

The Extended Ritz Method Versus Dynamic Programming in Water Reservoirs Management Under Uncertainty

MARCO BAGLIETTO,* CRISTIANO CERVELLERA,** MARCELLO SANGUINETI,*
and RICCARDO ZOPPOLI*

(*) Department of Communications, Computer and System Sciences, DIST - University of Genoa, Via Opera Pia 13, 16145 Genoa, Italy {mbaglietto,marcello,rzop}@dist.unige.it

(**) Institute of Intelligent Systems for Automation (ISSIA-CNR), National Research Council of Italy, Genoa Branch - Via De Marini 6, 16149 Genoa, Italy , cervellera@ge.issia.cnr.it

Abstract: The Extended Ritz Method is a new technique for the solution of general functional optimization problems. The management of a system of reservoirs can be modeled as a T -stage stochastic optimal decision problem that is a particular case of functional optimization. This problem is usually solved by dynamic programming. Both methods may incur Bellman's risk of dimensionality. The possibility of mitigating such a risk is addressed.

Key words: water reservoir management, dynamic programming, Extended Ritz Method, curse of dimensionality, neural networks.

1 INTRODUCTION AND STATEMENT OF THE PROBLEM

As is well-known, T -stage stochastic optimal control problems can be solved analytically by dynamic programming (DP) if suitable hypotheses on the state equation (where the state is measurable and represents the amount of water in the reservoirs) and the cost function are verified. Typically, analytical solutions by DP can be obtained if the state equation is linear and the cost is quadratic. If such hypotheses do not hold true, as is generally the case with the problems at hand, one has to search for approximate solutions. Approximate solutions can be found by discretizing the state space and applying DP. In such a way, the DP functional equation has to be solved only for a finite number of state values. However, given a fixed number d of discretization levels in each of the n components of the state vector, the use of an n -dimensional grid causes the total number of discretized points to grow as d^n . This exponential growth is known as Bellman's "curse of dimensionality," and limits the use of regular grids to small values of n . Various methods to cope with the curse of dimensionality in DP applied to reservoirs management problems have been proposed, such as the adoption of simpler models (Turgeon 1981, Archibald, et al. 1997) and the use of smart approximators for the cost-to-go functions. The latter method seems to be the one that has been applied more effectively. In particular, cubic Hermite polynomials (Foufoula-Georgiou and Kitanidis 1988) and spline interpolation (Johnson, et al. 1993) can be mentioned. More recently, only reservoir control problems with up to seven state variables have been faced (Philbrick and Kitanidis 1999). These strategies use fewer grid points to obtain a given precision in the approximation of the cost-to-go functions but still need a uniform discretization of the state space, thus leaving the curse of dimensionality eventually unavoided. In recent years, attention has been focused

on efficient deterministic sampling of the state space. For instance, in (Chen 1999) a 9-dimensional inventory forecasting problem, similar to a water resources management one, has been faced by using special subsets of the full uniform grid known as Orthogonal Arrays.

Let us consider the discrete-time dynamic system

$$\mathbf{x}_{t+1} = \mathbf{f}_t(\mathbf{x}_t, \mathbf{u}_t, \boldsymbol{\xi}_t), \quad t = 0, 1, \dots, T-1 \quad (1)$$

$$\mathbf{u}_t \in U_t \subseteq \mathbb{R}^m, \quad (2)$$

where $\mathbf{x}_t \in \mathbb{R}^n$ is the state vector (assumed to be perfectly measurable), $\mathbf{x}_0 \in \hat{\mathbf{x}}$ is a given initial state, \mathbf{u}_t is the control vector, U_t is the set of admissible controls, and $\boldsymbol{\xi}_t \in \mathbb{R}^q$, $t = 0, \dots, T-1$ are random noises characterized by a known probability density. Then, we formalize the T -stage stochastic optimal control problem as follows.

Problem TS. Find the sequence of optimal control functions $\mathbf{u}_t^o = \boldsymbol{\mu}_t^o(\mathbf{x}_t)$, $t = 0, \dots, T-1$, that minimize the expected cost

$$J(\mathbf{x}_0) = \mathbf{E}_{\boldsymbol{\xi}_0, \dots, \boldsymbol{\xi}_{T-1}} \left\{ \sum_{t=0}^{T-1} h_t(\mathbf{x}_t, \mathbf{u}_t, \boldsymbol{\xi}_t) + h_T(\mathbf{x}_T) \right\} \quad (3)$$

subject to the constraints (1) and (2).

