Worst-case-expectation approach to optimization under uncertainty

Wajdi Tekaya®
Joint research with
Alexander Shapiro*, Murilo Pereira Soares# and Joari Paulo da Costa#
®: Cambridge Systems Associates,*: Georgia Tech;#: ONS, Brazil

ICSP’13
Bergamo, ITALY July 8-12, 2013

July 10, 2013
Outline

1 Motivation

2 Risk neutral approach
   - Risk neutral formulation
   - Methodology
   - SDDP algorithm

3 Worst-case-expectation approach
   - Worst-case-expectation formulation
   - The (WCE) algorithm
   - Analysis of the worst-case-expectation policy

4 Conclusion
Motivation: Hydrothermal operation planning problem
Motivation: Hydrothermal operation planning problem

Demand constraints:

\[ Z_t + Y_t + F_t = D_t \]

- \( Z_t \): turbined volume
- \( Y_t \): thermal generation
- \( F_t \): energy exchange
- \( D_t \): load

Involves costs \( c > 0 \)
Motivation: Hydrothermal operation planning problem

Demand constraints:

$$t_{\text{urbined}} + Y_t + F_t = D_t$$

incurs costs $c > 0$
Motivation: Hydrothermal operation planning problem

Demand constraints:

\[ Z_t + Y_t + F_t = D_t \]

incurs costs \( c > 0 \)

Balance equation:

\[ V_{t+1} + Z_t + S_t = V_t + \xi_t \]
Motivation: Hydrothermal operation planning problem

The purpose 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.

\[
Q_t(V_t, \xi_t) = \min \left\{ c^T [Y_t, F_t] + \mathbb{E}[Q_{t+1}(V_{t+1}, \xi_{t+1})] \right\}
\]

s.t.

\[
\begin{align*}
Z_t &+ Y_t + F_t = D_t \quad \text{(Demand constraints)} \\
V_{t+1} &+ Z_t + S_t = V_t + \xi_t \quad \text{(Balance equation)}
\end{align*}
\]

\[
0 \leq Z_t \leq \bar{Z}_t, \quad 0 \leq V_{t+1} \leq \bar{V}_t, \quad \underline{Y}_t \leq Y_t \leq \bar{Y}_t, \quad 0 \leq F_t \leq \bar{F}_t \quad \text{(Capacity constraints)}
\]
Outline

1 Motivation

2 Risk neutral approach
   - Risk neutral formulation
   - Methodology
   - SDDP algorithm

3 Worst-case-expectation approach
   - Worst-case-expectation formulation
   - The (WCE) algorithm
   - Analysis of the worst-case-expectation policy

4 Conclusion
Risk neutral formulation

Nested formulation:

\[
\begin{bmatrix}
\min_{A_1 x_1 = b_1 \atop x_1 \geq 0} c_1^T x_1 + E_{|\xi_1} \\
\min_{B_2 x_1 + A_2 x_2 = b_2 \atop x_2 \geq 0} c_2^T x_2 + E_{|\xi_2} \\
\cdots + E_{|\xi_{[T-1]}} \\
\min_{B_T x_T - 1 + A_T x_T = b_T \atop x_T \geq 0} c_T^T x_T
\end{bmatrix}
\]

Components of vectors \( c_t, b_t \) and matrices \( A_t, B_t \) are modelled as random variables forming the stochastic data process \( \xi_t = (c_t, A_t, B_t, b_t) \), \( t = 2, ..., T \), with \( \xi_1 = (c_1, A_1, b_1) \) being deterministic.
Risk neutral formulation

Nested formulation:

\[
\begin{aligned}
\text{Min}_{A_1 x_1 = b_1, x_1 \geq 0} & \quad c_1^T x_1 + \mathbb{E}_{\xi_1} \left[ \text{Min}_{B_2 x_1 + A_2 x_2 = b_2, x_2 \geq 0} c_2^T x_2 + \mathbb{E}_{\xi_2} \left[ \cdots + \mathbb{E}_{\xi_{T-1}} \left[ \text{Min}_{B_T x_{T-1} + A_T x_T = b_T, x_T \geq 0} c_T^T x_T \right] \right] \right] \right]
\end{aligned}
\]

Components of vectors \(c_t, b_t\) and matrices \(A_t, B_t\) are modelled as random variables forming the stochastic data process \(\xi_t = (c_t, A_t, B_t, b_t)\), \(t = 2, ..., T\), with \(\xi_1 = (c_1, A_1, b_1)\) being deterministic.

