From the theory of vulnerability to verified policy advice

Nicola Botta with P. Jansson and C. Ionescu

Part 2, Background: climate science, climate policy under uncertainty, 2024-05-06

Plan

Done:

- The computational structure of *possible*: Monadic dynamical systems
- Background: climate science, climate policy under uncertainty
  - Basic notions
  - Example 1: emission reduction policies
  - Optimality, policies

Today:

- Background: climate science, climate policy under uncertainty
  - Example 2: a generation dilemma (Heitzig et al. 2018)
  - Examples 1 and 2: common traits
  - Towards sequential decision problems
- Sequential decision problems

Plan

Today:

- Background: climate science, climate policy under uncertainty
  - Example 2: a generation dilemma (Heitzig et al. 2018)
  - Examples 1 and 2: common traits
  - Towards sequential decision problems
- Sequential decision problems

Next week:

- Bellman's equation, backward induction
- Verified policy advice in a nutshell

Optimality, policies

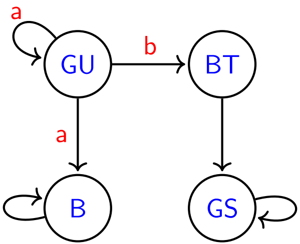
- With the understanding that what can be optimal under uncertainty are policies and with a notion of optimality, we can formulate the questions from the emission reduction example consistently
- How do optimal policies change if we account for the fact that technological innovation could become available later or earlier?
- How do optimal policies change if there is a non-zero probability of exceeding critical thresholds even if we stay within the IPCC emission corridor?
- How do optimal policies change if we account for the fact that climate decisions may not be implemented, for example, because of political instability or because of external shocks?

Example 2: a generation dilemma (Heitzig et al. 2018)

- The world can be in one of four states: GU, GS, B and BT
- B is a bad state, one in which resources are depleted and the wealth of the societies is low
- GS is a good, safe state. In GS, plenty of resources are available, societies are wealthy and there is no risk to turn into B, GU or BT
- GU is a good but unsafe state. In GU, plenty of resources are available, societies are wealthy but there is a significant risk to turn into B
- BT is a bad but temporary state
- In BT, societies are poor but it is certain that the next state will be good and safe: GS

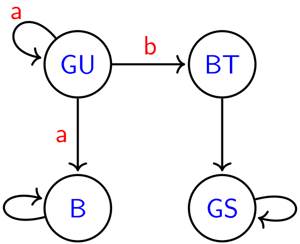
Example 2: a generation dilemma (Heitzig et al. 2018)

- A generation in B, BT or GS has no options: the next states will be B, GS and GS
- A generation in GU has two options: a and b
- If it picks a, the next generation will possibly be in GU again. But it can also end up in B
- If it picks b, the next generation will be in BT with certainty



- What should a generation in GU do? a or b?

Example 2: a generation dilemma (Heitzig et al. 2018)



Exercise 5.5

Should a generation in GU do a or b? The answer is: it depends. Explain on what it might depend.

Exercise 5.6

Put consistent probabilities on the edges of the transition graph above.

Examples 1 and 2: common traits

- Both decision problems have the form of a dilemma
- In both cases, the consequences of decisions are uncertain
- Decisions are taken sequentially, one after the other, see *Incorporating path dependency into decision-analytic methods: an application to global climate-change policy*
- Can we exploit these similarities? Can we develop a method for specifying and solving these and similar decisions problems rigorously? What does this mean?

Towards sequential decision problems

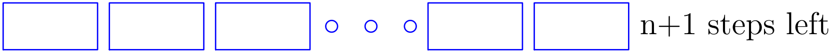
- We tackle these questions in three steps:

  1. Abstract away the details of specific decision problems
  2. Formulate a class of decision problems rigorously
  3. Derive generic, verified solution methods for this class

Towards sequential decision problems: step 1

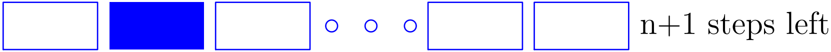
Towards sequential decision problems: step 1

There are $n + 1$ decision steps to go . . .



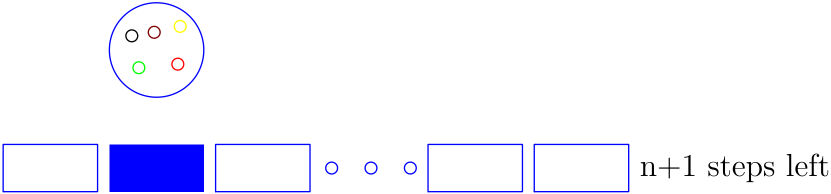
Towards sequential decision problems: step 1

. . . here is the current state,



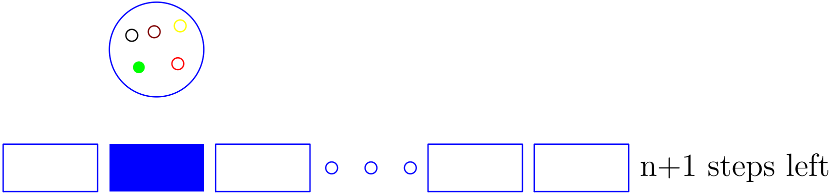
Towards sequential decision problems: step 1

. . . here are your options.



