Dynamic Programming
Complexity and Application

M. Akian J.P. Chancelier
J.P. Quadrat
INRIA Domaine de Voluceau, 78153 Le Chesnay, B.P.105 (France).

August 1988

Contents

1 INTRODUCTION

Many people in the automatic control community think that the dynamic programming equation (DPE) is not useful in practise because it leads to algorithms which have a too large complexity ; thus they are interested only by the LQG approach. Other ones work only on the mathematical aspect of some DPE. Our thesis is :

1. it is very difficult to avoid the dynamic programming approach on many concrete problems. We shall discuss our experience on a specific exemple ;

2. the "curse of dimensionality" can be avoided in a large amount by using the dynamic programming approach more for its time-space decoupling property than for computing the effective feedback — the Bellman function can be seen as a set of decoupling prices which can be obtained by an aggregated model ;

3. a very good numerical approach for solving DPE is the Multigrid point of view — algorithms based on this idea have a complexity linear with the number of discretization points ;

4. a main stone of expert systems able to solve optimization problem in presence of uncertainty will be the dynamic programming — we shall discuss Pandore an expert system on stochastic control ;

5. some algebraic treatment of dynamic programming can be done using the (max, +) algebra.

We shall develop these points in this paper.

2 MOTIVATION

For solving problems in presence of uncertainties we can have different points of view :

1. the oldest one is the design of a robust feedback without actual preoccupation of optimization — the H^∞ method is its most recent development ;

2. the LQG approach is very restrictive at the modelization level but is a very good tool for designing stabilizing feedback — used around an optimal deterministic trajectory it has a good approximation property in presence of small noises and on regularity assumptions Fleming [8],Bensoussan [2] ;

3. adaptative methods suppose the presence of a linear system that evolves slowly ; stochastic iteration can be used to adapt the linear model in line ; thus in some sense they optimize only a local (in time) system ; nevertheless the idea of stochastic iteration can be used to optimize parameters in a non–linear feedback, but, in this case, it subsists the difficulty of the choice of the class in which we want to optimize ;

4. the dynamic programming approach avoid a lot of these difficulties but has the main drawback of the "curse of the dimensionality" ; it has been more developped by the operational research community which is more concerned with the optimization problems than the stabilization ones.

We think that on a lot of real problems it is very difficult to avoid the dynamic approach and thus it is important to develop good numerical methods for solving it.

Let us take a storage system given to us by Electricity Of France. It is a pumping station system and we want decide if it is useful to build it. To take a rational decision we have to evaluate the gain that we can expect from this investment.

We can consider that this investment is marginal with respect to the existing ones ; thus we use the following simplified model :

$$\left\{ \begin{array}{l} dX_t = (U_1 - U_2)dt \ \ 0 \leq X_t \leq 1 \\ \min_{U_1,U_2} I\!\!E \int_0^T Y_t(\rho U_1 - U_2)dt \end{array} \right. \tag{1}$$

where :

- X_t denotes the level of water in the stock,
- Y_t the prices of the electricity,
- ρ the yield of the pumps,
- U_1 the pumping flow $1 \geq U_1 \geq 0$,
- U_2 the turbining flow $1 \geq U_2 \geq 0$,
- T the horizon of the management.

But Y_t — the price of electricity — is a stochastic process which is, in fact, itself a solution of an optimization problem. We model it here by a diffusion process in one dimension to avoid the dimension difficulty of the dynamic programming approach. Thus the model for the prices will be :

$$dY_t = b(t, Y_t)dt + \sigma(t, Y_t)dW_t \ , \tag{2}$$

in which we have to identify b and σ, a trajectory of Y_t being known as the solution of another optimization problem.

The resolution of a such simple problem can save million of dollars and the classical LQG problem is not very relevant for solving it.

Let us compare the cost obtained using three different strategies — on the actual datas.

1. The first one is the feedback obtained by solving the DPE equation after the identification of the process describing the price Y_t ;

Strategy	Cost
1	-8.7
2	-5.96
3	-6.38

Table 1: Comparaison of the three strategies on the actual datas.

2. The second one is an open loop control obtained by solving the deterministic control problem based on the average prices ;
3. The third one is a feedback defined by two parameters p, q :

$$\begin{array}{llll} \text{if} & Y_t > q, & U_1 = 1, & U_2 = 0, \\ \text{if} & Y_t < p, & U_1 = 0, & U_2 = 1, \\ \text{if} & p < Y_t < q, & U_1 = 0, & U_2 = 0. \end{array}$$

The form of this strategy has been suggered by the form of the DPE solution. The two parameters (p, q) are optimized by a dichotomy method.