\(\star\) Key assumption: stagewise independence (vector \(\xi_{t+1}\) is independent of \(\xi_{[t]} = (\xi_1, ..., \xi_t)\) for \(t = 1, ..., T - 1\).)
Risk neutral formulation

Nested formulation:

\[
\begin{align*}
\min_{A_1 x_1 = b_1, \quad x_1 \geq 0} &\quad c_1^\top x_1 + \mathbb{E}_{\xi_1} \\
\min_{B_2 x_1 + A_2 x_2 = b_2, \quad x_2 \geq 0} &\quad c_2^\top x_2 + \mathbb{E}_{\xi_2} \\
&\quad \cdots + \mathbb{E}_{\xi_{[T-1]}} \\
\min_{B_T x_{T-1} + A_T x_T = b_T, \quad x_T \geq 0} &\quad c_T^\top x_T
\end{align*}
\]

Components of vectors \(c_t, b_t\) and matrices \(A_t, B_t\) are modelled as random variables forming the stochastic data process \(\xi_t = (c_t, A_t, B_t, b_t), \quad t = 2, \ldots, T\), with \(\xi_1 = (c_1, A_1, b_1)\) being deterministic.

\(\star\) Key assumption: stagewise independence (vector \(\xi_{t+1}\) is independent of \(\xi_{[t]} = (\xi_1, \ldots, \xi_t)\) for \(t = 1, \ldots, T - 1\).)

Dynamic Programming equations:

\[
Q_t(x_{t-1}, \xi_t) = \min_{x_t \in \mathbb{R}^{n_t}} \left\{ c_t^\top x_t + Q_{t+1}(x_t) : B_t x_{t-1} + A_t x_t = b_t, x_t \geq 0 \right\}
\]

where: \(Q_{t+1}(x_t) = \mathbb{E} [Q_{t+1}(x_t, \xi_{t+1})]\) for \(2 \leq t \leq T\).

such that: \(Q_{T+1}(.) \equiv 0\) and \(B_1 = 0\).
Methodology

Three levels of approximations:

1. Modelling: we assume we have a “true” problem to solve.

2. The true problem is approximated by the so-called Sample Average Approximation (SAA) problem: a sample $\tilde{\xi}_1^1, ..., \tilde{\xi}_N^{N_t}$, of size $N_t$, from the distribution of the random vector $\xi_t$, $t = 1, ..., T$, is generated. These samples generate a scenarios tree with the total number of scenarios $N = \prod_{t=1}^{T} N_t$, each with equal probability $1/N$.

3. Total number of scenarios $N$ quickly becomes astronomically large with increase of $T$: The SDDP method suggests a computationally tractable approach to solving SAA, and hence the “true” problem.
SDDP algorithm

- It is based on building piecewise linear outer approximations of the convex cost-to-go functions.
- The distinguishing feature of the SDDP approach is random sampling from the set of scenarios in the forward step of the algorithm.
SDDP algorithm

- It is based on building piecewise linear outer approximations of the convex cost-to-go functions.
- The distinguishing feature of the SDDP approach is random sampling from the set of scenarios in the forward step of the algorithm.
- Almost sure convergence of the SDDP algorithm was proved in Philpott and Guan (2008) under mild regularity conditions.
SDDP algorithm

- It is based on building piecewise linear outer approximations of the convex cost-to-go functions.
- The distinguishing feature of the SDDP approach is random sampling from the set of scenarios in the forward step of the algorithm.
- Almost sure convergence of the SDDP algorithm was proved in Philpott and Guan (2008) under mild regularity conditions.
- An analysis of the SDDP algorithm applied to two stage stochastic programming indicates that its computational complexity grows fast with increase of the number of state variables.
SDDP algorithm: Illustration

(a) Stage 1  
(b) Stage 2  
(c) Stage 3

Figure: “true” problem
SDDP algorithm: Illustration

(a) Stage 1
(b) Stage 2
(c) Stage 3

Figure: SAA
SDDP algorithm: Illustration

Figure: SDDP iteration 1: Forward step
SDDP algorithm: Illustration

Figure: SDDP iteration 1: Forward step
SDDP algorithm: Illustration

Figure: SDDP iteration 1: Forward step
SDDP algorithm: Illustration

(a) Stage 1
\[ \mathcal{V}_2(\cdot) \equiv 0 \]

