



POLITECNICO
MILANO 1863

THE RED RIVER BASIN CASE STUDY

PROJECT REPORT

Gabriele Ferrari (996460)

Tommaso Zaini (970230)

Daniele Sala (996440)

Master of Environmental and Land Planning Engineering

Natural resources management a.a. 2021/2022

Professors:

Andrea Castelletti

Matteo Giuliani

Matteo Sangiorgio

Michele Mauri

Summary

Summary	1
INTRODUCTION	2
PART 1 - DATA-DRIVEN FORECAST MODEL	2
Why building a forecast model?	2
Identification procedure	3
Linear structure	4
ANN	5
Results	6
Other model structures explored (CART and Random Forest)	7
PART 2 – POLICY OPTIMIZATION VIA EMODPS	7
Problem formulation	7
Current performance quantification	8
Direct policy search	9
Pareto front and interesting solutions	10
Non mandatory part	10
POSSIBLE FUTURE IMPROVEMENT	12

INTRODUCTION

The purpose of the project is to explore new and better ways to manage the Hoa Binh dam, an infrastructure affecting a lot of different aspects of Vietnamese economy and society, from hydropower production to flood control and water supply. We explored two ways to do this:

1. Provide a reliable model of the average of the inflow to the Hoa Binh reservoir within the following five days.
2. Understand how the dam is managed and try to compute a new policy that will dominate the older one for two aspects: hydropower production and flood control in Hanoi.

The hydrology of the Red River Basin is peculiar. Half of his catchment area is in China, that does not provide data from the upstream part, due to strategic and political reasons. Furthermore, the energy coming from hydropower production covers about 46% of the entire Vietnamese electrical demand and, in order to supply this amount of energy, a large system of dams and reservoirs has been built all around the Vietnamese territory. Sticking with the economical point of view, the water of this system is exploited by all the agricultural districts of Vietnam, which are vital for the wealth of this country. Every year, during the monsoon season, this large system of water bodies overflows, damaging significantly the cities and the activities established all along the shores of rivers and lakes, sometimes even killing dozens of people. The management task is made more complex by climate change, which increases the occurrence of extreme events, potentially adding further damages to the already fragile Vietnamese society.

As we can see this is a complex multiobjective problem without a single solution, so our project aims to help the decision makers during the decision process, giving them useful information to better manage the Red River system.

PART 1 - DATA-DRIVEN FORECAST MODEL

The first part of the project focuses on the construction of a model forecasting, for each time step (i.e. day), the cumulative inflow of the following five days at the Hoa Binh reservoir. This reservoir is located towards the end of the Da river, about 60 km upstream of the confluence with the Thao river. Two types of hydrological variables are used as input for the model: flows and precipitations. The inflow to Hoa Binh is determined only by the catchment of the Da river, therefore only the flows measured in the stations of MuongTe, TamDuong, NamGiang, LaiChau and Tabu are exploited, as well as the HoaBinh inflow itself (at previous time steps). Instead, precipitation measurements over all the Red River basin are used, since, even though not located in the Da river catchment, they can still provide useful information to the forecast through positive or negative correlation. Thus, six precipitation input variables (MuongTe, TamDuong, Da, BaoLac, BacMe and HaGiang) are added. For the calibration and validation of the model, each input is provided for a length of seven years.

Why building a forecast model?

A model forecasting the cumulative inflow of the following 5 days can have multiple applications. Firstly, it can be used to improve alert systems and early warn local decision makers in order to implement anticipatory actions. Better forecasts of the inflow to the reservoir, and therefore of the level, can improve the flood damage prevention. Secondly, such forecast model can be used to improve the management policy of the regulated lakes in the basin, both for online and offline policies. Particularly, in on-line approach, both in OLFC and in POLFC, forecasts are used to update the optimal control sequence (or policy for POLFC) at each

time step. In OLFC forecasts are used to create a stochastic disturbance, in POLFC forecasts can feed the reduced model, making POLFC more efficient than off-line, but preserving a realistic description of the system. In off-line, with feedback + feedforward control, the inflow prediction can be added to improve the policy performance, changing the control based not only on the state (the storage), but also on the streamflow prediction. The application of the forecast model therefore depends on the approach adopted by the regulator of the Hoa Binh dam.