These numerical results show an important improvement obtained by the use of the DPE. It is natural to think that, in general, there is a loss of optimality of order 1 by a simplification of the class of the strategies in which we optimize, unless there is somewhere a small parameter.

These remarks show that unless we accept a loss of optimality of order 1 we have to solve the DPE — it is the only known way to optimize in the general class of strategies.

Let us explain now how we can avoid the "curse of dimensionality" in some cases by decoupling in time and space the problem thanks to the resolution of the DPE of an aggregate problem.

3 HOWARD ALGORITHM

Let us consider the DPE associated to the control of diffusion processes with a discounted cost :

$$\min_u \{A_\lambda(u)V + c(u)\} = 0$$

with :

$$A_\lambda(u)V = \sum_{i=1}^n b_i(x,u)\frac{\partial V}{\partial x_i} + \sum_j a_{ij}\frac{\partial V}{\partial x_i \partial x_j} - \lambda V \ ;$$

or its discretized version :

$$\min_u \{A_\lambda^h(u)V^h + c(u)\} = 0 \ ;$$

where h is the step of discretization and $A_\lambda^h(u)$ is obtained by discretization of $A_\lambda(u)$ by a finite difference or finite element method Kushner [13], Goursat-Quadrat [11].

The Howard algorithm is defined by the iteration :

1. $u^n \to V^n$ by resolution of :
$$A_\lambda(u^n)V^n + c(u^n) = 0 \ ,$$
2. $V^n \to u^{n+1}$ by :
$$\min_u A_\lambda(u)V^n + c(u) \ ,$$

can be interpreted as a Newton method for solving :

$$\min_u\{A_\lambda(u)V + c(u)\} \overset{df}{\equiv} \psi(V) = 0 \ .$$

Indeed the iteration of the Newton method can be written :

$$V^{n+1} = V^n - \psi'^{-1}(V^n)\psi(V^n) \ ,$$

but

$$\begin{aligned}\psi'(V^n) &= (\frac{\partial A_\lambda}{\partial u}(u^n) + \frac{\partial c}{\partial u}(u^n))\frac{\partial u^n}{\partial V}(V^n) + A_\lambda(u^n) \\ &= A_\lambda(u^n)\end{aligned}$$

in the case of optimization without constraints.

Now let us suppose that :

- we have to solve a large dimension problem for which the dynamic programming operator is $A^1_\lambda(u)$ and the cost $c^1(u)$;
- we know an aggregated or simplified model $A^2_\lambda(u)$ with the aggregated or simplified cost $c^2(u)$.

Suppose that we use the strategy defined by :

$$u^\star \in \arg\min\{A^1_\lambda(u)V^2 + c^1(u)\}$$

with V^2 solution of the DPE of the aggregated model :

$$\min_u\{A^2_\lambda(u)V^2 + c^2(u)\} = 0 \ .$$

Let us call $\delta = |V^1 - V^2|$. By the speed of convergence of the Newton method we see that $|V^\star - V^1| = \delta^2$ where $V^\star$ denotes the solution of $A^1_\lambda(u^\star)V^\star + c_1(u^\star) = 0$ — that is the cost of the strategy $u^\star$ for the true model.

The Newton method convergence conditions are not necessarily fulfiled, but even in this case, by the global convergence of the Howard algorithm we have $|V^\star - V^1| < |V^1 - V^2|$.

The problem :

$$\min\{A^1_\lambda(u)V^2 + c^1(u)\}$$

which gives the actual control must be solved on line along the realized trajectory and thus the dimensionality difficulty is partially resolved — up to a on-line computation.

Some variant of this idea has been used with success by Turgeon [19] and Lederer-Torrion-Bouttes [14] on a real exemple of management of electricity production.

These considerations show that DPE can be useful for some problems. Thus the remaining problem to be solved is to find an efficient numeric algorithm to compute the solution of the DPE.

4 COMPLEXITY

Let us consider the DPE in the discounted case that is :

$$\min_u\{A(u)V + c(u)\} - \lambda V = 0 \ , \lambda > 0 \ ;$$

with :

- $A(u)$ the generator of the controled stochastic process,
- u the control,
- $c(u)$ the instantaneous cost, ≥ 0,
- λ the actualization rate.

