19th International Congress on Modelling and Simulation, Perth, Australia, 12–16 December 2011
http://mssanz.org.au/modsim2011

A data-driven Dynamic Emulation Modelling approach for the management of large, distributed water resources systems

A. Castelletti[a], **S. Galelli**[b], **M. Restelli**[a], **R. Soncini-Sessa**[a]

[a]*Dipartimento di Elettronica e Informazione, Politecnico di Milano, Milano, Italy*
[b]*Singapore-Delft Water Alliance, National University of Singapore, Singapore*
Email: sdwgs@nus.edu.sg

Abstract: Water resources engineering and hydrology focus predominantly on physically-based models to characterize the dynamics of the physical, social and economic processes. Such a high fidelity models are usually computationally expensive and cannot be used in problems requiring hundreds or thousands of model runs to be satisfactory solved. Typical examples include optimal planning and management, data assimilation, and sensitivity analysis. An effective approach to overcome this limitation is to perform a top-down reduction of the physically-based model by identifying a simplified, computationally efficient emulator, constructed from and then used in place of the original physically-based model in highly resource-demanding tasks. In this work we propose a new data-driven Dynamic Emulation Modeling (DEMo) approach that combines the advantages of data-based modeling in representing complex, non-linear relationships, and preserves the state-space representation, which is both a precondition to infer an ex-post physically meaningful interpretation of the emulator and particularly effective in some applications (e.g. optimal management and data assimilation). The core mechanism of the proposed approach is a novel variable selection procedure based on a class of tree-based methods that is recursively applied to a data-set of input, state and output variables generated via simulation of the physically-based model. The approach embodies some very important properties: it is fully automated, independent on domain experts and system knowledge, and suitable for non-linear processes; it has a high potential in terms of complexity reduction; and, finally, it provides an ex-post interpretation of the emulator structure. The approach is demonstrated on a real-world case study concerning the optimal operation of a selective withdrawal reservoir suffering from algal blooms due to thermal stratification. The emulator, which is identified on a data-set generated with the 1D coupled hydrodynamic-ecological model DYRESM-CAEDYM, shows good performances in emulating the dynamic behaviour of the original model in characterizing the chlorophyll-a concentration in the euphotic layer.

Keywords: Emulation modelling, Data-driven modelling, Input variable selection, Large water systems

1 BACKGROUND

An emulator is a computationally efficient, low-order approximation of a physically-based model. Depending on whether the emulator preserves the dynamic nature of the original model or is a static relationship between inputs, (states), and outputs, two methodological approaches can be distinguished (Castelletti et al., 2011): Dynamic Emulation Modelling (DEMo) and non-dynamic emulation modelling. This latter has been considerably explored in the water resources literature, with applications in the planning of water distribution networks (Broad et al., 2005), groundwater (Yan and Minsker, 2006) and surface water resources (Castelletti et al., 2010). Non-dynamic emulators provide a static map of the planning decision into the objective functions of an optimization problem. As a consequence, they are limited to simulation-based optimization frameworks, whereas cannot be employed in any control problem, where a dynamic, yet approximated, description of the immediate-cost associated to each state transition is required.

Dynamic emulators, which maintain the dynamic property of the original model, can be categorized into structure and data-driven. The former are based on some projection of the high-dimension equations of the physically-based model onto a lower-dimension space, where the model equations are solved for the substituted projected states; the latter are based on the identification of the emulator as an I/O relationship over a data set of input-output samples generated by the physically-based model. The choice for one approach over the other depends on the level of complexity and non-linearities embedded into the original model.

Structure-driven dynamic emulators are well developed for linear, quadratic and weakly non-linear models, while theory is still under development for non-linear models (see Antoulas (2005) and references therein). A structure-driven emulator is naturally in the state-space form, which makes it directly and more effectively usable in any management problem, but because of the many difficulties in dealing with strong non-linearities, structure-driven DEMo has been only relatively adopted in the environmental field (Crout et al., 2009). Data-driven emulators are more flexible and powerful in characterizing the non-linear relationships between external drivers and output but usually are in an input-output representation, which is less efficient for management problems. Moreover, the original input-output representation can be hardly given a physical interpretation and the final emulator may lack of credibility by stakeholders and domain experts. Data-driven DEMo has been more extensively explored than its structure-driven twin in the environmental modelling literature (see van der Merwe et al. (2007) and references therein).

