AGU PUBLICATIONS

Water Resources Research

Supporting Information for

The impact of assuming perfect foresight for investment analysis in water resources systems

R. Payet-Burin[1,2], M. Kromman[2], S. J. Pereira-Cardenal[2], K. M. Strzepek[3], and P. Bauer-Gottwein[1]

1 Department of Environmental Engineering, Technical University of Denmark, Kgs. Lyngby, 2800, Denmark.

2 COWI A/S, Kgs. Lyngby, 2800, Denmark.

3 Joint Program on the Science and Policy of Global Change, MIT, Cambridge, MA 02139, USA

Contents of this file

Introduction

This supporting information provides (1) details on how nearest neighbors bootstrapping is implemented in the model, (2) the reservoir storage values in the Nile case from Stochastic Dynamic Programming, and (3) additional results to the Nile study case. The model and data for the different study cases is available at https://github.com/RaphaelPB/WHAT-IF

Text S1. Forecasting using Nearest Neighbors bootstrapping

Nearest neighbors bootstrapping (Lall & Sharma, 1996) is a method to generate synthetic time series or forecasts using observed time series while preserving important correlations and autocorrelations (Yates et al., 2003). The concept is to find a system state in the historical archive, which is similar to the current state of the system and then assume that future evolution will be similar to the evolution observed in the past. The approach can be divided in three steps: (1) Define the feature vector characterizing the current state of the system, (2) Find the nearest neighbors (closest past system states) in the observed time series, (3) Sample from the nearest neighbors to generate a forecast or a synthetic time series.

At a given time step (i), we consider here as feature vector (Di), the ensemble of: total runoff (Q_i), average precipitation (P_i), and average reference evapotranspiration (E_i) in the system.

$$D_i = [Q_i, P_i, E_i]$$

The ensemble of feature vectors of past occurrences (D) is defined for the ensemble of time steps corresponding to the same month as the current time step ($T_{month(i)}$).

$$D = ([Q_t, P_t, E_t] \,|t \in T_{month(i)})$$

$$D = ([Q_t, P_t, E_t] \;\forall\, t \in T \,|month(t) = month(i))$$

$$D = ([Q_t, P_t, E_t] \;\forall\, t \in T_{month(i)})$$

The distance (r_t) of a past state (D_t) to the current system state (D_i) is calculated using the Euclidian norm between feature vectors:

$$r_t = \sqrt{\sum_j w_j \cdot \left(D_i^j - D_t^j\right)^2}$$

where w_j are the weights of the elements of the feature vector (runoff, precipitation, and evapotranspiration here). We define the weights as the inverse of the standard deviation of the elements of the feature vector. For example, the runoff weight takes the form:

$$w_Q = {}^{1}\!/_{std(Q_t \,|\, t \in T_{month(i)})}$$

The k nearest neighbors are the k past system states with the lowest distance to the current state. The choice of k can be optimized, Lall and Sharma (1996) suggest the square root of the total amount of samples. Because we use 40 years or 360 months length time series, we choose k=20. The nearest neighbors are ranked from lowest to highest Euclidian distance. To sample among the nearest neighbors, we define the sampling Kernel based on the rank of the neighbors, as in Yates et al. (2003).

$$K_l = {}^{1/l}\!/_{\sum_{n=1..k} 1/n}$$

where l is the rank of the neighbors. This approach assumes that only the rank affects the probability; Akbari et al. (2011) describe alternative sampling Kernels.

To generate an ensemble forecast, the desired number of neighbors are sampled with probability K. To generate an average forecast, the nearest neighbors are weighted according to their probability to generate a single weighted forecast ($\widehat{D}_{i+m}$):

$$\widehat{D}_{i+m} = \sum_{l=1..k} K_l \cdot D_{t(l)+m}$$

Where m is the forecast lead time (in time steps) and $t(l)$ is the time step corresponding to the l-th ranked neighbor. We also generate weighted ensemble forecasts, by classifying the k-nearest neighbors in different categories based on the total predicted runoff (e.g. 50% lowest and highest predictions) and then computing the weighted average forecast within the categories. The likelihood of the categories is then the sum of the neighbor's likelihoods belonging to this category. This last method enables to generate an ensemble forecast with less members that still contains information from the k-nearest neighbors.