Identification procedure

Implicitly assuming that the physical system is cyclo-stationary, we perform a data-driven model identification, with different types of models. Whatever the tested model, the first step is to de-seasonalize the data according to the cyclo-stationary mean and standard deviation (calculated with window size equal to 10). In the context of this project, the performance metric used to assess the quality of a model is the R^2 , but further analysis can be conducted using other metrics. The aim of our identification procedure is to find a model that on one side has high performance, and on the other side is simple and computationally cheap.

Therefore, the exploration of the model structure will start with linear models and then will be complicated using non-linear parametric and non-parametric structures. The linear model has several advantages, among which its fast calibration with linear least squares, the simple model identification and the easier a posteriori interpretation, since the role of each parameter or variable is clearly identifiable. However, a natural process such as the streamflow generation in the Da river basin is likely nonlinear. For this reason, despite being more complex, hard to interpret, and possibly computationally intensive, non-linear models will likely allow for a more realistic representation of the phenomenon. Another major goal of our work is to create a model that produces accurate predictions with different datasets, in other words avoiding the risk of overfitting. For this purpose, the cross-validation plays a key role. Indeed, although the system is (cyclo-)stationary, there are variabilities due to stochasticity or occasional extreme events that can significantly affect the results. Using a single data split between calibration and validation would make the analysis prone to the risk of overfitting the model to those two specific datasets. To avoid this, we split the dataset using the K-Fold cross validation process with 5 years for training and 2 for validation, with the two years always contiguous at each iteration.

	Years						
	y1	y2	y3	y4	y5	y6	y7
K=1							
K=2							
K=3							
K=4							
K=5							
K=6							
K=7							

= calibration years
 = validation years

Table 1.1 Partitioning of the dataset for the k-fold cross-validation

Linear structure

The first linear model we build is a linear model fully exogenous, that exploits all the input variables to compute the 5-days cumulative inflow. The main challenges for this section are to understand:

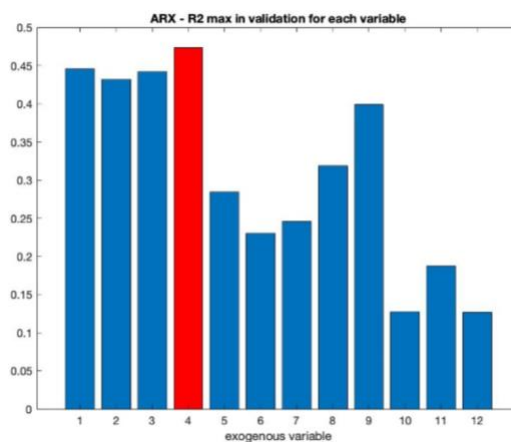
- In which order the variables should be added to the model
- Which variables should be included (feature selection)
- What time lag for each variable

To answer the first question, we do not carry out an a-priori analysis, such as a correlation analysis, but rather we directly test the ARX model: iteratively we add one by one each input variable and select at each step the one providing the maximum information, namely giving the maximum R^2 increase. In our procedure, the 3 questions mentioned before are solved at the same time using nested for-cycles, as explained in the following simplified pseudo-algorithm:

```

for i = 1 : number_of_inputs %go through every column (i.e. variable) of the dataset
    for j = 1 : max_lag %for each variable evaluate different orders of lag
        for k = 1 : number_of_partitionings_kfold %for each data-split of the cross-validation
            calibrate model (calculate parameters) with input i, lag j and training data k;
            validate model (calculate R2val(k)) with input i, lag j and test data k;
        end
        Compute average of R2val over the k datasets;
        Save the R2val, the variable i and its lag j, if it is the highest R2val (so far);
    end
end
end

```



	Input variable											
	1	2	3	4	5	6	7	8	9	10	11	12
R2cal	0.449	0.431	0.441	0.467	0.294	0.241	0.242	0.316	0.410	0.150	0.193	0.133
R2val	0.446	0.432	0.442	0.473	0.284	0.231	0.245	0.319	0.399	0.128	0.187	0.126
lag	2	2	2	1	1	1	1	1	1	1	1	1

Figure 1.1 and Table 1.2 - R^2 in validation adding one variable at a time. NamGiang flow (variable 4) with lag 1 is the one that gives maximum contribution at the first iteration

Once the first variable is found (in this case no. 4, the flow at NamGiang – Fig 1.1), the iterative procedure continues and the variables are added one by one, according to the marginal increment of R^2 they provide in validation (Figures 1.2 & 1.3). We will stop adding input variables when the R^2 in validation (averaged over the k datasets) stops growing, in other words when the “optimal” level of complexity is attained, just before the model starts to be overparametrized (Fig 1.4). At the end of the linear model identification, we select 8 input variables with different lags, for a total of 24 parameters and a final R^2 in validation of 0.591 (Table 1.3).

