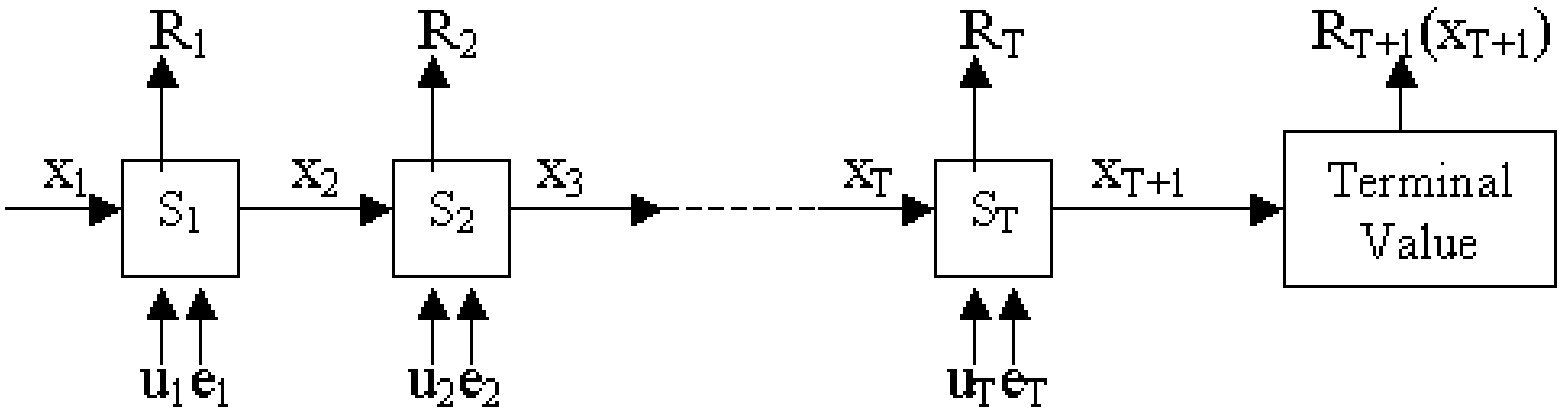
Stochastic Dynamic Programming

G Cornelis van Kooten

Stochastic Dynamic Programming (DP)

Flowchart for Stochastic DP System:



e_i= random effect occurring at stage *i*

Motivation

- Example from Buongiorno & Gilless (*Decision Methods in Forest Management* 2003 Chaps 16 & 17)

State *i*	Volume (m^3/ha)
L (low)	<400
M (medium)	400-700
H (high)	>700

Forest Management Example

Transition probability matrix with NO management

	Next state *j*		
Begin state *i*	L	M	H
L	0.40	0.60	0
M	0	0.30	0.70
H	0.05	0.05	0.90

Assumed time step is 20 years

Assume forest is initially in state L with probability 1. We want to know how it moves over time without management. Where will it end up?

$$p_0 = [1\ 0\ 0]$$

$$p_1 = p_0 P^{NM} = [1\ 0\ 0] \begin{bmatrix} 0.40 & 0.60 & 0 \\ 0 & 0.30 & 0.70 \\ 0.05 & 0.05 & 0.90 \end{bmatrix} = [0.40\ 0.60\ 0]$$

$$p_2 = p_1 P^{NM} = [0.40\ 0.60\ 0] \begin{bmatrix} 0.40 & 0.60 & 0 \\ 0 & 0.30 & 0.70 \\ 0.05 & 0.05 & 0.90 \end{bmatrix} = [0.16\ 0.42\ 0.42]$$

$$p_t = p_{t-1} P^{NM}, \ \text{for}\, t = 1, \ldots, T$$

$$\Rightarrow p^{NM^*} = [\pi_L\ \pi_M\ \pi_H] = [0.07\ 0.12\ 0.82]$$

- In the long run, the stand of trees will have >700 m^3 timber with probability 0.82 and 400-700 m^3 with probability 0.12.
- How long can one expect the stand to remain in one of the three categories? The **mean residence time** is

$$m_i = SL/(1-p_{ii})$$

where SL is the stage length (20 years) and p_{ii} is the diagonal element on P^{NM}, or probability that a stand in state *i* at the beginning of the period is still in that state at the end of the period.

