SDDP algorithm for Brazilian Power System Generation

Based on ONS-GT project

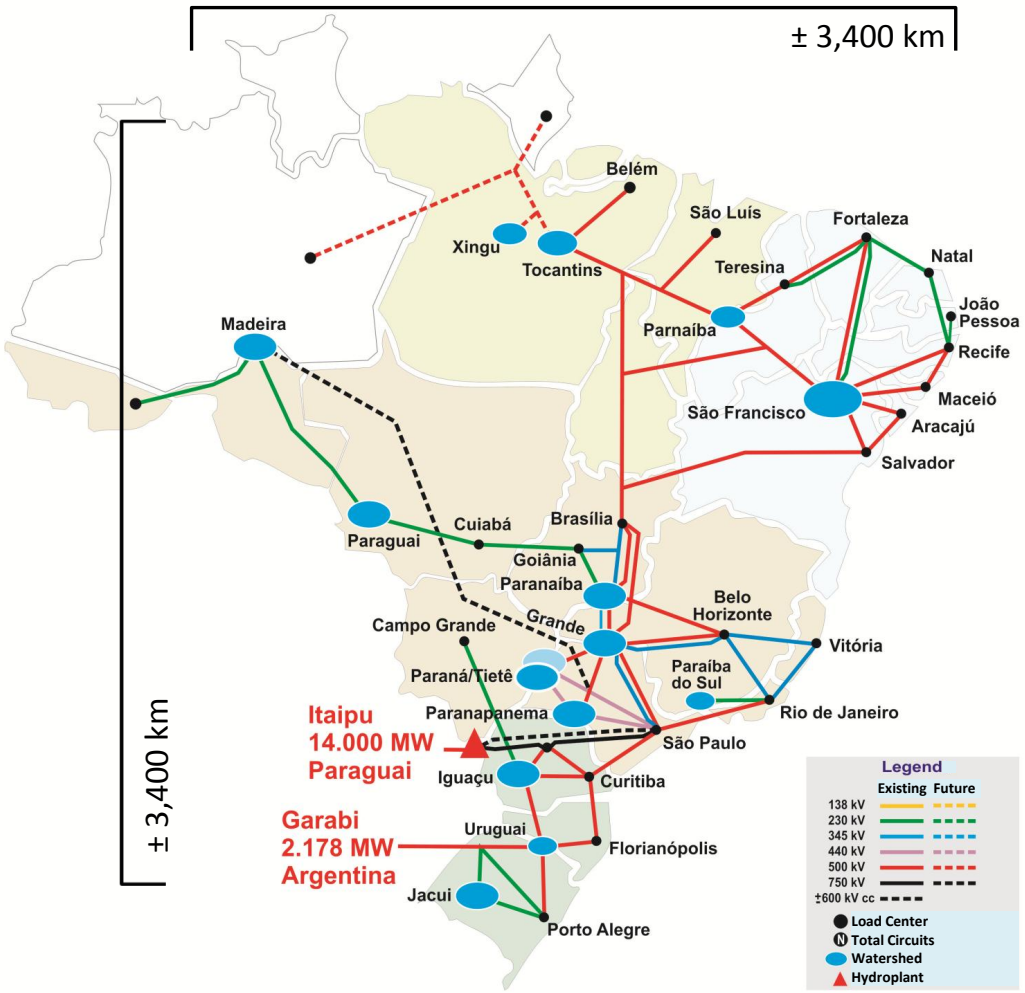
Alexander Shapiro, Wajdi Tekaya
Georgia Institute of Technology,
Atlanta, Georgia 30332-0205, USA

Joari Paulo da Costa, Murilo Pereira Soares
ONS - Operador Nacional do Sistema Eletrico

The Brazilian power system generation is hydro dominated (about 75% of the installed capacity) and characterized by large reservoirs presenting multi-year regulation capability, arranged in complex cascades over several river basins. The hydro plants use store water in the reservoirs to produce energy in the future, replacing fuel costs from the thermal units. Since the water inflows depend on rainfalls, the amount of future inflows is uncertain and cannot be predicted with a high accuracy.

The purpose of hydrothermal system operation planning is to define an operation strategy which, for each stage of the planning period, given the system state at the beginning of the stage, produces generation targets for each plant.

The Brazilian hydro power operation planning problem is a multistage, large scale (more than 200 power plants, of which 141 are hydro plants), stochastic optimization problem. On a high level, planning is for 5 years on monthly basis together with 5 additional years to smooth out the end of horizon effect. This results in 120-stage stochastic programming problem. Four energy equivalent reservoirs are considered, one in each one of the four interconnected main regions, SE, S, N and NE. The resulting policy obtained with the aggregate representation can be further refined, so as to provide decisions for each of the hydro and thermal power plants.


