Stochastic Dual Dynamic Programming

ESD.S30  Electric Power System Modeling for a Low Carbon Economy

Prof. Andres Ramos

http://www.iit.upcomillas.es/aramos/
Andres.Ramos@upcomillas.es
arght@mit.edu

References


Other resources

- Stochastic Programming Community Home Page. Stochastic Programming Resources (www, papers, tutorials, lecture notes, books) (http://stoprog.org/)
- Stochastic Programming Bibliography (http://www.eco.rug.nl/mally/spbib.html)
- STOCHASTIC PROGRAMMING E-PRINT SERIES (http://www.speps.org/)

- Red Temática de Optimización bajo Incertidumbre (ReTOBI) (http://www.optimizacionbajoincertidumbre.org/)

- International Conference in Stochastic Programming
1. General overview
2. Two-stage and multistage programming
3. Decomposition techniques
4. Benders' decomposition
5. Nested Benders' decomposition
6. Dantzig-Wolfe decomposition
7. Lagrangian relaxation
8. Scenario tree
9. Decomposition in two-stage and multistage stochastic programming
10. Improvements in decomposition techniques
11. Simulation in stochastic optimization
12. Stochastic dual dynamic programming

Two-stage and multistage programming
Two-stage (PL-2) and multistage (PL-P) linear programming

- Two-stage PL-2: decisions in two stages
- Multistage PL-P: decisions in multiple stages

- Stairway structure of the constraint matrix (block diagonal)
  - Each stage in only related with the previous one

- Problems of each stage are similar (they have the same structure)
- The matrix structure can be detected by visual inspection
Two-stage linear programming PL-2

\[
\begin{bmatrix}
A_1 \\
B_1 \\
\end{bmatrix}
\begin{bmatrix}
A_2
\end{bmatrix}
\]

\[
\min\left(c_1^T x_1 + c_2^T x_2\right)_{x_1, x_2}
\]

\[
\begin{align*}
A_1 x_1 &= b_1 \\
B_1 x_1 + A_2 x_2 &= b_2 \\
x_1, \quad x_2 &\geq 0
\end{align*}
\]
Two-stage stochastic linear programming PLE-2

- O.F. minimizes first-stage costs and expected value of second-stage costs

\[
\begin{align*}
\min \ & c_1^T x_1 + \sum_{\omega \in \Omega} p_\omega c_2^\omega x_2^\omega \\
A_1 x_1 & = b_1 \\
B_1^\omega x_1 + A_2^\omega x_2^\omega & = b_2^\omega \\
x_1, x_2^\omega & \geq 0
\end{align*}
\]

- If \( A_2^\omega \) doesn’t depend on \( \omega \) it is called fixed resource
- Structure of the constraint matrix
Deterministic equivalent problem

- State space is small

- Formulation of the deterministic equivalent problem

\[
\begin{align*}
\min_{x_1, x_2^1, x_2^2, x_2^3} & \quad c_1^T x_1 + p_1^1 c_2^1 x_2^1 + p_1^2 c_2^2 x_2^2 + p_1^3 c_2^3 T x_2^3 \\
A_1 x_1 & = b_1 \\
B_1^{x_1} x_1 + A_2^{x_2} x_2^1 & = b_2^{x_1} \\
B_1^{x_2} x_1 + A_2^{x_2} x_2^1 & = b_2^{x_2} \\
B_1^{x_3} x_1 + A_2^{x_3} x_2^1 & = b_2^{x_3} \\
x_1, x_2^1, x_2^2, x_2^3 & \geq 0
\end{align*}
\]
Multistage Linear Programming PL-P

\[
\begin{array}{c}
A_1 \\
B_1 \\
B_2 \\
\vdots \\
B_{P-1} \\
A_P \\
\end{array}
\]

\[
\begin{align*}
\min \sum_{p=1}^{P} c_p^T x_p \\
B_{p-1} x_{p-1} + A_p x_p &= b_p & p = 1, \ldots, P \\
x_p &\geq 0 \\
B_0 &\equiv 0
\end{align*}
\]
Medium term hydrothermal scheduling problem: constraint matrix

Inter-period Constraints
\[ R_{p-1} + i_p - P_p - S_p = R_p \]