If the stochastic process does not belong to a finite space we suppose that we have discretized it. Thus $A(u)$ is a matrix which satisfies :

$$\begin{array}{lcll}\sum_j A_{ij}(u) & = & 0 & \forall i\, \forall u \\ -1 < A_{ii}(u) & < & 0 & \forall i\, \forall u \\ A_{ij}(u) & \geq & 0 & j \neq i\, \forall u\end{array}$$

the three classical algorithms are — Ross [18], Derman[6] :

- *policy iteration* that we have discussed in the previous paragraph ;
- *value iteration* that is :
$$V^{n+1} = \frac{1}{1+\lambda}\min_u\{M(u)V^n + c(u)\}$$
with
$$M(u) = A(u) + I$$
Some variant have been proposed by Gonzales-Rofman [10] and Falcone [7], they corresponds in the linear case — $\mathcal{U}$ has only one element — respectively to the Gauss Seidel and overrelaxation variant of the Jacobi algorithm.
- *linear programming* "find the largest subsolution"
$$\left\{\begin{array}{l}\max_V(1,V) \\ A(u)V + c(u) - \lambda V \geq 0 \ \forall u \in \mathcal{U}\end{array}\right. \tag{3}$$
This problem is a linear programming one if the set $\mathcal{U}$ is finite.

Let us discuss this three algorithms. We have no experience on the linear programming one but it seems not to be a good point of view when the control set is large and when $\mathcal{U}$ has only one element it consumes more time than a classical algorithm solving a linear system — which is the problem to which is reduced the problem of control in this particular case.

Let us compare now the policy iteration and the value iteration. It is in general admited that the value iteration is faster. Indeed because the policy iteration is a Newton method the number of iterations is small. Let us say 5 — until now on many examples that we have tried no more than 4 iterations have been needed, see Quadrat [17] for some numerical experiments ; but at each iteration we have to solve a linear system ; thus the cost is approximatively 5 linear systems + 5 minimizations for all the points of discretization.

The value iteration when $\mathcal{U}$ is reduced to one element is the classical Jacobi iteration to solve a linear system. It can be improved by the classical way of acceleration of convergence. The cost of this algorithm is the number of iterations for solving the linear system times the cost of one optimization.

Now to improve the policy iteration we can use an iterative method to solve the linear systems. Let us suppose that we don't solve the linear system completely but we stop when we have achieved a given precision. The complete algorithm can be seen as a value iteration where sometimes we skip the minimization step — this is possible because the speed of convergence of the control (Newton method) is superlinear thus faster than the convergence of the cost (ill conditioning difficulties). Thus the best point of view is the policy iteration with iterative method to solve the linear system.

Now we have to decide when we make the minimization step. The super–linear convergence of the newton method suggests that if n is the index of iteration it is enough to make a minimization only when $\log n$ is integer thus the total cost is approximatively the cost of solving one linear system by an iterative method.

Let us now study the conditioning property of the linear system that we have to solve. The difficulty appears often when the discrete problem comes from a multidimensional control problem. In this case the conditioning of the matrices $A(u)$ is approximatively the same as the discretized Laplacian operator (Δ) one.

To study this problem let us look at the conditioning property of the discretized version of the following problem :

$$\begin{cases} \Delta V = f & \text{in }]0,\pi[^2 \\ V = 0 & \text{on the boundary } \Gamma \text{ of }]0,\pi[^2 \end{cases}$$

Suppose that the discretization is done by finite difference on a regular mesh. Let us denote by $N+1$ the number of point of discretization in each direction and let us denote by Δ^h the discretized operator.

The eignevalue of Δ^h are :

$$\lambda_{n,m} = -\frac{4(\sin^2(\frac{nh}{2}) + \sin^2(\frac{mh}{2}))}{h^2}$$

with $n, m = 1 \cdots N$. The eigenvector associated are :

$$\sin(inh)\sin(jmh) \quad i,j = 1,\ldots,N$$

Thus the conditionning of the matrix Δ^h is $\frac{1}{N^2}$ and the best iterative algorithm is of the form :

$$V^{n+1} = V^n + \tau(\Delta^h V^n - f) \tag{4}$$

— which converges — costs using the Marchouk [16] discussion : $kN^4 \log N$ if we want to stop when we have achieved a precision of $h^2 = \frac{1}{N^2}$ — which is the precision of the discretization. Because $M = N^2$ is the number of discretization point, the complexity is of order $kM^2 \log M$ and thus the complexity of DPE by policy iteration will be of this order also.

