High-Dimensional Decision Theory

Modibo K. Camara

September 26, 2019

Abstract

An agent makes choices with n-dimensional Euclidean consequences. She maximizes expected utility with an increasing utility function $u^n : \mathbb{R}^n \to \mathbb{R}$. Under permissive assumptions relating u^n with u^{n+1}, complexity theory can formalize the sense in which the agent's problem becomes challenging as the dimension n increases. If the P $\neq$ NP conjecture holds, a computational tractability axiom implies that u^n must be additively separable across dimensions for all n. This result provides an axiomatic foundation for narrow choice bracketing.

1 Introduction

An individual makes many interrelated decisions in her lifetime. Within the span of a few years, she may choose whether to buy a car, where to invest her savings, how to structure her debt, whether to take an expensive vacation, whether to start a family, etc. She must do this without knowing her career trajectory: whether she will be promoted, given a raise, or fired in the next five years. Yet her career trajectory is relevant to all of these decisions.

This paper concerns itself with the decision-making of this individual: namely, high-dimensional choice under uncertainty. This kind of problem is both ubiquitous in the real world – for individuals, organizations, and communities – and common in economic models, particularly those involving extensive form games or dynamic programming. Yet, it is poorly understood. My results demonstrate that new phenomena can arise under a high-dimensional analysis that are obscured under a low-dimensional analysis.

Taking high-dimensional choice seriously requires us, as a field, to contemplate several fundamental challenges to our theory and modeling practices. For pedagogical reasons, this paper will ignore most of them. I adopt a classical model of choice under uncertainty, introduced by Savage (1954). For any given menu of Savage acts, the agent chooses the act that maximizes her subjective expected utility. The space of consequences is a subset of $\mathbb{R}^n$. In order to scale the problem appropriately as the dimension n increases, I impose additional assumptions relating the agent's utility in the n-dimensional problem to her utility in the $(n + 1)$-dimensional problem. This has the flavor of a recursive utility representation, but permits a wide variety of utility functions.

The main novelty in high-dimensional decision-making is complexity. Even though we restrict attention to finite menus of acts, expected utility maximization still requires the agent to solve a high-dimensional combinatorial optimization problem. These problems can take a considerable amount of time to solve; in some cases they are intractable, even for state-of-the-art algorithms on modern computers. To reflect this difficulty, I introduce a computational tractability axiom. It requires the agent's choice correspondence to have polynomial time complexity. That is, there must exist an algorithm that, given a menu, can output the agent's choice from that menu with runtime that is at most some polynomial function of the dimension n.

The upshot of all this is a representation theorem: assuming that P $\neq$ NP, the agent's utility function must be additively separable across dimensions. If the utility function is not additively separable, I show that the agent's optimization problem is NP-hard via reduction from MAX2SAT and MIN2SAT. If the utility function is additively separable, there is a simple optimization algorithm with runtime that is linear in n.

This representation is consistent with a phenomenon observed in experimental economics, where individuals make decisions one-at-a-time without fully considering their joint implications. Read et al. (1999) describe this phenomenon:

> [Choice bracketing] designates the grouping of individual choices together into sets. A set of choices are bracketed together when they are made by taking into account the effect of each choice on all other choices in the set, but not on choices outside of the set. When the sets are small, containing one or very few choices, we say that bracketing is narrow, while when the sets are large, we say that it is broad. Broad bracketing allows people to consider all the hedonic consequences of their actions, and hence promotes utility maximization. Narrow bracketing, on the other hand, is like fighting a war one battle at a time with no overall guiding strategy, and it can have similar consequences.

There is considerable experimental evidence for narrow choice bracketing (NCB) (Andersen et al. 2018; Rabin and Weizsäcker 2009; Read et al. 1999). Some evidence and interpretations tend towards framing effects, where the type of bracketing depends on the framing of the problem (Brown et al. 2017; Haisley et al. 2008). Others tend towards bounded rationality, where NCB is a response to the complexity of combinatorial optimization (Stracke et al. 2017).

2 Representation Theorem

Consider a relatively-standard model of high-dimensional choice under uncertainty. Let $\theta \in \Theta = \{1, \ldots, m\}$ be an unknown state of the world. Let $(z_1, \ldots, z_n) \in Z \subseteq \mathbb{R}^n$ be an n-dimensional Euclidean consequence. Let $h^n : \Theta \to Z$ be an act, mapping states to n rewards. Let H^n be a menu of acts. Let $\mathcal{H}^n$ be a collection of menus. A choice correspondence ϕ^n maps menus $H^n \in \mathcal{H}^n$ in the collection to acts $h^n \in H^n$ in that menu. I drop superscripts whenever possible.

