Stochastic Optimization

Prof. Andres Ramos

https://www.iit.comillas.edu/aramos/
Andres.Ramos@comillas.edu
arght@mit.edu

December 2017
References


Other resources

- Stochastic Programming Society (https://www.stoprog.org/)
- Stochastic Programming Resources (www, papers, tutorials, lecture notes, books, software) (https://www.stoprog.org/resources)
- 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)
- International Conference in Stochastic Programming (ICSP)
1. General overview
2. Applications in electric systems
3. Two-stage and multistage programming
4. Decomposition techniques
5. Benders decomposition
6. Nested Benders decomposition
7. Dantzig-Wolfe decomposition
8. Lagrangian relaxation
9. Scenario tree
10. Decomposition in two-stage and multistage stochastic programming
11. Improvements in decomposition techniques
12. Simulation in stochastic optimization
13. Stochastic dual dynamic programming
Stochasticity or uncertainty

• Origin
  – Future information (prices or future demand)
  – Lack of reliable data
  – Measurement errors

• In electric energy systems planning
  – Demand (seasonal/daily variation, yearly, increment over time)
  – Hydro inflows
  – Availability of generation and network elements
  – Electricity or fuel prices

• Uncertainty relevant for each time scale
Decision under uncertainty

- **DETERMINISTIC** optimization
  - Best decision when future is known

- Simulation. Scenario analysis
  - What could happen if ...

- **STOCHASTIC** optimization
  - Best decision when future is uncertain but with a known probability
Deterministic vs. Stochastic Optimization

- **Deterministic**
  - Parameters known with certainty (it can be the mean value)

- **Stochastic**
  - Parameters modeled as stochastic variables with known distributions
    - Historical
    - Discrete
    - Continuous ⇒ simulation
Example: Newsvendor

- It is a mathematical model in operations management and applied economics used to determine optimal inventory levels. It is (typically) characterized by fixed prices and uncertain demand. If the inventory level is \( q \), each unit of demand above \( q \) is lost. This model is also known as the Newsvendor Problem or Newsboy Problem by analogy with the situation faced by a newspaper vendor who must decide how many copies of the day's paper to stock in the face of uncertain demand and knowing that unsold copies will be worthless at the end of the day.

The following two-stage problem consists of determining the optimal capacity investment in various types of power plants so as to meet next period demands for electricity. Four power plants are considered and they can operate in three different modes. The next period demand for each of the three modes are to be met. There is a budget constraint and also a constraint on the minimum total capacity.

- **Stages**
  - **First stage**: investment decisions
  - **Second stage**: operation decisions

Example: Generation expansion planning

http://www.siemens.com/
Example: Generation expansion planning

- Time scope divided into **three periods**
- **3 Stochastic demand scenarios** with change only in the first period
- Several type of generators available
- **Expansion decisions** must be **unique** for all the scenarios

Variables
- Generation expansion
- Unit output

Constraints
- Maximum budget available
- Minimum capacity to be installed
- Load-generation balance for each scenario
- Power output lower than the installed capacity
Generation expansion planning model (i)

$TITLE Optimal Generation Expansion Planning Problem

sets
  I generators / gen-1 * gen-4 /
  J periods / per-1 * per-3 /
  S demand scenarios / scen-1 * scen-3 /
Generation expansion planning model (ii)

parameters

\[ F(i) \] fixed investment cost [€ per MW]

\[
\begin{array}{l}
/ \text{gen-1} 10 \\
/ \text{gen-2} 7 \\
/ \text{gen-3} 16 \\
/ \text{gen-4} 6 \\
\end{array}
\]

\[ \text{PROB(s)} \] scenario probability [p.u.]

\[
\begin{array}{l}
/ \text{scen-1} 0.2 \\
/ \text{scen-2} 0.5 \\
/ \text{scen-3} 0.3 \\
\end{array}
\]

\[ \text{DEM(j)} \] scenario load [MW]

\[
\begin{array}{l}
/ \text{per-1} 3 \text{ scen-1} 5 \text{ scen-2} 7 \text{ scen-3} \\
/ \text{per-2} 3 \text{ scen-1} 3 \text{ scen-2} 3 \text{ scen-3} \\
/ \text{per-3} 2 \text{ scen-1} 2 \text{ scen-2} 2 \text{ scen-3} \\
\end{array}
\]

\[ \text{CAPMIN} \] minimum capacity to install [MW]

/ 12 /

\[ \text{BUDGLM} \] budget limit [€]

/ 120 /

\[
\begin{array}{l}
\text{table } V(i,j) \text{ variable operation cost [€ per MW]}
\end{array}
\]

\[
\begin{array}{l}
/ \text{gen-1} 40 \text{ per-1} 24 \text{ per-2} 4 \\
/ \text{gen-2} 45 \text{ per-1} 27 \text{ per-2} 4.5 \\
/ \text{gen-3} 32 \text{ per-1} 19.2 \text{ per-2} 3.2 \\
/ \text{gen-4} 55 \text{ per-1} 33 \text{ per-2} 5.5 \\
\end{array}
\]

\[
\begin{array}{l}
\text{table } \text{DEMS(s,j) stochastic demand [MW]}
\end{array}
\]

\[
\begin{array}{l}
/ \text{per-1} 3 \text{ scen-1} 5 \text{ scen-2} 7 \text{ scen-3} \\
/ \text{per-2} 3 \text{ scen-1} 3 \text{ scen-2} 3 \text{ scen-3} \\
/ \text{per-3} 2 \text{ scen-1} 2 \text{ scen-2} 2 \text{ scen-3} \\
\end{array}
\]
Generation expansion planning model (iii)

<table>
<thead>
<tr>
<th>Variables</th>
<th>Description</th>
<th>Unit</th>
</tr>
</thead>
<tbody>
<tr>
<td>$X(i)$</td>
<td>installed capacity</td>
<td>MW</td>
</tr>
<tr>
<td>$Y(j,i)$</td>
<td>operation output</td>
<td>MW</td>
</tr>
<tr>
<td>$YS(s,j,i)$</td>
<td>stochastic operation output</td>
<td>MW</td>
</tr>
<tr>
<td>TCOST</td>
<td>total cost</td>
<td>€</td>
</tr>
</tbody>
</table>

positive variables $X$, $Y$, $YS$
Generation expansion planning model (iv)

**equations**

<table>
<thead>
<tr>
<th>COST</th>
<th>total cost</th>
<th>[€]</th>
</tr>
</thead>
<tbody>
<tr>
<td>COSTS</td>
<td>stochastic total cost</td>
<td>[€]</td>
</tr>
<tr>
<td>BUDGET</td>
<td>budget limit</td>
<td>[€]</td>
</tr>
<tr>
<td>MININST</td>
<td>minimum capacity to install</td>
<td>[MW]</td>
</tr>
<tr>
<td>OUTPIN</td>
<td>power output &lt; installed</td>
<td>[MW]</td>
</tr>
<tr>
<td>OUTPINS</td>
<td>power output &lt; installed stochastic</td>
<td>[MW]</td>
</tr>
<tr>
<td>LOADBAL</td>
<td>load balance</td>
<td>[MW]</td>
</tr>
<tr>
<td>LOADBALS</td>
<td>stochastic load balance</td>
<td>[MW]</td>
</tr>
</tbody>
</table>

COST  ..  TCOST  =  \( \sum_i F(i) \times X(i) \) + \( \sum_{(j,i)} V(i,j) \times Y(j,i) \);

COSTS .. TCOST  =  \( \sum_i F(i) \times X(i) \) + \( \sum_{(s,j,i)} \text{PROB}(s) \times V(i,j) \times YS(s,j,i) \);

BUDGET  ..  \( \sum_i F(i) \times X(i) \) = \( \leq \text{BUDGLM} \);

MININST .. \( \sum_i X(i) \) = \( \geq \text{CAPMIN} \);

OUTPIN (j,i) .. \( Y(j,i) = \leq X(i) \);

OUTPINS(s,j,i) .. \( YS(s,j,i) = \leq X(i) \);

LOADBAL (j) .. \( \sum_i Y(j,i) = \geq \text{DEM}(j) \);

LOADBALS(s,j) .. \( \sum_i YS(s,j,i) = \geq \text{DEMS}(s,j) \);
Generation expansion planning model (v)

model DETERMINISTIC / COST, MININST, BUDGET, OUTPIN, LOADBAL /
model STOCHASTIC    / COSTS, MININST, BUDGET, OUTPINS, LOADBALS /

* Each deterministic scenario
  loop (s,
    DEM(j) = DEMS(s,j)
    solve DETERMINISTIC minimizing TCOST using LP
  ) ;

* Expected load scenario
  DEM(j) = sum[s, PROB(s) * DEMS(s,j)]
  solve DETERMINISTIC minimizing TCOST using LP

* Stochastic problem
  solve STOCHASTIC minimizing TCOST using LP
Analysis of the generation expansion results

- Deterministic (perfect information) decisions not necessarily appear in the optimal stochastic solution
- Optimal stochastic solution doesn’t appear necessarily in any deterministic scenario

<table>
<thead>
<tr>
<th></th>
<th>Det 1</th>
<th>Det 2</th>
<th>Det 3</th>
<th>Mean</th>
<th>Stochast</th>
</tr>
</thead>
<tbody>
<tr>
<td>Gen 1 [MW]</td>
<td>.</td>
<td>0.33</td>
<td>3.67</td>
<td>0.67</td>
<td>0.67</td>
</tr>
<tr>
<td>Gen 2 [MW]</td>
<td>.</td>
<td>.</td>
<td>.</td>
<td>.</td>
<td>2</td>
</tr>
<tr>
<td>Gen 3 [MW]</td>
<td>3</td>
<td>4.67</td>
<td>3.33</td>
<td>4.53</td>
<td>4.33</td>
</tr>
<tr>
<td>Gen 4 [MW]</td>
<td>9</td>
<td>7</td>
<td>5</td>
<td>6.8</td>
<td>5</td>
</tr>
<tr>
<td>Total Cost [€]</td>
<td>262</td>
<td>346.67</td>
<td>437.33</td>
<td>355.73</td>
<td>362.47</td>
</tr>
</tbody>
</table>
Alternatives for modeling the uncertainty

- **Wait and see o scenario analysis o what-if analysis o sensitivity analysis**
  - Decisions are taken once solved the uncertainty
  - The problem is solved independently for each scenario
  - The scenario with mean value of the parameters is just an special case
  - A priori, decisions will be different for each deterministic scenario
    (anticipative, clairvoyant, not implementable)
  - Solution of an scenario can be infeasible in the others

- **Here and now (stochastic) decisions**
  - Decisions have to be taken before solving uncertainty
  - Non anticipative decisions (only the available information so far can be used, no future information)
  - The only relevant decisions are those of the first stage, given that are the only to be taken immediately
  - Stochastic solution takes into account the stochasticity distribution
  - It allows to include risk averse attitudes, penalizing worst cases
Definitions in stochastic problems. Stochastic measures

- **Expected value with perfect information (EVWPI) o Wait and See (WS)**
  - Weighted mean of the objective function of each scenario knowing that is going to happen (356.93 for the example) (for minimization problems always ≤ the objective function for the stochastic problem, 280, 349.33 and 439.33 respectively, that give a weighted value of 362.47)

- **Value of the stochastic solution (VSS) o Expected Value of Including Uncertainty (EVIU)**
  - Difference between the expected objective function for the mean-value solution of the stochastic parameters Expectation of EV Problem (EEV) (280x0.2+347.73x0.5+454.73x0.3=366.28) and that of the stochastic problem RP (366.28–362.47=3.81)

$$VSS = EEV - RP$$

$VSS \geq 0$

- **Expected value of perfect information (EVPI) o mean regret**
  - Weighted average of the difference between the stochastic solution for each scenario and the perfect information solution in this scenario (always positive for minimization) (280–262=18, 349.33–346.67=2.66, 439.33–437.33=2) (18x0.2+2.66x0.5+2x0.3=5.54)

$$EVPI = RP - WS$$

$EVPI \geq 0$

- $WS \leq RP \leq EEV$
Decision tree

- Represents a sequence of **decisions** and **random events** that constitute the decision process.
- Elements of the tree:
  - **Chance node**: random points, are represented by a **circle**.
  - **Decision node**: decision points, are represented by a **square**.
  - **Initial node or root**: root of the tree with the initial decisions. Always the first thing to do is to take a decision.
  - **Final node or leaf**: finals points, are represented by triangles.
Example: Manufacturing. **Two-stage stochastic problem**