The speed of convergence is of order $\frac{1}{N^2}$ using block–relaxation technique. It can be improved up to the order $\frac{1}{N}$, which leads to a complexity of order $kM^{3/2} \log M$, but this block–relaxation is specialized to 2 dimensional problems.

Let us remark also that direct methods are more efficient than the iterative ones only for small dimensions (1 or 2) for higher dimension they have a complexity near of M^3.

Let us discuss now multigrid ideas which lead to a linear algorithm — thus of order $M = N^m$ whatever is the dimension m (M is the number of discretization points if N is the number of discretization in each dimension).

5 MULTIGRID ALGORITHM

5.1 Multigrid ideas

This technique is used in the context of the resolution of large linear systems obtained by discretization of partial differential equations. It is wanted to design an iterative algorithm with a contraction factor independent of the mesh size. This result is obtained using aggregation ideas. We build a pyramidal set of more and more aggregated models. The more aggregated ones giving a good approximation of the "slowest part" of the system.

More precisely we have to solve :

$$Ax + u = 0 \tag{5}$$

with $A : V \rightarrow V'$ linear, bounded coercive. We use the diagram :

$$\begin{tikzcd}
V \arrow[r, "A"] & V' \arrow[d, "C_k"] \\
\mathbb{R}^{N_k} \arrow[u, "B_k"] \arrow[r, "A_k"] & \mathbb{R}^{N_k} \arrow[d, "C_{k-1}"] \\
\mathbb{R}^{N_{k-1}} \arrow[u, "B_{k-1}"] \arrow[r, "A_{k-1}"] & \mathbb{R}^{N_{k-1}} \\
\vdots & \vdots \arrow[d, "C_1"] \\
\mathbb{R}^{N_1} \arrow[u, "B_1"] \arrow[r, "A_1"] & \mathbb{R}^{N_1}
\end{tikzcd}$$

with :

$$A_k = C_k A B_k$$

$$A_{k-1} = C_{k-1} C_k A B_k B_{k-1}$$

and :

$$N_1 < N_2 < \ldots < N_k \,.$$

To solve the approximated problem :

$$A_k x_k + u_k = 0 \ \text{ with } C_k u = u_k \tag{6}$$

we use an iterative method which is a numerical integration of the differential equation :

$$\dot{x}_k = A_k x_k + u_k \,. \tag{7}$$

By a correct choice of C_k and B_k, A_k keeps some properties of A — for example all its eigenvalues are negative. These properties insure the stability of (7) and

$$\lim_{t\to\infty} x_k(t) = -A_k^{-1} u_k .$$

But the eigenvalues of A_k are spread, in general, on a very large interval as we have seen on the previous paragraph.

Let us suppose now that :

- $\mathcal{R}(B_{k-1})$ [resp. $\mathcal{R}(B_{k-1}B_{k-2}\ldots)$] is the right invariant space of A_k containing the N_{k-1} [resp. $N_{k-2},\ldots$] slowest modes,
- $\mathcal{R}(C'_{k-1})$ [resp.$\mathcal{R}(C'_{k-1}C'_{k-2}\ldots)$] is the left invariant space of A_k containing the N_{k-1} [resp. $N_{k-2},\ldots$] slowest modes.

Then :

$$P_{k-1} = B_{k-1} C_{k-1}$$

$$[resp. P_{k-2} = B_{k-1} B_{k-2} C_{k-2} C_{k-1}, \ldots]$$

is a projector on $\mathcal{R}(B_{k-1})$ parallel to $\mathcal{N}(C_{k-1})$.