2 THE EXTENDED RITZ METHOD (ERIM)

Let us address the following functional optimization problem, which is by far more general than Problem TS.

Problem P.
$$\min_{\Gamma \in S_M} \mathbf{E}_z \left\{ F \left[\boldsymbol{\gamma}^1(y_1), \dots, \boldsymbol{\gamma}^M(y_M), z \right] \right\}.$$

$\Gamma \in S_M$ and each $\boldsymbol{\gamma}^i$ belongs to an infinite-dimensional real linear space H_i . z is a random vector with a known probability distribution. The arguments y_i depend on z and possibly on other decision functions $\boldsymbol{\gamma}^j$, $j \neq i$, through known mappings. F is a cost functional. Problem TS is an example of Problem P. Other examples of Problem P are given by team optimal decision problems, optimal estimation and filtering problems, and wide classes of variational problems where the unknowns are given by functions. In such examples of Problem P, DP can face Problem TS but not the other ones. As for Problem TS, deriving optimal solutions to Problem P analytically is feasible in very few situations. Not to get burdened with heavy notation and without loss of generality, the theoretical facts regarding functional optimization will be considered by addressing Problem P in which only one decision function $\boldsymbol{\gamma}(y) \in S \subseteq S_1$ is present.

There are several ways of approximately solving Problem P. Our approximate method, i.e., the ERIM, consists in constraining the functions $\boldsymbol{\gamma}(y)$ to take on the form $\boldsymbol{\gamma}_v(y, \mathbf{w}_v)$, where

$\gamma_v(\cdot, \cdot)$ has a fixed structure, $v \in \mathbb{N}^+$, and \mathbf{w}_v is a vector of parameters to be determined. A simple and quite natural choice for functions $\gamma_v(\mathbf{y}, \mathbf{w}_v)$ may lie in using, for each of the n_c components of γ_v , a *linear combination of v preassigned basis functions* of the form

$$\gamma_v(\mathbf{y}, \mathbf{w}_v) = \text{col} \left[\sum_{i=1}^v c_{ij} \phi(\mathbf{y}, \boldsymbol{\kappa}_i) : j=1, \dots, n_c \right], \quad (5)$$

where c_{ij} and $\boldsymbol{\kappa}_i$ are “free” parameters and \mathbf{w}_v has such parameters as components. We call functions with the structures (5) “*one hidden layer networks*” (“*OHL networks*”). OHL networks whose basis functions *contain (do not contain)* the vectors $\boldsymbol{\kappa}_i$ are defined as *nonlinear (linear) OHL networks*. For both types of networks, the number of parameters (i.e., the dimension of \mathbf{w}_v) is proportional to v . Then we call v *the complexity number of an OHL network*. Examples of nonlinear OHL networks are given by feedforward neural networks, radial basis functions, hinging hyperplanes, etc.

Substitution of $\gamma(\mathbf{y})$ with $\gamma_v(\mathbf{y}, \mathbf{w}_v)$ into the functional $F(\gamma)$ gives $F_v(\mathbf{w}_v) \stackrel{\text{Q}}{=} F[\gamma_v(\cdot, \mathbf{w}_v)]$. Therefore, the original functional optimization Problem TS is reduced, for $v = 1, 2, \dots$, to a sequence of “approximating” *nonlinear (in general) programming problems*, which may be hopefully solved by some descent algorithm. Each of them can be stated as follows.

Problem P_v .
$$\inf_{\mathbf{w}_v \in W_v} F_v(\mathbf{w}_v) = \inf_{\mathbf{w}_v \in W_v} \mathbb{E}_z [J_v(\mathbf{w}_v, z)].$$

$W_v \subseteq \mathbb{R}^{N(v)}$ is the set of admissible vectors \mathbf{w}_v , defined by the constraints that originally defined S in Problem TS, that is, $W_v \stackrel{\text{Q}}{=} \{\mathbf{w}_v : \gamma_v(\cdot, \mathbf{w}_v) \in A_v \cap S\}$. If the OHL networks are linear, then \mathbf{w}_v is the vector of the coefficients c_{ij} in (5), and the procedure leading to Problems P_v bears a close resemblance to the classical Ritz method for the calculus of variations. Instead, if the OHL networks are nonlinear, the procedure gives rise to an extension of the Ritz method, which we have called the “Extended Ritz Method.” If the optimal solutions to Problem TS belong to classes of d -variable functions provided with suitable regularity properties, the ERIM may outperform the Ritz method in that the number of free parameters increases moderately (e.g., polynomially) with d , whereas the latter method may be ruled out by the curse of dimensionality. This superiority should derive from the “flexibility” characterizing the parametrized basis functions.