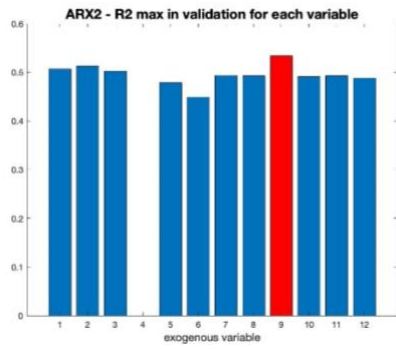


Figure 1.2 - R^2 in validation with 2 variables. The 2nd added input is the precipitation in Da.

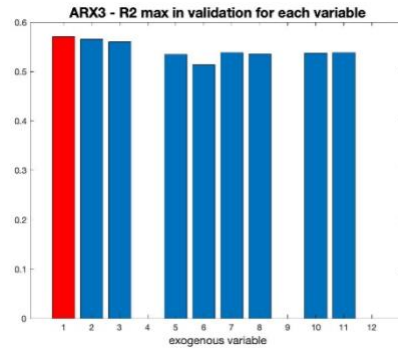


Figure 1.3 - R^2 in validation with 3 variables. The 3rd added input is the HoaBinh inflow.

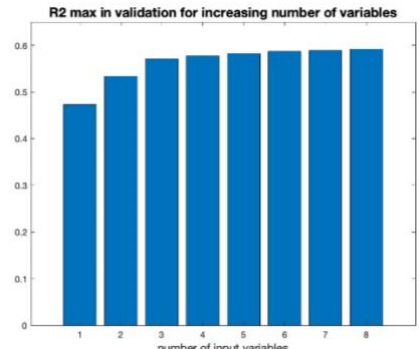


Figure 1.4 - R^2 in validation for increasing number of input variables

Variable added	4	9	1	10	3	8	7	11
Lag	1	4	1	5	2	7	3	1
R2val	0.473	0.533	0.570	0.577	0.582	0.587	0.589	0.591

Table 1.1 Progressive addition of variables and corresponding increase of R^2 in validation

ANN

Among the nonlinear parametric models, we have focused on Artificial Neural Networks, due to their flexibility in reproducing almost any kind of system. Indeed, according to the universal approximator principle, we can use feed forward ANN to approximate every function with given accuracy using a number of neurons growing less than exponentially with the size of the inputs. As previously mentioned, we aim at building a neural network that can be a good compromise between performance and complexity. In this case, the model identification can be divided into two macro-sequences. The meta-parameters identification and the optimal lag selection for each input.

The first meta-parameter is the number of hidden layers, which we choose a priori to set to 1 in order to keep the network light. To find the second meta-parameter, the number of neurons, we perform a cross-validation and for each of the k datasets we add one by one a neuron at a time, we train the NN, and validate it (calculate the R^2 with validation dataset). Afterwards we compute the mean of the R^2 in validation through the k datasets and find the number of neurons that gives the maximum R^2 (Figure 1.5).

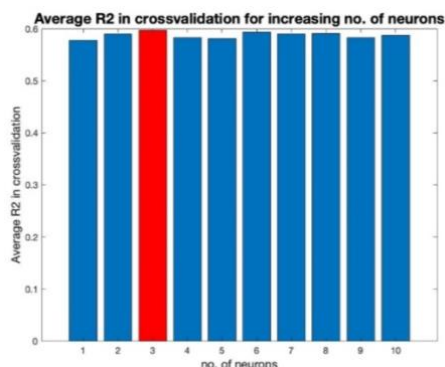


Figure 1.5 - Average R^2 in cross-validation for increasing number of neurons. 3 neurons is the optimal.

