Quadratic approximate dynamic programming for scheduling water resources: a case study

Agustin Castellano
Universidad de la República
Montevideo, Uruguay
acastellano@fing.edu.uy

Camila Martínez
Universidad ORT Uruguay
Montevideo, Uruguay
martinez_c@ort.edu.uy

Pablo Monzón
Universidad de la República
Montevideo, Uruguay
monzon@fing.edu.uy

Juan Bazerque
Universidad de la República
Montevideo, Uruguay
jbazerque@fing.edu.uy

Andrés Ferragut
Universidad ORT Uruguay
Montevideo, Uruguay
ferragut@ort.edu.uy

Fernando Paganini
Universidad ORT Uruguay
Montevideo, Uruguay
paganini@ort.edu.uy

***Abstract*—We address the problem of scheduling water resources in a power system via approximate dynamic programming. To this goal, we consider the use of quadratic approximate value functions for the finite horizon economic dispatch problem with convex stage cost and affine dynamics. Evaluating the achieved policy supposes solving a quadratic program at each time step, while value function fitting can be cast as a semidefinite program. We test our proposed algorithm on a simplified version of the Uruguayan power system, and obtain simulations that show promising performance.**

***Index Terms*—Approximate dynamic programming, economic dispatch, convex optimization, power systems.**

I. Introduction

Optimal operation of multi-reservoir systems for economic dispatch is a topic that has been extensively studied [1]–[3]. Succintly, the goal is to obtain a sequence of release decisions that achieve system operation with minimal cost over a planned horizon, while also meeting operational constraints. In systems involving large reservoirs decisions become coupled across time, while also being dependant on the availability of water —which is typically stochastic. The usual framework for solving these kinds of problems is (Stochastic) Dynamic Programming, where the state of the system typically includes the storage level in each reservoir. Standard practice involves discretizing the state variable and computing the value function at each point. However, the number of needed evaluations grows exponentially with the number of states, a phenomenon known as the Curse of Dimensionality [4]. In order to circumvent this issue, several (approximate) techniques have risen which allow for the problem to be solved in continuous spaces. One of such celebrated algorithms is SDDP which seeks to approximate the value function by a set of lower bounding affine functions [5], [6]. However, getting a rich enough approximation might entail the use of too many hyperplanes [cite needed]. Moreover, under quadratic stage cost and affine dynamics the resulting value functions are provably convex quadratic [7]. Given this, we sought to explore an alternative simpler parametric model. Specifically, we aim to tackle this problem by approximating each value function with a suitable convex quadratic function. This simplified model allows us to formulate the scheduling problem as a special case of convex approximate dynamic programming, therefore making the problem tractable on a continuous state manifold while also relaxing the need of computing exact averages, something typical of SDDP [cite needed].

There has been many a work regarding approximate convex dynamic programming. For a certain class of scalar storage problems, the value functions can be proven to be convex piecewise linear, and algorithms with proven convergence guarantees have been developed [8], [9]. Quadratic approximate dynamic programming has been used before (see e.g. [**?**]), especially for systems with quadratic cost and transition dynamics that are affine in the control (see [10] for examples including trajectory tracking and portfolio optimization). We build on these contributions for modelling the water scheduling problem as a quadratic approximate dynamic program.

The paper is outlined as follows. Section II briefly reviews dynamic programming and extends it so as to include inflow evolutions. Section III presents the proposed algorithm, which involves sequentially solving several quadratic programs [11, p.152] and one semidefinite program [12]. In Section IV we present our numerical results applied on the Uruguayan power system, while also detailing how to incorporate hydrologic uncertainty in our model. Conclusions are given in Section V.

II. Dynamic Programming

Consider a simple model of operation of a hydroelectric system over a horizon K, with time indexed as $k = 0, 1, \ldots, K-1$. A state vector $x_k \in \mathbb{R}^n$ represents current storage level at n reservoirs; a control vector $u_k \in \mathbb{R}^m$ models the actions the system operator may take, including the release and spill term on each hydroelectric plant and the economic dispatch decisions on other types of generators; water inflows $w_k \in \mathbb{R}^p$ at the reservoirs are modeled as correlated noise. Notice that we don't enforce $p = n$ since there might not be recorded inflows

at some of the reservoirs. The cost of operation of the system is modeled through a function $g_k(x_k, u_k, w_k)$, which may include the cost of thermal generation and a penalty for deviating from economic dispatch. Our goal is to obtain a sequence of control actions $\mathbf{u} = \{u_0, \ldots, u_{K-1}\}$ such that, for a given starting state x, the expected cost of running the system is minimized:

$$\mathbb{E}\left[\min_{\mathbf{u}} \sum_{k=0}^{K-1} g_k(x_k, u_k, w_k) \mid x_0 = x\right] \tag{1}$$
$$x_{k+1} = f_k(x_k, u_k, w_k)$$