2 DYNAMIC EMULATION MODELLING

This section formulates the Dynamic Emulation Modelling (DEMo) problem and summarize the essential background material for the subsequent sections, where the novel contribution of the paper is described and demonstrated.

2.1 DEMo problem formulation

Given a physically-based model $\mathcal{M}$ with state and exogenous driver vectors $\mathbf{X}_t$ and $\mathbf{W}_t$, the purpose of a DEMo exercise is to identify a computationally efficient, low-order model (the emulator) on a data-set appropriately generated via simulation of model $\mathcal{M}$. The emulator must be such that its output $\mathbf{y}_t$ accurately reproduces model $\mathcal{M}$'s output $\mathbf{Y}_t$, but has lower-dimension state and exogenous driver vectors $\mathbf{x}_t$ and $\mathbf{w}_t$, and takes the following general state-space form:

$$\mathbf{x}_{t+1} = \mathbf{f}_t(\mathbf{x}_t, \mathbf{w}_t, \mathbf{u}_t) \tag{1a}$$

$$\mathbf{y}_t = \mathbf{h}_t(\mathbf{x}_t, \mathbf{w}_t, \mathbf{u}_t) \tag{1b}$$

where $\mathbf{u}_t$ is the control vector, $\mathbf{f}_t(\cdot)$ a generally non-linear, time-variant, function that models the state transitions, and $\mathbf{h}_t(\cdot)$ the output transformation function. Since the emulator is to be used in management problems formulated as sequential decision-making processes, the model $\mathcal{M}$'s output $\mathbf{Y}_t$ we want to reproduce is the immediate cost associated to each state transition, whose aggregation over a pre-selected time horizon gives the objectives of the management problem (Castelletti et al. (2008) and references therein). According to the data-driven nature of the proposed DEMo approach, the emulator is identified on a data-set $\mathcal{F}$ of tuples $\{\mathbf{X}_t, \mathbf{W}_t, \mathbf{u}_t, \mathbf{Y}_t, \mathbf{X}_{t+1}\}$, with $t = 1, \ldots, H$, generated via simulation of model $\mathcal{M}$.

The adoption of a state-space representation brings some important features: i) the emulator is directly usable for optimal management problems, which require the knowledge of the system's state, without subsequently solving a - not always straightforward - minimal realization problem; ii) the resolution of the associated DEMo problem exploits all the information in the data-set $\mathcal{F}$, whereas by using the external form, the state transitions, however generated by the simulation, are totally ignored; iii) the state $\mathbf{x}_t$ can be source for reasonable physical interpretations, as practically demonstrated in the case study sections of the paper; iv) the emulator is generally compact and accurate, since the state $\mathbf{x}_t$ embeds the same amount of information contained in several time-lags of auto-regressive terms and exogenous drivers.

2.2 DEMo procedure

Assuming that a well-calibrated physically-based model $\mathcal{M}$ is available, the identification of a dynamic emulator can be performed in five steps (Castelletti et al., 2011):

Step 1. *Design of computer experiments (DOE) and simulation runs.* Since the DEMo exercise is completely data-driven, the data-set $\mathcal{F}$ must be as much as possible informative thus reproducing all possible model $\mathcal{M}$ dynamic behaviours, forced by the widest spectrum of inputs (exogenous drivers $\mathbf{W}_t$ and controls $\mathbf{u}_t$). Technically, the DOE consists in a sampling in the space of the physically-based model inputs aimed at defining a sequence of simulation runs for model $\mathcal{M}$ with the purpose of generating the data-set $\mathcal{F}$. Considering the severe limitations on the number on runs typically imposed by hydrodynamic-ecological models, proper techniques can be employed to effectively explore this space. These include statistical techniques (e.g. pseudo-random binary sequences (MacWilliams and Sloane, 1976)) or expert-based design (see, e.g., Galelli et al. (2010)).

Step 2. *Variable aggregation.* The spatially-distributed nature of model $\mathcal{M}$ can lead to a large dimensionality of the state and exogenous driver vectors. By processing the data in $\mathcal{F}$ with a suitable aggregation scheme, $\mathbf{X}_t$ and $\mathbf{W}_t$ are transformed in two lower-dimension vectors $\tilde{\mathbf{X}}_t$ and $\tilde{\mathbf{W}}_t$, so that the majority of the variation in the original vectors is captured. The aggregation scheme can rely on fully automatic techniques (e.g. principal component analysis (Jollife, 1986)) or can alternatively be based on expert-based skills (see Galelli et al. (2010)). Eventually, the data-set $\mathcal{F}$ is transformed into the lower-dimension data-set $\tilde{\mathcal{F}}$ of tuples $\{\tilde{\mathbf{X}}_t, \tilde{\mathbf{W}}_t, \mathbf{u}_t, \tilde{\mathbf{X}}_{t+1}, \mathbf{Y}_t\}$.

