Decision Making Under Severe Uncertainty

SIPTA Summerschool 2010

Matthias C. M. Troffaes

Department of Mathematical Sciences
Durham University

3 September, 2010

Outline

Outline

A Very Simple Example

Example (Machinery, Overtime, or Nothing?)

A company makes a product, and believes in increasing future demand. The manager asks you, the decision expert, whether he should buy new machinery, use overtime, or do nothing. The upcoming year, demand can either increase or remain the same.

If we buy new machinery, then the profit at the end of the year will be 440 (in thousands of pounds) if demand increases, and 260 otherwise. On the other hand, if we use overtime, then the profit will be 420 if demand increases, and 420 otherwise. If we do nothing, profit will be 370.

According to our best current judgement, demand will increase with probability at least 0.5, and at most 0.8.

What advice can we give the manager?

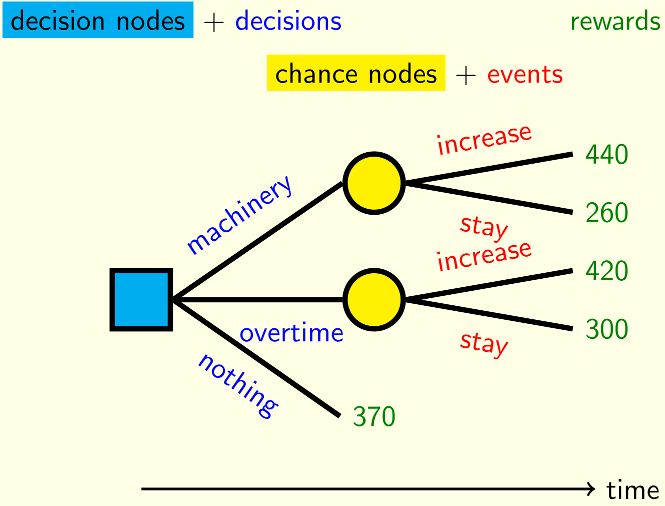
The Basic Elements of a Decision Problem

- decisions: $\{\text{buy new machinery}, \text{use overtime}, \text{do nothing}\}$
- events: $\{\text{demand increases}, \text{demand stays}\}$
- rewards: a monetary value, depending on decisions and events
- decision maker may have information about the events (e.g. bounds on the probabilities of the events)

Outline

Decision Trees

Graphical representation of decisions, events, and rewards:



Static and Sequential Decision Problems

The problem we are investigating is of a very simple type...

- we must make a single decision,
- which is followed by the occurence of an uncertain event, and
- which is in turn followed by a reward for us,
  depending on the decision we made, and the event that occurred

Informally...

Definition (Static Decision Problem)

Any decision tree that has

- a single decision node at its root,
- and no other decision node.

Definition (Sequential Decision Problem)

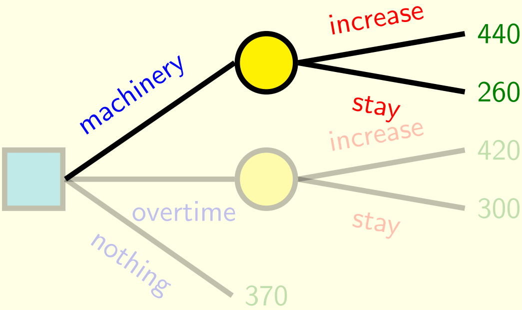
Any other decision tree.

Outline

The Problem of Choice: Gambles

Observation

in a static decision problem,
each decision branch corresponds to a gamble

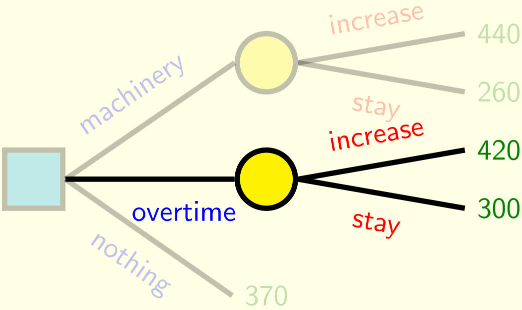


$$f_{\text{machinery}}(\text{increase}) = 440$$
$$f_{\text{machinery}}(\text{stay}) = 260$$

The Problem of Choice: Gambles

Observation

in a static decision problem,
each decision branch corresponds to a gamble



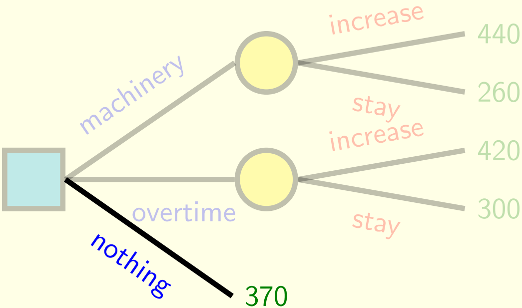
$$f_{\text{overtime}}(\text{increase}) = 420$$

$$f_{\text{overtime}}(\text{stay}) = 300$$

The Problem of Choice: Gambles

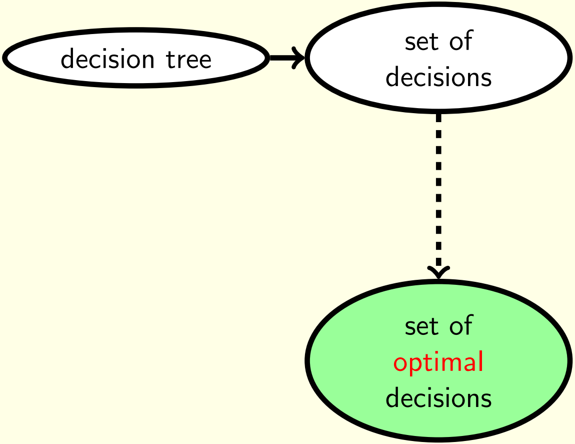
Observation

in a static decision problem,
each decision branch corresponds to a gamble

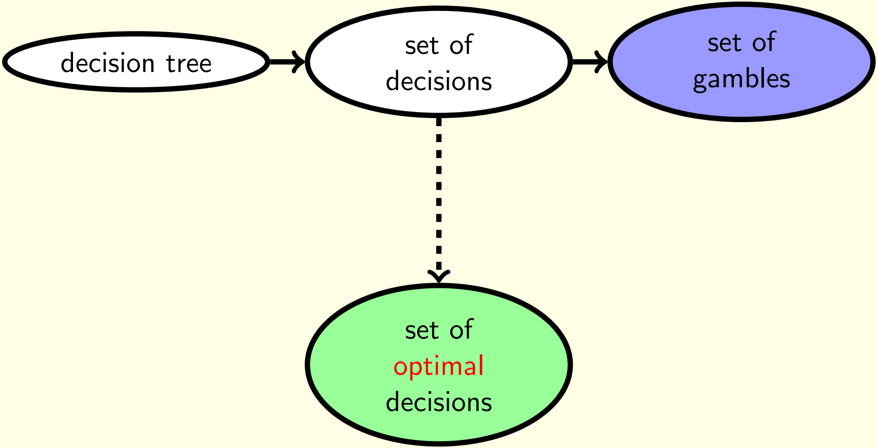


$$f_{\text{nothing}}(\text{increase}) = 370$$
$$f_{\text{nothing}}(\text{stay}) = 370$$

