Decomposition of Multistage Stochastic Programs with Recombining Scenario Trees

Christian Küchler*, Stefan Vigerske*

Humboldt-Universität zu Berlin
Institut für Mathematik

Operations Research 2007
Universität des Saarlandes, 6. September 2007

*supported by BMBF and WWTF
Overview

Motivation and Introduction

Modified Benders Decomposition Algorithm

Conclusions
Scenario Trees

- **Multistage Stochastic Programming**: Representation of the underlying stochastic process through a scenario tree.
Scenario Trees

- **Multistage Stochastic Programming**: Representation of the underlying stochastic process through a scenario tree.
- **Difficulty**: Number of nodes can grow exponentially.
- **Tradeoff**: *Approximation quality against computational convenience.*
Scenario Trees

- **Multistage Stochastic Programming**: Representation of the underlying stochastic process through a scenario tree.
- **Difficulty**: Number of nodes can grow exponentially.
- **Tradeoff**: *Approximation quality* against *computational convenience*.
- **Recombining scenario trees?**
Recombining Scenario Trees

- Binomial Model (Cox, Ross, and Rubinstein, 1979).
- Recombining Trees in *Stochastic Programming*?
Recombining Scenario Trees

- Binomial Model (Cox, Ross, and Rubinstein, 1979).
- Recombining Trees in *Stochastic Programming*?
- Pro: *good approximation* of many processes of practical interest and linearly growing number of nodes.
Recombining Scenario Trees

- Binomial Model (Cox, Ross, and Rubinstein, 1979).
- Recombining Trees in *Stochastic Programming*?
- Pro: good approximation of many processes of practical interest and linearly growing number of nodes.
- Con: time-coupling constraints.
Recombining Scenario Trees

- Idea: Not recombining scenarios - but coinciding subtrees.
Recombining Scenario Trees

- Idea: Not recombining scenarios - but coinciding subtrees.
- Node number is not reduced! Benefit?
Recombining Scenario Trees

- Idea: Not recombining scenarios - but coinciding subtrees.
- Node number is not reduced! Benefit?
  $\Rightarrow$ same subtrees $=$ same subproblems within Nested Benders Decomposition.
Recombining Scenario Trees

Definition

We say that nodes $\xi^R = \xi_i^R$ and $\xi^R = \xi_k^R$ can be recombined at time $R$, if both nodes share the same subtree, i.e.,

$$\mathbb{P}\left[(\xi_t)_{t=R,...,T} \in \cdot \left| \xi^R = \xi_i^R \right.\right] = \mathbb{P}\left[(\xi_t)_{t=R,...,T} \in \cdot \left| \xi^R = \xi_k^R \right.\right].$$
Consistency of Recombining Tree Approximations

Theorem (C. Küchler 2007)

A *recombining tree approximation is consistent* under the assumptions

- continuity of conditional distributions,
- complete recourse,
- existence of 'bounded' optimal solutions, and
- \((k)\)-short-term memory, i.e., for \(\Pr^t\)-a.e. \(u_{1,\ldots,t} \in \Xi^t, t = 1, \ldots, T - 1\),

\[
P \left[ \xi_{t+1} \in \cdot \mid \xi_{(1,\ldots,t)} = u_{1,\ldots,t} \right] = P \left[ \xi_{t+1} \in \cdot \mid \xi_{(t-k,\ldots,t)} = u_{(t-k,\ldots,t)} \right].
\]
Consistency of Recombining Tree Approximations

Theorem (C. Küchler 2007)

A recombining tree approximation is consistent under the assumptions

- continuity of conditional distributions,
- complete recourse,
- existence of 'bounded' optimal solutions, and
- \((k)\)-short-term memory, i.e., for \(\mathbb{P}^t\)-a.e. \(u(1,\ldots,t) \in \Xi^t, \ t = 1, \ldots, T - 1\),

\[
\mathbb{P} \left[ \xi_{t+1} \in \cdot \mid \xi_{(1,\ldots,t)} = u(1,\ldots,t) \right] = \mathbb{P} \left[ \xi_{t+1} \in \cdot \mid \xi_{(t-k,\ldots,t)} = u(t-k,\ldots,t) \right].
\]

