Computationally Tractable Choice

Modibo Camara

Northwestern University

Theory Bag Lunch

September 23, 2021

Introduction

Motivation

Bounded rationality: optimization limited by computational constraints

Observed behavior hard to explain via standard, unboundedly rational models

- e.g. satisficing
- e.g. consideration sets
- e.g. choice bracketing

Potential consequences:unreliable inference, misleading policy recommendations

Motivation

Bounded rationality: optimization limited by computational constraints

Observed behavior hard to explain via standard, unboundedly rational models

- e.g. satisficing
- e.g. consideration sets
- e.g. choice bracketing

Potential consequences:unreliable inference, misleading policy recommendations

Motivation

Bounded rationality: optimization limited by computational constraints

Observed behavior hard to explain via standard, unboundedly rational models

- e.g. satisficing
- e.g. consideration sets
- e.g. choice bracketing

Potential consequences:unreliable inference, misleading policy recommendations

This Paper

Relationship between rationality, computation, and behavior

1. Formalize computational constraints on decision-making
2. Do computational constraints imply behavioral heuristics?
3. Are rationality axioms still appealing, given computational constraints?

This Paper

Relationship between rationality, computation, and behavior

1. Formalize computational constraints on decision-making
2. Do computational constraints imply behavioral heuristics?
3. Are rationality axioms still appealing, given computational constraints?

This Paper

Relationship between rationality, computation, and behavior

1. Formalize computational constraints on decision-making
2. Do computational constraints imply behavioral heuristics?
3. Are rationality axioms still appealing, given computational constraints?

This Paper

Relationship between rationality, computation, and behavior

1. Formalize computational constraints on decision-making
2. Do computational constraints imply behavioral heuristics?
3. Are rationality axioms still appealing, given computational constraints?

Framework

Choice bracketing:

"A decisionmaker who faces multiple decisions tends to choose an option in each case without full regard to the other decisions... she faces." (Rabin and Weizsäcker 2009)

Model of choice under risk

Decisionmaker cares about high-dimensional outcomes $z = (z_1, \ldots, z_n)$

Choices are rational := maximize $\mathrm{E}[u]$ for some u

Choices are symmetric := order doesn't matter (e.g. $u(z_1, z_2) = u(z_2, z_1)$)

Framework

Choice bracketing:

"A decisionmaker who faces multiple decisions tends to choose an option in each case without full regard to the other decisions... she faces." (Rabin and Weizsäcker 2009)

Model of choice under risk

Decisionmaker cares about high-dimensional outcomes $z = (z_1, \dots, z_n)$

Choices are rational $:=$ maximize $\mathrm{E}[u]$ for some u

Choices are symmetric $:=$ order doesn't matter (e.g. $u(z_1, z_2) = u(z_2, z_1)$)

Framework

Choice bracketing:

"A decisionmaker who faces multiple decisions tends to choose an option in each case without full regard to the other decisions... she faces." (Rabin and Weizsäcker 2009)

Model of choice under risk

Decisionmaker cares about high-dimensional outcomes $z = (z_1, \ldots, z_n)$

Choices are rational := maximize $\mathrm{E}[u]$ for some u

Choices are symmetric := order doesn't matter (e.g. $u(z_1, z_2) = u(z_2, z_1)$)

Framework

Choice bracketing:

"A decisionmaker who faces multiple decisions tends to choose an option in each case without full regard to the other decisions... she faces." (Rabin and Weizsäcker 2009)

Model of choice under risk

Decisionmaker cares about high-dimensional outcomes $z = (z_1, \ldots, z_n)$

Choices are rational $:=$ maximize $\mathrm{E}[u]$ for some u

Choices are symmetric $:=$ order doesn't matter (e.g. $u(z_1, z_2) = u(z_2, z_1)$)

Framework

Choice bracketing:

"A decisionmaker who faces multiple decisions tends to choose an option in each case without full regard to the other decisions… she faces." (Rabin and Weizsäcker 2009)

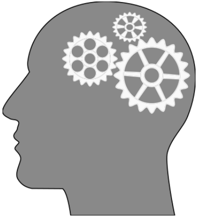
Model of choice under risk

Decisionmaker cares about high-dimensional outcomes $z = (z_1, \ldots, z_n)$

Choices are rational := maximize $\mathrm{E}[u]$ for some u

Choices are symmetric := order doesn't matter (e.g. $u(z_1, z_2) = u(z_2, z_1)$)

Generating Choices



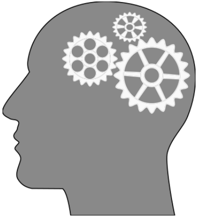
```
public void processData() {
   do {
      int data = getData();

      if (data < 0)
         performOperation1(data);
      else
         performOperation2(data);
   } while (hasMoreData());
}
```