Notice that this formulation —which first computes a minimum and then an expected value— differs from typical dynamic programming approaches [13], where the order is inverted. Implicitly, we are assuming at the $k-$th stage that the disturbance w_k is known. This means full knowledge of total inflows at the start of each time interval.

Dynamic Programming allows for decoupling of the optimization problem (1) across stages. For this purpose, let us define the cost-to-go function from stage k onwards:

$$V_k(x) = \mathbb{E}\left[\min_{u_k,\ldots,u_{K-1}} \sum_{j=k}^{K-1} g_k(x_k, u_k, w_k) \mid x_k = x\right] \tag{2}$$

As usual, the main idea behind this decoupling is to compute the cost-to-go for stage $k+1$ and, in a recursive manner, use this solution to compute the cost-to-go for stage k by using Bellman Equation [13]:

$$V_k(x_k) = \mathbb{E}\left[\min_{u_k \in \mathcal{U}_k(x_k, w_k)} \{g_k(x_k, u_k, w_k) + V_{k+1}(x_{k+1})\}\right] \tag{3}$$

where the set $\mathcal{U}_k(x_k, w_k)$ involves box-type constraints (e.g.: release and spill terms must be non-negative and bounded), power balance, etc. It can be shown that the value functions are convex, given that the stage cost is convex and the transition dynamics are affine in both the state and the control [10].

A. Hydrologic state space model

To capture correlations in water inflows across stages we expand the state variable to include a discrete Markov state $e_k = e$ that summarizes the current hydrological environment. Its dynamics are governed by an homogeneous Markov chain, with transition probabilities:

$$\mathcal{P}_{ee'} = P\left(e_{k+1} = e' \mid e_k = e\right) \tag{4}$$

This probabilities may be estimated from historical data. One possibility is letting e_k take two values (corresponding to dry and wet) as introduced in [14]. Local practice in Uruguay is to use a 5-level model which spans from very-dry to very-wet [15], with transitions given by a non-homogeneous Markov chain. We propose keeping this 5-level discretization while modelling the hydrologic state evolution as time invariant. This entails procuring a single transition matrix $\mathcal{P} \in \mathbb{R}^{5\times5}$ from the available data, which will be accomplished using Principal Component Analysis [16], [17]. A more thorough description of our proposed model is presented later in Section IV-B.

We separate the hydrologic state e from the reservoir levels x and solve the expected value in Bellman Equation in two steps. Since this hydrologic state can only take discrete values, we can compute a different value function $V_{k,e}(x)$ for each possible value of e. Then, for given $e_k = e$, we estimate the future cost-to-go by an expected value over the next hydrologic state e', computed according to the finite probabilistic model given in (4). The generalized Bellman iteration thus becomes:

$$V_{k,e}(x) = \mathbb{E}\left[\min_{u}\left\{g_k\left(x, u, w\right) + \sum_{e'} \mathcal{P}_{ee'}.V_{k+1,e'}\left(x'\right)\right\}\right] \tag{5}$$
$$x' = f_k(x, u, w)$$
$$u \in \mathcal{U}_k\left(x, w\right)$$

where the outmost expectation is taken over inflows w conditioned to $e_k = e$. The rightmost sum in (5) can be interpreted as an estimate of the future cost-to-go given the current hydrologic state. If the costs g_k and the dynamics f_k are affine (5) is a linearly constrained quadratic program [11, p.152] and can be efficiently solved using standard techniques.