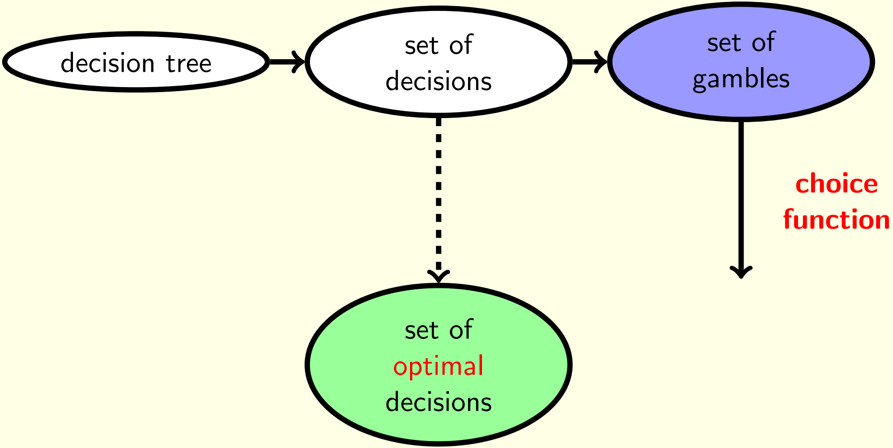
The Problem of Choice: Choice Function



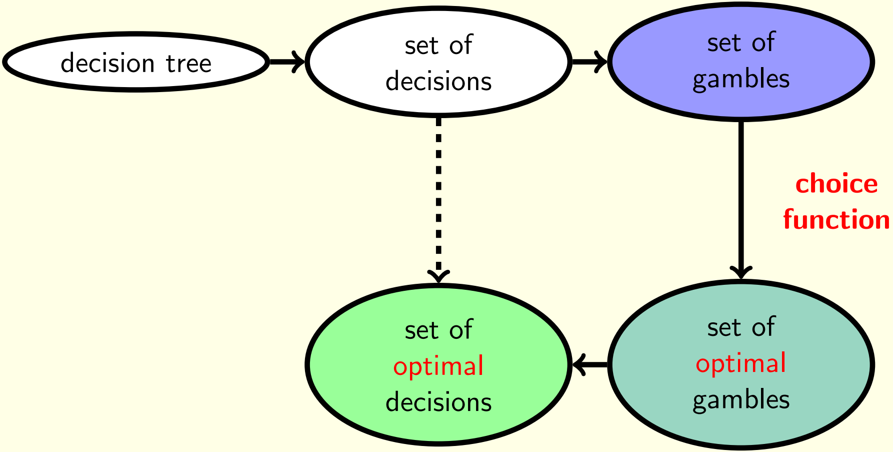
The Problem of Choice: Choice Function



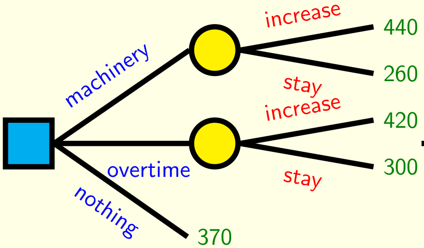
The Problem of Choice: Choice Function



The Problem of Choice: Choice Function



The Problem of Choice: Example



set of decisions
machinery
overtime
nothing

each row is a gamble

set of gambles

increase	stay
440	260
420	300
370	370

choice function

what is a good choice function, under severe uncertainty?

set of optimal gambles

increase	stay
420	300
370	370

set of optimal decisions
overtime
nothing

The Problem of Choice

- we know how to go from decision trees to gambles
- only one problem left to solve:

what is a good choice function?

- a standard choice function: maximize the expectation of the reward

however, as argued in the last few days...

- under severe uncertainty, we may not be able to identify a unique probability mass function p which describes our knowledge accurately
- still, we may be able to identify a lower prevision $\underline{P}$

what are good choice functions for lower previsions?

The Problem of Choice

Theorem (Approximate Representation Theorem)

For every lower prevision there is a finite set $\mathcal{P}$ of probability mass functions such that, for every gamble f on $\mathcal{X}$,

$$\underline{P}(f) \simeq \min_{p \in \mathcal{P}} \sum_{x \in \mathcal{X}} p(x) f(x)$$

Example (Machinery, Overtime, or Nothing?)

In our example, increase has probability at least 0.5 and at most 0.8, so

$$\mathcal{P} = \begin{array}{c|cc} & p_1 & p_2 \\ \hline \text{increase} & 0.5 & 0.8 \\ \text{stay} & 0.5 & 0.2 \end{array}$$ (each column is a probability mass function)

what are good choice functions
for finite sets of probability mass functions?

Recapitulating

		events		set of probabilities $\mathcal{P}$	
		increase	stay	p_1	p_2
events	increase			0.5	0.8
	stay			0.5	0.2
set of decisions	machinery	440	260		
	overtime	420	300		
	nothing	370	370		
		set of gambles			

which of the gambles are optimal?

Outline

Γ-Maximin

(Wald 1945 [15], Gilboa & Schmeidler 1989 [4])

Definition (Γ-Maximin Optimality Criterion)

Choose any gamble whose lower prevision is maximal.