Choices are tractable := exists reasonably quick algorithm that generates them

Source: https://commons.wikimedia.org/wiki/File:Java_keywords_highlighted.svg

Generating Choices

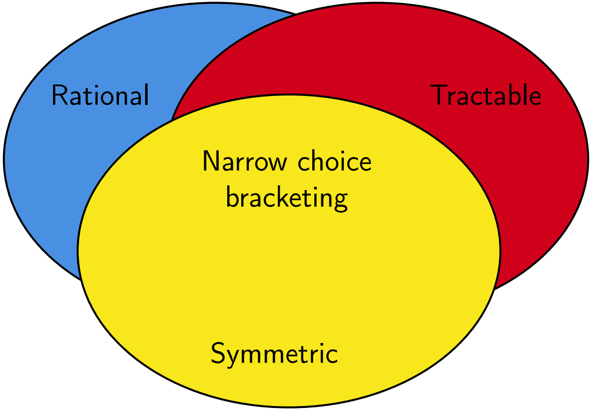
$\longrightarrow$



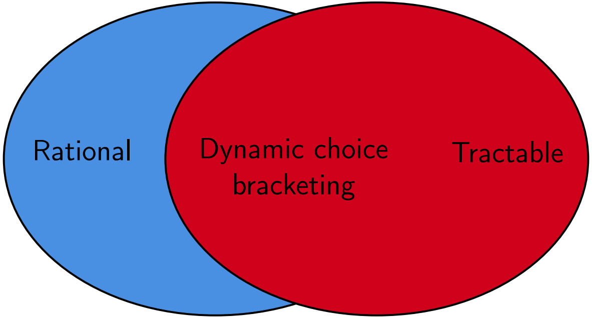
$\longrightarrow$

Choices are tractable := exists reasonably quick algorithm that generates them

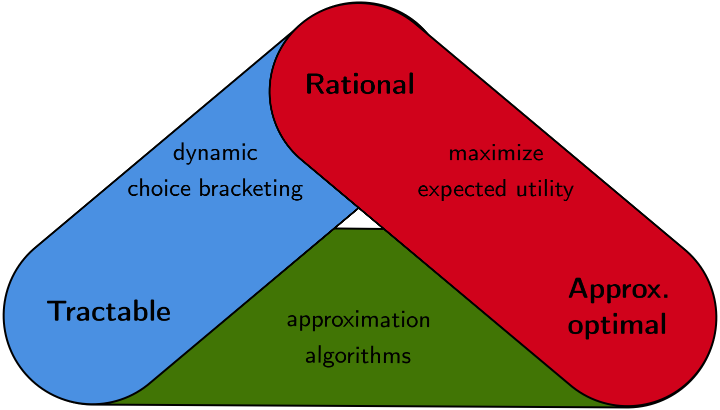
Rational + Tractable + Symmetric ≈ Narrow Choice Bracketing



Rational + Tractable ≈ Dynamic Choice Bracketing



Choice Trilemma



Related Literature

Computational constraints in decision theory (e.g. Al-Najjar, Casadesus-Masanell, and Ozdenoren 2003, Aragones, Gilboa, Postlewaite, and Schmeidler 2005, Apesteguia and Ballester 2010, **Echenique, Golovin, and Wierman 2011**, Mandler, Manzini, and Mariotti 2012, Mandler 2015, Gilboa, Postlewaite, and Schmeidler 2021, etc.)

Computational constraints in game theory , including equilibrium (e.g. Daskalakis, Goldberg, and Papadimitriou 2009), mechanism design (e.g. Nisan and Ronen 2001), repeated games (e.g. Rubinstein 1986), pricing (e.g. Rubinstein 1993), testing (Fortnow and Vohra 2009), social learning (Hązła, Jadbabaie, Mossel, and Rahimian 2021), etc.

Choice bracketing (e.g. Tversky and Kahneman 1981, Read, Loewenstein, and Rabin 1999, Haisley, Mostafa, and Loewenstein 2008, Rabin and Weizsäcker 2009, **Zhang 2021**, etc.)

Model

Choice under Risk

- lotteries l over outcomes z
- menus L of lotteries l
- collection $\mathcal{L}$ of menus L
- choice correspondence c maps menus $L \in \mathcal{L}$ to lotteries $l \in L$

Definition: choices c rational $\iff$ exists cardinal utility function $u : Z \to \mathbb{R}$ such that

$$c(L) = \arg\max_{l \in L} \mathrm{E}[u(l)]$$

Assumption: collection $\mathcal{L}$ includes all menus of three or fewer lotteries

Choice under Risk

- lotteries l over outcomes z
- menus L of lotteries l
- collection $\mathcal{L}$ of menus L
- choice correspondence c maps menus $L \in \mathcal{L}$ to lotteries $l \in L$

Definition: choices c rational $\iff$ exists cardinal utility function $u : Z \to \mathbb{R}$ such that

$$c(L) = \arg\max_{l \in L} \mathrm{E}[u(l)]$$

Assumption: collection $\mathcal{L}$ includes all menus of three or fewer lotteries

Choice under Risk

- lotteries l over outcomes z
- menus L of lotteries l
- collection $\mathcal{L}$ of menus L
- choice correspondence c maps menus $L \in \mathcal{L}$ to lotteries $l \in L$

Definition: choices c rational $\iff$ exists cardinal utility function $u : Z \to \mathbb{R}$ such that

$$c(L) = \arg\max_{l \in L} \mathrm{E}[u(l)]$$

Assumption: collection $\mathcal{L}$ includes all menus of three or fewer lotteries

