmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{pmatrix} + \frac{1}{2}\begin{pmatrix} 0 & 1 & 0 \\ 0 & 0 & 1 \\ 1 & 0 & 0 \end{pmatrix}.$$

- Birkhoff-von Neumann Theorem shows: For the one-to-one assignment problem, any random assignment can be implemented as a lottery over deterministic assignments. (More formally, any bistochastic matrix can be written as a convex combination of permutation matrices.)

Illustration

- Implementing random assignments is nontrivial since assignments need to be "correlated." Consider assigning 3 goods a, b, c to 3 agents $1, 2, 3$, one for each. Can express an arbitrary random assignment in a matrix form:

$$\mathbf{P} = \begin{pmatrix} 0.5 & 0.5 & 0 \\ 0 & 0.5 & 0.5 \\ 0.5 & 0 & 0.5 \end{pmatrix} = \frac{1}{2}\begin{pmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{pmatrix} + \frac{1}{2}\begin{pmatrix} 0 & 1 & 0 \\ 0 & 0 & 1 \\ 1 & 0 & 0 \end{pmatrix}.$$

- Birkhoff-von Neumann Theorem shows: For the one-to-one assignment problem, any random assignment can be implemented as a lottery over deterministic assignments. (More formally, any bistochastic matrix can be written as a convex combination of permutation matrices.)

Generalizing the RA method

The RA method (including HZ and BM) has been applied primarily to an one-to-one assignment problem. To gain practical applicability,

- the model need to be generalized to allow for many-to-one, many-to-many matchings, and unassignment.
- the method must be extended to accommodate a variety of constraints:
  - **Group-specific quota ("Controlled choice")**: School systems seek balance in student body based on race, ethnicity, gender, test scores (NYC, EdOpt), residence (Seoul).
    $\Rightarrow$ Sub-column constraint.
  - **Within agent constraint:** Scheduling and curriculum constraints in course allocation
    $\Rightarrow$ Sub-row constraint.
  - **Endogenous capacities:** Schools may run multiple programs the relative sizes of which are adjustable.
    $\Rightarrow$ Multi-column constraint

Generalizing the RA method

The RA method (including HZ and BM) has been applied primarily to an one-to-one assignment problem. To gain practical applicability,

- the model need to be generalized to allow for many-to-one, many-to-many matchings, and unassignment.
- the method must be extended to accommodate a variety of constraints:
  - **Group-specific quota ("Controlled choice")**: School systems seek balance in student body based on race, ethnicity, gender, test scores (NYC, EdOpt), residence (Seoul).
    $\Rightarrow$ Sub-column constraint.
  - **Within agent constraint:** Scheduling and curriculum constraints in course allocation
    $\Rightarrow$ Sub-row constraint.
  - **Endogenous capacities:** Schools may run multiple programs the relative sizes of which are adjustable.
    $\Rightarrow$ Multi-column constraint

Generalizing the RA method

The RA method (including HZ and BM) has been applied primarily to an one-to-one assignment problem. To gain practical applicability,

- the model need to be generalized to allow for many-to-one, many-to-many matchings, and unassignment.
- the method must be extended to accommodate a variety of constraints:
  - **Group-specific quota ("Controlled choice")**: School systems seek balance in student body based on race, ethnicity, gender, test scores (NYC, EdOpt), residence (Seoul).
    $\Rightarrow$ Sub-column constraint.
  - **Within agent constraint:** Scheduling and curriculum constraints in course allocation
    $\Rightarrow$ Sub-row constraint.
  - **Endogenous capacities:** Schools may run multiple programs the relative sizes of which are adjustable.
    $\Rightarrow$ Multi-column constraint

Generalizing the RA method

The RA method (including HZ and BM) has been applied primarily to an one-to-one assignment problem. To gain practical applicability,

- the model need to be generalized to allow for many-to-one, many-to-many matchings, and unassignment.
- the method must be extended to accommodate a variety of constraints:
  - **Group-specific quota ("Controlled choice")**: School systems seek balance in student body based on race, ethnicity, gender, test scores (NYC, EdOpt), residence (Seoul).
    $\Rightarrow$ Sub-column constraint.
  - **Within agent constraint:** Scheduling and curriculum constraints in course allocation
    $\Rightarrow$ Sub-row constraint.
  - **Endogenous capacities:** Schools may run multiple programs the relative sizes of which are adjustable.
    $\Rightarrow$ Multi-column constraint

Generalizing the RA method

The RA method (including HZ and BM) has been applied primarily to an one-to-one assignment problem. To gain practical applicability,

- the model need to be generalized to allow for many-to-one, many-to-many matchings, and unassignment.
- the method must be extended to accommodate a variety of constraints:
  - **Group-specific quota ("Controlled choice")**: School systems seek balance in student body based on race, ethnicity, gender, test scores (NYC, EdOpt), residence (Seoul).
    $\Rightarrow$ Sub-column constraint.
  - **Within agent constraint:** Scheduling and curriculum constraints in course allocation
    $\Rightarrow$ Sub-row constraint.
  - **Endogenous capacities:** Schools may run multiple programs the relative sizes of which are adjustable.
    $\Rightarrow$ Multi-column constraint

Generalizing the RA method

The RA method (including HZ and BM) has been applied primarily to an one-to-one assignment problem. To gain practical applicability,

- the model need to be generalized to allow for many-to-one, many-to-many matchings, and unassignment.
- the method must be extended to accommodate a variety of constraints:
  - **Group-specific quota ("Controlled choice")**: School systems seek balance in student body based on race, ethnicity, gender, test scores (NYC, EdOpt), residence (Seoul).
    $\Rightarrow$ Sub-column constraint.
  - **Within agent constraint:** Scheduling and curriculum constraints in course allocation
    $\Rightarrow$ Sub-row constraint.
  - **Endogenous capacities:** Schools may run multiple programs the relative sizes of which are adjustable.
    $\Rightarrow$ Multi-column constraint

Generalizing the RA method

The RA method (including HZ and BM) has been applied primarily to an one-to-one assignment problem. To gain practical applicability,

- the model need to be generalized to allow for many-to-one, many-to-many matchings, and unassignment.
- the method must be extended to accommodate a variety of constraints:
  - **Group-specific quota ("Controlled choice")**: School systems seek balance in student body based on race, ethnicity, gender, test scores (NYC, EdOpt), residence (Seoul).
    $\Rightarrow$ Sub-column constraint.
  - **Within agent constraint:** Scheduling and curriculum constraints in course allocation
    $\Rightarrow$ Sub-row constraint.
  - **Endogenous capacities:** Schools may run multiple programs the relative sizes of which are adjustable.
    $\Rightarrow$ Multi-column constraint

Generalizing the RA method

The RA method (including HZ and BM) has been applied primarily to an one-to-one assignment problem. To gain practical applicability,

- the model need to be generalized to allow for many-to-one, many-to-many matchings, and unassignment.
- the method must be extended to accommodate a variety of constraints:
    - **Group-specific quota ("Controlled choice")**: School systems seek balance in student body based on race, ethnicity, gender, test scores (NYC, EdOpt), residence (Seoul).
      $\Rightarrow$ Sub-column constraint.
    - **Within agent constraint:** Scheduling and curriculum constraints in course allocation
      $\Rightarrow$ Sub-row constraint.
    - **Endogenous capacities:** Schools may run multiple programs the relative sizes of which are adjustable.
      $\Rightarrow$ Multi-column constraint

Generalizing the RA method

The RA method (including HZ and BM) has been applied primarily to an one-to-one assignment problem. To gain practical applicability,

- the model need to be generalized to allow for many-to-one, many-to-many matchings, and unassignment.
- the method must be extended to accommodate a variety of constraints:
  - **Group-specific quota ("Controlled choice")**: School systems seek balance in student body based on race, ethnicity, gender, test scores (NYC, EdOpt), residence (Seoul).
    $\Rightarrow$ Sub-column constraint.
  - **Within agent constraint:** Scheduling and curriculum constraints in course allocation
    $\Rightarrow$ Sub-row constraint.
  - **Endogenous capacities:** Schools may run multiple programs the relative sizes of which are adjustable.
    $\Rightarrow$ Multi-column constraint

What we do

1. We generalize Birkhoff-von Neumann theorem for implementation of random assignments in general environment:
   - Identify a sufficient condition under which a random assignment can be implemented, called "bihierarchy"
   - Show that the sufficient condition is also necessary in bilateral matching
   - Develop a polynomial time algorithm for implementation
2. We extend the random assignment method to market-design applications
   - Generalize Bogomolnaia and Moulin's probabilistic serial mechanism for applications such as school choice
   - Generalize Hylland and Zeckhauser's pseudomarket mechanism for applications like course allocation
3. Find a way to improve ex post fairness in multi-unit assignment and two-sided matching

What we do

1. We generalize Birkhoff-von Neumann theorem for implementation of random assignments in general environment:
   - Identify a sufficient condition under which a random assignment can be implemented, called "bihierarchy"
   - Show that the sufficient condition is also necessary in bilateral matching
   - Develop a polynomial time algorithm for implementation
2. We extend the random assignment method to market-design applications
   - Generalize Bogomolnaia and Moulin's probabilistic serial mechanism for applications such as school choice
   - Generalize Hylland and Zeckhauser's pseudomarket mechanism for applications like course allocation
3. Find a way to improve ex post fairness in multi-unit assignment and two-sided matching

What we do

1. We generalize Birkhoff-von Neumann theorem for implementation of random assignments in general environment:
   - Identify a sufficient condition under which a random assignment can be implemented, called "bihierarchy"
   - Show that the sufficient condition is also necessary in bilateral matching
   - Develop a polynomial time algorithm for implementation
2. We extend the random assignment method to market-design applications
   - Generalize Bogomolnaia and Moulin's probabilistic serial mechanism for applications such as school choice
   - Generalize Hylland and Zeckhauser's pseudomarket mechanism for applications like course allocation
3. Find a way to improve ex post fairness in multi-unit assignment and two-sided matching