III. Algorithm

A. Backward pass

As has been argued before, our goal is to compute approximate value functions $\tilde{V}_{k,e}(x)$ quadratic in x, for every stage k and hydrologic state e. Each iteration of the backward dynamic programming algorithm is subdivided into two parts: a *sampling stage* and a *fitting stage*. The sampling stage consists of obtaining state-cost pairs (x, β) by solving an approximate Montecarlo-based version of (5):

$$\hat{\beta}_{k,e}(x, w_i) = \min_{u}\left\{g_k\left(x, u, w_i\right) + \sum_{e'} \mathcal{P}_{ee'}.\tilde{V}_{k+1,e'}\left(x'\right)\right\} \tag{6}$$

$$\beta_{k,e}(x) = \frac{1}{M}\sum_{i=0}^{M-1} \hat{\beta}_{k,e}(x, w_i) \tag{7}$$

Upon obtaining N pairs $\left(x_k^s, \beta_{k,e}^s\right)$, we fit the quadratic value function by solving:

$$\min_{P,q,r} \sum_{s=0}^{N-1} \left(x_k^{s\top} P x_k^s + q^\top x_k^s + r - \beta_{k,e}^s\right)^2 \tag{8}$$
$$\text{s. t.: } P \succeq 0$$

The computational complexity of our proposed method resides in solving $N \times M$ linearly constrained quadratic programs (as in (6)) and one semidefinite program (as in (8)) for each stage and hydrologic state.

B. Forward pass

Once all the value functions are approximated, the expected cost of running the system from a certain initial state x and certain hydrologic state e could be solved for by simply evaluating the fitted function $\tilde{V}_{0,e}(x)$. However, each stage of the backwards phase introduces errors on the approximations, and therefore the predicted cost $\tilde{V}_{0,e}$ might differ from the true cost substantially. In order to gauge the actual cost obtained by our methodology, a forward phase is carried out. This phase implements a Montecarlo simulation scheme which sequentially solves the one-stage optimization problem:

$$u_k = \underset{u \in \mathcal{U}_k(x_k, w_k)}{\arg\min} \left\{ g_k\left(x_k, u, w_k\right) + \sum_{e'} \mathcal{P}_{e_k e'} . \tilde{V}_{k+1, e'}\left(x_{k+1}\right) \right\} \tag{9}$$

$$x_{k+1} = f_k(x_k, u, w_k)$$

starting at $k = 0$ with initial storage level $x_0 = x$ and hydrological state $e_0 = e$. The incurred cost of operation over the planned horizon is the expected sum of the running cost per stages:

$$total\ cost(x, e) = \mathbb{E}\left[\sum_{k=0}^{K-1} g_k(x_k, u_k, w_k) \mid x_0 = x, e_0 = e \right] \tag{10}$$

where the expectation is taken over all possible sequences $\{(w_k, e_{k+1})\}_{k=0}^{k=K-1}$, and the control laws u_k are derived from (9). This simulated cost corresponds with what would be obtained by deploying our policy, and is therefore a better figure of merit for evaluating performance than the predictions $\tilde{V}_{0,e}(x)$.

Moreover, the obtained policy's performance can be contrasted with the performance of the *myopic policy*, which at time k seeks to minimize the current stage cost:

$$\begin{aligned} u_k^{myopic} = & \underset{u \in \mathcal{U}_k(x_k, w_k)}{\arg\min}\ g_k\left(x_k, u, w_k\right) \\ & x_{k+1} = f_k(x_k, u, w_k) \end{aligned} \tag{11}$$

Intuitively, at each step the myopic policy will use up (possibly all) the available water, minimizing the current cost and disregarding the utility of water in the future. While at first glance a reasonable thing to do, this behavior is generally suboptimal due to the expected inflows over the next steps and the spatial interconnection of the dams. For example, it could be better suited to store water now (at the expense of a higher cost) for use later, when a drought is expected.

Hope is set for our methodology to outperform the myopic policy. But how good can our policy really be? Although this question remains unanswered, we can construct a *lower bound* on the optimal performance. For a *given* inflow sequence $\mathbf{w} = \{w_0, \ldots, w_{K-1}\}$ the optimal decisions $\mathbf{u} = \{u_0, \ldots, u_{K-1}\}$ and the optimal cost can be obtained by solving the $K-$stage problem:

$$\begin{aligned} \mathbf{u}^{LB}(x, \mathbf{w}) = & \underset{\mathbf{u} \in \mathbf{U}(x, \mathbf{w})}{\arg\min} \sum_{k=0}^{K-1} g_k(x_k, u_k, w_k) \\ & x_{k+1} = f_k(x_k, u_k, w_k) \quad \forall k = 0, \ldots, K-1 \\ & x_0 = x \end{aligned} \tag{12}$$

