Dam Viable Management under Uncertainty

Michel De Lara

October 10, 2017

Contents

1 Problem data

We consider a dam manager intenting to maximize the intertemporal payoff obtained by selling power produced by water releases, when the water inflows (rain, outflows from upper dams) are random. However, the manager must also respect a minimal volume during the Summer months for tourism reasons.

1.1 Dam dynamics

We model the dynamics of the water volume in a dam by

$$\underbrace{S(t+1)}_{\text{future volume}} = \min\{S^\sharp, \underbrace{S(t)}_{\text{volume}} - \underbrace{q(t)}_{\text{turbined}} + \underbrace{a(t)}_{\text{inflow}}\} , \quad t \in \mathbb{T} := \{t_0, \dots, T-1\}$$

with

- time $t \in \overline{\mathbb{T}} := \{t_0, \dots, T\}$ is discrete (such as days), and t denotes the beginning of the period $[t, t+1[$,
- $S(t)$ volume (stock) of water at the beginning of period $[t, t+1[$, belonging to the discrete set $\mathbb{X} = \{0, 1, 2, \dots, S^\sharp\}$, made of water volumes, where $S^\sharp$ is the maximum dam volume,
- $a(t)$ inflow water volume (rain, etc.) during $[t, t+1[$, belonging to $\mathbb{W} = \{0, 1, 2, \dots, a^\sharp\}$
- decision-hazard: $a(t)$ is not available at the beginning of period $[t, t+1[$
- $q(t)$ turbined outflow volume during $[t, t+1[$, decided at the beginning of period $[t, t+1[$, supposed to depend on $S(t)$ but not on $a(t)$, belonging to the discrete set $\mathbb{U} = \{0, 1, 2, \dots, q^\sharp\}$, where $q^\sharp$ is the maximum which can be turbined by time unit (and produce electricity),
- $s(t) = \big[S(t) - q(t) + a(t) - S^\sharp\big]_+$ the spilled volume

The dam manager is supposed to make a decision, here turbining $q(t)$ at time t, before knowing the water input $a(t)$. Such a case is called *decision-hazard*. The constraint on the water turbine $q(t)$ is

$$0 \le q(t) \le S(t) . \tag{1}$$

A *scenario* is a sequence of uncertainties:

$$a(\cdot) := \big(a(t_0), \dots, a(T-1)\big) . \tag{2}$$

1.2 Criterion: intertemporal payoff

The manager original problem is one of *payoff maximization* where turbining one unit of water has unitary *price* $p(t)$. On the period from t_0 to T, the payoffs sum up to

$$\sum_{t=t_0}^{T-1} p(t)q(t) + \texttt{UtilFin}\big(T, S(T)\big) , \tag{3}$$

where

- the sequence
$$p(\cdot) = \big(p(t_0), \dots, p(T-1)\big) \tag{4}$$
of prices is supposed to be known in advance (in other models, it could be progressively revealed to the manager),

- the final term $\texttt{UtilFin}\big(T, S(T)\big)$ gives value to the water volume in the dam at the horizon T.

1.3 Constraint: minimal volume during the Summer months

For "tourism" reasons, the following constraint is imposed

$$\text{stock} \qquad S(t) \geq S^\flat \ , \quad \forall t \in \{\text{July}, \text{August}\} \ .$$

In what follows, we shall be more specific about the sense with which this constraint has to be satisfied, namely in probability.

1.4 Water turbined strategy

A strategy $\texttt{Rule} : \mathbb{T} \times \mathbb{X} \to \mathbb{U}$ assigns a water turbined $q = \texttt{Rule}(t, S)$ to any state S of dam stock volume and to any decision period $t \in \mathbb{T}$. Once given, we obtain uncertain volume trajectories $S(\cdot) := \big(S(t_0), \dots, S(T-1), S(T)\big)$ and turbined trajectories $q(\cdot) := \big(q(t_0), \dots, q(T-1)\big)$ produced by the "closed-loop" dynamics

$$\begin{array}{rcl} S(t_0) & = & S_0 \\ S(t+1) & = & \min\{S^\sharp, S(t) - q(t) + a(t)\} \\ q(t) & = & \texttt{Rule}\big(t, S(t)\big) \end{array} \tag{5}$$

and function of the scenario $a(\cdot)$. Thus, in the end, we obtain an uncertain payoff

$$\mathsf{Crit}^{\texttt{Rule}}\big(t_0, S_0, a(\cdot)\big) := \sum_{t=t_0}^{T-1} p(t) q(t) + \texttt{UtilFin}\big(T, S(T)\big) \ , \tag{6}$$

where $S(\cdot)$ and $q(\cdot)$ are given by (5).

1.5 Probabilistic model on water inputs and expected criterion

We suppose that sequences of uncertainties $\big(a(t_0), \dots, a(T-1)\big)$ are random variables with a known probability distribution $\mathbb{P}$ on the set $\{0, \dots, a^\sharp\}^{T-t_0}$.