± 3,400 km
± 3,400 km
Belém
São Luís
Fortaleza
Xingu
Tocantins
Teresina
Natal
João Pessoa
Madeira
Parnaíba
Recife
Maceió
São Francisco
Aracajú
Salvador
Brasília
Cuiabá
Paraguai
Goiânia
Paranaíba
Belo Horizonte
Campo Grande
Grande
Vitória
Paraíba do Sul
Paraná/Tietê
Rio de Janeiro
Itaipu
14.000 MW
Paraguai
Paranapanema
São Paulo
Iguaçu
Curitiba
Garabi
2.178 MW
Argentina
Uruguai
Florianópolis
Jacui
Porto Alegre
Legend
Existing Future
138 kV
230 kV
345 kV
440 kV
500 kV
750 kV
±600 kV cc
Load Center
Total Circuits
Watershed
Hydroplant

Nested formulation the linear multistage problem

$$\operatorname*{Min}_{\substack{A_1x_1=b_1\\ x_1\geq 0}} c_1^{\mathsf{T}}x_1+\mathbb{E}\left[\operatorname*{Min}_{\substack{B_2x_1+A_2x_2=b_2\\ x_2\geq 0}} c_2^{\mathsf{T}}x_2+\cdots+\mathbb{E}\left[\operatorname*{Min}_{\substack{B_Tx_{T-1}+A_Tx_T=b_T\\ x_T\geq 0}} c_T^{\mathsf{T}}x_T\right]\right].$$

Equivalent formulation

$$\begin{array}{ll}\operatorname*{Min}\limits_{x_1,x_2(\cdot),...,x_T(\cdot)} & \mathbb{E}\left[c_1^{\mathsf{T}}x_1+c_2^{\mathsf{T}}x_2(\xi_{[2]})...+c_T^{\mathsf{T}}x_T(\xi_{[T]})\right]\\ \text{s.t.} & A_1x_1=b_1,\ x_1\geq 0,\\ & B_tx_{t-1}(\xi_{[t-1]})+A_tx_t(\xi_{[t]})=b_t,\\ & x_t(\xi_{[t]})\geq 0,\ t=2,...,T.\end{array}$$

Here $\xi_t=(c_t,B_t,A_t,b_t)$, $t=2,...,T$, is considered as a random process, $\xi_1=(c_1,A_1,b_1)$ is supposed to be known, $\xi_{[t]}:=(\xi_1,...,\xi_t)$ denotes history of the data process up to time t.

Optimization is performed over feasible policies (also called decision rules). A policy is a sequence of (measurable) functions $x_t = x_t(\xi_{[t]})$, $t = 1, ..., T$. Each $x_t(\xi_{[t]})$ is a function of the data process up to time t, this ensures the *nonanticipative* property of a considered policy.

Policy suggests a decision rule for every possible realization of the data process.

A policy is *feasible* if it satisfies the feasibility constraints w.p.1, i.e., for almost every realization of the data process.

Ideally, the multistage problem is solved if one can construct a feasible policy minimizing the expected value of the (total) cost.

Three levels of approximation

"Any model is wrong but some are useful".

From a modeling point of view it is natural to assume that the random data process has a *continuous* distribution. We refer to such model as "true" or "continuous".

Next question is how to compute the involved expectations (multivariate integrals). A standard approach is to discretize the random process by generating a finite number of possible realizations (called scenarios). These scenarios can be represented by the corresponding *scenario tree*.

How to construct such scenario tree?

Simplifying assumption: we assume that the data process is stagewise independent. That is random vector ξ_{t+1} is in independent of $\xi_{[t]}$, $t = 1, ..., T-1$.

In some cases stagewise dependent problems can be reformulated in a stagewise independent form at the price of increasing number of state variables.

For example, suppose that only the right hand side vectors b_t are random and can be modeled as a (first order) autoregressive process

$$b_t = \mu + \Phi b_{t-1} + \varepsilon_t,$$

where μ and Φ are (deterministic) vector and regression matrix, respectively, and the error process ε_t, $t = 1, ..., T$, is stagewise independent. The corresponding feasibility constraints can be written in terms of x_t and b_t as

$$B_t x_{t-1} + A_t x_t \leq b_t, \;\; \Phi b_{t-1} - b_t + \mu + \varepsilon_t = 0.$$

That is, in terms of decision variables (x_t, b_t) this becomes a linear multistage stochastic programming problem governed by the stagewise independent random process $\varepsilon_1, ..., \varepsilon_T$.

Discretization by Monte Carlo sampling (second level of approximation)

Independent of each other random samples $\xi_t^j = (c_t^j, B_t^j, A_t^j, b_t^j)$, $j = 1, ..., N_t$, of respective ξ_t, $t = 2, ..., T$, are generated and the corresponding scenario tree is constructed by connecting every ancestor node at stage $t-1$ with the same set of children nodes $\xi_t^1, ..., \xi_t^{N_t}$. In that way the stagewise independence is preserved in the generated scenario tree. We refer to the constructed problem as the Sample Average Approximation (SAA) problem.

The total number of scenarios of the SAA problem is given by the product $\mathcal{N} = \prod_{t=2}^{T} N_t$ and quickly becomes astronomically large with increase of the number of stages even for moderate values of sample sizes N_t.