where $\mathbf{u} \in \mathbf{U}(x, \mathbf{w})$ means that $u_k \in \mathcal{U}_k(x_k, w_k)$ for each k, with the sets $\mathcal{U}_k(x_k, w_k)$ described in (3). Problem (12) solves for the whole decision sequence $\mathbf{u} = \{u_0, \ldots, u_{K-1}\}$ at once, by being given full knowledge of all the noise realizations $\mathbf{w}$ at the start of the planning horizon. This is in sharp contrast with our proposed algorithm, where at each stage k the controller only has access to the current noise w_k. The expected cost of running (12) over all the possible inflow sequences $\mathbf{w}$ is indeed a lower bound on (10), since no approximations are used in the objective function and more information is available for planning.

IV. Test case: the Uruguayan system

A. The Uruguayan system

Uruguay is a small country with a demand profile that seldom surpasses $2000MW$. It is comprised of 4 hydroelectric plants: 3 of which are located in a cascade-like fashion along the *Río Negro* basin; the fourth one is located in the *Río Uruguay*, and is shared with neighbouring Argentina. The combined installed power in said facilities is roughly $1500MW$. There are a number of wind farms in Uruguay, with a total installed power amounting to more than 75% of the country's peak load. In recent years, there has been a surge in the installation of solar farms as well [18].

We will employ a one-year horizon with weekly decisions ($K = 52$ weeks in a year, $k = 0, \ldots, K-1$). In that regard, non-dispatchable renewables (wind and solar) will be left out of our model since they typically vary on a much faster timescale. Generation will be provided by the four hydroelectric plants and by a single thermal generator representing the aggregate thermal generation of the whole system. The state vector $x_k \in \mathbb{R}^4$ represents the current volume at each of the four reservoirs. The control $u_k = \left[r_k^\top, s_k^\top, t_k\right]^\top \in \mathbb{R}_+^9$ consists of the release ($r_k \in \mathbb{R}_+^4$) and spill vectors ($s_k \in \mathbb{R}_+^4$) and the total thermal generation ($t_k \in \mathbb{R}_+$). The state dynamics are described by

$$x_{k+1} = f(x_k, u_k, w_k) = x_k + B\left(r_k + s_k\right) + w_k \tag{13}$$

where B is the coupling matrix that captures the interconnection between hydro plants:

$$B = \begin{pmatrix} -1 & 0 & 0 & 0 \\ 1 & -1 & 0 & 0 \\ 0 & 1 & -1 & 0 \\ 0 & 0 & 0 & -1 \end{pmatrix} \tag{14}$$

and the vector w_k gathers the weekly inflows at each reservoir, as detailed in the next Section. Finally, the cost function $g(t_k)$ is the cost incurred by thermal generation, modeled as linear and time-invariant.

B. Markov Model estimation

The series used in this study case consist of the weekly inflow data from the three main reservoirs in Uruguay collected for 105 years.

At first, the negative values are replaced by zeros and then, for this application, all the zeros are handled as Not Available.

1) Normalization: Each one of the three series of hydraulic inflow is divided by its weekly median to remove the seasonal variations.

Graficas de medianas anuales, no se si irían acá

Then, when logarithm is applied to the series obtained before, it can be observed that the new series present a normal distribution.

Histogramas de series lognormales

2) Model estimation: For the model estimation the rows with Not Available data are removed, so they are not considered in the model.

a) Clustering considering three-dimension data.: In this first model, the K-means algorithm is applied to the three variables data set. This algorithm consists in separating the data into K clusters in a way that the euclidean distance between each point to the centroid of the assigned group is minimized.

The basic idea of the algorithm is: given an initial but not optimal clustering, relocate each point to its new nearest center, update the clustering centers by calculating the mean of the member points, and repeat the relocating-and-updating process until convergence criteria (such as predefined number of iterations, difference on the value of the distortion function) are satisfied.cita

For this study, a five cluster model is chosen and it is important to highlight that numbers assigned (1, 2, 3, 4, 5) to each cluster are not related to the hydraulicity of the group because they are assigned randomly by the algorithm. Finally, the data which were not assigned to any cluster because one or more of their components were equal to zero, are studied replacing the zeros by a small value. Then to label them, the modified points are assigned to a cluster in a way that the euclidean distance between its centroid and the point is minimized. grafica cluster k-means

b) Clustering considering principal components analysis (PCA):

In this case, the clustering method is applied to a linear combination of the three inflow series. The coefficients used in the linear combination are the result of the principal component analysis of the series that considers the direction in which the variation of the series is maximized. This direction is now the principal axis, and then, the other axes are determined because of the orthogonality. The linear combination consists of projecting the three components of the points into these axes.