We suppose that the random variables $\big(a(t_0), \dots, a(T-1)\big)$ are independent with distribution $\pi_0(t), \dots, \pi_{a^\sharp}(t)$ on the set $\{0, \dots, a^\sharp\}$:

$$\mathbb{P}\{a(t) = 0\} = \pi_0(t), \dots, \mathbb{P}\{a(t) = a^\sharp\} = \pi_{a^\sharp}(t) \ . \tag{7}$$

Notice that the random variables $\big(a(t_0), \dots, a(T-1)\big)$ are independent, but that they are not necessarily identically distributed. This allows us to account for seasonal effects (more rain in autumn and winter).

To each strategy `Rule`, we associate the *expected payoff*

$$\mathbb{E}\Big[\mathsf{Crit}^{\texttt{Rule}}\big(t_0, S_0, a(\cdot)\big)\Big] = \mathbb{E}\big[\sum_{t=t_0}^{T-1} p(t)q(t) + \texttt{UtilFin}\big(T, S(T)\big)\big] \; , \tag{8}$$

where the expectation $\mathbb{E}$ is taken with respect to the probability $\mathbb{P}$.

2 Maximizing the expected payoff and computing the resulting probability to satisfy the constraint

2.1 Dynamic programming equation

The dynamic programming equation associated with the problem of *maximizing the expected payoff* (8)

$$\max \mathbb{E}\big[\sum_{t=t_0}^{T-1} p(t)q(t) + \texttt{UtilFin}\big(T, S(T)\big)\big] \tag{9}$$

is

$$\begin{array}{rcl} V(T,S) & = & \overbrace{\texttt{UtilFin}\big(T, S(T)\big)}^{\text{final payoff}} \; , \\ V(t,S) & = & \displaystyle\max_{q \in \{0,1,2,\dots,\min\{S, q^\sharp\}\}} \mathbb{E}_{a(t)}\big[\underbrace{p(t)q}_{\text{instant. payoff}} + V\big(t+1, \underbrace{\min\{S^\sharp, S-q+a(t)\}}_{\text{future stock volume}}\big)\big] \; , \end{array} \tag{10}$$

where the expectation $\mathbb{E}$ is taken with respect to the probability in (7).

2.2 Data for the numerical simulations

We know will make numerical simulations, and try different strategies. We shall consider a daily management over one year

$$t_0 = 1 \quad \text{and} \quad T = 365 \; , \tag{11}$$

with

$$S_0 = 0 \text{ hm}^3 \; , \quad S^\sharp = 100 \text{ hm}^3 \; , \quad q^\sharp = \frac{0.4}{7} \times S^\sharp \quad \text{and} \quad a^\sharp = \frac{0.5}{7} \times S^\sharp \tag{12}$$

where we say that, during one week, one can turbine at maximum 40% of the dam volume, and that during one week of full water inflows, an empty dam can be half-filled.

The sequence of prices is known in advance. We shall produce it by one sample from the expression

$$p(t) = (1+\epsilon(t))\overline{p} \qquad \text{with} \qquad \overline{p} = 66 \text{ MWh/hm}^3 \times 2.7 \text{ euros/MWh}^3 \tag{13}$$

where $\epsilon(t)$ is drawn from a sequence of i.i.d. uniform random variables in $[-1/2, 1/2]$.

The probability of water inflows (from zero to the maximum $a^\sharp$) is known in advance.[1]

Copy the following Scilab code into a file `DamData.sce`.

```
// exec DamData.sce

// -----------------------------------------------------
// DATA
// -----------------------------------------------------


// State
volume_max=100;
volume_min=0;

// Control
control_max=0.4/7*volume_max;
control_max=control_max+1;

// Time
tt0=1;
horizon=365;
TT=tt0:(horizon-1);
bTT=tt0:(horizon);

// Prices
price=66*2.7;
price=price*(1/2+0.5*(rand(TT)-1/2));

// Uncertainties
uncertainty_max=floor(0.5/7*volume_max);
uncertainty=[0:uncertainty_max];

// Probabilities
```

[1] We shall produce these probabilities by one sample drawn from a sequence of i.i.d. uniform random variables. This does not mean that these probabilities are random. This is just a trick to have probabilities differ from student to student.

```
unnormalized_proba=cumsum(ones(uncertainty))-1;
proba1=unnormalized_proba/sum(unnormalized_proba);
// more rain in winter
proba182=proba1($:-1:1);
// less rain in summer

// simulation of independent sequences of water inflows between 1 and
// uncertainty_max+1

Simulations=50;

WW=zeros(Simulations,horizon-tt0+1);

for ss=1:Simulations do
  for tt=bTT do
    proba=(1-sin(%pi*tt/365))*proba1+sin(%pi*tt/365)*proba182;
    WW(ss,tt)=dsearch(rand(),cumsum(proba));
  end
end

Scenarios=WW;

xset("window",1);xbasc();
plot2d2(bTT,Scenarios')
```

Copy the following Scilab macros into the same file `DamData.sce`.

