

ExpEcon Methods: Clever Elicitations & Choice Process Data

ECON 8877
P.J. Healy
Parts thanks to Kirby Nielsen

Updated 2026-01-28 at 01:17:19

Overview

1. MouseLab
2. Eye Tracking
3. fMRI
4. RTs
5. Krupka-Weber
6. Caplin Dean & Martin
7. Colin & Judd's audit study
8. Mistakes
9. Incompleteness
10. Strategy Method: conditional cooperators
11. Romero-Rosokha-Frechette
12. Advice/Muriel
13. Team Chat
14. Procedures

Search & Satisficing

- Two conditions ensure that data is consistent with satisficing:
 1. Subjects must always switch to higher value alternatives
 2. There must be some u^* such that search stops if and only if the utility of the chosen value is above u^*

- Design requires three things:
 1. Ranking of alternatives is clear to the experimenter
 - So we know that they switch to higher valued alternative
 2. But subjects still make mistakes
 - Still have to “search” through
 3. Can collect choice process data

Choice Objects

- Subjects choose between ‘sums’
four plus eight minus four
- Value of option is the value of the sum
- “Full information” ranking is obvious, but uncovering value takes effort
- 6 treatments
 - 2 x complexity (3 and 7 operations)
 - 3 x choice set size (10, 20, and 40 options)
- No time limit

Size 10, Complexity 3

Round
2 of 30

Current selection:

four plus eight minus four

Choose one:

- zero
- three plus five minus seven
- four plus two plus zero
- four plus three minus six
- four plus eight minus four
- three minus three plus one
- five plus one minus one
- eight plus two minus five
- three plus six minus five
- four minus two minus one
- five plus five minus one

Finished

Size 20, Complexity 7

- zero
- seven minus four minus two minus four minus two plus eleven minus four
- six plus five minus eight plus two minus nine plus one plus four
- seven minus two minus four plus three plus four minus three minus three
- seven plus five minus two minus two minus three plus zero minus two
- six plus seven plus six minus two minus six minus eight plus four
- six plus two plus five minus four minus two minus seven plus three
- six minus four minus one minus one plus five plus three minus six
- two plus six plus seven minus two minus four minus two plus zero
- two minus three minus five plus nine minus one plus five minus three
- three plus zero plus two plus zero plus one minus three minus one
- four plus three plus zero minus two plus three plus four minus ten
- seven plus two plus seven minus seven plus three minus two minus two
- three plus three minus two plus zero plus zero minus four plus five
- two minus two plus zero plus nine minus two minus one minus one
- three plus four minus three plus three minus four plus three minus four
- three plus five plus seven plus five minus two minus seven minus ten
- three plus six minus eight plus one plus two minus two plus zero
- three plus five plus zero plus four plus three minus four minus two
- eight minus one plus one minus four minus four minus five plus six

Results

Failure to choose optimal:

Size	Complexity	
	3	7
10	7%	24%
20	22%	56%
40	29%	65%

Results

Average loss (\$):

Size	Complexity	
	3	7
10	0.41	1.69
20	1.10	4.00
40	2.30	7.12

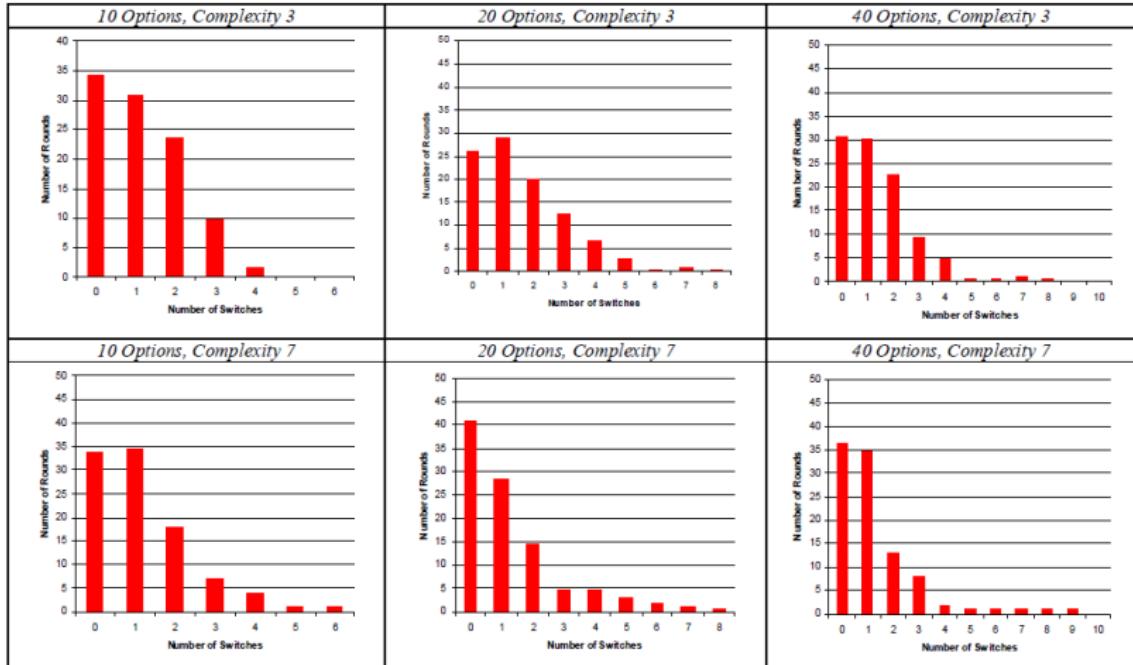
Results

- People frequently did not choose the best option, and left money on the table
- Presumably they do not prefer less money as things get bigger/more complex...
- Can satisficing explain it?