To find the principal direction, the covariance matrix of the data set is calculated, as well as its eigenvectors and eigenvalues. The principal direction is the one associated with the biggest eigenvalue, and before doing the projection into this direction, the eigenvector is normalized with the 1-norm. Once we have the projected series, the clustering method is applied. In this method the 20% percentile is calculated and then the data is labeled according to the five ranges determined between the percentiles. Afterward, the points which were not labeled into any cluster because one or more of their components were equal to zero, are assigned to the first cluster.

grafica cluster PCA

c) Markov process estimation:

For both clustering methods explained before, in order to estimate the Markov process, the transitions between different clusters are counted.

The parameters p_{ij} of the markovian matrix $\mathcal{P}$ represent the probability of making a transition from cluster i to cluster j. This parameters are calculated as follows:

$$\widehat{p}_{ij} = \frac{\sum_{t=1}^{T} \mathbf{1}_{x_{t-1=i}, x_{t=j}}}{\sum_{t=1}^{T} \mathbf{1}_{x_{t=i}}} \tag{15}$$

where x_t represents que state in time t, and the sum is computed along the complete state secuence.

C. Simulation using Markov transitions

Trials begin at an initial state x_0 and initial hydrologic state e_0. At each time step the hydrologic sequence is updated with the markovian matrix $\mathcal{P}$ derived in (15), and a disturbance vector w_k corresponding to said hydrologic state is sampled. In order to approximate the total cost of running the system, we substitute the expected value in (10) with a sample mean carried out over $T = 105$ forward passes.

1) Performance for varying training points: State-cost pairs are sampled by partinioning the state space in a grid-like fashion. Each of the four reservoirs $i = 0, \ldots, 3$ is uniformly partitioned in N_i steps, yielding a total number of $N = N_0 \times N_1 \times N_2 \times N_3$ state points. The cost at each point is obtained by averaging over $M = 10$ different noise realizations. It is worth emphasizing that the state variables are not discretized, but these grid points are knots where we anchor our quadratic model to find the specified parameters using (8). The results shown herafter are for varying N_0, which corresponds to the discretization of the largest reservoir *Bonete*. For the other reservoirs we fix $N_i = 3$. As an illustrating example, Fig. 1 shows a cut of the quadratic obtained for the fourtieth week of the year with $N_0 = 10$.

Sampling more state-cost pairs at every stage naturally increases the computational effort required to perform the backward pass (Table I). Nonetheless, our experiments show that there is no significant performance gain in the obtained policy if more points are used in the training phase (Fig. 2).

TABLE I
TRAINING TIME AS A FUNCTION OF DISCRETIZATION STEPS ON BONETE

$\mathbf{N_0}$	3	5	10	15
Time (s)	6113	11289	19312	29084

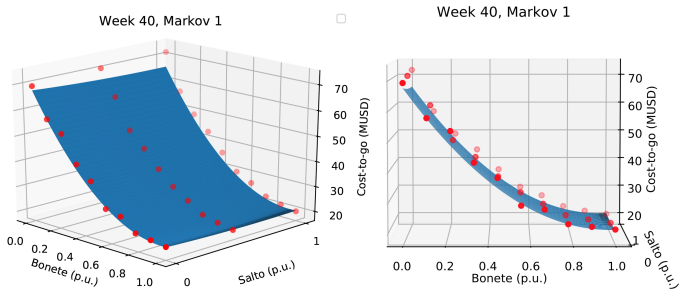

Fig. 1. Fitted quadratic function for the fourtieth week of the year and wet hydrologic state ($e_{40} = 1$) for $N_0 = 10$. In red: sampled state-cost pairs (using (6)–(7)). In blue: fitted quadratic function (using (8)). Note that the cost-to-go seems to primarily depend on the state of *Bonete* dam.