The differential equation :

$$\dot{z}_k = A_k[I + \alpha_{k-1}P_{k-1} + \cdots + \alpha_1 P_1] z_k + u_k \tag{8}$$

with $\alpha_i \geq 0$. has the same properties of stability than (7). Thus :

$$\lim_{t\to\infty} z_k(t) = z_\infty$$

which is such that :

$$(I + \alpha_{k-1}P_{k-1} + \cdots + \alpha_1 P_1) z_\infty = -A_k^{-1} u_k = x_\infty \tag{9}$$

The conditioning of the operator :

$$\tilde{A}_k = A_k[I + \cdots + \alpha_1 P_1]$$

can be improved by a good choice of the α_i, $i = 1 \cdots k-1$. Indeed the eigenvalues of $\tilde{A}_k$ are :

$$\left\{ \begin{array}{ll} \lambda_i(1 + \sum_{i=1}^{k-1} \alpha_i) & i = 1, N_1 \\ \lambda_i(1 + \sum_{i=2}^{k-1} \alpha_i) & i = N_1 + 1, \cdots, N_2 \\ \vdots & \vdots \\ \lambda_i & i = N_{k-1} + 1, \cdots, N_k \end{array} \right. \tag{10}$$

For example in the case of the discretized Laplacian in dimension 1 with $N_{k-1} = N_k/2$, $\alpha_{k-1} = 1, \alpha_{k-i} = 2\alpha_{k-i+1}$, the conditioning of $\tilde{A}_k \simeq \frac{1}{2}$ independently of k — that is the eigenvalues of $\tilde{A}_k$ are spead on the interval $[-N_k^2, -\frac{N_k^2}{2}]$.

In general it is difficult to find a set of operators (B_j, C_j). Nevertheless, using the continuity of the eigenvalues with respect to the perturbation of the operator it is sufficient to find a set of operators (B_j, C_j) having approximatively the invariance property needed. For example, we can take for B_j a linear interpolation and for $C_j = B'_j$. This choice is enough precise — in the case of discretization of elliptic operators — to give to $\tilde{A}_k$ a conditioning < 1, independently of N_k. These considerations lead to an algorithm which solved (6) in $\delta N_k \log N_k$ with a precision of $(\frac{1}{N_k})^\delta$.

The implementation of the algorithm is a discrete version of :

$$\left\{ \begin{array}{lcl} \dot{z}_k & = & A_k[z_k + \alpha_{k-1}B_{k-1}z_{k-1} \\ & & +\alpha_{k-2}B_{k-1}B_{k-2}z_{k-2} + \cdots] + u_k \\ \dot{z}_{k-1} & = & C_{k-1}A_k z_k + A_{k-1}[\alpha_{k-1}z_{k-1} \\ & & +\alpha_{k-2}B_{k-2}z_{k-2} + \ldots] + C_{k-1}u_k \\ \dot{z}_{k-2} & = & C_{k-2}C_{k-1}A_k z_k + \alpha_{k-1}C_{k-2}A_{k-1}z_{k-1} \\ & & +A_{k-2}[\alpha_{k-2}z_{k-2} + \ldots] + C_{k-2}C_{k-1}u_k \\ & \vdots & \end{array} \right. \tag{11}$$

The properties discussed here are proved in Hackbush [12] for example. To see the meaning of the equation (11) let us consider the particular case with only one level of aggregation and $\alpha_{k-1} = 1$:

$$\left\{ \begin{array}{lcl} \dot{z}_k & = & A_k[z_k + B_{k-1}z_{k-1}] + z_k \\ \dot{z}_{k-1} & = & C_{k-1}A_k z_k + A_{k-1}z_{k-1} + C_{k-1}u_k \end{array} \right. \tag{12}$$

which can be written :

$$\left\{ \begin{array}{lcl} \dot{z}_k & = & A_k[z_k + B_{k-1}z_{k-1}] + u_k \\ \dot{z}_{k-1} & = & C_{k-1}A_k B_{k-1}z_{k-1} + C_{k-1}[A_k z_k + u_k] \end{array} \right. \tag{13}$$

z_{k-1} is a correction term obtained by solving an aggregated system — the second equation of (13) — with a forcing term equal to an aggregate of the error between z_k and the equilibrium point $-A_k^{-1}u_k$. In fact, this is the initial point of view which has conducted to the multigrid methods.

5.2 The FMGH algorithm

Let us describe completely an algorithm based on the policy iteration and multigrid ideas — it is called full multigrid Howard algorithm (FMGH), it has been designed and experimented by M. Akian [1]. It solves the DPE for diffusion processes and discounted cost :

$$\min\{A^k(u^k)V^k + c(u^k)\} = 0 \,.$$

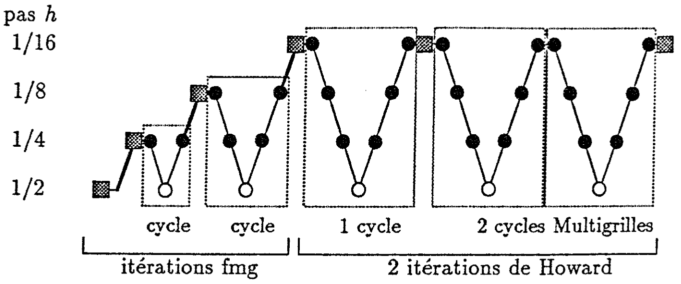

Figure 1: FMGH algorithm.