Mean residence time:

$$m_L = 33.3 \text{ yrs}, m_M = 28.6 \text{ yrs}, m_H = 200 \text{ yrs}$$

Mean recurrence time is found as:

$$m_{ii} = SL/\pi_i$$

Recall the values of π_i come from p^*=[.07 .12 .82], and m_{ii} is the time it takes for a stand in state *i* to return to that same state after exiting it.

$$m_{LL} = 285.7 \text{ yrs}$$

$$m_{MM} = 166.7 \text{ yrs}$$

$$m_{HH} = 24.4 \text{ yrs}$$

Transition probability matrix

With Management

Begin state *i*	Next state *j* L	M	H
L	0.40	0.60	0
M	0	0.30	0.70
H	0.40	0.60	0

Time step is 20 years

- Doing the same thing as before, we find:

	With Management		
State *i*	Mean residence time (m_i)	Steady-state probability (π_i)	Mean recurrence time (m_{ii})
L	33.3 yrs	0.22	90.9 yrs
M	28.6 yrs	0.46	43.5 yrs
H	20.0 yrs	0.32	62.5 yrs

Recall: $m_i = SL/(1-p_{ii})$ and $m_{ii} = SL/\pi_i$

Calculating long-run returns

- Suppose we have the following immediate return from harvest under management:

State *i*	Total volume (m^3/ha)	Average volume (m^3/ha)	Harvest and after harvest return to state *i*	
			m^3/ha	$/ha
L	<400	259	0	0
M	400-700	603	344	0
H	>700	817	558	7,254

Recursive Relationship

- Let V_{it} be the value of a stand in state i (= L, M, H) with t periods until the end of the time horizon.
- Let $\beta=1/(1+r)^{20}$
- Present value of expected returns with t+1 periods to go to the end of the time horizon:

$$V_{i,t+1} = R_i + \beta \left(p_{iL}\, V_{Lt} + p_{iM}\, V_{Mt} + p_{iH}\, V_{Ht}\right)$$

Begin with $V_{Lo}= V_{Mo}= V_{Ho}= 0$

Recursive relation: Stage 1

Assume discount rate of 5%

$$V_{L,1} = R_L + \beta \left(p_{LL}\, V_{L0} + p_{LM}\, V_{M0} + p_{LH}\, V_{H0}\right)$$

$$= 0 + 0.377 \{.4\ (0) + .6\ (0) + .0\ (0)\} = 0$$

$$V_{M,1} = R_M + \beta \left(p_{ML}\, V_{L0} + p_{MM}\, V_{M0} + p_{MH}\, V_{H0}\right)$$

$$= 0 + 0.377 \{.0\ (0) + .3\ (0) + .7\ (0)\} = 0$$

$$V_{H,1} = R_H + \beta \left(p_{HL}\, V_{L0} + p_{HM}\, V_{M0} + p_{HH}\, V_{H0}\right)$$