Running Examples

Firm management

n branches

Branch i generates income z_i

Values total income according to

$$v(z_1 + \ldots + z_n)$$

Sets policy l_i for branch i

Consumer choice

n goods

Consumes z_i units of good i

Values consumption according to

$$u(z_1, \ldots, z_n)$$

Puchases quantity l_i of good i

Running Examples

Firm management

n branches

Branch i generates income z_i

Values total income according to

$$v(z_1 + \ldots + z_n)$$

Sets policy l_i for branch i

Consumer choice

n goods

Consumes z_i units of good i

Values consumption according to

$$u(z_1, \ldots, z_n)$$

Puchases quantity l_i of good i

High-Dimensional Choice

Outcomes z are infinite sequences $z \in \mathbb{R}^\infty$

- E.g. z_i = income from asset i
- E.g. z_i = amount of good i

Restrict attention to n-dimensional outcomes ($z_i = 0$ for all $i > n$)

Assumption: prefer higher outcomes ($z \geq z'$ implies $z \in c(\{z, z'\})$)

High-Dimensional Choice

Outcomes z are infinite sequences $z \in \mathbb{R}^\infty$

- E.g. $z_i =$ income from asset i
- E.g. $z_i =$ amount of good i

Restrict attention to n-dimensional outcomes ($z_i = 0$ for all $i > n$)

Assumption: prefer higher outcomes ($z \geq z'$ implies $z \in c(\{z, z'\})$)

Product Menus

Lottery $l = (l_1, \ldots, l_n)$ consists of n partial lotteries $l_i \in \Delta(\mathbb{R})$

Partial menu L_i consists of partial lotteries l_i

Definition: product menu = Cartesian product of n partial menus, i.e.

$$L = L_1 \times \ldots \times L_n \times \{0\} \times \{0\} \times \ldots$$

Assumption: collection $\mathcal{L}$ includes all product menus

Product Menus

Lottery $l = (l_1, \ldots, l_n)$ consists of n partial lotteries $l_i \in \Delta(\mathbb{R})$

Partial menu L_i consists of partial lotteries l_i

Definition: product menu = Cartesian product of n partial menus, i.e.

$$L = L_1 \times \ldots \times L_n \times \{0\} \times \{0\} \times \ldots$$

Assumption: collection $\mathcal{L}$ includes all product menus

Product Menus

Lottery $l = (l_1, \ldots, l_n)$ consists of n partial lotteries $l_i \in \Delta(\mathbb{R})$

Partial menu L_i consists of partial lotteries l_i

Definition: product menu = Cartesian product of n partial menus, i.e.

$$L = L_1 \times \ldots \times L_n \times \{0\} \times \{0\} \times \ldots$$

Assumption: collection $\mathcal{L}$ includes all product menus

Product Menus

Lottery $l = (l_1, \ldots, l_n)$ consists of n partial lotteries $l_i \in \Delta(\mathbb{R})$

Partial menu L_i consists of partial lotteries l_i

Definition: product menu = Cartesian product of n partial menus, i.e.

$$L = L_1 \times \ldots \times L_n \times \{0\} \times \{0\} \times \ldots$$

Assumption: collection $\mathcal{L}$ includes all product menus

Axiom of Computational Tractability

Informal definition: choices c are tractable if there exists algorithm with

Input: n-dimensional menu $L \in \mathcal{L}$

Output: lottery $l \in c(L)$

Runtime: $\mathrm{poly}(n)$

Formal definitions | Why polynomial-time? | Is this worst-case complexity?

Computational Hardness Conjectures

$*$ P $\neq$ NP

$**$ NP $\not\subseteq$ P/poly

$***$ Non-uniform ETH

$\downarrow$ Increasing strength

Choice Bracketing

Narrow Choice Bracketing

Definition: narrow choice bracketing $\iff$ optimize each partial menu L_i separately:

$$\max_{l_i \in L_i} \mathrm{E}[u_i(l_i)]$$

Definition: u is additively separable if

$$u(z) = u_1(z_1) + \ldots + u_n(z_n)$$

Observation: concepts are observationally equivalent in product menus, since

$$\underbrace{\max_{l \in L} \mathrm{E}[u_1(l_1) + \ldots + u_n(l_n)]}_{\text{additive separability}} = \underbrace{\max_{l_1 \in L_1} \mathrm{E}[u_1(l_1)] + \ldots + \max_{l_n \in L_n} \mathrm{E}[u_n(l_n)]}_{\text{narrow choice bracketing}}$$

Narrow Choice Bracketing

Definition: narrow choice bracketing $\iff$ optimize each partial menu L_i separately:

$$\max_{l_i \in L_i} \mathrm{E}[u_i(l_i)]$$

Definition: u is additively separable if

$$u(z) = u_1(z_1) + \ldots + u_n(z_n)$$

Observation: concepts are observationally equivalent in product menus, since

$$\underbrace{\max_{l \in L} \mathrm{E}[u_1(l_1) + \ldots + u_n(l_n)]}_{\text{additive separability}} = \underbrace{\max_{l_1 \in L_1} \mathrm{E}[u_1(l_1)] + \ldots + \max_{l_n \in L_n} \mathrm{E}[u_n(l_n)]}_{\text{narrow choice bracketing}}$$