The following diagram explains the algorithm. In this diagram each horizontal line is associated with a level of discretization and the graphic symbols have the following meanings :

▨ → minimization :
$V^k \rightarrow u^k \in \arg\min A^k(u)V^k + c(u)$;

o → solution of a linear system :
$u^k \rightarrow V^k$ solution of $A^k(u^k)V^k + c(u^k) = 0$;

• → one Gauss–Seidel iteration
$(u^k, V^k) \rightarrow V^k := (L^k)^{-1}(M^k V^k + c(u^k))$
with : $A^k(u^k) = M^k - L^k$, L^k lower triangular
M^k strictly upper triangular ;

\ → restriction : $V^k \rightarrow V^{k-1} = C^k V^k$;

/ → state interpolation : $V^{k-1} \rightarrow V^k = B^k V^{k-1}$;

/ → state and control interpolation :
$u^{k-1} \rightarrow u^k = B'^k u^{k-1}$.

— in general the state interpolation operator and the control one are not the same because V and u do not belong to the same functional space.

Let us give now some numerical experiments of this algorithm. We solve, in dimension $d = 1, 2, 3$, the equations :

$$-V + \Delta V + \min_{u \in I\!R^d}(u.grad(V) + u.u) + c = 0$$

in $\Omega =]0, 1[^d$,

$$V = 0 \text{ in } \partial\Omega,$$

for different c choosen such that the solutions of the equations are :

1. in dimension 1

$$(a)\ v(x) = \sin(\pi x)$$

$$(b)\ v(x) = \sin(\pi x) + 0.1\sin(10\pi x)$$

2. in dimension 2

$$(a)\ v(x, y) = \sin(\pi x)\sin(\pi y)$$

$$(b)\ v(x, y) = \sin(\pi x)\sin(\pi y) + 0.1\sin(10\pi x)\sin(10\pi y)$$

Pb. nb.		1-a	1-b	2-a	2-b	3-a	3-b
k	h_k^2	ε_k					
0	0.25	0.6	1.4	0.44	0.44	0.32	3.8
1	0.06	0.7	32.7	0.36	24.6	0.09	1.4
2	15.10^{-3}	1	61	0.31	29.8	0.24	1.09
3	39.10^{-4}	1.1	41.2	0.28	29.7	0.53	
4	10^{-3}	1.1	21.8	.27	27.3		
5	24.10^{-5}	1.1	16.8				
6	61.10^{-6}	1.1	12.5				

Table 2: Evolution of the error of FMGH algorithm with the mesh size h_k.

Pb.nb.	1-a	1-b	2-a	2-b	3-a	3-b
k	6		4		2	
ε_k^3	0.6	15.8	0.46	8.39	0.32	1.01

Table 3: Limit error of the FMGH when the number of Howard cycles increases.

3. in dimension 3

$$(a)\ v(x, y, z) = \sin(\pi x)\sin(\pi y)\sin(\pi z)$$

$$(b)\ v(x, y, z) = \sin(\pi x)\sin(\pi y)\sin(\pi z)$$
$$+0.3\sin(3\pi x)\sin(3\pi y)\sin(3\pi z)$$

The first table shows the evolution of the error ε_k in the ascending part — decreasing of the mesh size — of the algorithm . It shows that the number of relaxation steps at each level is enough to maintain the precision wanted — of the form Ch^2 with C independent of h for a h^2 scheme of discretization. Thus *with a linear computing cost in the number of discretization points we obtain an approximated solution of precision* h^2.

6 PANDORE

DP approach is useful but it is one possible approach. Moreover this point of view is time consuming because we have to develop a special computer program for each application based always on this same idea. To help the non specialist and save time of the specialist we have designed an expert system on stochastic control called "Pandore" able to make by itself studies in this domain.

Pandore knows four points of view (to solve stochastic control problems of diffusion processes) : — dynamic programming, — decoupling, — stochastic iteration, — regular perturbation method. The more developped one is the dynamic programming.