- Manufacturing decision of an amount of a product with a cost of 2 €/unit to satisfy a random demand. If no enough product is manufactured it can be **bought** from an external provider at a price of 4 €/unit.

\[
\begin{align*}
\text{min} & \quad 2F_1 + 4C_2^1 + 0.3C_2^2 + 0.5C_2^2 + 0.2C_2^3 \\
F_1 + C_2^1 & \geq 400 \\
F_1 + C_2^2 & \geq 500 \\
F_1 + C_2^3 & \geq 600 \\
F_1, C_2^1, C_2^2, C_2^3 & \geq 0
\end{align*}
\]
Two-stage stochastic problem

1. **Today**, a set of first-stage decisions are taken

2. **At night**, some (exogenous) random events occur

3. **Tomorrow** a set of corrective actions in the second stage are taken to mitigate (fix) the effects of the random events over the today decisions. Second stage decisions are the recourses.
Recourse. Recourse function

- Capability of taking a **corrective action after** occurring a random event
- **Recourse function**: objective function associated to the corrective actions.
- Depends on the **previous decisions** and on the **random events**.
- If stages are in the time domain the recourse function is the **future cost function (cost-to-go function)**
- Two-stage stochastic linear optimization (Generation expansion planning):
  - First stage decisions are deterministic, unique (investment)
  - Second stage (recourse) decisions are stochastic (operation)
Type of recourse

- **Complete**
  - All the first-stage decisions are feasible for any second-stage scenario

- **Relatively complete**
  - All the first-stage feasible decisions are feasible for any second-stage scenario

- **Partial**
Penalizing the constraint violation

- Slack (deficit and surplus) variables are introduced in the stochastic constraints and are penalized in the objective function
Example: Manufacturing. **Two-stage stochastic problem**

$$
\begin{align*}
\min & \quad 2F_1 + 4(0.3C_2^1 + 0.5C_2^2 + 0.2C_2^3) \\
\text{s.t.} & \quad F_1 + C_2^1 \geq 400 \\
& \quad F_1 + C_2^2 \geq 500 \\
& \quad F_1 + C_2^3 \geq 600 \\
& \quad F_1, C_2^1, C_2^2, C_2^3 \geq 0
\end{align*}
$$

$$
\begin{align*}
\min & \quad 2(0.3F_1^1 + 0.5F_1^2 + 0.2F_1^3) + 4(0.3C_2^1 + 0.5C_2^2 + 0.2C_2^3) \\
\text{s.t.} & \quad F_1^1 = F_1^2 \\
& \quad F_1^2 = F_1^3 \\
& \quad F_1^1 + C_2^1 \geq 400 \\
& \quad F_1^2 + C_2^2 \geq 500 \\
& \quad F_1^3 + C_2^3 \geq 600 \\
& \quad F_1^1, F_1^2, F_1^3, C_2^1, C_2^2, C_2^3 \geq 0
\end{align*}
$$

**Non anticipativity constraints**
Tree representation

- **Implicit formulation** in the definition of parameters and variables
  - Decomposition technique (Benders, etc.)

- **Explicit formulation** by constraints
  - *Scenario decomposition (splitting variables, non-anticipativity constraints)* and Lagrangian relaxation
Example: Manufacturing. **Three-stage stochastic problem.**

**Decision tree**

- **Decision**
  - Uncertainty revealed
- **Decision**
  - Uncertainty revealed

![Decision Tree Diagram](image-url)
Example: Manufacturing. **Three-stage stochastic problem**

\[
\begin{align*}
\min & \quad 2F_1 + 4(0.3C_2^1 + 0.5C_2^2 + 0.2C_2^3) + 2(0.3F_1^1 + 0.5F_2^2 + 0.2F_2^3) + \\
& \quad 4(0.12C_3^1 + 0.15C_3^2 + 0.03C_3^3 + 0.15C_3^4 + 0.25C_3^5 + \\
& \quad 0.1C_3^6 + 0.08C_3^7 + 0.08C_3^8 + 0.04C_3^9) \\
\end{align*}
\]

\[
\begin{align*}
F_1 + C_2^1 & \geq 400 \\
F_1 + C_2^2 & \geq 500 \\
F_1 + C_3^1 & \geq 600 \\
F_1 + C_2^1 - 400 + F_2^1 + C_3^1 & \geq 200 \\
F_1 + C_2^2 - 400 + F_2^1 + C_3^2 & \geq 400 \\
F_1 + C_2^1 - 400 + F_2^1 + C_3^3 & \geq 550 \\
F_1 + C_2^1 - 500 + F_2^1 + C_3^4 & \geq 300 \\
F_1 + C_2^1 - 500 + F_2^1 + C_3^5 & \geq 500 \\
F_1 + C_2^1 - 500 + F_2^1 + C_3^6 & \geq 700 \\
F_1 + C_2^1 - 600 + F_2^1 + C_3^7 & \geq 600 \\
F_1 + C_2^1 - 600 + F_2^1 + C_3^8 & \geq 800 \\
F_1 + C_2^1 - 600 + F_2^1 + C_3^9 & \geq 900 \\
F_1, C_2^1, C_2^2, C_2^3, F_1^1, F_2^2, F_3^1, C_3^1, F_3^2, C_3^2, C_3^3, C_3^4, C_3^5, C_3^6, C_3^7, C_3^8, C_3^9 & \geq 0
\end{align*}
\]

\[
\begin{align*}
F_1^* & = 600 \\
C_2^* & = C_2^* = C_2^* = 0 \\
F_2^* & = 200; F_2^* = 400; F_2^* = 800 \\
C_3^* & = C_3^* = C_3^* = C_3^* = C_3^* = C_3^* = 0 \\
C_3^* & = 150; C_3^* = 200; C_3^* = 100
\end{align*}
\]
Probability or scenario tree

- Represents the evolution in realization of uncertainty along the time, different values of the random parameters along the time.
- **Scenario**: any path from the root to the leafs
- The scenarios that share information up to a certain time period share the same decisions in the tree (**implementable decisions**)
- The probability tree represents the dynamics of the random parameters and the non-anticipativity of the decisions and, therefore, is implicit in the constraint matrix
Example: Manufacturing. **Three-stage stochastic problem.**
Constraint matrix

|   | $F_1^1$ | $C_2^1$ | $C_2^2$ | $C_2^3$ | $F_2^1$ | $F_2^2$ | $F_2^3$ | $C_3^1$ | $C_3^2$ | $C_3^3$ | $C_3^4$ | $C_3^5$ | $C_3^6$ | $C_3^7$ | $C_3^8$ | $C_3^9$ |
|---|---------|---------|---------|---------|---------|---------|---------|---------|---------|---------|---------|---------|---------|---------|---------|
|   |         |         |         |         |         |         |         |         |         |         |         |         |         |         |         |         |
|   |         |         |         |         |         |         |         |         |         |         |         |         |         |         |         |         |
|   |         |         |         |         |         |         |         |         |         |         |         |         |         |         |         |         |
|   |         |         |         |         |         |         |         |         |         |         |         |         |         |         |         |         |
|   |         |         |         |         |         |         |         |         |         |         |         |         |         |         |         |         |
|   |         |         |         |         |         |         |         |         |         |         |         |         |         |         |         |         |
|   |         |         |         |         |         |         |         |         |         |         |         |         |         |         |         |         |
|   |         |         |         |         |         |         |         |         |         |         |         |         |         |         |         |         |
|   |         |         |         |         |         |         |         |         |         |         |         |         |         |         |         |         |
Example: Manufacturing. **Three-stage stochastic problem.**

Reordered constraint matrix

<table>
<thead>
<tr>
<th></th>
<th>$F_1$</th>
<th>$C_2^1$</th>
<th>$C_2^2$</th>
<th>$C_2^3$</th>
<th>$C_3^1$</th>
<th>$C_3^2$</th>
<th>$C_3^3$</th>
<th>$C_3^4$</th>
<th>$C_3^5$</th>
<th>$C_3^6$</th>
<th>$C_3^7$</th>
<th>$C_3^8$</th>
<th>$C_3^9$</th>
</tr>
</thead>
<tbody>
<tr>
<td>$F_1$</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>$C_2^1$</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>$C_2^2$</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>$C_2^3$</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>$C_3^1$</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>$C_3^2$</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>$C_3^3$</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>$C_3^4$</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>$C_3^5$</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>$C_3^6$</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>$C_3^7$</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>$C_3^8$</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>$C_3^9$</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Stochastic Optimization
2017-2018
Example: Manufacturing. Three-stage stochastic problem. Problem reformulation

\[
\begin{align*}
\min & \quad 2F_1 + 4(C_2^1 + 0.5C_2^2 + 0.2C_2^3) + 2(0.3F_2^1 + 0.5F_2^2 + 0.2F_2^3) + \\
& \quad + 4(C_3^1 + 0.12C_3^2 + 0.03C_3^3 + 0.15C_3^4 + 0.25C_3^5) + \\
& \quad + 0.1C_3^6 + 0.08C_3^7 + 0.08C_3^8 + 0.04C_3^9) \\
& \quad F_1 + C_2^1 \geq 400 \\
& \quad F_1 + C_2^2 \geq 500 \\
& \quad F_1 + C_2^3 \geq 600 \\
& \quad F_1 + C_2^4 - 400 = E_2^1 \\
& \quad F_1 + C_2^5 - 500 = E_2^2 \\
& \quad F_1 + C_2^6 - 600 = E_2^3 \\
& \quad E_2^1 + F_2^1 + C_3^4 \geq 200 \\
& \quad E_2^1 + F_2^2 + C_3^5 \geq 400 \\
& \quad E_2^1 + F_2^3 + C_3^6 \geq 550 \\
& \quad E_2^2 + F_2^4 + C_3^7 \geq 300 \\
& \quad E_2^2 + F_2^5 + C_3^8 \geq 500 \\
& \quad E_2^2 + F_2^6 + C_3^9 \geq 700 \\
& \quad E_2^3 + F_2^7 + C_3^{10} \geq 600 \\
& \quad E_2^3 + F_2^8 + C_3^{11} \geq 800 \\
& \quad E_2^3 + F_2^9 + C_3^{12} \geq 900
\end{align*}
\]

An inventory variable is introduced at the start of period 2.
Example: Manufacturing. Three-stage stochastic problem. Constraint matrix

<table>
<thead>
<tr>
<th>$F_1$</th>
<th>$C_1^2$</th>
<th>$C_2^2$</th>
<th>$C_3^2$</th>
<th>$E_1^2$</th>
<th>$F_2^1$</th>
<th>$E_2^2$</th>
<th>$F_2^3$</th>
<th>$E_3^2$</th>
<th>$F_2^3$</th>
<th>$C_3^1$</th>
<th>$C_3^2$</th>
<th>$C_3^3$</th>
<th>$C_3^4$</th>
<th>$C_3^5$</th>
<th>$C_3^6$</th>
<th>$C_3^7$</th>
<th>$C_3^8$</th>
<th>$C_3^9$</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>$E_2^1$</td>
<td>$F_2^1$</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>$E_2^1$</td>
<td>$F_2^1$</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>$E_2^1$</td>
<td>$F_2^1$</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Separable scenarios
Fixed Cost Transportation Problem (FCTP)

- It is a transportation problem where the arc connecting two nodes \((i\) and \(j\)) has a **fixed cost** \(f_{ij}\) associated to its installation and a **variable cost** \(c_{ij}\) by the use. We want to minimize the total fixed (investment) and variable (transportation) costs subject to the constraints of demand supply \(b_j\) at the destinations and maximum capacity at the origins \(a_i\).
### Deterministic and stochastic FCTP

#### Deterministic

- **Minimize**
  \[ \min_{x_{ij}, y_{ij}} \sum_{ij} (f_{ij} y_{ij} + c_{ij} x_{ij}) \]

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

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

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

- \[ x_{ij} \geq 0, y_{ij} \in \{0,1\} \]

#### Stochastic

- **Minimize**
  \[ \min_{x_{ij}^\omega, y_{ij}} \sum_{ij} \left( f_{ij} y_{ij} + \sum_{\omega} p_{\omega} c_{ij} x_{ij}^\omega \right) \]