Recipe (Γ-Maximin Optimality Criterion)

1. set up the table with gambles and probabilities
2. calculate the expectation of each gamble
   with respect to each probability mass function
3. calculate the minimum expectation of each gamble
4. choose the decision with the highest minimum expectation

Γ-Maximin: Example

Example (Machinery, Overtime, or Nothing)

	increase	stay	p_1	p_2	$\underline{P}$
increase			0.5	0.8	
stay			0.5	0.2	
machinery	440	260			
overtime	420	300			
nothing	370	370			
	(1)		(2)		(3) & (4)

Γ-Maximax

(Satia and Lave 1973 [10], probably others as well)

- Γ-maximin seems overly pessimistic; something more optimistic?

Definition (Γ-Maximax Optimality Criterion)

Choose any gamble whose *upper* prevision is maximal.

Recipe (Γ-Maximax Optimality Criterion)

1. set up the table with gambles and probabilities
2. calculate the expectation of each gamble
   with respect to each probability mass function
3. calculate the *maximum* expectation of each gamble
4. choose the decision with the highest maximum expectation

Γ-Maximax: Example

Example (Machinery, Overtime, or Nothing)

	increase	stay	p_1	p_2	$\overline{P}$
increase			0.5	0.8	
stay			0.5	0.2	
machinery	440	260			
overtime	420	300			
nothing	370	370			
	(1)		(2)		(3) & (4)

Interval Dominance

(Satia and Lave 1973 [10], Kyburg 1983 [8], *many* others)

- get every reasonable option (from pessimistic to optimistic) at once?

Definition (Interval Dominance Optimality Criterion)

Choose any gamble whose upper prevision
exceeds the largest lower prevision.

Recipe (Interval Dominance Optimality Criterion)

1. set up the table with gambles and probabilities
2. calculate the expectation of each gamble
   with respect to each probability mass function
3. calculate the *minimum and maximum* expectation of each gamble
4. choose the decisions whose maximum expectation
   exceeds the overall largest minimum expectation

Interval Dominance: Example

Example (Machinery, Overtime, or Nothing)

	increase	stay	p_1	p_2	$\underline{P}$	$\overline{P}$
increase			0.5	0.8		
stay			0.5	0.2		
machinery	440	260				
overtime	420	300				
nothing	370	370				
	(1)		(2)		(3) & (4)	

Maximality

- exploits the behavioural interpretation of lower previsions
- refines interval dominance (see Exercise 3 later!)

Definition (Partial Ordering Determined by a Lower Prevision)

A lower prevision determines a partial ordering between gambles

$$f \succ g \text{ whenever } \underline{P}(f - g) > 0$$

(willing to pay a small amount in order to trade g for f)
($f - g + \epsilon$ is desirable for some $\epsilon > 0$)

Maximality

(Condorcet 1785 [2], Sen 1977 [13], Walley 1991 [16])

Definition (Maximality Optimality Criterion)

Choose any gamble which is undominated with respect to $\succ$.

i.e. any gamble f such that $g \not\succ f$ for all relevant gambles g
i.e. any gamble f such that $\underline{P}(g - f) \leq 0$ for all relevant gambles g

Recipe (Maximality Optimality Criterion)

1. set up the table with gambles and probabilities
2. set up the table with differences between gambles
3. calculate the sign of the expectation of each difference with respect to each probability mass function
4. calculate the minimum sign of the expectation of each difference
5. choose the decisions whose differences are all negative or zero

(if you do this cleverly, you may not need to consider *every* difference!)

Maximality: Example

Example (Machinery, Overtime, or Nothing)

	increase	stay	p_1	p_2	$\underline{P}$
increase			0.5	0.8	
stay			0.5	0.2	
machinery	440	260			
overtime	420	300			
nothing	370	370			
	(1)		(3)		(4)

	machinery	overtime	nothing
−machinery	0		
−overtime		0	
−nothing			0

(2) & (5)

Outline

A Simple Example

(adapted from Kikuti et al. [7, Fig. 2])

Example (The Oil Wildcatter)

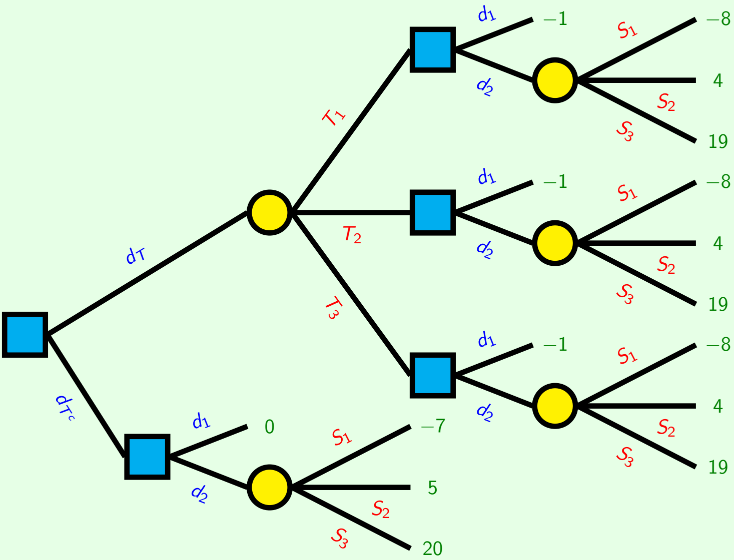
An oil wildcatter must decide whether to drill for oil (d_2) or not (d_1). Drilling costs 7 and provides a return of 0, 12, or 27 depending on the richness of the site. The events S_1 to S_3 represent the different yields, with S_1 being the least profitable and S_3 the most. The subject may pay 1 to test the site before deciding whether to drill; this gives one of three results T_1 to T_3, where T_1 is the most pessimistic and T_3 the most optimistic. (All rewards in units of $10000.)

$\mathcal{P} =$

	p_1	p_2	p_3	p_4	p_5	p_6
$T1\&S1$	0.18	0.18	0.18	0.18	0.26	0.40
$T1\&S2$	0.06	0.06	0.06	0.06	0.20	0.06
$T1\&S3$	0.03	0.03	0.03	0.03	0.03	0.03
$T2\&S1$	0.03	0.03	0.03	0.23	0.03	0.03
$T2\&S2$	0.18	0.18	0.40	0.23	0.18	0.18
$T2\&S3$	0.03	0.03	0.03	0.00	0.03	0.03
$T3\&S1$	0.03	0.03	0.03	0.03	0.03	0.03
$T3\&S2$	0.06	0.20	0.06	0.06	0.06	0.06
$T3\&S3$	0.40	0.26	0.18	0.18	0.18	0.18

Should the wildcatter pay for the test or not? Then, should he drill or not?