Secondly, for the ANN we choose to feed all the 12 input variables, since we don't want to neglect a priori the information given also by the eastern part of the Red River basin. However, every input has a different

lag. To find the optimal lag for each variable, we add an additional day lag at a time until the R^2 stops growing. We finally get an ANN with 3 neurons and with the following lags for each input variable:

input variable	1	2	3	4	5	6	7	8	9	10	11	12
optimal lag	3	5	2	3	2	1	4	7	7	4	5	2

Table 1.2 - Optimal lag for each input variable

The total number of inputs is thus 45 and the number of parameters is 142.

Results

The final R^2 is around 0.67 in calibration and 0.63 in validation, therefore the ANN is chosen as the best performing model among the tested ones.

The first feature to remark about the built ANN model is that it performs similarly in calibration and validation (Figures 1.6 & 1.7), which suggests that it does not overfit the data and it is likely not overparametrized. In general, the model provides more accurate predictions during the dry season (winter and spring), when the inflow has very little variability, while it shows some delay and more discrepancy from observed inflows in the wet season (summer and fall), where the inflow to Hoa Binh is much more variable, due to the increased precipitations. The delay is a structural feature of the model, which uses as input hydrological variables in the previous time steps; thus, a rapid change in precipitation and streamflow (in the order of hours) will be transformed with a delay into the predicted inflow by the model (which has a lead time of 5 days). The fact that the model predicts less accurately peaks is related to the fact that we are using one single forecast model. Using and calibrating different models for different values of inflows (for example one for the dry season and one for the wet season, selected according to a threshold value of flow) could partially overcome this inaccuracy.

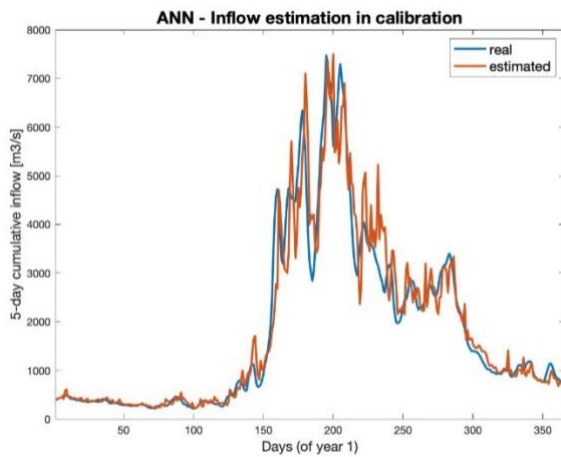


Figure 1.6 - Observed vs predicted Hoa Binh inflow for year 1, in calibration

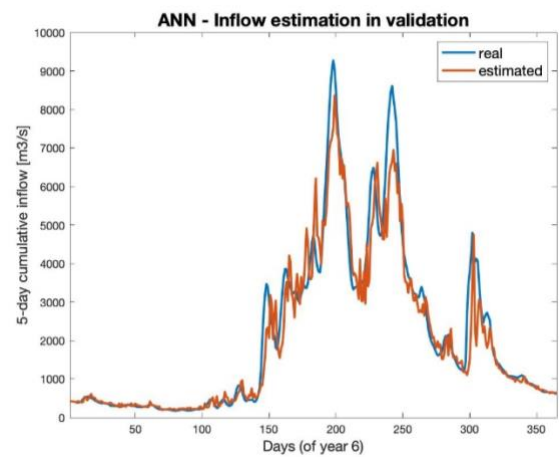


Figure 1.7 - Observed vs predicted Hoa Binh inflow for year 6, in validation

As far as the prediction error is concerned, it has zero mean and a standard deviation of 0.61 (de-seasonalized). The distribution is close to a gaussian one, even though it is slightly asymmetrical (left-tailed with skewness < 0) (Fig 1.8): this means that the forecast model tends to overestimate more frequently than to underestimate the inflow. Moreover, the error shows a temporal autocorrelation with the previous three time steps, meaning that it is not a white noise. This suggests that the forecast model has margin of improvement. The autocorrelation could be partially reduced adopting a recurrent NN instead of a

feedforward NN, so that the error at previous time steps can be included through recursion (which is impossible with a feedforward architecture).

