Relative Value Iteration for Infinite-Horizon SDDP: Application to Hydroelectric Problem

M. Azéma, B.F.P. da Costa, F. Durand, V. Leclère

June 2, 2026

Outline

Outline

Finite Horizon: Multistage Stochastic Optimization

$$\begin{aligned} \min_{u_0,u_1,\ldots,u_T} \quad & \mathbb{E}\left[\sum_{t=0}^{T} \beta^t \, c_t(x_t, u_t)\right]. \\ \text{s.t.} \quad & u_t \in U_t(x_t) \\ & x_{t+1} = f_t(x_t, u_t, \xi_t) \in X_{t+1} \\ & x_0 = \underline{x}_0 \\ & u_t \preceq \xi_{[t]} \end{aligned}$$

(MDP) Notation

- In each stage t, the system is in *state* $x_t \in X_t$;
- An *action* $u_t \in U_t(x_t)$ is chosen;
  - incurs the *immediate cost* $c_t(x_t, u_t)$, and
  - transitions to the *next state* $x_{t+1} = f_t(x_t, u_t, \xi_t)$;
  - affected by the uncertainty, modeled by the random variable ξ_t.
- Discount factor (for inflation, ...): $0 < \beta < 1$

Finite Horizon: Multistage Stochastic Optimization

A complex and important system

- In Brazil, approximately 70% of the total energy supply comes from hydroelectric power.
- A long-term scheduling problem involving multiple interconnected reservoirs with uncertain water inflows, which depend on seasonal variations.

- State x_t: water levels in the reservoirs
- Action u_t: water release decisions, production of nuclear plants, ...
- Uncertainty ξ_t: water inflows, which depend on seasonal variations.
- Discount factor β: 0.99
- Finite horizon: 120 months (10 years)

Finite Horizon: Dynamic Programming

Under *stagewise independence* of the uncertainties ξ_t,

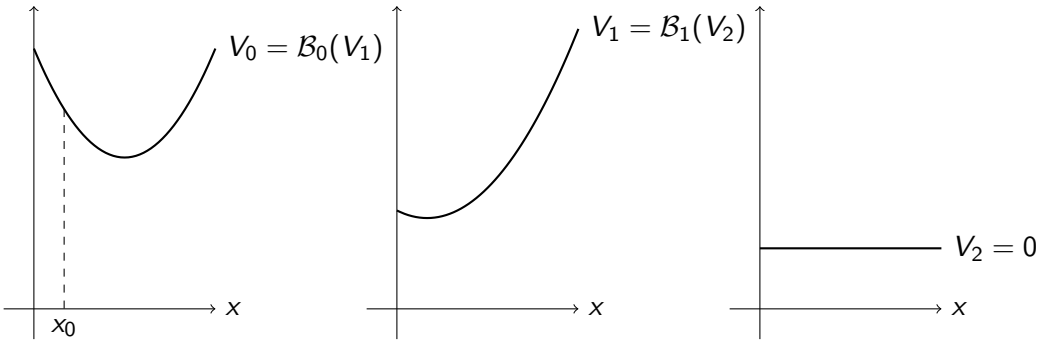
$$\begin{aligned} \min_{u_0, u_1, \ldots, u_T} \quad & \mathbb{E}\left[\sum_{t=0}^{T} \beta^t \, c_t(x_t, u_t)\right]. \\ \text{s.t.} \quad & u_t \in U_t(x_t) \\ & x_{t+1} = f_t(x_t, u_t, \xi_t), x_0 = \underline{x}_0 \\ & u_t \preceq \xi_{[t]} \end{aligned}$$

Dynamic Programming

$$\mathcal{B}_t(V)(x) = \mathbb{E}\left[\min_{u \in U_t(x)} c_t(x, u) + \beta \, V(\underbrace{f_t(x, u, \xi_t)}_{x_{t+1}})\right].$$

$$\begin{aligned} V_{T+1} &= 0 \\ V_t &= \mathcal{B}_t(V_{t+1}) \end{aligned}$$

Finite Horizon: Value Iteration

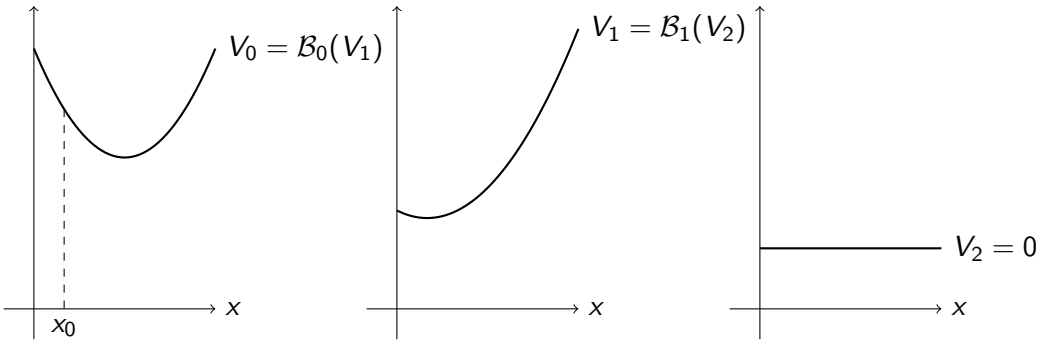


Problem: Computing $\mathcal{B}_t(V_{t+1})(x)$ for all x is impossible.

But, $\mathcal{B}_t$ preserves *convexity* and $V_{T+1} = 0$ is convex, so V_t is convex for all t.

$\implies$ SDDP: approximate V_t by a piecewise linear convex function.

Finite Horizon: Value Iteration

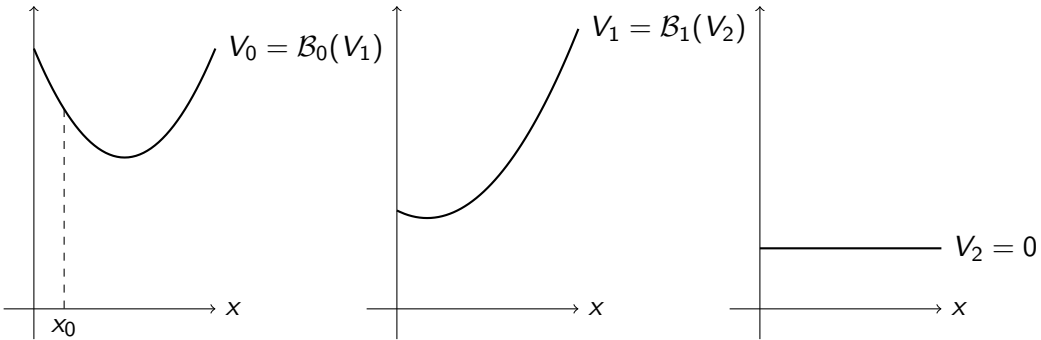


Problem: Computing $\mathcal{B}_t(V_{t+1})(x)$ for all x is impossible.