Step 3. *Variable selection.* Based on the information content of $\tilde{\mathcal{F}}$, model $\mathcal{M}$ is further simplified by selecting the components of $\tilde{\mathbf{X}}_t$ and $\tilde{\mathbf{W}}_t$ that will constitute the emulator's state $\mathbf{x}_t$ and exogenous driver $\mathbf{w}_t$ vectors. Generally, this operation relies on some automated techniques (e.g. variable selection methods; see Castelletti et al. (2011) and references therein), since $\tilde{\mathbf{X}}_t$ and $\tilde{\mathbf{W}}_t$ are often too large to be handled by a human operator.

Step 4. *Structure identification and parameter estimation.* In this step, the functions $\mathbf{f}_t(\cdot)$ and $\mathbf{h}_t(\cdot)$ are built. This is a 'traditional' identification problem, composed of the selection of a suitable model structure and parameter estimation, performed in a data-driven fashion, based on the information content of $\tilde{\mathcal{F}}$.

Step 5. *Evaluation and physical interpretation.* Once the emulator has been calibrated, its ability in reproducing the model $\mathcal{M}$ input-output behaviour is cross-validated on the data-set $\tilde{\mathcal{F}}$. The final emulator is physically interpreted based on the analysis of the arguments of eqs. (1a) and (1b).

Once the emulator has been successfully validated against the data and the operator/expert, it is ready to be employed in the solution of the management problem.

2.3 RVS-IIS algorithm

The core of the proposed approach is the combined Recursive Variable Selection - Iterative Input Selection (RVS-IIS) algorithms (Castelletti et al., 2011), through which the most relevant variables for the emulation of the output $\mathbf{Y}_t$ of the physically-based model $\mathcal{M}$ are selected. The RVS algorithm proceeds iteratively in three steps over each component v^o of $\mathbf{Y}_t$. i) The most relevant variables in explaining v^o are selected (with the IIS algorithm) in the set $\mathcal{V}_i = \{\tilde{\mathbf{X}}, \tilde{\mathbf{W}}, \mathbf{u}\}$ of the candidate input variables on the basis of the information content of the data-set $\tilde{\mathcal{F}}$. This gives the arguments of the component of the output transformation function (eq. (1b)) associated to v^o. ii) For each state variable selected in the previous step, a new run of the IIS algorithm is performed to select the variables relevant to describe its dynamics.

This gives the arguments of the corresponding component of the vector state transition function (eq. (1a)) associated to the state variable considered. iii) If the second step leads to the selection of further state variables not yet included in $\mathbf{x}_t$, the step is recursively repeated until all the selected state variables are given a dynamic description. Once the RVS-IIS algorithms is over, the arguments of eqs. (1a) and (1b) are known.

The Iterative Input Selection (IIS) algorithm called at each invocation of RVS is a model-free, forward-selection method. For a given output variable v^o, the IIS algorithm proceeds by first identifying the best performing input v^* in the set $\mathcal{V}_i$ of candidate variables using an *input ranking procedure* based on a statistical measure of significance. Then, given v^*, it builds an *underlying model* $\hat{m}(\cdot)$ to explain v^o. To account for redundancy, IIS repeats the ranking process using the residuals $\hat{v}^o$ of model $\hat{m}(\cdot)$ as new output variable $\hat{v}^o$. These operations are re-iterated until the accuracy of $\hat{m}(\cdot)$ does not significantly improves any further.

The effectiveness of the RVS-IIS strategy strongly depends on the choice of a suitable combination of the input ranking procedure and the underlying model. The combination adopted must support the ability of RVS-IIS in accounting for significance and redundancy: RVS-IIS must be effective in selecting only the most relevant input variables, while trying to avoid the inclusion of redundant ones, which would unnecessarily add to the emulator complexity. Moreover, the ideal algorithm should account for non-linear dependencies, as hydrodynamic-ecological models are usually characterized by complex, non-linear behaviours with multiple coupled variables. Finally, the algorithm must be computationally efficient, since in a complex and spatially-distributed domain the number of candidate variables is generally considerably large.