Intra-period Constraints

Variables:
- \( R_{p-1} \): reservoir level
- \( i_p \): inflow
- \( P_p \): hydro output
- \( S_p \): reservoir spillage
Medium term hydrothermal scheduling problem: constraint matrix
Multistage stochastic linear programming PLE-P

- O. F. minimizes expected costs of all the stages

\[
\min_{x_p^\omega} \sum_{p=1}^{P} \sum_{\omega_p \in \Omega_p} p_{p}^{\omega_p} c_{p}^{\omega_p} x_{p}^{\omega_p}
\]

\[
B_x^{\omega_p} = A_p x_p^{\omega_p} = b_p^{\omega_p} \quad p = 1, \ldots, P
\]

\[
x_{p}^{\omega_p} \geq 0
\]

\[
B_{01}^{\omega_1} \equiv 0
\]

- Size grows exponentially with the number of scenarios
- Probabilities \( p_{p}^{\omega_p} \) are conditioned
- Constraint matrix

![Constraint Matrix](image)
3. **Decomposition techniques**

1. General overview
2. Two-stage and multistage programming
3. Decomposition techniques
4. Benders' decomposition
5. Nested Benders' decomposition
6. Dantzig-Wolfe decomposition
7. Lagrangian relaxation
8. Scenario tree
9. Decomposition in two-stage and multistage stochastic programming
10. Improvements in decomposition techniques
11. Simulation in stochastic optimization
12. Stochastic dual dynamic programming

**Decomposition techniques**
Decomposition techniques

- **Divide and conquer strategy**
  - Time division by periods
  - Spatial division by (thermal) units
  - Division by scenarios
- Allow the solution of huge problems (not directly solvable) with a certain structure by solving iteratively small size problems
- Objective function and feasible region have to be convex whenever to obtain dual variables is needed
- Dantzig-Wolfe 1960, Benders 1962, Geoffrion 1972 (generalized Benders)
Decomposition techniques: classification

• According to the **difficulties**
  – **Variables** (Benders)
  – **Constraints** (Dantzig-Wolfe or lagrangian relaxation)

• According to the **exchanged information** between master and subproblem
  – **Primal** (Benders)
  – **Dual** (Dantzig-Wolfe or lagrangian relaxation)
Coordinating mechanisms. Hydrothermal model

- **Primal (quantities)**
  - Master assigns an amount of water to release in each period or the reservoir levels at the end of the period.
  - Each subproblem returns the marginal price (water value) associated to the use of the previous amount of water.

- **Dual (prices)**
  - Master gives a value to the water.
  - Each subproblem returns the future cost function taking into account this value.
Medium term hydrothermal scheduling problem

- Solvable by **Bd, DW-LR or nested Benders’ decomposition**
  - Variables of hydro release complicate the solution ⇒ Benders
  - Constraints of hydro release complicate the solution ⇒ Dantzig-Wolfe, lagrangian relaxation

- Criterion:
  - Engineering: context dependent
  - Mathematical:
    - What complicates? (foreseeable number of iterations)
    - Respective size of master and subproblems
Algorithm: Benders

- **Master problem: inter-period constraints**
- **Subproblem: intra-period constraints**
Benders' decomposition

- Reservoir level or hydro release is a given for the subproblem
Algorithm: Dantzig-Wolfe or lagrangian relaxation

- **Master problem**: inter-period constraints
- **Subproblem**: intra-period constraints
DW or LR decomposition

- Reservoir level is a variable for the subproblem
Algorithm: nested Benders’ decomposition

Subproblem 1

Subproblem 2

Subproblem 3:
Subproblem for subproblem 2
Master problem for subproblem 4

Subproblem 4
Nested Benders’ decomposition

Subproblem 1

Subproblem 2

Subproblem 3

Subproblem 4

Reservoir level or Hydro release

Water value

Reservoir level or Hydro release

Water value

Reservoir level or Hydro release

Water value

Water value
1. General overview
2. Two-stage and multistage programming
3. Decomposition techniques
4. Benders' decomposition
5. Nested Benders' decomposition
6. Dantzig-Wolfe decomposition
7. Lagrangian relaxation
8. Scenario tree
9. Decomposition in two-stage and multistage stochastic programming
10. Improvements in decomposition techniques
11. Simulation in stochastic optimization
12. Stochastic dual dynamic programming

Benders' decomposition
When to use Benders’ decomposition?