Narrow Choice Bracketing

Definition: narrow choice bracketing $\iff$ optimize each partial menu L_i separately:

$$\max_{l_i \in L_i} \mathrm{E}[u_i(l_i)]$$

Definition: u is additively separable if

$$u(z) = u_1(z_1) + \ldots + u_n(z_n)$$

Observation: concepts are observationally equivalent in product menus, since

$$\underbrace{\max_{l \in L} \mathrm{E}[u_1(l_1) + \ldots + u_n(l_n)]}_{\text{additive separability}} = \underbrace{\max_{l_1 \in L_1} \mathrm{E}[u_1(l_1)] + \ldots + \max_{l_n \in L_n} \mathrm{E}[u_n(l_n)]}_{\text{narrow choice bracketing}}$$

Symmetry

Definition: choices c symmetric if $\{l, l'\} \in c(\{l, l'\})$ for any permutation l' of l

Representation Theorem

Theorem

Assume hardness conjecture $(*)$

Choices c are rational, tractable, and symmetric
$\implies$ c is rationalized by an additively separable utility function

Partial converse: choices c rationalized by additively separable utility function
$\implies$ c is tractable on the collection of product menus

Representation Theorem

Theorem

Assume hardness conjecture $(*)$

Choices c are rational, tractable, and symmetric
$\implies$ c is rationalized by an additively separable utility function

Partial converse: choices c rationalized by additively separable utility function
$\implies$ c is tractable on the collection of product menus

Representation Theorem

Theorem

Assume hardness conjecture $(*)$

Choices c are rational, tractable, and symmetric
$\Longrightarrow$ c is rationalized by an additively separable utility function

Partial converse: choices c rationalized by additively separable utility function
$\Longrightarrow$ c is tractable on the collection of product menus

Firm management

Running Example

Branch i generates income z_i

Firm cares about total income according to

$$u(z) = v(z_1 + \ldots + z_n)$$

Corollary: $\max \mathrm{E}[u]$ is tractable $\iff$ firm is risk-neutral (v linear)

Firm management

Running Example

Branch i generates income z_i

Firm cares about total income according to

$$u(z) = v(z_1 + \ldots + z_n)$$

Corollary: $\max \mathrm{E}[u]$ is tractable $\iff$ firm is risk-neutral (v linear)

Proof Sketch

Expected Utility Maximization

EUM (example)

Input: n-dimensional product menu L

Output: lottery l that maximizes expected utility where

$$u(z) = \sqrt{z_1 + \ldots + z_n}$$

Next: prove example is intractable (proof of hardness)

In paper: prove EUM is intractable for *any* relevant u (dichotomy theorem)

Expected Utility Maximization

EUM (example)

Input: n-dimensional product menu L

Output: lottery l that maximizes expected utility where

$$u(z) = \sqrt{z_1 + \ldots + z_n}$$

Next: prove example is intractable (proof of hardness)

In paper: prove EUM is intractable for *any* relevant u (dichotomy theorem)

Boolean Formulas

Variable $x_i \in \{\text{true}, \text{false}\}$

Clause = assertions combined by "or" statements

e.g. $(x_1 \vee \neg x_2 \vee x_3)$

Formula = clauses combined by "and" statements

e.g. $\text{clause}_1 \wedge \text{clause}_2 = (x_1 \vee x_2) \wedge (\neg x_1 \vee x_3)$

Boolean Formulas

Variable $x_i \in \{\text{true}, \text{false}\}$

Clause = assertions combined by "or" statements

e.g. $(x_1 \vee \neg x_2 \vee x_3)$

Formula = clauses combined by "and" statements

e.g. $\text{clause}_1 \wedge \text{clause}_2 = (x_1 \vee x_2) \wedge (\neg x_1 \vee x_3)$

Boolean Formulas

Variable $x_i \in \{\text{true}, \text{false}\}$

Clause = assertions combined by "or" statements

e.g. $(x_1 \vee \neg x_2 \vee x_3)$

Formula = clauses combined by "and" statements

e.g. $\text{clause}_1 \wedge \text{clause}_2 = (x_1 \vee x_2) \wedge (\neg x_1 \vee x_3)$

Proof by Contradiction

MIN 2-SAT

Input: formula with variables $x_1, \ldots, x_n$ where each clause_j has ≤ 2 assertions

Output: values $x_1, \ldots, x_n$ that minimize $\#$ of $\text{clause}_j = \text{true}$

P $\neq$ NP $\implies$ MIN 2-SAT not tractable (Kohli, Krishnamurti, and Mirchandani 1994)

Claim: EUM tractable $\implies$ MIN 2-SAT tractable

Proof by Contradiction

MIN 2-SAT

Input: formula with variables $x_1, \ldots, x_n$ where each clause_j has ≤ 2 assertions

Output: values $x_1, \ldots, x_n$ that minimize $\#$ of $\text{clause}_j = \text{true}$

P $\neq$ NP $\implies$ MIN 2-SAT not tractable (Kohli, Krishnamurti, and Mirchandani 1994)