But, $\mathcal{B}_t$ preserves *convexity* and $V_{T+1} = 0$ is convex, so V_t is convex for all t.

$\Longrightarrow$ SDDP: approximate V_t by a piecewise linear convex function.

Finite Horizon: Value Iteration

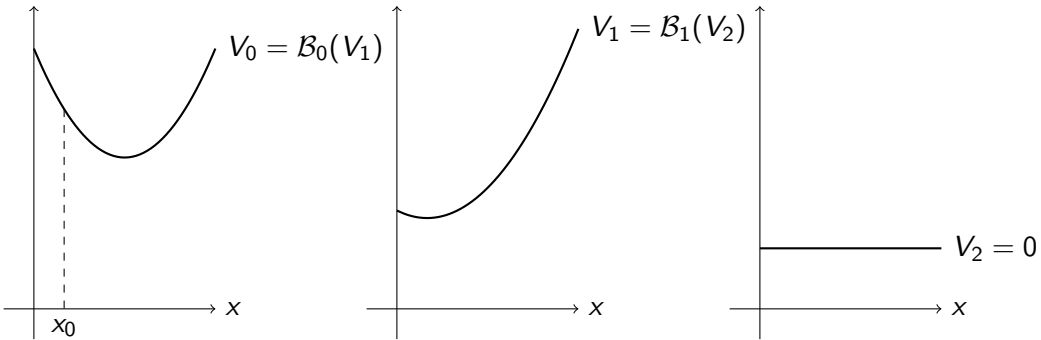


Problem: Computing $\mathcal{B}_t(V_{t+1})(x)$ for all x is impossible.

But, $\mathcal{B}_t$ preserves *convexity* and $V_{T+1} = 0$ is convex, so V_t is convex for all t.

$\Longrightarrow$ SDDP: approximate V_t by a piecewise linear convex function.

Finite Horizon: Value Iteration



Problem: Computing $\mathcal{B}_t(V_{t+1})(x)$ for all x is impossible.

But, $\mathcal{B}_t$ preserves *convexity* and $V_{T+1} = 0$ is convex, so V_t is convex for all t.

$\implies$ SDDP: approximate V_t by a piecewise linear convex function.

Finite Horizon: SDDP[1]

Algorithm 1: SDDP

Initialize value functions $V_t^0(x)$ with valid lower bounds

for $k = 0 : K$ **do**

  Simulate a realization of uncertainty $\xi_0^k, \ldots, \xi_T^k$

  for $t = 0 : T$ **do** `// forward pass`

    Simulate the policy using V_{t+1}^{k-1} to generate trial points $x_t^k \in X$

$$x_{t+1}^k = f_t(x_t^k, u_t^k, \xi_t^k), \quad u_t^k \in \arg\min_{u \in U_t(x_t^k)} c_t(x_t^k, u) + \beta V_{t+1}^{k-1}(f_t(x_t^k, u, \xi_t^k))$$

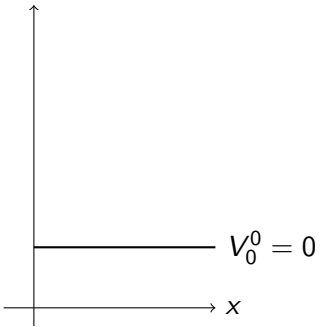
  for $t = T : 0$ **do** `// backward pass`

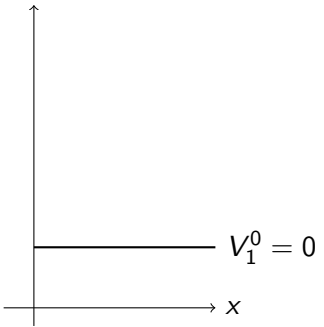
    Compute cut_t^k, a tangent to $\mathcal{B}_t(V_{t+1}^k)$ at x_t^k

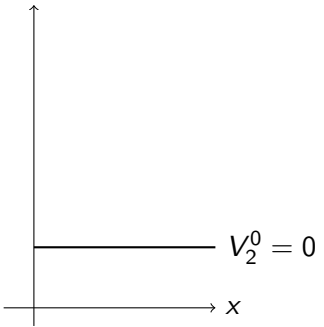
    Update: $V_t^k = \max(V_t^{k-1}, \text{cut}_t^k)$

[1] Mario VF Pereira and Leontina MVG Pinto (1991). "Multi-stage stochastic optimization applied to energy planning". In: *Mathematical programming* 52, pp. 359–375.