- Variables $x_1$ complicate the solution of the problem
- Implicitly $n_1 << n_2$
- Number of iterations related with $n_1$
- Matrix structure induces separability of subproblems
- Master and subproblem have different nature
  - Master in discrete variables (MIP)
  - Subproblem with nonlinear (convex) objective function (NLP)
- Benders’ decomposition needs convex o.f. and convex feasible region of the subproblem
Relaxed master problem and subproblem

**Master:** first stage + cuts

\[
\begin{align*}
& \min_{x_1, \theta_2} c_1^T x_1 + \theta_2 \\
& A_1 x_1 = b_1 \\
& \pi_2^T B_1 x_1 + \theta_2 \geq f_2^l + \pi_2^T B_1 x_1^l \quad l = 1, \ldots, j \\
& x_1 \geq 0
\end{align*}
\]

**Subproblem:** second stage with known decisions of the first stage

\[
\begin{align*}
& f_2^j = \min_{x_2} c_2^T x_2 \\
& A_2 x_2 = b_2 - B_1 x_1^l : \pi_2^j \\
& x_2 \geq 0
\end{align*}
\]
Benders’ master and subproblem

• Master
  – One cut is added in each iteration
  – Each cut defines a new feasible region
  – Optimal solution of previous iteration becomes infeasible
  – It is worthy to use dual simplex method
  – Size \((m_1 + j) \times (n_1 + 1)\)
  – It can be nonconvex (MIP, NLP)

• Subproblem
  – Each iteration modifies the RHS of the constraints
  – It is worthy to use primal simplex method (if size is adequate)
  – Size \(m_2 \times n_2\)
  – It must be convex (LP, NLP)
Benders’ algorithm (i)

- **Upper bound** of the optimal value of the o.f. of problem PL-2
  \[ z = c_1^T x_1^j + c_2^T x_2^j \]

- **Lower bound** of the problem, value obtained by the o.f. of the relaxed master problem
  \[ z = c_1^T x_1^j + \theta_2^j \]

- **Convergence condition**
  \[ \frac{|z - z|}{|z|} = \frac{|c_2^T x_2^j - \theta_2^j|}{c_1^T x_1^j + c_2^T x_2^j} \leq \varepsilon \]

or repetition of the last master proposal
Benders’ algorithm (ii)

- Successive approximation of the second-stage objective function by cuts.
- Benders’ cuts (cutting planes, support hyperplanes) are an outer linearization of the recourse function.
- Lower bound is monotonous increasing.
  Upper bound is not necessarily monotonous decreasing.
  - Upper bound is the minimum of previous upper bounds
- In the first iteration the value of $x_1^0$ can be fixed, if the problem nature is known, or by solving the master problem without cuts $\theta_2 = 0$
- In each iteration we have a quasi-optimal feasible solution
Benders’ algorithm (iii)

1. **Initialization:**
   
   \[ j = 0 \quad z = -\infty \quad \bar{z} = \infty \quad \varepsilon = 10^{-4} \]

2. **Solving the master problem**

   \[
   \begin{align*}
   \min_{x_1, \theta_2} & \quad c_1^T x_1 + \theta_2 \\
   \text{subject to} & \quad A_1 x_1 = b_1 \\
   & \quad \pi_2^T B_2 x_1 + \delta(\theta_2) \geq \pi_2^T b_2 \quad l = 1, \ldots, j \\
   & \quad x_1 \geq 0
   \end{align*}
   \]

   Determine the solution \((x_1^j, \theta_2^j)\) and the lower bound

   If no optimality cuts \(\theta_2 = 0\)

3. **Solving the subproblem of sum of infeasibilities**

   \[
   \begin{align*}
   f_2^j &= \min_{v^+, v^-} \quad e^T v^+ + e^T v^- \\
   & \quad A_2 x_2 + I v^+ - I v^- = b_2 - B_1 x_1^j : \pi_2^j \\
   & \quad x_2, v^+, v^- \geq 0
   \end{align*}
   \]

   If \(f_2^j \geq 0\) infeasibility cut

   If \(f_2^j = 0\) go to step 4.

4. **Solving the Benders’ subproblem**

   \[
   f_2^j = \min_{x_2} \quad c_2^T x_2 \\
   & \quad A_2 x_2 = b_2 - B_1 x_1^j : \pi_2^j \\
   & \quad x_2 \geq 0
   \]

   Obtain \(x_2^j\) and update the upper bound.