Among the different classes of underlying model, we here propose to resort to Extremely Randomized Trees (Extra-Trees), a tree-based regression method proposed by Geurts et al. (2006) that can provide these required features. As a consequence, the choice of the input ranking procedure has fallen on a method based on Extra-Trees: in fact, as proposed in Wehenkel (1998), Extra-Trees can also be used as a ranking procedure, since their particular structure can be exploited to infer the relative importance of the input variables.

3 CASE STUDY: TONO DAM

The case study concerns the optimal operation of Tono Dam, an artificial water reservoir located at the confluence of Kango and Fukuro rivers in Western Japan. The dam, whose construction works are to be completed in early 2012, will form an impounded reservoir with a gross capacity of 12.4 x 10^6 m^3. The main purpose of the dam is to provide water to a downstream agricultural district. However, the water impoundment behind the dam is very likely to suffer from algal blooms due to thermal stratification, and, moreover, the dam operation is expected to alter the natural downstream water temperature pattern. To account for these water quality objectives, the reservoir will be equipped with a Selective Withdrawal System (SWS), which enables water releases at different depth and blending, thus allowing for a mechanical control of the outflow temperature.

To demonstrate the effectiveness of the DEMo approach, this paper illustrates the identification of a dynamic emulator over the sample data set obtained from a 1D physically-based model, which is used to compute the water quality objective related to algal blooms[1]. Given the sequential nature of the decision-making problem underlying the optimal reservoir operation, this objective is formulated as the expected discounted integral of the immediate cost associated to each state transition, and it is thus the output variable considered in the DEMo exercise, i.e.

$$g_t^{wq} = Chla_t^E \tag{2}$$

where $Chla_t^E$ is the average concentration of Chlorophyll-a [μg $Chla$/L] in the time interval $[t-1, t)$ in the euphotic zone, defined as the first three meters below the water surface.

Since the reservoir is created by damming two rivers in a narrow section of their course and will have a relatively small surface area, vertical phenomena are supposedly dominating and the 1D coupled

[1] Water quantity dynamics is described by a simple mass balance equation and does not require an emulator.

hydrodynamic-ecological model DYRESM-CAEDYM originally used in Yajima et al. (2010) is adopted in this study. The model exogenous driver vector $\mathbf{W}_t$ includes 50 components, accounting for the main hydro-meteorological processes and water pollution loads, while the control vector $\mathbf{u}_t$ has two components, the release decisions u_t^{-3} and u_t^{-13} [m^3/s] from the siphons located at -3 and -13 m depth from the reservoir surface. As for the state vector $\mathbf{X}_t$, a total of about 10^2 computational cells for the 10 state variables/cell gives $\sim 10^3$ components for $\mathbf{X}_t$.

4 APPLICATION RESULTS

This section provides a step-by-step description of the results obtained by applying to Tono Dam case study the DEMo procedure implementing the tree-based RVS-IIS algorithms.

4.1 DOE and simulation runs

With the purpose of spanning as much as possible the model state-control space, a set of trajectories for $\mathbf{W}_t$ and $\mathbf{u}_t$ are designed. As for $\mathbf{W}_t$, the time series of observational data over the period 1995-2006 are available, and, considering both the variety of conditions they include and the length of the series, no further data generation is required. Concerning $\mathbf{u}_t$, 100 control scenarios are generated as pseudo-random sequences and the DYRESM-CAEDYM model is run with 1 m vertical grid resolution and a simulation step of 1 min. The simulated data, sampled with a daily time-step, are finally stored in the data-set $\mathcal{F}$ of tuples $\{\mathbf{X}_t, \mathbf{W}_t, \mathbf{u}_t, \mathbf{X}_{t+1}, \mathbf{Y}_t\}$, with dimensionality equal to $\sim 10^3$, 50 and 2. The first half of the $\sim 4.50 \cdot 10^5$ generated tuples is used for the variable selection process, while the second for model calibration and validation.