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

- \[ \sum_{i} x_{ij}^\omega \leq b_j^\omega \quad \forall j \omega \]

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

- \[ x_{ij}^\omega \geq 0, y_{ij} \in \{0,1\} \]
$title Deterministic fixed-charge transportation problem (DFCTP)

**relative optimality tolerance in solving MIP problems**

```
option OptcR = 0;
```

```
sets
I origins / i1 * i4 /
J destinations / j1 * j3 /
```

```
parameters
A(i) product offer / i1 20, i2 30, i3 40, i4 20 /
B(j) product demand / j1 20, j2 50, j3 30 /
```

```
table C(i,j) per unit variable transportation cost
  j1  j2  j3
i1   1   2   3
i2   3   2   1
i3   2   3   4
i4   4   3   2
```

```
table F(i,j) fixed transportation cost
  j1  j2  j3
i1  10  20  30
i2  20  30  40
i3  30  40  50
i4  40  50  60
```

```
abort $(sum[i, A(i)] < sum[j, B(j)]) 'Infeasible problem'
```

```
positive variable
X(i,j) arc flow
binary variable
Y(i,j) arc investment decision
variables
Z1 objective function
```

```
equations
EQ_OBJ complete problem objective function
Offer (i) offer at origin
Demand (j) demand at destination
FlowLimit(i,j) arc flow limit;
```

```
EQ_OBJ .. Z1 =e= sum[i, F(i,j)*Y(i,j)] + sum[i,j], C(i,j)*X(i,j) ;
Offer (i) .. sum[j, X(i,j)] =l= A(i) ;
Demand (j) .. sum[i, X(i,j)] =g= B(j) ;
FlowLimit(i,j) .. X(i,j) =l= min[A(i),B(j)] * Y(i,j) ;
```

```
model Complete / EQ_OBJ Offer Demand FlowLimit /;
X.up(i,j) = min[A(i),B(j)]
solve Complete using MIP minimizing Z1;
```

$title Stochastic fixed-charge transportation problem (SFCTP)

**relative optimality tolerance in solving MIP problems**

```
option OptcR = 0;
```

```
sets
I origins / i1 * i4 /
J destinations / j1 * j3 /
S scenarios / s1 * s3 /
```

```
parameters
A(i) product offer / i1 20, i2 30, i3 40, i4 20 /
P(s) scenario probability / s1 0.5, s2 0.3, s3 0.2 /
```

```
table B(s,j) product demand
  j1  j2  j3
s1  21  51  31
s2  32  22  52
s3  53  33  23
```

```
table C(i,j) per unit variable transportation cost
  j1  j2  j3
i1   1   2   3
i2   3   2   1
i3   2   3   4
i4   4   3   2
```

```
table F(i,j) fixed transportation cost
  j1  j2  j3
i1  10  20  30
i2  20  30  40
i3  30  40  50
i4  40  50  60
```

```
loop (s, abort $(sum[i, A(i)] < sum[j, B(s,j)]) 'Infeasible problem')
```

```
positive variable
X(s,i,j) arc flow
binary variable
Y(i,j) arc investment decision
variables
Z1 objective function
```

```
equations
EQ_OBJ complete problem objective function
Offer (s,i) offer at origin
Demand (s,j) demand at destination
FlowLimit(s,i,j) arc flow limit;
```

```
EQ_OBJ .. Z1 =e= sum[i,j], F(i,j)*Y(i,j)] + sum[i,j], P(s)*C(i,j)*X(s,i,j) ;
Offer (s,i) .. sum[j, X(s,i,j)] =l= A(i) ;
Demand (s,j) .. sum[i, X(s,i,j)] =g= B(s,j) ;
FlowLimit(s,i,j) .. X(s,i,j) =l= min[A(i),B(s,j)] * Y(i,j) ;
```

```
model Complete / EQ_OBJ Offer Demand FlowLimit /;
X.up(s,i,j) = min[A(s),B(s,j)]
solve Complete using MIP minimizing Z1;
```
$title Deterministic fixed-charge transportation problem (FCTP)

# relative optimality tolerance in solving MIP problems
option Optcr = 0

sets
I origins / i1 * i4 /     J destinations / j1 * j3 /

parameters
A(i) product offer
     / i1 20, i2 30, i3 40, i4 20 /
B(j) product demand
     / j1 11, j2 44, j3 66 /

table C(i,j) per unit variable transportation cost
j1  j2  j3
i1   1   2   3
i2   3   2   1
i3   2   3   4
i4   4   3   2

table F(i,j) fixed transportation cost
j1  j2  j3
i1  10  20  30
i2  20  30  40
i3  30  40  50
i4  40  50  60

positive variable
X(i,j) arc flow

binary variable
Y(i,j) arc investment decision

variables
Z1 objective function

equations
EQ_OBJ complete problem objective function
Offer (i) offer at origin
Demand (j) demand at destination
FlowLimit(i,j) arc flow limit

EQ_OBJ =.. Z1 =e= sum[(i,j), F(i,j)*Y(i,j)] + sum[(i,j), C(i,j)*X(i,j)];
offer (i) .. sum[j, X(i,j)] =l= A(i);
Demand (j) .. sum[i, X(i,j)] =g= B(j);
FlowLimit(i,j) .. X(i,j) =l= 100 * Y(i,j);
model Complete / all /;
x.up(i,j) = 100

set S scenarios / s1 * s3 /
parameter P(s) scenario probability / s1 0.5, s2 0.3, s3 0.2 /
YS(s,i,j) arc investment decision
XS(s,i,j) arc flow

table BS(s,j) product demand
j1  j2  j3
s1  21  51  31
s2  32  22  52
s3  53  33  23

file emp / '%emp.info%' / ; emp.pc=2
put emp
put "* problem %gams.1%' / "jrandvar 
loop (j,
   put B.tn(j) " ")
loop (s,
   put P(s)
   loop (j,
      put BS(s,j)
   )
   )
put / "stage 2 B X Offer Demand FlowLimit"
putclose emp

set dict / s . scenario . ''
   B . randvar . BS
   X . level . XS
   Y . level . YS /
loop (s, abort $(sum[i, A(i)] < sum[j, BS(s,j)]) 'Infeasible problem' ) ;
solve Complete minimizing Z1 using emp scenario dict
display XS, YS
Hydrothermal Scheduling

- The objective of the problem is to determine a generation schedule for a hydrothermal generating system such that expected operating costs are minimized. These costs are composed of fuel costs for the thermal units and penalties for failure to meet power demand. The hydrothermal system consists of four hydroelectric reservoirs and one thermal generation unit. Constraints to the problem are the mass balance equations for the water supply across the reservoirs and across the stages; the demand constraints and the capacity of the operating units. The schedule has to be developed for three time periods.
Example: Hydrothermal Scheduling Problem

• Scenario analysis (deterministic)
  – Run the model supposing that the natural inflows will be the same as **any of the previous historical inflows** (i.e., year 1989 or 2004, etc.) for the time scope
  – Run the model supposing that the natural inflows for each period will be exactly the **mean of the historical values** (i.e., average year) for the time scope

• Stochastic optimization
  – Run the model taking into account that the **distribution of future natural inflows** will be the same as it has been in the past
Multistage stochastic optimization

• Taking optimal decisions in different stages in presence of random parameters with known distributions to minimize the **expected value**

• **General formulation** of the stochastic optimization problem:

\[
\min_{x \in X} \mathbb{E}_\omega \{ f(\omega, x) \} = \min_{x \in X} \int_{\Omega} f(\omega, x) \cdot dP(\omega)
\]

• Uncertainty is represented by a **scenario tree**
Solving a stochastic model

- Stochastic parameters
- Scenario tree generation
- Stochastic optimization
- Stability of the stochastic solution?

YES

NO
Stability of the stochastic solution

- The main stochastic solutions (i.e., the first-stage ones) must be “the same” against the uncertainty modeling (structure and number of scenarios of the tree)
- A scenario tree must be generated such as the solution of the stochastic model ought to be independent of it
- Analyze the stochastic solutions for different increasing scenario trees and stop when there is no change in the optimal solution (or viceversa)
Optimization-simulation combination

- Use the model in an open-loop control mechanism with rolling horizon
  1. First planning by stochastic optimization
  2. Second simulation of the random parameters

- Stochastic optimization
  - Determines optimal policies taking into account the uncertainty

- Simulation
  - Evaluates possible future outcomes of random parameters given the optimal policies obtained previously
ROM General overview

(https://www.iit.comillas.edu/aramos/ROM.htm)
Cross-validation

- Stochastic data split in two sets:
  - **Training set** to determine the scenario tree (in-sample)
  - **Test/validation set** to evaluate/validate the optimal stochastic solutions (out-of-sample)

- Performance measures based on out-of-sample values
Probabilistic or chance constraints

- The **probability of satisfying a constraint** with stochastic parameters should be greater than a certain probability.
- Or, alternatively, the **probability of satisfying a set of constraints** with stochastic parameters should be greater than a certain probability.

\[
P \left( \sum_j a_j^\omega x_j \leq b^\omega \quad \forall \omega \right) \geq \alpha
\]

- In case of discrete distributions a binary variable is needed for each event (a realization of the discrete parameters).
Robust optimization

- **Risk-averse criterion** (minimax or Savage’s criterion)
- There may exist catastrophic scenarios or huge non-linearity in the objective function
- There is no probabilities associated to the scenarios or they are not used
- Objective function:
  - Minimize the maximum regret
  - Minimize the maximum value of the objective function
- Robust optimization
  - Robust solution if it is similar to the optimal one in all the scenarios
  - Robust model if it is almost feasible in all the scenarios
  - Robust optimization balances both objectives: similar to optimal solution and feasible
Applications in electric systems

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

- Tutorial talks prior to the [XIV Int. Conf. on Stochastic Programming](http://www.univie.ac.at/spxi/tutorial/TutPres_Vladimirou.ppt) (Búzios, Brazil, 2016):
  - Finance ([http://www.univie.ac.at/spxi/tutorial/TutPres_Vladimirou.ppt](http://www.univie.ac.at/spxi/tutorial/TutPres_Vladimirou.ppt))
  - Pension fund management ([http://www.univie.ac.at/spxi/tutorial/TutPres_Hochreiter.pdf](http://www.univie.ac.at/spxi/tutorial/TutPres_Hochreiter.pdf))
  - Electric systems ([http://www.univie.ac.at/spxi/tutorial/TutPres_Philpott.pps](http://www.univie.ac.at/spxi/tutorial/TutPres_Philpott.pps))
  - Gas markets ([http://www.univie.ac.at/spxi/tutorial/TutPres_Tomasgard.pdf](http://www.univie.ac.at/spxi/tutorial/TutPres_Tomasgard.pdf))
  - Logistics ([http://www.univie.ac.at/spxi/tutorial/TutPres_Wallace.ppt](http://www.univie.ac.at/spxi/tutorial/TutPres_Wallace.ppt))
Stochastic Daily Unit Commitment Model Mathematical formulation

- **Objective function**
  - Minimize the total *expected* variable costs plus penalties for energy not served

- **Variables**
  - BINARY: Commitment, startup and shutdown of thermal units
  - Thermal output

- **Operation constraints**
  - Load balance and operating reserve
  - Thermal operation constraints

- **Mixed integer linear programming (MIP)**
Indices

- **Time scope**
  - 1 day
- **Period**
  - 1 hour
- **Scenario**

\[
\begin{align*}
\text{Hour} & \quad n \\
\text{Scenario} & \quad \omega
\end{align*}
\]
Demand (5 weekdays)

Chronological Load Curve (5 Working Days)

Demand \([MW]\) \(D_n\)
Intermittent generation (IG)

Intermittent generation \([MW]\) \(i_{g_n}^\omega\)
Technical characteristics of thermal units ($t$)

- Maximum and minimum output
- Fuel cost
- Slope and intercept of the heat rate straight line
- Operation and maintenance (O&M) variable cost
  - No load cost = fuel cost x heat rate intercept
  - Variable cost = fuel cost x heat rate slope + O&M cost
- Cold startup cost
- Up and down ramps