Claim: EUM tractable $\implies$ MIN 2-SAT tractable

Proof by Contradiction

MIN 2-SAT

Input: formula with variables $x_1, \ldots, x_n$ where each clause_j has ≤ 2 assertions

Output: values $x_1, \ldots, x_n$ that minimize $\#$ of $\text{clause}_j = \text{true}$

P $\neq$ NP $\implies$ MIN 2-SAT not tractable (Kohli, Krishnamurti, and Mirchandani 1994)

Claim: EUM tractable $\implies$ MIN 2-SAT tractable

Reducing MIN 2-SAT to EUM

Input: formula with n variables $\longrightarrow$ n-dimensional product menu L

Output: $x_i \in \{\text{true}, \text{false}\} \longrightarrow$ partial lottery $l_i \in \left\{ l_i^{\text{true}}, l_i^{\text{false}} \right\}$

Goal: minimize probability that random clause j is true

- Equivalent to minimizing number of true clauses (MIN 2-SAT)
- Equivalent to maximizing indicator that j is false (EUM)

Strategy: construct lotteries so that

$$\text{clause}_j = \text{false} \implies \sqrt{l_1^{x_1} + \ldots + l_n^{x_n}} = 2.83$$

$$\text{clause}_j = \text{true} \implies \sqrt{l_1^{x_1} + \ldots + l_n^{x_n}} = 2$$

Reducing MIN 2-SAT to EUM

Input: formula with n variables $\longrightarrow$ n-dimensional product menu L

Output: $x_i \in \{\text{true}, \text{false}\} \longrightarrow$ partial lottery $l_i \in \left\{ l_i^{\text{true}}, l_i^{\text{false}} \right\}$

Goal: minimize probability that random clause j is true

- Equivalent to minimizing number of true clauses (MIN 2-SAT)
- Equivalent to maximizing indicator that j is false (EUM)

Strategy: construct lotteries so that

$$\text{clause}_j = \text{false} \implies \sqrt{l_1^{x_1} + \ldots + l_n^{x_n}} = 2.83$$

$$\text{clause}_j = \text{true} \implies \sqrt{l_1^{x_1} + \ldots + l_n^{x_n}} = 2$$

Reducing MIN 2-SAT to EUM

Input: formula with n variables $\longrightarrow$ n-dimensional product menu L

Output: $x_i \in \{\text{true}, \text{false}\}$ $\longrightarrow$ partial lottery $l_i \in \{l_i^{\text{true}}, l_i^{\text{false}}\}$

Goal: minimize probability that random clause j is true

- Equivalent to minimizing number of true clauses (MIN 2-SAT)
- Equivalent to maximizing indicator that j is false (EUM)

Strategy: construct lotteries so that

$$\text{clause}_j = \text{false} \implies \sqrt{l_1^{x_1} + \ldots + l_n^{x_n}} = 2.83$$

$$\text{clause}_j = \text{true} \implies \sqrt{l_1^{x_1} + \ldots + l_n^{x_n}} = 2$$

Reducing MIN 2-SAT to EUM

Input: formula with n variables $\longrightarrow$ n-dimensional product menu L

Output: $x_i \in \{\text{true}, \text{false}\} \longrightarrow$ partial lottery $l_i \in \{l_i^{\text{true}}, l_i^{\text{false}}\}$

Goal: minimize probability that random clause j is true

- Equivalent to minimizing number of true clauses (MIN 2-SAT)
- Equivalent to maximizing indicator that j is false (EUM)

Strategy: construct lotteries so that

$$\text{clause}_j = \text{false} \implies \sqrt{l_1^{x_1} + \ldots + l_n^{x_n}} = 2.83$$

$$\text{clause}_j = \text{true} \implies \sqrt{l_1^{x_1} + \ldots + l_n^{x_n}} = 2$$

Constructing I_i^{true}

1. Randomly choose a clause $j = 1, \ldots, m$, where

$$\mathbf{1}(\mathrm{clause}_j) = \mathbf{1}(x_{j1} \vee x_{j2}) = \underbrace{\mathbf{1}(x_{j1} \wedge x_{j2})}_{(1)} + \underbrace{\mathbf{1}(\neg x_{j1} \wedge x_{j2})}_{(2)} + \underbrace{\mathbf{1}(x_{j1} \wedge \neg x_{j2})}_{(3)}$$

2. Randomly choose an integer $k \in \{1, 2, 3\}$
   - If (k) includes $\neg x_i$, return

$$z_i = \begin{cases} 0.34 & k = 1 \\ 1 & k = 2, 3 \end{cases}$$

   - Otherwise, return $z_i = 0$

Constructing I_i^{false}

1. Randomly choose a clause $j = 1, \ldots, m$, where

$$\mathbf{1}(\text{clause}_j) = \mathbf{1}(x_{j1} \vee x_{j2}) = \underbrace{\mathbf{1}(x_{j1} \wedge x_{j2})}_{(1)} + \underbrace{\mathbf{1}(\neg x_{j1} \wedge x_{j2})}_{(2)} + \underbrace{\mathbf{1}(x_{j1} \wedge \neg x_{j2})}_{(3)}$$