Remark:

- Consider time series models \(\xi_{t+1} = f(\xi_t, \ldots, \xi_{t-k}, \varepsilon_t)\), with \(\varepsilon_t\) independent of \(\sigma(\xi_1, \ldots, \xi_{t-k-1})\) and \(f\) Lipschitz-continuous

\[\Rightarrow (\xi_t)_t\] can be displayed (without great loss of precision) by a scenario tree, where at certain time points scenarios with similar short-term history recombine

- scenario tree construction methods of Heitsch and Römisch can be adapted

Christian Küchler, Stefan Vigerske

Decomposition of MSPs with Recombining Scenario Trees
Problem formulation

Linear multistage stochastic program:

\[
\begin{align*}
\min_{(x_t)_{t=1,...,T}} & \quad \mathbb{E} \left[ \sum_{t=1}^{T} \langle b_t(\xi_t), x_t \rangle \right] \\
\text{s.t.} & \quad A_{t,0}(\xi_t)x_t + A_{t,1}(\xi_t)x_{t-1} = h_t(\xi_t), \quad t = 2, \ldots, T, \\
& \quad x_t \in X_t, \quad x_t \in \sigma(\xi^t), \quad t = 1, \ldots, T.
\end{align*}
\]
Problem formulation

Linear multistage stochastic program - dynamic formulation:

\[
\min_{(x_t)_{t=1,...,R}} \mathbb{E} \left[ \sum_{t=1}^{R} \langle b_t(\xi_t), x_t \rangle + Q_R(x_R, \xi^R) \right]
\]

s.t. \( A_{t,0}(\xi_t)x_t + A_{t,1}(\xi_t)x_{t-1} = h_t(\xi_t), \quad t = 2, \ldots, R, \)

\( x_t \in X_t, \quad x_t \in \sigma(\xi^t), \quad t = 1, \ldots, R. \)

with cost-to-go function

\[
Q_R(x_R, \xi^R_i) = \min_{(x_t)_{t=R+1,...,T}} \mathbb{E} \left[ \sum_{t=R+1}^{T} \langle b_t(\xi_t), x_t \rangle \right| \xi^R = \xi^R_i]
\]

s.t. \((x_t)_{t=R+1,...,T}\) is admissible.
Overview

Motivation and Introduction

Modified Benders Decomposition Algorithm

Conclusions
Cutting plane approximation of cost-to-go function

• $Q_R(\cdot, \xi^R_i)$ is convex and piecewise linear
Cutting plane approximation of cost-to-go function

- $Q_R(\cdot, \xi_i^R)$ is convex and piecewise linear
  ⇒ approximation by supporting hyperplanes: optimality cuts and feasibility cuts
Benders Decomposition and Recombining Scenario Trees
Benders Decomposition and Recombining Scenario Trees

Motivation and Introduction

Modified Benders Decomposition Algorithm

Conclusions

Christian Küchler, Stefan Vigerske

Decomposition of MSPs with Recombining Scenario Trees
Benders Decomposition and Recombining Scenario Trees

• Benders Decomposition: replace $Q_R(\cdot, \xi^R_i)$ by a cutting plane approximation $Q^L_R(\cdot, \xi^R_i)$
• recombining scenario tree allows simultaneous approximation by reuse of cutting planes

Christian Küchler, Stefan Vigerske
Decomposition of MSPs with Recombining Scenario Trees
Basic Benders Decomposition Algorithm

1. Define functions $Q_R^L(\cdot, \xi^R_i) := -\infty$, underestimating $Q_R(\cdot, \xi^R_i)$
2. Solve the master problem

$$\min \mathbb{E} \left[ \sum_{t=1}^{R} \langle b_t(\xi_t), x_t \rangle + Q_R^L(x_R, \xi^R) \right]$$

s.t. $(x_t)_{t=1,\ldots,R}$ is admissible.