How large should be the sample sizes $N_2, ..., N_T$ in order for the SAA problem to give a reasonable approximation of the true problem?

Because of the exponential growth of the number of scenarios $\mathcal{N}$ it is hopeless to try to solve multistage stochastic programs by enumerating all scenarios. An alternative approach is suggested by the dynamic programming.

Dynamic programming equations for the SAA problem

Going backward in time the so-called *cost-to-go* functions are defined recursively for $t = T, ..., 2$, as follows

$$Q_t^j(x_{t-1}) = \inf_{\substack{B_t^j x_{t-1} + A_t^j x_t = b_t^j \\ x_t \geq 0}} \left\{ (c_t^j)^{\mathsf{T}} x_t + \mathcal{Q}_{t+1}(x_t) \right\},$$

$j = 1, ..., N_t$, with $\mathcal{Q}_{T+1}(\cdot) \equiv 0$ and

$$\mathcal{Q}_{t+1}(x_t) = \frac{1}{N_{t+1}} \sum_{j=1}^{N_{t+1}} Q_{t+1}^j(x_t).$$

At the first stage the following problem should be solved

$$\operatorname*{Min}_{x_1} c_1^{\mathsf{T}} x_1 + \mathcal{Q}_2(x_1) \text{ s.t. } A_1 x_1 = b_1,\ x_1 \geq 0.$$

The optimal value of the first stage problem gives the optimal value of the corresponding multistage problem.

A policy $\bar{x}_t = \bar{x}_t(\xi_{[t]})$, $t = 1, \ldots, T$, is *optimal* if $\bar{x}_1$ is an optimal solution of the first stage problem and for $t = 2, \ldots, T$, it holds for every realization of the data process that

$$\bar{x}_t(\xi_{[t]}) \in \arg \min_{\substack{B_t \bar{x}_{t-1} + A_t x_t = b_t \\ x_t \geq 0}} \left\{ c_t^{\mathsf{T}} x_t + \mathcal{Q}_{t+1}(x_t) \right\}.$$

In the dynamic programming formulation the problem is reduced to solving a sequence of finite dimensional problems, indexed by t and depending on $\xi_{[t]}$.

For linear programs the cost-to-go (value) functions $Q_t^j(x_{t-1})$ and $\mathcal{Q}_{t+1}(x_t)$ are *convex*.

Because of the stagewise independence condition, the cost-to-go functions $\mathcal{Q}_{t+1}(x_t)$ are functions of x_t and do not depend on the data process ξ_t.

Curse of dimensionality

One of the main difficulties in solving the dynamic programming equations is how to represent the cost-to-go functions in a computationally feasible way.

For dimension of x_t say greater than 3 and large number of stages it is practically impossible to solve the dynamic programming equations with high accuracy. Several alternatives were suggested in recent literature.

Approximate dynamic programming (third level of approximation)

Basic idea is to approximate the cost-to-go functions by a class of computationally manageable functions. Since functions $\mathcal{Q}_t(\cdot)$ are convex it is natural to approximate these functions by piecewise linear functions given by maximum of cutting hyperplanes.

Stochastic Dual Dynamic Programming (SDDP) method.

For trial decisions $\bar{x}_t$, $t = 1, ..., T-1$, at the backward step of the SDDP algorithm, piecewise linear approximations $\mathfrak{Q}_t(\cdot)$ of the cost-to-go functions $\mathcal{Q}_t(\cdot)$ are constructed by solving problems

$$\operatorname*{Min}_{x_t \in \mathbb{R}^{n_t}} (c_t^j)^{\mathsf{T}} x_t + \mathfrak{Q}_{t+1}(x_t) \text{ s.t. } B_t^j \bar{x}_{t-1} + A_t^j x_t = b_t^j, \ x_t \geq 0,$$

$j = 1, ..., N_t$, and their duals, going backward in time $t = T, ..., 1$.

Denote by v^0 and $\widehat{v}_N$ the respective optimal values of the true and SAA problems.

By construction

$$\mathcal{Q}_t(\cdot) \geq \mathfrak{Q}_t(\cdot),\ t = 2, ..., T.$$

Therefore the optimal value of

$$\operatorname*{Min}_{x_1 \in \mathbb{R}^{n_1}} c_1^{\mathsf{T}} x_1 + \mathfrak{Q}_2(x_1) \ \text{ s.t. } A_1 x_1 = b_1,\ x_1 \geq 0$$

gives a lower bound for the optimal value $\widehat{v}_N$ of the SAA problem.

We also have that

$$v^0 \geq \mathbb{E}[\widehat{v}_N].$$

Therefore *on average* $\widehat{v}_N$ is also a lower bound for the optimal value of the true problem.