What we do

1. We generalize Birkhoff-von Neumann theorem for implementation of random assignments in general environment:
    - Identify a sufficient condition under which a random assignment can be implemented, called "bihierarchy"
    - Show that the sufficient condition is also necessary in bilateral matching
    - Develop a polynomial time algorithm for implementation
2. We extend the random assignment method to market-design applications
    - Generalize Bogomolnaia and Moulin's probabilistic serial mechanism for applications such as school choice
    - Generalize Hylland and Zeckhauser's pseudomarket mechanism for applications like course allocation
3. Find a way to improve ex post fairness in multi-unit assignment and two-sided matching

What we do

1. We generalize Birkhoff-von Neumann theorem for implementation of random assignments in general environment:
   - Identify a sufficient condition under which a random assignment can be implemented, called "bihierarchy"
   - Show that the sufficient condition is also necessary in bilateral matching
   - Develop a polynomial time algorithm for implementation
2. We extend the random assignment method to market-design applications
   - Generalize Bogomolnaia and Moulin's probabilistic serial mechanism for applications such as school choice
   - Generalize Hylland and Zeckhauser's pseudomarket mechanism for applications like course allocation
3. Find a way to improve ex post fairness in multi-unit assignment and two-sided matching

What we do

1. We generalize Birkhoff-von Neumann theorem for implementation of random assignments in general environment:
    - Identify a sufficient condition under which a random assignment can be implemented, called "bihierarchy"
    - Show that the sufficient condition is also necessary in bilateral matching
    - Develop a polynomial time algorithm for implementation
2. We extend the random assignment method to market-design applications
    - Generalize Bogomolnaia and Moulin's probabilistic serial mechanism for applications such as school choice
    - Generalize Hylland and Zeckhauser's pseudomarket mechanism for applications like course allocation
3. Find a way to improve ex post fairness in multi-unit assignment and two-sided matching

What we do

1. We generalize Birkhoff-von Neumann theorem for implementation of random assignments in general environment:
   - Identify a sufficient condition under which a random assignment can be implemented, called "bihierarchy"
   - Show that the sufficient condition is also necessary in bilateral matching
   - Develop a polynomial time algorithm for implementation
2. We extend the random assignment method to market-design applications
   - Generalize Bogomolnaia and Moulin's probabilistic serial mechanism for applications such as school choice
   - Generalize Hylland and Zeckhauser's pseudomarket mechanism for applications like course allocation
3. Find a way to improve ex post fairness in multi-unit assignment and two-sided matching

What we do

1. We generalize Birkhoff-von Neumann theorem for implementation of random assignments in general environment:
   - Identify a sufficient condition under which a random assignment can be implemented, called "bihierarchy"
   - Show that the sufficient condition is also necessary in bilateral matching
   - Develop a polynomial time algorithm for implementation
2. We extend the random assignment method to market-design applications
   - Generalize Bogomolnaia and Moulin's probabilistic serial mechanism for applications such as school choice
   - Generalize Hylland and Zeckhauser's pseudomarket mechanism for applications like course allocation
3. Find a way to improve ex post fairness in multi-unit assignment and two-sided matching

Model

- N, O are the sets of agents and goods,
- A (generalized) random assignment is a matrix $P = (P_{ia}) \in \mathbb{R}^{|N| \times |O|}$.
- $\mathcal{H} \subset 2^{N \times O}$ is a collection of subsets of $N \times O$, called a **constraint structure**.
- Integers $\underline{q}_S \leq \overline{q}_S$ for each $S \in \mathcal{H}$.
  - Each set $S \in \mathcal{H}$ is understood to be a "constraint set," that is, a set of elements on which a constraint is imposed. $\underline{q}_S$ and $\overline{q}_S$ are floor and ceiling (minimum and maximum) constraints, respectively. That is, we will consider random assignment P satisfying

$$\underline{q}_S \leq \sum_{(i,a) \in S} P_{ia} \leq \overline{q}_S,$$

for each $S \in \mathcal{H}$.

Model

- N, O are the sets of agents and goods,
- A (generalized) random assignment is a matrix $P = (P_{ia}) \in \mathbb{R}^{|N| \times |O|}$.
- $\mathcal{H} \subset 2^{N \times O}$ is a collection of subsets of $N \times O$, called a **constraint structure**.
- Integers $\underline{q}_S \leq \overline{q}_S$ for each $S \in \mathcal{H}$.
  - Each set $S \in \mathcal{H}$ is understood to be a "constraint set," that is, a set of elements on which a constraint is imposed. $\underline{q}_S$ and $\overline{q}_S$ are floor and ceiling (minimum and maximum) constraints, respectively. That is, we will consider random assignment P satisfying

$$\underline{q}_S \leq \sum_{(i,a) \in S} P_{ia} \leq \overline{q}_S,$$

for each $S \in \mathcal{H}$.

Model

- N, O are the sets of agents and goods,
- A (generalized) random assignment is a matrix $P = (P_{ia}) \in \mathbb{R}^{|N| \times |O|}$.
- $\mathcal{H} \subset 2^{N \times O}$ is a collection of subsets of $N \times O$, called a **constraint structure**.
- Integers $\underline{q}_S \leq \overline{q}_S$ for each $S \in \mathcal{H}$.
  - Each set $S \in \mathcal{H}$ is understood to be a "constraint set," that is, a set of elements on which a constraint is imposed. $\underline{q}_S$ and $\overline{q}_S$ are floor and ceiling (minimum and maximum) constraints, respectively. That is, we will consider random assignment P satisfying

$$\underline{q}_S \leq \sum_{(i,a) \in S} P_{ia} \leq \overline{q}_S,$$

for each $S \in \mathcal{H}$.

Model

- N, O are the sets of agents and goods,
- A (generalized) random assignment is a matrix $P = (P_{ia}) \in \mathbb{R}^{|N| \times |O|}$.
- $\mathcal{H} \subset 2^{N \times O}$ is a collection of subsets of $N \times O$, called a **constraint structure**.
- Integers $\underline{q}_S \leq \overline{q}_S$ for each $S \in \mathcal{H}$.
  - Each set $S \in \mathcal{H}$ is understood to be a "constraint set," that is, a set of elements on which a constraint is imposed. $\underline{q}_S$ and $\overline{q}_S$ are floor and ceiling (minimum and maximum) constraints, respectively. That is, we will consider random assignment P satisfying

$$\underline{q}_S \leq \sum_{(i,a) \in S} P_{ia} \leq \overline{q}_S,$$

for each $S \in \mathcal{H}$.

Model

- N, O are the sets of agents and goods,
- A (generalized) random assignment is a matrix $P = (P_{ia}) \in \mathbb{R}^{|N| \times |O|}$.
- $\mathcal{H} \subset 2^{N \times O}$ is a collection of subsets of $N \times O$, called a **constraint structure**.
- Integers $\underline{q}_S \leq \overline{q}_S$ for each $S \in \mathcal{H}$.
  - Each set $S \in \mathcal{H}$ is understood to be a "constraint set," that is, a set of elements on which a constraint is imposed. $\underline{q}_S$ and $\overline{q}_S$ are floor and ceiling (minimum and maximum) constraints, respectively. That is, we will consider random assignment P satisfying

$$\underline{q}_S \leq \sum_{(i,a) \in S} P_{ia} \leq \overline{q}_S,$$

for each $S \in \mathcal{H}$.

Model

- N, O are the sets of agents and goods,
- A (generalized) random assignment is a matrix $P = (P_{ia}) \in \mathbb{R}^{|N|\times|O|}$.
- $\mathcal{H} \subset 2^{N\times O}$ is a collection of subsets of $N \times O$, called a **constraint structure**.
- Integers $\underline{q}_S \leq \overline{q}_S$ for each $S \in \mathcal{H}$.
  - Each set $S \in \mathcal{H}$ is understood to be a "constraint set," that is, a set of elements on which a constraint is imposed. $\underline{q}_S$ and $\overline{q}_S$ are floor and ceiling (minimum and maximum) constraints, respectively. That is, we will consider random assignment P satisfying

$$\underline{q}_S \leq \sum_{(i,a)\in S} P_{ia} \leq \overline{q}_S,$$

for each $S \in \mathcal{H}$.

Model

- N, O are the sets of agents and goods,
- A (generalized) random assignment is a matrix $P = (P_{ia}) \in \mathbb{R}^{|N| \times |O|}$.
- $\mathcal{H} \subset 2^{N \times O}$ is a collection of subsets of $N \times O$, called a **constraint structure**.
- Integers $\underline{q}_S \leq \overline{q}_S$ for each $S \in \mathcal{H}$.
  - Each set $S \in \mathcal{H}$ is understood to be a "constraint set," that is, a set of elements on which a constraint is imposed. $\underline{q}_S$ and $\overline{q}_S$ are floor and ceiling (minimum and maximum) constraints, respectively. That is, we will consider random assignment P satisfying

$$\underline{q}_S \leq \sum_{(i,a) \in S} P_{ia} \leq \overline{q}_S,$$

for each $S \in \mathcal{H}$.

Decomposability

- Constraint structure $\mathcal{H}$ is **universally decomposable** if, for each $(\underline{q}_S, \overline{q}_S)_{S\in\mathcal{H}}$ and P with $\underline{q}_S \leq \sum_{(i,a)\in S} P_{ia} \leq \overline{q}_S$ for all $S \in \mathcal{H}$, there exists a convex decomposition

$$P = \sum_{k=1}^{K} \lambda^k P^k, \text{ such that}$$

  1. each P^k is integer-valued, and
  2. $\underline{q}_S \leq \sum_{(i,a)\in S} P^k_{ia} \leq \overline{q}_S$, for each k and $S \in \mathcal{H}$.