Assumption 1 (Richness)**.** *The space of consequences includes a cube, i.e.* $[\underline{z}, \bar{z}]^n \subseteq Z$ *for some real numbers* $\underline{z} < \bar{z}$*. Furthermore, the collection* $\mathcal{H}$ *includes all product menus, i.e.*

$$H := H_1 \times H_2 \times \ldots \times H_n$$

where H_i *consists of functions* $\Theta \to [\underline{z}, \bar{z}]$.

The following assumptions relate choice $\phi(H)$ across menus H.

Assumption 2 (Expected Utility)**.** *There exists a prior distribution* $\pi \in \Delta(\Theta)$ *such that for all dimensions n and menus* $H^n \in \mathcal{H}^n$*, the agent's choice satisfies:*

$$\phi(H^n) = \arg\max_{h^n \in H^n} \sum_{j=1}^{m} \pi_j \cdot u^n(h^n(j))$$

where $u^n : Z \to \mathbb{R}$ *is a continuous utility function.*

Assumption 3 (Monotonicity)**.** *The utility function* u^n *is weakly increasing in general and strictly increasing along the diagonal.*

Assumption 4 (Symmetry)**.** *The prior* π *is uniform. The utility function* u^n *is permutation-invariant.*

The following assumptions relate choice ϕ^n across dimensions n.

Assumption 5 (Recursive Utility)**.** *Assume 1. For some* $e \in [\underline{z}, \bar{z}]$*, let* $\gamma(z) = u^2(z, e)$ *be strictly increasing in z. Then define*

$$u^n(z_1, \ldots, z_n) = u^2\left(z_1, \gamma^{-1}\left(u^{n-1}(z_2, \ldots, z_n)\right)\right)$$

Example 1. *Utility functions that satisfy assumption our assumptions so far include:*

- $g\left(\sum_{i=1}^n f(z_i)\right)$ *for* f, g *strictly increasing and* $0 \in f([\underline{z}, \bar{z}])$
- $g\left(\prod_{i=1}^n f(z_i)\right)$ *for* f, g *strictly increasing and* $1 \in f([\underline{z}, \bar{z}])$
- $g\left(\max_i f(z_i)\right)$ *for* f, g *strictly increasing and* $a \in f([\underline{z}, \bar{z}])$
- $g\left(\min_i f(z_i)\right)$ *for* f, g *strictly increasing and* $b \in f([\underline{z}, \bar{z}])$

Assumption 6 (Tractability)**.** *There exists an algorithm that takes in* $H^n \in \mathcal{H}^n$ *and outputs* $h^n \in \phi^n(H^n)$ *with worst-case runtime that is* $O(\text{poly}(n, m))$.

This definition of tractability has been tremendously influential in theoretical computer science (see Cobham-Edmonds thesis). Keep in mind that tractability does not imply facility. For any choice correspondence and any algorithm, there will be a sufficiently large number $\bar{n}$ that makes it computationally infeasible. Instead, time complexity classifies problems by how quickly they become infeasible as n grows. In practice, there is a subsantial gap between (super)-exponential-time and (sub)-polynomial-time problems. To illustrate, suppose our agent can perform one operation per second. If her choice correspondence requires up to n^2 operations and she lives for one-hundred years, the agent can accept up to $\bar{n} = 56,156$ decisions. If she accepted more, she would violate completeness by failing to make a choice in some hypothetical menu. However, if her choice correspondence takes up to 2^n operations, she can only accept up to $\bar{n} = 22$ decisions.

There is a common objection to this definition of tractability – that it classifies problems by their worst-case runtime – but it is not relevant to our model. Yes, expected utility maximization may be difficult for some menus while simultaneously being easy for the "typical" menu, where even simple heuristics like NCB will find the optimal solution. However, in axiomatic decision theory, a representation and its underlying axioms apply to *all* menus in a collection; indeed, researchers lean heavily on this fact in their proofs. As a consequence, if optimization is computationally infeasible for *any* menu then the *entire* representation must be ruled out. The reader is more than welcome to question this practice, in which case theorem 1 can be viewed as a negative result.

Next, I state the main result. The proof is deferred to section 3.