In the macro `trajectories`, the output `CC` is the mean payoff averaged over the scenarios: by the law of large numbers, `CC` is an approximation of the expected payoff if the number of scenarios is large enough (Monte Carlo method).

```
// -----------------------------------------------------
//  MACROS
// -----------------------------------------------------

// Dynamics
function ssdot=dynamics(ss,qq,aa)
  ssdot=max(volume_min,min(volume_max,ss-qq+aa));
endfunction

// Instantaneous payoff function
function c=instant_payoff(tt,ss,qq,aa)
  c=price(tt)*qq;
```

```
endfunction


// Final payoff function
function c=final_payoff(tt,ss)
  c=0;
endfunction


// Trajectories simulations

function [SS,QQ,CC]=trajectories(SS0,scenarios,policy)
  SS=[];
  QQ=[];
  CC=[];
  nb_simulations=size(scenarios,'r');
  for simu=1:nb_simulations do
    ss=SS0;
    qq=[];
    cc=0;
    aa=scenarios(simu,:);
    for tt=TT do
      qq=[qq,policy(tt,ss($))];
      ss=[ss,dynamics(ss($),qq($),aa(tt))];
      cc=cc+instant_payoff(tt,ss($),qq($),aa(tt));
    end
    cc=cc+final_payoff(TT($),ss($));
    SS=[SS;ss];
    QQ=[QQ;qq];
    CC=[CC;cc];
  end
  //
  disp('The payoff is '+string(mean(CC)));
endfunction
```

2.3 Scilab code for the additive stochastic dynamic programming equation

Copy the following Scilab code into the file `Damoptimality.sce`.

```
///////////////////////////////////////////////////////////////////////
//    STOCHASTIC ADDITIVE DYNAMIC PROGRAMMING EQUATION
///////////////////////////////////////////////////////////////////////
```

```
// -----------------------------------------------------
// DATA
// -----------------------------------------------------

states=[0:volume_max];
controls=[0:control_max];

cardinal_states=prod(size(states));
cardinal_controls=prod(size(controls));
cardinal_uncertainty=prod(size(uncertainty));

state_min=min(states);
state_max=max(states);

// -----------------------------------------------------
//  MACROS
// -----------------------------------------------------

function [FEEDBACK,VALUE]=SDP(FINAL_PAYOFF)
  VALUE=zeros(bTT'*states);
  FEEDBACK=zeros(TT'*states);

  VALUE(horizon,:)=FINAL_PAYOFF;// vector

  // backward time
  for tt=TT($:-1:1) do
    loc=zeros(cardinal_controls,cardinal_states);
    // local variable containg the values of the function to be minimized
    for jj=1:cardinal_controls do
      hh=controls(jj);
      loc(jj,:)=0;
      // the following loop computes an expectation
      for dd=1:cardinal_uncertainty do
        ww=uncertainty(dd);
        loc(jj,:)=loc(jj,:)+ ...
                  proba(dd)*(-1/%eps*bool2s(states < hh)+bool2s(states >= hh)) .* ...
                  (instant_payoff(tt,states,hh,ww)+ ...
                   VALUE(tt+1,dynamics(states,hh,ww)-state_min+1));
      end;
    end
    //
```

```
    [mmn,jjn]=min(loc,'r');
    [mmx,jjx]=max(loc,'r');
    // mm is the extremum achieved
    // jj is the index of the extremum argument
    //
    VALUE(tt,:)=mmx;
    // maximal payoff
    FEEDBACK(tt,:)=controls(jjx);
    // optimal feedback
  end
endfunction
```

We first consider that the final "value of water" is zero:

$$\mathtt{UtilFin}(T, S_0) = 0 \; . \tag{14}$$

```
// We start with a zero value of water at the end of the year
zero_final_payoff_vector=zeros(states);

// ------------------------------------------------------
//  SIMULATIONS
// ------------------------------------------------------

[FEEDBACK,VALUE]=SDP(zero_final_payoff_vector);

// optimal strategy
function uu=optimal_rule(tt,xx)
  uu=FEEDBACK(tt,xx-state_min+1);
endfunction


// Trajectories simulations and visualization
SS0=0;
[SS,HH,CC]=trajectories(SS0,Scenarios,optimal_rule);
xset("window",10);// xbasc();
plot2d(bTT,SS')
xtitle('Stock volumes in a dam following an optimal strategywith a zero final value of w
       '(time)','(volume)')

// Payoff histogram
xset("window",20);xbasc();
histplot(100,CC)
xtitle('Histogram of the optimal payoff with a zero final value of water')
```

```
disp('The minimum of the optimal payoff is '+string(min(CC)));
disp('The mean of the optimal payoff is '+string(mean(CC)));
disp('The maximum of the optimal payoff is '+string(max(CC)));
```

Question 1 *Picture the trajectories of the stocks corresponding to the optimal strategy. Evaluate the optimal expected payoff, and compare it with the value function $V(t_0, 0)$ evaluated at the initial time t_0 and the initial stock $S_0 = 0$. Explain why these two quantities should be close. What do you observe for the final stocks? Explain why.*