$$= 7254 + 0.377 \{.4\ (0) + .6\ (0) + .0\ (0)\}$$

$$= 7254$$

Recursive relation: Stage 2

$$V_{L,2} = R_L + \beta (p_{LL} V_{L1} + p_{LM} V_{M1} + p_{LH} V_{H1})$$

$$= 0 + 0.377 \{.4 (0) + .6 (0) + .0 (7254)\} = 0$$

$$V_{M,2} = R_M + \beta (p_{ML} V_{L1} + p_{MM} V_{M1} + p_{MH} V_{H1})$$

$$= 0 + 0.377 \{.0 (0) + .3 (0) + .7 (7254)\}$$

$$= 1914$$

$$V_{H,2} = R_H + \beta (p_{HL} V_{L1} + p_{HM} V_{M1} + p_{HH} V_{H1})$$

$$= 7254 + 0.377 \{.4 (0) + .6 (0) + .0 (7254)\}$$

$$= 7254$$

Recursive relation: Stage 3

$$V_{L,3} = R_L + \beta (p_{LL} V_{L2} + p_{LM} V_{M2} + p_{LH} V_{H2})$$

$$= 0 + 0.377 \{.4 (0) + .6 (1914) + .0 (7254)\}$$

$$= 433$$

$$V_{M,3} = R_M + \beta (p_{ML} V_{L2} + p_{MM} V_{M2} + p_{MH} V_{H2})$$

$$= 0 + 0.377 \{.0 (0) + .3 (1914) + .7 (7254)\}$$

$$= 2130$$

$$V_{H,3} = R_H + \beta (p_{HL} V_{L2} + p_{HM} V_{M2} + p_{HH} V_{H2})$$

$$= 7254 + 0.377 \{.4 (0) + .6 (1914) + .0 (7254)\}$$

$$= 7687$$

Recursive relation: Stage *n*

Since $\beta < 1$, convergence eventually occurs (in this case for $t > 10$). The result is that, for each potential starting state, we find the following value:

$$V_{L,n} = \$624/\text{ha}$$

$$V_{M,n} = \$2{,}343/\text{ha}$$

$$V_{H,n} = \$7{,}878/\text{ha}$$

Long-run expected return is found by multiplying the above values by $[\pi_L \ \pi_M \ \pi_H] = [0.22 \ 0.46 \ 0.32]$

Expected return = \$3736/ha

Stochastic DP

So far we have had no decision to make.

Let *p(i,j,k)* be the probability that, if system is in state *i* at time *t*, it will be in state *j* at *t+1* if *u=k*.

Bellman's Equation:

$$V_t(x_t, u_t) = \max_{u_t} E\left[R(x_t, u_t) + \beta V_{t+1}(x_{t+1})\right]$$

$$= \max_k \left[ER(i,k) + \beta \sum_{j=1}^{n} p(i, j, k) V_{t+1}(j) \right]$$

$$V_t(i) = \max_{d(t)} \left[ER^d(i) + \beta \sum_{j=1}^{n} p^d(i, j) V_{t+1}(j) \right]$$

Transition probabilities replace state equation, or equation of motion

$$P^{u_1} = \begin{bmatrix} p_{11}^1 & p_{12}^1 & p_{13}^1 \\ p_{21}^1 & p_{22}^1 & p_{23}^1 \\ p_{31}^1 & p_{32}^1 & p_{33}^1 \end{bmatrix} \qquad P^{u_2} = \begin{bmatrix} p_{11}^2 & p_{12}^2 & p_{13}^2 \\ p_{21}^2 & p_{22}^2 & p_{23}^2 \\ p_{31}^2 & p_{32}^2 & p_{33}^2 \end{bmatrix}$$

One transition matrix for each decision

Sum of each row =1.0

Columns are **single-peaked**

Markov Assumption of DP: All the information about the past is contained in the present value of the state variable.

SDP Definitions

- **Policy iteration:** If any state is **reachable** from any other state, then there is convergence toward an optimal policy that holds for any *t*.
- **Optimal policy:** Optimal decision for any value of state variable at any *t*.
- **Value iteration:** The optimal policy depends not only on the value of the state variable, but also on *t*. Some states are not **reachable** from any other state (viz., soil erosion) – there can be an **absorbing state**

Forestry example: Transition matrices

Begin state *i*	NO CUT: Next state *j* L	NO CUT: Next state *j* M	NO CUT: Next state *j* H	CUT: Next state *j* L	CUT: Next state *j* M	CUT: Next state *j* H
L	0.40	0.60	0.00	0.40	0.60	0.00
M	0.00	0.30	0.70	0.40	0.60	0.00
H	0.05	0.05	0.90	0.40	0.60	0.00

Returns to each decision/state

State	Immediate Return R_{ik} ($/ha)	
i	NO CUT	CUT
L	0	0
M	0	4,472
H	0	7,254

Recursive Relationship

Present value of expected returns with t+1 periods to go to the end of the time horizon:

$V_{i,t+1}$ =

Max { $[R_{iN} + \beta (p_{iLN} V_{Lt} + p_{iMN} V_{Mt} + p_{iHN} V_{Ht})]$,

$$[R_{iC} + \beta (p_{iLC} V_{Lt} + p_{iMC} V_{Mt} + p_{iHC} V_{Ht})]\}$$

where p_{ijk} is the probability that a stand moves from state *i* to state *j* when the decision is *k* (= N,C)