The approximate cost-to-go functions $\mathfrak{Q}_2, ..., \mathfrak{Q}_T$ and a feasible first stage solution $\bar{x}_1$ define a feasible policy. That is for a realization (sample path) $\xi_1, ..., \xi_T$ of the data process, $\bar{x}_t = \bar{x}_t(\xi_{[t]})$ are computed recursively in $t = 2, ..., T$ as a solution of

$$\underset{x_t}{\text{Min}}\, c_t^\top x_t + \mathfrak{Q}_{t+1}(x_t) \text{ s.t. } B_t \bar{x}_{t-1} + A_t x_t \leq b_t.$$

In the *forward step* of the SDDP algorithm M sample paths (scenarios) are generated and the corresponding $\bar{x}_t$, $t = 2, ..., T$, are used as trial points in the next iteration of the backward step.

It is absolutely essential for convergence of this algorithm that at each iteration in the forward step the paths (scenarios) are *resampled*, i.e., generated independently of the previous iteration.

Note that the functions $\mathfrak{Q}_2, ..., \mathfrak{Q}_T$ and $\bar{x}_1$ define a feasible policy also for the *true* problem.

Convergence of the SDDP algorithm

It is possible to show that, under mild regularity conditions, the SDDP algorithm converges as the number of iterations go to infinity. That is, the computed optimal values and generated policies converge w.p.1 to their counterparts of the considered SAA problem. However, the convergence can be very slow and one should take such mathematical proofs very cautiously.

Moreover, it should be remembered that the SAA problem is just an approximation of the "true" problem. It is possible to show that, in a certain probabilistic sense, the SAA problem converges to the "true" problem as all sample sizes N_t, $t = 2, ..., T$, tend to infinity.
It was found in our numerical experiments that optimal solutions of the SAA problems started to stabilize for sample sizes of about $N_t = 100$, $t = 2, ..., T$.

Stopping criteria

The policy value $\mathbb{E}\left[\sum_{t=1}^{T} c_t^{\mathsf{T}} \bar{x}_t(\xi_{[t]})\right]$ can be estimated in the forward step of the algorithm. That is, let $\xi_2^i, ..., \xi_T^i$, $i = 1, ..., M$, be sample paths (scenarios) generated at a current iteration of the forward step, and

$$\vartheta_i := \sum_{t=1}^{T} (c_t^i)^{\mathsf{T}} \bar{x}_t^i, \ i = 1, ..., M,$$

be the corresponding cost values. Then $\mathbb{E}[\vartheta_i] = \mathbb{E}\left[\sum_{t=1}^{T} c_t^{\mathsf{T}} \bar{x}_t(\xi_{[t]}^i)\right]$, and hence

$$\bar{\vartheta} = \frac{1}{M} \sum_{i=1}^{M} \vartheta_i$$

gives an unbiased estimate of the policy value.

Also

$$\widehat{\sigma}^2 = \frac{1}{M-1}\sum_{i=1}^{M}(\vartheta_i - \bar{\vartheta})^2$$

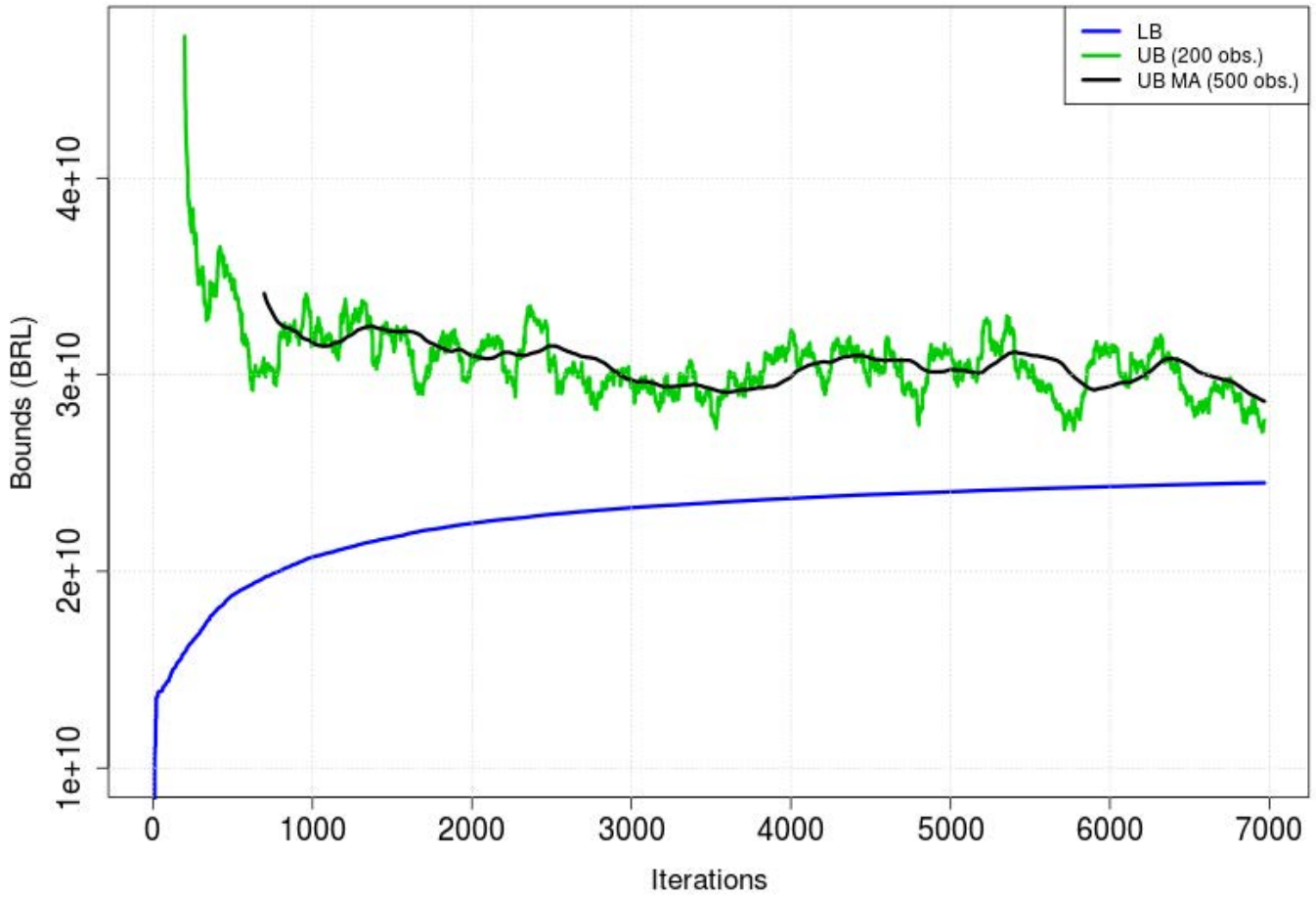
estimates variance of the sample $\vartheta_1, ..., \vartheta_M$. Hence