2.4 Optimal strategy when the final "value of water" is not zero

Till now, there was no gain in leaving water in the dam at the ultimate decision period. From now on, we consider that the "value of water" `UtilFin`(T, S_0) is given by

$$\texttt{UtilFin}(T, S_0) = \frac{1}{1 + r_f} V(t_0, S_0) \tag{15}$$

where $\frac{1}{1+r_f}$ is a discount factor. We shall take $r_f = 0.1$ when $T = 365$ days.

```
//////////////////////////////////////////////////////////////////////
//    VALUE OF WATER
//////////////////////////////////////////////////////////////////////

final_payoff_vector=(1/(1+0.1))*VALUE(1,:);
```

Copy the following Scilab code into the file `DamOptimality.sce`.

```
// ---------------------------------------------------
//  SIMULATIONS
// ---------------------------------------------------

[FEEDBACK,VALUE]=SDP(final_payoff_vector);

// optimal strategy
function uu=optimal_rule(tt,xx)
  uu=FEEDBACK(tt,xx-state_min+1);
endfunction


// Trajectories simulations and visualization
[SS,HH,CC]=trajectories(SS0,Scenarios,optimal_rule);
xset("window",10);// xbasc();
```

```
plot2d(bTT,SS')
xtitle('Stock volumes in a dam following an optimal strategy with a final value of water
       '(time)','(volume)')

// Payoff histogram
xset("window",20);xbasc();
histplot(100,CC)
xtitle('Histogram of the optimal payoff with a final value of water')

disp('The minimum of the optimal payoff is '+string(min(CC)));
disp('The mean of the optimal payoff is '+string(mean(CC)));
disp('The maximum of the optimal payoff is '+string(max(CC)));
```

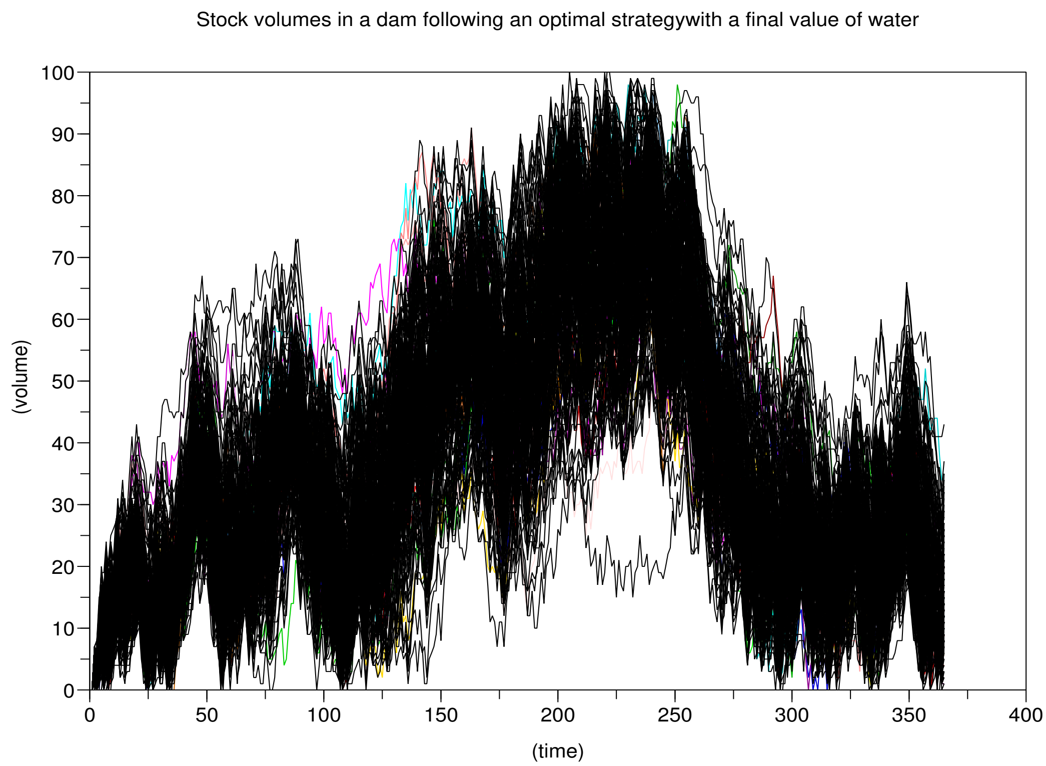

Figure 1: Stock volumes in a dam following an optimal strategy, with a final value of water

Question 2 *Picture the trajectories of the stocks corresponding to the optimal strategy. Evaluate the optimal expected payoffs for different values of the initial stock* S_0*, and compare them with the value function* $V(t_0, S_0)$ *evaluated at the initial time* t_0 *and the initial stock*

S_0. Display the histogram of the optimal payoff. Compare the mean of the optimal payoff with the upper and lower bounds of the distribution.

