A Comparison of Sample-based Stochastic Optimal Control Methods for Power System Management

ISMP'09

Pierre Carpentier[2], Guy Cohen[3], Pierre Girardeau[1,2,3]

[1]EDF R&D, France

[2]ENSTA-ParisTech, France

[3]École des Ponts-ParisTech, France

August 24, 2009

Outline

Plan

1. Introduction
   - A mid-term power management problem
   - Two classical techniques
2. Scenario tree methods
   - Presentation
   - Estimating error
3. Particle methods
   - Main ideas
   - Example

Motivation

Goal

We are interested in solving **large-scale stochastic optimal control** problems:

- "Optimize the behaviour" of some controlled dynamical system driven by observed random variables.

An example

A power producer wants to find strategies over units that:

- **minimize its production cost** over all units,
- while satisfying power **demand constraints** and **physical constraints** over units.
- Demand, water inflows, breakdowns are **random variables**.
- **Strategies** are based on past **observations**.

Motivation

Goal

We are interested in solving **large-scale stochastic optimal control** problems:

- "Optimize the behaviour" of some controlled dynamical system driven by observed random variables.

An example

A power producer wants to find strategies over units that:

- **minimize its production cost** over all units,
- while satisfying power **demand constraints** and **physical constraints** over units.
- Demand, water inflows, breakdowns are **random variables**.
- **Strategies** are based on past **observations**.

Motivation

Goal

We are interested in solving **large-scale stochastic optimal control** problems:

- "Optimize the behaviour" of some controlled dynamical system driven by observed random variables.

Difficulties

- Large amount of units and time steps.
- Random variables are involved.
- Decisions have to be made sequentially and are based on available information:
  - $\hookrightarrow$ **Optimization variables are functions depending on observation of past noises**.

Classical techniques

Dynamic Programming [Bellman, 1957, Bertsekas, 2000]

- DP provides a way to obtain **strategies depending on the state** of the system.
- But it suffers from the curse of dimensionality:
  $\hookrightarrow$ **complexity grows exponentially** w.r.t. the state dimension

Classical techniques

Dynamic Programming [Bellman, 1957, Bertsekas, 2000]

- DP provides a way to obtain **strategies depending on the state** of the system.
- But it suffers from the curse of dimensionality:
  $\hookrightarrow$ **complexity grows exponentially** w.r.t. the state dimension

Classical techniques

We here **focus on scenario-based techniques**.

Stochastic Programming

- SP does not require to map the whole state space.
- But the **computational burden grows exponentially with the time horizon** [Shapiro, 2006].

We here further analyse this negative feature and propose leads to overcome this difficulty.

Notations

We consider a discrete finite time horizon $\{0, \dots, T\}$. Three types of random variables on $(\Omega, \mathcal{A}, \mathbb{P})$ are involved :

- A state $\mathbf{X}_t$ associated with the system ;
- A control $\mathbf{U}_t$ with which the system can be managed ;
- A noise $\mathbf{W}_t$ influencing the system.

We suppose that random variables lie in $L^2(\Omega, \mathcal{A}, \mathbb{P})$.

The problem

$$\min_{\mathbf{X},\mathbf{U}} \quad \mathbb{E}\left(\sum_{t=0}^{T-1} L_t\left(\mathbf{X}_t,\mathbf{U}_t,\mathbf{W}_{t+1}\right) + K\left(\mathbf{X}_T\right)\right),$$

$$\text{s.t.} \quad \mathbf{X}_{t+1} = f_t\left(\mathbf{X}_t,\mathbf{U}_t,\mathbf{W}_{t+1}\right),$$

$$\mathbf{U}_t \preceq \left(\mathbf{W}_0,\dots,\mathbf{W}_t\right).$$

In the case of a power system

- L_t is the production cost over power units at time t, depending on stocks ($\mathbf{X}_t$), controls applied ($\mathbf{U}_t$) and water inflows or breakdowns ($\mathbf{W}_{t+1}$).
- K is valueing the remaining stock $\mathbf{X}_T$ at the end of the period.
- f_t represents the dynamics of the stocks.