⇒ obtain solution points $x_R(\xi^R_i)$.

3. Solve subproblem $Q_R(x_R(\xi^R_i), \xi^R_i)$ for all $\xi^R_i$
   ⇒ use dual solutions to simultaneously improve $Q_R^L(\cdot, \xi^R_i)$ (optimality and feasibility cuts)
4. If a $Q_R^L(\cdot, \xi^R_i)$ has changed, go to 2.
Basic Benders Decomposition Algorithm

1. Define functions $Q^L_R(\cdot, \xi^R_i) := -\infty$, underestimating $Q_R(\cdot, \xi^R_i)$
2. Solve the master problem

$$
\min \mathbb{E} \left[ \sum_{t=1}^R \langle b_t(\xi_t), x_t \rangle + Q^L_R(x_R, \xi^R) \right]
$$

s.t. $(x_t)_{t=1,...,R}$ is admissible.

$\Rightarrow$ obtain MANY solution points $x_R(\xi^R_i)$.

3. Solve subproblem $Q_R(x_R(\xi^R_i), \xi^R_i)$ FOR ALL $\xi^R_i$

$\Rightarrow$ use dual solutions to simultaneously improve $Q^L_R(\cdot, \xi^R_i)$ (optimality and feasibility cuts)

4. If a $Q^L_R(\cdot, \xi^R_i)$ has changed, go to 2.
Basic Benders Decomposition Algorithm

1. Define functions $Q^L_R(\cdot, \xi^R_i) := -\infty$, underestimating $Q_R(\cdot, \xi^R_i)$
2. Solve the master problem

$$
\min \mathbb{E} \left[ \sum_{t=1}^{R} \langle b_t(\xi_t), x_t \rangle + Q^L_R(x_R, \xi^R) \right]
$$

s.t. $(x_t)_{t=1,\ldots,R}$ is admissible.

$\Rightarrow$ obtain MANY solution points $x_R(\xi^R_i)$.

3. Solve subproblem $Q_R(x_R(\xi^R_i), \xi^R_i)$ FOR ALL $\xi^R_i$

$\Rightarrow$ use dual solutions to simultaneously improve $Q^L_R(\cdot, \xi^R_i)$ (optimality and feasibility cuts)

4. If a $Q^L_R(\cdot, \xi^R_i)$ has changed, go to 2.

Observation: Often solution points $x_R(\xi^R_i)$ are close and share the same $Q_R(\cdot, \xi^R_i)$ (which is Lipschitz continuous)
Thinning the decision space

⇒ **Thinning:** For some parameter $\rho \in [0, 1]$:
If $\|x_R - x'_R\| < \rho$, evaluate only $Q_R(x_R, \xi^R_i)$, not $Q_R(x'_R, \xi^R_i)$.

Christian Küchler, Stefan Vigerske

Decomposition of MSPs with Recombining Scenario Trees
Thinning the decision space

⇒ **Thinning**: For some parameter \( \rho \in [0, 1] \):

If \( \|x_R - x'_R\| < \rho \), evaluate only \( Q_R(x_R, \xi_i^R) \), not \( Q_R(x'_R, \xi_i^R) \).

- large \( \rho \) ⇒ throw away many points \( x'_R \) ⇒ rough approximation of \( Q_R(\cdot, \xi_i^R) \)
Thinning the decision space

⇒ **Thinning**: For some parameter $\rho \in [0, 1]$: If $\|x_R - x'_R\| < \rho$, evaluate only $Q_R(x_R, \xi_i^R)$, not $Q_R(x'_R, \xi_i^R)$.

- large $\rho$ ⇒ throw away many points $x'_R$ ⇒ rough approximation of $Q_R(\cdot, \xi_i^R)$
- Decrease $\rho$ to improve accuracy.
Thinning the decision space

⇒ **Thinning**: For some parameter \( \rho \in [0, 1] \):
   If \( \| x_R - x'_R \| < \rho \), evaluate only \( Q_R(x_R, \xi^R_i) \), not \( Q_R(x'_R, \xi^R_i) \).