<table>
<thead>
<tr>
<th>Technical characteristic</th>
<th>Unit</th>
<th>Equation</th>
</tr>
</thead>
<tbody>
<tr>
<td>Max and min output</td>
<td>MW</td>
<td>$\bar{P}_t; p_t$</td>
</tr>
<tr>
<td>No load cost</td>
<td>€/h</td>
<td>$f_t$</td>
</tr>
<tr>
<td>Variable cost</td>
<td>€/MWh</td>
<td>$v_t$</td>
</tr>
<tr>
<td>Startup cost</td>
<td>€</td>
<td>$su_t$</td>
</tr>
<tr>
<td>Shutdown cost</td>
<td>€</td>
<td>$sd_t$</td>
</tr>
<tr>
<td>Ramp up</td>
<td>MW/h</td>
<td>$rup_t$</td>
</tr>
<tr>
<td>Ramp down</td>
<td>MW/h</td>
<td>$rdw_t$</td>
</tr>
</tbody>
</table>
Scenario tree

- Represents how the stochasticity is revealed over time, i.e., the different states of the random parameters and simultaneously the non anticipative decisions over time
- **Nodes**: where decisions are taken.
- **Scenarios**: realizations of the random process.
Weekly load, a 4-scenario tree example

Historical series (green)
Scenario tree (black)
Weekly load, a 32-scenario tree example

Scenario tree (colored)
Scenario tree for the SDUC

- Commitment decisions of thermal units (the set of committed units) are unique under different stochastic scenarios (intermittent generation IG, demand, etc.)
Scenario tree example with IG uncertainty

Hour 1
- IG Output 400 MW
- IG Output 430 MW
- IG Output 560 MW
- IG Output 600 MW

Hour 2
- IG Output 410 MW
  - Prob: 0.3
- IG Output 420 MW
  - Prob: 0.2
- IG Output 630 MW
  - Prob: 0.25
- IG Output 550 MW
  - Prob: 0.25
Variables

- **Commitment, startup and shutdown** of thermal units (BINARY)

  Commitment, startup and shutdown \( \{0,1\} \ UC_{nt}, SU_{nt}, SD_{nt} \)

- **Production** of thermal units

  Production of a thermal unit \([MW]\) \(P_{nt}\)

- **Intermittent generation**

  Intermittent generation \([MW]\) \(IG_{n}\)

- **Energy not served**

  Energy not served \([MW]\) \(ENS_{n}\)
Constraints: Operating power reserve

Committed output of thermal units
\[ \geq \text{Demand} \]
+ Operating reserve for each load level and scenario

\[ \sum_{t} p_t UC_{nt} \geq D_n + O_n \quad \forall n \]
Constraints: Generation and load balance

Generation of thermal units
+ Energy not served
= Demand for each load level and scenario

\[ \sum_{t} P_{nt}^{\omega} + IG_{n}^{\omega} + ENS_{n}^{\omega} = D_{n} \quad \forall \omega n \]
Constraints: Production in consecutive load levels

Unit output in any hour - Unit output in previous one ≤ ramp up
Unit output in any hour - Unit output in previous one ≥ – ramp down

\[ P^n_t - P^n_{n-1t} \leq rup_t \quad \forall \omega nt \]
\[ P^n_t - P^n_{n-1t} \geq -rdw_t \quad \forall \omega nt \]
Constraints: Commitment and startup

Commitment of a thermal unit in an hour
– Commitment of a thermal unit in the previous hour
= Startup of a thermal unit in this hour
– Shutdown of a thermal unit in this hour

\[ UC_{nt} - UC_{n-1t} = SU_{nt} - SD_{nt} \quad \forall nt \]
Constraints: Commitment and production

Production of a thermal unit on every scenario
≤ Commitment of a thermal unit x the maximum output

Production of a thermal unit on every scenario
≥ Commitment of a thermal unit x the minimum output

\[ UC_{nt} \bar{p}_t \leq P^\omega_{nt} \leq UC_{nt} \bar{p}_t \quad \forall \omega_{nt} \]

- If the thermal unit is committed \( UC_{nt} = 1 \) it can produce between its minimum and maximum output
- If the thermal unit is not committed \( UC_{nt} = 0 \) it can’t produce
Constraints: Operation limits

**Power output between limits for each unit**

\[ 0 \leq P_{nt}^\omega \leq \bar{p}_t \quad \forall \omega nt \]

**Commitment, startup and shutdown for each unit**

\[ UC_{nt}, SU_{nt}, SD_{nt} \in \{0, 1\} \quad \forall nt \]

**Intermittent generation limit**

\[ 0 \leq IG_n^\omega \leq ig_n^\omega \quad \forall \omega n \]
Multiobjective function

- Minimize
  - Thermal unit expected variable costs

\[
\sum_{nt} s_{U_t} S_{U_{nt}} + \sum_{nt} s_{D_t} S_{D_{nt}} + \sum_{nt} f_{UC_{nt}} + \sum_{\omega nt} p_{\omega v_t} P_{nt}
\]

- Expected penalty introduced in the objective function for energy not served

\[
\sum_{\omega n} p_{\omega v_t} ENS_{\omega n}
\]
Short Run Marginal Cost (SRMC)

- **Short Run Marginal Cost** = Dual variable of generation and load balance when binary variables are fixed [€/MWh]
  - Change in the objective function due to a marginal increment in the demand

\[
\sum_{t} P^\omega_{nt} + IG^\omega_n + ENS^\omega_n = D_n : \sigma^\omega_n \quad \forall \omega n
\]

\[
SRMC^\omega_n = \sigma^\omega_n \quad \forall \omega n
\]
**StarGenLite_SDUC** Stochastic Daily Unit Commitment Model ([https://www.iit.comillas.edu/aramos/StarGenLite_SDUC.zip](https://www.iit.comillas.edu/aramos/StarGenLite_SDUC.zip))

- **Files**
  - Microsoft Excel interface for input and output data: *StarGenLite_SDUC.xlsm*
  - GAMS file: *StarGenLite_SDUC.gms*

- **How to use it**
  - **Save the Excel workbook if data have changed**
  - Run the model
  - The model creates
    - *tmp.xlsx* with the output data and
    - *StarGenLite_SDUC.lst* as the listing file of the GAMS execution
  - Load the results into the Excel interface

[Run] [Load results]
Medium Term Stochastic Hydrothermal Coordination Model Prototype. Mathematical formulation

• **Objective function**
  – Minimize the total expected variable costs plus penalties for energy and power not served

• **Variables**
  – **BINARY:** Commitment, startup and shutdown of thermal units
  – Thermal, storage hydro and pumped storage hydro output
  – Reservoir levels

• **Operation constraints**
  – **Inter-period**
    • Storage hydro and pumped storage hydro scheduling
      *Water balance with stochastic inflows*
  – **Intra-period**
    • Load balance and operating reserve
    • Detailed hydro basin modeling
    • Thermal, storage hydro and pumped-storage hydro operation constraints

• **Mixed integer linear programming (MIP)**
Indices

- **Time scope**
  - 1 year

- **Period**
  - 1 month

- **Subperiod**
  - weekdays and weekends

- **Load level**
  - peak, shoulder and off-peak

<table>
<thead>
<tr>
<th>Period</th>
<th>Subperiod</th>
<th>Load level</th>
</tr>
</thead>
<tbody>
<tr>
<td>$p$</td>
<td>$s$</td>
<td>$n$</td>
</tr>
</tbody>
</table>
**Demand (5 weekdays)**

- **Chronological Load Curve**
  - Hours: 1, 5, 9, 13, 17, 21, 25, 29, 33, 37, 41, 45, 49, 53, 57, 61, 65, 69, 73, 77, 81, 85, 89, 93, 97, 101, 105, 109, 113, 117
  - Demand [MW]: 0, 5000, 10000, 15000, 20000, 25000, 30000, 35000, 40000, 45000, 50000

- **Load Duration Curve**
  - Hours: 1, 5, 9, 13, 17, 21, 25, 29, 33, 37, 41, 45, 49, 53, 57, 61, 65, 69, 73, 77, 81, 85, 89, 93, 97, 101, 105, 109, 113, 117
  - Demand [MW]: 0, 5000, 10000, 15000, 20000, 25000, 30000, 35000, 40000, 45000, 50000

- **Load Duration Curve in 3 Load Levels**
  - Hours: 1, 5, 9, 13, 17, 21, 25, 29, 33, 37, 41, 45, 49, 53, 57, 61, 65, 69, 73, 77, 81, 85, 89, 93, 97, 101, 105, 109, 113, 117
  - Demand [MW]: 0, 5000, 10000, 15000, 20000, 25000, 30000, 35000, 40000, 45000, 50000
Demand

- Monthly demand with several load levels
  - Peak, shoulder and off-peak for weekdays and weekends
- All the weekdays of the same month are similar (same for weekends)
Technical characteristics of thermal units ($t$)

- Maximum and minimum output
- Fuel cost
- Slope and intercept of the heat rate straight line
- Operation and maintenance (O&M) variable cost
  - No load cost = fuel cost $\times$ heat rate intercept
  - Variable cost = fuel cost $\times$ heat rate slope + O&M cost
- Cold startup cost
- Equivalent forced outage rate (EFOR)

<table>
<thead>
<tr>
<th>Characteristic</th>
<th>Unit</th>
<th>Symbol</th>
</tr>
</thead>
<tbody>
<tr>
<td>Max and min output</td>
<td>MW</td>
<td>$\bar{P}_t, \underline{P}_t$</td>
</tr>
<tr>
<td>No load cost</td>
<td>€/h</td>
<td>$f_t$</td>
</tr>
<tr>
<td>Variable cost</td>
<td>€/MWh</td>
<td>$v_t$</td>
</tr>
<tr>
<td>Startup cost</td>
<td>€</td>
<td>$su_t$</td>
</tr>
<tr>
<td>EFOR</td>
<td>p.u.</td>
<td>$q_t$</td>
</tr>
</tbody>
</table>
Technical characteristics of hydro plants ($h$)

- Maximum and minimum output
- Production function (efficiency for conversion of water release in m$^3$/s to electric power MW)
- Efficiency of pumped storage hydro plants
  - Only this ratio of the energy consumed to pump the water is recovered by turbining this water

Max and min output $[MW]$ $\bar{P}_h, P_h$
Production function $[kWh / m^3]$ $c_h$
Efficiency $[p.u.]$ $\eta_h$
Technical characteristics of hydro reservoirs ($r$)

- Maximum and minimum reserve
- Initial reserve
  - Final reserve = initial reserve
- Stochastic inflows

Max and min reserve \[ [hm^3] \begin{align*} \bar{r}_r, \underline{r}_r \end{align*} \]  
Initial and final reserve \[ [hm^3] \begin{align*} r'_r \end{align*} \]  
Stochastic inflows \[ [m^3/s] \begin{align*} i_{pr}^\omega \end{align*} \]
Scenario tree example

In each node a decision is made and afterwards stochastic parameters are revealed.
Scenario tree. Ancestor and descendant

Tree structure
- Scenario $\omega$
- Period $p$
- Scenario tree $(p, \omega)$

Tree relations
- $\omega' \in a(\omega)$
- $(p2, sc03) \in a[(p3, sc03)]$

Tree data
- Scenario probability [p.u.] $P_p^\omega$
- Stochastic inflows [m$^3$/s] $i_{pr}^\omega$

Scenario tree:
- Scenario 1
- Scenario 2
- Scenario 3
- Scenario 4

Note: The diagram shows the relationships between scenarios and periods, with probabilities and stochastic inflows indicated.
Only one spillage per reservoir can be considered

Hydro topology

Hydro plant upstream of reservoir \( h \in up(r) \) \( hur(h, r) \) \((h_1, r_3)\)
Pumped hydro plant upstream of reservoir \( h \in up(r) \) \( hpr(h, r) \) \((h_3, r_2)\)
Reservoir upstream of hydro plant \( h \in dw(r) \) \( ruh(r, h) \) \((r_2, h_2)\)
Reservoir upstream of pumped hydro plant \( h \in dw(r) \) \( rph(r, h) \) \((r_3, h_3)\)
Reservoir upstream of reservoir \( r' \in up(r) \) \( rur(r, r) \) \((r_1, r_3)\)
Other system parameters

- Energy not served cost
- Operating power reserve not served cost
- Operating power reserve

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Unit</th>
<th>Symbol</th>
</tr>
</thead>
<tbody>
<tr>
<td>Energy not served cost</td>
<td>€ / MWh</td>
<td>$v'$</td>
</tr>
<tr>
<td>Operating power reserve not served cost</td>
<td>€ / MW</td>
<td>$v''</td>
</tr>
<tr>
<td>Operating reserve</td>
<td>MW</td>
<td>$O_{ps1}$</td>
</tr>
</tbody>
</table>
Variables