3. If stopping rule is met

   If not go to step 2.
Fixed cost transportation problem

- Complete problem

\[
\min \sum_{ij} (c_{ij} x_{ij} + f_{ij} y_{ij})
\]

\[
\sum_{j} x_{ij} \leq a_i \quad \forall i
\]

\[
\sum_{i} x_{ij} \geq b_j \quad \forall j
\]

\[
x_{ij} \leq M_{ij} y_{ij} \quad \forall ij
\]

\[
x_{ij} \geq 0, y_{ij} \in \{0,1\}
\]
Fixed cost transportation problem

- **Master**

\[
\begin{align*}
\min \theta & + \sum_{ij} f_{ij} y_{ij} \\
\delta^l \theta & + \sum_{ij} \pi^l_{ij} M_{ij} y_{ij} \geq f^l & + \sum_{ij} \pi^l_{ij} M_{ij} y^l_{ij} & \quad l = 1, \ldots, k \\
y_{ij} & \in \{0, 1\}
\end{align*}
\]

- **Subproblem**

\[
\begin{align*}
\min_{ij} & \sum_{ij} c_{ij} x_{ij} \\
\sum_{j} x_{ij} & \leq a_i & \forall i \\
\sum_{i} x_{ij} & \geq b_j & \forall j \\
x_{ij} & \leq M_{ij} y_{ij}^k & \forall ij : \pi_{ij}^k \\
x_{ij} & \geq 0
\end{align*}
\]
Case study. Solution

- Possible arcs

- Solutions along decomposition
Case study. Convergence

- Lower bound is not increasingly monotonous because there are infeasible iterations

<table>
<thead>
<tr>
<th>Iteración</th>
<th>Cota Inferior</th>
<th>Cota Superior</th>
</tr>
</thead>
<tbody>
<tr>
<td>1 a 6</td>
<td>$-\infty$</td>
<td>$\infty$</td>
</tr>
<tr>
<td>7</td>
<td>140</td>
<td>390</td>
</tr>
<tr>
<td>8</td>
<td>140</td>
<td>390</td>
</tr>
<tr>
<td>9</td>
<td>120</td>
<td>390</td>
</tr>
<tr>
<td>10</td>
<td>360</td>
<td>380</td>
</tr>
<tr>
<td>11</td>
<td>360</td>
<td>380</td>
</tr>
<tr>
<td>12</td>
<td>360</td>
<td>380</td>
</tr>
<tr>
<td>13</td>
<td>360</td>
<td>380</td>
</tr>
<tr>
<td>14</td>
<td>360</td>
<td>380</td>
</tr>
<tr>
<td>15</td>
<td>380</td>
<td>380</td>
</tr>
</tbody>
</table>
Convergence of hydrothermal scheduling model
Achtung! Achtung!

- **Degeneration in LP problems**
  - In real cases it is frequent to find multiple optima (degeneration in dual problem) with the same or different basis. Given that decomposition techniques are based on dual variables you must be very careful in its computation.
  - For example, in hydrothermal scheduling model formulated as LP it can exist spatial degeneration (system can produce with one plant or another) and temporal (system can produce now or in the future).
Primal degeneration

- Variable $x_6$ is degenerated (basic variable with value 0)

\[
\begin{align*}
\min z &= -3x_1 - 5x_2 \\
2x_2 + x_4 &= 12 \\
3x_1 + 2x_2 + x_5 &= 18 \\
x_1 + x_2 + x_6 &= 8 \\
x_1, x_2, x_3, x_4, x_5, x_6 &\geq 0
\end{align*}
\]
1. General overview
2. Two-stage and multistage programming
3. Decomposition techniques
4. Benders' decomposition
5. **Nested Benders’ decomposition**
6. Dantzig-Wolfe decomposition
7. Lagrangian relaxation
8. Scenario tree
9. Decomposition in two-stage and multistage stochastic programming
10. Improvements in decomposition techniques
11. Simulation in stochastic optimization
12. Stochastic dual dynamic programming