(b) Stage 2
\[ \mathcal{V}_3(\cdot) \equiv 0 \]

(c) Stage 3
\[ \mathcal{V}_4(\cdot) \equiv 0 \]

\( x_1 \bullet \)

\( x_2 \bullet \)

\( x_3 \bullet \)

Figure: SDDP iteration 1: Forward step
SDDP algorithm: Illustration

Figure: SDDP iteration 1: Backward step
SDDP algorithm: Illustration

Figure: SDDP iteration 1: Backward step

(a) Stage 1

(b) Stage 2

(c) Stage 3
SDDP algorithm: Illustration

(a) Stage 1
Ω_2^0(·) \equiv 0

(b) Stage 2
Ω_3^1(·)

(c) Stage 3
Ω_4^1(·)

Figure: SDDP iteration 1: Backward step
Figure: SDDP iteration 1: Backward step
SDDP algorithm: Illustration

(a) Stage 1
(b) Stage 2
(c) Stage 3

Figure: SDDP iteration 2: Forward step
SDDP algorithm: Illustration

(a) Stage 1

(b) Stage 2

(c) Stage 3

Figure: SDDP iteration 2: Forward step
SDDP algorithm: Illustration

(a) Stage 1

(b) Stage 2

(c) Stage 3

Figure: SDDP iteration 2: Forward step
SDDP algorithm: Illustration

Figure: SDDP iteration 2: Backward step
SDDP algorithm: Illustration

Figure: SDDP iteration 2: Backward step
SDDP algorithm: Illustration

Figure: SDDP iteration 2: Backward step...
Outline

1 Motivation

2 Risk neutral approach
   - Risk neutral formulation
   - Methodology
   - SDDP algorithm

3 Worst-case-expectation approach
   - Worst-case-expectation formulation
   - The (WCE) algorithm
   - Analysis of the worst-case-expectation policy

4 Conclusion
Worst-case-expectation formulation

Suppose that $\xi_t = (\xi_1^t, \xi_2^t)$:

- uncertain parameters: $(\xi_2^1, ..., \xi_T^1) \in \Xi^1 = \Xi_2^1 \times \cdots \times \Xi_T^1$ compact
- random parameters: $\xi_2^2, ..., \xi_T^2$ with a specified probability distribution
Worst-case-expectation formulation

Suppose that $\xi_t = (\xi_t^1, \xi_t^2)$:

- uncertain parameters: $(\xi_2^1, ..., \xi_T^1) \in \Xi^1 = \Xi^1_2 \times \cdots \times \Xi^1_T$ compact
- random parameters: $\xi_2^2, ..., \xi_T^2$ with a specified probability distribution

Nested formulation:

\[
\begin{align*}
\min_{A_1 x_1 = b_1} c_1^T x_1 + \rho_2 |\xi_1 & \left[ \min_{B_2 x_1 + A_2 x_2 = b_2} c_2^T x_2 + \cdots + \rho_T |\xi_{[T-1]} \left[ \min_{B^T x_{[T-1]} + A^T x_T = b_T} c_T^T x_T \right] \right] \\
\end{align*}
\]

where $\rho_t |\xi_{[t-1]} [ \cdot ] = \sup_{(\xi_2^1, ..., \xi_T^1) \in \Xi^1} \mathbb{E}_{|\xi_{[t-1]} [ \cdot ]} \left[ \cdot \right]$, $t = 2, ..., T$. 

}\]
Worst-case-expectation formulation

Suppose that $\xi_t = (\xi^1_t, \xi^2_t)$:

- uncertain parameters: $(\xi^2_2, \ldots, \xi^2_T) \in \Xi^1 = \Xi^1_2 \times \cdots \times \Xi^1_T$ compact
- random parameters: $\xi^2_2, \ldots, \xi^2_T$ with a specified probability distribution

Nested formulation:

$$\begin{align*}
\min_{\substack{A_1 x_1 = b_1 \\
x_1 \geq 0}} & \quad c^T_1 x_1 + \rho_2|\xi_1 \\
\min_{\substack{B_2 x_1 + A_2 x_2 = b_2 \\
x_2 \geq 0}} & \quad c^T_2 x_2 + \cdots + \rho_T|\xi_{[T-1]} \\
\min_{x_T \geq 0} & \quad c^T_T x_T
\end{align*}$$

where $\rho_t|\xi_{[t-1]}[\cdot] = \sup_{(\xi^1_2, \ldots, \xi^1_T) \in \Xi^1} \mathbb{E}|\xi^2_{[t-1]}[\cdot]$, $t = 2, \ldots, T$.