We now assume 1) an optimal solution γ° to Problem TS to exist, 2) for any $v \in \mathbb{N}^+$, an optimal solution \mathbf{w}_v° to Problem P_v to exist. In order to simplify the notation, we define:

$$F^\circ \stackrel{\text{Q}}{=} F(\gamma^\circ) = \min_{\gamma \in S} F(\gamma); \quad F_v^\circ \stackrel{\text{Q}}{=} F[\gamma_v(\cdot, \mathbf{w}_v^\circ)] = \min_{\mathbf{w}_v \in W_v} F_v(\mathbf{w}_v); \quad \gamma^\circ(\cdot) \stackrel{\text{Q}}{=} \gamma_v(\cdot, \mathbf{w}_v^\circ).$$

We need to equip the space H with a norm $\|\cdot\|$. Then, we replace H with the normed linear space $H \stackrel{\text{Q}}{=} (H, \|\cdot\|)$. We are now in a position to state formally that a sequence of

Problems P_v approximate better and better Problem TS if both following assumptions hold true:

A1. The sequence $\{\gamma_v^o\}_{v=1}^{\infty}$ is such that $\lim_{v \rightarrow \infty} F(\gamma_v^o) = F^o$.

A2. The sequence $\{\gamma_v^o\}_{v=1}^{\infty}$ has a limit function γ^o , i.e., $\lim_{v \rightarrow \infty} \|\gamma_v^o - \gamma^o\| = 0$.

It is useful to introduce the following definition.

Definition 1. A sequence $\{\gamma_v^o\}_{v=1}^{\infty}$ that verifies both Assumption A1 and Assumption A2 is defined as “P-optimizing sequence.” The OHL networks making up this sequence are defined as “P-optimizing networks.” We also define as “linear (nonlinear) P-optimizing sequences” the ones implemented by linear (nonlinear) OHL networks.

We now introduce an important type of P-optimizing networks by addressing the speed of convergence to zero of the error sequences $\{F(\gamma_v^o) - F^o\}_{v=1}^{\infty}$ and $\{\|\gamma_v^o - \gamma^o\|\}_{v=1}^{\infty}$, that is, the speed at which the optimal solutions of Problems $P_1, P_2, \dots, P_v, \dots$ converge to the optimal solutions of Problem P. Of course, it is very important that the convergence should remain sufficiently fast even when the dimension d of the vector \mathbf{y} takes on large values. In order to make the above concept explicit, we establish a relationship among the following three quantities: i) the complexity number v , ii) the dimension d , and iii) the approximation accuracy ε . As regards the third term, we say that Problem P_v approximates Problem P with the accuracy ε if both errors $F(\gamma_v^o) - F^o$ and $\|\gamma_v^o - \gamma^o\|$ are at most equal to ε .

Let us now consider a certain P-optimizing sequence. The expression “the convergence of the optimal solution of Problem P_v to an optimal solution of Problem P is sufficiently fast” can be

described quantitatively by the two following conditions: A) there exist $p, q \in \mathbb{N}^+$, such that

$F(\gamma_v^o) - F^o \leq O(d^p / v^q)$; B) there exist $p', q' \in \mathbb{N}^+$ such that $\|\gamma_v^o - \gamma^o\| \leq O(d^{p'} / v^{q'})$. If both

conditions A and B hold true, it is immediate to show that in order to have an error of approximate solution not larger than ε it is sufficient to choose a number v such that

$$v \geq \max \left[\left(\frac{c}{\varepsilon} \right)^{1/q} d^{p/q}, \left(\frac{c'}{\varepsilon} \right)^{1/q'} d^{p'/q'} \right]. \quad (6)$$

To sum up, P-optimizing sequences for which (6) is verified enable one to derive solutions to Problem P (approximate up to any desired degree of accuracy) by using networks containing a number of parameters growing at most as a power of d . This might avoid the curse of dimensionality. Therefore, we are motivated to define the following class of OHL networks.