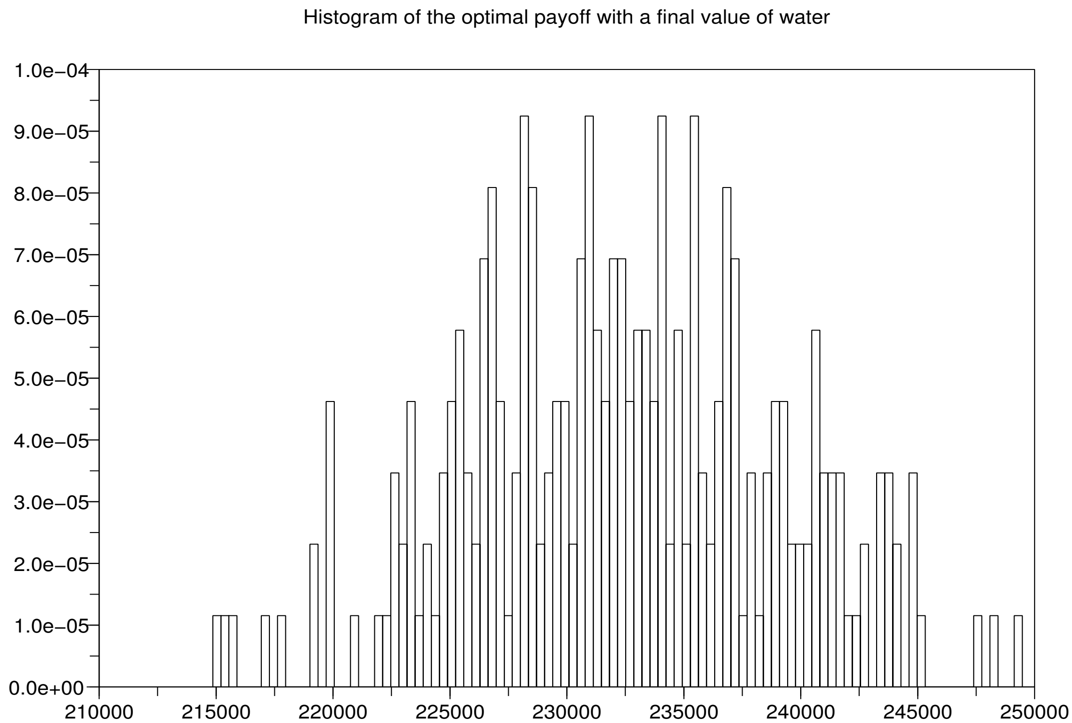

Figure 2: Histogram of the optimal payoff, with a final value of water

2.5 Evaluation of the probability to satisfy the tourism constraint

Let $S^\star(\cdot)$ denote an optimal stock trajectory.

Question 3 *Evaluate the probability*

$$\mathbb{P}\left\{S^\star(t) \geq S^\flat \ , \quad \forall t \in \{\text{July}, \text{August}\}\right\} \tag{16}$$

that the water volume $S^\star(t)$ remains above F% of $S^\sharp$ during the months of July and August, where F% varies between 0% and 100%.

```
// -----------------------------------------------------
// PROBABILITY CONSTRAINT EVALUATION
```

