Danish Informatics Network in the Agricultural Sciences

Computer demonstration: A general sofware system for dynamic programming.

Anders Ringgaard Kristensen

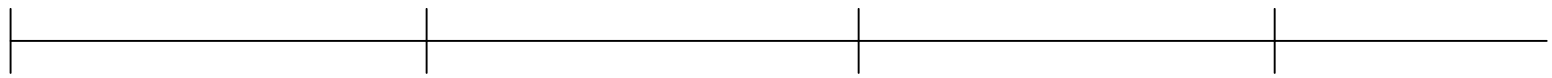
Dept. Anim. Sci. and Anim. Health

Royal Vet. and Agric. Univ.

Outline

- Introduction to dynamic programming and Multi-level hierarchic Markov processes
- The purpose of the software
- Facilities
- Description
- Example: A sow model
- Discussion

A Markov decision process



Stage 1 Stage 2 Stage 3

Stage length e.g. 1 reproduction cycle

At the beginning of each stage, the *state, i,* of the sow is observed:

i=1: Small litter size

i=2: Average litter size

i=3: Big litter size

The state is in this example defined by the value of only one *state variable* (trait)

Actions

- Having observed the state *i*, an *action*, *d*, is taken:
  - *d*=1: Keep the sow
  - *d*=2: Replace the sow at the end of the stage

Rewards

Depending on state i and action d, a reward r^d_i is gained

r^d_i	d=1 (Keep)	d=2 (Replace)
i=1 (Small litter)	10,000	9,000
i=2 (Average)	12,000	11,000
i=3 (Big litter)	14,000	13,000

Output

Depending on state *i* and action *d* a physic ouput m^d_i (in this case number of piglets is involved).

m^d_i	*d*=1 (Keep)	*d*=2 (Replace)
i=1 (Small litter)	8	8
i=2 (Average litter)	11	11
i=1 (Big litter)	14	14

Transition probabilities

Transitions between states are governed by transition probabilities p^d_{ij}

p^d_{ij}	d=1 (Keep)			d=2 (Replace)		
	j=1	j=2	j=3	j=1	j=2	j=3
i=1	0.6	0.3	0.1	1/3	1/3	1/3
i=2	0.2	0.6	0.2	1/3	1/3	1/3
i=3	0.1	0.3	0.6	1/3	1/3	1/3

A *policy s* is a map (rule) assigning to each state an action. An example of a policy for this model is to replace if i=1 and keep if i>1. Thus, in functional notation: $s(1)=2$ ("Replace), and $s(2)=s(3)=1$ ("Keep").

Problem: To determine an *optimal* policy.

What is ml-HMP

- Benefits
  - The curse of dimensionality
  - Decisions on multiple time scales
- A founder process which is an ordinary Markov decision process
- Each combination of state and actions may be extented to a child which is again a Markov decision process
- A child process may be further extended to a ”grand child” level…

Further information

- Kristensen, A.R. & E. Jørgensen. 2000. Multi-level hierarchic Markov processes as a framework for herd management support. *Annals of Operations Research 94*, 69-89.