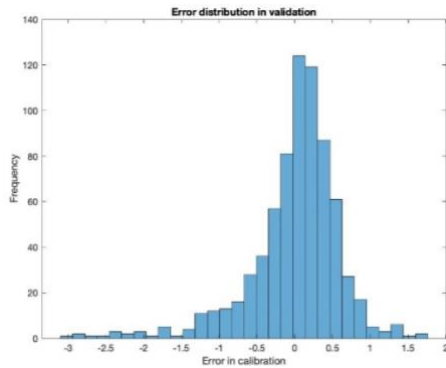


Figure 1.8 - Statistical distribution of the error in validation

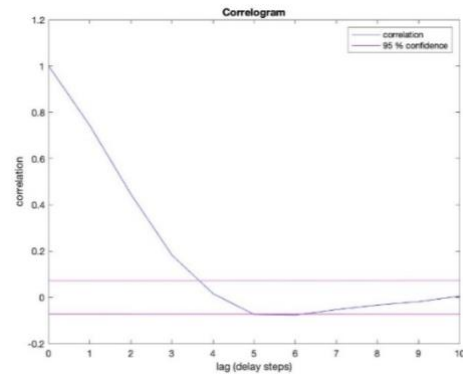


Figure 1.9 - Autocorrelation of the error

Other model structures explored (CART and Random Forest)

Tree-based models are a non-parametric approach based on analogy to make output predictions. We build a CART using a function that automatically optimizes the hyperparameters like the splitting points and the termination criteria and then operates the recursive binary splitting. Despite an R^2 of 0.72 in calibration, which is the best of all the tentative models, the R^2 in validation is around 0.48, the lowest, showing an overfitting of the CART on the training dataset. Another way to exploit tree models is via Random Forests that, given some meta-parameters, randomly select the splitting variable and build an ensemble of trees. In this case the final R^2 in validation is 0.57.

These models are outperformed by linear ones and by ANN, so we decided to leave them apart and concentrate on the other structures.

PART 2 – POLICY OPTIMIZATION VIA EMODPS

The second part of the project aims at optimally allocating water resources in the Red River basin. Specifically, our work focused on optimizing the control policy of the Hoa Binh reservoir, considering two main objectives: the maximization of the hydropower production and the minimization of the flood events in the downstream city of Hanoi, the Vietnam's capital city. The first part consists of assessing the performance of the current policy, the second part addresses the actual optimization via Evolutionary Multi-Objective Direct Policy Search (EMODPS).

Problem formulation

The red river basin extends mostly in China and Vietnam, with a little part in Laos. However, China doesn't share information with neighboring states, making the task of dam managers in Vietnam more difficult. In this report we analyze the management problem of the Hoa Binh dam, which has direct influence on flooding in Hanoi, the capital city with 16 million inhabitants. For this reason, we try to find a policy that combines minimization of flooding in this city and maximization of hydropower production. The

schematization of the system is reported in *Figure 2.1*: the Da River feeds the reservoir, while downstream it joins two tributary rivers, Lo and Thao, before reaching the city of Hanoi.



Figure 2.1 - Schematic representation of the system in our policy optimization problem

The hydropower objective is the mean daily production (in KWh/d), which is a function of the outflow of the dam. The flooding objective is the mean daily squared water level excess in Hanoi (in cm^2) and is a function of the water level in the city.

Current performance quantification

The current policy, which will be used as baseline for the following optimization, is a Standard Operating Policy, meaning that the release (i.e. the decision variable) is a piecewise linear function of the level of the lake (representing the state). The policy must be included between a minimum release and a maximum release. The minimum release is necessary to guarantee the minimum environmental flow of the river, thus preserving the downstream environment and activities. The maximum release, instead, is the water release at open spillways, in case of flood events.