Text S2 Nile study case: storage value with Stochastic Dynamic Programing (SDP)

Loucks and Van Beek (2005) describe the development of the SDP method. The code implementing the SDP framework is https://github.com/RaphaelPB/WHAT-IF under the "Nile synthetic case" branch in the file "Nile_SDP_water_value.py". The SDP framework is not part of the WHAT-IF tool and was only implemented on this specific study case to compare results with the Model Predictive Control framework. **Figure S1** shows example of the reservoir storage value for three scenarios obtained from backwards runs in the SDP framework. When comparing the full range of scenarios exploring total runoff and water demand, the reservoir storage value is calculated individually for each scenario.

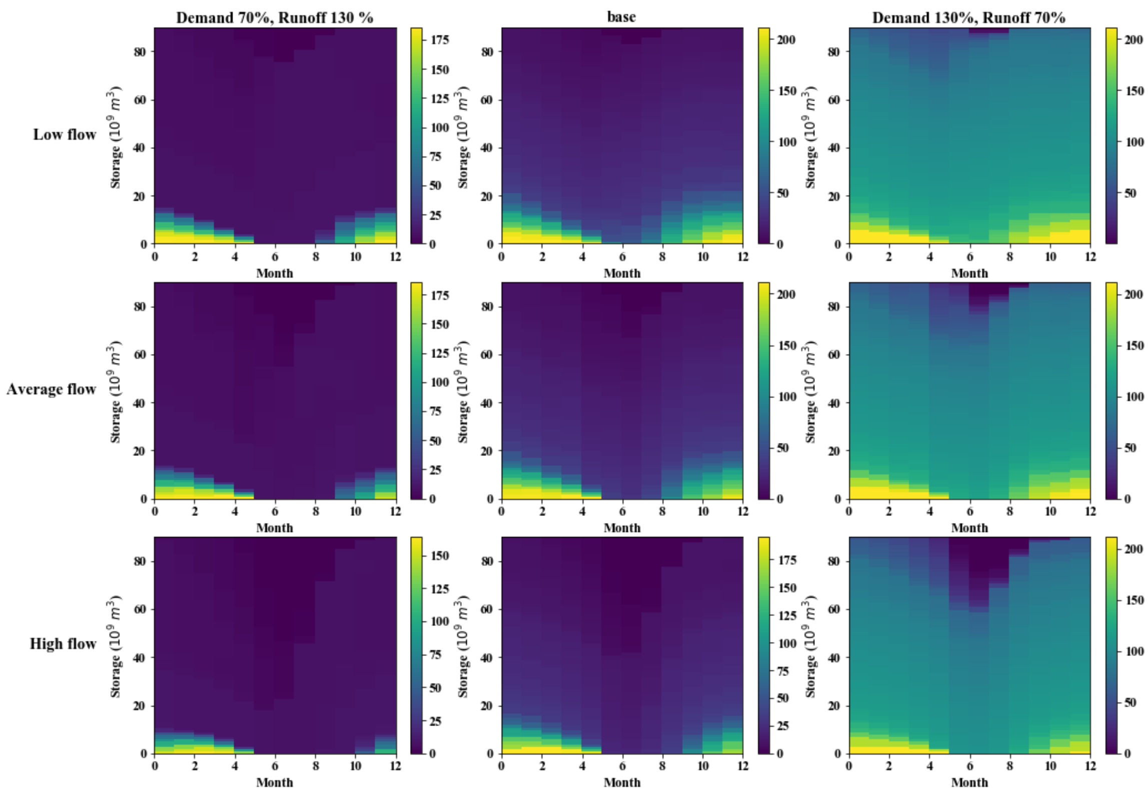

Figure S1. Reservoir storage value evaluated through the SDP backward runs. The value of reservoir storage depends on the current inflow and is different for each scenario.

Text S3 Nile study case: additional results to the projects and scenario evaluation

This section provides two additional figures to the paper: **Figure S2** shows the differences between the frameworks on key indicators not restricted to total system benefits (Allocation value, Hydropower value, Spills and Storage), **Figure S3** shows the impact on total system benefits and cost-benefit analysis when evaluating the project development as in the Paper, but displays the results in terms of relative value (percentage).