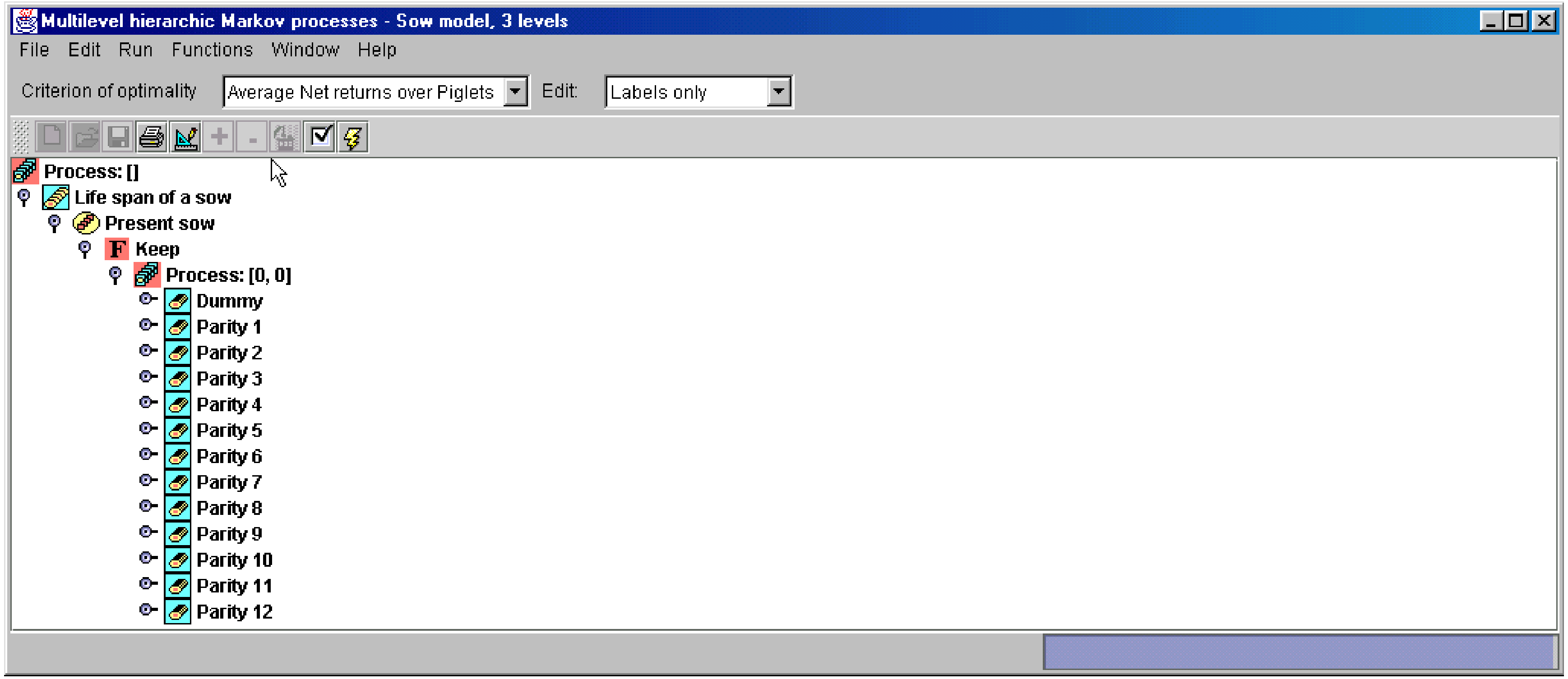
Purpose

- Apprentice level
  - Comprehension
  - Small examples
- Professional user
  - Real world models
  - Only intermediate
- No standard software: A bottle neck for application

Facilities, GUI

- Graphical user interface:
  - Visual editing of model structure
  - Icons for process, stage, state and action
  - Entering of parameters
  - Special icons for various ”tricks”

The graphical interface



Facilities, functions

- Optimization
  - Criteria of optimality
    - Discounting
    - Average rewards over time
    - Average rewards over output
  - Policy iteration
  - Value iteration
- Markov chain simulation

Windows

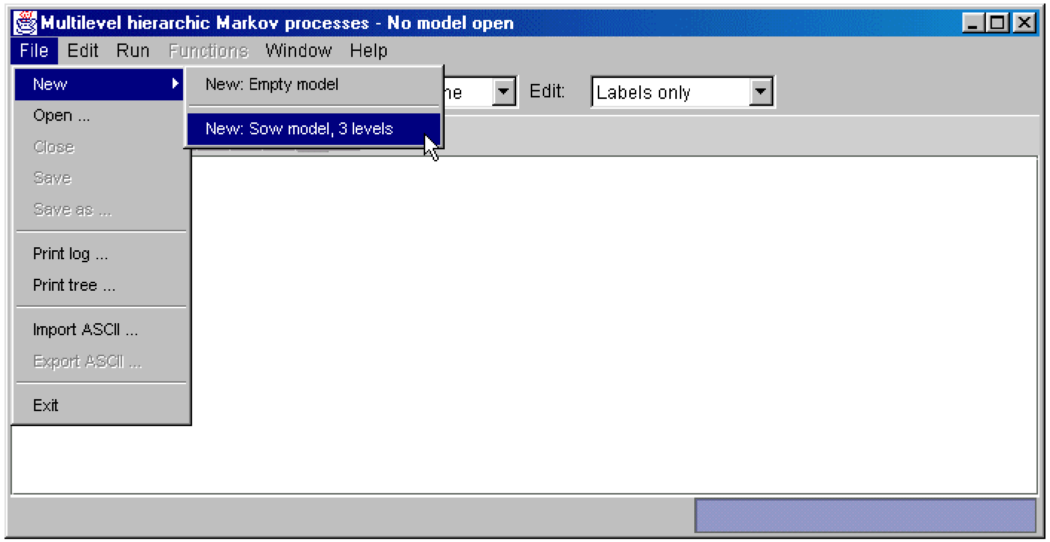
- Process tree
- Optimization log
  - The iterations
  - Time spent on optimization
- Results
  - Optimal policy
  - Present (relative) values of actions
  - Future profitabilities
  - Editing of policies for Markov chain simulation

Technical description

- Model: Array of levels
- Level: Array of processes
- Process: Array of stages
- Stage: Array of states
- State: Array of actions
- Action:
  - Defined by child process
  - Defined by parameters

Plugins

- ModelProvider class (abstract)
  - Generates an entire model
  - Interface for installing and removing
  - Install into the ”New” menu



Example: Sow model

- Founder
  - Stage: Life span of sow
  - State: Dummy
  - Action: Dummy
- Child level 1
  - Stage: A reproductive cycle: mating-mating
  - State: Estimated litter size potential & previous litter size
  - Actions: Boar 1, Boar 2

Example cont.

- Child level 2
  - Stages: Mating, gestation, suckling
  - State: Health, Health & infertile, litter size
  - Action: Mating policy, dummy, Keep-Replace
- Number of states: ~100,000
- Optimization: A few minutes

Discussion

- Visible models
  - Demonstration
  - Model development
- Export of data
- General versus specific software
- Model size