Eliciting Choice Process Data

1. Allow subjects to select any alternative at any time
 - Can change as often as they like
2. One random time between 0 and 120 seconds will be selected, unknown to subjects
 - Subjects will be paid whatever they had selected at this random time
 - Incentivizes subject to keep current best alternative selected
 - Treat sequence of selections as choice process data
3. Round ends in two ways
 - After 120 seconds
 - When subject presses the finish button
 - Discard the former

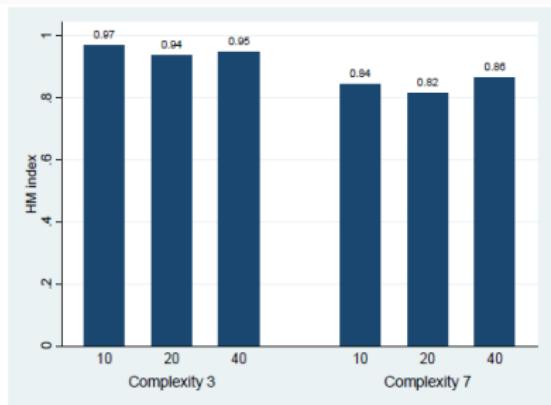
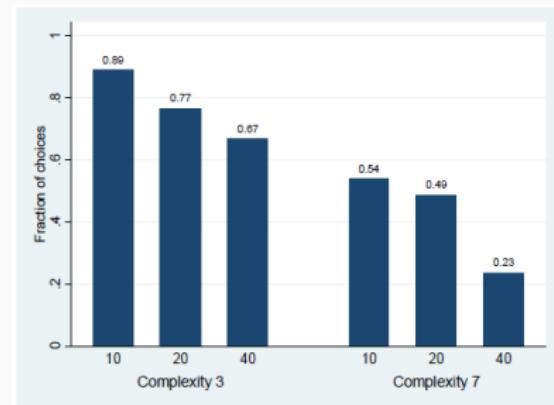
Benefit of Process Data



Condition 1

- Satisficing condition 1: Must always switch to higher-valued objects
- Graph the fraction of switches that satisfy this condition
- Compare to the fraction of choices that satisfy optimality

Condition 1



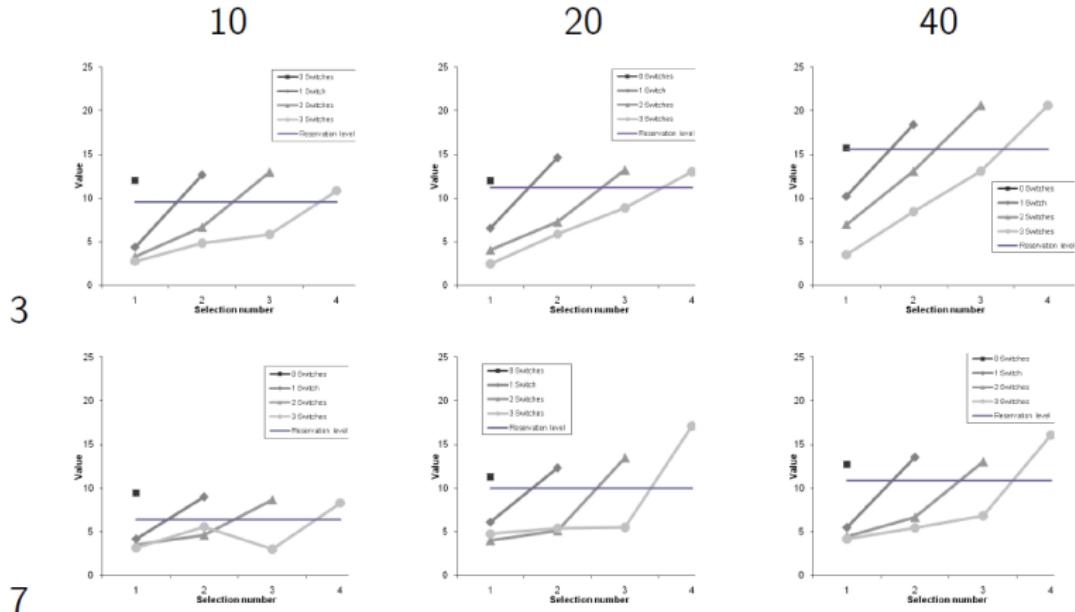
Satisficing

- Does seem to be that subjects are searching sequentially
- Switch when they find a better option

Satisficing

- Does seem to be that subjects are searching sequentially
- Switch when they find a better option
- But are they satisficing?
- Do they stop searching after encountering some value, u^* ?

Condition 1



Estimating Reservation Levels

Size	Complexity	
	3	7
10	9.54	6.36
20	11.18	9.95
40	15.54	10.84

Estimating Reservation Levels

- Reservation levels decrease with complexity
 - Predicted by theory
 - Increasing k
- Reservation levels increase with choice set size
 - Not predicted by theory

Awareness

- It looks “as if” individual are satisficing... do they know they are? How would we elicit this?
- Does it matter?