A Simple Example: Decision Tree



Outline

Normal Form Decision

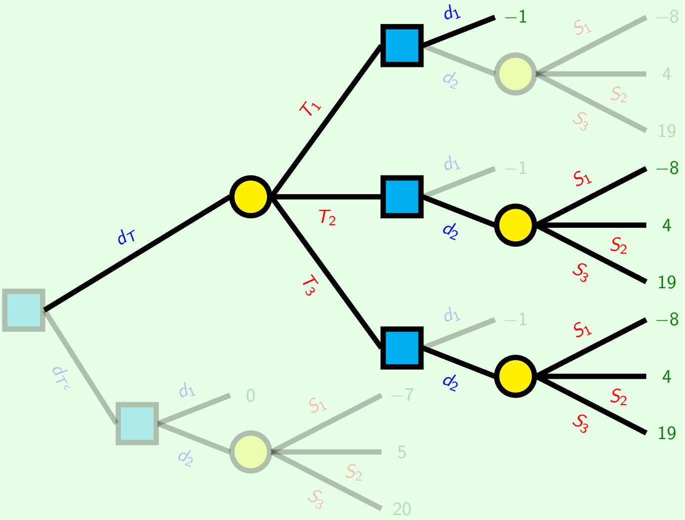
Definition

A normal form decision fixes at every decision node exactly one decision.

Observation

in a sequential decision problem,
each normal form decision corresponds to a gamble

Normal Form Decision: Example



	T_1S_1	T_1S_2	T_1S_3	T_2S_1	T_2S_2	T_2S_3	T_3S_1	T_3S_2	T_3S_3
$d_T(T_1d_1)(T_2d_2)(T_3d_2)$	-1	-1	-1	-8	4	19	-8	4	19

Normal Form Decision: Example

- We can find the gamble this for every normal form decision:

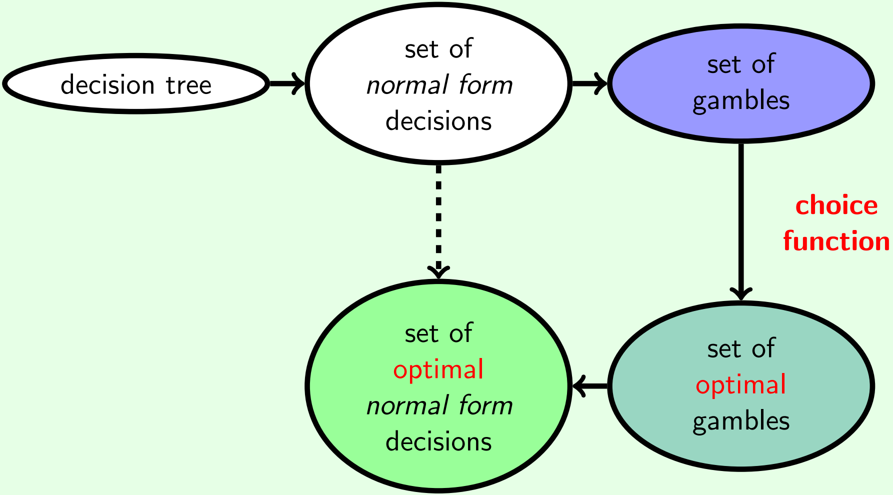
	T_1S_1	T_1S_2	T_1S_3	T_2S_1	T_2S_2	T_2S_3	T_3S_1	T_3S_2	T_3S_3
$d_T d_1$	-1	-1	-1	-1	-1	-1	-1	-1	-1
$d_T(T_1d_1)(T_2d_1)(T_3d_2)$	-1	-1	-1	-1	-1	-1	-8	4	19
$d_T(T_1d_1)(T_2d_2)(T_3d_1)$	-1	-1	-1	-8	4	19	-1	-1	-1
$d_T(T_1d_1)(T_2d_2)(T_3d_2)$	-1	-1	-1	-8	4	19	-8	4	19
$d_T(T_1d_2)(T_2d_1)(T_3d_1)$	-8	4	19	-1	-1	-1	-1	-1	-1
$d_T(T_1d_2)(T_2d_1)(T_3d_2)$	-8	4	19	-1	-1	-1	-8	4	19
$d_T(T_1d_2)(T_2d_2)(T_3d_1)$	-8	4	19	-8	4	19	-1	-1	-1
$d_T d_2$	-8	4	19	-8	4	19	-8	4	19
$d_{T^c} d_1$	0	0	0	0	0	0	0	0	0
$d_{T^c} d_2$	-7	5	20	-7	5	20	-7	5	20

- . . . so, we have
  - a set of normal form decisions
  - a gamble for each normal form decision
  - a credal set, so we can calculate lower/upper previsions of gambles and of their differences

. . . everything keeps working as before!!
(except that now we have normal form decisions, instead of simple decisions)

Outline

The Problem of Sequential Choice in Normal Form



Example: Normal Form Solution

```
>>> opt = OptLowPrevMaxMin(lpr)
>>> list(opt(gambles))
[-7.0, 5.0, 20.0, -7.0, 5.0, 20.0, -7.0, 5.0, 20.0]
>>> opt = OptLowPrevMaxMax(lpr)
>>> list(opt(gambles))
[-7.0, 5.0, 20.0, -7.0, 5.0, 20.0, -7.0, 5.0, 20.0]
>>> opt = OptLowPrevMaxInterval(lpr)
>>> list(opt(gambles))
[-1.0, -1.0, -1.0, -1.0, -1.0, -1.0, -8.0, 4.0, 19.0]
[-1.0, -1.0, -1.0, -8.0, 4.0, 19.0, -8.0, 4.0, 19.0]
[-8.0, 4.0, 19.0, -1.0, -1.0, -1.0, -8.0, 4.0, 19.0]
[-8.0, 4.0, 19.0, -8.0, 4.0, 19.0, -8.0, 4.0, 19.0]
[-7.0, 5.0, 20.0, -7.0, 5.0, 20.0, -7.0, 5.0, 20.0]
>>> opt = OptLowPrevMax(lpr)
>>> list(opt(gambles))
[-1.0, -1.0, -1.0, -8.0, 4.0, 19.0, -8.0, 4.0, 19.0]
[-7.0, 5.0, 20.0, -7.0, 5.0, 20.0, -7.0, 5.0, 20.0]
```

(aside, this shows very nicely that you should always try maximality, particularly when interval dominance gives you a lot of optimal gambles where it does not seem to make sense!)