- **Commitment, startup and shutdown** of thermal units (BINARY)
  
  Commitment, startup and shutdown \( \{0, 1\} \) \( \omega \)

- **Production** of thermal units and hydro plants
  
  Production of a thermal or hydro unit \([MW]\) \( P_{psnt}^\omega, P_{psnh}^\omega \)

- **Consumption** of pumped storage hydro plants
  
  Consumption of a hydro plant \([MW]\) \( C_{psnh}^\omega \)

- **Reservoir levels**
  
  Reservoir level \([hm^3]\) \( R_{pr}^\omega \)

- **Energy and power not served**
  
  Energy and power not served \([MW]\) \( ENS_{psn}^\omega, PNS_{ps}^\omega \)
Constraints: Operating power reserve

Committed output of thermal units
+ Maximum output of hydro plants
+ Power not served
≥ Demand
+ Operating reserve for peak load level, subperiod, period and scenario

\[ \sum_t \overline{p}_t^{UC_{\omega}}_{pst} + \sum_h \overline{p}_h + PNS_{\omega}^{ps} \geq D_{ps1} + O_{ps1} \quad \forall \omega, ps \]
Constraints: Generation and load balance

Generation of thermal units
+ Generation of storage hydro plants
– Consumption of pumped storage hydro plants
+ Energy not served
=
Demand for each load level, subperiod, period and scenario

\[
\sum_{t} P_{psnt}^\omega + \sum_{h} P_{psnh}^\omega - \sum_{h} C_{psnh}^\omega / \eta_h + ENS_{psn}^\omega = D_{psn} \quad \forall \omega psn
\]
Constraints: Production in consecutive load levels

Output of a unit in shoulder \( \leq \) Output of a unit in peak
Output of a unit in off-peak \( \leq \) Output of a unit in shoulder

\[
P^\omega_{psn+1t} \leq P^\omega_{psnt} \quad \forall \omega psnt
\]
\[
P^\omega_{psn+1h} \leq P^\omega_{psnh} \quad \forall \omega psnh
\]
Constraints: Commitment, startup and shutdown

- All the weekdays of the same month are similar (same for weekends)
- Commitment decision of a thermal unit
Constraints: Commitment, startup and shutdown

- **Startup** of thermal units can only be made in the transition between consecutive weekend and weekdays

  Commitment of a thermal unit in a weekday
  - Commitment of a thermal unit in the weekend of previous period
  = Startup of a thermal unit in this weekday
  - Startup of a thermal unit in this weekday

  \[
  UC_{p, st}^\omega - UC_{p-1, st+1}^\omega = SU_{p, st}^\omega - SD_{p, st}^\omega \quad \forall \omega \in a(\omega)
  \]

- **Shutdown** only in the opposite transition

  Commitment of a thermal unit in a weekend
  - Commitment of a thermal unit in the previous weekday
  = Startup of a thermal unit in this weekend
  - Shutdown of a thermal unit in this weekend

  \[
  UC_{p, st+1}^\omega - UC_{p, st}^\omega = SU_{p+1, st}^\omega - SD_{p+1, st}^\omega \quad \forall \omega \in a(\omega)
  \]
Constraints: Commitment and production

Production of a thermal unit
\[ \leq \text{Commitment of a thermal unit } \times \text{the maximum output reduced by availability rate} \]

Production of a thermal unit
\[ \geq \text{Commitment of a thermal unit } \times \text{the minimum output reduced by availability rate} \]

\[ UC^\omega_{pst} p_t (1 - q_t) \leq P^\omega_{psnt} \leq UC^\omega_{pst} p_t (1 - q_t) \quad \forall \omega psnt \]

• If the thermal unit is committed \((UC^\omega_{pst} = 1)\) it can produce between its minimum and maximum output
• If the thermal unit is not committed \((UC^\omega_{pst} = 0)\) it can’t produce
Constraints: Water balance for each reservoir

Reservoir volume at the beginning of the period
– Reservoir volume at the end of the period
+ Natural inflows
– Spills from this reservoir
+ Spills from upstream reservoirs
+ Turbined water from upstream storage hydro plants
– Turbined and pumped water from this reservoir
+ Pumped water from upstream pumped hydro plants = 0 for each reservoir, period and scenario

\[
\begin{align*}
R^\omega_{pr} - R^\omega_{pr-1} + i^\omega_{pr} - S^\omega_{pr} + \sum_{r' \in \text{up}(r)} S^\omega_{pr'} & + \sum_{sn \in \text{up}(r)} d_{psn} P^\omega_{psnh} / c_h - \sum_{sn \in \text{dw}(r)} d_{psn} P^\omega_{psnh} / c_h \\
+ \sum_{sn \in \text{up}(r)} d_{psn} C^\omega_{psnh} / c_h - \sum_{sn \in \text{dw}(r)} d_{psn} C^\omega_{psnh} / c_h & = 0 \quad \forall \omega \in \text{a}(\omega) \quad \omega' \in \text{a}(\omega)
\end{align*}
\]
Constraints: Operation limits

Reservoir volumes between limits for each hydro reservoir

\[ r_r \leq R^\omega_{pr} \leq \bar{r}_r \quad \forall \omega pr \]
\[ R_{0r} = R^\omega_{Pr} = r'_r \quad \forall \omega r \]

Power output between limits for each unit

\[ 0 \leq P^\omega_{psnt} \leq p_t(1 - q_t) \quad \forall \omega psnt \]
\[ 0 \leq P^\omega_{psnh}, C^\omega_{psnh} \leq \bar{p}_h \quad \forall \omega psnh \]

Commitment, startup and shutdown for each unit

\[ UC^\omega_{pst}, SU^\omega_{pst}, SD^\omega_{pst} \in \{0, 1\} \quad \forall \omega pst \]
Multiobjective function

- Minimize
  - Thermal unit expected variable costs
    \[
    \sum_{\omega_{pst}} p_{p}^\omega s_{t} U_{pst}^{\omega} + \sum_{\omega_{psn}} p_{p}^\omega d_{psn} f_{t} U_{pst}^{\omega} + \sum_{\omega_{psnt}} p_{p}^\omega d_{psn} v_{t} P_{psnt}^{\omega}
    \]
  - Expected penalties introduced in the objective function for energy and power not served
    \[
    \sum_{\omega_{psn}} p_{p}^\omega d_{psn} v' E_{psn}^{\omega} + \sum_{\omega_{ps}} p_{p}^\omega v'' P_{ps}^{\omega}
    \]
Short Run Marginal Cost (\(SRMC\))

- **Dual variable** of generation and load balance [€/MW]
  - Change in the objective function due to a marginal increment in the demand
    \[
    \sum_{t} P_{p_{tn}}^{\omega} + \sum_{h} P_{p_{nh}}^{\omega} - \sum_{h} C_{p_{nh}}^{\omega} / \eta_{h} + ENS_{p_{sn}}^{\omega} = D_{p_{sn}} : \sigma_{p_{sn}}^{\omega} \quad \forall \omega_{p_{sn}}
    \]

- **Short Run Marginal Cost** = dual variable / load level duration. Expressed in [€/MWh]
  \[
  SRMC_{p_{sn}}^{\omega} = \sigma_{p_{sn}}^{\omega} / d_{p_{sn}} \quad \forall \omega_{p_{sn}}
  \]
Water value

- **Dual variable** of water balance for each reservoir [€/hm$^3$]
  
  - Change in the objective function due to a marginal increment in the reservoir inflow

\[
\frac{R_{\omega'}^{pr}}{p-1r} - R_{pr}^{pr} + i_{pr}^{\omega} - S_{pr}^{\omega} + \sum_{r' \in up(r)} S_{pr'}^{\omega} + \sum_{sn \in up(r)} d_{psn} P_{psnh}^{\omega} / c_{h} - \sum_{sn \in dw(r)} d_{psn} P_{psnh}^{\omega} / c_{h} + \sum_{sn \in up(r)} d_{psn} C_{psnh}^{\omega} / c_{h} - \sum_{sn \in dw(r)} d_{psn} C_{psnh}^{\omega} / c_{h} = 0 \\
: \pi_{pr}^{\omega} \forall \omega pr \quad \omega' \in a(\omega)
\]

- Turbining water has no variable cost. However, an additional hm$^3$ turbined allows to substitute energy produced by thermal units with the corresponding variable cost (this is called water value)
StarGenLite_SHTCM Medium Term Stochastic Hydrothermal Coordination Model (https://www.iit.comillas.edu/aramos/StarGenLite_SHTCM.zip)

- **Files**
  - Microsoft Excel interface for input and output data
    StarGenLite_SHTCM.xlsm
  - GAMS file StarGenLite_SHTCM.gms

- **How to use it**
  - **Save the Excel workbook if data have changed**
  - Run the model
  - The model creates
    - tmp.xlsx with the output data and
    - StarGenLite_SHTCM.lst as the listing file of the GAMS execution
  - Load the results into the Excel interface
Two-stage and multistage programming

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

\[
\begin{align*}
    c_1 & \in \mathbb{R}^{n_1} & c_2 & \in \mathbb{R}^{n_2} \\
    A_1 & \in \mathbb{R}^{m_1 \times n_1} & A_2 & \in \mathbb{R}^{m_2 \times n_2} \\
    b_1 & \in \mathbb{R}^{m_1} & b_2 & \in \mathbb{R}^{m_2} \\
    x_1 & \in \mathbb{R}^{n_1} & x_2 & \in \mathbb{R}^{n_2}
\end{align*}
\]

\[
\begin{align*}
    \min(c_1^T x_1 + c_2^T x_2) \\
    \text{subject to} \\
    A_1 x_1 = b_1 \\
    B_1 x_1 + A_2 x_2 = b_2 \\
    x_1, x_2 \geq 0
\end{align*}
\]
Structures of the constraint matrix

• Block diagonal with variables that complicate, constraints or both
Two-stage stochastic linear programming PLE-2

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

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

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

$$\begin{bmatrix}
A_1 \\
B_1^1 & A_2^1 \\
B_1^2 & A_2^2 \\
B_1^3 & A_2^3
\end{bmatrix}$$
Deterministic equivalent problem (DEP)

- State space is finite (and small)

- Formulation of the deterministic equivalent problem

\[
\begin{align*}
\min_{x_1, x_2^1, x_2^2, x_2^3} & \quad c_1^T x_1 + p_1 c_2^1 x_2^1 + p_1 c_2^2 x_2^2 + p_1 c_2^3 x_2^3 \\
A_1 x_1 & = b_1 \\
B_2^1 x_1 + A_2^1 x_2^1 & = b_2^1 \\
B_2^2 x_1 + A_2^2 x_2^2 & = b_2^2 \\
B_2^3 x_1 + A_2^3 x_2^3 & = b_2^3 \\
x_1, x_2^1, x_2^2, x_2^3 & \geq 0
\end{align*}
\]
Two-stage stochastic linear programming PLE-2 (ii)

- Minimization of the **maximum regret**

\[
\min_{\alpha, x_1^*, x_2^*} \quad \alpha \quad - c_1^T x_1 - c_2^T x_2^* \geq - f^\omega \quad \forall \omega \in \Omega
\]

\[
A_1 x_1 = b_1
\]

\[
B_1^\omega x_1 + A_2^\omega x_2^\omega = b_2^\omega
\]

\[
x_1, \quad x_2^\omega \geq 0
\]

\( f^\omega \) o.f. with perfect information for scenario \( \omega \)

- Minimization of the **maximum cost**

\[
\min_{\alpha, x_1^*, x_2^*} \quad \alpha \quad - c_1^T x_1 - c_2^T x_2^* \geq 0 \quad \forall \omega \in \Omega
\]

\[
A_1 x_1 = b_1
\]

\[
B_1^\omega x_1 + A_2^\omega x_2^\omega = b_2^\omega
\]

\[
x_1, \quad x_2^\omega \geq 0
\]
**Multistage linear programming PL-P**

\[
\begin{align*}
    \min_{x_p} & \sum_{p=1}^{P} c_p^T x_p \\
    B_{p-1} x_{p-1} + A_p x_p &= b_p & p = 1, \ldots, P \\
    x_p &\geq 0 \\
    B_0 x_0 &= 0
\end{align*}
\]
Medium term hydrothermal scheduling problem: constraint matrix