Dynamic Programming equations:

$$Q_t(x_{t-1}, \xi^1_t, \xi^2_t) = \min_{B_t x_{t-1} + A_t x_t = b_t, x_t \geq 0} \left\{ c^T_t x_t + Q_{t+1}(x_t) \right\}$$

where: $Q_{t+1}(x_t) = \sup_{\xi^1_{t+1} \in \Xi^1_{t+1}} \mathbb{E} \left\{ Q_{t+1}(x_t, \xi^1_{t+1}, \xi^2_{t+1}) \right\}$ for $2 \leq t \leq T$.

such that: $Q_{T+1}(.) \equiv 0$ and $B_1 = 0$.

Key assumption: stagewise independence
The (WCE) algorithm: Backward step

At stage $t = T$ for a given $b^1_T$ and $b^T_{Tj} = (b^1_T, b^2_T)$, we solve $N$ problems

$$\min_{x_T \in \mathbb{R}^n_T} c_T x_T$$

$$b_T - 1 + A_T x_T = b_{Tj}, \quad x_T \geq 0, \quad j = 1, \ldots, N.$$

We have:

$$q_T(\bar{x}_T - 1, b^1_T) = \sum_{j=1}^N Q_T j(\bar{x}_T - 1, b^1_T)$$

Suppose that we can sample from sets $\{\Xi^1_T\}_{2 \leq t \leq T}$. Sample $L$ points $b^1_T \ell, \ell = 1, \ldots, L$, from $\Xi^1_T$ and compute subgradient (at $\bar{x}_T - 1$)

$$\gamma_T \ell = \sum_{j=1}^N \nabla Q_T j(\bar{x}_T - 1, b^1_T \ell).$$

Add the corresponding cutting planes $q_T(\bar{x}_T - 1, b^1_T \ell) + \gamma_T T \ell (x_T - 1 - \bar{x}_T - 1)$, $\ell = 1, \ldots, L$, to the collection of cutting planes of $Q_T(\cdot)$.
The (WCE) algorithm: Backward step

At stage $t = T$ for a given $b_T^1$ and $b_Tj = (b_T^1, b_T^2)$, we solve $N$ problems

$$\min_{x_T \in \mathbb{R}^{n_T}} c_T^T x_T \quad \text{s.t.} \quad B_T \bar{x}_{T-1} + A_T x_T = b_Tj, \quad x_T \geq 0, \quad j = 1, ..., N.$$ 

We have:

$$q_T(\bar{x}_{T-1}, b_T^1) = N^{-1} \sum_{j=1}^{N} Q_{Tj}(\bar{x}_{T-1}, b_T^1).$$

$$Q_T(\bar{x}_{T-1}) = \sup_{b_T^1 \in \Xi_T^1} q_T(\bar{x}_{T-1}, b_T^1)?$$
The (WCE) algorithm: Backward step

At stage $t = T$ for a given $b^1_T$ and $b^j_T = (b^1_T, b^2_T)$, we solve $N$ problems

$$\min_{x_T \in \mathbb{R}^{n_T}} c^T_T x_T \text{ s.t. } B_T \bar{x}_{T-1} + A_T x_T = b^j_T, \ x_T \geq 0, \ j = 1, \ldots, N.$$ 

We have: $q_T(\bar{x}_{T-1}, b^1_T) = N^{-1} \sum_{j=1}^N Q_j(\bar{x}_{T-1}, b^1_T)$.

$$Q_T(\bar{x}_{T-1}) = \sup_{b^1_T \in \Xi^1_T} q_T(\bar{x}_{T-1}, b^1_T)?$$

Suppose that we can sample from sets $\{\Xi^1_t\}_{2 \leq t \leq T}$.