From a specification of the model it is able to make a complete study that is : — verify the well posedness of the problem, — write the DPE, — prove the existence of a solution of the DPE in some cases, — discretize it by finite differences, — write a Fortran routine, — make a numerical test, — plot the numerical result, — write a Latex report summarizing all the study.

It knows, until now, only the classical algorithms. A generator of programs based on the multigrid written by M. Akian [1] exists and will be installed in the system soon. It is written in Lisp, Prolog and Macsyma and works on Lisp Symbolics machine. Studies like the pumping station problem have been solved using this system. A complete exemple solved by Pandore with some explanations about its structure is given in Chancelier-Gomez-Quadrat-Sulem [3].

7 FORMAL CALCULUS

Until now we have discussed numeric approximation of the solution of the DPE. Let us study now some particular cases for which it is possible to use the formal calculus. Let us suppose that we have a discrete time deterministic dynaming programming problem for which all the datas — the dynamic and cost — are piecewise linear functions ; then it must be clear — we shall give a proof later on — that the solution is piecewise linear also. Thus in this case we have only to code and manipulate this kind of functions.

A piecewise linear function can be coded by a rational function in $(\max,+)$ algebra and thus this problem — of computing the solution of the DPE — becomes a problem of manipulation and simplification of rational functions in the $(\max,+)$ algebra. A $(\max,+)$ formal polynomial calculus can be developped for that. Let us detail these points.

7.1 The $(\max,+)$ algebra

We consider $\mathbb{R}$ the set of real numbers with the two operations "max" and "+" which satisfy the following axioms :

"max" is an idempotent semigroup. It : — is associative, — is commutative, — has a neutral element $-\infty$ called ε, — is idempotent $\max(x,x)=x$.

"+" is a group, moreover : — "+" is distributive with respect to the max, — ε is absorbing $\varepsilon+x=\varepsilon$.

Such kind of idempotent semi-ring is called a dioid.

We shall denote in this paragraph max by $\oplus$ and "+" by $\otimes$ thus $2\otimes x\oplus 3$ means $\max(3,2+x)$ and we shall make the classical omission of the "$\times$" — thus $2x$ means $2\otimes x$ — and we omit also very often the circle around de "+" — thus in the new notations $2x+3$ means $\max(3,2+x)$ in the old ones. This change of notations is done because it insists on the analogy with the classical calculus.

7.2 The $(\max,+)$ rational functions

Proposition 1 *The set of piecewise linear functions are the* $(\max,+)$ *rational functions with integer slopes.*

Proof. Given a $(\max,+)$ rational function $R(x)=\frac{P(x)}{Q(x)}$ — with $P(x)$ and $Q(x)$ polynomial. It is the difference of two polynomials because the division is the inverse of the multiplication that is the difference with the usual notation. Now a polynomial $P(x)=\sum_{i=0}^{n}a_ix^i$ which means with classical notation $P(x)=\max_{i=1,n}(a_i+ix)$ that is the superior envelop of increasing piecewise–linear integer–slopes functions.

In this algebra there is the analogue of the fundamental algebra theorem.

Proposition 2 *$P(x)$ polynomial of degree n can be factorized with n linear factors :*

$$P(x)=a\prod_{i=1}^{n}(x+x_i)$$

The x_i are the corner points of $P(x)$ that is the points where the slopes changes of value.

Proof. Let us denote by x_i the corner points of $P(x)$ and k_i the multiplicity of the corner point that is the value of the increase of the slopes — which is an integer. Then $Q(x)$ defined by :

$$Q(x)=a_n\prod_{i}(x+x_i)^{k_i}$$

— where a_n is the coefficient of x^n in $P(x)$ — is an increasing piecewise–linear integer–slopes function. It changes of slope at the x_i and the value of these changes are the same as the ones of $P(x)$. Moreover $P(x)$ and $Q(x)$ coincides for x large because the leading monomials are the same. Thus $P(x)$ and $Q(x)$ are the same.

Corollaire 1 *A rational function $R(x)$ can be factored :*

$$R(x)=a\frac{\prod_j(x+x_i)}{\prod_k(x+x_k)}$$

where the "roots" x_i are the positive changes of slope and the "pole" x_k are the negative changes of slope.