The policy shape is determined by the value of 5 parameters: h_1 , h_2 , m_1 , m_2 and w :

- h_1 : minimum height in order to start releasing water;
- h_2 : maximum height after which we activate the spillways;
- w : constant water demand;
- m_1 and m_2 : slope of the lines that connect zero-release and the maximum release with w ;

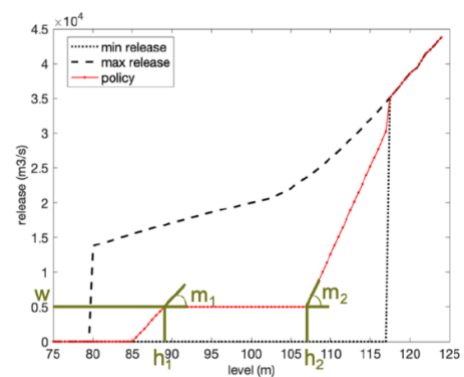


Figure 2.2 - Structure of the piecewise control law

In order to quantify the performance of the current policy, a simulation of the reservoir operation is carried out over the horizon 1995-2005 (excluding the first two months for warmup). Thus, a simulation function is used, which calculates at each time step, according to the release decision, the storage, the water level in Hoa Binh, the level in Hanoi and the consequent values of the two objectives. As a result, over this 10-year

horizon, the current policy produced objective values equal to $1.69 \cdot 10^7$ kWh/d for hydropower production and $569,6 \text{ cm}^2$ for (squared) water level excess in Hanoi. *Figure 2.3* shows the streamflow in Da, the level of the reservoir, the release and the level in Hanoi in period 1995-2005 with the current SOP.

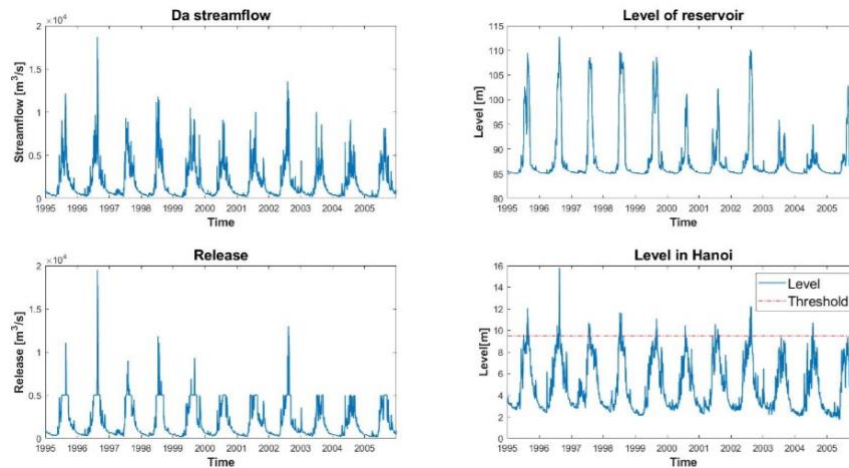


Figure 2.3 - Output trajectories with the current policy

Direct policy search

To optimize the policy, we use a direct policy search with NSGA II, that is an evolutive multi-objective algorithm able to explore parameters improving the performance of the policy. *Figure 2.4* shows how parameters evolve from the initial population to the final generation.

The algorithm creates a random initial population, computes the performance of each policy and mates individuals two by two. The selection is operated via tournament selection, that consists in pairwise comparisons between two random individuals, after which only the one with highest fitness survives. The fitness is assessed with a linear combination of the value of the two objective functions. As a result, the algorithm progressively finds better performing solutions and the final population is concentrated in the bottom-left part of the plot. In our case we use an initial population of 70 individual and 50 generations. SOP is added in this plot to quantify how much performances are improved.