2. Randomly choose an integer $k \in \{1, 2, 3\}$
   - If (k) includes x_i, return

$$z_i = \begin{cases} 0.34 & k = 1 \\ 1 & k = 2, 3 \end{cases}$$

   - Otherwise, return $z_i = 0$

Dynamic Choice Bracketing

Inseparability Graph

Running Example

Consumer cares about quantities $z_1, \ldots, z_n$ of goods $1, \ldots, n$

Definition: goods i and j are separable if

$$u(z) = u_i(z_i, z_{-ij}) + u_j(z_j, z_{-ij})$$

Definition: inseparability graph G_n with n nodes

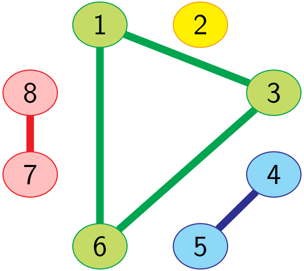
- Goods i and j share edge iff not separable

Example of Choice Bracketing

Running Example

Bracket together goods $\{1, 3, 6\}$, $\{2\}$, $\{4, 5\}$, $\{7, 8\}$, ...

$$u(z) = u_1(z_1, z_3, z_6) + u_2(z_2) + u_3(z_4, z_5) + u_4(z_7, z_8) + \ldots$$

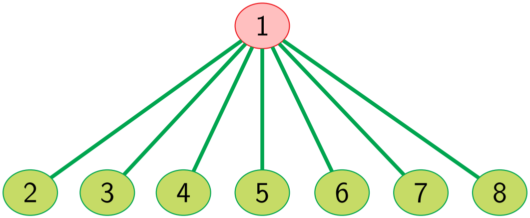


Example of Dynamic Choice Bracketing

Running Example

Conditional on good $i = 1$, narrowly bracket goods $i = 2, \ldots, n$

$$u(z) = u_1(z_1, z_2) + \ldots + u_{n-1}(z_1, z_n)$$



Hadwiger Separability

Hadwiger number $\mathrm{Had}(G)$ measures how dense graph G is.

Definition: u is Hadwiger separable if $\mathrm{Had}(G_n) = O(\log n)$

Observationally equivalent to "dynamic choice bracketing"

- Like choice bracketing, selectively ignores links between decisions
- Unlike choice bracketing, brackets may change in the process of making choice

---

Definition of Hadwiger number

Representation Theorem

Theorem

Assume hardness conjecture $(***)$

Choices c are rational and tractable

$\implies$ c is rationalized by a Hadwiger separable utility function

Partial converse: choices c rationalized by Hadwiger separable utility function

$\implies$ c is tractable on the collection of product menus

Proof outline | Linking representation theorems

Revisiting Rationality

When Approximation is Necessary

Suppose you care about utility u

1st-best: maximize $\mathrm{E}[u]$

2nd-best: approximately maximize $\mathrm{E}[u]$ s.t. tractability

3rd-best: approximately maximize $\mathrm{E}[u]$ s.t. rationality & tractability

u is Hadwiger separable $\implies$ 1st-best $=$ 2nd-best $=$ 3rd-best

u not Hadwiger separable $\implies$ 1st-best $>$ 2nd-best $\overset{?}{>}$ 3rd-best

When Approximation is Necessary

Suppose you care about utility u

1st-best: maximize $\mathrm{E}[u]$

2nd-best: approximately maximize $\mathrm{E}[u]$ s.t. tractability

3rd-best: approximately maximize $\mathrm{E}[u]$ s.t. rationality & tractability

u is Hadwiger separable $\Longrightarrow$ 1st-best $=$ 2nd-best $=$ 3rd-best

u not Hadwiger separable $\Longrightarrow$ 1st-best $>$ 2nd-best $\overset{?}{>}$ 3rd-best

Quantifying Approximate Optimality

Definition: choices c achieve an approximation ratio of

$$\inf_{L \in \mathcal{L}} \left(\frac{\text{actual payoff in menu } L}{\text{optimal payoff in menu } L} \right) = \inf_{L \in \mathcal{L}} \frac{\mathrm{E}[u(c(l))]}{\max_{l \in L} \mathrm{E}[u(l)]}$$

Definition: optimal approximation ratio := best ratio achieved by tractable c.

Quantifying Approximate Optimality

Definition: choices c achieve an approximation ratio of

$$\inf_{L \in \mathcal{L}} \left(\frac{\text{actual payoff in menu } L}{\text{optimal payoff in menu } L} \right) = \inf_{L \in \mathcal{L}} \frac{\mathrm{E}[u(c(l))]}{\max_{l \in L} \mathrm{E}[u(l)]}$$

Definition: optimal approximation ratio $:=$ best ratio achieved by tractable c.