**Intra-period Constraints**

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

**Inter-period Constraints**

\[R_{p-1}\] 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

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

\[
B_{p-1}^{\omega_p} x_{p-1}^{\omega_p} + A_{p}^{\omega_p} x_{p}^{\omega_p} = b_{p}^{\omega_p} \quad p = 1, \ldots, P
\]

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

\[
B_{0}^{\omega_p} \equiv 0
\]

- Size *grows exponentially* with the number of scenarios
- Probabilities \( p_{p}^{\omega_p} \) are conditional
- Constraint matrix
4. Decomposition techniques

- General overview
- Applications in electric systems
- Two-stage and multistage programming
- Benders decomposition
- Nested Benders decomposition
- Dantzig-Wolfe decomposition
- Lagrangian relaxation
- Scenario tree
- Decomposition in two-stage and multistage stochastic programming
- Improvements in decomposition techniques
- Simulation in stochastic optimization
- Stochastic dual dynamic programming
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 obtaining 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** from master to subproblem
  – **Primal** (Benders)
  – **Dual** (Dantzig-Wolfe or Lagrangian relaxation)
Coordinating mechanisms. Hydrothermal model

- **Primal (quantities)**
  - Master assigns an *amount of outflow* 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 outflow *complicate* the solution \( \Rightarrow \) Benders
  - *Constraints* of hydro outflow *complicate* the solution \( \Rightarrow \) 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 outflow 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
Reservoir level or Hydro outflow
Water value

Subproblem 2
Reservoir level or Hydro outflow
Water value

Subproblem 3
Reservoir level or Hydro outflow
Water value

Subproblem 4
Computer applications

• Currently there are no standard solvers, powerful and stable for stochastic problems. In many cases
  Stochastic optimization = “do-it-yourself”

• Solver with decomposition methods (https://www.stoprog.org/resources)
  – MSLiP (Horand I. Gassman)
  – SLP-IOR (Janos Mayer)
  – FortSP (OptiRisk Systems)
  – SPiNE (Gautam Mitra)
  – DDSIP (Claus C. Carøe)
  – Python-based Stochastic Programming PySP (Jean-Paul Watson)
  – Bouncing Nested Benders Solvers BNBS (Fredrik Altenstedt)
  – Stochastic Modeling Interface COIN-SMI (open-source interface for modeling stochastic linear programming problems)

• ILOG Concert + C++
• GAMS o AMPL + decomposition methods
• GAMS/DECIS for two-stage stochastic programming
5. Benders decomposition

Benders decomposition
**Benders decomposition**

- Benders decomposition or
  - Primal (because primal values are fixed),
  - L-shaped (because the structure of the constraint matrix),
  - Resource (master problem assigns resources),
  - Outer approximation (recourse function is outer approximated)
- Splits the two-stage linear programming problem PL-2 in master and subproblem


Decomposing in Bender’s Tavern in Denver, CO, USA (INFORMS Nov 2004)
Bd Algorithm deduction (i)

- Two-stage linear programming PL-2

\[
\begin{align*}
\min (c_1^T x_1 + c_2^T x_2) \\
A_1 x_1 &= b_1 \\
B_1 x_1 + A_2 x_2 &= b_2 \\
x_1, x_2 &\geq 0
\end{align*}
\]

- It can also be expressed as

\[
\begin{align*}
\min_{x_1, \theta_2(x_1)} (c_1^T x_1 + \theta_2(x_1)) \\
A_1 x_1 &= b_1 \\
x_1 &\geq 0
\end{align*}
\]

- Being \( \theta_2(x_1) \in \mathbb{R} \) the recourse function (piecewise linear)

\[
\theta_2(x_1) = \min_{x_2} c_2^T x_2
\]

- And \( \pi_2 \) the dual variables of the second-stage constraints
Bd Algorithm deduction (ii)

- We express the subproblem is its dual form
  \[ \theta_2(x_1) = \max_{\pi_2} (b_2 - B_1 x_1)^T \pi_2 \]
  \[ A_2^T \pi_2 \leq c_2 \]

- Feasible region is independent of \( x_1 \)
- Vertices of polyhedron \( \Pi = \{ \pi_2^1, \pi_2^2, \ldots, \pi_2^\nu \} \)
- Maximum will be at one vertex
  \[ \theta_2(x_1) = \max \left\{ (b_2 - B_1 x_1)^T \pi_2^l \right\} \quad l = 1, \ldots, \nu \]

- Expressed as a linear problem (complete master problem)
  \[ \theta_2(x_1) = \min_{\theta_2} \theta_2 \]
  \[ \theta_2 \geq (b_2 - B_1 x_1)^T \pi_2^1 \]
  \[ \vdots \]
  \[ \theta_2 \geq (b_2 - B_1 x_1)^T \pi_2^\nu \]
Bd Algorithm deduction (iii)

- One cut is introduced in each iteration
- **Relaxed Master** problem

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

- **Dual variable** generated by the subproblem is different in each iteration. As the number of vertices is finite the number of algorithm iterations is also finite
- **Valid cut**: outer cut of the recourse function (not necessarily tangent). It is **NOT** necessary to solve the subproblem up to optimality
Bd Algorithm deduction (iv)

- Alternative formulation of Benders cuts
  \[
  \theta_2 \geq \pi_2^T (b_2 - B_1 x_1) = \pi_2^T (b_2 - B_1 x_1 + B_1 x_1^j - B_1 x_1^j) = \\
  = \pi_2^T [b_2 - B_1 x_1^j - B_1 (x_1 - x_1^j)] = \pi_2^T [b_2 - B_1 x_1^j] + \pi_2^T [-B_1 (x_1 - x_1^j)]
  \]

- Being
  \[
  f_2^j = \pi_2^T [b_2 - B_1 x_1^j]
  \]

- Benders cuts can also be expressed as (linearization around a point)
  \[
  \theta_2 - f_2^l \geq \pi_2^T B_1 (x_1^l - x_1^l)
  \]
  \[
  \theta_2 + \pi_2^T B_1 x_1 \geq f_2^l + \pi_2^T B_1 x_1^l
  \]

- \(\pi_2^T B_1\) is a subgradient of \(\theta_2(x_1)\)
Bd Relaxed Master problem and Subproblem

- Bd Relaxed Master: first stage + cuts
  \[
  \begin{align*}
  \min_{x_1, \theta_2} & \quad c^T_1 x_1 + \theta_2 \\
  A_1 x_1 & = b_1 \\
  \pi^T_2 B_1 x_1 + \theta_2 & \geq f^l_2 + \pi^T_2 B_1 x_1^l & l = 1, \ldots, j \\
  x_1 & \geq 0 
  \end{align*}
  \]

- Bd Subproblem: second stage with known decisions of the first stage
  \[
  f^j_2 = \min_{x_2} c^T_2 x_2 \\
  A_2 x_2 = b_2 - B_1 x_1^j \\
  : \quad \pi^j_2 \\
  x_2 \geq 0
  \]
Bd Feasibility cuts (i)

- So far we have assumed a feasible subproblem for master decisions (complete recourse or relatively complete)
- If subproblem is
  - Feasible we build optimality cuts
  - Infeasible we build feasibility cuts
  - Unbounded, then PL-2 is unbounded
- Phase I of simplex method (minimize the sum of infeasibilities)

\[
\begin{align*}
\min & \quad e^T v^+ + e^T v^- \\
\text{s.t.} & \quad A_2^T v^+ - I v^- = b_2 - B_1^T x_1 : \pi_2 \\
& \quad x_2, v^+, v^- \geq 0
\end{align*}
\]
Farkas’ lemma

• Let’s define primal and dual problems

\[
\begin{align*}
\min & \quad c^T x \\
\text{subject to} & \quad Ax = b \\
& \quad x \geq 0
\end{align*}
\]

\[
\begin{align*}
\max & \quad b^T y \\
\text{subject to} & \quad A^T y \leq c
\end{align*}
\]

• Exactly one of these system of equations has solution
  – If the first is true necessarily the second is false

Primal feasibility condition

\[
\begin{align*}
A^T x &= b \\
x &\geq 0
\end{align*}
\]

\[
\begin{align*}
A^T y &\leq 0 \\
 b^T y &> 0
\end{align*}
\]
Bd Feasibility cuts (ii)

- Subproblem

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

will be feasible if for the dual variables that satisfy
\[ A_2^T \pi_2 \leq 0 \]
also met
\[ (b_2 - B_1 x_1^j)^T \pi_2 \leq 0 \]

- According to Farka’s lemma it can then be formulated as

\[ \max_{\pi_2} (b_2 - B_1 x_1)^T \pi_2 \]
\[ A_2^T \pi_2 \leq 0 \]

- If this o.f. is strictly positive a feasibility cut will be generated to avoid this (infeasible subproblem)

\[ (b_2 - B_1 x_1)^T \pi_2^j \leq 0 \]
Bd Feasibility cuts (iii)

- As the subproblem is a cone, it could be unbounded, some bounds on the dual variables are introduced:

\[
\max_{\pi_2} (b_2 - B_1 x_1)^T \pi_2 \\
A_2^T \pi_2 \leq 0 \\
-1 \leq \pi_2 \leq 1
\]

- The dual problem of this problem is the infeasibility minimization subproblem:

\[
\min_{x_2, v^+, v^-} e^T v^+ + e^T v^- \\
A_2 x_2 + Iv^+ - Iv^- = b_2 - B_1 x_1 : \pi_2 \\
x_2, v^+, v^- \geq 0
\]
Bd Feasibility cuts (iv)

- Relaxed Master problem

\[
\begin{align*}
\min & \quad c^T_1 x_1 + \theta_2 \\
\text{s.t.} & \quad A_i x_1 = b_i \\
& \quad \pi_2^T B_1 x_1 + \delta^l \theta_2 \geq \pi_2^T b_2 \quad l = 1, \ldots, j \\
& \quad x_1 \geq 0
\end{align*}
\]

\[
\delta^l = 1 \quad \text{for optimality cuts and } \delta^l = 0 \quad \text{for feasibility cuts}
\]

- Feasibility cuts are optimality cuts with infinite slope

- These cuts avoid master solutions that make infeasible the subproblem and allow to keep those that are feasible

- Alternatively, instead of feasibility cuts we can penalize the infeasibilities in the subproblem
Bd Relaxed Master and Subproblem

• Bd Relaxed 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 or lazy constraints option in Gurobi and CPLEX
  – Size \((m_1 + j) \times (n_1 + 1)\)
  – It can be nonconvex (MIP, NLP)

• Bd Subproblem
  – Each iteration modifies the RHS of the constraints
  – It is worthy to use primal simplex method (if size is reasonable)
  – Size \(m_2 \times n_2\)
  – It must be convex (LP, NLP). Generalized Benders decomposition (GBD)
**Bd Convergence**

**PROblema Maestro**

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

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

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

or repetition of the last master proposal
**Bd Algorithm (i)**

- 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 monotonically increasing.
  - Upper bound is not necessarily monotonically 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
Bd Algorithm (ii)

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

2. Solving the Bd Relaxed Master problem

\[
\begin{align*}
& \min_{x, \theta} c^T x + \theta \\
& A x + \theta \theta \geq b \quad A x = B x \quad x \geq 0
\end{align*}
\]

Determine the solution \((x_j, \theta_j)\) and the lower bound

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

3. Solving the Bd Subproblem of sum of infeasibilities

\[
f_j^l = \min_{x, v^+, v^-} e^T v^+ + e^T v^- \\
A x + Iv^+ - Iv^- = b - B x \quad x \geq 0
\]

If \(f_j^l \geq 0\) feasibility cut

If \(f_j^l = 0\) go to step 4.

4. Solving the Bd Subproblem

Obtain \(x_j^l\) and update the upper bound.

\[
|z - z| = \frac{c^T x_j^l - \theta_j}{c^T x_j^l + c^T x_j^l} \leq \varepsilon
\]

3. If stopping rule is met

If not go to step 2.
Case study 1: complete problem

\[
\min_{x, y} -2x - y
\]

\[
x \leq 4
\]

\[
x + y \leq 5
\]

\[
2x + 3y \leq 12
\]

\[
x, y \geq 0
\]

Optimal solution \((4,1)\)

Value of the o.f. -9
Case study 1

Objective function in 3D

Contour lines

Lower values of the o.f.