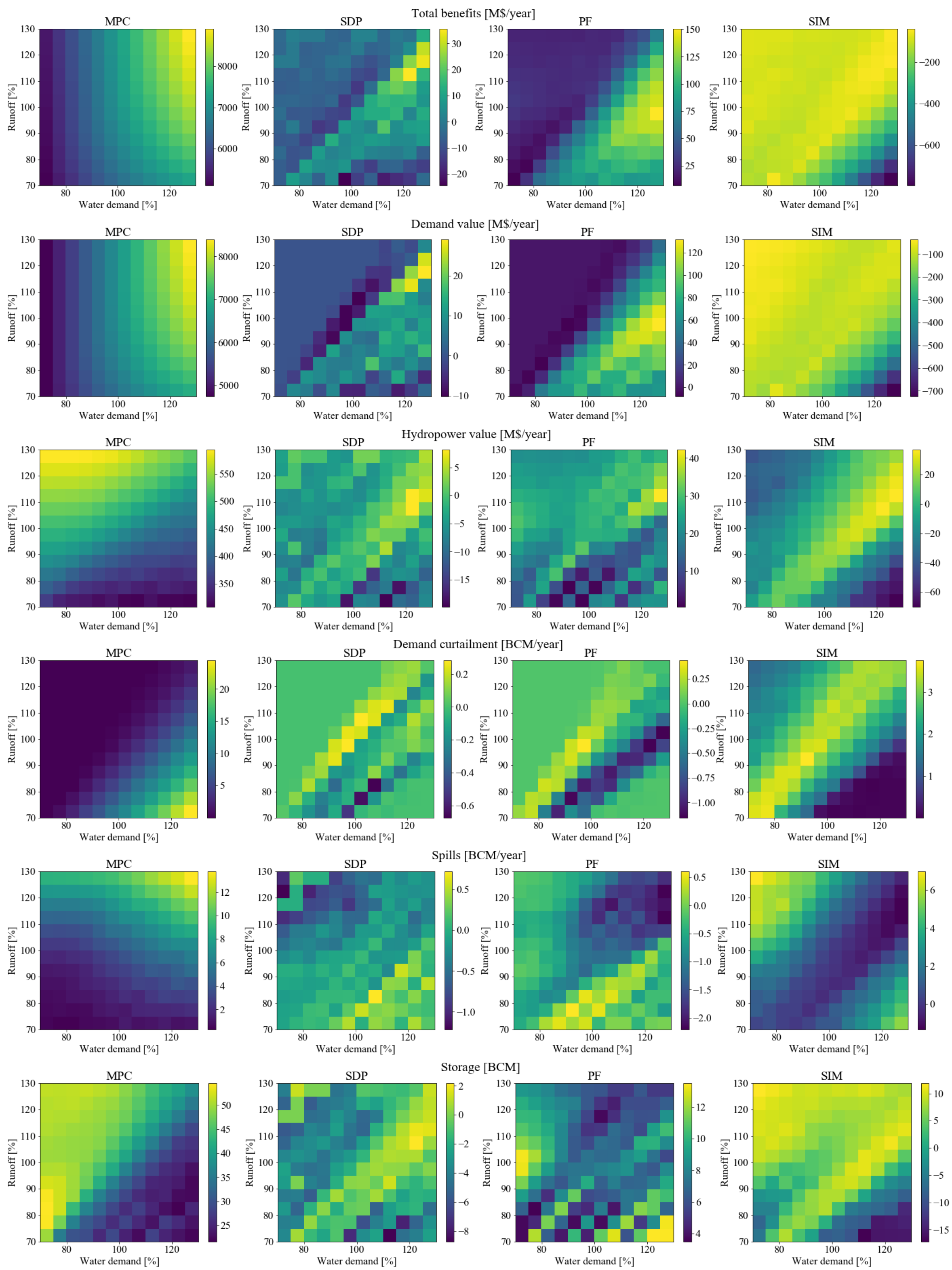

Figure S2. Key indicators on the Nile study case. MPC in absolute values, other frameworks are relative to MPC