- Decomposability means "Every P satisfying all the given constraints in $\mathcal{H}$ can be expressed as a convex combination of integral matrices satisfying the constraints." In other words, any random assignment satisfying constraints in $\mathcal{H}$ can be implemented as a lottery over deterministic assignments that satisfy constraints in $\mathcal{H}$.

Decomposability

- Constraint structure $\mathcal{H}$ is **universally decomposable** if, for each $(\underline{q}_S, \overline{q}_S)_{S \in \mathcal{H}}$ and P with $\underline{q}_S \leq \sum_{(i,a) \in S} P_{ia} \leq \overline{q}_S$ for all $S \in \mathcal{H}$, there exists a convex decomposition

$$P = \sum_{k=1}^{K} \lambda^k P^k, \text{ such that}$$

1. each P^k is integer-valued, and
2. $\underline{q}_S \leq \sum_{(i,a) \in S} P^k_{ia} \leq \overline{q}_S$, for each k and $S \in \mathcal{H}$.

- Decomposability means "Every P satisfying all the given constraints in $\mathcal{H}$ can be expressed as a convex combination of integral matrices satisfying the constraints." In other words, any random assignment satisfying constraints in $\mathcal{H}$ can be implemented as a lottery over deterministic assignments that satisfy constraints in $\mathcal{H}$.

Decomposability

- Constraint structure $\mathcal{H}$ is **universally decomposable** if, for each $(\underline{q}_S, \overline{q}_S)_{S\in\mathcal{H}}$ and P with $\underline{q}_S \leq \sum_{(i,a)\in S} P_{ia} \leq \overline{q}_S$ for all $S \in \mathcal{H}$, there exists a convex decomposition

$$P = \sum_{k=1}^{K} \lambda^k P^k, \text{ such that}$$

1. each P^k is integer-valued, and
2. $\underline{q}_S \leq \sum_{(i,a)\in S} P^k_{ia} \leq \overline{q}_S$, for each k and $S \in \mathcal{H}$.

- Decomposability means "Every P satisfying all the given constraints in $\mathcal{H}$ can be expressed as a convex combination of integral matrices satisfying the constraints." In other words, any random assignment satisfying constraints in $\mathcal{H}$ can be implemented as a lottery over deterministic assignments that satisfy constraints in $\mathcal{H}$.

Decomposability

- Constraint structure $\mathcal{H}$ is **universally decomposable** if, for each $(\underline{q}_S, \overline{q}_S)_{S\in\mathcal{H}}$ and P with $\underline{q}_S \leq \sum_{(i,a)\in S} P_{ia} \leq \overline{q}_S$ for all $S \in \mathcal{H}$, there exists a convex decomposition

$$P = \sum_{k=1}^{K} \lambda^k P^k, \text{ such that}$$

  1. each P^k is integer-valued, and
  2. $\underline{q}_S \leq \sum_{(i,a)\in S} P^k_{ia} \leq \overline{q}_S$, for each k and $S \in \mathcal{H}$.

- Decomposability means "Every P satisfying all the given constraints in $\mathcal{H}$ can be expressed as a convex combination of integral matrices satisfying the constraints." In other words, any random assignment satisfying constraints in $\mathcal{H}$ can be implemented as a lottery over deterministic assignments that satisfy constraints in $\mathcal{H}$.

Decomposability

- Constraint structure $\mathcal{H}$ is **universally decomposable** if, for each $(\underline{q}_S, \overline{q}_S)_{S\in\mathcal{H}}$ and P with $\underline{q}_S \leq \sum_{(i,a)\in S} P_{ia} \leq \overline{q}_S$ for all $S \in \mathcal{H}$, there exists a convex decomposition

$$P = \sum_{k=1}^{K} \lambda^k P^k, \text{ such that}$$

  1. each P^k is integer-valued, and
  2. $\underline{q}_S \leq \sum_{(i,a)\in S} P^k_{ia} \leq \overline{q}_S$, for each k and $S \in \mathcal{H}$.

- Decomposability means "Every P satisfying all the given constraints in $\mathcal{H}$ can be expressed as a convex combination of integral matrices satisfying the constraints." In other words, any random assignment satisfying constraints in $\mathcal{H}$ can be implemented as a lottery over deterministic assignments that satisfy constraints in $\mathcal{H}$.

Decomposability

- Constraint structure $\mathcal{H}$ is **universally decomposable** if, for each $(\underline{q}_S, \overline{q}_S)_{S \in \mathcal{H}}$ and P with $\underline{q}_S \leq \sum_{(i,a) \in S} P_{ia} \leq \overline{q}_S$ for all $S \in \mathcal{H}$, there exists a convex decomposition

$$P = \sum_{k=1}^{K} \lambda^k P^k, \text{ such that}$$

  1. each P^k is integer-valued, and
  2. $\underline{q}_S \leq \sum_{(i,a) \in S} P^k_{ia} \leq \overline{q}_S$, for each k and $S \in \mathcal{H}$.

- Decomposability means "Every P satisfying all the given constraints in $\mathcal{H}$ can be expressed as a convex combination of integral matrices satisfying the constraints." In other words, any random assignment satisfying constraints in $\mathcal{H}$ can be implemented as a lottery over deterministic assignments that satisfy constraints in $\mathcal{H}$.

Decomposability

- Constraint structure $\mathcal{H}$ is **universally decomposable** if, for each $(\underline{q}_S, \overline{q}_S)_{S\in\mathcal{H}}$ and P with $\underline{q}_S \leq \sum_{(i,a)\in S} P_{ia} \leq \overline{q}_S$ for all $S \in \mathcal{H}$, there exists a convex decomposition

$$P = \sum_{k=1}^{K} \lambda^k P^k, \text{ such that}$$

  1. each P^k is integer-valued, and
  2. $\underline{q}_S \leq \sum_{(i,a)\in S} P^k_{ia} \leq \overline{q}_S$, for each k and $S \in \mathcal{H}$.

- Decomposability means "Every P satisfying all the given constraints in $\mathcal{H}$ can be expressed as a convex combination of integral matrices satisfying the constraints." In other words, any random assignment satisfying constraints in $\mathcal{H}$ can be implemented as a lottery over deterministic assignments that satisfy constraints in $\mathcal{H}$.

Decomposability

- Constraint structure $\mathcal{H}$ is **universally decomposable** if, for each $(\underline{q}_S, \overline{q}_S)_{S\in\mathcal{H}}$ and P with $\underline{q}_S \leq \sum_{(i,a)\in S} P_{ia} \leq \overline{q}_S$ for all $S \in \mathcal{H}$, there exists a convex decomposition

$$P = \sum_{k=1}^{K} \lambda^k P^k, \text{ such that}$$

  1. each P^k is integer-valued, and
  2. $\underline{q}_S \leq \sum_{(i,a)\in S} P^k_{ia} \leq \overline{q}_S$, for each k and $S \in \mathcal{H}$.
- Decomposability means "Every P satisfying all the given constraints in $\mathcal{H}$ can be expressed as a convex combination of integral matrices satisfying the constraints." In other words, any random assignment satisfying constraints in $\mathcal{H}$ can be implemented as a lottery over deterministic assignments that satisfy constraints in $\mathcal{H}$.

Hierarchy

- What property of the constraint structure $\mathcal{H}$ enables decomposability?
- $\mathcal{H} \subseteq 2^{N \times O}$ is a **hierarchy** if $S \cap S' = \emptyset$ or $S \subset S'$ or $S' \subset S$ for any $S, S' \in \mathcal{H}$.

$$\mathbf{P} = \begin{pmatrix} P_{1a} & P_{1b} & P_{1c} \\ P_{2a} & P_{2b} & P_{2c} \\ P_{3a} & P_{3b} & P_{3c} \end{pmatrix}$$

Hierarchy

- What property of the constraint structure $\mathcal{H}$ enables decomposability?
- $\mathcal{H} \subseteq 2^{N \times O}$ is a **hierarchy** if $S \cap S' = \emptyset$ or $S \subset S'$ or $S' \subset S$ for any $S, S' \in \mathcal{H}$.

$$\mathbf{P} = \begin{pmatrix} P_{1a} & P_{1b} & P_{1c} \\ P_{2a} & P_{2b} & P_{2c} \\ P_{3a} & P_{3b} & P_{3c} \end{pmatrix}$$

Not a Hierarchy

Hierarchy

- What property of the constraint structure $\mathcal{H}$ enables decomposability?
- $\mathcal{H} \subseteq 2^{N \times O}$ is a **hierarchy** if $S \cap S' = \emptyset$ or $S \subset S'$ or $S' \subset S$ for any $S, S' \in \mathcal{H}$.

$$\mathbf{P} = \begin{pmatrix} P_{1a} & P_{1b} & P_{1c} \\ P_{2a} & P_{2b} & P_{2c} \\ P_{3a} & P_{3b} & P_{3c} \end{pmatrix}$$

Hierarchy Not a Hierarchy

Hierarchy

- What property of the constraint structure $\mathcal{H}$ enables decomposability?
- $\mathcal{H} \subseteq 2^{N \times O}$ is a **hierarchy** if $S \cap S' = \emptyset$ or $S \subset S'$ or $S' \subset S$ for any $S, S' \in \mathcal{H}$.

$$\mathbf{P} = \begin{pmatrix} \boxed{P_{1a} \quad P_{1b} \quad P_{1c}} \\ P_{2a} \quad P_{2b} \quad P_{2c} \\ P_{3a} \quad P_{3b} \quad P_{3c} \end{pmatrix}$$

Hierarchy Not a Hierarchy

Hierarchy

- What property of the constraint structure $\mathcal{H}$ enables decomposability?
- $\mathcal{H} \subseteq 2^{N \times O}$ is a **hierarchy** if $S \cap S' = \emptyset$ or $S \subset S'$ or $S' \subset S$ for any $S, S' \in \mathcal{H}$.

