Decomposition of Multistage Stochastic Programs with Recombining Scenario Trees

Christian Küchler*, Stefan Vigerske*

Humboldt-Universität zu Berlin
Institut für Mathematik

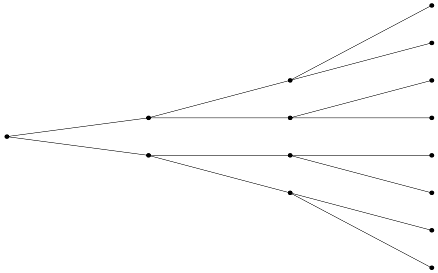
Operations Research 2007
Universität des Saarlandes, 6. September 2007

*supported by BMBF and WWTF

Overview

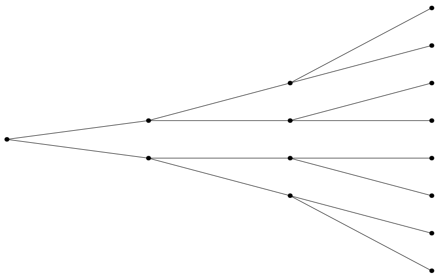
Scenario Trees

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



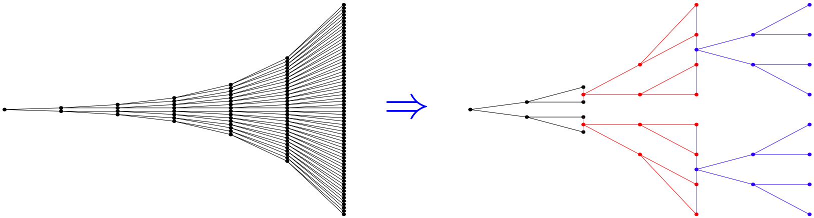
Scenario Trees

- 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*.



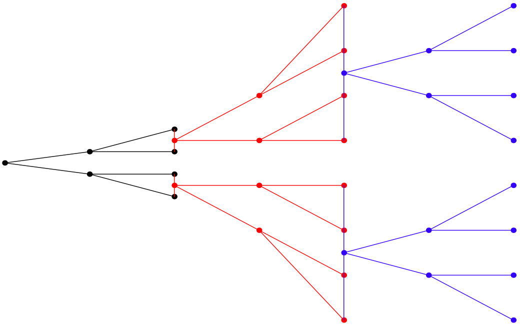
Scenario Trees

- 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?



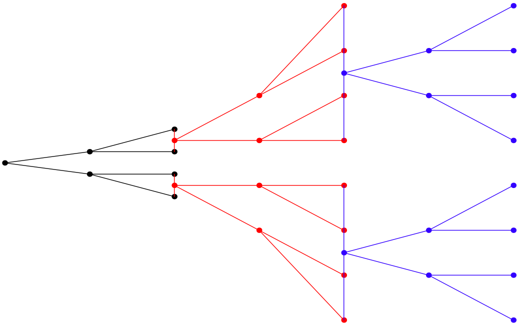
Recombining Scenario Trees

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



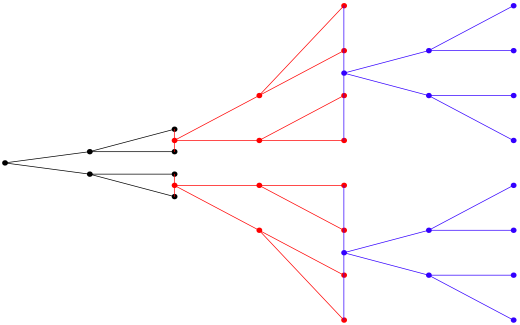
Recombining Scenario Trees

- 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.



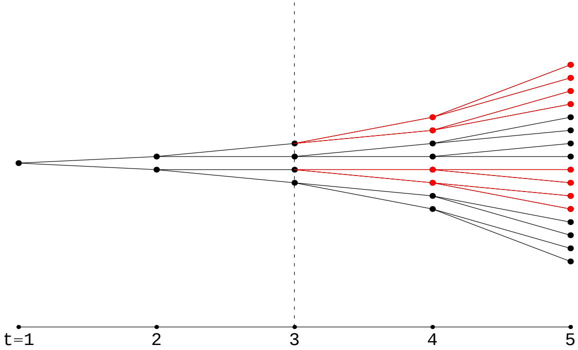
Recombining Scenario Trees

- 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.