Recursive Relationship (cont)

Proceed as before, but now keep track of the best decision –

cut (C)

no cut (N)

Again, begin with $V_{Lo} = V_{Mo} = V_{Ho} = 0$

$r = 5\%$ so $\beta \approx 0.377$

Recursive relation: Stage 1

$$V_{L,1} = \text{Max}\{ [R_{LN} + \beta (p_{LLN} V_{L0} + p_{LMN} V_{M0} + p_{LHN} V_{H0})], [R_{LC} + \beta (p_{LLC} V_{L0} + p_{LMC} V_{M0} + p_{LHC} V_{H0})]\}$$

$$= \text{Max}\ \{[0 + 0.377(.4\ (0) + .6\ (0) + .0\ (0))], [0 + 0.377(.4\ (0) + .6\ (0) + .0\ (0))]\} = 0\ (\text{N})$$

$$V_{M,1} = \text{Max}\{ [R_{MN} + \beta (p_{MLN} V_{L0} + p_{MMN} V_{M0} + p_{MHN} V_{H0})], [R_{MC} + \beta (p_{MLC} V_{L0} + p_{MMC} V_{M0} + p_{MHC} V_{H0})]\}$$

$$= \text{Max}\ \{[0 + 0.377(.0\ (0) + .3\ (0) + .7\ (0))], [4472 + 0.377(.4\ (0) + .6\ (0) + .0\ (0))]\} = 4472\ (\text{C})$$

Recursive relation: Stage 1 (cont)

$$V_{H,1} = \text{Max}\{\ [R_{HN} + \beta\ (p_{HLN}\ V_{L0} + p_{HMN}\ V_{M0} + p_{HHN}\ V_{H0})],$$
$$[R_{HC} + \beta\ (p_{HLC}\ V_{L0} + p_{HMC}\ V_{M0} + p_{HHC}\ V_{H0})]\}$$

$$= \text{Max}\ \{[0 + 0.377(.05\ (0) + .05\ (0) + .9\ (0))],$$
$$[7254 + 0.377(.4\ (0) + .6\ (0) + 0\ (0))]\}$$

$$= 7254\ (C)$$

Decision: [No cut, cut, cut] = [N C C]

Recursive relation: Stage 2

$$V_{L,2} = \text{Max}\{ [R_{LN} + \beta (p_{LLN} V_{L1} + p_{LMN} V_{M1} + p_{LHN} V_{H1})], [R_{LC} + \beta (p_{LLC} V_{L1} + p_{LMC} V_{M1} + p_{LHC} V_{H1})]\}$$

$$= \text{Max}\{[0 + 0.377(.4\ (0) + .6\ (4472) + .0\ (7254))], [0 + 0.377(.4\ (0) + .6\ (4472) + .0\ (7254))]\} = 1011\ (\text{NC})$$

$$V_{M,2} = \text{Max}\{ [R_{MN} + \beta (p_{MLN} V_{L1} + p_{MMN} V_{M1} + p_{MHN} V_{H1})], [R_{MC} + \beta (p_{MLC} V_{L1} + p_{MMC} V_{M1} + p_{MHC} V_{H1})]\}$$

$$= \text{Max}\{[0 + 0.377(.0\ (0) + .3\ (4472) + .7\ (7254))], [4472 + 0.377(.4\ (0) + .6\ (4472) + .0\ (7254))]\} = 5483\ (\text{C})$$

Recursive relation: Stage 2 (cont)

$$V_{H,2} = \text{Max}\{\ [R_{HN} + \beta\ (p_{HLN}\ V_{L1} + p_{HMN}\ V_{M1} + p_{HHN}\ V_{H1})],$$
$$[R_{HC} + \beta\ (p_{HLC}\ V_{L1} + p_{HMC}\ V_{M1} + p_{HHC}\ V_{H1})]\}$$

$$= \text{Max}\ \{[0 + 0.377(.05\ (0) + .05\ (4472) + .9\ (7254))],$$
$$[7254 + 0.377(.4\ (0) + .6\ (4472) + 0\ (7254))]\}$$

$$= 8265\ (C)$$

Decision: [N C C]

Long-run solution

- After 10 iterations, the algorithm converges on an equilibrium solution given below:

State *i*	Decision	Net present value ($/ha)	Long-run probability (π_i)
L	No cut	1,623	0.40
M	Cut	6,095	0.60
H	Cut	8,877	0.00

Calculation of long-run probabilities is illustrated below

How do we find the long-run probability vector and expected returns?