$$\mathbf{P} = \begin{pmatrix} \boxed{\boxed{P_{1a} \quad P_{1b}} \quad P_{1c}} \\ P_{2a} \quad P_{2b} \quad P_{2c} \\ P_{3a} \quad P_{3b} \quad P_{3c} \end{pmatrix}$$

Hierarchy Not a Hierarchy

Hierarchy

- What property of the constraint structure $\mathcal{H}$ enables decomposability?
- $\mathcal{H} \subseteq 2^{N \times O}$ is a **hierarchy** if $S \cap S' = \emptyset$ or $S \subset S'$ or $S' \subset S$ for any $S, S' \in \mathcal{H}$.

$$\mathbf{P} = \begin{pmatrix} \boxed{\boxed{P_{1a} \quad P_{1b}} \quad P_{1c}} \\ \boxed{P_{2a} \quad P_{2b} \quad P_{2c}} \\ P_{3a} \quad P_{3b} \quad P_{3c} \end{pmatrix}$$

Hierarchy Not a Hierarchy

Hierarchy

- What property of the constraint structure $\mathcal{H}$ enables decomposability?
- $\mathcal{H} \subseteq 2^{N \times O}$ is a **hierarchy** if $S \cap S' = \emptyset$ or $S \subset S'$ or $S' \subset S$ for any $S, S' \in \mathcal{H}$.

$$\mathbf{P} = \begin{pmatrix} P_{1a} & P_{1b} & P_{1c} \\ P_{2a} & P_{2b} & P_{2c} \\ P_{3a} & P_{3b} & P_{3c} \end{pmatrix}$$

Hierarchy Not a Hierarchy

Hierarchy

- What property of the constraint structure $\mathcal{H}$ enables decomposability?
- $\mathcal{H} \subseteq 2^{N \times O}$ is a **hierarchy** if $S \cap S' = \emptyset$ or $S \subset S'$ or $S' \subset S$ for any $S, S' \in \mathcal{H}$.

$$\mathbf{P} = \begin{pmatrix} P_{1a} & P_{1b} & P_{1c} \\ P_{2a} & P_{2b} & P_{2c} \\ P_{3a} & P_{3b} & P_{3c} \end{pmatrix}$$

Hierarchy Not a Hierarchy

Decomposition Theorem

- $\mathcal{H} \subseteq 2^{N \times O}$ is a **bihierarchy** if it can be partitioned into two hierarchies.

Theorem

If $\mathcal{H}$ forms a bihierarchy, then it is universally decomposable.

- Proof Sketch: Recognize that the set of feasible random assignments $\{P : \underline{q}_S \leq \sum_{(i,a) \in S} P_{ia} \leq \overline{q}_S$, for each $S \in \mathcal{H}\}$ forms a convex polyhedron. Any random assignment is thus a convex combination of extreme points. Suffices to show that the extreme points are integer-valued. This result follow from Hoffman and Kruskal (1956) and Edmonds (1970).
- More important is "constructive algorithm" that works fast. We provide one based on a network flow method.

Decomposition Theorem

- $\mathcal{H} \subseteq 2^{N \times O}$ is a **bihierarchy** if it can be partitioned into two hierarchies.

Theorem

If $\mathcal{H}$ forms a bihierarchy, then it is universally decomposable.

- Proof Sketch: Recognize that the set of feasible random assignments $\{P : \underline{q}_S \leq \sum_{(i,a) \in S} P_{ia} \leq \overline{q}_S$, for each $S \in \mathcal{H}\}$ forms a convex polyhedron. Any random assignment is thus a convex combination of extreme points. Suffices to show that the extreme points are integer-valued. This result follow from Hoffman and Kruskal (1956) and Edmonds (1970).
- More important is "constructive algorithm" that works fast. We provide one based on a network flow method.

Decomposition Theorem

- $\mathcal{H} \subseteq 2^{N \times O}$ is a **bihierarchy** if it can be partitioned into two hierarchies.

Theorem

If $\mathcal{H}$ forms a bihierarchy, then it is universally decomposable.

- Proof Sketch: Recognize that the set of feasible random assignments $\{P : \underline{q}_S \leq \sum_{(i,a)\in S} P_{ia} \leq \overline{q}_S$, for each $S \in \mathcal{H}\}$ forms a convex polyhedron. Any random assignment is thus a convex combination of extreme points. Suffices to show that the extreme points are integer-valued. This result follow from Hoffman and Kruskal (1956) and Edmonds (1970).
- More important is "constructive algorithm" that works fast. We provide one based on a network flow method.

Decomposition Theorem

- $\mathcal{H} \subseteq 2^{N \times O}$ is a **bihierarchy** if it can be partitioned into two hierarchies.

Theorem

If $\mathcal{H}$ forms a bihierarchy, then it is universally decomposable.

- Proof Sketch: Recognize that the set of feasible random assignments $\{P : \underline{q}_S \leq \sum_{(i,a)\in S} P_{ia} \leq \overline{q}_S$, for each $S \in \mathcal{H}\}$ forms a convex polyhedron. Any random assignment is thus a convex combination of extreme points. Suffices to show that the extreme points are integer-valued. This result follow from Hoffman and Kruskal (1956) and Edmonds (1970).
- More important is "constructive algorithm" that works fast. We provide one based on a network flow method.

- What can go wrong without bihierarchy?
  - 2 goods and 2 agents, $\mathcal{H} = \{\{(1,a),(1,b)\},\{(1,a),(2,a)\},\{(1,b),(2,a)\}\}$, with each constraint being one.

$$\mathbf{P} = \begin{pmatrix} 0.5 & 0.5 \\ 0.5 & 0.5 \end{pmatrix} =? \begin{pmatrix} 1 & 0 \\ 1 & \end{pmatrix}$$

Lemma

If $\mathcal{H}$ has an odd cycle of intersecting sets, then $\mathcal{H}$ is not universally decomposable.

- What can go wrong without bihierarchy?
  - 2 goods and 2 agents,

    $\mathcal{H} = \{\{(1,a),(1,b)\},\{(1,a),(2,a)\},\{(1,b),(2,a)\}\}$, with each constraint being one.

$$\mathbf{P} = \begin{pmatrix} 0.5 & 0.5 \\ 0.5 & 0.5 \end{pmatrix} =? \begin{pmatrix} 1 & 0 \\ 1 & \end{pmatrix}$$

Lemma

If $\mathcal{H}$ has an odd cycle of intersecting sets, then $\mathcal{H}$ is not universally decomposable.

- What can go wrong without bihierarchy?
  - 2 goods and 2 agents, $\mathcal{H} = \{\{(1,a),(1,b)\},\{(1,a),(2,a)\},\{(1,b),(2,a)\}\}$, with each constraint being one.

$$\mathbf{P} = \begin{pmatrix} 0.5 & 0.5 \\ 0.5 & 0.5 \end{pmatrix} =? \begin{pmatrix} 1 & 0 \\ 1 & \end{pmatrix}$$

Lemma

If $\mathcal{H}$ has an odd cycle of intersecting sets, then $\mathcal{H}$ is not universally decomposable.

- What can go wrong without bihierarchy?
  - 2 goods and 2 agents,

    $\mathcal{H} = \{\{(1,a),(1,b)\},\{(1,a),(2,a)\},\{(1,b),(2,a)\}\}$, with each constraint being one.

$$\mathbf{P} = \begin{pmatrix} 0.5 & 0.5 \\ 0.5 & 0.5 \end{pmatrix} =? \begin{pmatrix} 1 & 0 \\ 1 & \end{pmatrix}$$

Lemma

If $\mathcal{H}$ has an odd cycle of intersecting sets, then $\mathcal{H}$ is not universally decomposable.

- What can go wrong without bihierarchy?
  - 2 goods and 2 agents,

    $\mathcal{H} = \{\{(1,a),(1,b)\},\{(1,a),(2,a)\},\{(1,b),(2,a)\}\}$, with each constraint being one.

$$\mathbf{P} = \begin{pmatrix} 0.5 & 0.5 \\ 0.5 & 0.5 \end{pmatrix} =? \begin{pmatrix} 1 & 0 \\ 1 & \end{pmatrix}$$

Lemma

If $\mathcal{H}$ has an odd cycle of intersecting sets, then $\mathcal{H}$ is not universally decomposable.

- What can go wrong without bihierarchy?
  - 2 goods and 2 agents,

    $\mathcal{H} = \{\{(1,a),(1,b)\},\{(1,a),(2,a)\},\{(1,b),(2,a)\}\}$, with each constraint being one.

$$\mathbf{P} = \begin{pmatrix} 0.5 & 0.5 \\ 0.5 & 0.5 \end{pmatrix} =? \begin{pmatrix} 1 & 0 \\ 1 & \end{pmatrix}$$

Lemma

If $\mathcal{H}$ has an odd cycle of intersecting sets, then $\mathcal{H}$ is not universally decomposable.

- What can go wrong without bihierarchy?
  - 2 goods and 2 agents,

    $\mathcal{H} = \{\{(1,a),(1,b)\},\{(1,a),(2,a)\},\{(1,b),(2,a)\}\}$, with each constraint being one.

$$\mathbf{P} = \begin{pmatrix} 0.5 & 0.5 \\ 0.5 & 0.5 \end{pmatrix} =? \begin{pmatrix} 1 & 0 \\ 1 & \end{pmatrix}$$

Lemma

If $\mathcal{H}$ has an odd cycle of intersecting sets, then $\mathcal{H}$ is not universally decomposable.

- What can go wrong without bihierarchy?
    - 2 goods and 2 agents,

      $\mathcal{H} = \{\{(1,a),(1,b)\},\{(1,a),(2,a)\},\{(1,b),(2,a)\}\}$, with each constraint being one.

$$\mathbf{P} = \begin{pmatrix} 0.5 & 0.5 \\ 0.5 & 0.5 \end{pmatrix} =? \begin{pmatrix} 1 & 0 \\ 1 & \end{pmatrix}$$

Lemma

If $\mathcal{H}$ has an odd cycle of intersecting sets, then $\mathcal{H}$ is not universally decomposable.