The problem

$$
\begin{aligned}
\min_{\mathbf{X},\mathbf{U}} \quad & \mathbb{E}\left(\sum_{t=0}^{T-1} L_t\left(\mathbf{X}_t, \mathbf{U}_t, \mathbf{W}_{t+1}\right) + K\left(\mathbf{X}_T\right)\right), \\
\text{s.t.} \quad & \mathbf{X}_{t+1} = f_t\left(\mathbf{X}_t, \mathbf{U}_t, \mathbf{W}_{t+1}\right), \\
& \mathbf{U}_t \preceq \left(\mathbf{W}_0, \dots, \mathbf{W}_t\right).
\end{aligned}
$$

What is expected?

We are looking for strategies $\widehat{\mathbf{U}}_t$:

$$
\left(\mathbf{W}_0, \dots, \mathbf{W}_t\right) \longrightarrow \widehat{\mathbf{U}}_t = \widehat{\gamma}_t\left(\mathbf{W}_0, \dots, \mathbf{W}_t\right)
$$

or, within the Dynamic Programming framework (stagewise independence assumption for noises $\mathbf{W}_t$) :

$$
\mathbf{X}_t \longrightarrow \widehat{\mathbf{U}}_t = \widehat{\gamma}_t\left(\mathbf{X}_t\right).
$$

The problem

$$
\begin{aligned}
\min_{\mathbf{X},\mathbf{U}} \quad & \mathbb{E}\left(\sum_{t=0}^{T-1} L_t\left(\mathbf{X}_t, \mathbf{U}_t, \mathbf{W}_{t+1}\right) + K\left(\mathbf{X}_T\right)\right), \\
\text{s.t.} \quad & \mathbf{X}_{t+1} = f_t\left(\mathbf{X}_t, \mathbf{U}_t, \mathbf{W}_{t+1}\right), \\
& \mathbf{U}_t \preceq \left(\mathbf{W}_0, \dots, \mathbf{W}_t\right).
\end{aligned}
$$

The case of stochastic methods

Note that for stochastic methods, the derived **strategies depend on some drawings** ξ made along with the algorithm:

$$
\mathbf{X}_t \longrightarrow \widehat{\mathbf{U}}_t = \widehat{\gamma}_t^{\xi}\left(\mathbf{X}_t\right).
$$

For example: ξ may be the drawing of a scenario tree.

The problem

$$\min_{\mathbf{X},\mathbf{U}} \quad \mathbb{E}\left(\sum_{t=0}^{T-1} L_t\left(\mathbf{X}_t, \mathbf{U}_t, \mathbf{W}_{t+1}\right) + K\left(\mathbf{X}_T\right)\right),$$
$$\text{s.t.} \quad \mathbf{X}_{t+1} = f_t\left(\mathbf{X}_t, \mathbf{U}_t, \mathbf{W}_{t+1}\right),$$
$$\mathbf{U}_t \preceq \left(\mathbf{W}_0, \ldots, \mathbf{W}_t\right).$$

Mean Squared Error for the strategy

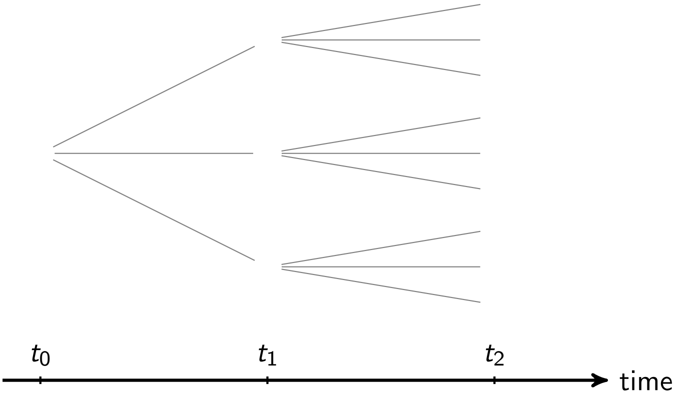
We then evaluate the error by computing the **distance to the true optimal strategy $\mathbf{U}^* = \gamma^*\left(\mathbf{X}\right)$**:

$$\text{IMSE} = \mathbb{E}_{\boldsymbol{\xi}}\left(\left\|\widehat{\gamma}^{\boldsymbol{\xi}}\left(\cdot\right) - \gamma^*\left(\cdot\right)\right\|^2_{L^2(\mathbb{X})}\right).$$