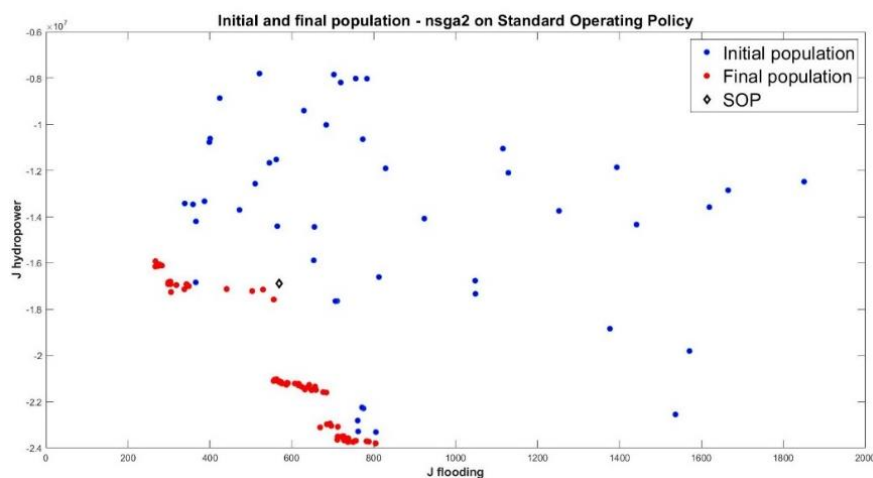


Figure 2.4 - Initial and final populations (i.e. parametrizations) using NSGAII on Standard Operating Policy

Pareto front and interesting solutions

Pareto fronts are built by selecting only individuals of initial and final populations that are non-dominated. From the final pareto front we also selected three interesting solutions: Best Hydropower, with the maximum hydropower production; Best Floods, with the minimum water level excess in Hanoi and Best Compromise, that is the closest point to the utopia (Fig 2.5).

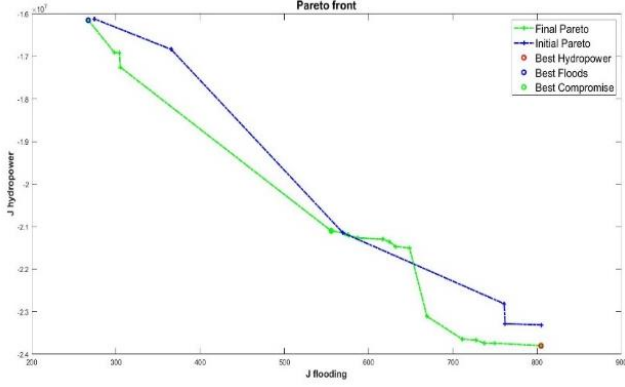


Figure 2.5 - Evolution of the Pareto front through optimization on SOP

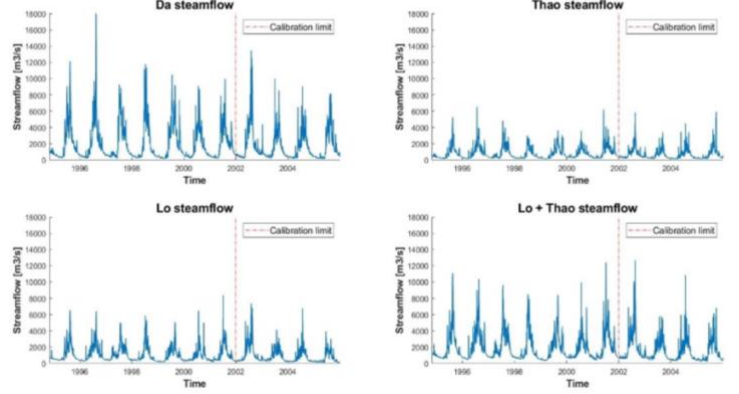


Figure 2.6 - Input variables of the policy

Non mandatory part

By watching plots of input variables (Fig 2.6), we noticed a great variability between the peaks of each year. Therefore, to avoid the risk of overfitting, we split the time series in two parts, one for calibration and another for validation.

In order to deepen the search for an optimal (or suboptimal) policy, we also decide to extend our analysis to Gaussian radial basis functions. We choose this family of functions because they are flexible and capable of representing policies for a large class of management problems and they allow to include exogenous information, thus working with higher dimensionality.

$$y_{t+1} = \sum_{i=1}^n b_i \phi_i(\mathbf{z}_t)$$

$$\phi_i(\mathbf{z}_t) = \exp \left[- \sum_{j=1}^{N_z} \frac{(z_t^j - c_i^j)^2}{(b_i^j)^2} \right]$$

The release function therefore becomes a nonlinear one. To design the control policy we decide to use a sum of four radial functions with 13 parameters: 4 means, 4 standard deviations, 4 weights and a scale factor. This last parameter converts the value of the function, which is a number from 0 to 1, to a value of release that is in the order of magnitude of 10^4 .