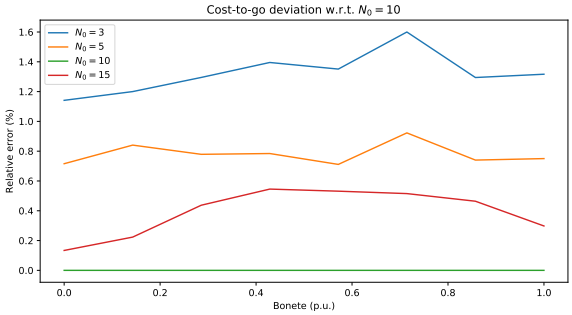

Fig. 2. Percentual cost deviation with respect to $N_0 = 10$ as a function of Bonete's initial level. Each line corresponds to a different policy trained with varying degree of discretization of Bonete (N_0). Note that all the policies attain a similar cost.

2) Bounds on performance and comparison with myopic policy: We can compare the predicted cost-to-go at the start of the year $V_{0,e_0}(x_0)$ with the simulated total cost $\sum_{k=0}^{K-1} g_k$ achieved by running the system forward starting from e_0 and x_0, following the learned policy (see (9)–(10)). Fig. 3 shows a comparison between the predicted and simulated cost as a function of the level of the largest reservoir, while starting from a neither-dry-nor-wet hydrologic state ($e_0 = 2$). A lower bound is constructed by solving the $K-$stages problem (12) given full knowledge of the noise realizations. Our experiments show that the predictions $\tilde{V}_{0,e}$ are typically optimistic.

The policy achieved by our proposed algorithm typically outperforms the so-called *myopic policy* (11), in particular for non-empty initial reservoir levels, as portrayed in Fig. 4.

D. Simulation using historical series

We also perform simulations using the historical series of inflows that were used for fitting our markov model. We compare the cost attained by our policy with the cost attained by a policy that was trained with the Markov model currently in use in Uruguay, and obtain better performance (see Fig. 5).

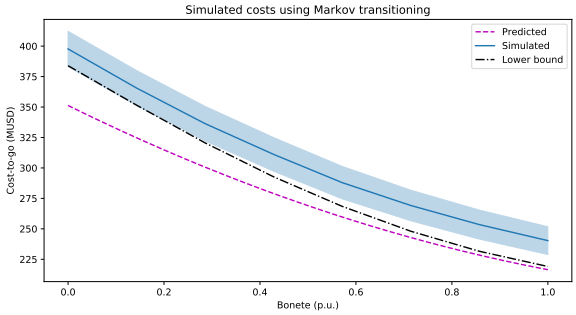

Fig. 3. Annual cost as a function of largest reservoir initial level for $e_0 = 2$: comparison between simulated and predicted costs, along with a performance bound. Simulated costs are averaged over $T = 105$ different trials. Mean cost is plotted in solid blue; shaded interval is defined as $\pm\sqrt{\sigma}/T$ where σ is the sample deviation.

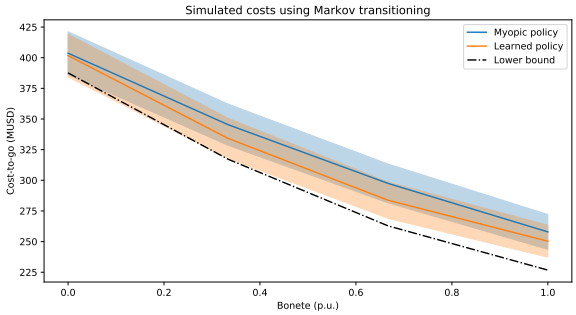

Fig. 4. Comparison between the myopic (blue) and learned policy (orange) for varying initial storage levels, along with a performance bound. Our policy achieves a 4% reduction on cost w.r.t. the myopic policy when storage levels are half-full, and performs at most 9% worse than the lower bound policy.

V. Conclusions

We proposed the use of convex quadratic functions to approximate the cost-to-go of a simple economic dispatch problem. We showed that training our method involves solving a large number of convex optimization problems, typically quadratic and semidefinite programs, while computing our policy consists of solving a quadratic program at each step. We benchmarked our algorithm on the Uruguayan power system, obtaining performance that surpasses that of a myopic policy, while incurring a cost that is not far from a theoretical lower bound.

Acknowledgements

This work was partially supported by ANII under grant FSE_1_2017_1_145060 and by UTE through Project UTE-FJR-UdelaR-ORT PT 001 2018

References

[1] W. W.-G. Yeh, "Reservoir management and operations models: A state-of-the-art review," *Water resources research*, vol. 21, no. 12, pp. 1797–1818, 1985.