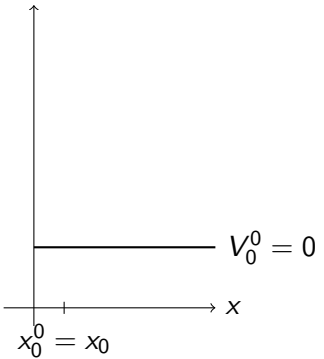
Finite Horizon: SDDP

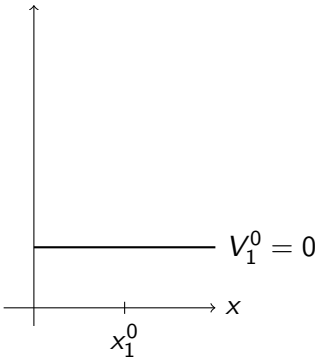


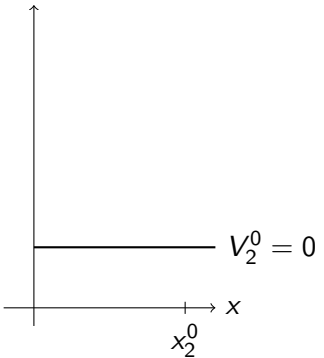




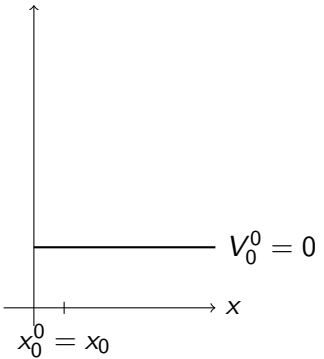
Finite Horizon: SDDP

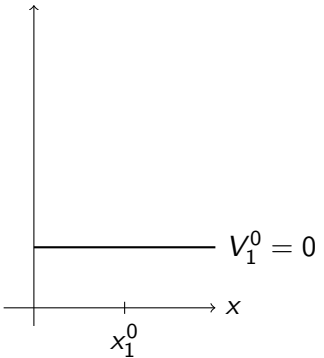


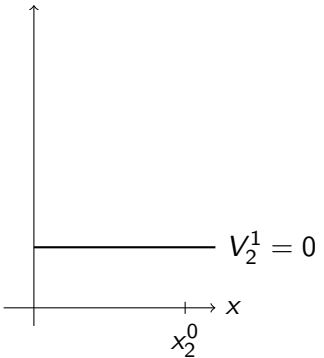




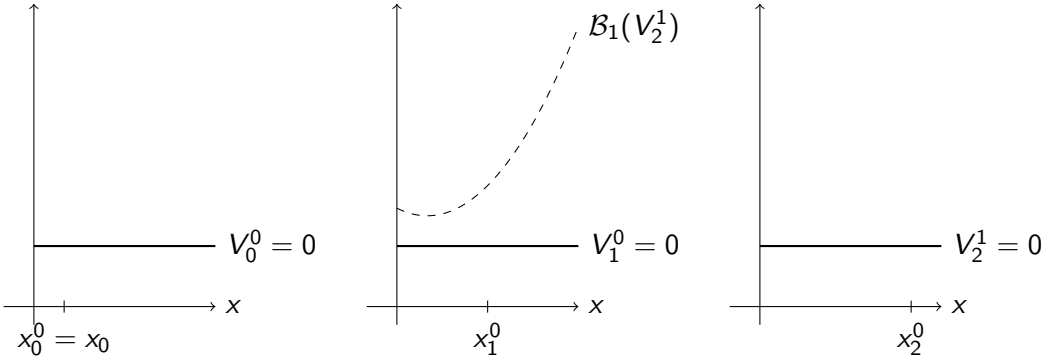
Finite Horizon: SDDP



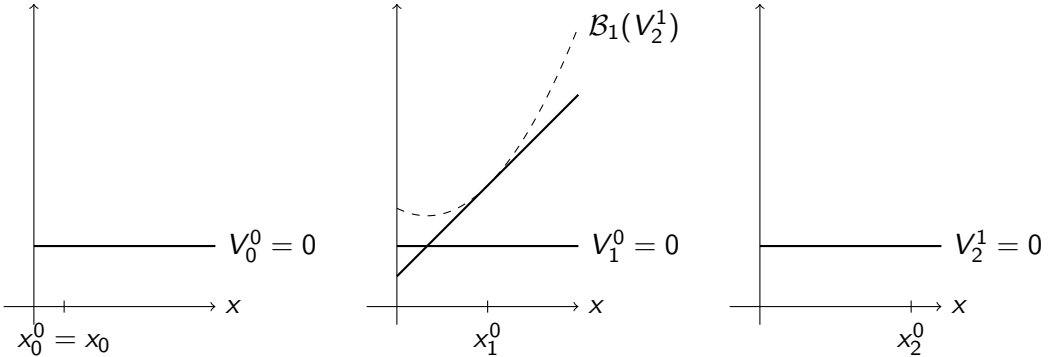




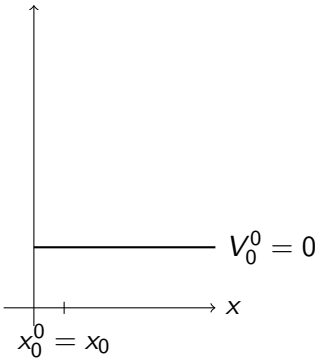
Finite Horizon: SDDP

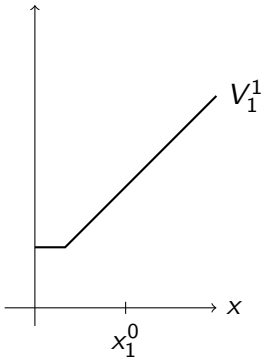


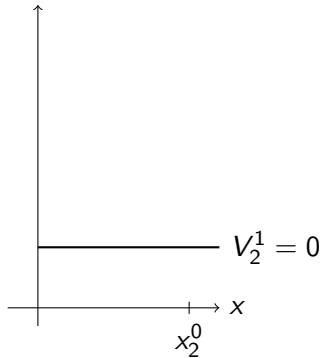
Finite Horizon: SDDP



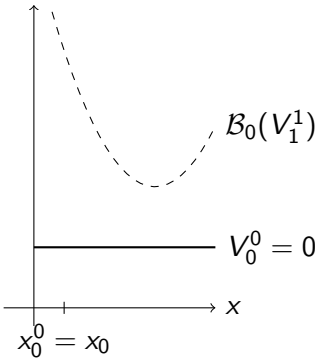
Finite Horizon: SDDP

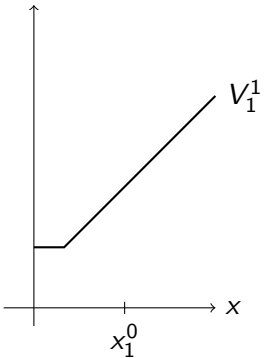


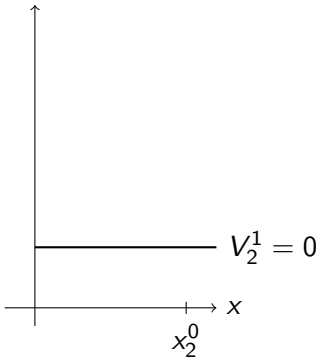




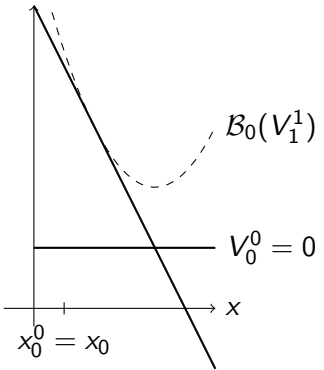
Finite Horizon: SDDP

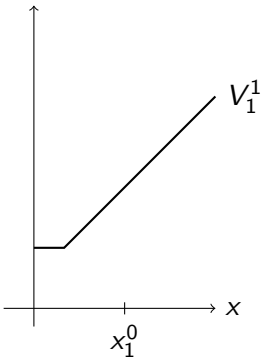


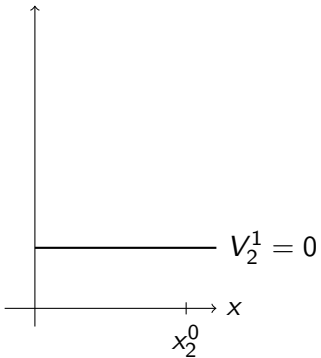




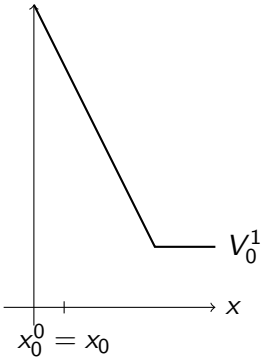
Finite Horizon: SDDP

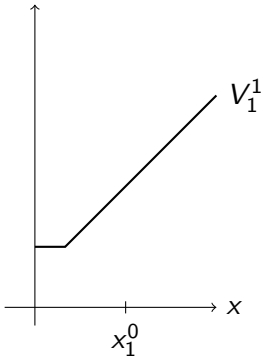


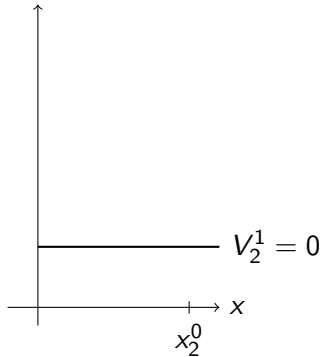




Finite Horizon: SDDP







Computational effort

Forward Pass At each iteration, we simulate a realization of uncertainty $\xi_0^k, \ldots, \xi_T^k$ and compute trial points x_t^k by solving T optimization problems of the form

$$u_t^k \in \arg \min_{u \in U_t(x_t^k)} c_t(x_t^k, u) + \beta V_{t+1}^{k-1}(f_t(x_t^k, u, \xi_t^k)).$$

Backward Pass To compute the cut, we need to solve for each possible realization of ξ_t the following optimization problem:

$$\begin{aligned} \min_{u \in U_t(x_t), x_t} \quad & c_t(x_t, u) + \beta V_{t+1}^k(f_t(x_t, u, \xi_t^k)) \\ \text{s.t.} \quad & x_t = x_t^k \qquad [\lambda_t^k] \end{aligned}$$

This requires solving $T \times S$ optimization problems, where S is the number of possible realizations of ξ_t.

Outline

Periodic infinite-horizon

$$\begin{aligned} \min_{u_0,u_1,\ldots} \quad & \mathbb{E}\left[\sum_{t=0}^{\infty} \beta^t c_t(x_t, u_t)\right] \\ \text{s.t.} \quad & u_t \in U_t(x_t) \\ & x_{t+1} = f_t(x_t, u_t, \xi_t) \end{aligned}$$

T-periodicity

- The functions c_t, f_t, the sets U_t and the distribution of ξ_t are *periodic* with period T.

Periodic infinite-horizon: Motivations[2][3]