Plan

Scenario trees

We discretize the information structure using a scenario tree :



Scenario trees

Principle

- **Discretize randomness** using a scenario tree to represent the non-anticipativity constraint.
- Use deterministic optimization techniques to **solve the discretized problem**.
- If strategies (functions) are needed:
  - either use an **interpolation technique to obtain controls outside nodes**,
  - or use a rolling horizon algorithm (time consuming).

- There is a large literature on **how to build a scenario tree** from scenarios [Pflug, 2001, Heitsch and Römisch, 2009].
- Here we do not insist on this point and suppose that at each node, n branches start, so that **there are** $N = n^T$ **leaves to the tree**.

Scenario trees

Principle

- **Discretize randomness** using a scenario tree to represent the non-anticipativity constraint.
- Use deterministic optimization techniques to **solve the discretized problem**.
- If strategies (functions) are needed:
  - either use an **interpolation technique to obtain controls outside nodes**,
  - or use a rolling horizon algorithm (time consuming).

- There is a large literature on **how to build a scenario tree** from scenarios [Pflug, 2001, Heitsch and Römisch, 2009].
- Here we do not insist on this point and suppose that at each node, n branches start, so that **there are** $N = n^T$ **leaves to the tree**.

Back to our multi-stage problem

The problem

$$
\begin{aligned}
\min_{\mathbf{X},\mathbf{U}} \quad & \mathbb{E}\left(\sum_{t=0}^{T-1} L_t\left(\mathbf{X}_t, \mathbf{U}_t, \mathbf{W}_{t+1}\right) + K\left(\mathbf{X}_T\right)\right), \\
\text{s.c.} \quad & \mathbf{X}_{t+1} = f_t\left(\mathbf{X}_t, \mathbf{U}_t, \mathbf{W}_{t+1}\right), \\
& \mathbf{U}_t \preceq \mathbf{W}_0, \ldots, \mathbf{W}_t.
\end{aligned}
$$

Integrated Mean Squared Error

- On this problem we apply SP and obtain a strategy $\widehat{\gamma}^{\boldsymbol{\xi}}$, depending on a random variable $\boldsymbol{\xi}$ (the scenarios drawn for building the tree).
- We would like to evaluate the distance between $\widehat{\gamma}^{\boldsymbol{\xi}}$ and the optimum γ^*:

IMSE for the strategy

$$\mathsf{IMSE} = \mathbb{E}_{\boldsymbol{\xi}}\left(\left\|\widehat{\gamma}^{\boldsymbol{\xi}}(\cdot) - \gamma^*(\cdot)\right\|^2_{L^2(\mathbb{X})}\right).$$

Integrated Mean Squared Error

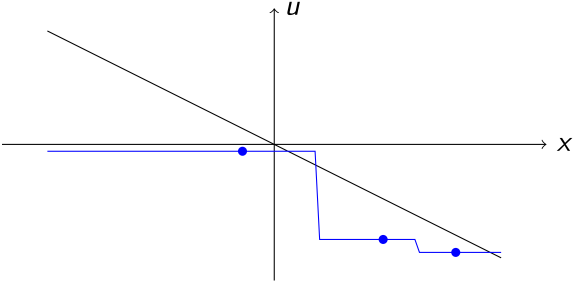
- On this problem we apply SP and obtain a strategy $\widehat{\gamma}^{\boldsymbol{\xi}}$, depending on a random variable $\boldsymbol{\xi}$ (the scenarios drawn for building the tree).
- We would like to evaluate the distance between $\widehat{\gamma}^{\boldsymbol{\xi}}$ and the optimum γ^*:

IMSE for the strategy

$$\mathrm{IMSE} = \mathbb{E}_{\boldsymbol{\xi}}\left(\left\|\widehat{\gamma}^{\boldsymbol{\xi}}(\cdot) - \mathbb{E}_{\boldsymbol{\xi}}\left(\widehat{\gamma}^{\boldsymbol{\xi}}(\cdot)\right)\right\|^2_{L^2(\mathbb{X})}\right) \rightarrow \text{ variance}$$

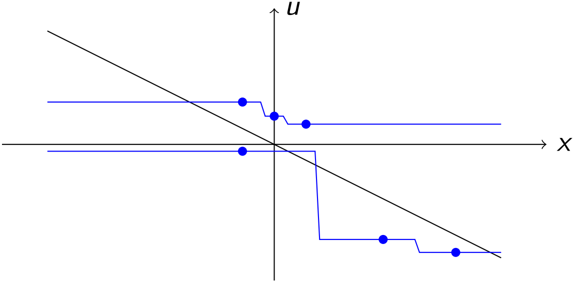
$$+ \left\|\gamma^*(\cdot) - \mathbb{E}_{\boldsymbol{\xi}}\left(\widehat{\gamma}^{\boldsymbol{\xi}}(\cdot)\right)\right\|^2_{L^2(\mathbb{X})} \rightarrow \text{ bias}$$

Bias and variance on an example



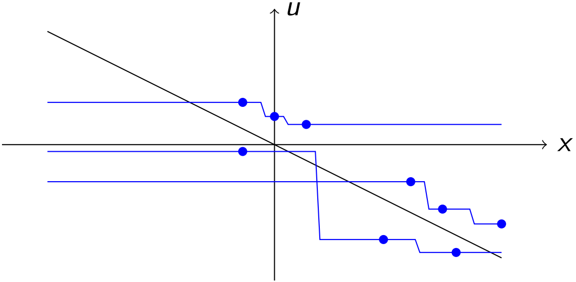
- Black: optimum. Blue: sample strategy (dots are tree nodes).
- Here the regression operator is just a **nearest neighbour** operator, so that **strategies are piecewise constant**.
- Red: average strategy.
- However the **average strategy seems to be smooth**.

Bias and variance on an example



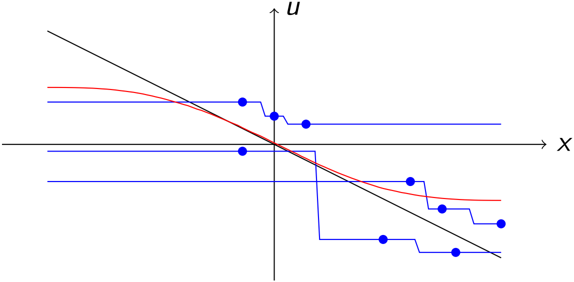
- Black: optimum. Blue: sample strategy (dots are tree nodes).
- Here the regression operator is just a **nearest neighbour** operator, so that **strategies are piecewise constant**.
- Red: average strategy.
- However the **average strategy seems to be smooth**.

Bias and variance on an example



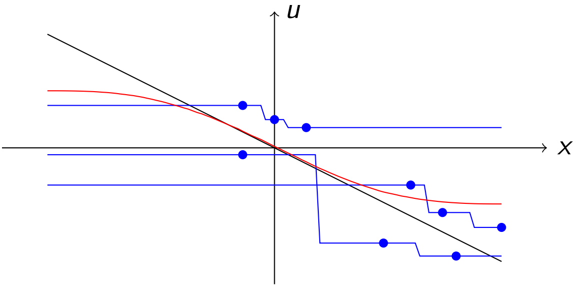
- Black: optimum. Blue: sample strategy (dots are tree nodes).
- Here the regression operator is just a **nearest neighbour** operator, so that **strategies are piecewise constant**.
- Red: average strategy.
- However the **average strategy seems to be smooth**.

Bias and variance on an example



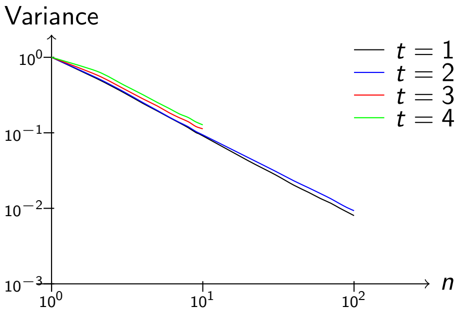
- Black: optimum. Blue: sample strategy (dots are tree nodes).
- Here the regression operator is just a **nearest neighbour** operator, so that **strategies are piecewise constant**.
- Red: average strategy.
- However the **average strategy seems to be smooth**.