- Create a new transition matrix by taking, for each decision, the row associated with that decision.
- Example: Suppose the transition matrices are ‘reachable’ and the optimal policy is u_1 if in state 2 and u_2 when in state 1 or 3. Then:

$$P = \begin{bmatrix} p_{11}^2 & p_{12}^2 & p_{13}^2 \\ p_{21}^1 & p_{22}^1 & p_{23}^1 \\ p_{31}^2 & p_{32}^2 & p_{33}^2 \end{bmatrix} = \begin{bmatrix} p^{u_2}[1] \\ p^{u_1}[2] \\ p^{u_2}[3] \end{bmatrix}$$

Let $\pi = [\pi_1\ \pi_2\ \pi_3]$ be the probabilities of being in states 1, 2 and 3 in the long run. We can solve π as was done earlier, or by solving $\pi = \pi P^n$, which is the same as finding:

$$\pi = \lim_{n\to\infty} P^n = \begin{bmatrix} \pi \\ \pi \\ \pi \end{bmatrix} = \begin{bmatrix} [\pi_1\ \pi_2\ \pi_3] \\ [\pi_1\ \pi_2\ \pi_3] \\ [\pi_1\ \pi_2\ \pi_3] \end{bmatrix}$$

Note: Each row is the same

Problem: Since probabilities have the property that

$$0 \leq \text{prob} \leq 1,$$

as $n \to \infty$, P collapses to a null matrix. Another approach is needed

We can find π as follows:

Let $$I = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix} \qquad D = \begin{bmatrix} 0 & 0 & 1 \\ 0 & 0 & 1 \\ 0 & 0 & 1 \end{bmatrix}$$

Then $\Pi = \begin{bmatrix} \pi_1 & \pi_2 & \pi_3 \\ \pi_1 & \pi_2 & \pi_3 \\ \pi_1 & \pi_2 & \pi_3 \end{bmatrix} = \begin{bmatrix} \pi \\ \pi \\ \pi \end{bmatrix} = D\,(I + D - P)^{-1}$

and:

Expected returns = ER = $[\pi_1\ \pi_2\ \pi_3] \times \begin{bmatrix} R_1 \\ R_2 \\ R_3 \end{bmatrix}$

where R_i refers to the returns to state *i* under the optimal policy regime.

- In the previous cut/no harvest timber management example, the decision rule is **no cut** whenever in state L, and **cut** in states M and H
- Taking the ‘L row’ from the ‘no cut’ matrix and the ‘M’ and ‘H’ rows from the ‘cut’ matrix gives:

$$P = \begin{bmatrix} 0.4 & 0.6 & 0 \\ 0.4 & 0.6 & 0 \\ 0.4 & 0.6 & 0 \end{bmatrix}$$

$$I+D-P = \begin{bmatrix} 0.6 & -0.6 & 1.0 \\ -0.4 & 0.4 & 1.0 \\ -0.4 & -0.6 & 2.0 \end{bmatrix}$$

$$Inv(I+D-P) = \begin{bmatrix} 1.4 & 0.6 & -1.0 \\ 0.4 & 1.6 & -1.0 \\ 0.4 & 0.6 & 0 \end{bmatrix}$$

$$D \times Inv(I+D-P) = \begin{bmatrix} 0.4 & 0.6 & 0 \\ 0.4 & 0.6 & 0 \\ 0.4 & 0.6 & 0 \end{bmatrix}$$

Conclusion

- From the SDP algorithm (previous table on slide 56), the long-run expected returns (ER) for each state:

ER if initially in L:	$1,623/ha
ER if initially in M:	$6,095/ha
ER if initially in H:	$8,877/ha

- The long-run expected return is found by multiplying the π vector by the returns vector, which gives $4,306.20/ha.
- Note that you never let trees grow to reach state H since they are harvested in state M.

```r
# Copyright: G Cornelis van Kooten

S <- 5    # Number of states
Con <- 2    # Number of controls
iter <- 15  # Number of iterations or time horizon

price <- 121/2000  # $ per lb based on $78, $106 and $121 per US ton
rate <- 0.05  # Discount rate
beta <- 1/(1+rate) # Discount fator
cost <- c(31.27, 109.86)  # Costs of fallow 31.27 and cropping $/acre
ControlName <- c('Summerfallow', 'Plant')
decision <- array(0, dim=c(iter, S))  # Creat cell array for string variables

salvage <- rep(0, S)  #Salvage value for each state
for (j in 1:S) { val[1,j] <- salvage[j]}

# Probablility transition matrices
fallow <- array(c(0.0225, 0.4400, 0.4055, 0.1130, 0.0190,
            0.0000, 0.0770, 0.4350, 0.3565, 0.1315,
            0.0000, 0.0210, 0.2665, 0.4175, 0.2950,
            0.0000, 0.0070, 0.1560, 0.3845, 0.4525,
            0.0000, 0.0025, 0.0870, 0.3130, 0.5975), dim=c(S,S))

crop <- array(c(0.1750, 0.3975, 0.2475, 0.1085, 0.0715,
          0.1025, 0.3425, 0.2830, 0.1500, 0.1220,
          0.0790, 0.3090, 0.2895, 0.1680, 0.1545,
          0.0640, 0.2845, 0.2905, 0.1795, 0.1815,
          0.0545, 0.2620, 0.2885, 0.1895, 0.2055), dim=c(S,S))
fallow <- t(fallow)
crop <- t(crop)
mid <- c(0.75,2.00,3.00,4.00,5.55)

vfal <- rep(0,S)
vcrop <- rep(0,S)
val <- matrix(0, iter, S)

for (yrs in 2:iter)
{           # Loop years #
   for (m in 1:S)   #For each current state
      {
        rwdcrp <- 0
        rwdfal <- 0
        for (n in 1:S) # For each future state
          {
            rwdcrp <- beta*crop[m,n ]*val[yrs-1, n] + rwdcrp
            rwdfal <- beta*fallow[m,n]*val[yrs-1, n] + rwdfal
          }  # Close for each future state
        vfal[m] <- - cost[1] + rwdfal
        vcrop[m] <- price*(76.4+3137*(1-0.636^mid[m])) - cost[2] + rwdcrp
        val[yrs,m] <- max(rbind(vfal[m], vcrop[m]))
        control <- which.max(rbind(vfal[m], vcrop[m]))
        decision[yrs,m] <- ControlName[control]

      }     # Close for each current state #
}       # Close loop years #
val
decision
```

```
val
decision

# Calculate long run returns

#install.package('stringr')
library('stringr')
item1 <- sum(str_count(decision[iter,], 'Summerfallow'))
#item2 <- sum(str_count(decision[iter,], 'Plant'))
item2 <- item1+1
LRtrans <- rbind(fallow[1:item1,], crop[item2:S,])
P <- LRtrans
D1<- matrix(0,S,S-1)
D2 <- rep(1,S)
dim(D2) <- c(S,1)
D <- cbind(D1,D2)
I <- diag(S) # make identity matrix

cropreturn <- rep(0,S)
for (j in 1:S) {cropreturn[j] <- price*(76.4+3137*(1-0.636^mid[j])) - cost[2]   }

returns <- c( rep(-cost[1], item1), cropreturn[item2:S])  # Periodic returns for the optimal decision at each state

BigPi <- D%*%solve(I+D-P)
LittlePi <- BigPi[1,]
ExpectedReturn <- LittlePi%*%returns
VarReturn <- sum(LittlePi*((returns-ExpectedReturn)^2))
SDreturn <- sqrt(VarReturn)

ExpectedReturn
SDreturn

```