Necessity of Bihierarchy

Not generally but in a natural bilateral matching setting.

Theorem: Maximal domain

Suppose $\mathcal{H}$ contains all "rows" ($\{i\} \times O, \forall i \in N$) and all "columns" ($N \times \{a\}, \forall a \in O$). If $\mathcal{H}$ is not bihierarchical, then $\mathcal{H}$ is not universally decomposable.

In many applications, row and column constraints are present. If this is the case, a bihierarchical structure is necessary for BvN decomposition.

Necessity of Bihierarchy

Not generally but in a natural bilateral matching setting.

Theorem: Maximal domain

Suppose $\mathcal{H}$ contains all "rows" ($\{i\} \times O, \forall i \in N$) and all "columns" ($N \times \{a\}, \forall a \in O$). If $\mathcal{H}$ is not bihierarchical, then $\mathcal{H}$ is not universally decomposable.

In many applications, row and column constraints are present. If this is the case, a bihierarchical structure is necessary for BvN decomposition.

Necessity of Bihierarchy

Not generally but in a natural bilateral matching setting.

Theorem: Maximal domain

Suppose $\mathcal{H}$ contains all "rows" ($\{i\} \times O, \forall i \in N$) and all "columns" ($N \times \{a\}, \forall a \in O$). If $\mathcal{H}$ is not bihierarchical, then $\mathcal{H}$ is not universally decomposable.

In many applications, row and column constraints are present. If this is the case, a bihierarchical structure is necessary for BvN decomposition.

Example of a bihierarchy: Classical One to One Assignment

$$\mathbf{P} = \begin{pmatrix} P_{1a} & P_{1b} & P_{1c} \\ P_{2a} & P_{2b} & P_{2c} \\ P_{3a} & P_{3b} & P_{3c} \end{pmatrix}$$

The Birkhoff-von Neumann Theorem is a corollary of the Theorem.

Example of a bihierarchy: Classical One to One Assignment

$$\mathbf{P} = \begin{pmatrix} P_{1a} & P_{1b} & P_{1c} \\ P_{2a} & P_{2b} & P_{2c} \\ P_{3a} & P_{3b} & P_{3c} \end{pmatrix}$$

The Birkhoff-von Neumann Theorem is a corollary of the Theorem.

Example of a bihierarchy: Classical One to One Assignment

$$\mathbf{P} = \begin{pmatrix} P_{1a} & P_{1b} & P_{1c} \\ P_{2a} & P_{2b} & P_{2c} \\ P_{3a} & P_{3b} & P_{3c} \end{pmatrix}$$

The Birkhoff-von Neumann Theorem is a corollary of the Theorem.

Example of a bihierarchy: Flexible Capacities

Suppose a and b are two programs within a school; each program has maximum capacity of 2, and the school has maximum capacity of 3.

$$\mathbf{P} = \begin{pmatrix} P_{1a} & P_{1b} & P_{1c} \\ P_{2a} & P_{2b} & P_{2c} \\ P_{3a} & P_{3b} & P_{3c} \end{pmatrix}$$

Example of a bihierarchy: Flexible Capacities

Suppose a and b are two programs within a school; each program has maximum capacity of 2, and the school has maximum capacity of 3.

$$\mathbf{P} = \begin{pmatrix} \boxed{\begin{matrix} P_{1a} \\ P_{2a} \\ P_{3a} \end{matrix}} & \boxed{\begin{matrix} P_{1b} \\ P_{2b} \\ P_{3b} \end{matrix}} & \begin{matrix} P_{1c} \\ P_{2c} \\ P_{3c} \end{matrix} \end{pmatrix}$$

Example of a bihierarchy: Flexible Capacities

Suppose a and b are two programs within a school; each program has maximum capacity of 2, and the school has maximum capacity of 3.

$$\mathbf{P} = \begin{pmatrix} P_{1a} & P_{1b} & P_{1c} \\ P_{2a} & P_{2b} & P_{2c} \\ P_{3a} & P_{3b} & P_{3c} \end{pmatrix}$$

Example of a bihierarchy: Group Specific Quota

Suppose students 1 and 2 are ethnic majority, and 2 and 3 are male. If school a has a limit on ethnic majority while school b has a limit on male,

$$\mathbf{P} = \begin{pmatrix} P_{1a} & P_{1b} & P_{1c} \\ P_{2a} & P_{2b} & P_{2c} \\ P_{3a} & P_{3b} & P_{3c} \end{pmatrix}$$

Example of a bihierarchy: Group Specific Quota

Suppose students 1 and 2 are ethnic majority, and 2 and 3 are male. If school a has a limit on ethnic majority while school b has a limit on male,

$$\mathbf{P} = \begin{pmatrix} P_{1a} & P_{1b} & P_{1c} \\ P_{2a} & P_{2b} & P_{2c} \\ P_{3a} & P_{3b} & P_{3c} \end{pmatrix}$$

Example of a bihierarchy: Group Specific Quota

Suppose students 1 and 2 are ethnic majority, and 2 and 3 are male. If school a has a limit on ethnic majority while school b has a limit on male,

$$\mathbf{P} = \begin{pmatrix} P_{1a} & P_{1b} & P_{1c} \\ P_{2a} & P_{2b} & P_{2c} \\ P_{3a} & P_{3b} & P_{3c} \end{pmatrix}$$

Example of a bihierarchy: Group Specific Quota

Suppose students 1 and 2 are ethnic majority, and 2 and 3 are male. If school a has a limit on ethnic majority while school b has a limit on male,

$$\mathbf{P} = \begin{pmatrix} P_{1a} & P_{1b} & P_{1c} \\ P_{2a} & P_{2b} & P_{2c} \\ P_{3a} & P_{3b} & P_{3c} \end{pmatrix}$$

Application: Single-Unit Assignment

- Social planner needs to assign at most one object to each agent (e.g., school choice, housing allocation).
- Each agent has strict preferences over O.
- Some additional constraints are allowed; affirmative action constraints, flexible capacity, etc.
- Suppose constraint sets $\mathcal{H}$ form a bihierchy.
  - $\mathcal{H}$ contains "rows."
  - There are only ceiling constraints.
- **Random priority** (RP) mechanism: randomly order agents, and let each agent receive the favorite remaining good following the order, subject to the constraints described above. Ex post efficient but not ex ant efficient.

Application: Single-Unit Assignment

- Social planner needs to assign at most one object to each agent (e.g., school choice, housing allocation).
- Each agent has strict preferences over O.
- Some additional constraints are allowed; affirmative action constraints, flexible capacity, etc.
- Suppose constraint sets $\mathcal{H}$ form a bihierchy.
  - $\mathcal{H}$ contains "rows."
  - There are only ceiling constraints.
- **Random priority** (RP) mechanism: randomly order agents, and let each agent receive the favorite remaining good following the order, subject to the constraints described above. Ex post efficient but not ex ant efficient.

Application: Single-Unit Assignment

- Social planner needs to assign at most one object to each agent (e.g., school choice, housing allocation).
- Each agent has strict preferences over O.
- Some additional constraints are allowed; affirmative action constraints, flexible capacity, etc.
- Suppose constraint sets $\mathcal{H}$ form a bihierchy.
  - $\mathcal{H}$ contains "rows."
  - There are only ceiling constraints.
- **Random priority** (RP) mechanism: randomly order agents, and let each agent receive the favorite remaining good following the order, subject to the constraints described above. Ex post efficient but not ex ant efficient.

Application: Single-Unit Assignment

- Social planner needs to assign at most one object to each agent (e.g., school choice, housing allocation).
- Each agent has strict preferences over O.
- Some additional constraints are allowed; affirmative action constraints, flexible capacity, etc.
- Suppose constraint sets $\mathcal{H}$ form a bihierchy.
  - $\mathcal{H}$ contains "rows."
  - There are only ceiling constraints.
- **Random priority** (RP) mechanism: randomly order agents, and let each agent receive the favorite remaining good following the order, subject to the constraints described above. Ex post efficient but not ex ant efficient.

Application: Single-Unit Assignment

- Social planner needs to assign at most one object to each agent (e.g., school choice, housing allocation).
- Each agent has strict preferences over O.
- Some additional constraints are allowed; affirmative action constraints, flexible capacity, etc.
- Suppose constraint sets $\mathcal{H}$ form a bihierchy.
  - $\mathcal{H}$ contains "rows."
  - There are only ceiling constraints.
- **Random priority** (RP) mechanism: randomly order agents, and let each agent receive the favorite remaining good following the order, subject to the constraints described above. Ex post efficient but not ex ant efficient.

Inefficiency of RP

Let $N = \{1, 2, 3, 4\}$, $O = \{a, b, c, \emptyset\}$. Each good has quota of one, and only two out of three goods can actually be produced.

1 and 2 like $a, b, \emptyset$ (in this order),

3 and 4 like $c, b, \emptyset$.

RP produces random assignment:

$$RP = \begin{pmatrix} 5/12 & 1/12 & 0 & 1/2 \\ 5/12 & 1/12 & 0 & 1/2 \\ 0 & 1/12 & 5/12 & 1/2 \\ 0 & 1/12 & 5/12 & 1/2 \end{pmatrix}.$$

Everyone prefers

$$P' = \begin{pmatrix} 1/2 & 0 & 0 & 1/2 \\ 1/2 & 0 & 0 & 1/2 \\ 0 & 0 & 1/2 & 1/2 \\ 0 & 0 & 1/2 & 1/2 \end{pmatrix}.$$

Inefficiency of RP

Let $N = \{1, 2, 3, 4\}$, $O = \{a, b, c, \emptyset\}$. Each good has quota of one, and only two out of three goods can actually be produced.