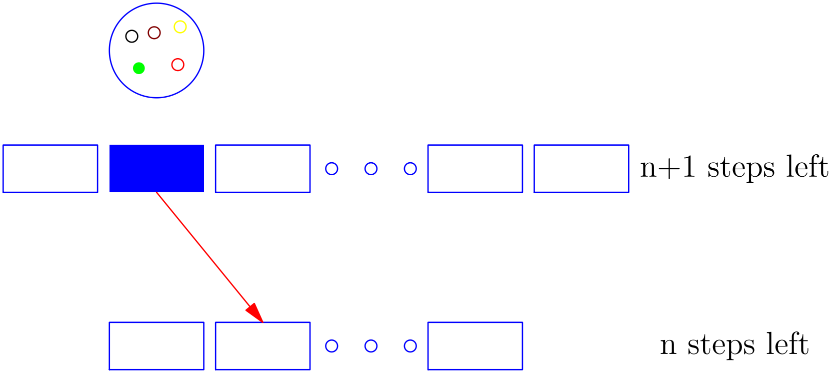
Towards sequential decision problems: step 1

Pick one!



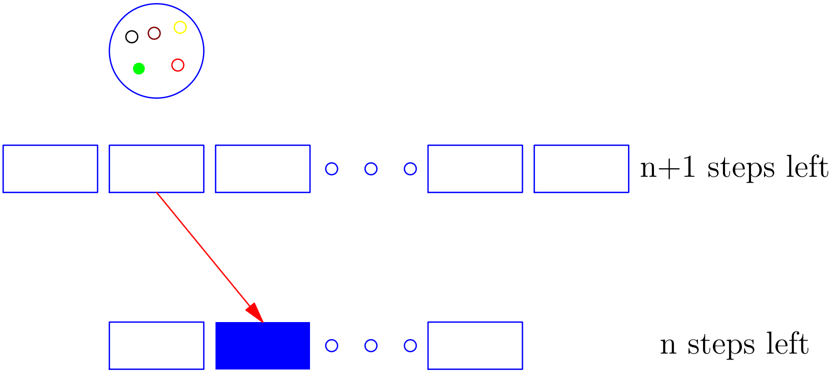
Towards sequential decision problems: step 1

Move to a new state and . . .



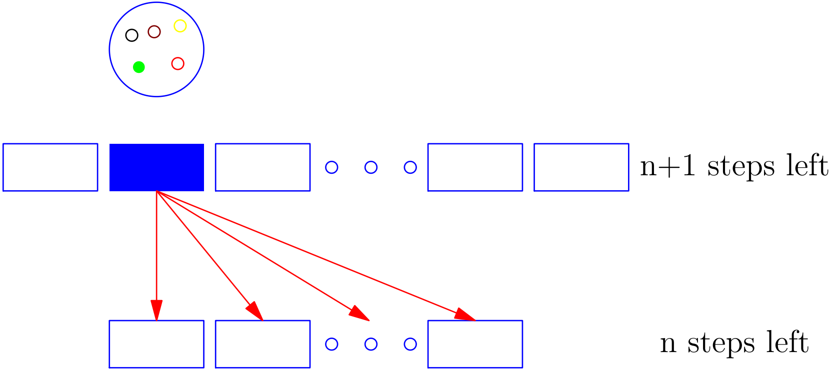
Towards sequential decision problems: step 1

. . . collect rewards and face the next decision step!



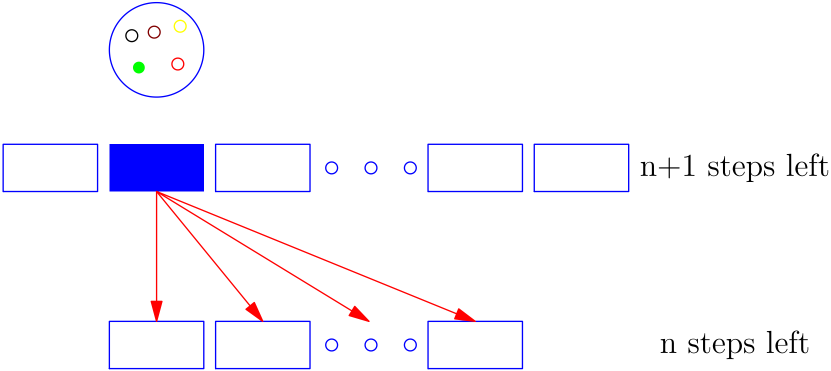
Towards sequential decision problems: step 1

What if there are more than one next possible states?