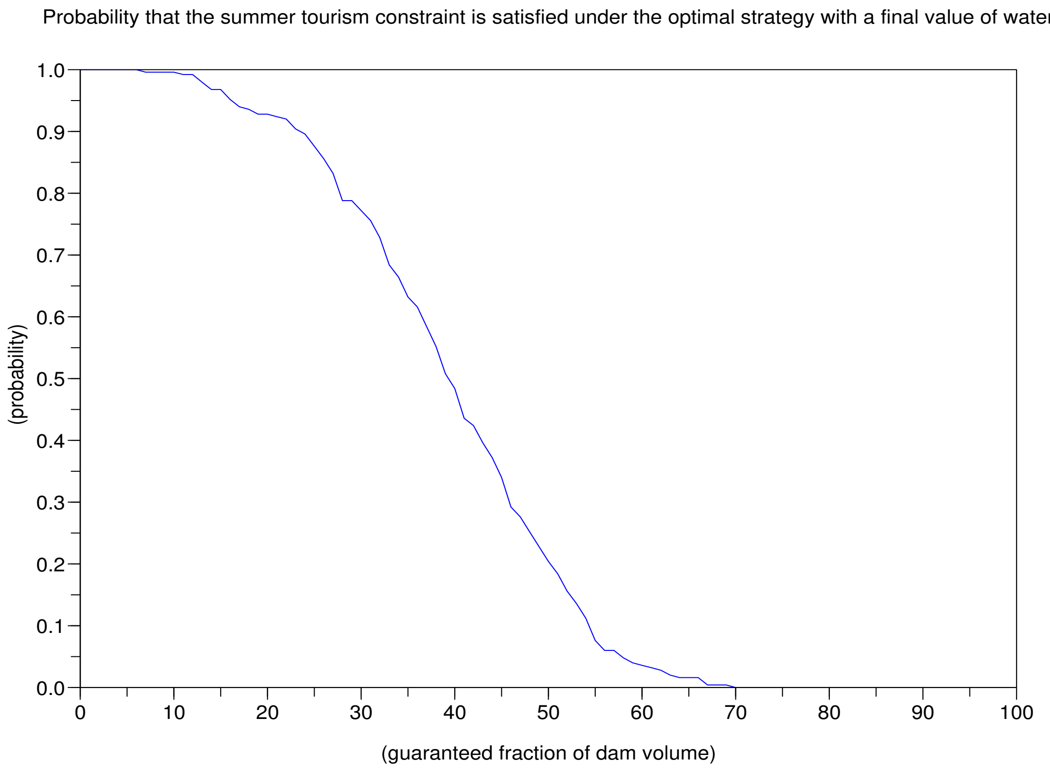

Figure 3: Probability that the summer tourism constraint is satisfied under the optimal strategy, with a final value of water

```
// -----------------------------------------------------
Summer=([1:horizon] >= horizon/2 & [1:horizon] <= horizon/2+2*30*horizon/364);
VP=[]
for jj=0:100 do
  VP=[VP,mean(bool2s(min(SS(:,Summer),'c') >= jj/100*volume_max))];
end

xset("window",30);xbasc();
plot(0:100,VP)
xtitle('Probability that the summer tourism constraint is satisfied under the optimal st
       '(guaranteed fraction of dam volume)','(probability)')
```

3 Maximizing the viability probability to guarantee jointly payoff and summer water volume

The payoff at time $t \in \mathbb{T}$ is

$$B(t) = \sum_{s=t_0}^{t} \overbrace{p(s)q(s)}^{\text{water turbined profit}}$$

and

$$B(T) = \sum_{t=t_0}^{T-1} \overbrace{p(t)q(t)}^{\text{water turbined profit}} + \overbrace{\texttt{UtilFin}\big(T, S(T)\big)}^{\text{final stock value}}$$

We propose an alternative stochastic viability formulation to (9)-(16) under the form

$$\max \mathbb{P} \left\{ a(\cdot) \;\middle|\; \begin{array}{l} \text{water input scenarios along which} \\ \text{the stocks} S(t) \geq S^{\flat} \ , \quad \forall t \in \{\text{July}, \text{August}\} \\ \text{and} \\ \text{the final profit} B(T) \geq B^{\flat} \end{array} \right\} \tag{17}$$

3.1 Multiplicative dynamic programming equation

The dynamic programming equation associated with the problem of *maximizing the viability probability* (17) is

$$\begin{aligned}
V(T, S, B) &= \overbrace{\mathbf{1}_{\{B \geq B^{\flat}\}}}^{\text{final constraint}} \ , \\
V(T-1, S, B) &= \max_{q \in \{0,1,2,\ldots,\min\{S, q^{\sharp}\}\}} \mathbb{E}_{a(T-1)} \big[\mathbf{1}_{\{S \geq S^{\flat} \ , T-1 \in \{\text{July}, \text{August}\}}} \\
&\quad \times V\big(t+1, \min\{S^{\sharp}, S - q + a(T-1)\}, \\
&\quad B + p(T-1)q + \texttt{UtilFin}\big(T, \min\{S^{\sharp}, S - q + a(T-1)\}\big)\big) \big] \ , \\
V(t, S, B) &= \max_{q \in \{0,1,2,\ldots,\min\{S, q^{\sharp}\}\}} \mathbb{E}_{a(t)} \big[\overbrace{\mathbf{1}_{\{S(t) \geq S^{\flat} \ , t \in \{\text{July}, \text{August}\}\}}}^{\text{instantaneous constraint}} \\
&\quad \times V\big(t+1, \underbrace{\min\{S^{\sharp}, S(t) - q(t) + a(t)\}}_{\text{future stock volume}}, \underbrace{B + p(t)q}_{\text{future payoff}}\big) \big] \ , \quad \forall t = t_0, \ldots, T-2 \ ,
\end{aligned} \tag{18}$$

where the expectation $\mathbb{E}$ is taken with respect to the probability (7). Notice that the equation for $t = T-1$ takes into account the term $\texttt{UtilFin}\big(S(T)\big)$ in the payoff.

3.2 Scilab code for the multiplicative stochastic dynamic programming equation

Copy the following Scilab code into the file `DamViability.sce`.