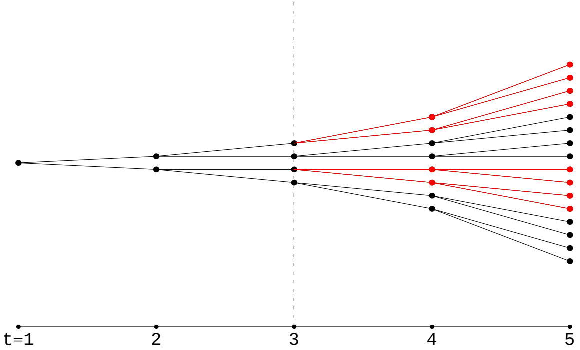
Recombining Scenario Trees

- Idea: Not recombining scenarios - but coinciding subtrees.



Recombining Scenario Trees

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

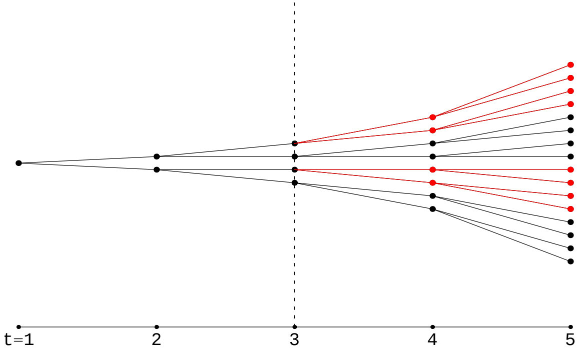


Recombining Scenario Trees

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

  ⇒ same subtrees = same subproblems

  within Nested Benders Decomposition.

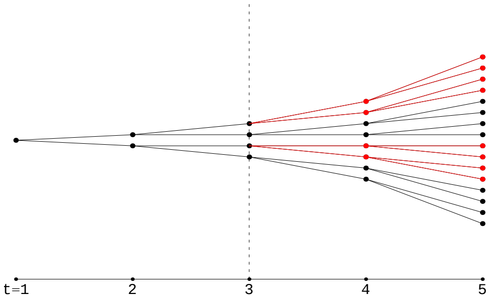


Recombining Scenario Trees

Definition

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

$$\mathbb{P}\left[(\boldsymbol{\xi}_t)_{t=R,\ldots,T} \in \cdot \,\middle|\, \boldsymbol{\xi}^R = \xi_i^R\right] = \mathbb{P}\left[(\boldsymbol{\xi}_t)_{t=R,\ldots,T} \in \cdot \,\middle|\, \boldsymbol{\xi}^R = \xi_k^R\right].$$



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.* $u_{(1,\dots,t)} \in \Xi^t$, $t = 1, \dots, T-1$,

$$\mathbb{P}\left[\xi_{t+1} \in \cdot \,\middle|\, \xi_{(1,\dots,t)} = u_{(1,\dots,t)}\right] = \mathbb{P}\left[\xi_{t+1} \in \cdot \,\middle|\, \xi_{(t-k,\dots,t)} = u_{(t-k,\dots,t)}\right].$$

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. $u_{(1,\ldots,t)} \in \Xi^t$, $t = 1, \ldots, T-1$,*

$$\mathbb{P}\left[\boldsymbol{\xi}_{t+1} \in \cdot \,\middle|\, \boldsymbol{\xi}_{(1,\ldots,t)} = u_{(1,\ldots,t)}\right] = \mathbb{P}\left[\boldsymbol{\xi}_{t+1} \in \cdot \,\middle|\, \boldsymbol{\xi}_{(t-k,\ldots,t)} = u_{(t-k,\ldots,t)}\right].$$

Remark:

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

$\Rightarrow$ $(\boldsymbol{\xi}_t)_t$ 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

Problem formulation

Linear multistage stochastic program:

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

Problem formulation

Linear multistage stochastic program - dynamic formulation:

$$\begin{aligned}
\min_{(x_t)_{t=1,\ldots,R}} \quad & \mathbb{E}\left[\sum_{t=1}^{R} \langle b_t(\xi_t), x_t\rangle + Q_R(x_R, \xi^R)\right] \\
\text{s.t.} \quad & A_{t,0}(\xi_t)x_t + A_{t,1}(\xi_t)x_{t-1} = h_t(\xi_t), \qquad t = 2,\ldots,R, \\
& x_t \in X_t, \quad x_t \in \sigma(\xi^t), \qquad t = 1,\ldots,R.
\end{aligned}$$