Theorem 1. *Assume 1, 2, 3, 4, 5, 6. The utility function u is additively separable if $P \neq NP$.*

Example 2. *Theorem 1 uses assumption 5 to rule out the following counterexamples:*

- *The utility function is $\prod_{i=1}^{n} z_i$ for $n < 2^{100}$ but becomes $\sum_{i=1}^{n} z_i$ for $n \geq 2^{100}$.*
- *The utility function is $(z_1 z_2) + (z_3 z_4) + \ldots + (z_{n-1} z_n)$, which pairs together decisions.*

Both cases induce tractable optimization problems, but neither violates the spirit of theorem 1.

3 Proof of Theorem 1

Let $x_1, \ldots, x_n$ be boolean variables. Let $c_1, \ldots, c_m$ be clauses of up to n literals (i.e. x_i or $\neg x_i$). An assignment gives a truth value to each x_i. A clause is satisfied if at least one literal is true. For example, if $n = 3$, $c_1 = (x_1 \vee x_2)$, and $c_2 = (\neg x_1 \vee x_3)$, then the solution $(\mathtt{true}, \mathtt{true}, \mathtt{true})$ satisfies both clauses while $(\mathtt{false}, \mathtt{false}, \mathtt{false})$ only satisfies clause c_2.

If each clause is restricted to two literals, MAX2SAT requires us to find an assignment that maximizes the number of satisfied clauses. This problem is NP-hard (Johnson 1974). Similarly, if each clause is restricted to two literals, MIN2SAT requires us to find an assignment that minimizes the number of satisfied clauses. This problem is also NP-hard (Kohli et al. 1994). I rely on these two facts in the proofs of lemmas 1 and 2, respectively.

Lemma 1. *Assume 1, 2, 3, 4, 5, 6. If there exist $a, b \in Z$ such that*

$$u^2(a, a) + u^2(b, b) > u^2(a, b) + u^2(b, a) \tag{1}$$

then the choice correspondence ϕ^n is NP-hard.

Proof. I show that if the choice correspondence ϕ^n were tractable, then we could solve MAX2SAT in polynomial time. First, recall that the objective for MAX2SAT is

$$\max_{x_1,\ldots,x_n} \sum_{j=1}^{m} \mathbf{1}(c_j = \mathrm{true}) \tag{2}$$

where

$$c_j = (x_{j1} \vee x_{j2}) = (x_{j1} \wedge x_{j2}) \vee (\neg x_{j1} \wedge x_{j2}) \vee (x_{j1} \wedge \neg x_{j2}) \tag{3}$$

Our goal is to construct a menu H such that, given the solution $\phi(H)$, we could solve program (2) in poly(n, m) time. For every clause c_j, create three states j_1, j_2, j_3. For every variable x_i, create a dimension i. Let each submenu H_i consist of a two acts: h_i^T, which indicates $x_i = \mathrm{true}$, and h_i^F, which indicates $x_i = \mathrm{false}$. We defer the definition of these objects until later. For now, define a menu $H = H_1 \times \ldots \times H_n$ where an act h corresponds to an assignment x as described.

As written, the objective for BDT is

$$\max_{h \in H} \sum_{j=1}^{m} \sum_{k=1}^{3} u(h_1(j_k), \ldots, h_n(j_k))$$

Clearly, the following condition is sufficient for our purposes:

$$\sum_{k=1}^{3} u(h_1^{x_1}(j_k), \ldots, h_n^{x_n}(j_k)) \propto \underbrace{\mathbf{1}\left((x_{j1} \wedge x_{j2}) \vee (\neg x_{j1} \wedge x_{j2}) \vee (x_{j1} \wedge \neg x_{j2})\right)}_{\mathbf{1}(c_j = \mathrm{true})} \tag{4}$$

Our first task is to simplify this expression. On the left-hand side we refer to all n variables whereas on the right-hand side we refer only to two variables. If $x_i \notin c_j$ and $\neg x_i \notin c_j$, set $h_i(j_1) = h_i(j_2) = h_i(j_3) = e$ where e is the fixed element in assumption 5. Then, using symmetry, we have

$$\begin{aligned} u^n\left(h_1^{x_1}(j_k), \ldots, h_n^{x_n}(j_k)\right) &= u^n\left(e, \ldots, e, h_1^{x_{j1}}(j_k), h_n^{x_{j2}}(j_k)\right) \\ &= u^2\left(h_1^{x_{j1}}(j_k), h_n^{x_{j2}}(j_k)\right) \end{aligned}$$