Bias and variance on an example



- **Bias** is the distance between the optimal strategy and the averaged approximate strategy.
- **Variance** is the mean distance between approximate strategies and the averaged approximate strategy.

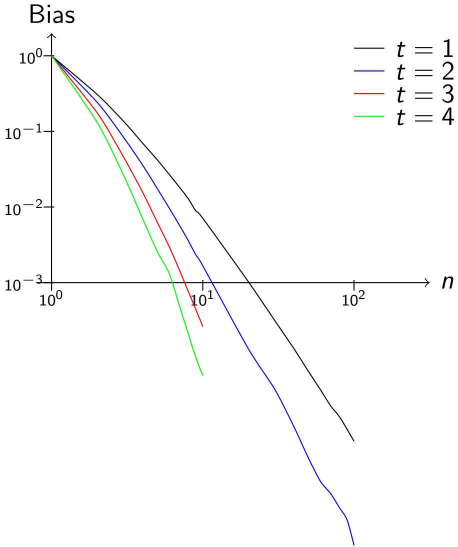
4-stage example: curve of variance



Variance order associated with the 4 SP strategies:

- does not depend on time.
- decreases like n^{-1}.

4-stage example: curve of bias



Conclusion on scenario trees complexity

- Obviously, variance dominates in the IMSE.
  - ↪ But results **depend on the state dimension** and this dominance relation could change.

The error of the method does not depend on the number of scenarios used (the leaves of the tree), but **on the branching factor**.

↪ This confirms Shapiro's result from another viewpoint:

- We here work on the **distance between optimal and SP strategies**.
- Shapiro's result is concerned about the probability that the **distance between optimal and SP costs** is smaller than some quantity.

Conclusion on scenario trees complexity

- Obviously, variance dominates in the IMSE.
  ↪ But results **depend on the state dimension** and this dominance relation could change.

The error of the method does not depend on the number of scenarios used (the leaves of the tree), but **on the branching factor**.

↪ This confirms Shapiro's result from another viewpoint:

- We here work on the **distance between optimal and SP strategies**.
- Shapiro's result is concerned about the probability that the **distance between optimal and SP costs** is smaller than some quantity.

Conclusion on scenario trees complexity

- Obviously, variance dominates in the IMSE.
  ↪ But results **depend on the state dimension** and this dominance relation could change.

The error of the method does not depend on the number of scenarios used (the leaves of the tree), but **on the branching factor**.

↪ This confirms Shapiro's result from another viewpoint:

- We here work on the **distance between optimal and SP strategies**.
- Shapiro's result is concerned about the probability that the **distance between optimal and SP costs** is smaller than some quantity.

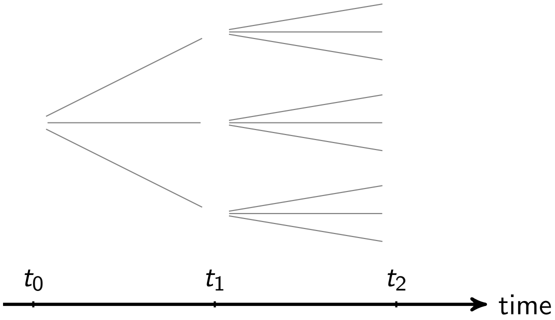
Plan

1. Introduction
   - A mid-term power management problem
   - Two classical techniques
2. Scenario tree methods
   - Presentation
   - Estimating error
3. Particle methods
   - Main ideas
   - Example

Complexity of scenario trees

Where does this bad rate come from?

The convergence rate is in $\frac{1}{n}$, because for each node we only use n samples to compute the conditional expectations :



But the branching factor n is often much smaller than the number of scenarios N.