Definition 2. A P -optimizing sequence $\{\gamma_v^o\}_{v=1}^\infty$ that verifies both conditions A and B is defined as “polynomially complex P -optimizing sequence.” The OHL networks making up this sequence are defined as “polynomially complex P -optimizing networks.”

Mathematical assumptions on Problem P and the OHL networks to allow conditions A and B to be verified are given in (Kůrková and Sanguineti 2004). To solve Problem P_v , we address gradient algorithms mainly for their simplicity. This leads us to introduce the concept of “stochastic approximation” in a straightforward way. $J_v(\mathbf{w}_v, \mathbf{z})$ is required to be a C^1 function with respect to \mathbf{w}_v for all \mathbf{z} . Under mild additional assumptions, also $F_v(\mathbf{w}_v)$ is a C^1 function. However, due to the very general assumptions under which Problem P_v has been stated, it is practically impossible to compute the gradient $\nabla_{\mathbf{w}_v} E_z J_v(\mathbf{w}_v, \mathbf{z})$ analytically. Indeed, we should calculate, at each iteration step, a multiple integral of a very complex function and its gradient. Stochastic approximation enables us to use only $\nabla_{\mathbf{w}_v} J_v(\mathbf{w}_v, \mathbf{z})$ (which can be easily computed) instead of $\nabla_{\mathbf{w}_v} E_z J_v(\mathbf{w}_v, \mathbf{z})$. To sum up, we use the updating algorithm

$$\mathbf{w}_v(k+1) = \mathbf{w}_v(k) - \alpha(k) \nabla_{\mathbf{w}_v} J_v[\mathbf{w}_v(k), \mathbf{z}(k)], \quad k=0, 1, \dots, \quad (7)$$

where the sequence $\{\mathbf{z}(k)\}$ is generated randomly according to the known probability distribution of \mathbf{z} , and $\alpha(k)$ is a suitably decreasing positive stepsize.

3. SOLUTION OF THE T-STAGE PROBLEM TS BY THE ERIM

We have seen that the ERIM consists in constraining the admissible decision functions $\mathbf{u}_t = \mu_t(\mathbf{x}_t)$, $t=0, \dots, T-1$, to take on the structures of OHL networks, that is,

$$\hat{\mathbf{u}}_t = \hat{\mu}(\mathbf{x}_t, \mathbf{w}_{N_t}), \quad t=1, \dots, T-1. \quad (8)$$

We have assumed that the OHL networks remain of the same type stage after stage. Then the integer t does not appear as an argument of the networks, for they depend on t only through the parameter vector \mathbf{w}_{N_t} and its dimension, i.e., v_t . Let us now substitute

$\mu_1(\mathbf{x}_1), \dots, \mu_{T-1}(\mathbf{x}_{T-1})$ with the OHL networks (8) into (3). Similarly, we substitute $\mathbf{u}_1, \dots, \mathbf{u}_{T-1}$

with the networks (8) into (1). Finally, we eliminate $\mathbf{x}_1, \dots, \mathbf{x}_T$ from (3) by using the state equation repeatedly. After these operations, the cost (3) is no longer a functional. Indeed, it

does not depend any more on the functions μ_1, \dots, μ_{T-1} but turns out to be a function of the variables that escaped from the elimination. We can write it in the compact form $E_{\xi} J_v(\mathbf{u}_0, \mathbf{w}_v, \mathbf{x}_0, \xi)$, where $\xi \in \mathbb{Q} \text{ col}(\xi_0, \xi_1, \dots, \xi_{T-1})$ and $\mathbf{w} \in \mathbb{Q} \text{ col}(\mathbf{w}_{1v_1}, \dots, \mathbf{w}_{T-1, v_{T-1}})$. The integer v in J_v and \mathbf{w}_v is related to the numbers v_1, \dots, v_{T-1} in the networks (8). When the dimensions of such networks (8) grow, v can be expressed, through some simple conventional rule, as a function of the integers v_1, \dots, v_{T-1} .