The fact that we can treat e as a null element follows from

$$\begin{aligned} u^n(e, z_1, \ldots, z_{n-1}) &= u^2(e, \gamma^{-1}(u^{n-1}(z_1, \ldots, z_{n-1})) \\ &= \gamma\left(\gamma^{-1}(u^{n-1}(z_1, \ldots, z_{n-1}))\right) \\ &= u^{n-1}(z_1, \ldots, z_{n-1}) \end{aligned}$$

At this point, we can focus on the problem where $n = 2$, $m = 1$. Without loss of generality, let $c_j = (x_1 \vee x_2)$. This will simplify our notation. The left-hand side of condition (4) becomes

$$u^2\left(h_1^{x_1}(1), h_2^{x_2}(1)\right) + u^2\left(h_1^{x_1}(2), h_2^{x_2}(2)\right) + u^2\left(h_1^{x_1}(3), h_2^{x_2}(3)\right) \tag{5}$$

while the right-hand side becomes

$$\mathbf{1}\left((x_1 \wedge x_2) \vee (\neg x_1 \wedge x_2) \vee (x_1 \wedge \neg x_2)\right) \tag{6}$$

The reader may have gathered that I intend to somehow associate $u\left(h_1^{x_1}(1), h_2^{x_2}(1)\right)$ with $(x_1 \wedge x_2)$, $u\left(h_1^{x_1}(2), h_2^{x_2}(2)\right)$ with $(\neg x_1 \wedge x_2)$, and $u^2\left(h_1^{x_1}(3), h_2^{x_2}(3)\right)$ with $(x_1 \wedge \neg x_2)$. Indeed, this is true. This would be immediate if, for example, $u(z_1, z_2) = \min\{z_1, z_2\}$ since we could set

$$h_1^T(1) = 1; \quad h_1^T(2) = 0; \quad h_1^T(3) = 1; \quad h_2^T(1) = 1; \quad h_2^T(2) = 1; \quad h_2^T(3) = 0$$

and $h_i^F(k) = 1 - h_i^T(k)$ for all i, k. In that case, $\min\left(h_1^{x_1}(1), h_2^{x_2}(1)\right) = \mathbf{1}(x_1 \wedge x_2)$ and so forth. Since only one of the three expressions in the disjunctive normal form (DNF) (3) can be true at one time, the sum (5) would be one if and only if (3) were true, and zero otherwise. In other words, (5) would equal (6) and we would be done.

The case for more general utility functions u is not quite so straightforward. Roughly, we can think of u as a function of two literals (e.g. (x_1, x_2), $(\neg x_1, x_2)$, $(x_1, \neg x_2)$). Since u is increasing, it will assign high utility when both variables are true and low utility when both variables are false. However, it will also assign medium utility when one variable is true and the other false. Essentially, this implies a three-valued logic where "medium" corresponds to a statement that is not quite true but also not quite false. This makes it difficult to map our problem onto MAX2SAT, which deals with a traditional two-valued logic.

To clear this hurdle, we rely on our supermodularity condition (1) and asymmetry across the states $k = 1, \ldots, 3$. While it will not be true that $u\left(h_1^{x_1}(1), h_2^{x_2}(1)\right) \propto (x_1 \wedge x_2)$, our sufficient condition (4) will hold because it sums across these three states. Formally, define

$$h_1^T(1) = c; \quad h_1^T(2) = a; \quad h_1^T(3) = b; \quad h_2^T(1) = c; \quad h_2^T(2) = b; \quad h_2^T(3) = a$$

$$h_1^F(1) = a; \quad h_1^F(2) = b; \quad h_1^F(3) = a; \quad h_2^F(1) = a; \quad h_2^F(2) = a; \quad h_2^F(3) = b$$

By setting $b > c > a$, we devalue the satisfaction of the first expression in the DNF (3) relative to the latter two. Now, condition (4) is true if and only if

$$u\left(h_1^T(1), h_2^T(1)\right) + u\left(h_1^T(2), h_2^T(2)\right) + u\left(h_1^T(3), h_2^T(3)\right) = u(c,c) + u(a,b) + u(b,a) = B \tag{7}$$
$$u\left(h_1^T(1), h_2^F(1)\right) + u\left(h_1^T(2), h_2^F(2)\right) + u\left(h_1^T(3), h_2^F(3)\right) = u(c,a) + u(a,a) + u(b,b) = B \tag{8}$$
$$u\left(h_1^F(1), h_2^T(1)\right) + u\left(h_1^F(2), h_2^T(2)\right) + u\left(h_1^F(3), h_2^T(3)\right) = u(a,c) + u(b,b) + u(a,a) = B \tag{9}$$
$$u\left(h_1^F(1), h_2^F(1)\right) + u\left(h_1^F(2), h_2^F(2)\right) + u\left(h_1^F(3), h_2^F(3)\right) = u(a,a) + u(b,a) + u(a,b) = A \tag{10}$$

for some $B > A$. This is because the first three choices make the DNF (3) true, which demands a high value B, and the last one makes it false, which demands a low value A.

