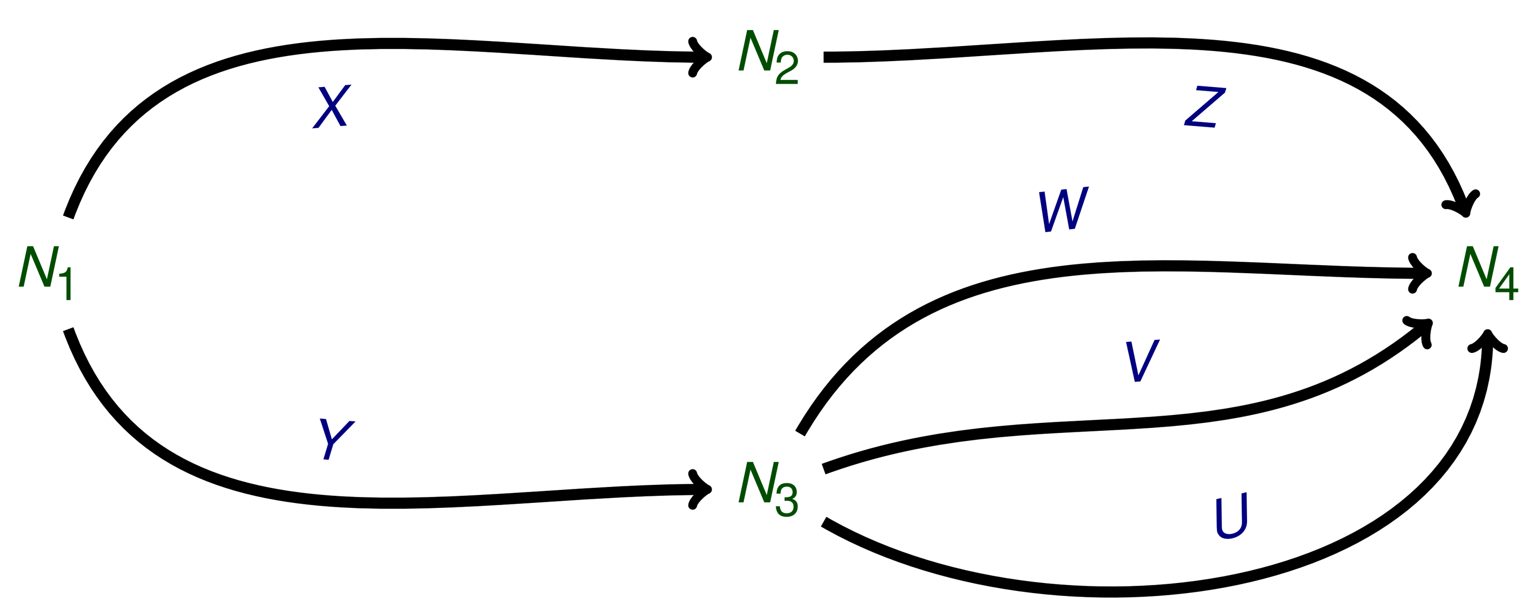
# Dynamic Programming and Subtree Perfectness for Deterministic Discrete-Time Systems with Uncertain Rewards

Nathan Huntley Matthias C. M. Troffaes

Durham University, Department of Mathematical Sciences

27 July, 2011




## Background

### Normal Form Solutions

- aim: find optimal paths from $N_1$ to $N_4$
- set of paths judged optimal is a normal form solution

### Rewards

- binary operator $+$: for combining rewards.
- left identity element: $0 + r = r$
- left inverse: $(-r) + r = 0$
- no additional structure assumed (no utility, no full ranking)

### Gambles and Choice Functions

- each path corresponds to a gamble (an uncertain reward)

$$N_1 \to N_2 \to N_4 = X + Z$$

- find optimal paths by finding optimal gambles
- choice function opt maps sets of gambles to (optimal) subsets

### Standard Normal Form Solution

- apply choice function to the set of all the problem's gambles
- choose one of the paths judged optimal by this process

inefficient for large problems!
but we can do more cleverly...

### References

[1] G. De Cooman and M.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.

[2] N. Huntley and M. C. M. Troffaes.
Characterizing factuality in normal form sequential decision making.
In Thomas Augustin, Frank P. A. Coolen, Serafin Moral, and Matthias C. M. Troffaes, editors, *ISIPTA'09: Proceedings of the Sixth International Symposium on Imprecise Probability: Theories and Applications*, pages 239–248, 2009.

[3] R. Selten.
Reexamination of the perfectness concept for equilibrium points in extensive games.
*International Journal of Game Theory*, 4(1):25–55, Mar 1975.

[4] M. C. M. Troffaes, N. Huntley, and R. Shirota Filho.
Sequential decision processes under act-state independence with arbitrary choice functions.
In E. Huellermeier, R. Kruse, and F. Hoffmann, editors, *Information Processing and Management of Uncertainty in Knowledge-Based Systems*, pages 98–107. Springer, 2010.

## Backward Induction

### Backward Induction Method

- apply choice function recursively to gambles from last stage
- first, apply opt on $\{W, V, U\}$, say $\mathrm{opt}(\{W, V, U\}) = \{V, U\}$
- next, only need opt on $\{X + Z, Y + W, Y + V, Y + U\}$

### Problem

- will not always coincide with the standard normal form solution
- e.g. Γ-maximin and interval dominance do not [1]
- partial answer in [1]:
  sufficient condition for backward induction
  under assumption of utility
- our contribution:
  necessary and sufficient conditions for backward induction
  necessary and sufficient conditions for subtree perfectness
  no utility assumed

### Necessary and Sufficient Conditions for Backward Induction

- Insensitivity to Omission of Non-Optimal Elements

$$\mathrm{opt}(\mathcal{X}) \subseteq \mathcal{Y} \subseteq \mathcal{X} \Rightarrow \mathrm{opt}(\mathcal{Y}) = \mathrm{opt}(\mathcal{X}).$$

- Preservation of Non-Optimality Under Addition of Elements

$$\mathcal{Y} \subseteq \mathcal{X} \Rightarrow \mathrm{opt}(\mathcal{Y}) \supseteq \mathrm{opt}(\mathcal{X}) \cap \mathcal{Y}.$$

- Backward Addition Property

$$\mathrm{opt}(X + \mathcal{Y}) \subseteq X + \mathrm{opt}(\mathcal{Y}).$$

## Subtree Perfectness

### Example of Failure of Subtree Perfectness

- say $\mathrm{opt}(\{W, V, U\} = \{V, U\}$, so $W$ is deleted
- but global solution is $\{X + Z, Y + U\}$, so also $V$ is deleted
- removal of $V$ not caused by the local choice at $N_3$
  but by global choice when $X + Z$ was also considered at $N_1$
- choice at $N_3$ is not completely determined by the options available at $N_3$
- this is a failure of subtree perfectness: choice in subproblem is influenced by problem into which it is embedded [3]

### Necessary and Sufficient Conditions for Subtree Perfectness

- Intersection Property

$$\mathrm{opt}(\mathcal{Y}) = \mathrm{opt}(\mathcal{X}) \cap \mathcal{Y}.$$

- Addition Property

$$\mathrm{opt}(X + \mathcal{Y}) = X + \mathrm{opt}(\mathcal{Y}).$$

### Final Remarks

- Insensitivity to Omission implied by Intersection
- Intersection is equivalent to Total Preorder
- indeterminate choice functions can never be subtree perfect
  similar results well known for other decision problems [2, 4]

Durham University