$$\bar{\vartheta} + z_\alpha \widehat{\sigma}/\sqrt{M}$$

gives an *upper* bound for the policy value with confidence of about $100(1-\alpha)\%$. Here z_α is the corresponding critical value.

At the same time this gives an upper bound for the optimal value of the corresponding multistage problem, SAA or the "true" problem depending from what data process the random scenarios were generated.

Typical example of behavior of the lower and upper bounds produced by the SDDP algorithm for an SAA problem

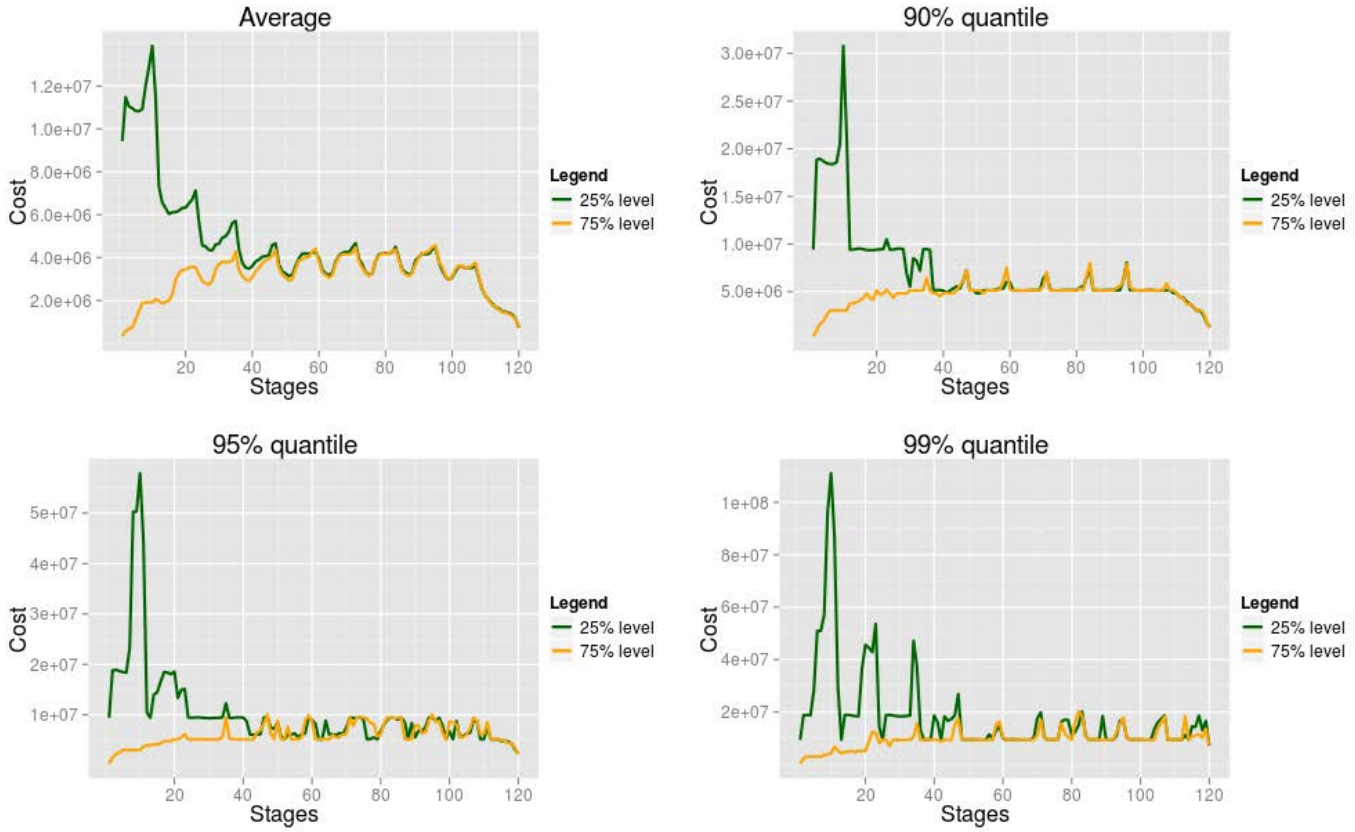


8 state variables, 120 stages, 1 cut per iteration

Theoretical analysis and numerical experiments indicate that computational complexity of the SDDP algorithm grows fast with increase of the number of state variables. The optimality gap jumped from 4% to 20% when the number of state variables was increased from 4 to 8 as a result of considering an autoregressive model.

Sensitivity to initial conditions

Individual stage costs for the risk neutral approach in two cases: all the reservoirs start at 25% or at 75% of the maximum capacity. The yellow curve denotes the 75% initial reservoir level and the dark green denotes the 25% initial level.


Average
Cost
1.2e+07
1.0e+07
8.0e+06
6.0e+06
4.0e+06
2.0e+06
20
40
60
80
100
120
Stages
Legend
25% level
75% level
90% quantile
Cost
3.0e+07
2.5e+07
2.0e+07
1.5e+07
1.0e+07
5.0e+06
20
40
60
80
100
120
Stages
Legend
25% level
75% level
95% quantile
Cost
5e+07
4e+07
3e+07
2e+07
1e+07
20
40
60
80
100
120
Stages
Legend
25% level
75% level
99% quantile
Cost
1e+08
8e+07
6e+07
4e+07
2e+07
20
40
60
80
100
120
Stages
Legend
25% level
75% level

Variability of SAA problems

Table shows the 95% confidence interval for the lower bound and average policy value at iteration 3000 over a sample of 20 SAA problems. Each of the policy value observations was computed using 2000 scenarios. The last 2 columns of the table shows the range divided by the average of the lower bound (where the range is the difference between the maximum and minimum observation) and the standard deviation divided by the average value. This problem has relatively low variability (approx. 4%) for both of the lower bound and the average policy value.