Why periodic?

- Reduce computational complexity. Instead of 120 value functions, only 12 are needed.
- Avoid "end-of-horizon" effects (120 months may seem long, but may still be insufficient). For example, in the Brazilian hydro problem with $\beta = 0.99$ and $T = 120$, they show that only the 10 first stages are not affected by end-of-horizon effects.

[2] Alexander Shapiro and Lingquan Ding (2020). "Periodical multistage stochastic programs". In: *SIAM Journal on Optimization* 30.3, pp. 2083–2102.

[3] Alexander Shapiro and Yi Cheng (2023). "Dual bounds for periodical stochastic programs". In: *Operations Research* 71.1, pp. 120–128.

Bellman equations for periodic infinite-horizon

$$\mathcal{B}_t(V)(x) = \mathbb{E}\left[\min_{u \in U_t(x)} c_t(x,u) + \beta\, V(f_t(x,u,\xi_t))\right].$$

$$V_t = \mathcal{B}_t(V_{t+1}) \quad t = 1, \ldots, T-1$$
$$V_T = \mathcal{B}_T(V_1)$$

Infinite Horizon: 1-Periodic model

$$\mathcal{B}(V)(x) = \mathbb{E}\left[\min_{u \in U(x)} c(x,u) + \beta\, V(f(x,u,\xi))\right].$$

$$V = \mathcal{B}(V)$$

Properties of the Bellman operator $\mathcal{B}$

- The discount factor ensures that the Bellman operator $\mathcal{B}$ is a *contraction*.
- Convergence of $\mathcal{B}^n(V^0)$ to the fixed point: *slower* as $\beta \to 1$.
- If $V \leq V'$, then $\mathcal{B}(V) \leq \mathcal{B}(V')$.
- If the costs c are positive, then $\mathcal{B}(0) \geq 0$, and the fixed point V is positive.
- $\mathcal{B}$ preserves convexity and the fixed point V is convex.

Infinite Horizon: Periodic SDDP

Algorithm 2: 1-Periodic SDDP:

Initialize value function $V_0(x)$
$n = 0$;
for $k = 1 : K$ **do**
  Choose forward pass length T_k
  for $t = 1 : T_k$ **do** // forward pass
    Simulate the policy using V_n to generate trial points $x_t^k \in X$
  for $t = T_k : 1$ **do** // backward pass
    Compute cut_t^k, a cut to $\mathcal{B}(V_n)$ at x_t^k
    Update: $V_{n+1} = \max(V_n, \text{cut}_t^k)$
    $n = n + 1$

A toy example, $\beta = 0.9$:

- 1d state space;
- As β is close to 1, *several initial cuts will be subsumed*.
- The initial cuts aim at approximating a constant proportional to $(1-\beta)^{-1} = \sum_{t=0}^{\infty} \beta^t$.