- large \( \rho \) ⇒ throw away many points \( x'_R \) ⇒ rough approximation of \( Q_R(\cdot, \xi^R_i) \)
- Decrease \( \rho \) to improve accuracy.

Example:

<table>
<thead>
<tr>
<th>time horizon</th>
<th>subcases/day</th>
<th>no recombination</th>
<th>with recombination</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td># subproblems</td>
<td>time</td>
</tr>
<tr>
<td>2 days</td>
<td>2</td>
<td>9</td>
<td>10s</td>
</tr>
<tr>
<td>2 days</td>
<td>4</td>
<td>9</td>
<td>12s</td>
</tr>
<tr>
<td>3 days</td>
<td>2</td>
<td>73</td>
<td>99s</td>
</tr>
<tr>
<td>3 days</td>
<td>4</td>
<td>73</td>
<td>94s</td>
</tr>
<tr>
<td>4 days</td>
<td>2</td>
<td>585</td>
<td>859s</td>
</tr>
<tr>
<td>4 days</td>
<td>4</td>
<td>585</td>
<td>789s</td>
</tr>
</tbody>
</table>

Power scheduling under uncertain wind energy input. Hourly discretization, binary branching (3x per day), recombination after each day, final \( \rho = 0.0001 \).
Motivation and Introduction

Modified Benders Decomposition Algorithm

Conclusions

Thinining the decision space

- Longer time horizons $T$, many nodes $\xi^R_i$,
decreasing $\rho \Rightarrow$ many $Q_R(x^R, \xi^R_i)$ to evaluate.

<table>
<thead>
<tr>
<th>time horizon</th>
<th>subtrees/day</th>
<th>rough ((\rho = 0.1))</th>
<th>complete ((\rho = 0.001))</th>
</tr>
</thead>
<tbody>
<tr>
<td>2 weeks</td>
<td>2</td>
<td>11s</td>
<td>711s</td>
</tr>
<tr>
<td>2 weeks</td>
<td>4</td>
<td>21s</td>
<td>2512s</td>
</tr>
<tr>
<td>1 month</td>
<td>2</td>
<td>21s</td>
<td>&gt;3h</td>
</tr>
<tr>
<td>1 month</td>
<td>4</td>
<td>35s</td>
<td>&gt;3h</td>
</tr>
<tr>
<td>3 months</td>
<td>2</td>
<td>60s</td>
<td>&gt;3h</td>
</tr>
<tr>
<td>3 months</td>
<td>4</td>
<td>60s</td>
<td>&gt;3h</td>
</tr>
</tbody>
</table>

1,330 variables per master problem, 3 months without recombination $\approx 2 * 10^{81}$ master problems.
Thinning the decision space

• Longer time horizons $T$, many nodes $\xi^R_i$, decreasing $\rho \Rightarrow$ many $Q_R(x^R, \xi^R_i)$ to evaluate.

<table>
<thead>
<tr>
<th>time horizon</th>
<th>subtrees/day</th>
<th>rough $\rho = 0.1$</th>
<th>complete $\rho = 0.001$</th>
</tr>
</thead>
<tbody>
<tr>
<td>2 weeks</td>
<td>2</td>
<td>11s</td>
<td>711s</td>
</tr>
<tr>
<td>2 weeks</td>
<td>4</td>
<td>21s</td>
<td>2512s</td>
</tr>
<tr>
<td>1 month</td>
<td>2</td>
<td>21s</td>
<td>&gt;3h</td>
</tr>
<tr>
<td>1 month</td>
<td>4</td>
<td>35s</td>
<td>&gt;3h</td>
</tr>
<tr>
<td>3 months</td>
<td>2</td>
<td>60s</td>
<td>&gt;3h</td>
</tr>
<tr>
<td>3 months</td>
<td>4</td>
<td>60s</td>
<td>&gt;3h</td>
</tr>
</tbody>
</table>

1.330 variables per master problem, 3 months without recombination $\approx 2 \times 10^{81}$ master problems.