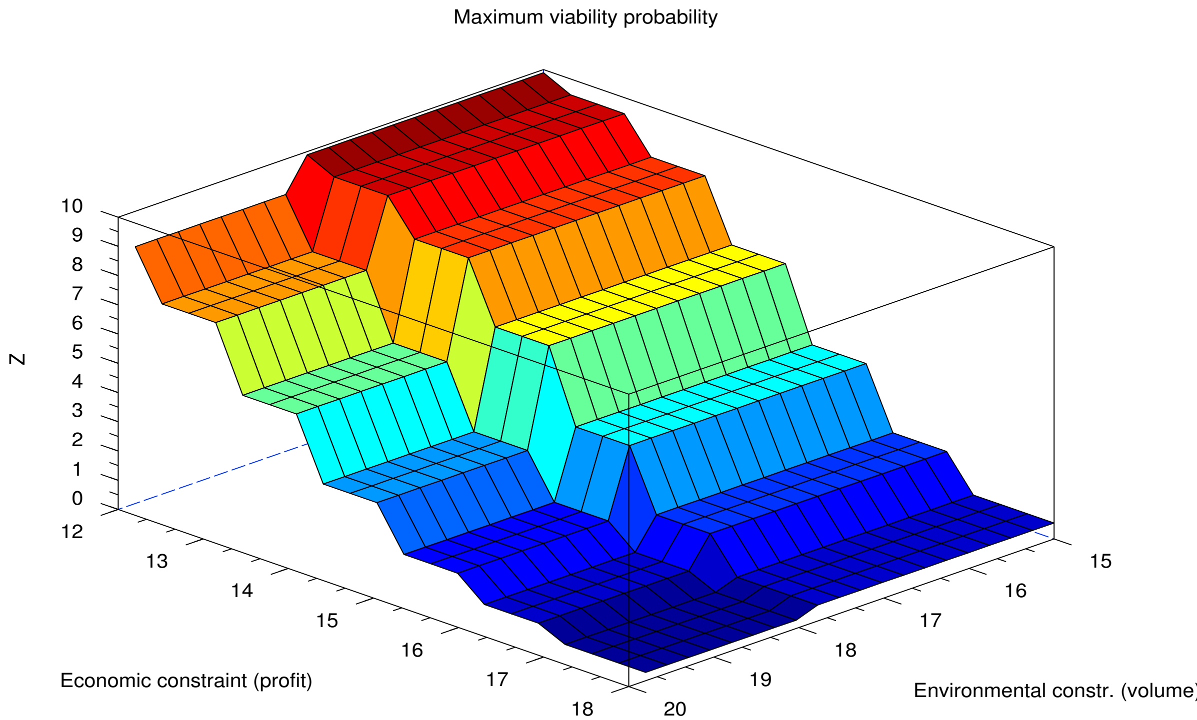

Figure 4: Maximal viability probability as a function of guaranteed thresholds $S^\flat$ and $B^\flat$

```
// exec DamViability.sce

// -----------------------------------------------------
// DATA
// -----------------------------------------------------

horizon=10;

SSmax=volume_max;
nb_SS=SSmax+1;
state_PP=0:(horizon*SSmax);
nb_PP=length(state_PP);
PPmax=max(state_PP);
controls=0:nb_SS;
nb_CC=length(controls);

// For a practical work lower the number of points
```

```
volume_max=10;
SSmax=volume_max;
nb_SS=SSmax+1;
state_PP=0:(horizon*SSmax);
nb_PP=length(state_PP);
PPmax=max(state_PP);
// controls=0:nb_SS;
controls=linspace(0,SSmax+1,10);
nb_CC=length(controls);


uncertainties=0:1;
nb_WW=length(uncertainties);
proba=ones(1,nb_WW)/nb_WW;

/////////////////////////////////////////////////////////////////////////
//    MULTIPLICATIVE STOCHASTIC DYNAMIC PROGRAMMING EQUATION
/////////////////////////////////////////////////////////////////////////


// -----------------------------------------------------
//  MACROS
// -----------------------------------------------------

function [FEEDBACK,VALUE]=MSDP(SS_min,PP_min)
  //  SS_min= SSmin * Summer ;
  //  PP_min=PPmin;

  VALUE=list();
  FEEDBACK=list();
  VALUE(horizon)=ones(nb_SS,1)*bool2s(state_PP >= PP_min);
  shift=[(horizon-1):(-1):1];
  for tt=shift do
    VVdot=VALUE(tt+1);
    VV=zeros(VVdot);
    for ss=1:nb_SS do
      SS=ss-1;
      if SS >= SS_min(tt) then
        for pp=1:nb_PP do
          PP=pp-1;
          locext=[];
```

```
      for cc=1:ss do
        // control constraint
        UU=cc-1;
        locint=0;
        for oo=1:nb_WW do
          ww=uncertainties(oo);
          SSdot=(min(SSmax,SS-UU+ww));// physical value
          ssdot=SSdot+1;// corresponding index
          Ppdot=(min(PPmax-1, ...
                     PP+price(tt)*UU+bool2s(tt==horizon-1)*final_payoff(tt,ssdot)));
          //physical value
          ppdot=min(round(ppdot/PPmax)+1,nb_PP);//corresponding index
          locint=locint+proba(oo)*VVdot(ssdot,ppdot);
        end;// of the expectation loop
        locext=[locext,locint];
      end;// of the control loop
      VV(ss,pp)=max(locext);
    end;// of the pp loop
  end;// of the if condition on SS
 end;// of the ss loop
 VALUE(tt)=VV;
end;// of the time tt loop
endfunction
```

3.3 Maximal viability probability function and viability kernels

Question 4 *Compute the maximal viability probability. Deduce the viability kernels with confidence levels 100%, 95% and 90%.*

```
stacksize('max');

PPmin=0.13*PPmax;
SSmin=0.89*SSmax;

[FEEDBACK,VALUE]=MSDP(SSmin*Summer,PPmin)
```