Towards sequential decision problems: step 1

Apply monadic systems theory!



Next week

- Steps 1-3: Sequential decision problems
- Bellman's equation, backward induction
- Verified policy advice in a nutshell

Exercise 5.7

Try to formalize the cartoon of step 1 (abstract away the details of specific decision problems) in Agda

From the theory of vulnerability to verified policy advice

Nicola Botta with P. Jansson and C. Ionescu

Part 3, 2024-05-06

Plan

Done:

- The computational structure of *possible*: Monadic dynamical systems
- Background: climate science, climate policy under uncertainty

Today:

- Sequential decision problems
- Bellman's equation, backward induction
- Verified policy advice in a nutshell

Sequential decision problems

Sequential decision problems

- As in the vulnerability theory, we build a theory for specifying and solving *finite horizon sequential decision problems* (SDP) in terms of a number of postulates or partial definitions
- These are the problem specification components of the theory/library
- The rest are problem solution components
- The theory is applied by fully defining the specification components

Specification: monadic uncertainty, number of decision steps

- The problem is affected by monadic uncertainty

$$
\begin{array}{llll}
M & : & Set \to Set & \\
fmap_M & : & \{A\ B : Set\} \to (A \to B) \to M\ A \to M\ B & \\
\eta_M & : & \{A : Set\} \to A \to M\ A & \\
\mu_M & : & \{A : Set\} \to M\ (M\ A) \to M\ A &
\end{array}
$$

- We want to make n decision steps

$$n : \mathbb{N}$$

- At decision step $t : \mathbb{N}$, we have already taken t decisions

Specification: states, controls, transition function

- The set of states of the problem can be different at different decision steps

$$X : \mathbb{N} \to Set$$

- The set of controls can be different at different decision steps and in different states

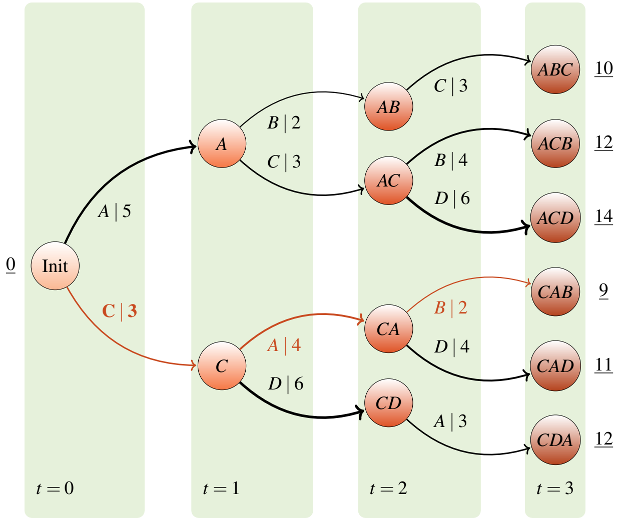
$$Y : (t : \mathbb{N}) \to X\ t \to Set$$

- Selecting a control $y : Y\ t\ x$ in $x : X\ t$ yields an M-structure of possible next states

$$next : (t : \mathbb{N}) \to (x : X\ t) \to Y\ t\ x \to M\ (X\ (suc\ t))$$

- *next* describes an infinite, layered DAG with states in the nodes

Specification: states, controls, transition function



Specification: values, reward function

- Each decision step yields a reward in a value set Val

$$Val \ : \ Set$$

- As in vulnerability theory, we require Val to be a preorder, here a total one

$$\begin{aligned}
\leqslant \quad &: Val \to Val \to Set \\
refl_{\leqslant} \quad &: (x : Val) \to x \leqslant x \\
trans_{\leqslant} \quad &: (x\ y\ z : Val) \to x \leqslant y \to y \leqslant z \to x \leqslant z \\
total_{\leqslant} \quad &: (x\ y : Val) \to Either\ (x \leqslant y)\ (y \leqslant x)
\end{aligned}$$

- We will also need Val to have a reference "zero" element and an "addition"

$$\begin{aligned}
0_{Val} \quad &: Val \\
\oplus \quad &: Val \to Val \to Val
\end{aligned}$$

Specification: reward function, solving a SDP

- Different combinations of current state, control and next state can lead to different rewards

$$reward : (t : \mathbb{N}) \to (x : X\ t) \to Y\ t\ x \to X\ (suc\ t) \to Val$$

- Solving a SDP means finding a sequence of policies that maximizes a measure of the $\oplus$-sum of the rewards along all possible trajectories

Exercise 6.1

Make sure that you fully understand what solving a SDP means.

Exercise 6.2

Define M, X, Y and *next* for the generation dilemma. What could be *Val* and *reward* for this problem?