Figure: Value functions for several SDDP iterations.

A toy example, $\beta = 0.9$:

- 1d state space;
- As β is close to 1, *several initial cuts will be subsumed*.
- The initial cuts aim at approximating a constant proportional to $(1-\beta)^{-1} = \sum_{t=0}^{\infty} \beta^t$.

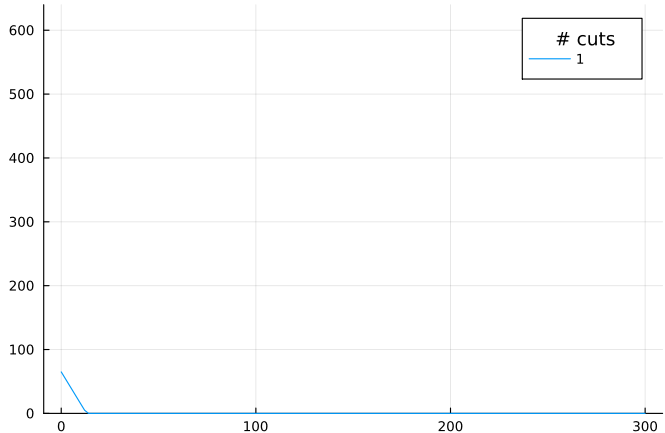


Figure: Value functions for several SDDP iterations.

A toy example, $\beta = 0.9$:

- 1d state space;
- As β is close to 1, *several initial cuts will be subsumed*.
- The initial cuts aim at approximating a constant proportional to $(1-\beta)^{-1} = \sum_{t=0}^{\infty} \beta^t$.

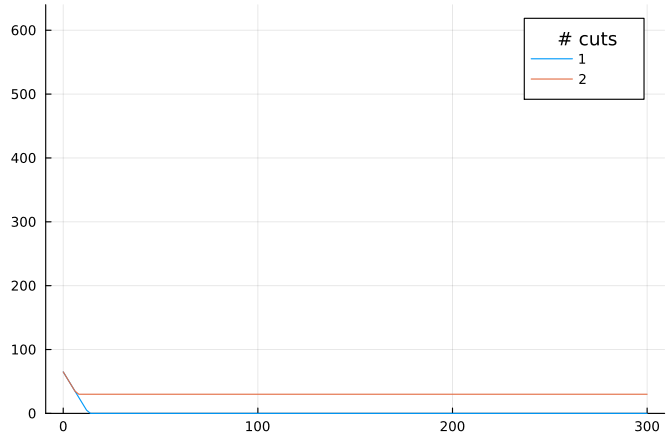


Figure: Value functions for several SDDP iterations.

A toy example, $\beta = 0.9$:

- 1d state space;
- As β is close to 1, *several initial cuts will be subsumed*.
- The initial cuts aim at approximating a constant proportional to $(1-\beta)^{-1} = \sum_{t=0}^{\infty} \beta^t$.

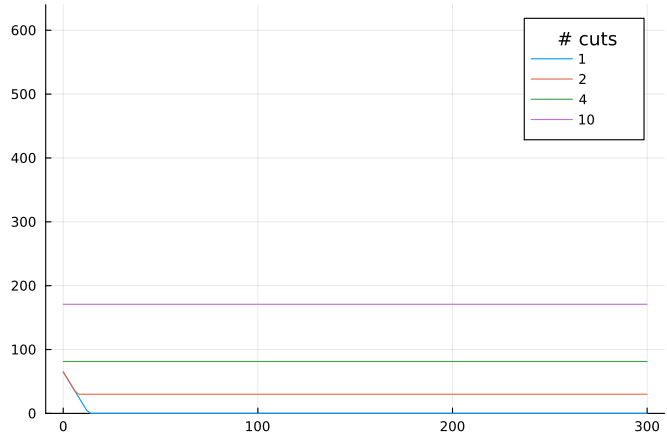


Figure: Value functions for several SDDP iterations.

A toy example, $\beta = 0.9$:

- 1d state space;
- As β is close to 1, *several initial cuts will be subsumed*.
- The initial cuts aim at approximating a constant proportional to $(1-\beta)^{-1} = \sum_{t=0}^{\infty} \beta^t$.

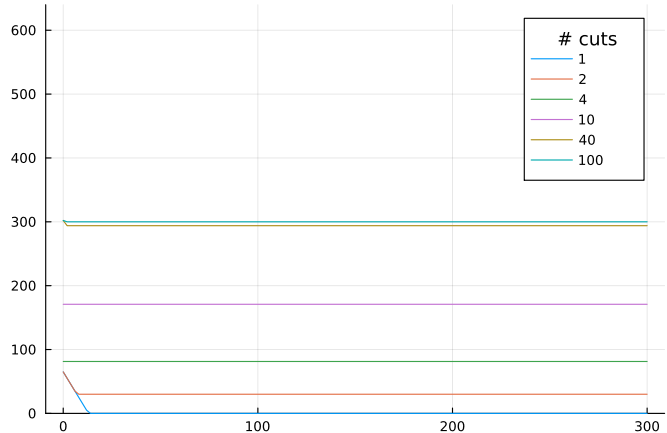


Figure: Value functions for several SDDP iterations.

A toy example, $\beta = 0.9$:

- 1d state space;
- As β is close to 1, *several initial cuts will be subsumed*.
- The initial cuts aim at approximating a constant proportional to $(1-\beta)^{-1} = \sum_{t=0}^{\infty} \beta^t$.

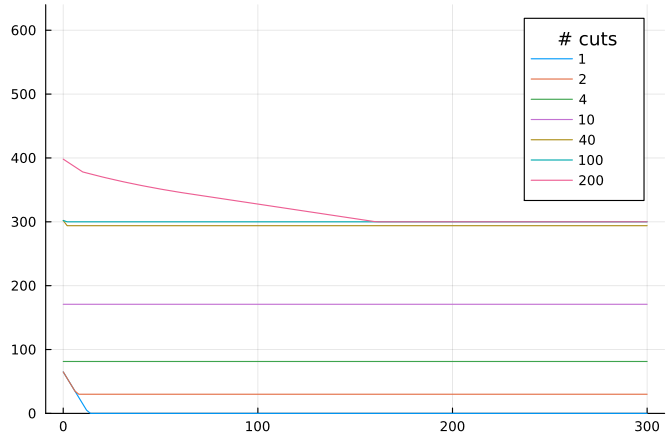


Figure: Value functions for several SDDP iterations.