Conditions (8) and (9) are equivalent since u is symmetric across dimensions. Conditions (7) and (8) hold if and only if $\psi(c) = 0$ where

$$\psi(z) = u(z, z) - u(z, a) - u(a, a) - u(b, b) + 2u(a, b)$$

Note that $\psi(a) < 0$ and $\psi(b) \geq 0$ by assumption (1). Since u is continuous, it follows from the intermediate value theorem that there exists $c \in (a, b]$ such that $\psi(c) = 0$. Finally, the fact that $A < B$ follows from

$$u(a, a) + u(b, a) + u(a, b) < u(c, c) + u(a, b) + u(b, a) \iff u(a, a) < u(c, c)$$

which is true since $c > a$ and u is strictly increasing along the diagonal. This completes the reduction, since the consequences $h_i(j)$ can be defined similarly for other variables and states. Moreover, defining the menu H only requires us to define $O(nm)$ such consequences.

Notice that c depends on u^2 but not on any other aspect of the problem, including n, m, or the clauses $c_1, \dots, c_m$. This proof is non-constructive in the sense that I only prove the existence of a polynomial-time reduction (parameterized by c) from MAX2SAT to BDT, but do not provide an algorithm to find c itself. But whether c is easy or hard to compute is irrelevant for our purposes, so long as it does not need to be re-computed for different inputs to MAX2SAT. □

Lemma 2. *Assume 1, 2, 3, 4, 5, 6. If there exist $a, b \in Z$ such that*

$$u^2(a, a) + u^2(b, b) < u^2(a, b) + u^2(b, a) \tag{11}$$

then the choice correspondence ϕ^n is NP-hard.

Proof. I show that if the choice correspondence ϕ^n were tractable, then we could solve MIN2SAT in polynomial time. First, recall that the objective for MIN2SAT is

$$\max_{x_1,\dots,x_n} \sum_{j=1}^{m} \mathbf{1}(c_j = \text{false}) \tag{12}$$

Our goal is to construct a menu H such that, given the solution $\phi(H)$, we could solve program (12) in poly(n, m) time. My approach will be almost identical to the proof of lemma 1, so I skip ahead to the point of divergence.

Here, I define acts similarly to the previous lemma. However, because our objective is to minimize rather than maximize the number of satisfied clauses, we give false literals a high utility and true literals a low utility. Formally, define

$$h_1^T(1) = a; \quad h_1^T(2) = b; \quad h_1^T(3) = a; \quad h_2^T(1) = a; \quad h_2^T(2) = a; \quad h_2^T(3) = b$$

$$h_1^F(1) = c; \quad h_1^F(2) = a; \quad h_1^F(3) = b; \quad h_2^F(1) = c; \quad h_2^F(2) = b; \quad h_2^F(3) = a$$

Now, our (negatively proportional) analog to condition (4) is true if and only if

$$u\left(h_1^T(1), h_2^T(1)\right) + u\left(h_1^T(2), h_2^T(2)\right) + u\left(h_1^T(3), h_2^T(3)\right) = u(a,a) + u(b,a) + u(a,b) = A \quad (13)$$
$$u\left(h_1^T(1), h_2^F(1)\right) + u\left(h_1^T(2), h_2^F(2)\right) + u\left(h_1^T(3), h_2^F(3)\right) = u(a,c) + u(b,b,) + u(a,a) = A \quad (14)$$
$$u\left(h_1^F(1), h_2^T(1)\right) + u\left(h_1^F(2), h_2^T(2)\right) + u\left(h_1^F(3), h_2^T(3)\right) = u(c,a) + u(a,a) + u(b,b) = A \quad (15)$$
$$u\left(h_1^F(1), h_2^F(1)\right) + u\left(h_1^F(2), h_2^F(2)\right) + u\left(h_1^F(3), h_2^F(3)\right) = u(c,c) + u(a,b) + u(b,a) = B \quad (16)$$

for some $B > A$. This is because the first three choices make the DNF (3) true, which demands a low value A, and the last one makes it false, which demands a high value B.