	95% C.I. left ($\times 10^9$)	Average ($\times 10^9$)	95% C.I. right ($\times 10^9$)	range average	sdev. average
Lower bound	22.290	22.695	23.100	15.92%	4.07%
Average policy	27.333	27.836	28.339	17.05%	4.12%

SAA variability for risk neutral SDDP

Risk averse approach

How to control risk, i.e., to reduce chances of extreme costs, at every stage of the time process.

Value-at-Risk of a random outcome (variable) Z at level $\alpha \in (0,1)$:

$$\text{V@R}_\alpha(Z) = \inf\{t : F_Z(t) \geq 1-\alpha\},$$

where $F_Z(t) = \Pr(Z \leq t)$ is the cdf of Z. That is, $\text{V@R}_\alpha(Z)$ is the $(1-\alpha)$-quantile of the distribution of Z.

Note that $\text{V@R}_\alpha(Z) \leq c$ is equivalent to $\Pr(Z > c) \leq \alpha$. Therefore it could be a natural approach to impose constraints (chance constraints) of $\text{V@R}_\alpha(Z) \leq c$ for $Z = cost$, chosen constant c and significance level α at every stage of the process.

There are two problems with such approach. It is difficult to handle chance constraints numerically and could lead to infeasibility problems.

Average Value-at-Risk (also called *Conditional Value-at-Risk*)

$$\mathsf{AV@R}_\alpha(Z) = \inf_{t\in\mathbb{R}} \left\{ t + \alpha^{-1}\mathbb{E}[Z-t]_+ \right\}$$

Note that the minimum in the above is attained at $t^* = \mathsf{V@R}_\alpha(Z)$. If the cdf $F_Z(z)$ is continuous, then

$$\mathsf{AV@R}_\alpha(Z) = \mathbb{E}\Big[Z | Z \geq \mathsf{V@R}_\alpha(Z) \Big].$$

It follows that $\mathsf{AV@R}_\alpha(Z) \geq \mathsf{V@R}_\alpha(Z)$. Therefore the constraint $\mathsf{AV@R}_\alpha(Z) \leq c$ is a conservative approximation of the chance constraint $\mathsf{V@R}_\alpha(Z) \leq c$.

In the problem of minimizing expected cost $\mathbb{E}[Z]$ subject to the constraint $\mathsf{AV@R}_\alpha(Z) \leq c$, we impose an infinite penalty for violating this constraint. This could result in infeasibility of the obtained problem. Instead we can impose a finite penalty and consider problem of minimization of $\mathbb{E}[Z] + \kappa \mathsf{AV@R}_\alpha(Z)$ for some constant $\kappa > 0$. Note that this is equivalent to minimization of $\rho(Z)$, where

$$\rho(Z) = (1 - \lambda)\mathbb{E}[Z] + \lambda \mathsf{AV@R}_\alpha(Z)$$

for $\lambda \in (0, 1)$ and $\kappa = \frac{\lambda}{1-\lambda}$.