Why **Not** Solve Sequential Problems This Way?

- even for this very simple problem, the number of normal form decisions was already pretty large
- a lot of calculations required, particularly with maximality
- for larger problems, not even manageable by computer

the good news...

- there are backward induction schemes that can deal with arbitrary choice functions

the bad news...

- these algorithms only yields the actual optimal normal form solution if the choice function satisfies rather restrictive properties

but not all is lost!

- maximality satisfies these properties!!
- however (almost) no other criterion does

Outline

Backward Induction

idea: use solutions of subtrees to eliminate options in the full tree

For general choice functions several algorithms have been proposed:

- Seidenfeld 1988 [11] [12] (extensive form)
- Harmanec 1999 [5] (extensive form)
- De Cooman & Troffaes 2005 [3] (normal form)
- Kikuti et. al 2005 [7] (apparently, normal form)
- Huntley & Troffaes 2008 [6] (normal form)

Normal Form Backward Induction

(Huntley & Troffaes, 2008 [6])

Recipe (Normal Form Backward Induction)

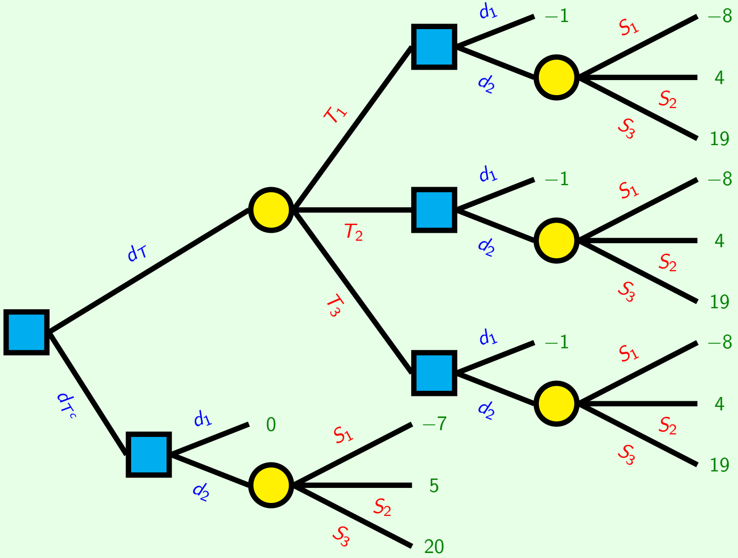
reiterate these steps, until all nodes have been dealt with:

1. find normal form decisions, and corresponding gambles, at final nodes
2. apply choice function conditional on past events, on each set of gambles
3. replace each final node by its set of optimal gambles

Theorem

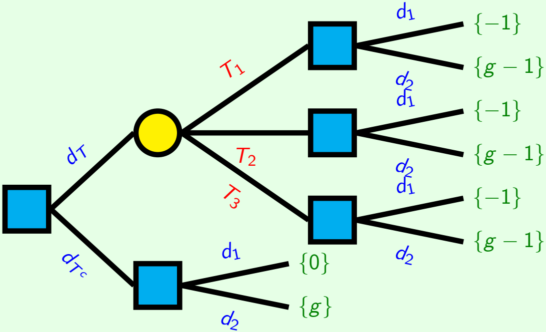
If you do this using maximality as optimality criterion, then you are guaranteed to end up with the optimal normal form decisions at the root.

Normal Form Backward Induction: Example



Normal Form Backward Induction: Example (Stage 1)

1. normal form decisions, and corresponding gambles: trivial
2. apply conditional choice: trivial (single gamble for each node!)
3. replace nodes with sets of optimal gambles



with $g = \begin{array}{ccc} S_1 & S_2 & S_3 \\ \hline -7 & 5 & 20 \end{array}$

Normal Form Backward Induction: Example (Stage 2)

T_1 branch

1. normal form decisions and gambles

	S_1	S_2	S_3
d_1	-1	-1	-1
d_2	-8	4	19

2. maximality conditional on T_1
   - apply the definition of conditional probability on each of the given unconditional probabilities

$\mathcal{P}|T_1 =$

	p_1	p_2	p_3
S_1	0.531	0.667	0.817
S_2	0.408	0.222	0.122
S_3	0.061	0.111	0.061

   - apply maximality using the resulting conditional probabilities

	S_1	S_2	S_3	p_1	p_2	p_3	$\underline{P}$
S_1				0.531	0.667	0.817	
S_2				0.408	0.222	0.122	
S_3				0.061	0.111	0.061	
d_1	-1	-1	-1				
d_2	-8	4	19				
$d_2 - d_1$							
$d_1 - d_2$							

	d_1	d_2
$-d_1$	0	
$-d_2$		0

Normal Form Backward Induction: Example (Stage 2)

T_2 branch

1. normal form decisions and gambles

	S_1	S_2	S_3
d_1	-1	-1	-1
d_2	-8	4	19

2. maximality conditional on T_2
   - apply the definition of conditional probability on each of the given unconditional probabilities

$$\mathcal{P}|T_2 = \begin{array}{c|ccc} & p_1 & p_2 & p_3 \\ \hline S_1 & 0.065 & 0.125 & 0.500 \\ S_2 & 0.870 & 0.750 & 0.500 \\ S_3 & 0.065 & 0.125 & 0.000 \end{array}$$

   - apply maximality using the resulting conditional probabilities

	S_1	S_2	S_3	p_1	p_2	p_3	$\underline{P}$
S_1				0.065	0.125	0.500	
S_2				0.870	0.750	0.500	
S_3				0.065	0.125	0.000	
d_1	-1	-1	-1				
d_2	-8	4	19				
d_2-d_1							
d_1-d_2							

	d_1	d_2
$-d_1$	0	
$-d_2$		0

Normal Form Backward Induction: Example (Stage 2)

T_3 branch

1. normal form decisions and gambles

	S_1	S_2	S_3
d_1	-1	-1	-1
d_2	-8	4	19

2. maximality conditional on T_3
   - apply the definition of conditional probability on each of the given unconditional probabilities