**Nested Benders’ decomposition**
Nested Benders’ Decomposition (i)

- Recursive application of decomposition technique.
- Let us see the PL-P problem:

\[
\begin{align*}
\min_{x_p} & \sum_{p=1}^{P} c_p^T x_p \\
B_{p-1} x_{p-1} + A_p x_p &= b_p & p = 1, \ldots, P \\
x_p &\geq 0 \\
B_0 &\equiv 0
\end{align*}
\]

- We apply Benders’ decomposition:
  - Stage 1 master, stage 2 to \( P \) subproblem
  - We decompose the subproblem that begins in stage 2
    - Stage 2 master, stage 3 to \( P \) subproblem
    - We decompose the subproblem that begins in stage 3
      - Stage 3 master, stage 4 to \( P \) subproblem
      - We decompose the subproblem that begins in stage 4
Nested Benders’ Decomposition (ii)

- In stage \( p \)
  - The problem of this stage is solved
  - As a master it receives cuts from \( p+1 \) and passes the solution to \( p+1 \),
  - As a subproblem it builds cuts from \( p-1 \) and receives the solution from \( p-1 \).
Nested Benders’ Decomposition

• Generic problem to solve

\[
\begin{align*}
\min_{\theta_{p+1}} & \quad c^T \! x_p + \theta_{p+1} \\
A_p x_p & = b_p - B_{p-1} x_{p-1} \\
\pi^T_p B_p x_p + \theta_{p+1} & \geq q_p = \pi^T_{p+1} b_{p+1} + \eta^T_{p+1} q_{p+1} \\
x_p & \geq 0 \\
\theta_{p+1} & \equiv 0 \\
B_0 & \equiv 0 \\
\pi^j_{p+1} & \equiv 0 \\
\eta^j_{p+1} & \equiv 0 
\end{align*}
\]

\[
\begin{align*}
\min_{\theta_{p+1}} & \quad c^T \! x_p + \theta_{p+1} \\
A_p x_p & = b_p - B_{p-1} x_{p-1} \\
\pi^T_p B_p x_p + \theta_{p+1} & \geq \xi_{p+1}^l + \eta^T_{p+1} B_p x_{p+1}^l \\
x_p & \geq 0 \\
\theta_{p+1} & \equiv 0 \\
B_0 & \equiv 0 \\
\pi^j_{p+1} & \equiv 0 \\
\eta^j_{p+1} & \equiv 0 
\end{align*}
\]

• Problem converges when first stage does
1. General overview
2. Two-stage and multistage programming
3. Decomposition techniques
4. Benders' decomposition
5. Nested Benders' decomposition
6. Dantzig-Wolfe decomposition
7. Lagrangian relaxation
8. Scenario tree
9. Decomposition in two-stage and multistage stochastic programming
10. Improvements in decomposition techniques
11. Simulation in stochastic optimization
12. Stochastic dual dynamic programming

Decomposition in two-stage and multistage stochastic programming
Two-stage stochastic linear programming PLE-2

- O.F. minimizes first-stage costs and expected value of second-stage costs

\[
\begin{align*}
\min_{x_1, x_2} & \quad c_1^T x_1 + \sum_{\omega \in \Omega} p^\omega c_2^\omega x_2^\omega \\
A_1 x_1 & = b_1 \\
B_1^\omega x_1 + A_2^\omega x_2^\omega & = b_2^\omega \\
x_1, x_2^\omega & \geq 0
\end{align*}
\]

- If \( A_2^\omega \) doesn’t depend on \( \omega \) it is called fixed resource
- Structure of the constraint matrix

\[
\begin{pmatrix}
A_1 & B_1^1 & A_2^1 \\
B_2^1 & A_2^2 \\
B_3^1 & A_2^3
\end{pmatrix}
\]
Deterministic equivalent problem

- **State space is small**
- **Formulation of the deterministic equivalent problem**

\[
\begin{align*}
\min_{x_1, x_2, x_3} & \quad c_1^T x_1 + p_1^1 c_2^1 x_2^1 + p_1^2 c_2^2 x_2^2 + p_1^3 c_2^3 x_2^3 \\
& \quad A_1 x_1 = b_1 \\
& \quad B_1^1 x_1 + A_2^1 x_2^1 = b_2^1 \\
& \quad B_1^2 x_1 + A_2^2 x_2^2 = b_2^2 \\
& \quad B_1^3 x_1 + A_2^3 x_2^3 = b_2^3 \\
& \quad x_1, x_2^1, x_2^2, x_2^3, \geq 0
\end{align*}
\]