with cost-to-go function

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

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

Overview

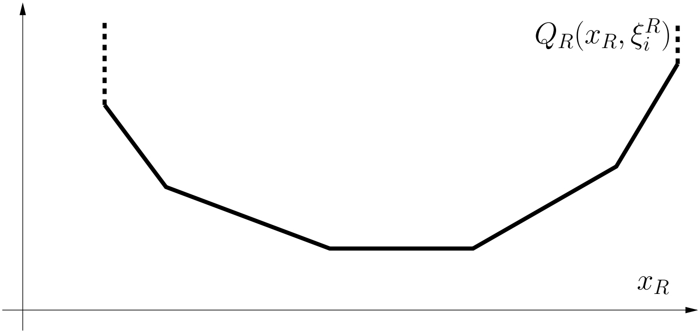
Motivation and Introduction

Modified Benders Decomposition Algorithm

Conclusions

Cutting plane approximation of cost-to-go function

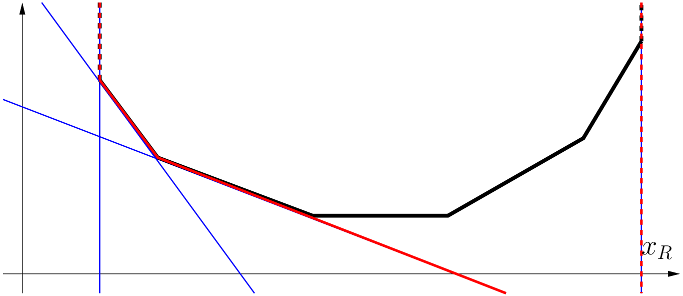
- $Q_R(\cdot, \xi_i^R)$ is convex and piecewise linear



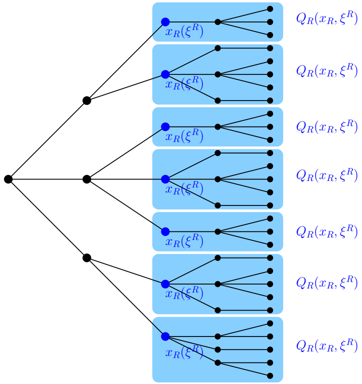
Cutting plane approximation of cost-to-go function

- $Q_R(\cdot, \xi_i^R)$ is convex and piecewise linear

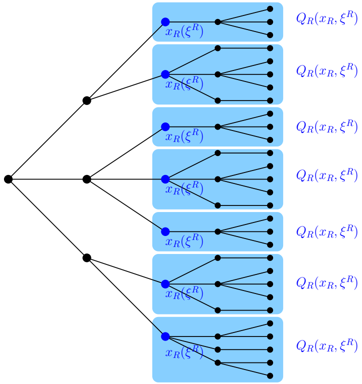
$\Rightarrow$ approximation by supporting hyperplanes: optimality cuts and feasibility cuts



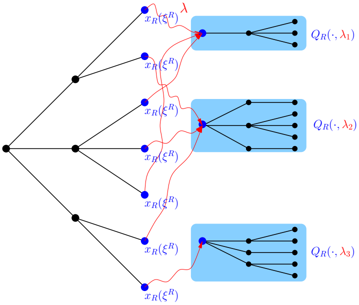
Benders Decomposition and Recombining Scenario Trees



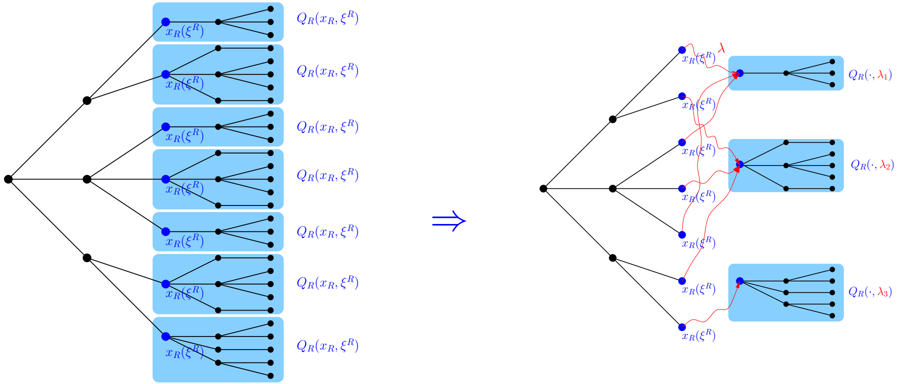
Benders Decomposition and Recombining Scenario Trees



$\Rightarrow$



Benders Decomposition and Recombining Scenario Trees



- Benders Decomposition: replace $Q_R(\cdot, \xi_i^R)$ by a cutting plane approximation $Q_R^L(\cdot, \xi_i^R)$
- recombining scenario tree allows simultaneous approximation by reuse of cutting planes