• **Empirical observation:** The rough phase ($\rho = 0.1$) already gives very accurate approximations.
Thinning the decision space

- Longer time horizons $T$, many nodes $\xi_i^R$, decreasing $\rho \Rightarrow$ many $Q_R(x_R, \xi_i^R)$ to evaluate.

<table>
<thead>
<tr>
<th>time horizon</th>
<th>subtrees/day</th>
<th>rough $\rho = 0.1$</th>
<th>complete $\rho = 0.001$</th>
</tr>
</thead>
<tbody>
<tr>
<td>2 weeks</td>
<td>2</td>
<td>11s</td>
<td>711s</td>
</tr>
<tr>
<td>2 weeks</td>
<td>4</td>
<td>21s</td>
<td>2512s</td>
</tr>
<tr>
<td>1 month</td>
<td>2</td>
<td>21s</td>
<td>&gt;3h</td>
</tr>
<tr>
<td>1 month</td>
<td>4</td>
<td>35s</td>
<td>&gt;3h</td>
</tr>
<tr>
<td>3 months</td>
<td>2</td>
<td>60s</td>
<td>&gt;3h</td>
</tr>
<tr>
<td>3 months</td>
<td>4</td>
<td>60s</td>
<td>&gt;3h</td>
</tr>
</tbody>
</table>

1.330 variables per master problem, 3 months without recombination $\approx 2 \times 10^{81}$ master problems.

- Empirical observation: The rough phase ($\rho = 0.1$) already gives very accurate approximations.
- How can we evaluate the approximation quality of the rough phase?
- How to estimate the differences $Q_R(\cdot, \xi_i^R) - Q_{R_i}^L(\cdot, \xi_i^R)$?
Evaluating the approximation quality

• How to estimate the differences $Q_R(\cdot, \xi^R_i) - Q^L_R(\cdot, \xi^R_i)$?
Evaluating the approximation quality

- How to estimate the differences $Q_R(\cdot, \xi^R_i) - Q^L_R(\cdot, \xi^R_i)$?
- Construct upper bounds $Q^U_R(\cdot, \xi^R_i)$ of $Q_R(\cdot, \xi^R_i)$ and use the gap $Q^U_R(\cdot, \xi^R_i) - Q^L_R(\cdot, \xi^R_i)$. 
Evaluating the approximation quality

- How to estimate the differences $Q_R(\cdot, \xi_i^R) - Q_L^R(\cdot, \xi_i^R)$?
- Construct upper bounds $Q^U_R(\cdot, \xi_i^R)$ of $Q_R(\cdot, \xi_i^R)$ and use the gap $Q^U_R(\cdot, \xi_i^R) - Q_L^R(\cdot, \xi_i^R)$.
- build $Q^U_R(\cdot, \xi_i^R)$ by taking a convex combination of points where $Q_R(\cdot, \xi_i^R)$ has been evaluated before.

Christian Küchler, Stefan Vigertske
Decomposition of MSPs with Recombining Scenario Trees
Extended Nested Benders Algorithm

Upper bounds allow *error estimate* during the solution process.

<table>
<thead>
<tr>
<th>time horizon</th>
<th>subtrees/day</th>
<th>no upper bounds</th>
<th>with upper bounds</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td>rough</td>
<td>rough phase gap</td>
</tr>
<tr>
<td>2 weeks</td>
<td>2</td>
<td>11s</td>
<td>27s</td>
</tr>
<tr>
<td></td>
<td>4</td>
<td>21s</td>
<td>57s</td>
</tr>
<tr>
<td>1 month</td>
<td>2</td>
<td>21s</td>
<td>59s</td>
</tr>
<tr>
<td></td>
<td>4</td>
<td>35s</td>
<td>151s</td>
</tr>
<tr>
<td>3 months</td>
<td>2</td>
<td>60s</td>
<td>195s</td>
</tr>
<tr>
<td></td>
<td>4</td>
<td>60s</td>
<td>868s</td>
</tr>
</tbody>
</table>
Extended Nested Benders Algorithm

Upper bounds allow error estimate during the solution process.
⇒ adaptive stopping criteria

1. Local: Do not solve subproblem $Q_R(x_R, \xi_i^R)$ if gap is small (at $x_R$).
2. Global: Stop Algorithm if the first stage gap is small.