- **In Benders’ decomposition subproblem results separable and has the same structure in the constraints**
Decomposition in PLE-2

- **Master monocut**

\[
\begin{align*}
\min_{x_1, \theta_2} & \quad c_1^T x_1 + \theta_2 \\
A_1 x_1 & = \quad b_1 \\
\sum_{\omega \in \Omega} p_\omega \pi_2^{\omega T} B_1^\omega x_1 + \theta_2 & \geq \sum_{\omega \in \Omega} p_\omega \pi_2^{\omega T} b_2^\omega & l = 1, \ldots, j \\
x_1 & \geq \quad 0
\end{align*}
\]

- **Subproblem**

\[
\begin{align*}
\min_{x_2^\omega} & \quad c_2^T x_2^\omega \\
A_2^\omega x_2^\omega & = \quad b_2^\omega - B_1^\omega x_1^\omega \\
x_2^\omega & \geq \quad 0
\end{align*}
\]

and **multicut**

\[
\begin{align*}
\min_{x_1, \theta_2} & \quad c_1^T x_1 + \sum_{\omega \in \Omega} p_\omega \theta_2^\omega \\
A_1 x_1 & = \quad b_1 \\
\pi_2^{\omega T} B_1^\omega x_1 + \theta_2^\omega & \geq \pi_2^{\omega T} b_2^\omega & \omega \in \Omega & l = 1, \ldots, j \\
x_1 & \geq \quad 0
\end{align*}
\]
Monocut vs. multicut

- Monocut \((n_1+1) \times (m_1+j)\).
- Multicut \((n_1+\Omega) \times (m_1+j\Omega)\).
- Multicut convenient when \(m_2\) is large and \(\Omega\) no much larger than \(n_1\). Requires less Benders iterations but more cumbersome.
- Multicut approximates independently each scenario. Monocut approximates the weighted sum of scenarios.
Multistage stochastic linear programming PLE-P

- O.F. minimizes expected costs of all the stages

\[
\min_{x_p^p} \sum_{p=1}^{P} \sum_{\omega_p \in \Omega_p} p_{\omega_p}^p c_{\omega_p}^p T x_{\omega_p}^p
\]

\[
B_{p-1}^\omega x_{p-1}^\omega + A_p^\omega x_p^\omega = b_p^\omega \quad p = 1, \ldots, P
\]

\[
x_p^\omega \geq 0
\]

\[
B_0^\omega \equiv 0
\]

- Probabilities \( p_{\omega_p}^p \) are conditional
- Constraint matrix

[Diagram of constraint matrix]

Escuela Técnica Superior de Ingeniería ICAI
Multistage stochastic problem. Nested Benders’ decomposition
Stochastic multistage decomposition

Step 0  Set $J_t^\xi = J_t^\zeta = 0$. Set $\theta_t^\xi \equiv 0$ at the initial iteration

Step 1  **Forward pass:**
Repeat for $t = 1, \ldots, T$
  Repeat for each node $\xi_t$ of stage $t$
  Solve $(RP_t^\xi)$
  If feasible: obtain solution $x_t^\xi$
    If $t = 1$ obtain lower bound $\underline{z} = v(RP_1^\xi)$
    If infeasible: stop forward pass, set $T' = t$ and go to Step 4

Step 2  **Upper bound computation:**
Evaluate objective function of the complete problem with the primal solutions so far obtained. $\bar{z} = v(P)$

Step 3  (stopping rule)
If $\bar{z} - z < \text{tol}$ stop, $x_t^\xi$ is optimal solution, else go to Step 4

Step 4  **Backward pass**
Repeat for $t = T', \ldots, 1$
  Repeat for each node $\xi_t$ of stage $t$
  Solve $(RP_t^\xi)$
  If feasible: obtain objective $\theta_t^{\xi,i} = v(RP_t^\xi)$ and dual values $\pi_t^{\xi,i}$
    Augment $J_t^\xi = J_t^\zeta + 1$
    If infeasible: obtain sum of infeasibilities $\bar{\theta}_t^{\xi,i}$ and dual values $\bar{\pi}_t^{\xi,i}$
      Augment $J_t^\xi = J_t^\zeta + 1$
  Go to step 1
1. General overview
2. Two-stage and multistage programming
3. Decomposition techniques
4. Benders' decomposition
5. Nested Benders' decomposition
6. Dantzig-Wolfe decomposition
7. Lagrangian relaxation
8. Scenario tree
9. Decomposition in two-stage and multistage stochastic programming