Basic Benders Decomposition Algorithm

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

$$\min \quad \mathbb{E}\left[\sum_{t=1}^{R} \langle b_t(\boldsymbol{\xi}_t), x_t \rangle + Q_R^L(x_R, \boldsymbol{\xi}^R)\right]$$
$$\text{s.t. } (x_t)_{t=1,\dots,R} \text{ is admissible.}$$

   $\Rightarrow$ obtain solution points $x_R(\xi_i^R)$.
3. Solve subproblem $Q_R(x_R(\xi_i^R), \xi_i^R)$ for all ξ_i^R
   $\Rightarrow$ use dual solutions to simultaneously improve $Q_R^L(\cdot, \xi_i^R)$ (optimality and feasibility cuts)
4. If a $Q_R^L(\cdot, \xi_i^R)$ has changed, go to 2.

Basic Benders Decomposition Algorithm

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

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

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

$\Rightarrow$ obtain MANY solution points $x_R(\xi_i^R)$.

3. Solve subproblem $Q_R(x_R(\xi_i^R), \xi_i^R)$ FOR ALL ξ_i^R
   $\Rightarrow$ use dual solutions to simultaneously improve $Q_R^L(\cdot, \xi_i^R)$ (optimality and feasibility cuts)
4. If a $Q_R^L(\cdot, \xi_i^R)$ has changed, go to 2.

Basic Benders Decomposition Algorithm

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

$$\min \quad \mathbb{E}\left[\sum_{t=1}^{R} \langle b_t(\boldsymbol{\xi}_t), x_t\rangle + Q_R^L(x_R,\boldsymbol{\xi}^R)\right]$$
$$\text{s.t. } (x_t)_{t=1,\dots,R} \text{ is admissible.}$$

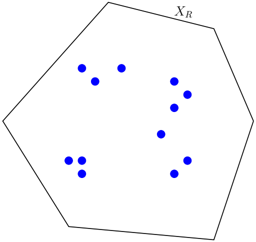
   $\Rightarrow$ obtain MANY solution points $x_R(\xi_i^R)$.

3. Solve subproblem $Q_R(x_R(\xi_i^R),\xi_i^R)$ FOR ALL ξ_i^R
   $\Rightarrow$ use dual solutions to simultaneously improve $Q_R^L(\cdot,\xi_i^R)$ (optimality and feasibility cuts)
4. If a $Q_R^L(\cdot,\xi_i^R)$ has changed, go to 2.

Observation: Often solution points $x_R(\xi_i^R)$ are close and share the same $Q_R(\cdot,\xi_i^R)$ (which is Lipschitz continuous)

Thinning the decision space

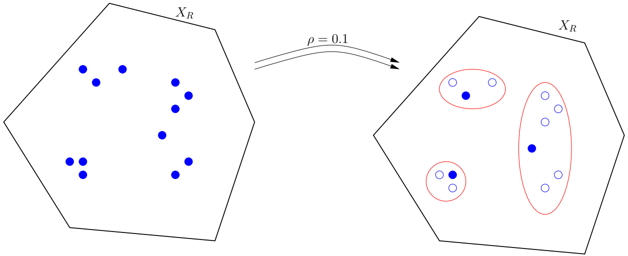
⇒ Thinning: For some parameter $\rho \in [0,1]$:
If $\|x_R - x'_R\| < \rho$, evaluate only $Q_R(x_R, \xi_i^R)$, not $Q_R(x'_R, \xi_i^R)$.



Thinning the decision space

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

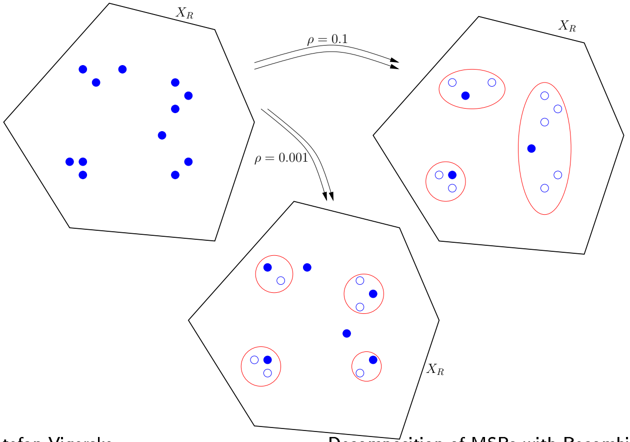
- large $\rho \Rightarrow$ throw away many points $x'_R \Rightarrow$ rough approximation of $Q_R(\cdot, \xi_i^R)$



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 $\rho \Rightarrow$ throw away many points $x'_R \Rightarrow$ rough approximation of $Q_R(\cdot, \xi_i^R)$
- Decrease ρ to improve accuracy.