<table>
<thead>
<tr>
<th>time horizon</th>
<th>subtrees/day</th>
<th>no upper bounds</th>
<th>with upper bounds</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td>rough</td>
<td>complete</td>
</tr>
<tr>
<td>2 weeks</td>
<td>2</td>
<td>11s</td>
<td>711s</td>
</tr>
<tr>
<td>2 weeks</td>
<td>4</td>
<td>21s</td>
<td>2512s</td>
</tr>
<tr>
<td>1 month</td>
<td>2</td>
<td>21s</td>
<td>&gt;3h</td>
</tr>
<tr>
<td>1 month</td>
<td>4</td>
<td>35s</td>
<td>&gt;3h</td>
</tr>
<tr>
<td>3 months</td>
<td>2</td>
<td>60s</td>
<td>&gt;3h</td>
</tr>
<tr>
<td>3 months</td>
<td>4</td>
<td>60s</td>
<td>&gt;3h</td>
</tr>
</tbody>
</table>
Overview

Motivation and Introduction

Modified Benders Decomposition Algorithm

Conclusions
Conclusions

Summary:

- Recombination of scenarios = Assignment of same subtrees
- Modified Benders Decomposition to exploit this structure: *simultaneous cutting plane approximation*

Thank you!

Conclusions

Summary:

- Recombination of scenarios = Assignment of same subtrees
- Modified Benders Decomposition to exploit this structure: *simultaneous cutting plane approximation*
- Thinning of exponentially growing control point sets
Conclusions

Summary:

- Recombination of scenarios = Assignment of same subtrees
- Modified Benders Decomposition to exploit this structure: simultaneous cutting plane approximation
- Thinning of exponentially growing control point sets
- Upper bounds for estimate on approximation error
Conclusions

Summary:

- Recombination of scenarios = Assignment of same subtrees
- Modified Benders Decomposition to exploit this structure: simultaneous cutting plane approximation
- Thinning of exponentially growing control point sets
- Upper bounds for estimate on approximation error
- Consistency of recombining tree approximation
Conclusions

Summary:

- Recombination of scenarios = Assignment of same subtrees
- Modified Benders Decomposition to exploit this structure: simultaneous cutting plane approximation
- Thinning of exponentially growing control point sets
- Upper bounds for estimate on approximation error
- Consistency of recombining tree approximation

Further Developments:

- Better use of **warmstarts** in repeated solve of the same subproblem
- Improving the Nested Benders **sequencing protocol**
Conclusions

Summary:

- Recombination of scenarios = Assignment of same subtrees
- Modified Benders Decomposition to exploit this structure: simultaneous cutting plane approximation
- Thinning of exponentially growing control point sets
- Upper bounds for estimate on approximation error
- Consistency of recombining tree approximation

Further Developments:

- Better use of warmstarts in repeated solve of the same subproblem
- Improving the Nested Benders sequencing protocol
- Support of integer variables
Conclusions

Summary:

- Recombination of scenarios = Assignment of same subtrees
- Modified Benders Decomposition to exploit this structure: *simultaneous cutting plane approximation*
- Thinning of exponentially growing control point sets
- Upper bounds for estimate on approximation error
- Consistency of recombining tree approximation

Further Developments:

- Better use of *warmstarts* in repeated solve of the same subproblem
- Improving the Nested Benders *sequencing protocol*
- Support of *integer variables*?

Thank you!