- Sample $L$ points $b^1_T$, $\ell = 1, \ldots, L$, from $\Xi^1_T$ and compute subgradient (at $\bar{x}_{T-1}$)

$$\gamma_T^\ell = N^{-1} \sum_{j=1}^N \nabla Q_j(\bar{x}_{T-1}, b^1_T)$$

- Add the corresponding cutting planes

$$q_T(\bar{x}_{T-1}, b^1_T) + \gamma_T^\ell (x_{T-1} - \bar{x}_{T-1}), \ell = 1, \ldots, L,$$

to the collection of cutting planes of $\mathcal{Q}_T(\cdot)$. 

Wajdi Tekaya (CSA)
The (WCE) algorithm: Sampling from the uncertainty set

- We consider as uncertainty set $\Xi^1_t$, $t = 2, \ldots, T$, an ellipsoid, centered at $\bar{\xi}_t$:

$$
\Xi^1_t := \{\xi : (\xi - \bar{\xi}_t)^T A (\xi - \bar{\xi}_t) \leq r_t\}
$$

where $A$ is a positive definite matrix and $r_t > 0$. 

![Ellipsoid Diagram]
The (WCE) algorithm: Sampling from the uncertainty set

- We consider as uncertainty set $\Xi^1_t$, $t = 2, ..., T$, an ellipsoid, centered at $\bar{\xi}_t$:

$$\Xi^1_t := \{\xi : (\xi - \bar{\xi}_t)^T A (\xi - \bar{\xi}_t) \leq r_t\}$$

where $A$ is a positive definite matrix and $r_t > 0$.

- Dominating points in $\Xi^1_t$:

$$D = \{\xi \in \Xi^1_t : \text{does not exist } \xi' \in \Xi^1_t \text{ such that } \xi' \neq \xi \text{ and } \xi \leq \xi'\}$$

$$= \bigcup_{a \geq 0, \|a\| = 1} \arg \max \{a^T \xi : (\xi - \bar{\xi})^T A (\xi - \bar{\xi}) \leq r\}.$$ 

where $r = (\|\bar{\xi}^1\|_2 \times u)^2$, for a given $u > 0$, and $A = \Sigma^{-1}$. 
The (WCE) algorithm: Forward step (same as before)

\(\Omega_2(\cdot), \ldots, \Omega_T(\cdot)\) and a feasible solution \(\bar{x}_1\) ⇒ an implementable policy:

- For a realization \(\xi_t = (c_t, A_t, B_t, b_t), t = 2, \ldots, T\), decisions \(\bar{x}_t, t = 1, \ldots, T\), are computed recursively going forward with \(\bar{x}_1\) being the chosen feasible solution of the first stage problem, and \(\bar{x}_t\) being an optimal solution of

\[
\min_{x_t} c_t^T x_t + \Omega_{t+1}(x_t) \quad \text{s.t.} \quad A_t x_t = b_t - B_t \bar{x}_{t-1}, \quad x_t \geq 0, \quad t = 2, \ldots, T
\]

- These optimal solutions can be used as trial decisions in the backward step of the algorithm.
The (WCE) algorithm: Summary

In the numerical approach outlined above, the key steps are:

- Problem: Evaluating expectation, Approach: Construct the SAA problem.
- Problem: Computing maximum for the uncertain parameters, Approach: Sampling from the uncertainty sets.
Analysis of the worst-case-expectation policy
Analysis of the worst-case-expectation policy

Figure: 120 stages policy values for risk neutral and (WCE) ($u = 3\%$)
Analysis of the worst-case-expectation policy

Figure: Individual stage costs for risk neutral and (WCE) ($u = 3\%$)
Outline

1 Motivation

2 Risk neutral approach
   - Risk neutral formulation
   - Methodology
   - SDDP algorithm

3 Worst-case-expectation approach
   - Worst-case-expectation formulation
   - The (WCE) algorithm
   - Analysis of the worst-case-expectation policy

4 Conclusion
Conclusion

- A methodology that combines robust and stochastic programming approaches is suggested.
- A variant of the SDDP algorithm for solving this class of problems is suggested.
- Comparison with risk neutral approach:
  - The worst-case-expectation approach constructs a policy that is less sensitive to unexpected demand increase with a reasonable loss on average when compared to the risk neutral method.
- The computational experiments for our problem show that there is practically no increase in CPU time when compared to the risk neutral approach.
References

Thank you for your attention!