Upper values of the o.f.
Case study 1: complete master problem

\[
\min_{\theta} \ 2x + \theta(x)
\]
\[
x \leq 4
\]
\[
x \geq 0
\]
Case study 1: recourse function (to be discovered iteratively)

\[ \theta(x) = \min_y y \]
\[ y \leq 5 - x \]
\[ 3y \leq 12 - 2x \]
\[ y \geq 0 \]

Recourse function \( \theta(x) \)
¡UNKNOWN!
Case study 1: recourse function iteration 1

\[ \theta(x) = \min_y \ y \]
\[ y \leq 5 - 0 \]
\[ 3y \leq 12 - 2 \cdot 0 \]
\[ y \geq 0 \]

Initial master proposal \( x = 0 \)
Initial value of \( \theta(x = 0) = -4 \)
Derivative w.r.t. \( x = 2/3 \)

Upper bound -4
Case study 1: master problem iteration 1

\[ \min_{x, \theta} -2x + \theta \]
\[ x \leq 4 \]
\[ \theta - \frac{2}{3} x \geq -4 \]
\[ x \geq 0 \]

First cut
\[ \theta - 2x/3 = -4 \]

Second master proposal \( x = 4 \)
Value of \( \theta = -4/3 \)
Lower bound = \(-28/3\)
Case study 1: recourse function iteration 2

\[ \theta(x) = \min_{y} y \]
\[ y \leq 5 - 4 \]
\[ 3y \leq 12 - 2 \cdot 4 \]
\[ y \geq 0 \]

Solution \( x = 4 \)
Value of \( \theta(x = 4) = -1 \)
Derivative w.r.t. \( x = 1 \)

Upper bound -9
Case study 1: **master problem iteration 2**

First cut
\[ \theta - \frac{2}{3} x = -4 \]

Second cut
\[ \theta - x = -5 \]

Third master proposal \( x = 4 \)
Value of \( \theta = -1 \)

Lower bound = -9
Case study 2: complete problem

\[
\min_{x,y} 2(x - 3)^2 + (y - 3)^2 \\
x \\ x + y \\ 2x + 3y \\ x, y \geq 0
\]
Case study 2

Objective function in 3D

Contour lines

Lower values of the o.f.

Upper values of the o.f.
Case study 2: complete master problem

\[
\min_{x} 2(x - 3)^2 + \theta(x) \\
x \leq 4 \\
x \geq 0
\]

All the Benders cuts ¡UNKNOWN!
Case study 2: recourse function

\[ \theta(x) = \min_y (y - 3)^2 \]

\[ y \leq 5 - x \]

\[ 3y \leq 12 - 2x \]

\[ y \geq 0 \]
Case study 2: recourse function iteration 1

\[ \theta(x) = \min_y (y - 3)^2 \]

\[ y \leq 5 - 2 \]
\[ 3y \leq 12 - 2 \cdot 2 \]
\[ y \geq 0 \]

Initial master proposal \( x = 2 \)
Initial value of \( \theta(x = 2) = 0.111 \)
Derivative w.r.t. \( x = 0.444 \)

Upper bound 2.111
Case study 2: **master problem iteration 1**

\[ \min_{x, \theta} 2(x - 3)^2 + \theta \]

\[ x \leq 4 \]

\[ \theta - 0.4\bar{x} \geq -0.7 \]

\[ x \geq 0 \]

First cut

\[ 0 - 0.444x = -0.777 \]

Second master proposal \( x = 3 \)

Value of \( \theta = 0.555 \)

Lower bound = 0.555
Case study 2: recourse function iteration 2

\[ \theta(x) = \min_y (y - 3)^2 \]
\[ y \leq 5 - 3 \]
\[ 3y \leq 12 - 2 \cdot 3 \]
\[ y \geq 0 \]

Degenerate solution: \( x = 3 \)
Value of \( \theta(x = 3) = 1 \)
Derivative + w.r.t. \( x = 2 \)
Derivative - w.r.t. \( x = 1.333 \)

Upper bound 1
Case study 2: master problem iteration 2

\[ \min_{x, \theta} 2(x - 3)^2 + \theta \]
\[ x \leq 4 \]
\[ \theta - 0.44x \geq -0.7 \]
\[ \theta - 1.3x \geq -3 \]
\[ x \geq 0 \]

First cut
\[ \theta - 0.444x = -0.777 \]

Second cut
\[ \theta - 1.333x = -3 \]

Third master proposal
\[ x = 2.5 \]
Value of \( \theta = 0.333 \)

Lower bound = 0.8333
Fixed Cost Transportation Problem (FCTP)

- It is a transportation problem where the arc connecting two nodes (i and j) has a **fixed cost** \( f_{ij} \) associated to its installation and a **variable cost** \( c_{ij} \) by the use. We want to minimize the total fixed (investment) and variable (transportation) costs subject to the constraints of demand supply \( b_j \) at the destinations and maximum capacity at the origins \( a_i \).
Fixed-Charge Transportation Problem (FCTP)

- **Bd Relaxed Master**
  \[
  \min \theta + \sum_{ij} (f_{ij}y_{ij})
  \]
  \[
  \delta^l \theta - \theta^l \geq \sum_{ij} \pi_{ij}^l M_{ij} (y_{ij}^l - y_{ij}) \quad l = 1, \ldots, k
  \]
  \[y_{ij} \in \{0,1\}\]

- **Bd Subproblem**
  \[
  \min \sum_{ij} (c_{ij}x_{ij})
  \]
  \[
  \sum_{j} x_{ij} \leq a_i \quad \forall i
  \]
  \[
  \sum_{i} x_{ij} \geq b_j \quad \forall j
  \]
  \[
  x_{ij} \leq M_{ij} y_{ij} \quad \forall ij
  \]
  \[
  x_{ij} \geq 0, y_{ij} \in \{0,1\}\]

- **Complete problem**
  \[
  \min \sum_{ij} (f_{ij}y_{ij} + c_{ij}x_{ij})
  \]
  \[
  \sum_{j} x_{ij} \leq a_i \quad \forall i
  \]
  \[
  \sum_{i} x_{ij} \geq b_j \quad \forall j
  \]
  \[
  x_{ij} \leq M_{ij} y_{ij} \quad \forall ij
  \]
  \[
  x_{ij} \geq 0, y_{ij} \in \{0,1\}\]

**Flows** (second stage)
- Capacity of each origin
- Demand of each destination
- Flow can pass only for installed connections

**Investment decisions** (first stage)
- Complete problem
- BD Subproblem
- BD Relax Master
- O.F. of the subproblem at iteration $l$
- Dual variables of linking constraints at iteration $l$
- Master proposal at iteration $l$
- Change in the o.f. proportional to changes in master proposals
Fixed-charge transportation problem. Bd Solution

- Possible arcs

- Solutions along Bd decomposition iterations
Fixed-charge transportation problem. Bd Convergence

<table>
<thead>
<tr>
<th>Iteration</th>
<th>Lower Bound</th>
<th>Upper Bound</th>
</tr>
</thead>
<tbody>
<tr>
<td>1 to 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>140</td>
<td>390</td>
</tr>
<tr>
<td>10</td>
<td>360</td>
<td>390</td>
</tr>
<tr>
<td>11</td>
<td>370</td>
<td>390</td>
</tr>
<tr>
<td>12</td>
<td>380</td>
<td>380</td>
</tr>
</tbody>
</table>
Fixed-charge transportation problem (FCTP) solved by Benders decomposition

* relative optimality tolerance in solving MIP problems

**option** OptcR = 0

**sets**
- **L** iterations / l1 * l20 /
- **LL(l)** iterations subset
- **I** origins / i1 * i4 /
- **J** destinations / j1 * j3 /

* Begin problem data

**parameters**
- **A(i)** product offer
  / i1 20, i2 30, i3 40, i4 20 /
- **B(j)** product demand
  / j1 20, j2 50, j3 30 /

**table** **C(i,j)** per unit variable transportation cost

<table>
<thead>
<tr>
<th></th>
<th>j1</th>
<th>j2</th>
<th>j3</th>
</tr>
</thead>
<tbody>
<tr>
<td>i1</td>
<td>1</td>
<td>2</td>
<td>3</td>
</tr>
<tr>
<td>i2</td>
<td>2</td>
<td>3</td>
<td>4</td>
</tr>
<tr>
<td>i3</td>
<td>3</td>
<td>4</td>
<td>5</td>
</tr>
<tr>
<td>i4</td>
<td>4</td>
<td>5</td>
<td>6</td>
</tr>
</tbody>
</table>

**table** **F(i,j)** fixed transportation cost

<table>
<thead>
<tr>
<th></th>
<th>j1</th>
<th>j2</th>
<th>j3</th>
</tr>
</thead>
<tbody>
<tr>
<td>i1</td>
<td>10</td>
<td>20</td>
<td>30</td>
</tr>
<tr>
<td>i2</td>
<td>20</td>
<td>30</td>
<td>40</td>
</tr>
<tr>
<td>i3</td>
<td>30</td>
<td>40</td>
<td>50</td>
</tr>
<tr>
<td>i4</td>
<td>40</td>
<td>50</td>
<td>60</td>
</tr>
</tbody>
</table>

* End problem data

**abort** $(\text{sum}[i, A(i)] < \text{sum}[j, B(j)])$ 'Infeasible problem'

**parameters**
- **BdTol** relative Benders tolerance / 1e-6 /
- **Z.Lower** lower bound / -inf /
- **Z.Upper** upper bound / inf /
- **Y.L (l,i,j)** first stage variables values in iteration l
- **PI.L (l,i,j)** dual variables of second stage constraints in iteration l
- **Delta(l)** cut type (feasibility 0 optimality 1) in iteration l
- **Z2.L (l)** subproblem objective function value in iteration l
FCTP solved by Benders decomposition (ii)

**positive variable**

\( X(i,j) \) arc flow

**binary variable**

\( Y(i,j) \) arc investment decision

**variables**

- \( Z_1 \): first stage objective function
- \( Z_2 \): second stage objective function
- \( \Theta \): recourse function

**equations**

- **EQ \_Z1**: first stage objective function
- **EQ \_Z2**: second stage objective function
- **EQ \_OBJ**: complete problem objective function

- **Offer** \((i)\)**: offer at origin
- **Demand** \((j)\)**: demand at destination
- **FlowLimit** \((i,j)\)**: arc flow limit

\[
\text{EQ}_Z1 \quad \text{..} 
Z_1 \ = \sum (i,j) \  F(i,j) \times Y(i,j) \ + \Theta \\
\text{EQ}_Z2 \quad \text{..} 
Z_2 \ = \sum (i,j) \  C(i,j) \times X(i,j) \\
\text{EQ}_OBJ \quad \text{..} 
Z_1 \ = \sum (i,j) \  F(i,j) \times Y(i,j) \ + \sum (i,j) \  C(i,j) \times X(i,j) \\
\text{Offer} \quad (i) \quad \text{..} 
\sum (j) \ X(i,j) \ = \ A(i) \\
\text{Demand} \quad (j) \quad \text{..} 
\sum (i) \ X(i,j) \ = \ B(j) \\
\text{FlowLimit} \quad (i,j) \quad \text{..} 
X(i,j) \ = \ min \{A(i),B(j)\} \times Y(i,j) \\
\text{Bd\_Cuts} \quad (l) \quad \text{Benders cuts} \ ;
\]

\[
\text{model} \quad \text{Master\_Bd} \quad / \quad \text{EQ}_Z1 \quad \text{Bd\_Cuts} \ ;
\text{model} \quad \text{Subproblem\_Bd} \quad / \quad \text{EQ}_Z2 \quad \text{Offer} \quad \text{Demand} \quad \text{FlowLimit} \ / \\
\text{model} \quad \text{Complete} \quad / \quad \text{EQ}_OBJ \quad \text{Offer} \quad \text{Demand} \quad \text{FlowLimit} \ / \\
X.up(i,j) \ = \ min \{A(i),B(j)\} \\
\]

*to allow CPLEX correctly detect rays in an infeasible problem*
*only simplex method can be used and no preprocessing neither scaling options*
*optimality and feasibility tolerances are very small to avoid primal degeneration*