This leads to the following (nested) formulation of risk averse multistage problem.

$$\begin{aligned}\min_{A_1x_1\leq b_1} \quad & c_1^{\mathsf{T}}x_1 + \rho_{2|\xi_1}\Big[\inf_{\substack{B_2x_1+A_2x_2=b_2\\ x_2\geq 0}} c_2^{\mathsf{T}}x_2 + \ldots \\ & + \rho_{T-1|\xi_{[T-2]}}\Big[\inf_{\substack{B_{T-1}x_{T-2}+A_{T-1}x_{T-1}=b_{T-1}\\ x_{T-1}\geq 0}} c_{T-1}^{\mathsf{T}}x_{T-1} \\ & + \rho_{T|\xi_{[T-1]}}\big[\inf_{\substack{B_Tx_{T-1}+A_Tx_T=b_T\\ x_T\geq 0}} c_T^{\mathsf{T}}x_T\big]\Big]\Big],\end{aligned}$$

with

$$\rho_{t|\xi_{[t]}}(\cdot) := (1-\lambda)\mathbb{E}_{|\xi_{[t]}}[\cdot] + \lambda \mathsf{AV@R}_{\alpha|\xi_{[t]}}(\cdot)$$

being conditional analogue of $\rho(\cdot)$.

The value of this problem corresponds to the total objective

$$\begin{aligned}\bar{\rho}(Z_1 + ... + Z_T) &= \rho_{|\xi_{[1]}}\Big(\cdots \rho_{|\xi_{[T-1]}}(Z_1 + ... + Z_T)\Big) \\ &= Z_1 + \rho_{|\xi_{[1]}}\Big(Z_2 + \cdots + \rho_{|\xi_{[T-1]}}(Z_T)\Big)\end{aligned}$$

The dynamic programming equations of the risk averse formulation of the SAA program take the form

$$Q_t^j(x_{t-1}) = \inf_{x_t}\Big\{(c_t^j)^{\mathsf{T}}x_t + \mathcal{Q}_{t+1}(x_t) : B_t^j x_{t-1} + A_t^j x_t = b_t^j,\ x_t \geq 0\Big\},$$

$j = 1, ..., N_t$, $t = T, \ldots, 2$, and

$$\mathcal{Q}_{t+1}(x_t) = \rho\left(Q_{t+1}^1(x_t), ..., Q_{t+1}^{N_{t+1}}(x_t)\right),$$

with $\mathcal{Q}_{T+1}(\cdot) \equiv 0$ and the first stage problem

$$\operatorname*{Min}_{A_1 x_1 \leq b_1} c_1^{\mathsf{T}}x_1 + \rho\left(Q_2^1(x_1), ..., Q_2^{N_2}(x_1)\right).$$

For $\rho(\cdot) = (1-\lambda)\mathbb{E}[\cdot] + \lambda \mathsf{AV@R}_\alpha(\cdot)$, and $(Z_1, ..., Z_N) = \left(Q^1_{t+1}(x_t), ..., Q^N_{t+1}(x_t)\right)$ we have that

$$\mathcal{Q}_{t+1}(x_t) = \frac{1-\lambda}{N_{t+1}} \sum_{j=1}^{N_{t+1}} Z_j + \lambda \left(Z_\iota + \frac{1}{\alpha N_{t+1}} \sum_{j: Z_j > Z_\iota} \left[Z_j - Z_\iota \right] \right),$$

where Z_ι is the $(1-\alpha)$-quantile of $Z_1, ..., Z_{N_{t+1}}$. Note that if $N_{t+1} < (1-\alpha)^{-1}$, then $Z_\iota = \max\{Z_1, ..., Z_{N_{t+1}}\}$.

A subgradient of $\mathcal{Q}_{t+1}(x_t)$ is given by

$$\nabla \mathcal{Q}_{t+1}(x_t) = \frac{1-\lambda}{N} \sum_{j=1}^{N_{t+1}} \nabla Q^j_{t+1}(x_t) +$$

$$\lambda \left(\nabla Q^\iota_{t+1}(x_t) + \frac{1}{\alpha N_{t+1}} \sum_{j: Z_j > Z_\iota} \left[\nabla Q^j_{t+1}(x_t) - \nabla Q^\iota_{t+1}(x_t) \right] \right).$$

These formulas allow construction of cuts in the backward step of the SDDP algorithm. In the forward step trial points are generated in the same way as in the risk neutral case.