Thinning the decision space

$\Rightarrow$ 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 $\rho \Rightarrow$ throw away many points $x'_R \Rightarrow$ rough approximation of $Q_R(\cdot, \xi_i^R)$
- Decrease ρ to improve accuracy.

Example:

		no recombination		with recombination	
time horizon	subtrees/day	# subproblems	time	# subproblems	time
2 days	2	9	10s	3	4s
2 days	4	9	12s	5	7s
3 days	2	73	99s	5	5s
3 days	4	73	94s	9	9s
4 days	2	585	859s	7	6s
4 days	4	585	789s	13	13s

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

Thinning the decision space

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

time horizon	subtrees/day	rough ($\rho = 0.1$)	complete ($\rho = 0.001$)
2 weeks	2	11s	711s
2 weeks	4	21s	2512s
1 month	2	21s	>3h
1 month	4	35s	>3h
3 months	2	60s	>3h
3 months	4	60s	>3h

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 ξ_i^R, decreasing $\rho \Rightarrow$ many $Q_R(x_R, \xi_i^R)$ to evaluate.

time horizon	subtrees/day	rough ($\rho = 0.1$)	complete ($\rho = 0.001$)
2 weeks	2	11s	711s
2 weeks	4	21s	2512s
1 month	2	21s	>3h
1 month	4	35s	>3h
3 months	2	60s	>3h
3 months	4	60s	>3h

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

- Empiricial observation: The rough phase ($\rho = 0.1$) already gives very accurate approximations.

Thinning the decision space

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

time horizon	subtrees/day	rough ($\rho = 0.1$)	complete ($\rho = 0.001$)
2 weeks	2	11s	711s
2 weeks	4	21s	2512s
1 month	2	21s	>3h
1 month	4	35s	>3h
3 months	2	60s	>3h
3 months	4	60s	>3h

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

- Empiricial 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 $Q_R(\cdot, \xi_i^R) - Q_R^L(\cdot, \xi_i^R)$?

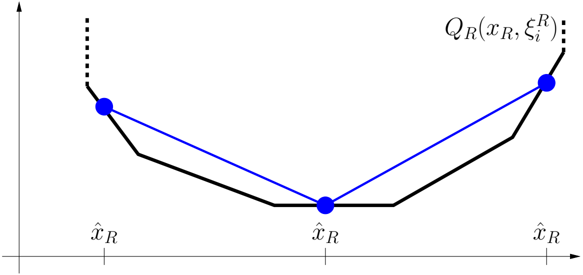
Evaluating the approximation quality

- How to estimate the differences $Q_R(\cdot, \xi_i^R) - Q_R^L(\cdot, \xi_i^R)$?
- Construct upper bounds $Q_R^U(\cdot, \xi_i^R)$ of $Q_R(\cdot, \xi_i^R)$ and use the gap

$$Q_R^U(\cdot, \xi_i^R) - Q_R^L(\cdot, \xi_i^R).$$

Evaluating the approximation quality

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



$Q_R(\cdot, \xi_i^R)$ and its upper bound $Q_R^U(\cdot, \xi_i^R)$.

Extended Nested Benders Algorithm

Upper bounds allow error estimate during the solution process.

		no upper bounds	with upper bounds	
time horizon	subtrees/day	rough	rough	rough phase gap
2 weeks	2	11s	27s	0.04%
2 weeks	4	21s	57s	0.03%
1 month	2	21s	59s	0.12%
1 month	4	35s	151s	0.03%
3 months	2	60s	195s	0.09%
3 months	4	60s	868s	0.09%

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.

		no upper bounds		with upper bounds		
time horizon	subtrees/day	rough	complete	rough	rough phase gap	complete
2 weeks	2	11s	711s	27s	0.04%	536s
2 weeks	4	21s	2512s	57s	0.03%	4535s
1 month	2	21s	>3h	59s	0.12%	>3h
1 month	4	35s	>3h	151s	0.03%	>3h
3 months	2	60s	>3h	195s	0.09%	829s
3 months	4	60s	>3h	868s	0.09%	>3h

Overview

Conclusions

Summary:

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

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
- consistenty 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
- consistenty 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
- consistenty 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
- consistenty 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!

More details: paper at `http://www.math.hu-berlin.de/~stefan`