$\mathcal{P}|T_3 =$

	p_1	p_2	p_3
S_1	0.061	0.111	0.061
S_2	0.408	0.222	0.122
S_3	0.531	0.667	0.817

   - apply maximality using the resulting conditional probabilities

	S_1	S_2	S_3	p_1	p_2	p_3	$\underline{P}$
S_1				0.061	0.111	0.061	
S_2				0.408	0.222	0.122	
S_3				0.531	0.667	0.817	
d_1	-1	-1	-1				
d_2	-8	4	19				
d_2-d_1							
d_1-d_2							

	d_1	d_2
$-d_1$	0	
$-d_2$		0

Normal Form Backward Induction: Example (Stage 2)

d_{T^c} branch

1. normal form decisions and gambles

	T_1S_1	T_1S_2	T_1S_3	T_2S_1	T_2S_2	T_2S_3	T_3S_1	T_3S_2	T_3S_3
d_1	0	0	0	0	0	0	0	0	0
d_2	−7	5	20	−7	5	20	−7	5	20

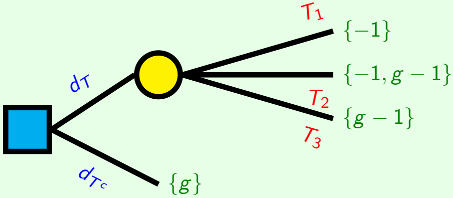
2. maximality
   - no past events, so use unconditional probabilities
   - apply maximality as usual

	T_1S_1	T_1S_2	T_1S_3	T_2S_1	T_2S_2	T_2S_3	T_3S_1	T_3S_2	T_3S_3	p_1	p_2	p_3	p_4	p_5	p_6	$\underline{P}$
$T1\&S1$										0.18	0.18	0.18	0.18	0.26	0.40	
$T1\&S2$										0.06	0.06	0.06	0.06	0.20	0.06	
$T1\&S3$										0.03	0.03	0.03	0.03	0.03	0.03	
$T2\&S1$										0.03	0.03	0.03	0.23	0.03	0.03	
$T2\&S2$										0.18	0.18	0.40	0.23	0.18	0.18	
$T2\&S3$										0.03	0.03	0.03	0.00	0.03	0.03	
$T3\&S1$										0.03	0.03	0.03	0.03	0.03	0.03	
$T3\&S2$										0.06	0.20	0.06	0.06	0.06	0.06	
$T3\&S3$										0.40	0.26	0.18	0.18	0.18	0.18	
d_1	0	0	0	0	0	0	0	0	0							
d_2	−7	5	20	−7	5	20	−7	5	20							
$d_2 - d_1$																
$d_1 - d_2$																

	d_1	d_2
$-d_1$	0	
$-d_2$		0

Normal Form Backward Induction: Example (Stage 2)

3. replace nodes with sets of optimal gambles



Normal Form Backward Induction: Example (Stage 3)

d_T branch

1. normal form decisions and gambles

	T_1S_1	T_1S_2	T_1S_3	T_2S_1	T_2S_2	T_2S_3	T_3S_1	T_3S_2	T_3S_3
$s_1=(T_1d_1)(T_2d_1)(T_3d_2)$	-1	-1	-1	-1	-1	-1	-8	4	19
$s_2=(T_1d_1)(T_2d_2)(T_3d_2)$	-1	-1	-1	-8	4	19	-8	4	19

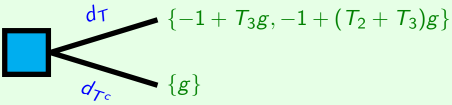
2. maximality
   - no past events, so use unconditional probabilities
   - apply maximality as usual

	T_1S_1	T_1S_2	T_1S_3	T_2S_1	T_2S_2	T_2S_3	T_3S_1	T_3S_2	T_3S_3	p_1	p_2	p_3	p_4	p_5	p_6	$\underline{P}$
$T1\&S1$										0.18	0.18	0.18	0.18	0.26	0.40	
$T1\&S2$										0.06	0.06	0.06	0.06	0.20	0.06	
$T1\&S3$										0.03	0.03	0.03	0.03	0.03	0.03	
$T2\&S1$										0.03	0.03	0.03	0.23	0.03	0.03	
$T2\&S2$										0.18	0.18	0.40	0.23	0.18	0.18	
$T2\&S3$										0.03	0.03	0.03	0.00	0.03	0.03	
$T3\&S1$										0.03	0.03	0.03	0.03	0.03	0.03	
$T3\&S2$										0.06	0.20	0.06	0.06	0.06	0.06	
$T3\&S3$										0.40	0.26	0.18	0.18	0.18	0.18	
s_1	-1	-1	-1	-1	-1	-1	-8	4	19							
s_2	-1	-1	-1	-8	4	19	-8	4	19							
s_2-s_1																
s_1-s_2																

	s_1	s_2
$-s_1$	0	
$-s_2$		0

Normal Form Backward Induction: Example (Stage 3)

3. replace nodes with sets of optimal gambles



Normal Form Backward Induction: Example (Stage 4)

root node

1. normal form decisions and gambles

	T_1S_1	T_1S_2	T_1S_3	T_2S_1	T_2S_2	T_2S_3	T_3S_1	T_3S_2	T_3S_3
$s_1=(T_1d_1)(T_2d_1)(T_3d_2)$	-1	-1	-1	-1	-1	-1	-8	4	19
$s_2=(T_1d_1)(T_2d_2)(T_3d_2)$	-1	-1	-1	-8	4	19	-8	4	19
d_2	-7	5	20	-7	5	20	-7	5	20

2. maximality
   - no past events, so use unconditional probabilities
   - apply maximality as usual

Normal Form Backward Induction: Example (Stage 4)

root node

	T_1S_1	T_1S_2	T_1S_3	T_2S_1	T_2S_2	T_2S_3	T_3S_1	T_3S_2	T_3S_3	p_1	p_2	p_3	p_4	p_5	p_6	$\underline{P}$
$T1\&S1$										0.18	0.18	0.18	0.18	0.26	0.40	
$T1\&S2$										0.06	0.06	0.06	0.06	0.20	0.06	
$T1\&S3$										0.03	0.03	0.03	0.03	0.03	0.03	
$T2\&S1$										0.03	0.03	0.03	0.23	0.03	0.03	
$T2\&S2$										0.18	0.18	0.40	0.23	0.18	0.18	
$T2\&S3$										0.03	0.03	0.03	0.00	0.03	0.03	
$T3\&S1$										0.03	0.03	0.03	0.03	0.03	0.03	
$T3\&S2$										0.06	0.20	0.06	0.06	0.06	0.06	
$T3\&S3$										0.40	0.26	0.18	0.18	0.18	0.18	
s_1	-1	-1	-1	-1	-1	-1	-8	4	19							
s_2	-1	-1	-1	-8	4	19	-8	4	19							
d_2	-7	5	20	-7	5	20	-7	5	20							
$s_2 - s_1$																
$d_2 - s_1$																
$s_1 - s_2$																
$d_2 - s_2$																
$s_1 - d_2$																
$s_2 - d_2$																

	s_1	s_2	d_2
$-s_1$	0		
$-s_2$		0	
$-d_2$			0

Normal Form Backward Induction: Example (Stage 4)

3. replace nodes with sets of optimal gambles

$$\{-1 + (T_2 + T_3)g, g\}$$

- these gambles correspond to the normal form decisions:

$$d_T(T_1 d_1)(T_2 d_2)(T_3 d_2),\ d_{T^c} d_2$$

- we have solved the problem!!