1 and 2 like $a, b, \emptyset$ (in this order),

3 and 4 like $c, b, \emptyset$.

RP produces random assignment:

$$RP = \begin{pmatrix} 5/12 & 1/12 & 0 & 1/2 \\ 5/12 & 1/12 & 0 & 1/2 \\ 0 & 1/12 & 5/12 & 1/2 \\ 0 & 1/12 & 5/12 & 1/2 \end{pmatrix}.$$

Everyone prefers

$$P' = \begin{pmatrix} 1/2 & 0 & 0 & 1/2 \\ 1/2 & 0 & 0 & 1/2 \\ 0 & 0 & 1/2 & 1/2 \\ 0 & 0 & 1/2 & 1/2 \end{pmatrix}.$$

Inefficiency of RP

Let $N = \{1, 2, 3, 4\}$, $O = \{a, b, c, \emptyset\}$. Each good has quota of one, and only two out of three goods can actually be produced.

1 and 2 like $a, b, \emptyset$ (in this order),

3 and 4 like $c, b, \emptyset$.

RP produces random assignment:

$$RP = \begin{pmatrix} 5/12 & 1/12 & 0 & 1/2 \\ 5/12 & 1/12 & 0 & 1/2 \\ 0 & 1/12 & 5/12 & 1/2 \\ 0 & 1/12 & 5/12 & 1/2 \end{pmatrix}.$$

Everyone prefers

$$P' = \begin{pmatrix} 1/2 & 0 & 0 & 1/2 \\ 1/2 & 0 & 0 & 1/2 \\ 0 & 0 & 1/2 & 1/2 \\ 0 & 0 & 1/2 & 1/2 \end{pmatrix}.$$

Inefficiency of RP

Let $N = \{1, 2, 3, 4\}$, $O = \{a, b, c, \emptyset\}$. Each good has quota of one, and only two out of three goods can actually be produced.

1 and 2 like $a, b, \emptyset$ (in this order),

3 and 4 like $c, b, \emptyset$.

RP produces random assignment:

$$RP = \begin{pmatrix} 5/12 & 1/12 & 0 & 1/2 \\ 5/12 & 1/12 & 0 & 1/2 \\ 0 & 1/12 & 5/12 & 1/2 \\ 0 & 1/12 & 5/12 & 1/2 \end{pmatrix}.$$

Everyone prefers

$$P' = \begin{pmatrix} 1/2 & 0 & 0 & 1/2 \\ 1/2 & 0 & 0 & 1/2 \\ 0 & 0 & 1/2 & 1/2 \\ 0 & 0 & 1/2 & 1/2 \end{pmatrix}.$$

Probabilistic Serial Mechanism (Bogomolnaia and Moulin)

- The agents regard the goods as "divisible" in probability units. Time runs continuously from 0 to 1, and each agent simultaneously "eats" her favorite "available" good at unit speed at each moment of time.
- The end outcome is a random assignment, implementable by BvN.
- The random assignment is "ordinally efficient," and "envy free."

Probabilistic Serial Mechanism (Bogomolnaia and Moulin)

- The agents regard the goods as “divisible” in probability units. Time runs continuously from 0 to 1, and each agent simultaneously “eats” her favorite “available” good at unit speed at each moment of time.
- The end outcome is a random assignment, implementable by BvN.
- The random assignment is “ordinally efficient,” and “envy free.”

Probabilistic Serial Mechanism (Bogomolnaia and Moulin)

- The agents regard the goods as "divisible" in probability units. Time runs continuously from 0 to 1, and each agent simultaneously "eats" her favorite "available" good at unit speed at each moment of time.
- The end outcome is a random assignment, implementable by BvN.
- The random assignment is "ordinally efficient," and "envy free."

Probabilistic Serial Mechanism (Bogomolnaia and Moulin)

- The agents regard the goods as “divisible” in probability units. Time runs continuously from 0 to 1, and each agent simultaneously “eats” her favorite “available” good at unit speed at each moment of time.
- The end outcome is a random assignment, implementable by BvN.
- The random assignment is “ordinally efficient,” and “envy free.”

Generalizing the Probabilistic Serial Mechanism

- As before: time runs continuously from 0 to 1, and each agent "eats" her favorite "available" good at speed one at each moment of time.
- But, modify "available": we say that object a is "available" to agent i if and only if the total amount of probability shares eaten away within S is less than the quota $\overline{q}_S$ for every constraint set $S \ni (i, a)$.
- The end outcome is a random assignment that satisfies bihierarchical constraints. Therefore it is implementable.
- We show: The assignment is ordinally efficient, and "feasible" envy free.
- May enhance the applicability of PS.

Generalizing the Probabilistic Serial Mechanism

- As before: time runs continuously from 0 to 1, and each agent "eats" her favorite "available" good at speed one at each moment of time.
- But, modify "available": we say that object a is "available" to agent i if and only if the total amount of probability shares eaten away within S is less than the quota $\overline{q}_S$ for every constraint set $S \ni (i, a)$.
- The end outcome is a random assignment that satisfies bihierarchical constraints. Therefore it is implementable.
- We show: The assignment is ordinally efficient, and "feasible" envy free.
- May enhance the applicability of PS.

Generalizing the Probabilistic Serial Mechanism

- As before: time runs continuously from 0 to 1, and each agent "eats" her favorite "available" good at speed one at each moment of time.
- But, modify "available": we say that object a is "available" to agent i if and only if the total amount of probability shares eaten away within S is less than the quota $\overline{q}_S$ for every constraint set $S \ni (i, a)$.
- The end outcome is a random assignment that satisfies bihierarchical constraints. Therefore it is implementable.
- We show: The assignment is ordinally efficient, and "feasible" envy free.
- May enhance the applicability of PS.

Generalizing the Probabilistic Serial Mechanism

- As before: time runs continuously from 0 to 1, and each agent "eats" her favorite "available" good at speed one at each moment of time.
- But, modify "available": we say that object a is "available" to agent i if and only if the total amount of probability shares eaten away within S is less than the quota $\overline{q}_S$ for every constraint set $S \ni (i, a)$.
- The end outcome is a random assignment that satisfies bihierarchical constraints. Therefore it is implementable.
- We show: The assignment is ordinally efficient, and "feasible" envy free.
- May enhance the applicability of PS.

Generalizing the Probabilistic Serial Mechanism

- As before: time runs continuously from 0 to 1, and each agent "eats" her favorite "available" good at speed one at each moment of time.
- But, modify "available": we say that object a is "available" to agent i if and only if the total amount of probability shares eaten away within S is less than the quota $\overline{q}_S$ for every constraint set $S \ni (i, a)$.
- The end outcome is a random assignment that satisfies bihierarchical constraints. Therefore it is implementable.
- We show: The assignment is ordinally efficient, and "feasible" envy free.
- May enhance the applicability of PS.

Generalizing Hylland Zeckhauser

The Hylland Zeckhauser mechanism produces competitive equilibrium outcome in random assignment in one-to-one assignment. We generalize the mechanism to environments in which

- agents demand arbitrary multiple units with additively separable preferences over objects
- agent faces constraints over hierarchical sets, e.g., in course allocation
  - Scheduling constraints: "no two classes that meet at the same time," or
  - Curricular constraints: "no more than two classes in finance"

Generalizing Hylland Zeckhauser

The Hylland Zeckhauser mechanism produces competitive equilibrium outcome in random assignment in one-to-one assignment. We generalize the mechanism to environments in which

- agents demand arbitrary multiple units with additively separable preferences over objects
- agent faces constraints over hierarchical sets, e.g., in course allocation
  - Scheduling constraints: "no two classes that meet at the same time," or
  - Curricular constraints: "no more than two classes in finance"

Generalizing Hylland Zeckhauser

The Hylland Zeckhauser mechanism produces competitive equilibrium outcome in random assignment in one-to-one assignment. We generalize the mechanism to environments in which

- agents demand arbitrary multiple units with additively separable preferences over objects
- agent faces constraints over hierarchical sets, e.g., in course allocation
  - Scheduling constraints: "no two classes that meet at the same time," or
  - Curricular constraints: "no more than two classes in finance"

Generalizing Hylland Zeckhauser

The Hylland Zeckhauser mechanism produces competitive equilibrium outcome in random assignment in one-to-one assignment. We generalize the mechanism to environments in which

- agents demand arbitrary multiple units with additively separable preferences over objects
- agent faces constraints over hierarchical sets, e.g., in course allocation
  - Scheduling constraints: "no two classes that meet at the same time," or
  - Curricular constraints: "no more than two classes in finance"

Generalizing Hylland Zeckhauser

The Hylland Zeckhauser mechanism produces competitive equilibrium outcome in random assignment in one-to-one assignment. We generalize the mechanism to environments in which

- agents demand arbitrary multiple units with additively separable preferences over objects
- agent faces constraints over hierarchical sets, e.g., in course allocation
  - Scheduling constraints: "no two classes that meet at the same time," or
  - Curricular constraints: "no more than two classes in finance"

Application: Course Allocation

- Course-allocation mechanisms currently used have flaws in fairness and efficiency (Budish and Cantillon, 2009).
- For the case of simple additive-separable preferences, the HZ generalization is attractive: efficient, interim envy free, and strategyproof in the large economy.
- Even nonlinear preferences, such as diminishing marginal utilities, can be encoded by the judicious design of message spaces: Milgrom (2010)'s assignment messages.

Application: Course Allocation

- Course-allocation mechanisms currently used have flaws in fairness and efficiency (Budish and Cantillon, 2009).
- For the case of simple additive-separable preferences, the HZ generalization is attractive: efficient, interim envy free, and strategyproof in the large economy.
- Even nonlinear preferences, such as diminishing marginal utilities, can be encoded by the judicious design of message spaces: Milgrom (2010)'s assignment messages.