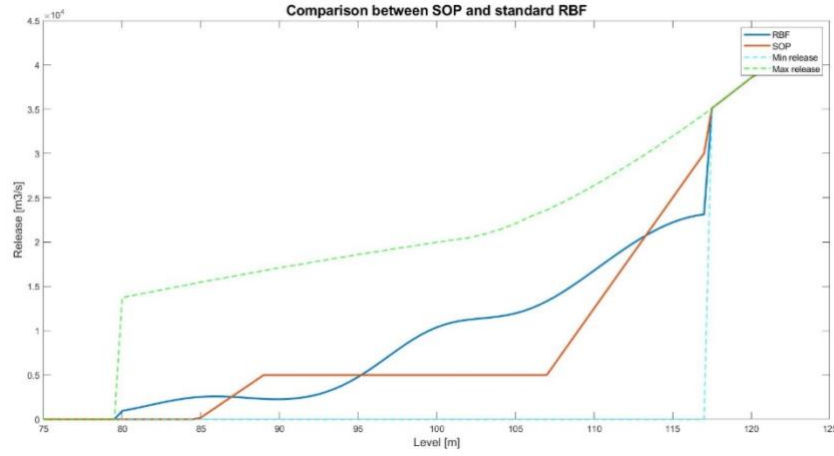


Figure 2.7 - RBF policy and Standard Operating policy before the optimization

Figure 2.7 shows how the standard radial basis function appears in comparison with the standard operating policy. It turns out to be a smoother relation between level and release, allowing for better performance through optimization.

We optimize both the policies over calibration period by using NSGAI1 with 70 individuals and 50 generations. Then, we evaluate the performance of the two final populations over validation period and look for the non-dominated policies of each population.

Figure 2.8 shows the final results with SOP and RBF, compared with pre-optimization standard operating policy and radial basis function. RBF policy clearly outperforms the piecewise linear function. This is due to the smoother shape of the RBF function and therefore its higher adaptability to the specific problem. However, it is worth remarking that this is the result over the four years of validation. Having longer time series would provide more robust solutions.

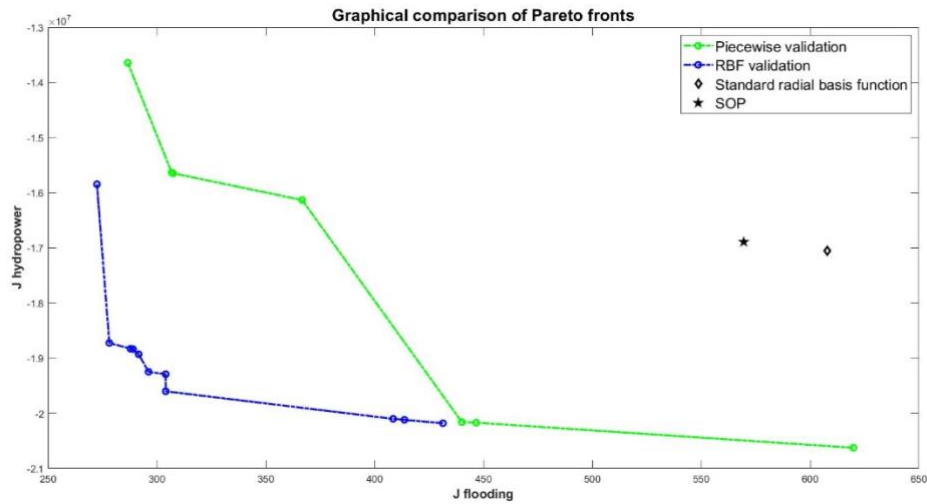


Figure 2.8 - RBF policy and SOP after the optimization with NSGAI1

POSSIBLE FUTURE IMPROVEMENT

For both the forecast model construction and the policy design the work can be further developed, further analysis can be carried out in order to obtain more accurate predictions or provide a better performing and more robust policy.

Regarding the model of the 5-days cumulative inflow other more complicated structures can be tested, like weakly non-linear models (i.e. piecewise linear), different neural network architectures or other types of tree-based structures.

For the second part the value of exogenous information can be assessed to see if adding other information leads to a further improvement of the performances of the policy. For example, we can build a model to forecast the inflows coming from Thao and Lo River instead of using their historical series, or use the model obtained in the first part to build a multidimensional release function. To do so, radial basis functions are very useful because the Gaussian bases allows multivariate nonlinear transformation of different inputs.