Outline

What's Next...

Things I have **not** told you today:

- relationships between choice functions [14]
- more choice functions
  - E-admissibility [9]
  - info-gap, satisficing [1]
  - extensive form methods [11] [12] [5]
- nasty properties of some choice functions:
  do not rely on your general intuition about optimality
- clever things you can do when your decision problem has additional structure

we've only scratched the surface, but hopefully you have learnt something, and have some idea of how decisions could be made under severe uncertainty, and where to look further

References I

Yakov Ben-Haim.
Info-Gap Decision Theory: Decisions Under Severe Uncertainty.
Academic Press, 2001.

Marquis de Condorcet.
Essai sur l'Application de l'Analyse à la Probabilité des Décisions Rendues à la Pluralité des Voix.
L'Imprimerie Royale, Paris, 1785.

Gert de Cooman and Matthias C. M. Troffaes.
Dynamic programming for deterministic discrete-time systems with uncertain gain.
International Journal of Approximate Reasoning, 39(2–3):257–278, Jun 2005.

References II

Itzhak Gilboa and David Schmeidler.
Maxmin expected utility with non-unique prior.
Journal of Mathematical Economics, 18(2):141–153, 1989.

David Harmanec.
A generalization of the concept of Markov decision process to imprecise probabilities.
pages 175–182, Ghent, 1999. Imprecise Probabilities Project.

Nathan Huntley and Matthias C. M. Troffaes.
An efficient normal form solution to decision trees with lower previsions.
In Didier Dubois, M. Asunción Lubiano, Henri Prade, María Ángeles Gil, Przemyslaw Grzegorzewski, and Olgierd Hryniewicz, editors, *Soft Methods for Handling Variability and Imprecision*, Advances in Soft Computing, pages 419–426. Springer, Sep 2008.

References III

Daniel Kikuti, Fabio G. Cozman, and Cassio P. de Campos.
Partially ordered preferences in decision trees: Computing strategies with imprecision in probabilities.
In Ronen Brafman and Ulrich Junker, editors, *Multidisciplinary IJCAI-05 Workshop on Advances in Preference Handling*, pages 118–123, 2005.

H. E. Kyburg.
Rational belief.
Technical report, University of Rochester, 1983.

Isaac Levi.
The Enterprise of Knowledge. An Essay on Knowledge, Credal Probability, and Chance.
MIT Press, Cambridge, 1980.

References IV

Jay K. Satia and Jr. Roy E. Lave.
Markovian decision processes with uncertain transition probabilities.
Operations Research, 21(3):728–740, 1973.

T. Seidenfeld.
Decision theory without 'independence' or without 'ordering': What is the difference?
Economics and Philosophy, 4:267–290, 1988.

Teddy Seidenfeld.
A contrast between two decision rules for use with (convex) sets of probabilities: Gamma-maximin versus E-admissibility.
Synthese, 140(1–2):69–88, May 2004.

Amartya Sen.
Social choice theory: A re-examination.
Econometrica, 45(1):53–89, January 1977.

References V

Matthias C. M. Troffaes.
Decision making under uncertainty using imprecise probabilities.
International Journal of Approximate Reasoning, 45(1):17–29, May 2007.

Abraham Wald.
Statistical decision functions which minimize the maximum risk.
The Annals of Mathematics, 46(2):265–280, 1945.

Peter Walley.
Statistical Reasoning with Imprecise Probabilities.
Chapman and Hall, London, 1991.

Outline

Exercise 1: Machinery, Overtime, or Nothing?

Consider again the same very simple example. We have done additional market research, and we now know that demand will increase with probability at least 0.6, and at most 0.65.

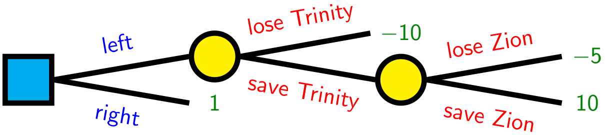
What advice can we give the manager now? Investigate with each optimality criterion.

Hint: $\mathcal{P} =$

	p_1	p_2
increase	0.6	0.65
stay	0.4	0.35

Exercise 2: Saving Zion (Or Maybe Not?)

There are two doors. The door to your right leads to the Source and the salvation of Zion. The door to your left leads back to the Matrix, to her... and to the end of your species. As you adequately put, the problem is choice. But we already know what you are going to do, don't we?

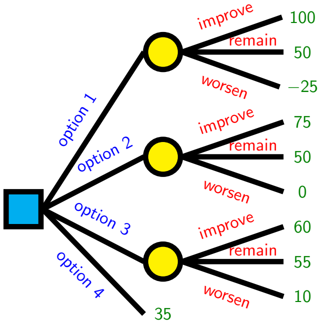


$$\mathcal{P} = \begin{array}{c|ccc} & p_1 & p_2 & p_3 \\ \hline \text{lose Trin} & 0.1 & 0.4 & 0.3 \\ \text{save Trin \& lose Zion} & 0.45 & 0.3 & 0.2 \\ \text{save Trin \& save Zion} & 0.45 & 0.3 & 0.5 \end{array}$$

Left, or right? Investigate with your favorite optimality criterion.

Exercise 3: A Risky Investment

You have the option to invest some money. The market can either improve, remain, or worsen. The set of probabilities for your lower prevision are tabulated below. You have the choice between 4 options, summarized in the decision tree below.



$\mathcal{P} =$

	p_1	p_2
improve	0.0	0.3
remain	0.6	0.3
worsen	0.4	0.4

Which options should you definitely not consider? First consider interval dominance, then consider maximality. Which of these two criteria gives the better answer?