4.2 Variable aggregation

An expert-based aggregation scheme is employed for the state vector $\mathbf{X}_t$. Among the different layers of the model spatial domain, the five layers located at -3, -7, -13 m of depth and at the bottom and sediments outlets are selected, as they are considered to be representative of the euphotic, middle and benthic zone. For each selected layer, all the state variables are considered, so that the original $\sim 10^3$ components are reduced to 50. The exogenous drivers in $\mathbf{W}_t$ are already lumped in space and do not require any further aggregation. The dimensionality of the first three vectors composing the data-set $\tilde{\mathcal{F}}$ is thus respectively equal to 50, 50 and 2. This gives a total of 102 candidate variables to appear in the emulator.

4.3 Variable selection

The elementary operation in the tree-based RVS-IIS algorithm is the selection of the most relevant variables in explaining each component v^o of the output $\mathbf{Y}_t$, namely the immediate cost g_t^{wq}. The performances of the emulator being built are evaluated in k-fold cross-validation (with $k = 10$) using the coefficient of determination R^2, while the algorithm tolerance is set to 10^{-3}. This means that when the selection of a further variable leads to an increase of R^2 lower than 10^{-3}, the algorithm is stopped.

The variable selection process for the dynamic emulator of the immediate cost g^{wq} takes two calls of the RVS-IIS algorithm to single out a state vector $\mathbf{x}_t^{g^{wq}}$ with 2 components, an exogenous driver vector $\mathbf{w}_t^{g^{wq}}$ with 4 components and the original control vector. The selected variables are presented in Figure 1. The immediate cost depends on i) the average concentration of Chlorophyll-a $Chla^E$ (i.e. g^{wq}) in the first three meters below the surface; ii) the level h, which, as empirically demonstrated in Yajima et al. (2010), might be a limiting factor that contributes in containing algal blooms; iii) the exogenous drivers c (cloud cover), w^s (solar radiation) and $DOPL^k$ (dissolved oxygen in Kango river) and the controls u^{-3} and u^{-13}, which are all limiting factors of the Chlorophyll-a growth. The only variable that does not present an explicit physical meaning is the suspended solid concentration $ssol_4^k$ in Kango river. Numerical results show that the contribution of release decisions is weak and indirect with respect to the output g^{wq} ($\Delta R^2 = 0.0035$ for u^{-3} and $\Delta R^2 = 0.0045$ for u^{-13}), which largely depends on the state variable $Chla^{-3}$ ($\Delta R^2 = 0.9727$). This is probably due to the discrepancy between the reservoir operational time step (1 day) and the slow dynamics of the Chlorophyll-a growth that is in the order of few days: controls are changed at a too high frequency to affect directly the algae dynamics.

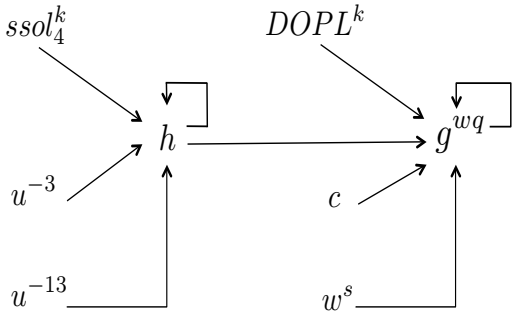

Figure 1. Causal network of the variables involved in the emulation of g^{wq}.

4.4 Structure identification and parameter estimation

This step requires to select an appropriate structure (class of functions) for the emulator, which is then calibrated and validated. Considering the good performances provided by Extra-Trees as underlying model in the variable selection process, they are adopted with the same setting also in this step. The final structure of the emulator is thus a cascade of tree-based models, which is validated with a k-fold cross-validation (with $k = 2$) on the second half of the data-set $\tilde{\mathcal{F}}$.

4.5 Evaluation and physical interpretation

The dynamic emulator performances obtained in k-fold cross-validation are reported in Figure 2, where a comparison of the trajectories for the output g_t^{wq} (as computed by DYRESM-CAEDYM and the emulator) in one-step ahead prediction is shown. The emulator shows good capabilities in approximating g_t^{wq} behaviour, apart from the under-estimation of largest peaks.

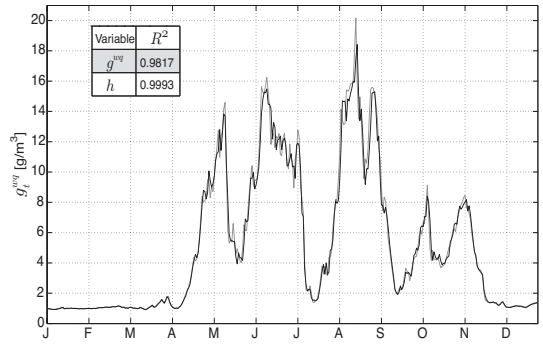

Figure 2. Performance, in 2-fold cross-validation (R^2) of the dynamic emulator output transformation function (1st row) and state transition equation (2nd row), and comparison over the year 1997 between the output g_t^{wq} simulated by DYRESM-CAEDYM (grey line) and predicted by the emulator (black line).