In order to apply the algorithm (7), we have to derive, at each iteration step k , the components of the gradient $\nabla_{\mathbf{u}_0, \mathbf{w}_v} J_v[\mathbf{u}_0(k), \mathbf{w}_v(k), \mathbf{x}_0, \xi(k)]$, i.e., the partial derivatives $\partial J_v / \partial \mathbf{u}_0$, $\partial J_v / \partial \mathbf{w}_{v_t}$, $t=1, \dots, T-1$. To simplify the notation, in the following we shall drop the subscript v_t and the index k . It can be shown (see, for example, (Parisini and Zoppoli 1994)) that the required partial derivatives can be computed by the following relationships:

$$\begin{aligned} \frac{\partial J_v}{\partial \mathbf{w}_t} &= \frac{\partial J_v}{\partial \mathbf{u}_t} \frac{\partial \mathbf{h}(\mathbf{x}_t, \mathbf{w}_t)}{\partial \mathbf{w}_t}, \quad t=1, \dots, T-1 \\ \lambda_t^\top &= \frac{\partial h(\mathbf{x}_t, \mathbf{u}_t, \xi_t)}{\partial \mathbf{u}_t} + \lambda_{t+1}^\top \frac{\partial f_t(\mathbf{x}_t, \mathbf{u}_t, \xi_t)}{\partial \mathbf{x}_t} + \frac{\partial J_v}{\partial \mathbf{u}_t} \frac{\partial \mathbf{h}(\mathbf{x}_t, \mathbf{w}_t)}{\partial \mathbf{x}_t}, \quad t=T-1, \dots, 1 \\ \lambda_T^\top &= \frac{\partial h_T(\mathbf{x}_T)}{\partial \mathbf{x}_T}. \end{aligned}$$

$\partial J_v / \partial \mathbf{u}_0$ can be derived in a simple way. A good deal of experimental results have shown the possibility of solving Problem TS (and many other functional optimization problems) by the ERIM without incurring the curse of dimensionality. Generally, OHL neural networks have been used; see, for example, (Zoppoli, et. al. 2002) and the references therein. From a theoretical point of view, the results obtained in (Kůrková and Sanguineti 2004) have to be extended from Problem P to the more complex Problem TS. Toward this end, studies are currently in progress.

4. APPROXIMATE SOLUTION OF THE T-STAGE PROBLEM TS BY DP

For the sake of brevity, we do not write the backward recursive equation of DP. We limit ourselves to describing some choices made in order to mitigate the danger of the curse of dimensionality. 1. To compute the optimal cost-to-go $J_t^o(\mathbf{x}_t)$ at each decisional stage, we define the sets $X_{L,t} \in \mathbb{Q} \{ \mathbf{x}_t^{(l)}, l=1, \dots, L \}$, $t=1, \dots, T-1$, made up of L *sample vectors* in the

state space. Such vectors have been chosen by Montecarlo techniques to avoid the use of regular grids. Other choices, based on deterministic sequences, are possible and currently under test. Among them, we can cite statistical designs such as Orthogonal Arrays and Latin Hypercubes (Chen, et. al. 2003) (which are different kinds of “smart subsets” of the full uniform grid), and number-theoretic samplings such as low-discrepancy sequences (Niederreiter 1992), commonly employed for quasi-Monte Carlo numerical integration. 2. In applying DP, the minimizations with respect to \mathbf{u}_t cannot be performed “exactly” since, in general, each next-stage vector $\mathbf{x}_{t+1} = \mathbf{f}(\mathbf{x}_t^{(l)}, \mathbf{u}_t, \xi_t)$, $\mathbf{x}_t^{(l)} \in X_{tL}$, does not coincide with any quantized state $\mathbf{x}_{t+1}^{(m)} \in X_{t+1,L}$. Then, outside the points of X_{tL} , we need approximate values of the costs $J_t^o(\mathbf{x}_t)$. Such values can be obtained by optimizing (we use a least-squares criterion) OHL networks of the form (cf. (5))

$$J_t(\mathbf{x}_t, \mathbf{w}_{N_t}^J) = \sum_{i=1}^{v_t} c_{it} \varphi(\mathbf{x}_t, \boldsymbol{\kappa}_{it}), \quad t = 1, \dots, T-1. \quad (8)$$

3. Not to incur the curse of dimensionality, the networks (8) have to be provided with different properties from the ones given in Definitions 1 and 2. Actually, instead of approximating optimal solutions to functional optimization problems, they are required to approximate functions, again by a moderate number of parameters. This task is generally simpler and much more investigated in the literature. OHL networks provided with such different properties have been defined as *polynomially complex approximating networks* and are described, for example, in (Zoppoli, et. al. 2002).