Application: Course Allocation

- Course-allocation mechanisms currently used have flaws in fairness and efficiency (Budish and Cantillon, 2009).
- For the case of simple additive-separable preferences, the HZ generalization is attractive: efficient, interim envy free, and strategyproof in the large economy.
- Even nonlinear preferences, such as diminishing marginal utilities, can be encoded by the judicious design of message spaces: Milgrom (2010)'s assignment messages.

Application: Multi-Unit Assignment with Ex Post Fairness

- Suppose agents may be assigned to multiple objects, and they have linear preferences in the values of assigned objects, $\{v_{ia}\}$.
- There are multiple ways to implement a random assignment, some less fair than others.
- Example: $N = \{1, 2\}$; $O = \{a, b, c, d\}$, both have preferences $a \succ b \succ c \succ d$; each agent demands 2 units.

  A random assignment

$$\mathbf{P} = \begin{pmatrix} 0.5 & 0.5 & 0.5 & 0.5 \\ 0.5 & 0.5 & 0.5 & 0.5 \end{pmatrix}$$

  can be decomposed as

$$= \frac{1}{2}\begin{pmatrix} 1 & 1 & 0 & 0 \\ 0 & 0 & 1 & 1 \end{pmatrix} + \frac{1}{2}\begin{pmatrix} 0 & 0 & 1 & 1 \\ 1 & 1 & 0 & 0 \end{pmatrix}$$

  can also be decomposed as

$$= \frac{1}{2}\begin{pmatrix} 1 & 0 & 0 & 1 \\ 0 & 1 & 1 & 0 \end{pmatrix} + \frac{1}{2}\begin{pmatrix} 0 & 1 & 1 & 0 \\ 1 & 0 & 0 & 1 \end{pmatrix}$$

Application: Multi-Unit Assignment with Ex Post Fairness

- Suppose agents may be assigned to multiple objects, and they have linear preferences in the values of assigned objects, $\{v_{ia}\}$.
- There are multiple ways to implement a random assignment, some less fair than others.
- Example: $N = \{1, 2\}$; $O = \{a, b, c, d\}$, both have preferences $a \succ b \succ c \succ d$; each agent demands 2 units.

  A random assignment

$$\mathbf{P} = \begin{pmatrix} 0.5 & 0.5 & 0.5 & 0.5 \\ 0.5 & 0.5 & 0.5 & 0.5 \end{pmatrix}$$

  can be decomposed as

$$= \frac{1}{2}\begin{pmatrix} 1 & 1 & 0 & 0 \\ 0 & 0 & 1 & 1 \end{pmatrix} + \frac{1}{2}\begin{pmatrix} 0 & 0 & 1 & 1 \\ 1 & 1 & 0 & 0 \end{pmatrix}$$

  can also be decomposed as

$$= \frac{1}{2}\begin{pmatrix} 1 & 0 & 0 & 1 \\ 0 & 1 & 1 & 0 \end{pmatrix} + \frac{1}{2}\begin{pmatrix} 0 & 1 & 1 & 0 \\ 1 & 0 & 0 & 1 \end{pmatrix}$$

Application: Multi-Unit Assignment with Ex Post Fairness

- Suppose agents may be assigned to multiple objects, and they have linear preferences in the values of assigned objects, $\{v_{ia}\}$.
- There are multiple ways to implement a random assignment, some less fair than others.
- Example: $N = \{1, 2\}$; $O = \{a, b, c, d\}$, both have preferences $a \succ b \succ c \succ d$; each agent demands 2 units.

A random assignment

$$\mathbf{P} = \begin{pmatrix} 0.5 & 0.5 & 0.5 & 0.5 \\ 0.5 & 0.5 & 0.5 & 0.5 \end{pmatrix}$$

can be decomposed as

$$= \frac{1}{2}\begin{pmatrix} 1 & 1 & 0 & 0 \\ 0 & 0 & 1 & 1 \end{pmatrix} + \frac{1}{2}\begin{pmatrix} 0 & 0 & 1 & 1 \\ 1 & 1 & 0 & 0 \end{pmatrix}$$

can also be decomposed as

$$= \frac{1}{2}\begin{pmatrix} 1 & 0 & 0 & 1 \\ 0 & 1 & 1 & 0 \end{pmatrix} + \frac{1}{2}\begin{pmatrix} 0 & 1 & 1 & 0 \\ 1 & 0 & 0 & 1 \end{pmatrix}$$

Application: Multi-Unit Assignment with Ex Post Fairness

- Suppose agents may be assigned to multiple objects, and they have linear preferences in the values of assigned objects, $\{v_{ia}\}$.
- There are multiple ways to implement a random assignment, some less fair than others.
- Example: $N = \{1, 2\}$; $O = \{a, b, c, d\}$, both have preferences $a \succ b \succ c \succ d$; each agent demands 2 units.
  A random assignment

$$\mathbf{P} = \begin{pmatrix} 0.5 & 0.5 & 0.5 & 0.5 \\ 0.5 & 0.5 & 0.5 & 0.5 \end{pmatrix}$$

can be decomposed as

$$= \frac{1}{2}\begin{pmatrix} 1 & 1 & 0 & 0 \\ 0 & 0 & 1 & 1 \end{pmatrix} + \frac{1}{2}\begin{pmatrix} 0 & 0 & 1 & 1 \\ 1 & 1 & 0 & 0 \end{pmatrix}$$

can also be decomposed as

$$= \frac{1}{2}\begin{pmatrix} 1 & 0 & 0 & 1 \\ 0 & 1 & 1 & 0 \end{pmatrix} + \frac{1}{2}\begin{pmatrix} 0 & 1 & 1 & 0 \\ 1 & 0 & 0 & 1 \end{pmatrix}$$

Application: Multi-Unit Assignment with Ex Post Fairness

- Suppose agents may be assigned to multiple objects, and they have linear preferences in the values of assigned objects, $\{v_{ia}\}$.
- There are multiple ways to implement a random assignment, some less fair than others.
- Example: $N = \{1, 2\}$; $O = \{a, b, c, d\}$, both have preferences $a \succ b \succ c \succ d$; each agent demands 2 units.

  A random assignment

$$\mathbf{P} = \begin{pmatrix} 0.5 & 0.5 & 0.5 & 0.5 \\ 0.5 & 0.5 & 0.5 & 0.5 \end{pmatrix}$$

  can be decomposed as

$$= \frac{1}{2}\begin{pmatrix} 1 & 1 & 0 & 0 \\ 0 & 0 & 1 & 1 \end{pmatrix} + \frac{1}{2}\begin{pmatrix} 0 & 0 & 1 & 1 \\ 1 & 1 & 0 & 0 \end{pmatrix}$$

  can also be decomposed as

$$= \frac{1}{2}\begin{pmatrix} 1 & 0 & 0 & 1 \\ 0 & 1 & 1 & 0 \end{pmatrix} + \frac{1}{2}\begin{pmatrix} 0 & 1 & 1 & 0 \\ 1 & 0 & 0 & 1 \end{pmatrix}$$

Application: Multi-Unit Assignment with Ex Post Fairness

- Suppose agents may be assigned to multiple objects, and they have linear preferences in the values of assigned objects, $\{v_{ia}\}$.
- There are multiple ways to implement a random assignment, some less fair than others.
- Example: $N = \{1,2\}$; $O = \{a,b,c,d\}$, both have preferences $a \succ b \succ c \succ d$; each agent demands 2 units.

  A random assignment

$$\mathbf{P} = \begin{pmatrix} 0.5 & 0.5 & 0.5 & 0.5 \\ 0.5 & 0.5 & 0.5 & 0.5 \end{pmatrix}$$

  can be decomposed as

$$= \frac{1}{2}\begin{pmatrix} 1 & 1 & 0 & 0 \\ 0 & 0 & 1 & 1 \end{pmatrix} + \frac{1}{2}\begin{pmatrix} 0 & 0 & 1 & 1 \\ 1 & 1 & 0 & 0 \end{pmatrix}$$

  can also be decomposed as

$$= \frac{1}{2}\begin{pmatrix} 1 & 0 & 0 & 1 \\ 0 & 1 & 1 & 0 \end{pmatrix} + \frac{1}{2}\begin{pmatrix} 0 & 1 & 1 & 0 \\ 1 & 0 & 0 & 1 \end{pmatrix}$$

Application: Multi-Unit Assignment with Ex Post Fairness

- Suppose agents may be assigned to multiple objects, and they have linear preferences in the values of assigned objects, $\{v_{ia}\}$.
- There are multiple ways to implement a random assignment, some less fair than others.
- Example: $N = \{1, 2\}$; $O = \{a, b, c, d\}$, both have preferences $a \succ b \succ c \succ d$; each agent demands 2 units.
  A random assignment

$$\mathbf{P} = \begin{pmatrix} 0.5 & 0.5 & 0.5 & 0.5 \\ 0.5 & 0.5 & 0.5 & 0.5 \end{pmatrix}$$

can be decomposed as

$$= \frac{1}{2} \begin{pmatrix} 1 & 1 & 0 & 0 \\ 0 & 0 & 1 & 1 \end{pmatrix} + \frac{1}{2} \begin{pmatrix} 0 & 0 & 1 & 1 \\ 1 & 1 & 0 & 0 \end{pmatrix}$$

can also be decomposed as

$$= \frac{1}{2} \begin{pmatrix} 1 & 0 & 0 & 1 \\ 0 & 1 & 1 & 0 \end{pmatrix} + \frac{1}{2} \begin{pmatrix} 0 & 1 & 1 & 0 \\ 1 & 0 & 0 & 1 \end{pmatrix}$$

Application: Multi-Unit Assignment with Ex Post Fairness

- Suppose agents may be assigned to multiple objects, and they have linear preferences in the values of assigned objects, $\{v_{ia}\}$.
- There are multiple ways to implement a random assignment, some less fair than others.
- Example: $N = \{1, 2\}$; $O = \{a, b, c, d\}$, both have preferences $a \succ b \succ c \succ d$; each agent demands 2 units.