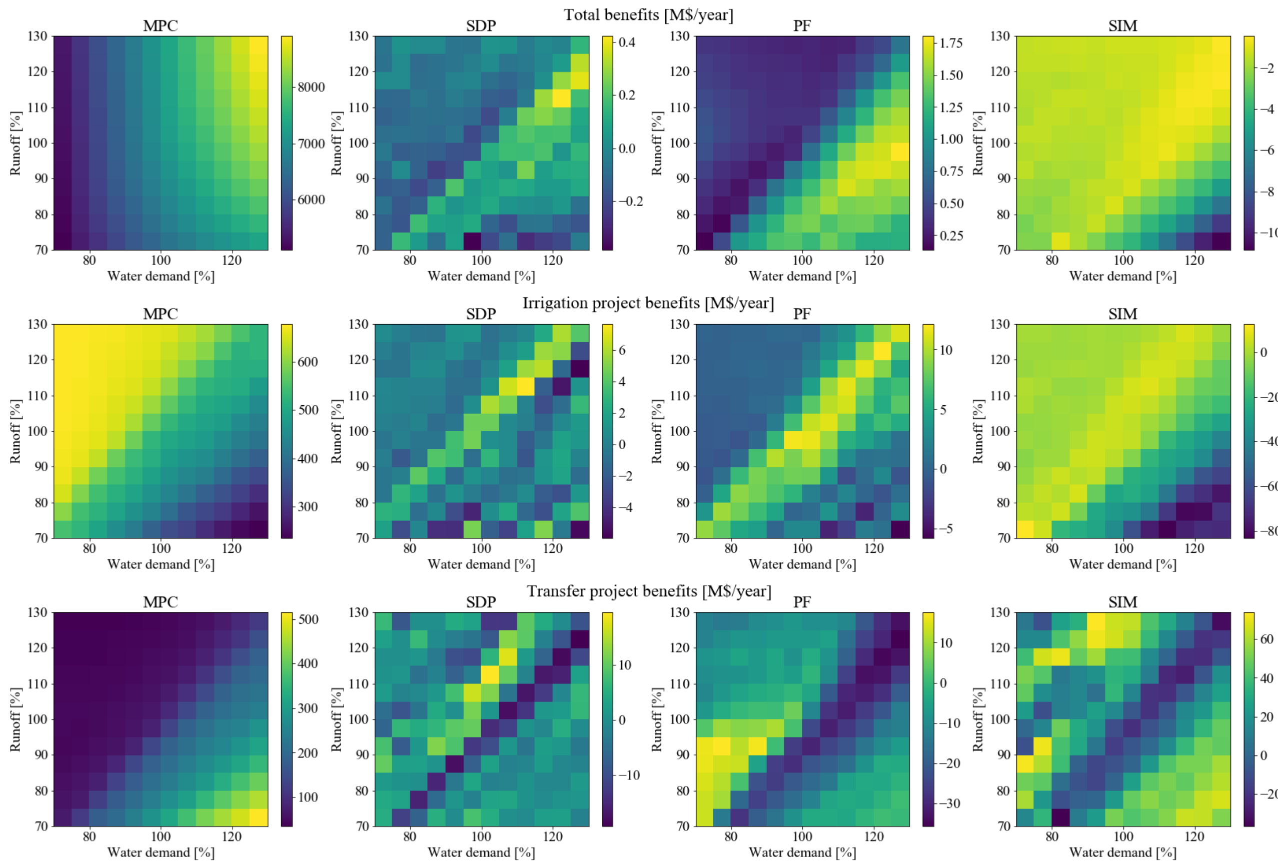

Figure S3. Economic evaluation of project development with the different frameworks. MPC in absolute values, other frameworks are in relative changes (%): (framework-MPC)/MPC

References

Akbari, M., van Overloop, P. J., & Afshar, A. (2011). Clustered K Nearest Neighbor Algorithm for Daily Inflow Forecasting. *Water Resources Management*, *25*(5), 1341–1357. https://doi.org/10.1007/s11269-010-9748-z

Lall, U., & Sharma, A. (1996). A nearest neighbor bootstrap for time series resampling. *Water Resources Research*, *32*(3), 679–693. Retrieved from http://www.agu.org/journals/wr/wr9603/95WR02966/0.html

Loucks, D. P., & Van Beek, E. (2005). Water Resources Systems Planning and Management and Applications: An Introduction to Methods, Models and Applications. *Water Resources Planning and Management*. Paris, France: UNESCO and WL | Delft Hydraulics. Retrieved from https://unesdoc.unesco.org/ark:/48223/pf0000143430

Yates, D., Gangopadhyay, S., Rajagopalan, B., & Strzepek, K. (2003). A technique for generating regional climate scenarios using a nearest-neighbor algorithm. *Water Resources Research*, *39*(7), 1–15. https://doi.org/10.1029/2002WR001769