7.3 Piecewise linear DPE

Let us consider the finite horizon dynamic control problem in the classical notation :

$$\begin{cases} x_{i+1}=P_{u_i}(x_i) \\ \max_{u_0,u_1,\ldots,u_{n-1}}\sum_{i=0}^{n-1}c_{u_i}(x_i) \end{cases} \quad u\in\mathcal{U} \text{ finite set}$$

Let us suppose that the $P_u(x)$ and $c_u(x)$ are piecewise linear functions with integer slopes — that is rational in the $(\max, +)$ algebra. Then the DPE $V_i(x)$ satisfies :

$$\begin{cases} V_i(x) &= \max_u\{V_{i+1}(P_u(x)) + c_u(x)\} \\ V_n(x) &= 0 \end{cases}$$

which can be written with the $(\max, +)$ notations :

$$\begin{cases} V_i(x) &= \sum_{u\in\mathcal{U}} c_u(x)V_{i+1}(P_u(x)) \\ V_n(x) &= 0 \end{cases} \qquad (14)$$

We call such DPE a rational one in the $(\max, +)$ sense.

Théorème 1 *The solution of a rational DPE in the* $(\max, +)$ *sense is rational.*

Proof. In (14) we use only rational transformations and start with a rational function thus the result is rational •

The resolution of (14) can be done formally using the $(\max, +)$ operations and can be seen as maintaining a canonical form — for example the factorized one — that is computing the corner points of the solution. The experience accumulated in the formal calculus can be adapted at this new situation to solve the problem.

This point of view may be developped. It can solve the difficulty of the numerical diffusion which appears when we want solve the deterministic DPE by numerical methods.

References

[1] M.Akian, *Résolution numérique d'équations d'Hamilton Jacobi Bellman au moyen d'algorithmes multigrilles et d'itérations sur les politiques*, Lectures Notes in Control and Inf. Sc. Nb.111 Springer Verlag, 1988.

[2] A. Bensoussan, *Méthodes de perturbation en contrôle optimal*, Wiley/Gauthier-Villars.

[3] J.P. Chancelier, C. Gomez, J.P. Quadrat, A. Sulem, *Pandore*, Nato Course Ciocco 1987, to appear.

[4] G. Cohen, P. Moller, J.P. Quadrat, M. Viot, *Algebraic tools for the performance evaluation of discrete event system*, to appear special issue IEEE Proceedings on Discrete Event Dynamic Systems.

[5] R.A. Cuninghame-Green, *Minimax algebra*, Springer Verlag, Berlin 1979.

[6] C.Derman, *Finite State Markovian Decision Processes.* Academic Press 1970.

[7] M.Falcone, *Numerical solution of deterministic continuous control problems.* Int. Symp. on Num. Anal. Madrid 1985.

[8] W. Fleming, *Control for small noise intensities*, SIAM J. Control, Vol. 9, n3, 1971.

[9] M. Gondran, M. Minoux, *Linear algebra in dioids : a survey of recents results*, Annals of Discrete Math., Vol. 19, pp. 147-164, 1984.

[10] R.Gonzales,E.Rofman,*On determ. control problems: an approx. procedure for the optimal cost.The static case* Siam J. of control Vol 23,2 85.

[11] M. Goursat, J.P. Quadrat, *Optimal stochastic control : numerical methods*, in systems and control encyclopedia, Edited M. Singh, 1987.

[12] W. Hackbush, U. Trottenberg, *Multigrids methods*, Lectures notes in Math. n 960, Springer Verlag, 1981.

[13] H.J. Kushner, *Probability methods in stochastic control and for elliptic equations*, Academic Press, 1977.

[14] P. Lederer, Ph. Torrion, J.P. Bouttes, *Maquette du système production-consommation*, Note Nb. 744 Electricté de France, Etudes Economiques Générales.

[15] P.L. Lions, *Optimal control : Hamilton Jacobi equation*, in Systems and control encyclopedia edited by M. Singh, 1987.

[16] G. Marchouk, *Méthodes de calcul numérique*, Edition Mir, Trad. Française, 1980.

[17] J.P. Quadrat,*Analyse Numérique de l'équation de Bellman stochastique*, Rapport Laboria N.140.

[18] S.M.Ross, *Introduction to stochastic dynamic programming*, Academic Press, 1983.

[19] A. Turgeon, *Optimal operation of multireservoir power systems with stochastic inflows water resources*, Water Research April 1980.