A toy example, $\beta = 0.9$:

- 1d state space;
- As β is close to 1, *several initial cuts will be subsumed*.
- The initial cuts aim at approximating a constant proportional to $(1-\beta)^{-1} = \sum_{t=0}^{\infty} \beta^t$.

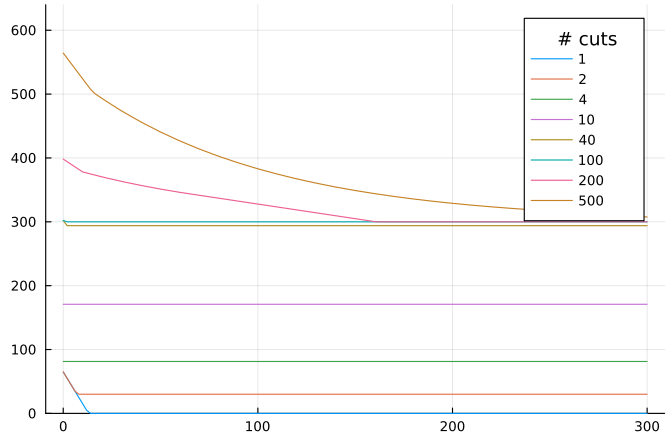


Figure: Value functions for several SDDP iterations.

Outline

Relative Value Iteration[4][5]

$$\mathcal{B}(V)(x) = \mathbb{E}\left[\min_{u \in U(x)} c(x,u) + \beta\, V(f(x,u,\xi))\right].$$

- Developed for *Markov Decision Processes* with finite state and action spaces for the *ergodic case* ($\beta = 1$).
- Idea: the optimal policy is *invariant* under translation of the value function by a constant. We only need to find V such that $\mathcal{B}(V) = V + \text{constant}$.

$$V_0 = 0$$
$$V_{n+1} = \mathcal{B}(V_n) - \mathcal{B}(V_n)(x^{\text{ref}})$$

[4] Douglas J White (1963). "Dynamic programming, Markov chains, and the method of successive approximations". In: *Journal of Mathematical Analysis and Applications* 6.3, pp. 373–376.

[5] Ari Arapostathis and Vivek S Borkar (2019). "Average cost optimal control under weak ergodicity hypotheses: Relative value iterations". In: *arXiv:1902.01048*.

Toward RV-SDDP

RVI

$V_0 = 0$

$V_{n+1} = \mathcal{B}(V_n) - \mathcal{B}(V_n)(x^{\text{ref}})$

monotonic RVI

$V_0^{\Delta} = 0$

$V_{n+1}^{\Delta} = \max(V_n^{\Delta}, \mathcal{B}(V_n^{\Delta}) - \Delta_n)$

converges to V_{∞}^{Δ}

Theorem

Let $(\Delta_n)_{n\in\mathbb{N}}$ be a sequence of non-negative shift values. If

$$\delta = \liminf_{n\to\infty} \Delta_n \leq \min_{y\in X} \left\{ \mathcal{B}(V_{\infty}^{\Delta})(y) - V_{\infty}^{\Delta}(y) \right\},$$

then the function V_{∞}^{Δ} satisfies: $\mathcal{B}(V_{\infty}^{\Delta}) = V_{\infty}^{\Delta} + \delta$.

Toward RV-SDDP

RVI

$V_0 = 0$

$V_{n+1} = \mathcal{B}(V_n) - \mathcal{B}(V_n)(x^{\text{ref}})$

monotonic RVI

$V_0^\Delta = 0$

$V_{n+1}^\Delta = \max(V_n^\Delta, \mathcal{B}(V_n^\Delta) - \Delta_n)$

converges to V_∞^Δ

Theorem

Let $(\Delta_n)_{n\in\mathbb{N}}$ be a sequence of non-negative shift values. If

$$\delta = \liminf_{n\to\infty} \Delta_n \leq \min_{y\in X} \left\{ \mathcal{B}(V_\infty^\Delta)(y) - V_\infty^\Delta(y) \right\},$$

then the function V_∞^Δ satisfies: $\mathcal{B}(V_\infty^\Delta) = V_\infty^\Delta + \delta$.

Possible shifts

monotonic RVI

$$V_0^{\Delta} = 0, \qquad V_{n+1}^{\Delta}(x) = \max\left(V_n^{\Delta}(x), \mathcal{B}(V_n^{\Delta})(x) - \Delta_n\right).$$

Convergence condition

$$\delta := \liminf_{n\to\infty} \Delta_n \leq \min_{y\in X} \left\{\mathcal{B}(V_\infty^{\Delta})(y) - V_\infty^{\Delta}(y)\right\}.$$

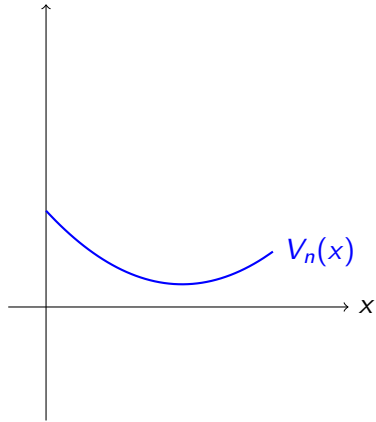
Possible choices for the shift Δ_n:

1. $\Delta_n = 0$ for all $n \to V_{n+1}^{\Delta} = \mathcal{B}(V_n^{\Delta})$: VI algorithm. ✓
2. $\Delta_n = \min_{x\in X} \left\{\mathcal{B}(V_n^{\Delta})(x) - V_n^{\Delta}(x)\right\}$. ✓
3. $\Delta_n = \mathcal{B}(V_n^{\Delta})(y) - V_n^{\Delta}(y)$, for some random $y \in X$. ✓
4. $\Delta_n = \mathcal{B}(V_n^{\Delta})(x^{ref})$. ✗