How to perform better?
Some leads

- The problem comes from the structure of the tree.
  - ↪ We are looking for functions :

$$\gamma : \mathbf{W}_0, \ldots, \mathbf{W}_t \longrightarrow \mathbf{U}_t .$$

- **Particle methods** try to overcome this difficulty by:
  - using DP ideas: **define a state** so that there is **stagewise independence between noises $\mathbf{W}_t$**.
  - but do not compute the strategy on every state like in classical DP:
    - ↪ **adaptive mesh method based on scenarios**.

How to perform better?

Some leads

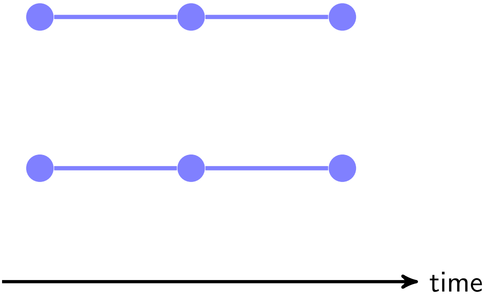
- The problem comes from the structure of the tree.
  ↪ We are looking for functions :

$$\gamma : \mathbf{W}_0, \ldots, \mathbf{W}_t \longrightarrow \mathbf{U}_t .$$

- **Particle methods** try to overcome this difficulty by:
  - using DP ideas: **define a state** so that there is **stagewise independence between noises $\mathbf{W}_t$**.
  - but do not compute the strategy on every state like in classical DP:

    ↪ **adaptive mesh method based on scenarios**.

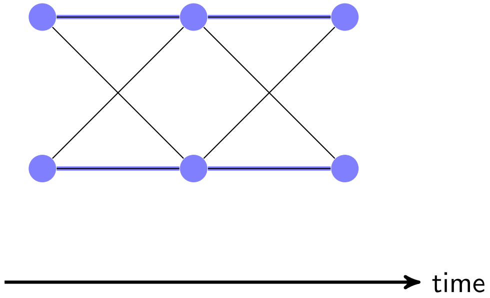
Scheme



- Suppose we have **two scenarios**.

(To learn more, see [Carpentier et al., 2009, Dallagi et al., 2007].)

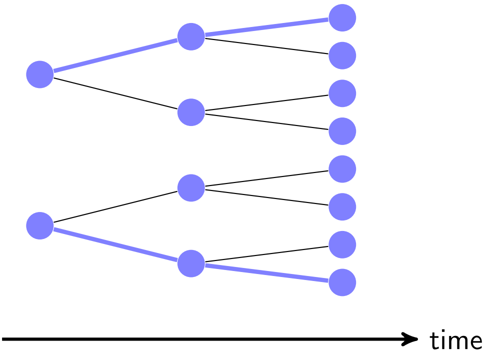
Scheme



- **Stagewise independence assumption** allows us to mix scenarios...

(To learn more, see [Carpentier et al., 2009, Dallagi et al., 2007].)

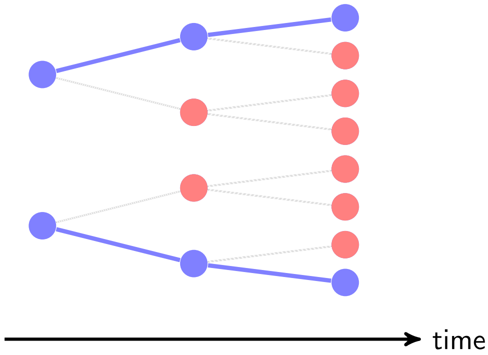
Scheme



- ...and build the “full” scenario tree, on which we write **optimality conditions** for the problem.

(To learn more, see [Carpentier et al., 2009, Dallagi et al., 2007].)

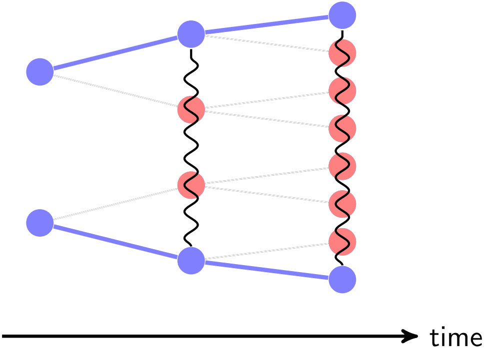
Scheme



- But computing the state and control values on each node would be too heavy.