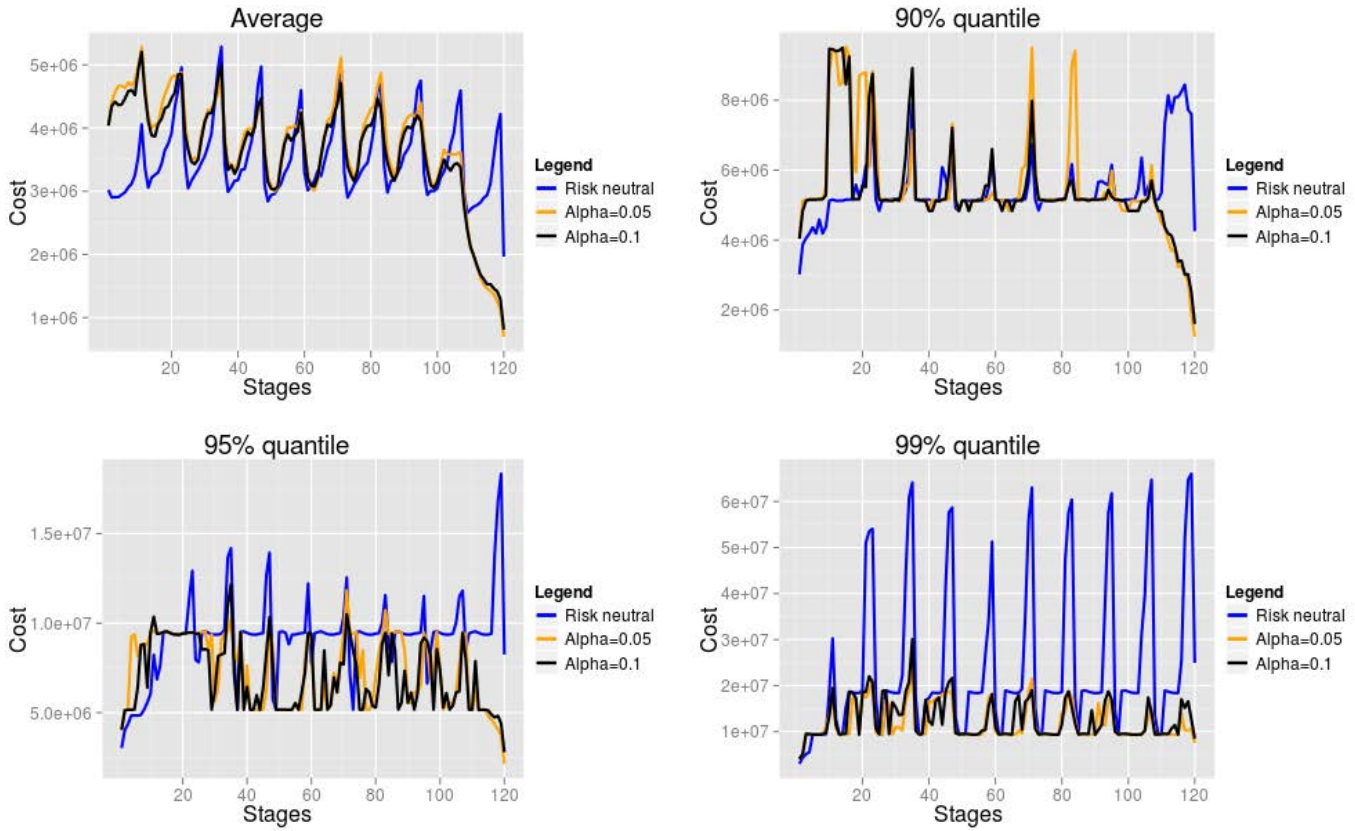
Remarks

Unfortunately there is no easy way for evaluating value of the risk objective of generated policies, and hence constructing a corresponding upper bound. Some suggestions were made in the recent literature. However, in larger problems the optimality gap (between the upper and lower bounds) never approaches zero in any realistic time. Therefore stopping criteria based on stabilization of the lower bound (and may be optimal solutions) could be reasonable. Also it should be remembered that there is no intuitive interpretation for the risk objective $\bar{\rho}(cost)$ of the total cost. Rather the goal is to control risk at every stage of the process.

It was observed in some numerical experiments that the algorithm converged faster for the risk averse optimization problems than for their risk neutral counterparts. This could be explained by that risk penalties make the problem somewhat better conditioned.

In principle it is possible to choose the parameters λ and α dynamically conditional on observed data realizations. This requires a further investigation.

Removal of redundant cuts can reduce computational times.



Individual stage costs for $\lambda = 0.15$ and $\alpha = 0.05$ and $\alpha = 0.1$.

Reports with numerical experiments can be downloaded from:

http://www2.isye.gatech.edu/∼ashapiro/publications.html

Papers:

Shapiro, A., Tekaya, W., Paulo da Costa, J. and Pereira Soares, M., "Risk neutral and risk averse Stochastic Dual Dynamic Programming method", *European Journal of Operations Research*, 224, 375-391, 2013.

Shapiro, A., Tekaya, W., Pereira Soares, M. and Paulo da Costa, J., "Worst-case-expectation approach to optimization under uncertainty", *Operations Research*, 61, 1435-1449, 2013.