RV-SDDP: First idea

$V_0^\Delta = 0$

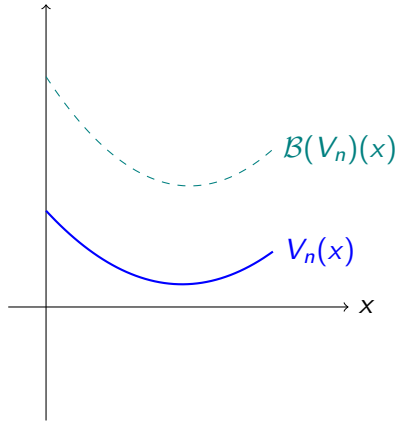
$V_{n+1}^\Delta(x) = \max(V_n^\Delta(x), \text{cut}_n(x) - \Delta_n)$



RV-SDDP: First idea

$V_0^\Delta = 0$

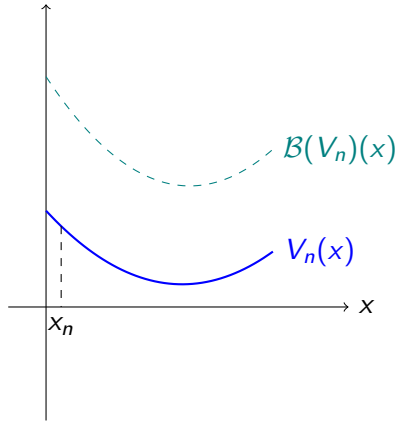
$V_{n+1}^\Delta(x) = \max(V_n^\Delta(x), \mathrm{cut}_n(x) - \Delta_n)$



RV-SDDP: First idea

$V_0^\Delta = 0$

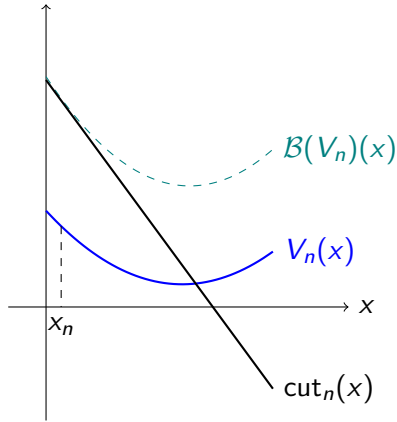
$V_{n+1}^\Delta(x) = \max(V_n^\Delta(x), \text{cut}_n(x) - \Delta_n)$



RV-SDDP: First idea

$V_0^\Delta = 0$

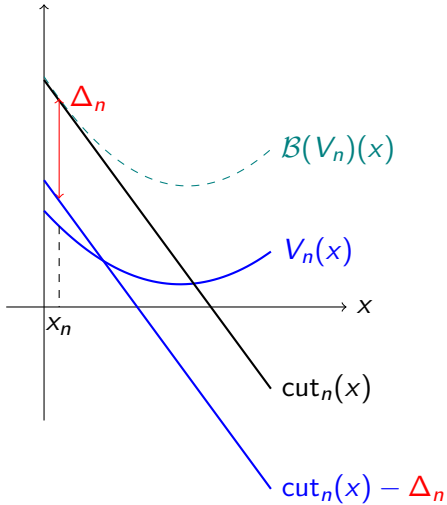
$V_{n+1}^\Delta(x) = \max(V_n^\Delta(x), \text{cut}_n(x) - \Delta_n)$



RV-SDDP: First idea

$V_0^\Delta = 0$

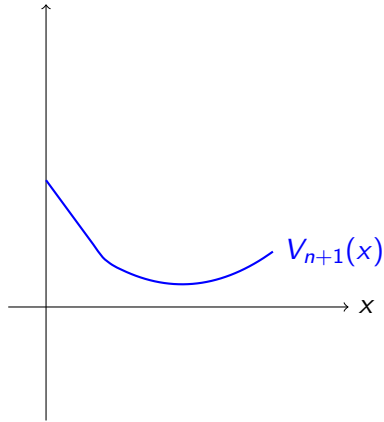
$V_{n+1}^\Delta(x) = \max(V_n^\Delta(x), \text{cut}_n(x) - \Delta_n)$



RV-SDDP: First idea

$V_0^\Delta = 0$

$V_{n+1}^\Delta(x) = \max(V_n^\Delta(x), \text{cut}_n(x) - \Delta_n)$



RV-SDDP: Final idea

Naive update

$$V_0^\Delta = 0$$
$$V_{n+1}^\Delta(x) = \max(V_n^\Delta(x), \mathrm{cut}_n(x) - \Delta_n)$$

Improved update

$$V_0^\Delta = 0$$
$$V_{n+1}^\Delta(x) = \max(V_n^\Delta(x), \max_{1\le j\le n} \mathrm{cut}_j(x) - \Delta_n)$$

Equivalent closed-form expression

$$V_n^\Delta(x) = \max\left\{0, \max_{1\le j\le n} \left(\mathrm{cut}_j(x) - \Delta_j^n\right)\right\}$$
$$\Delta_j^n = \min_{j\le \ell\le n} \Delta_\ell, \qquad \forall j = 1,\dots,n.$$

Interpretation

At iteration n, we *add a new cut* cut_n and we *update the shifts* Δ_j^n for all the previous cuts.

RV-SDDP theorem

$$V_0^\Delta = 0$$
$$V_{n+1}^\Delta = \max(V_n^\Delta, \max_{1 \le j \le n} \text{cut}_j(x) - \Delta_n) \qquad \text{converges to } V_\infty^\Delta$$

Theorem

Let $(\Delta_n)_{n \in \mathbb{N}}$ be a sequence of non-negative shift values. If

$$\delta = \liminf_{n \to \infty} \Delta_n \le \min_{y \in X} \left\{ \mathcal{B}(V_\infty^\Delta)(y) - V_\infty^\Delta(y) \right\},$$

then the function V_∞^Δ satisfies: $\mathcal{B}(V_\infty^\Delta) = V_\infty^\Delta + \delta$.