[2] J. W. Labadie, "Optimal operation of multireservoir systems: state-of-the-art review," *Journal of water resources planning and management*, vol. 130, no. 2, pp. 93–111, 2004.

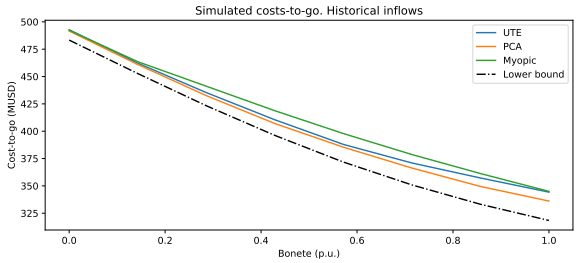

Fig. 5. Mean annual cost as a function of largest reservoir (Bonete). Inflow sequences correspond with historical data. The policy trained with our proposed Markovian model (PCA) achieves a 2.4% reduction on cost w.r.t. a policy trained with the model currently used in Uruguay (UTE).

[3] D. Rani and M. M. Moreira, "Simulation–optimization modeling: a survey and potential application in reservoir systems operation," *Water resources management*, vol. 24, no. 6, pp. 1107–1138, 2010.

[4] W. B. Powell, *Approximate Dynamic Programming: Solving the curses of dimensionality*. John Wiley & Sons, 2007, vol. 703.

[5] M. V. Pereira and L. M. Pinto, "Multi-stage stochastic optimization applied to energy planning," *Mathematical programming*, vol. 52, no. 1-3, pp. 359–375, 1991.

[6] R. Porteiro, A. Ferragut, and F. Paganini, "Towards multi-timescale energy provisioning using stochastic dual dynamic programming," in *2018 IEEE 9th Power, Instrumentation and Measurement Meeting (EPIM)*. IEEE, 2018, pp. 1–6.

[7] S. Barratt and S. Boyd, "Stochastic control with affine dynamics and extended quadratic costs," *arXiv preprint arXiv:1811.00168*, 2018.

[8] J. Nascimento and W. B. Powell, "An optimal approximate dynamic programming algorithm for concave, scalar storage problems with vector-valued controls," *IEEE Transactions on Automatic Control*, vol. 58, no. 12, pp. 2995–3010, 2013.

[9] J. M. Nascimento and W. B. Powell, "An optimal approximate dynamic programming algorithm for the lagged asset acquisition problem," *Mathematics of Operations Research*, vol. 34, no. 1, pp. 210–237, 2009. [Online]. Available: https://doi.org/10.1287/moor.1080.0360

[10] A. Keshavarz and S. Boyd, "Quadratic approximate dynamic programming for input-affine systems," *International Journal of Robust and Nonlinear Control*, vol. 24, no. 3, pp. 432–449, 2014.

[11] S. Boyd, S. P. Boyd, and L. Vandenberghe, *Convex optimization*. Cambridge university press, 2004.

[12] L. Vandenberghe and S. Boyd, "Semidefinite programming," *SIAM review*, vol. 38, no. 1, pp. 49–95, 1996.

[13] D. P. Bertsekas, *Dynamic programming and optimal control*, 4th ed., vol. I, no. 2.

[14] A. B. Philpott and V. L. De Matos, "Dynamic sampling algorithms for multi-stage stochastic programs with risk aversion," *European Journal of operational research*, vol. 218, no. 2, pp. 470–483, 2012.

[15] G. Casaravilla, R. Chaer, and P. Alfaro, "SimSEE: Simulador de sistemas de energía eléctrica," *Proyecto PDT 47/12. Technical Report 7, Universidad de la República (Uruguay). Facultad de Ingeniería. Instituto de Ingeniería Elétrica, Number 7-Dec, Tech. Rep.*, 2008.

[16] S. Wold, K. Esbensen, and P. Geladi, "Principal component analysis," *Chemometrics and intelligent laboratory systems*, vol. 2, no. 1-3, pp. 37–52, 1987.

[17] H. Abdi and L. J. Williams, "Principal component analysis," *Wiley interdisciplinary reviews: computational statistics*, vol. 2, no. 4, pp. 433–459, 2010.

[18] "ADME: Administración del mercado eléctrico," www.adme.com.uy, accessed April 2020.