Optimal Approximation Algorithms violate Rationality Axioms

Theorem

Assume hardness conjecture $(**)$

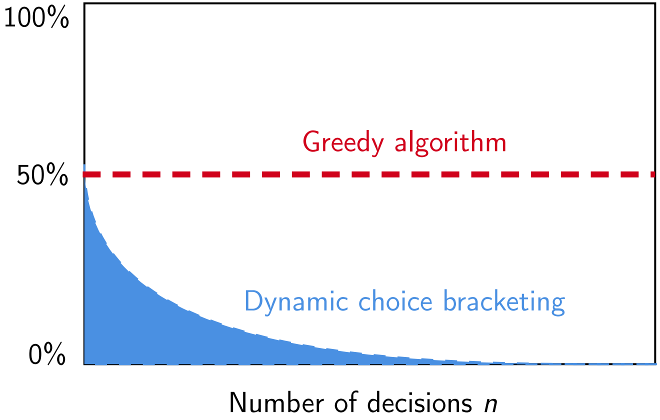
There exist utility functions u such that:

- Rational & tractable choices c cannot achieve any constant approximation ratio
- Optimal approximation ratio is $\geq 1/2$

Proof outline | Relevant utility functions | Why approximation ratio?

Illustrating the Performance Gap

Performance Guarantees for $u(z) = \sqrt{z_1 + \ldots + z_n}$



Conclusion

The Bigger Picture

Tractability axioms are a promising tool for decision theory

- Can tractability axioms motivate *other* behavioral heuristics?

Rationality axioms should be relaxed further

- Computer scientists study approximate solutions to intractable problems
- Can we learn from them?

Broader role for bounded rationality in econometrics?

- Tractability tends towards "simpler" behaviors
- Simpler behaviors tend to be easier to estimate
- E.g. Hadwiger separable functions are *much* easier to estimate

The Bigger Picture

Tractability axioms are a promising tool for decision theory

- Can tractability axioms motivate *other* behavioral heuristics?

Rationality axioms should be relaxed further

- Computer scientists study approximate solutions to intractable problems
- Can we learn from them?

Broader role for bounded rationality in econometrics?

- Tractability tends towards "simpler" behaviors
- Simpler behaviors tend to be easier to estimate
- E.g. Hadwiger separable functions are *much* easier to estimate

The Bigger Picture

Tractability axioms are a promising tool for decision theory

- Can tractability axioms motivate *other* behavioral heuristics?

Rationality axioms should be relaxed further

- Computer scientists study approximate solutions to intractable problems
- Can we learn from them?

Broader role for bounded rationality in econometrics?

- Tractability tends towards “simpler” behaviors
- Simpler behaviors tend to be easier to estimate
- E.g. Hadwiger separable functions are *much* easier to estimate

Appendix

Why Poly-Time?

Jason: discussion needs improvement; does not succintly give audience what they want
Interpretation: if it is tractable then *maybe* a human could do it; if it is not tractable then there's no way
explain poly-time; graph of 2^n vs n^2 vs n^{100} sufficiently zoomed out
show a vertical line depicting time constraint (e.g. mortality)
qualitative difference between poly-time and exponential-time

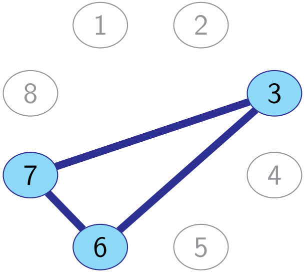
Why Not Model Computational Costs?

Why not model this as optimization subject to cost? -agents rationally choose not to optimize in menu L because it is too hard
hardness is not a property of the menu; it is a property of the choice correspondence (like IIA)
picture where you have a sequence of menus with yes or no answers
always answer yes gives you right answer on first menu and wrong on second (EASY)
always answer no gives you wrong answer on first menu and right on second (EASY)
give the correct answer (HARD)
for every menu there exists a quick algorithm that gets it right; but there is no algorithm that recreates the choice correspondence

Isn't this worst-case analysis?

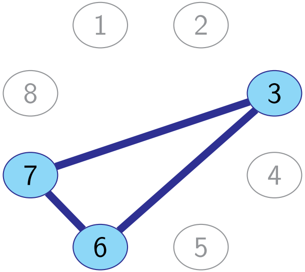
contrapositive of tractability: there exists no menu where the agent takes more than $O(poly n)$ time
motivation: you are mortal
observation: this is about choices, not optimization; tractability only means worst-case complexity if the agent never fails to optimize
contrapositive of rationality: there exists no menu where the agent fails to optimize
motivation: "introspection"
observation: "introspection" is not a well-defined algorithm, and no well-defined algorithm can bypass the impossibility results in this paper, assuming that the computational hardness conjectures hold
If you don't like worst-case analysis, need to revisit the definition of rationality.

Example 2: Consideration set of size $\leq k$.



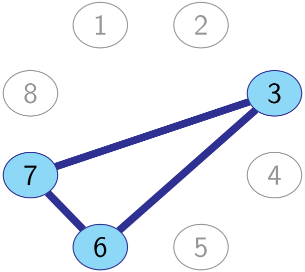
In both cases, Hadwiger separable if $k = O(\log n)$.

Example 2: Consideration set of size $\leq k$.



In both cases, Hadwiger separable if $k = O(\log n)$.

Example 2: Consideration set of size $\leq k$.



In both cases, Hadwiger separable if $k = O(\log n)$.