  A random assignment

$$\mathbf{P} = \begin{pmatrix} 0.5 & 0.5 & 0.5 & 0.5 \\ 0.5 & 0.5 & 0.5 & 0.5 \end{pmatrix}$$

  can be decomposed as

$$= \frac{1}{2}\begin{pmatrix} 1 & 1 & 0 & 0 \\ 0 & 0 & 1 & 1 \end{pmatrix} + \frac{1}{2}\begin{pmatrix} 0 & 0 & 1 & 1 \\ 1 & 1 & 0 & 0 \end{pmatrix}$$

  can also be decomposed as

$$= \frac{1}{2}\begin{pmatrix} 1 & 0 & 0 & 1 \\ 0 & 1 & 1 & 0 \end{pmatrix} + \frac{1}{2}\begin{pmatrix} 0 & 1 & 1 & 0 \\ 1 & 0 & 0 & 1 \end{pmatrix}$$

Application: Multi-Unit Assignment with Ex Post Fairness

- Suppose agents may be assigned to multiple objects, and they have linear preferences in the values of assigned objects, $\{v_{ia}\}$.
- There are multiple ways to implement a random assignment, some less fair than others.
- Example: $N = \{1, 2\}$; $O = \{a, b, c, d\}$, both have preferences $a \succ b \succ c \succ d$; each agent demands 2 units.

  A random assignment

$$\mathbf{P} = \begin{pmatrix} 0.5 & 0.5 & 0.5 & 0.5 \\ 0.5 & 0.5 & 0.5 & 0.5 \end{pmatrix}$$

  can be decomposed as

$$= \frac{1}{2}\begin{pmatrix} 1 & 1 & 0 & 0 \\ 0 & 0 & 1 & 1 \end{pmatrix} + \frac{1}{2}\begin{pmatrix} 0 & 0 & 1 & 1 \\ 1 & 1 & 0 & 0 \end{pmatrix}$$

  can also be decomposed as

$$= \frac{1}{2}\begin{pmatrix} 1 & 0 & 0 & 1 \\ 0 & 1 & 1 & 0 \end{pmatrix} + \frac{1}{2}\begin{pmatrix} 0 & 1 & 1 & 0 \\ 1 & 0 & 0 & 1 \end{pmatrix}$$

Application: Multi-Unit Assignment with Ex Post Fairness

Theorem: One-sided utility guarantee

Given any random assignment $\mathbf{P} = (P_{ia})$, there exists a BvN decomposition of $\mathbf{P}$ such that, for each $i \in N$, each ex post assignment in the decomposition gives i the expected utility within $\Delta_i := \max\{v_{ia} - v_{ib} | a, b \in O, P_{ia}, P_{ib} \notin \mathbb{Z}\}$ of that under $\mathbf{P}$.

Proof Idea

Add a hierarchical set of “artificial” constraints in a way that bounds the extent to which each agent’s utility can vary over different resolutions of the random assignment.

	a	b	c	d
1	0.5	0.5	0.5	0.5
2	0.5	0.5	0.5	0.5

This method works for more general (heterogenous preferences) cases.

Proof Idea

Add a hierarchical set of “artificial” constraints in a way that bounds the extent to which each agent’s utility can vary over different resolutions of the random assignment.

	a	b	c	d
1	0.5	0.5	0.5	0.5
2	0.5	0.5	0.5	0.5

This method works for more general (heterogenous preferences) cases.

Proof Idea

Add a hierarchical set of "artificial" constraints in a way that bounds the extent to which each agent's utility can vary over different resolutions of the random assignment.

	a	b	c	d
1	0.5	0.5	0.5	0.5
2	0.5	0.5	0.5	0.5

This method works for more general (heterogenous preferences) cases.

Proof Idea

Add a hierarchical set of "artificial" constraints in a way that bounds the extent to which each agent's utility can vary over different resolutions of the random assignment.

	a	b	c	d
1	0.5	0.5	0.5	0.5
2	0.5	0.5	0.5	0.5

This method works for more general (heterogenous preferences) cases.

Application: Two-Sided Matching

Theorem: Two-sided utility guarantee

Suppose both N and O are agents with strict preferences on the other side. Given any random assignment $\mathbf{P} = [P_{ia}]$, there exists a BvN decomposition of $\mathbf{P}$ such that, for each $i \in N$ and $a \in O$, each ex post assignment in the decomposition gives i the expected utility within $\Delta_i := \max\{v_{ia} - v_{ib} | a, b \in O, P_{ia}, P_{ib} \notin \mathbb{Z}\}$ of that under $\mathbf{P}$, and $a \in O$ the expected utility within $\Delta_a := \max\{v_{ia} - v_{ja} | i, j \in N, P_{ia}, P_{ja} \notin \mathbb{Z}\}$ of that under $\mathbf{P}$.

Example: Interleague Play Matchup Design

Suppose 8 (baseball) teams in two leagues, NL and AL, 4 teams in each league, must engage in interleague play — 6 games for each team against the teams in the other league. Wish to design equitable matchups.

List the teams in order of past performance (win/loss).

		AL			
		a	b	c	d
NL	1	1.5	1.5	1.5	1.5
	2	1.5	1.5	1.5	1.5
	3	1.5	1.5	1.5	1.5
	4	1.5	1.5	1.5	1.5

Example: Interleague Play Matchup Design

Suppose 8 (baseball) teams in two leagues, NL and AL, 4 teams in each league, must engage in interleague play — 6 games for each team against the teams in the other league. Wish to design equitable matchups.

List the teams in order of past performance (win/loss).

		AL			
		a	b	c	d
NL	1	1.5	1.5	1.5	1.5
	2	1.5	1.5	1.5	1.5
	3	1.5	1.5	1.5	1.5
	4	1.5	1.5	1.5	1.5

Example: Interleague Play Matchup Design

Suppose 8 (baseball) teams in two leagues, NL and AL, 4 teams in each league, must engage in interleague play — 6 games for each team against the teams in the other league. Wish to design equitable matchups.

List the teams in order of past performance (win/loss).

		AL			
		a	b	c	d
NL	1	1.5	1.5	1.5	1.5
	2	1.5	1.5	1.5	1.5
	3	1.5	1.5	1.5	1.5
	4	1.5	1.5	1.5	1.5

Example: Interleague Play Matchup Design

Suppose 8 (baseball) teams in two leagues, NL and AL, 4 teams in each league, must engage in interleague play — 6 games for each team against the teams in the other league. Wish to design equitable matchups.

List the teams in order of past performance (win/loss).

AL

NL		*a*	*b*	*c*	*d*
	1	1.5	1.5	1.5	1.5
	2	1.5	1.5	1.5	1.5
	3	1.5	1.5	1.5	1.5
	4	1.5	1.5	1.5	1.5

Example: Interleague Play Matchup Design

Suppose 8 (baseball) teams in two leagues, NL and AL, 4 teams in each league, must engage in interleague play — 6 games for each team against the teams in the other league. Wish to design equitable matchups.

List the teams in order of past performance (win/loss).

NL		AL *a*	*b*	*c*	*d*
	1	1.5	1.5	1.5	1.5
	2	1.5	1.5	1.5	1.5
	3	1.5	1.5	1.5	1.5
	4	1.5	1.5	1.5	1.5

One Possible Outcome

		AL			
		a	*b*	*c*	*d*
NL	1	2	1	1	2
	2	1	2	2	1
	3	1	2	2	1
	4	2	1	1	2

Beyond Bilateral Assignment

- The methodology can be extended to a general hypergraph $\mathcal{X} = (X, \mathcal{H})$ where X is a finite set and $\mathcal{H}$ is a collection of subsets from X.
- But we obtain a pair of impossibility of decomposition in
  1. Matching with more than 2-sides.
  2. One-sided ("roommate") matching.

Beyond Bilateral Assignment

- The methodology can be extended to a general hypergraph $\mathcal{X} = (X, \mathcal{H})$ where X is a finite set and $\mathcal{H}$ is a collection of subsets from X.
- But we obtain a pair of impossibility of decomposition in
  1. Matching with more than 2-sides.
  2. One-sided ("roommate") matching.

Beyond Bilateral Assignment

- The methodology can be extended to a general hypergraph $\mathcal{X} = (X, \mathcal{H})$ where X is a finite set and $\mathcal{H}$ is a collection of subsets from X.
- But we obtain a pair of impossibility of decomposition in
    1. Matching with more than 2-sides.
    2. One-sided ("roommate") matching.

Beyond Bilateral Assignment

- The methodology can be extended to a general hypergraph $\mathcal{X} = (X, \mathcal{H})$ where X is a finite set and $\mathcal{H}$ is a collection of subsets from X.
- But we obtain a pair of impossibility of decomposition in
  1. Matching with more than 2-sides.
  2. One-sided ("roommate") matching.

Beyond Bilateral Assignment

- The methodology can be extended to a general hypergraph $\mathcal{X} = (X, \mathcal{H})$ where X is a finite set and $\mathcal{H}$ is a collection of subsets from X.
- But we obtain a pair of impossibility of decomposition in
  1. Matching with more than 2-sides.
  2. One-sided ("roommate") matching.

Conclusion

We generalize the RA method by identifying most realistic constraint structure that guarantees implementation.

We show how the method can be applied to produce desirable random allocations in a variety of settings including single- and multi-unit demand assignment as well as two-sided matching.

Conclusion

We generalize the RA method by identifying most realistic constraint structure that guarantees implementation.

We show how the method can be applied to produce desirable random allocations in a variety of settings including single- and multi-unit demand assignment as well as two-sided matching.

Conclusion

We generalize the RA method by identifying most realistic constraint structure that guarantees implementation.

We show how the method can be applied to produce desirable random allocations in a variety of settings including single- and multi-unit demand assignment as well as two-sided matching.