10. Improvements in decomposition techniques
11. Simulation in stochastic optimization
12. Stochastic dual dynamic programming
Improvements in decomposition techniques (i)

• **Optimization method** used for the problems
  Subproblems solved many times with modifications. In Benders’ decomposition the master adds constraints and the subproblem change the constraint RHS.
  – **Simplex dual method** is the initial candidate. Try simplex method or interior point method.
  – **Warm start**: Use of previous bases (option BRATIO in GAMS).
  – Use of initial point taken from the deterministic equivalent problem for a scenario.

• **Advanced start** procedures generate preliminary cuts prior to initiating a formal Benders algorithm
Improvements in decomposition techniques (ii)

- **Tree traversing strategies**: Ways to traverse through the tree from the root to the leaves. The “best” tree traversing strategy will properly balance the quality of the cuts (and hence the lower bound) with the computational effort required to generate them
  - **Fast-pass**: from 1 to P and from P-1 to 1
  - **Shuffle**: solves the stage with largest error between lower and upper bounds. It is centered in final stages, never goes backward until the error of a stage is bounded (*fast-forward*)
  - **Cautious**: goes forward when the error in a stage is small enough. It is centered in initial stages, never goes forward until the error of a stage is bounded (*fast-backward*)

Improvements in decomposition techniques (iii)

• **Formulation and cut aggregation**
  – Linear or nonlinear type (*linearization around a point*)
  – **Monocut or multicut**
  – More cuts \(\Rightarrow\) more information to the master \(\Rightarrow\) less iterations. On the other hand, more variables and more constraints in the master
Improvements in decomposition techniques (iv)

- Tree partition or node aggregation (multicoordination)
  - Advantage: reduction in decomposition algorithm iterations
  - Disadvantage: potential increase in problem solution time (interior point method)
  - Methods
    - By nodes
    - By scenarios
    - By subtrees
    - By complete scenarios
    - By graph partition
Node and scenario partition
**Subtree partition**

- **Ascendant node aggregation** (from the leaves to the root) is the one with the best performance

Decomposition in grid computing

• Distributed computing

• GAMS grid
  – Use of multiple cores of a computer
Multistage stochastic integer programming PLE-P

- Decomposition for multistage problems with integer variables in each stage
1. General overview
2. Two-stage and multistage programming
3. Decomposition techniques
4. Benders' decomposition
5. Nested Benders’ decomposition
6. Dantzig-Wolfe decomposition
7. Lagrangian relaxation
8. Scenario tree
9. Decomposition in two-stage and multistage stochastic programming
10. Improvements in decomposition techniques
11. Simulation in stochastic optimization
12. Stochastic dual dynamic programming

Simulation in stochastic optimization
Why do we need simulation?

- It is used when the number of states of random parameters too high.
- Computation of expectation in the recourse function (multic和平) or expectation in the cut terms (monocut).
- Equivalent to integrate or sample in the random parameter hyperspace with known probability density function. A sample is a combination of random parameter values.
- Each sample is computationally cumbersome (solving an LP problem).
- Simulate is equivalent to integrate or sample in hyperspace of random parameters with a known probability density function.
Types of sampling

• **External** sampling
  – We take samples to reduce the problem size and then we solve the stochastic optimization problem
    • In SDDP we take samples in the forward pass

• **Internal** sampling
  – We take samples at the same time that we solve the stochastic optimization problem
    • In a two-stage planning problem with the expected value for the second state substituted by the sample mean of the second stage
1. General overview
2. Two-stage and multistage programming
3. Decomposition techniques
4. Benders' decomposition
5. Nested Benders' decomposition
6. Dantzig-Wolfe decomposition
7. Lagrangian relaxation
8. Scenario tree
9. Decomposition in two-stage and multistage stochastic programming
10. Improvements in decomposition techniques
11. Simulation in stochastic optimization
12. Stochastic dual dynamic programming
Stochastic Dual Dynamic Programming (SDDP)