Exercise 4

The following is again the very simple example, solved in Python, for Γ-maximin, Γ-maximax, interval dominance, and maximality, respectively:

```
>>> from improb.lowprev.lowpoly import LowPoly
>>> from improb.decision.opt import *
>>> lpr = LowPoly(2, credalset=[[.5, .5], [.8, .2]])
>>> gambles = [[440, 260], [420, 300], [370, 370]]
>>> opt = OptLowPrevMaxMin(lpr)
>>> list(opt(gambles))
[[370, 370]]
>>> opt = OptLowPrevMaxMax(lpr)
>>> list(opt(gambles))
[[440, 260]]
>>> opt = OptLowPrevMaxInterval(lpr)
>>> list(opt(gambles))
[[440, 260], [420, 300], [370, 370]]
>>> opt = OptLowPrevMax(lpr)
>>> list(opt(gambles))
[[440, 260], [420, 300], [370, 370]]
```

Solve Exercises 1, 2, and 3 similarly.

[Hint: The first argument of `LowPoly` denotes the number of events.]

Exercise 5 (*)

Let g be any gamble on $\mathcal{X}$, with lower prevision L and upper prevision U. Let c be any constant. Suppose you have the choice between the uncertain gain g, or the certain gain c.

Under each of the criteria, determine which of g or c (or both!) are optimal, under the following circumstances:

- $c < L$
- $L \leq c \leq U$
- $c > U$

In Exercise 3, option 4 corresponds to an investment without risk, as it yields the value $c = 35$ independently of the market, however, we found that this value was too low relative to the other options to be optimal.

For what values for c would you change your mind? Again, investigate this using each of the criteria.

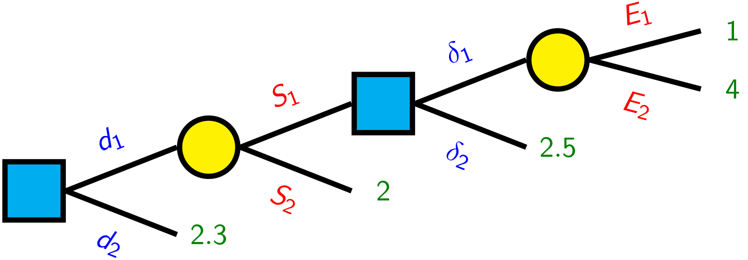
Exercise 6

State one advantage, and one disadvantage, of solving a sequential decision problem by normal form backward induction, compared to solving it by normal form.

Can you think of a situation in which normal form backward induction would be less efficient than normal form?

Exercise 7

Solve the following sequential decision problem for maximality, using either normal form, or normal form backward induction.



$\mathcal{P} =$

	p_1	p_2
$S_1 E_1$	0.2	0.1
$S_1 E_2$	0.3	0.4
S_2	0.5	0.5

[Hint: $\mathcal{P}|S_1 =$

	p_1	p_2
E_1	0.4	0.2
E_2	0.6	0.8

]

Exercise 8 (*)

The following program solves the previous exercise in Python for maximality, by normal form backward induction:

```
>>> from improb.lowprev.lowpoly import LowPoly
>>> from improb.decision.opt import *
>>> from improb.decision.tree import *
>>> pspace = PSpace(["S1", "S2"], ["E1", "E2"])
>>> E1 = pspace.make_event(["S1", "S2"], ["E1"], name="E1")
>>> E2 = pspace.make_event(["S1", "S2"], ["E2"], name="E2")
>>> S1 = pspace.make_event(["S1"], ["E1", "E2"], name="S1")
>>> S2 = pspace.make_event(["S2"], ["E1", "E2"], name="S2")
```

(continued on next slide)

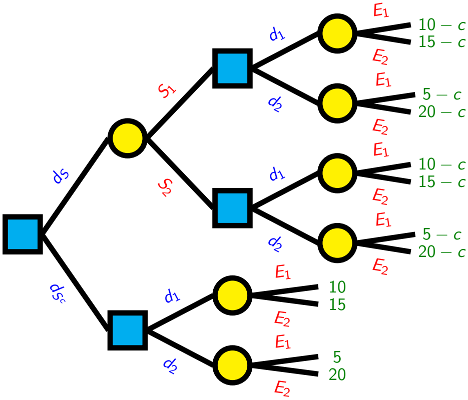
(continued from previous slide)

```
>>> t3 = Chance(pspace)
>>> t3[E1] = 1
>>> t3[E2] = 4
>>> t2 = Decision()
>>> t2["delta1"] = t3
>>> t2["delta2"] = 2.5
>>> t1 = Chance(pspace)
>>> t1[S1] = t2
>>> t1[S2] = 2
>>> t0 = Decision()
>>> t0["d1"] = t1
>>> t0["d2"] = 2.3
>>> print(t0)
>>> lpr = LowPoly(pspace)
>>> lpr.set_lower(S1 & E1, 0.1)
>>> lpr.set_lower(S1 & E2, 0.3)
>>> lpr.set_precise(S2, 0.5)
>>> opt = OptLowPrevMax(lpr)
>>> for gamble, normal_form_decision in t0.get_norm_back_opt(opt):
...     print(normal_form_decision)
```

Verify the solution. For what value of d_2 does d_2 become uniquely maximal? If you found this easy, also solve Exercises 9, 10, and 11, using Python.

Exercise 9 (*)

Tomorrow, a subject is going for a walk in the lake district. It may rain (E_1), or not (E_2). The subject can either take a waterproof (d_1), or not (d_2). But the subject may also choose to buy today's newspaper, at cost c, to learn about tomorrow's weather forecast (d_S), or not (d_{S^c}), before leaving for the lake district. The forecast has two possible outcomes: predicting rain (S_1), or not (S_2). *Solve for maximality, with $c = 1$.*



$\mathcal{P} =$

	p_1	p_2	p_3	p_4
$S_1 E_1$	0.378	0.378	0.378	0.478
$S_1 E_2$	0.162	0.162	0.262	0.162
$S_2 E_1$	0.072	0.172	0.072	0.072
$S_2 E_2$	0.388	0.288	0.288	0.288

Exercise 10 (**)

Consider again the lake district exercise.

For which values of c is it no longer maximal to buy the newspaper?

(This is the value of information of the newspaper.)

Exercise 11 (*)

Complete the details of the oil wildcatter example which we discussed during the lecture, by normal form backward induction, and thereby verify the solution.

(If you have Python, Octave, or Matlab, you can also try to verify the solution by normal form.)