(To learn more, see [Carpentier et al., 2009, Dallagi et al., 2007].)

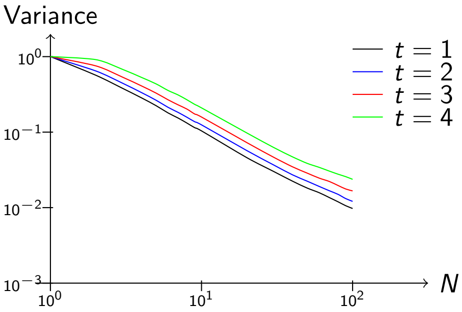
Scheme



- To solve optimality conditions, we **approximate the values of state and control**, when necessary, by using **regression operators**.

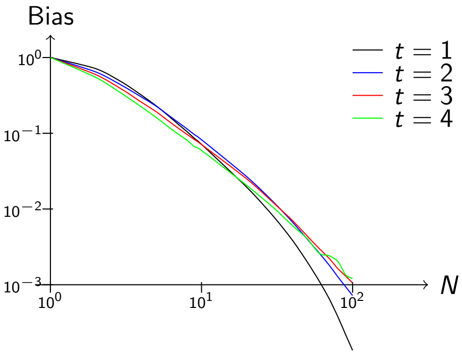
(To learn more, see [Carpentier et al., 2009, Dallagi et al., 2007].)

4-stage example: curve of variance for PM



- Variance still does not depend on time.
- It decreases like N^{-1}, N **being the number of scenarios**!

4-stage example: curve of bias for PM



Bias is still small compared to the variance.

Conclusion

Error estimates on scenario tree methods

- We evaluate the distance between optimal strategies and strategies given by scenario trees on a simple example.
- Variance dominates in the IMSE, and **the order is n^{-1}, n being the branching factor**.

Leads to overcome these difficulties

- **Adaptive mesh method** using scenarios.
- On simple examples we observe that **IMSE decreases like N^{-1}, N being the number of scenarios** ($N >> n$).

Conclusion

Error estimates on scenario tree methods

- We evaluate the distance between optimal strategies and strategies given by scenario trees on a simple example.
- Variance dominates in the IMSE, and **the order is n^{-1}, n being the branching factor**.

Leads to overcome these difficulties

- **Adaptive mesh method** using scenarios.
- On simple examples we observe that **IMSE decreases like N^{-1}, N being the number of scenarios** ($N >> n$).

Future works

$\longrightarrow$ Obtain **convergence results in more general cases**.

$\longrightarrow$ How does PM perform **when state dimension grows**?

- In order to satisfy the stagewise independence condition, we may have to increase the state dimension.

$\longrightarrow$ Work on the **numerical solving of the optimality conditions**:

- Regression operators introduce non-linearities so that the discretized problem is more difficult to solve.

References

Bellman, R. (1957).
Dynamic Programming.
Princeton University Press, New Jersey.

Bertsekas, D. (2000).
Dynamic Programming and Optimal Control, volume 1 & 2.
Athena Scientific, 2 edition.

Carpentier, P., Cohen, G., and Dallagi, A. (2009).
Particle methods for stochastic optimal control problems.
arXiv:0907.4663v1.

Dallagi, A., Carpentier, P., Cohen, G., and SOWG (2007).
Particle methods on stochastic optimal control: an alternative to stochastic programming and dynamic programming.
EURO'07 conference, "http://www.ualberta.ca/~dallagi/PresentationEURO2007.pdf".

Heitsch, H. and Römisch, W. (2009).
Scenario tree modelling for multistage stochastic programs.
Mathematical Programming, 118:371–406.

Pflug, G. (2001).
Scenario tree generation for multiperiod financial optimization by optimal discretization.
Math. Program., (89):251–271.

Shapiro, A. (2006).
On complexity of multistage stochastic programs.
Operations Research Letters, 34:1–8.