3.4 Maximal viability probability as a function of guaranteed thresholds

Copy the following Scilab code into the file `DamViability.sce`.

```
//precision=10;
precision=2;
```

```
Thresholds_EE=linspace(0.1,0.15,precision)*PPmax;
nb_EE=length(Thresholds_EE);
//
Thresholds_BB=linspace(0.75,0.99,precision)*SSmax;
nb_BB=length(Thresholds_BB);

ViabProba=zeros(nb_BB,nb_EE);

for bb=1:nb_BB do
  SS_min=Thresholds_BB(bb)*Summer;
  for ee=1:nb_EE do
    PP_min=Thresholds_EE(ee);
    [FEEDBACK,VALUE]=MSDP(SS_min,PP_min);
    VV=VALUE(1);
    ViabProba(bb,ee)=VV(nb_SS-2,1);
    // initial state
  end
  // of the ee loop
end
// of the bb loop

save("ViabStoch.dat",ViabProba)
```

Question 5 *Launch the above code (maybe you will have to reduce the time step, or the horizon, and adapt the code in consequence if the computation takes too much time). Visualize the maximal viability probability starting from an almost full dam. Draw iso-probability curves. Comment on what you observe.*

```
SP=10^6;
SP=10;
// scale probability
SC=10;
SC=2;

xset('window',1);// xclear(); // xbasc();
xset('colormap',jetcolormap(20));
//    plot3d1(Thresholds_BB,SC*Thresholds_EE,SP*ViabProba);
//    plot3d1((1:nb_BB)/nb_BB,(1:nb_EE)/nb_EE,SP*ViabProba);
plot3d1(SC*Thresholds_BB,Thresholds_EE,SP*ViabProba);
xtitle("Maximum viability probability","Environmental constr. (volume)", ...
       "Economic constraint (profit)")
```

```
// ------------------------------------------------------------------

xset('window',10);contour(Thresholds_BB,Thresholds_EE,ViabProba,[0.7:0.05:1]);
xtitle("Maximum viability probability","Environmental constraint (volume)", ...
       "Economic constraint (profit)")
```

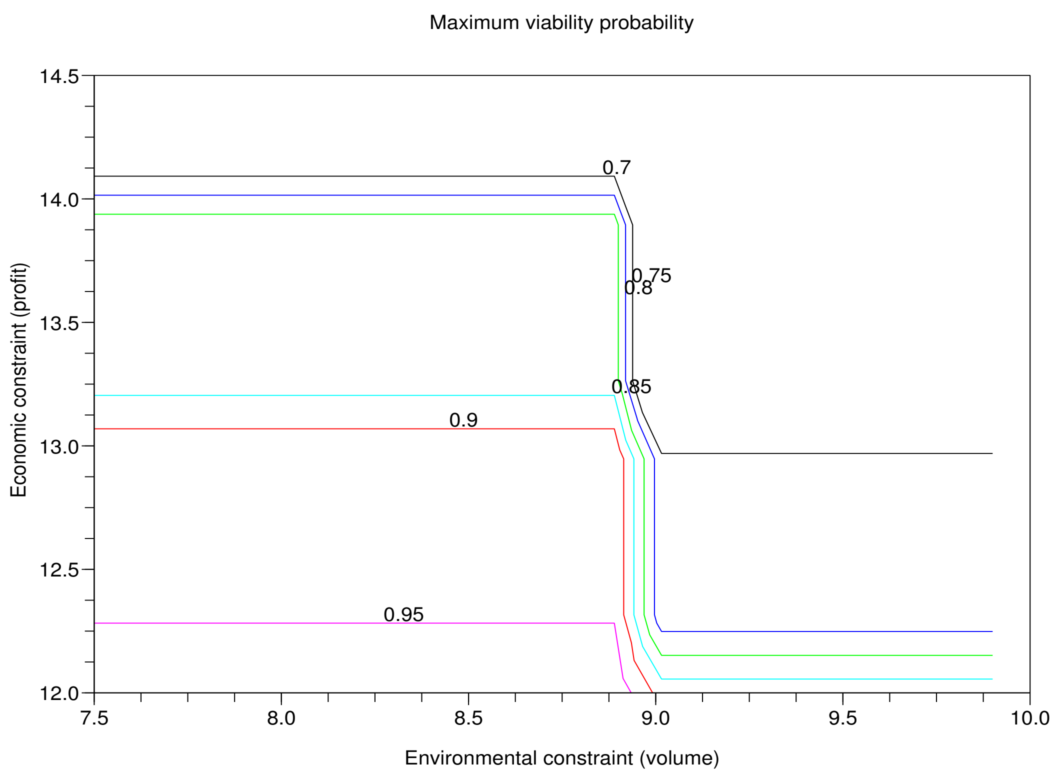

Figure 5: Iso-values for the maximal viability probability as a function of guaranteed thresholds $S^{\flat}$ and $B^{\flat}$

References

D. P. Bertsekas. *Dynamic Programming and Optimal Control.* Athena Scientific, Belmont, Massachusets, second edition, 2000. Volumes 1 and 2.

M. De Lara and L. Doyen. *Sustainable Management of Natural Resources. Mathematical Models and Methods.* Springer-Verlag, Berlin, 2008.