Remark. It is important to point out that the mutual independence among the random vectors $\xi_0, \xi_1, \dots, \xi_{T-1}$ is required for the application of DP but not for the ERIM.

5. OPTIMAL CONTROL OF A SYSTEM OF RESERVOIRS

The problem presented in this section has been faced in the literature many times, very often by using some techniques related to DP (Foufoula-Georgiou and Kitanidis 1988, Johnson, et. al. 1993, Yakowitz 1982, Philbrick and Kitanidis 2001, Baglietto, et. al. 2000), for which it has become a sort of benchmark. A 10-reservoir system with $T=5$ (see Fig.1), has been chosen. The system is the same as presented (for the deterministic case) in (Yakowitz 1982). The state equation has the form

$$x_{t+1}^j = \min(x_t^j - u_t^j + \sum_{h \in I_j^+} u_t^h + \xi_t^j, x_{\max}^j), \quad t = 0, \dots, 5, \quad j = 1, \dots, 10$$

$$0 \leq u_t^j \leq x_t^j + \sum_{h \in I_j^+} u_t^h + \xi_{\min}^j.$$

x_t^j is the amount of water in the j -th reservoir at the beginning of the t -th period, u_t^j is the amount of water released from the j -th reservoir during the t -th period (planned at the beginning of the period), I_j^+ is the set of indexes defining the reservoirs that release water

directly into the j -th one, and ξ_t^j is the natural inflow of water into the j -th reservoir during the t -th period. The inflows are stochastic and are given by the sums of deterministic minimum river inflows and stochastic rain inflows. The random variables ξ_t^j are mutually independent with uniform probability density functions between $\xi_{\min} = \text{col}(1, 1, 1, 0, 0.7, 0.7, 0, 1.5, 0, 0)$ and $\xi_{\max} = \text{col}(1.5, 1.5, 1.5, 0, 1.2, 1.2, 0, 2, 0, 0)$. The presence of a floodway for each reservoir is assumed, so that the state vector is never larger than $\mathbf{x}_{\max} = \text{col}(10, 10, 10, 10, 10, 10, 10, 10, 18, 25)$. The benefit function to be maximized is given by

$$J(\mathbf{x}_0) = \mathbb{E} \left\{ \sum_{t=0}^5 (c_p^\top \mathbf{u}_t + c_f u_t^{10}) - 100 \sum_{j=1}^{10} \max(0, x_{\min}^j - x_5^j)^2 \right\} \quad (9)$$

where $c_p = \text{col}(1.1, 1, 1, 1.2, 1.1, 1, 1.2, 1, 1.2, 2.5)$ describes the benefits for the release of water from each reservoir (for example, power generation) and $c_f = 1.9$ is the coefficient corresponding to an irrigation benefit, supposed to come only from the 10-th reservoir. J contains a penalty function that takes into account the constraint $x_5 \leq x_{\min}$, where $\mathbf{x}_{\min} = \text{col}(3, 3, 3, 3, 3, 3, 3, 3, 5, 6)$. In applying DP, OHL neural networks with $v = 40$ basis functions were used to approximate the optimal cost-to-go functions. Each network was optimized by using $L=1800$ sample pairs (state vector - optimal cost-to-go function). In applying the ERIM, OHL neural networks with $v = 40$ were used again to determine the suboptimal control functions $\mathbf{u}(\mathbf{x}_t, \mathbf{w}_{N_t}^o)$, $t=1, \dots, 4$. Since stochastic approximation utilizes the gradient algorithm (7), suitable differentiable penalty functions have been added to the benefit cost (9) in order to implement the various constraints. The results obtained by both methods have been tested by computing a mean benefit over different sequences of inflows, randomly extracted according to their probability density functions, each for $\mathbf{x}_0 = \mathbf{x}_{\min}$. The mean benefit for DP turned out to be 282, whereas the ERIM gave a benefit of 285.