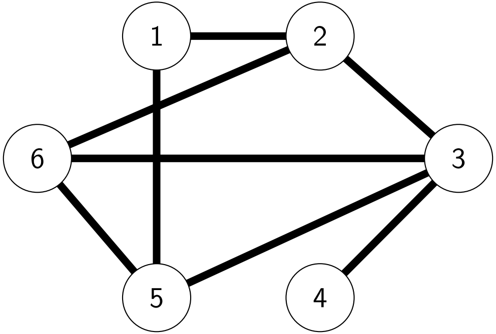
The Hadwiger Number

Definition

1. *The graph H is a minor of G if it can be formed from G by (a) deleting nodes i and (b) contracting edges (i, j).*
2. *The Hadwiger number $\mathrm{Had}(G)$ is the number of nodes in the largest complete minor H of G.*

The Hadwiger Number, Illustrated

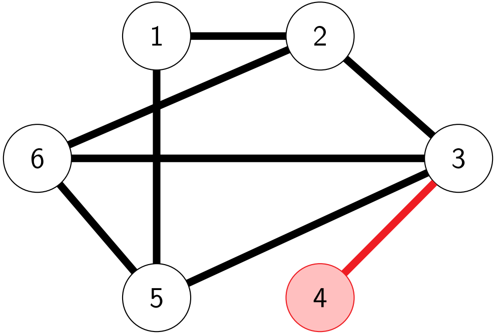
Let G be the following graph.



What is $\mathrm{Had}(G)$?

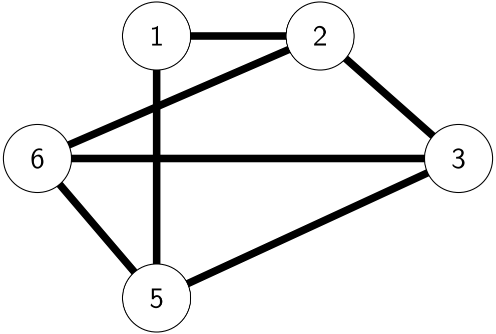
The Hadwiger Number, Illustrated

Delete node 4.



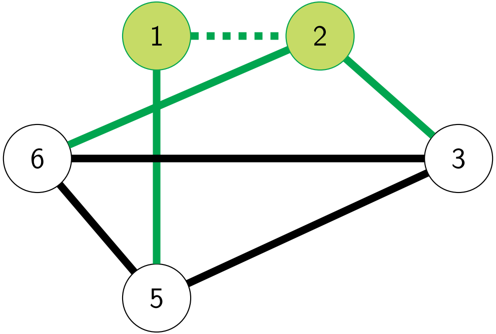
The Hadwiger Number, Illustrated

Obtain the minor H of graph G.



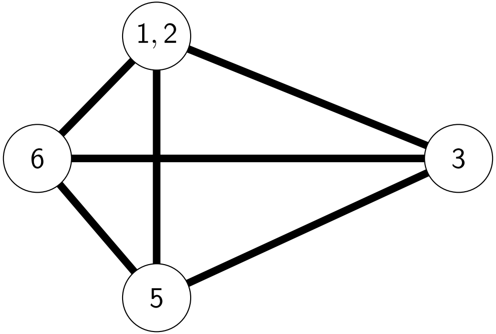
The Hadwiger Number, Illustrated

Contract the edge between nodes 1 and 2.



The Hadwiger Number, Illustrated

Obtain the minor H' of graph G. Note that it is complete.

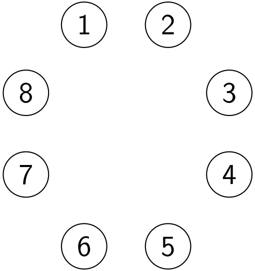


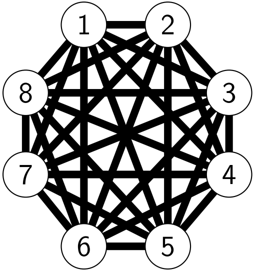
Remember, $\mathrm{Had}(G)$ is the number of nodes in the largest complete minor. In fact, H' is the largest complete minor, so $\mathrm{Had}(G) = 4$.

Connecting the Representation Theorems

Claim: u symmetric $\implies$ $\{u$ Hadwiger separable $\iff$ u additively separable$\}$

Proof: u symmetric $\implies$ inseparability graph G_n either (1) empty or (2) complete





(1) G_n empty $\implies$ u additively separable (2) G_n complete $\implies$ $\mathrm{Had}(G_n) = n$

Proof Outline

Hardness

Construction more elaborate than before

$\max_{l \in L} \mathrm{E}[u(l)]$ tractable

$\implies$

3-SAT with $\mathrm{Had}(G_n)$ variables solvable in $o(2^n)$ time

Contradicts hardness conjecture $(***)$ unless $\mathrm{Had}(G_n) = O(\log n)$

Tractability

Non-trivial

Show that if the algorithm ever gets “stuck” solving a high-dimensional optimization problem, then the Hadwiger number must be large

Outline of Proof

Example: $u(z) = \sqrt{z_1 + \ldots + z_n}$
Adapt greedy algorithm for MAX 2-SAT (Johnson 1974)
Let choices c be rational and tractable
$\implies$ dynamic choice bracketing
Construct menu L where dynamic choice bracketing is bad heuristic
Relies on characterizations of chromatic number of graphs (Szekeres and Wilf 1968)