- Nested Benders’ decomposition with
  - Forward pass: node sampling instead of solving all the nodes
  - Backward pass: solution of all the nodes of the recombining tree. It approximates for each scenario the recourse function for the sampled values obtained in the forward iteration

- Therefore we have stochastic convergence
  - Lower bound is deterministic while upper bound is stochastic
  - Stopping criterion: lower bound enters the confidence interval of the upper bound
  - I.e., if the stopping criterion is 1% for a confidence level of 95%. The algorithm stops with we are sure with a 95% that the relative difference between upper and lower bounds is lower than 1%
Stochastic Dual Dynamic Programming SDDP

Step 0
Set $I^0_i = J^0_i = 0$. Set $\theta^0_i \equiv 0$ at the initial iteration.

Step 1
Simulate $N$ scenarios $h^\xi_i$, $n : 1, \ldots, N$, $t = 1, \ldots, T$.

Forward pass:
Repeat for $n : 1, \ldots, N$:
Repeat for $t = 1, \ldots, T$:
Solve $(RP^\xi_t)$ with right hand side value $(h^\xi_t)^n$ and obtain solution $(x^\xi_t)^n$.

If $t = 1$ obtain lower bound $\bar{z} = v(RP^\xi_1)$.
If infeasible: stop forward pass for simulation $n$.

Step 2
Upper bound computation:
Evaluate objective function of the complete (deterministic) problem for each of the primal solutions so far obtained.
$\bar{z} = \frac{1}{N} \sum_{i=1}^{T} c_i (x^\xi_t)^n$.

Step 3
(stopping rule)
If $\bar{z} - \bar{z} < \text{tol}$ stop, $x^\xi_t$ is optimal solution, else go to Step 4.

Step 4
Backward pass
Repeat for $t = T, \ldots, 1$:
Repeat for each node $\xi_t$ of stage $t$:
Repeat for each proposal obtained in forward pass, modifying the right hand side value of subproblem $(RP^\xi_t)$:
Solve $(RP^\xi_t)$.
If feasible: obtain objective $\theta^\xi,i = v(RP^\xi_t)$ and dual values $\pi^\xi,i$.
Augment $I^\xi_t = I^\xi_t + 1$.
If infeasible: obtain sum of infeasibilities $\bar{\theta}^\xi,i$ and dual values $\bar{\pi}^\xi,i$.
Augment $J^\xi_t = J^\xi_t + 1$.

Go to step 1.
Stochastic convergence in SDDP (i)

The first 100 iterations are ignored, needed to obtain a small confidence interval.

Stop the algorithm if the lower bound enters into the confidence interval of the upper bound for a confidence level of 95%.

Average upper bound of the last 100 iterations.
Stochastic convergence in SDDP (ii)
Stochastic convergence in SDDP (ii) (detail)
## Comparison of decomposition methods

<table>
<thead>
<tr>
<th></th>
<th>Benders</th>
<th>SDDP</th>
</tr>
</thead>
<tbody>
<tr>
<td>Suitable for stochastic problems with tree structure</td>
<td>Suitable for stochastic problems where stochasticity is introduced independently by scenarios</td>
<td></td>
</tr>
<tr>
<td>Scenario tree with no fixed structure (symmetrical or non symmetrical)</td>
<td>Recombining scenario tree (dependence of one scenario with respect to other is modeled by transition probabilities)</td>
<td></td>
</tr>
<tr>
<td>Solution of the deterministic equivalent problem</td>
<td>Impossible to solve the deterministic equivalent problem</td>
<td></td>
</tr>
<tr>
<td>Multicut</td>
<td>Multicut</td>
<td></td>
</tr>
<tr>
<td>Flexible node aggregation (tree partition)</td>
<td>Rigid node aggregation (tree partition) conditioned by the branching periods</td>
<td></td>
</tr>
<tr>
<td>In forward pass all the scenarios are solved</td>
<td>In forward pass only one scenario is solved</td>
<td></td>
</tr>
<tr>
<td>Deterministic stopping criterion</td>
<td>Stochastic stopping criterion</td>
<td></td>
</tr>
<tr>
<td>Exponential time increase with number of scenarios</td>
<td>Linear time increase with number of scenarios</td>
<td></td>
</tr>
</tbody>
</table>
Prof. Andres Ramos

http://www.iit.upcomillas.es/aramos/

Andres.Ramos@upcomillas.es

arght@mit.edu