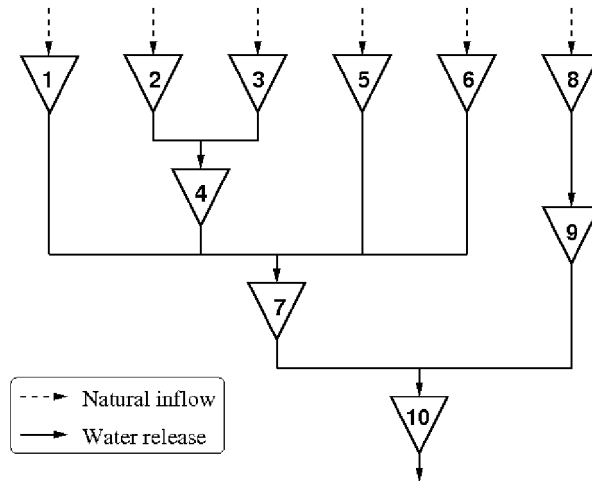


Fig. 1 Ten reservoirs configuration.

We point out that DP, together with the sampling of the state space based on low-discrepancy sequences, was recently proven to be successful in solving the optimal control problem for a 30-dimensional model (Cervellera, et. al. 2005). The new model is still made up of 10 reservoirs. The larger dimension of the state space is due to a more realistic modeling of the

random inflows implemented by an autoregressive process. A comparison between DP and the ERIM is currently under development for this 30-dimensional model.

REFERENCES

- Archibald, T. W., Mckinnon, K. I. M., and Thomas, L. C. (1997), An aggregate stochastic dynamic programming model of multireservoir systems, *Water Resour. Res.*, Vol. 33, pp. 333-340.
- Baglietto, M., Cervellera, C., Parisini, T., Sanguineti, M., and Zoppoli, R. (2000), Approximating networks, dynamic programming and stochastic approximation, *Proc. American Control Conference*, Chicago, IL, pp. 3304-3308, 2000.
- Chen, V. C. P., Ruppert, D., And Shoemaker, C. A. (1999), Applying experimental design and regression splines to high-dimensional continuous-state stochastic dynamic programming, *Operations Research*, Vol. 47, pp. 38-53.
- Chen, V.C.P., Tsui, K.-L., Barton, R. R., Allen, J. K. (2003), A review of design and modeling in computer experiments, in: C.R. Rao, R. Khattree (Eds.), *Handbook in Industrial Statistics*, Vol. 22, Elsevier Science, Amsterdam.
- Cervellera, C., Chen, V. C. P., and Wen, A. (2005), Optimization of a Large-Scale Water Reservoir Network by Stochastic Dynamic Programming with Efficient State Space Discretization, *European Journal of Operational Research*, to appear.
- Foufoula-Georgiou, E. and Kitanidis, P.K. (1988), Gradient dynamic programming for stochastic optimal control of multidimensional water resources systems, *Water Resour. Res.*, Vol. 24, pp. 1345-1359.
- Johnson, S. A., Stedinger, J. R., Shoemaker, C., Li. Y., and Tejada-Guibert, J. A. (1993), Numerical solution of continuous-state dynamic programs using linear and spline interpolation, *Operations Research*, Vol. 41, pp. 484-500.
- Kůrková, V., and Sanguineti, M. (2004), Error estimates for approximate optimization by the extended Ritz method, *SIAM J. on Optimization*, vol. 15, pp. 461-487, 2005.
- Niederreiter, H.(1992), *Random Number Generation and Quasi-Monte Carlo Methods*, SIAM, Philadelphia.
- Parisini, T., and Zoppoli, R.(1994), Neural Networks for Feedback Feedforward Nonlinear Control Systems, *IEEE Trans. Neural Networks*, Vol. 5, pp. 436-449.
- Philbrick, C. R. Jr., and Kitanidis, P. K. (2001), Improved dynamic programming methods for optimal control of lumped-parameter stochastic systems, *Operations Research*, Vol. 49, pp. 398-412.
- Turgeon, A. (1981), A decomposition method for the long-term scheduling of reservoirs in series, *Water Resour. Res.*, Vol. 17, pp. 1565-1570.
- Yakowitz, S. (1982), Dynamic programming applications in water resources, *Water Resour. Res.*, Vol. 18, pp. 673-696.
- Zoppoli, R., Sanguineti, M., and Parisini, T. (2002), Approximating Networks and the Extended Ritz Method for the Solution of Functional Optimization Problems, *J. Optimization Theory and Applications*, Vol. 112, pp. 403-439.