5 CONCLUSIONS

This paper presents a novel approach to the identification of dynamic emulators particularly suitable for all those high resource-demanding problems, such as optimal management, that might benefit from a state-space representation of the emulator both in computational terms and for the increased model

credibility. The core of the proposed approach is the Recursive Variable Selection-Iterative Input Selection (RVS-IIS) algorithm that uses Extremely Randomized Trees as both the underlying model and the ranking procedure in the selection of the more significant input variables to the emulator. The approach is demonstrated on a real world case study implementing a 1D (DYRESM-CAEDYM) coupled hydrodynamic-ecological model of a reservoir: results show that the tree-based RVS-IIS algorithm is particularly effective in reducing the dimensionality of the original models, while accurately explaining the output variables. With a proper variable aggregation scheme and in a relatively small number of the RVS-IIS algorithm iterations, the high-dimensional state and exogenous driver vectors of the original physically-based models are respectively reduced from $\sim 10^3$ (1D model) to 10^1.

ACKNOWLEDGEMENT

The research presented in this work was carried out as part of the SDWA Multi-Objective Multiple-Reservoir Management research programme (R-303-001-005-272).

REFERENCES

Antoulas, A. (2005). An overview of approximation methods for large-scale dynamical systems. *Annual Reviews in Control 29*(2), 181–190.

Broad, D., G. Dandy, and H. Maier (2005). Water distribution system optimization using metamodels. *Journal of Water Resources Planning and Managment 131*(3), 172–180.

Castelletti, A., S. Galelli, M. Ratto, R. Soncini-Sessa, and P. Young (2011). Dynamic Emulation Modelling: a general framework for environmental problems. *Environmental Modelling & Software*. submitted - available from authors.

Castelletti, A., S. Galelli, M. Restelli, and R. Soncini-Sessa (2011). Data-driven dynamic emulation modelling for the optimal management of environmental systems. *Environmental Modelling & Software*. doi:10.1016/j.envsoft.2011.09.003.

Castelletti, A., F. Pianosi, and R. Soncini-Sessa (2008). Water reservoir control under economic, social and environmental constraints. *Automatica 44*(6), 1595–1607.

Castelletti, A., F. Pianosi, R. Soncini-Sessa, and J. Antenucci (2010). A multi-objective response surface approach for improved water quality planning in lakes and reservoirs. *Water Resources Research 46*(W06502). doi:10.1029/2009WR008389.

Crout, N., D. Tarsitano, and A. Wood (2009). Is my model too complex? Evaluating model formulation using model reduction. *Environmental Modelling & Software 24*(1), 1–7.

Galelli, S., C. Gandolfi, R. Soncini-Sessa, and D. Agostani (2010). Building a metamodel of an irrigation district distributed-parameter model. *Agricultural Water Management 97*(2), 187–200.

Geurts, P., D. Ernst, and L. Wehenkel (2006). Extremely randomized trees. *Machine Learning 63*(1), 3–42.

Jollife, I. (1986). *Principal Component Analysis*. New York, NY.: Springer.

MacWilliams, F. and N. Sloane (1976). Pseudo-random sequences and arrays. *Proceedings of the IEEE 64*(12), 1715–1729.

van der Merwe, R., T. Leen, Z. Lu, S. Frolov, and A. Baptista (2007). Fast neural network surrogates for very high dimensional physics-based models in computational oceanography. *Neural Networks 20*(4), 462–478.

Wehenkel, L. (1998). *Automatic learning techniques in power systems*. Boston, MA.: Kluwer Academic.

Yajima, H., A. Castelletti, and R. Soncini-Sessa (2010). Optimal operation of the selective withdrawal system in tono dam reservoir. *Annual Journal of Hydraulic Engineering (JSCE) 54*, –. in Japanese.

Yan, S. and B. Minsker (2006). Optimal groundwater remediation design using an adaptive neural network genetic algorithm. *Water Resources Research 42*(W05407). doi:10.1029/2005WR004303.