Conditions (14) and (15) are equivalent since u is symmetric across dimensions. Conditions (13) and (14) hold if and only if $\psi(c) = 0$ where

$$\psi(z) = 2u(b,a) - u(b,b) - u(a,a) + u(a,a) - u(a,z)$$

Note that $\psi(a) = 2u(b,a) - u(b,b) - u(a,a) > 0$ and $\psi(b) = u(b,a) - u(b,b) \leq 0$ by assumption (11). Since u is continuous, it follows from the intermediate value theorem that there exists $c \in (a,b]$ such that $\psi(c) = 0$. Finally, the fact that $B > A$ follows from

$$u(c,c) + u(a,b) + u(b,a) > u(a,a) + u(b,a) + u(a,b) \iff u(c,c) > u(a,a)$$

which is true since $c > a$ and u is strictly increasing along the diagonal. This completes the reduction, for the same reasons as in the previous lemma. □

Armed with lemmas 1 and 2, it remains to show that

$$u^2(a,a) + u^2(b,b) = u^2(a,b) + u^2(b,a) \quad \forall a, b \quad (17)$$

implies that u^n is additively separable. I prove this by induction. First, rearrange (17) and apply symmetry to find

$$u^2(a,b) = \frac{1}{2}u^2(a,a) + \frac{1}{2}u^2(b,b) \quad \forall a, b$$

This combined with assumption 5 implies that

$$u^n(z_1, \ldots, z_n) = \frac{1}{2}u^2(z_1, z_1) + \frac{1}{2}u^2\left(\gamma^{-1}\left(u^{n-1}(z_2, \ldots, z_n)\right), \gamma^{-1}\left(u^{n-1}(z_2, \ldots, z_n)\right)\right) \quad (18)$$

Assume that u^{n-1} is additively separable. By definition,

$$\gamma(z) = u^2(z,e) = \frac{1}{2}u^2(z,z) + \frac{1}{2}u^2(e,e)$$

Rearrange and set $y = \gamma(z)$ to find

$$\frac{1}{2}u^2(\gamma^{-1}(y), \gamma^{-1}(y)) = y - \frac{1}{2}u^2(e,e)$$

It follows that condition (18) simplifies to

$$u^n(z_1, \dots, z_n) = \frac{1}{2}u^2(z_1, z_1) + u^{n-1}(z_2, \dots, z_n) - \frac{1}{2}u^2(e, e)$$

Since u^{n-1} is additively separable, so is u^n. This completes the proof.

References

Andersen, S., Cox, J. C., Harrison, G. W., Lau, M. I., Rutström, E. E., & Sadiraj, V. (2018, December). Asset integration and attitudes to risk: theory and evidence. *The Review of Economics and Statistics*, *100*(5), 816–830.

Brown, J. R., Kapteyn, A., Luttmer, E. F., Mitchell, O. S., & Samek, A. (2017, December). *Behavioral impediments to valuing annuities: evidence on the effects of complexity and choice bracketing* (Working Paper No. 24101). National Bureau of Economic Research.

Haisley, E., Mostafa, R., & Loewenstein, G. (2008, August). Myopic risk-seeking: the impact of narrow decision bracketing on lottery play. *Journal of Risk and Uncertainty*, *37*(1), 57–75.

Johnson, D. S. (1974). Approximation algorithms for combinatorial problems. *Journal of Computer and System Sciences*, *9*(3), 256–278.

Kohli, R., Krishnamurti, R., & Mirchandani, P. (1994, May). The minimum satisfiability problem. *SIAM J. Discret. Math. 7*(2), 275–283.

Rabin, M. & Weizsäcker, G. (2009, September). Narrow bracketing and dominated choices. *American Economic Review*, *99*(4), 1508–43.

Read, D., Loewenstein, G., & Rabin, M. (1999, December). Choice bracketing. *Journal of Risk and Uncertainty*, *19*(1), 171–197.

Savage, L. J. (1954). *The Foundations of Statistics*. Wiley Publications in Statistics.

Stracke, R., Kerschbamer, R., & Sunde, U. (2017). Coping with complexity: experimental evidence for narrow bracketing in multi-stage contests. *European Economic Review*, *98*, 264–281.