Solution: policies, policy sequences

- ...finding a sequence of policies that maximizes a measure of the $\oplus$-sum of the rewards along all possible trajectories
- We need to formulate the problem precisely
- We start with policies and policy sequences

$$\begin{aligned}
&\mathit{Policy} &&: (t : \mathbb{N}) \rightarrow \mathit{Set} \\
&\mathit{Policy}\ t &&= (x : X\ t) \rightarrow Y\ t\ x
\end{aligned}$$

$$\begin{aligned}
&\textbf{data}\ \mathit{PolicySeq} : (t\ n : \mathbb{N}) \rightarrow \mathit{Set}\ \textbf{where} \\
&\quad \mathit{Nil} \quad : \{t : \mathbb{N}\} \quad \rightarrow \mathit{PolicySeq}\ t\ \mathit{zero} \\
&\quad _{::}_ : \{t\ n : \mathbb{N}\} \rightarrow \mathit{Policy}\ t \rightarrow \mathit{PolicySeq}\ (\mathit{suc}\ t)\ n \rightarrow \mathit{PolicySeq}\ t\ (\mathit{suc}\ n) \\
&\textbf{infixr}\ 5\ _{::}_
\end{aligned}$$

Exercise 6.3

Explain the $(\mathit{suc}\ t)\ n$ - $t\ (\mathit{suc}\ n)$ pattern in the definition of *PolicySeq*.

Solution: state-control sequences

- …a pol. seq. that max. a meas. of the $\oplus$-sum of the rewards along all possible trajectories
- We want the possible trajectories of a policy sequence to be sequences of state-control pairs

$$
\begin{array}{l}
\textbf{data}\ \mathit{XYSeq} : (t\ n : \mathbb{N}) \to \mathit{Set}\ \textbf{where} \\
\quad \mathit{Last} : \{t : \mathbb{N}\} \quad\ \to X\ t \to \mathit{XYSeq}\ t\ (\mathit{suc}\ \mathit{zero}) \\
\quad \_{\|}\_ \ \ : \{t\ n : \mathbb{N}\} \to \Sigma\,(X\ t)\,(Y\ t) \to \mathit{XYSeq}\,(\mathit{suc}\ t)\,(\mathit{suc}\ n) \to \mathit{XYSeq}\ t\ (\mathit{suc}\ (\mathit{suc}\ n))
\end{array}
$$

Exercise 6.4

A value of type $\mathit{XYSeq}\ t\ n$ is like a vector. What is its length? Can n be zero? Why is the first constructor of XYSeq called Last?

Solution: possible trajectories

- ... a pol. seq. that max. a meas. of the $\oplus$-sum of the rewards along all possible trajectories
- We compute the possible trajectories of a policy sequence as we did for monadic systems

$$
\begin{aligned}
&trj : \{t\ n : \mathbb{N}\} \to PolicySeq\ t\ n \to X\ t \to M\ (XYSeq\ t\ (suc\ n)) \\
&trj\ \{t\}\ Nil \qquad\quad x = \eta_M\ (Last\ x) \\
&trj\ \{t\}\ (p :: ps)\ x = \mathbf{let}\ y \quad\ = p\ x\ \mathbf{in} \\
&\qquad\qquad\qquad\qquad\quad \mathbf{let}\ mx' = next\ t\ x\ y\ \mathbf{in} \\
&\qquad\qquad\qquad\qquad\quad fmap_M\ ((x\ ,\ y) \,{\shortparallel\!\_}\,)\ (mx' \gg\!=_M\ trj\ ps)
\end{aligned}
$$

Exercise 6.5

Make sure that you understand the computation of possible trajectories. What are the types of y, mx' in the **let-in** clauses?

Solution: possible trajectories

- . . . a pol. seq. that max. a meas. of the $\oplus$-sum of the rewards along all possible trajectories
- Now we can compute the $\oplus$-sum of the rewards along all possible trajectories . . .

$$
\begin{aligned}
&sumR : \{t\ n : \mathbb{N}\} \to XYSeq\ t\ n \to Val \\
&sumR\ \{t\}\ (Last\ x) = 0_{Val} \\
&sumR\ \{t\}\ ((x\ ,\ y)\ ||\ xys) = reward\ t\ x\ y\ (head\ xys) \oplus sumR\ xys
\end{aligned}
$$

- . . . and the value of taking n decisions according to a policy sequence in an initial state

$$
\begin{aligned}
&val : \{t\ n : \mathbb{N}\} \to (ps : PolicySeq\ t\ n) \to (x : X\ t) \to Val \\
&val\ ps = measure \circ fmap_M\ sumR \circ trj\ ps
\end{aligned}
$$

Exercise 6.6

Notice that *val ps* is a vulnerability measure! What are *possible* and *harm* here?