RV-SDDP: possible shifts

RV-SDDP update

$$V_0^\Delta = 0, \qquad V_{n+1}^\Delta(x) = \max\left(V_n^\Delta(x), \max_{1\le j\le n} \mathrm{cut}_j(x) - \Delta_n\right).$$

Convergence condition

$$\delta := \liminf_{n\to\infty} \Delta_n \le \min_{y\in X} \left\{\mathcal{B}(V_\infty^\Delta)(y) - V_\infty^\Delta(y)\right\}.$$

Possible choices for the shift Δ_n:

1. $\Delta_n = 0$ for all $n \to V_{n+1}^\Delta = \max(V_n^\Delta, \mathrm{cut}_n)$: SDDP algorithm. ✓
2. $\Delta_n = \min_{x\in X} \left\{\mathcal{B}(V_n^\Delta)(x) - V_n^\Delta(x)\right\}$. ✓
3. $\Delta_n = \mathcal{B}(V_n^\Delta)(y) - V_n^\Delta(y)$, for some random $y \in X$. ✓

RV-SDDP: possible shifts

RV-SDDP update

$$V_0^\Delta = 0, \qquad V_{n+1}^\Delta(x) = \max\left(V_n^\Delta(x), \max_{1\le j\le n} \mathrm{cut}_j(x) - \Delta_n\right).$$

Convergence condition

$$\delta := \liminf_{n\to\infty} \Delta_n \le \min_{y\in X} \left\{\mathcal{B}(V_\infty^\Delta)(y) - V_\infty^\Delta(y)\right\}.$$

Possible choices for the shift Δ_n:

1. $\Delta_n = 0$ for all $n \to V_{n+1}^\Delta = \max(V_n^\Delta, \mathrm{cut}_n)$: SDDP algorithm. ✓
2. $\Delta_n = \min_{x\in X} \left\{\mathcal{B}(V_n^\Delta)(x) - V_n^\Delta(x)\right\}$. ✓
3. $\Delta_n = \mathcal{B}(V_n^\Delta)(y) - V_n^\Delta(y)$, for some random $y \in X$. ✓

RV-SDDP algorithm

Algorithm 3: RV-SDDP

Initialize value function $V_0(x)$
$n = 0$;
for $k = 1 : K$ **do**
  Choose forward pass length T_k
  for $t = 1 : T_k$ **do** // forward pass
    Simulate the policy using V_n to generate trial points $x_t^k \in X$
  for $t = T_k : 1$ **do** // backward pass
    Compute cut_n, a cut to $\mathcal{B}(V_n)$ at x_t^k
    Sample $y_n \in X$ uniformly and define $\Delta_n = \mathcal{B}(V_n)(y_n) - V_n(y_n)$
    Update shift values: $\Delta_n^n = \Delta_n$ and $\Delta_j^n = \min(\Delta_n, \Delta_j^{n-1})$ for $j = 1, \ldots, n-1$
    Update: $V_{n+1} = \max\left\{V_0, \max_{1 \le j \le n}\left(\text{cut}_j - \Delta_j^n\right)\right\}$
    $n = n + 1$

Outline

System Description

A Model of the Brazilian Hydro-Electric Grid

- A benchmark commonly used to compare the performance of different algorithms in the literature
- 4-dimensional state space
- 150-dimensional action space
- Periodic with period 12
- 82 possible water inflow scenarios each month, corresponding to historical records
- Historically $\beta = 0.99$

Two criteria:

1. *Cost of the policy* evaluated on a test set.
2. *Number of active cuts*: cuts that are not dominated by others.

Policy cost for different values of β

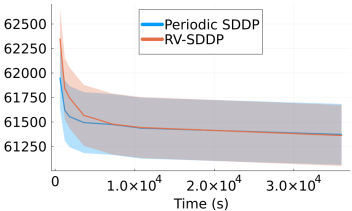


Figure: $\beta = 0.8$

- Timehorizon: 36
- Dataset size: 50000 scenarios

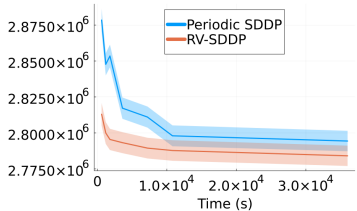


Figure: $\beta = 0.99$

- Timehorizon: 696
- Dataset size: 10000 scenarios

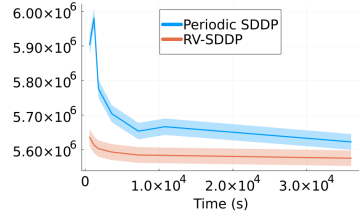
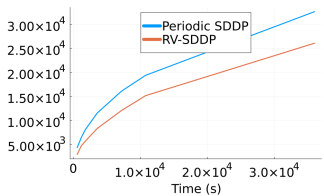


Figure: $\beta = 0.995$

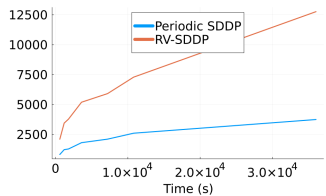
- Timehorizon: 1392
- Dataset size: 2000 scenarios

Number of active cuts for different values of β

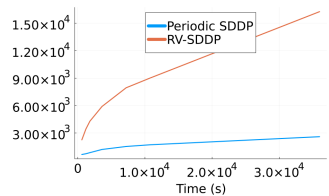
Number of active cuts: cuts that are not dominated by others.



(a) $\beta = 0.8$



(b) $\beta = 0.99$



(c) $\beta = 0.995$

Outline

Conclusion

- **Main idea:** Infinite-horizon SDDP suffers when the discount factor β is close to 1. We adapt Relative Value Iteration to infinite-horizon SDDP by shifting cuts.
- **Numerical evidence (Brazilian hydro benchmark):** RV-SDDP outperforms SDDP when the discount factor β is close to 1 and appears almost insensitive to β. 3 times more active cuts when $\beta = 0.99$.
- **Works not presented:** Proof of convergence for $\beta < 1$.
- **Future work:** Proof of convergence for $\beta = 1$.

Bibliographie I

- Arapostathis, Ari and Vivek S Borkar (2019). "Average cost optimal control under weak ergodicity hypotheses: Relative value iterations". In: *arXiv:1902.01048*.
- Pereira, Mario VF and Leontina MVG Pinto (1991). "Multi-stage stochastic optimization applied to energy planning". In: *Mathematical programming* 52, pp. 359–375.
- Shapiro, Alexander and Yi Cheng (2023). "Dual bounds for periodical stochastic programs". In: *Operations Research* 71.1, pp. 120–128.
- Shapiro, Alexander and Lingquan Ding (2020). "Periodical multistage stochastic programs". In: *SIAM Journal on Optimization* 30.3, pp. 2083–2102.
- White, Douglas J (1963). "Dynamic programming, Markov chains, and the method of successive approximations". In: *Journal of Mathematical Analysis and Applications* 6.3, pp. 373–376.