**file** COPT / cplex.opt /
**put** COPT putclose 'ScaInd -1' / 'LPMethod 1' / 'PreInd 0' / 'EpOpt 1e-9' / 'EpRHS 1e-9' / ;
**Subproblem\_Bd.OptFile** = 1 ;
FCTP solved by Benders decomposition (iii)

* parameter initialization

\[ LL(1) = \text{no} ; \]
\[ \text{Theta.fx} = 0 ; \]
\[ \text{Delta}(1) = 0 ; \]
\[ Z2_L(1) = 0 ; \]
\[ PI_L(1,i,j) = 0 ; \]
\[ Y_L(1,i,j) = 0 ; \]

* Benders algorithm iterations

\[ \text{loop (1 $(abs(1-Z_Lower/Z_Upper) > BdTol)$,} \]

* solving master problem

\[ \text{solve Master_Bd using MIP minimizing } Z1 ; \]

* storing the master solution

\[ Y_L(1,i,j) = Y.L(i,j) ; \]

* fixing first-stage variables and solving subproblem

\[ Y.fx(i,j) = Y.L(i,j) ; \]

* solving subproblem

\[ \text{solve Subproblem_Bd using RMIP minimizing } Z2 ; \]

* storing parameters to build a new Benders cut

\[ \text{if (Subproblem_Bd.ModelStat = 4,} \]
\[ \text{Delta}(1) = 0 ; \]
\[ Z2_L(1) = \text{Subproblem_Bd.SumInfeas} ; \]
\[ \text{else} \]
\[ \text{updating lower and upper bound} \]
\[ Z_Lower = \text{Z1.L} ; \]
\[ Z_Upper = \text{min}(Z_Upper, Z1.L - \text{Theta.L} + Z2.L) ; \]
\[ \text{Theta.lo} = -\text{inf} ; \]
\[ \text{Theta.up} = \text{inf} ; \]
\[ \text{Delta}(1) = 1 ; \]
\[ \text{Delta}(1) = \text{Subproblem_Bd.ObjVal} ; \]
\[ PI_L(1,i,j) = \text{FlowLimit.m(i,j)} ; \]
\[ Y.lo(i,j) = 0 ; \]
\[ Y.up(i,j) = 1 ; \]

* increase the set of Benders cuts

\[ LL(1) = \text{yes} ; \]

\[ \text{solve Complete using MIP minimizing } Z1 \]
Deterministic & Stochastic FCTP

**Deterministic**

\[
\min_{x_{ij}, y_{ij}} \sum_{ij} (f_{ij} y_{ij} + c_{ij} x_{ij}) \\
\sum_{j} x_{ij} \leq a_i \quad \forall i \\
\sum_{i} x_{ij} \geq b_j \quad \forall j \\
x_{ij} \leq M_{ij} y_{ij} \quad \forall ij \\
x_{ij} \geq 0, y_{ij} \in \{0,1\}
\]

**Stochastic**

\[
\min_{x_{ij}, y_{ij}} \sum_{ij} \left( f_{ij} y_{ij} + \sum_{\omega} c_{ij} x_{ij}^{\omega} \right) \\
\sum_{j} x_{ij}^{\omega} \leq a_i \quad \forall i \omega \\
\sum_{i} x_{ij}^{\omega} \geq b_j^{\omega} \quad \forall j \omega \\
x_{ij}^{\omega} \leq M_{ij} y_{ij} \quad \forall ij \omega \\
x_{ij}^{\omega} \geq 0, y_{ij} \in \{0,1\}
\]
Deterministic & Stochastic FCTP

**Deterministic fixed-charge transportation problem (DFCTP)**

* relative optimality tolerance in solving MIP problems

```plaintext
option OptcR = 0
```

```plaintext
sets
I origins / i1 * i4 /
J destinations / j1 * j3 /
parameters
A(i) product offer / i1 20, i2 30, i3 40, i4 20 /
B(j) product demand / j1 20, j2 50, j3 30 /

table C(i,j) per unit variable transportation cost
i1 i2 i3 i4
j1 1 2 3 2
j2 3 2 1 4
j3 2 3 4 3

table F(i,j) fixed transportation cost
i1 i2 i3 i4
j1 10 20 30 40
j2 20 30 40 50
j3 30 40 50 60

abort $(sum[i, A(i)] < sum[j, B(j)]) 'Infeasible problem'

positive variable
X(i,j) arc flow

binary variable
Y(i,j) arc investment decision
variables
Z1 objective function

equations

EQ_OBJ complete problem objective function
Offer (i) offer at origin
Demand (j) demand at destination
FlowLimit(i,j) arc flow limit :

EQ_OBJ.. Z1 =e= sum[i, A(i)] + sum[j, C(i,j)*X(i,j)] + sum[j, F(i,j)*Y(i,j)] ;
Offer (i) .. sum[j, X(i,j)] =l= A(i) ;
Demand (j) .. sum[i, X(i,j)] =g= B(j) ;
FlowLimit(i,j) .. X(i,j) =l= min[A(i),B(j)] * Y(i,j) ;

model Complete / EQ_OBJ Offer Demand FlowLimit / ;
x.up(i,j) = min[A(i),B(j)]

solve Complete using MIP minimizing Z1
```

**Stochastic fixed-charge transportation problem (SFCTP)**

* relative optimality tolerance in solving MIP problems

```plaintext
option OptcR = 0
```

```plaintext
sets
I origins / i1 * i4 /
J destinations / j1 * j3 /
S scenarios / s1 * s3 /
parameters
A(i) product offer / i1 20, i2 30, i3 40, i4 20 /
P(s) scenario probability / s1 0.5, s2 0.3, s3 0.2 /

table B(s,j) product demand
s1 s2 s3
j1 21 31 51
j2 32 22 52
j3 53 33 23

abort $(sum[i, A(i)] < sum[j, B(s,j)]) 'Infeasible problem'

positive variable
X(s,i,j) arc flow

binary variable
Y(i,j) arc investment decision
variables
Z1 objective function

equations

EQ_OBJ complete problem objective function
Offer (s,i) offer at origin
Demand (s,j) demand at destination
FlowLimit(s,i,j) arc flow limit :

EQ_OBJ.. Z1 =e= sum[(s,i,j), P(s)*C(i,j)*X(s,i,j)] ;
Offer (s,i) .. sum[(j, X(s,i,j)] =l= A(i) ;
Demand (s,j) .. sum[(i, X(s,i,j)] =g= B(s,j) ;
FlowLimit(s,i,j) .. X(s,i,j) =l= min[A(i),B(s,j)] * Y(i,j) ;

model Complete / EQ_OBJ Offer Demand FlowLimit / ;
x.up(s,i,j) = min[A(i),B(s,j)]

solve Complete using MIP minimizing Z1
Stochastic FCTP with EMP

$title Deterministic fixed-charge transportation problem (FCTP)
/* relative optimality tolerance in solving MIP problems
option Optcr = 0;

sets
   I origins / i1 * i4 /;
   J destinations / j1 * j3 /;

parameters
   A(i) product offer /
      i1 20, i2 30, i3 40, i4 20 /;
   B(j) product demand /
      j1 11, j2 44, j3 66 /;

table C(i,j) per unit variable transportation cost
   j1  j2  j3
   i1  1  2  3
   i2  3  2  1
   i3  2  3  4
   i4  4  3  2;

table F(i,j) fixed transportation cost
   j1  j2  j3
   i1  10  20  30
   i2  20  30  40
   i3  30  40  50
   i4  40  50  60;

positive variable
   X(i,j) arc flow;

binary variable
   Y(i,j) arc investment decision;

variables
   Z1 objective function;

equations
   EQ_OBJ complete problem objective function;
   Offer (i) offer at origin;
   Demand (j) demand at destination;
   FlowLimit(i,j) arc flow limit;

EQ_OBJ ..
   Z1 =e= sum[(i,j), F(i,j)*Y(i,j)] + sum[(i,j), C(i,j)*X(i,j)];

Offer (i) ..
   sum[j, X(i,j)] =l= A(i); Demand (j) ..
   sum[i, X(i,j)] =g= B(j);

FlowLimit(i,j) .. X(i,j) =l= 100 * Y(i,j);

model Complete / all /;
X.up(i,j) = 100;

set S scenarios / s1 * s3 /;
parameter P(s) scenario probability /
   s1 0.5, s2 0.3, s3 0.2 /;

YS(s,i,j) arc investment decision Xs(s,i,j) arc flow;

file emp / '%emp.info%' /;
emp.pc=2;
put emp
   put '* problem %gams.i%' /;
   put
   put "stage 2 B X Offer Demand FlowLimit"
   putclose emp;

set dict / s . scenario . ''
   B . randvar . BS
   X . level    . XS
   Y . level    . YS /;

loop (s, abort $(sum[i, A(i)] < sum[j, BS(s,j)]) 'Infeasible problem');
solve Complete minimizing Z1 using emp scenario dict;
display XS, YS;
Stochastic FCTP solved by Benders decomposition (i)

Title Stochastic Fixed-charge transportation problem (FCTP) solved by Benders decomposition

* relative optimality tolerance in solving MIP problems

option Optcr = 0

sets
   L       iterations          / l1 * l70 /
   LL(l)   iterations subset   
   I       origins            / i1 * i4 /
   J       destinations        / j1 * j3 /
   S       scenarios          / s1 * s3 /

* Begin problem data

parameters
   A(i)    product offer
            / i1 20, i2 30, i3 40, i4 30 /
   P(s)    scenario probability / s1 0.5, s2 0.3, s3 0.2 /

table B(s,j) product demand
    j1  j2  j3
  s1  21  51  31
  s2  32  22  52
  s3  53  33  23

table C(i,j) per unit variable transportation cost
    j1  j2  j3
  i1  1   2   3
  i2  2   3   1
  i3  2   3   4
  i4  4   3   2

table F(i,j) fixed transportation cost
    j1  j2  j3
  i1  10  20  30
  i2  20  30  40
  i3  30  40  50
  i4  40  50  60

* End problem data

loop (s, abort $(sum[i, A(i)] < sum[j, B(s,j)]) 'Infeasible problem' )

parameters
   BdTo1  relative Benders tolerance / 1e-6 /
   Z_Lower lower bound                  -inf /
   Z_Upper upper bound                  inf /
   Y_L (l, i,j) first stage variables values in iteration l
   PI_L (l,s,i,j) dual variables of second stage constraints in iteration l
   Delta(l) cut type (feasibility 0 optimality 1) in iteration l
   Z2L(l) subproblem objective function value in iteration l
Stochastic FCTP solved by Benders decomposition (ii)

**positive variable**
X(s,i,j)    arc flow

**binary variable**
Y(i,j)  arc investment decision

**variables**
Z1  first stage objective function
Z2  second stage objective function
 Theta  recourse function

**equations**
EQ_Z1  first stage objective function
EQ_Z2  second stage objective function
EQ_OBJ  complete problem objective function
Offer (s,i ) offer at origin
Demand (s, j) demand at destination
FlowLimit(s,i,j) arc flow limit
Bd_Cuts (l) Benders cuts

EQ_Z1  
.. Z1 =e= sum[(i,j), F(i,j)*Y(i,j)] + Theta ;

EQ_Z2  
.. Z2 =e= sum[(i,j), P(s)*C(i,j)*X(s,i,j)] ;

EQ_OBJ  
.. Z1 =e= sum[(i,j), F(i,j)*Y(i,j)] + sum[(s,i,j), P(s)*C(i,j)*X(s,i,j)] ;

Offer (s,i ) .. sum[j, X(s,i,j)] =l= A( i) ;
Demand (s, j) .. sum[i, X(s,i,j)] =g= B(s,j) ;
FlowLimit(s,i,j) .. X(s,i,j) =l= min[A(i),B(s,j)] * Y(i,j) ;

Bd_Cuts(l)  .. Delta(l) * Theta =g= Z2_L(l) - 
sum[(s,i,j), PI_L(l,s,i,j) * min[A(i),B(s,j)] * (Y_L(l,i,j) - Y(i,j))] ;

model Master_Bd  / EQ_Z1  Bd_Cuts /
model Subproblem_Bd  / EQ_Z2  Offer, Demand, FlowLimit /
model Complete  / EQ_OBJ  Offer, Demand, FlowLimit /

X.up(s,i,j) = min[A(i),B(s,j)]

* to allow CPLEX correctly detect rays in an infeasible problem
* only simplex method can be used and no preprocessing neither scaling options
* optimality and feasibility tolerances are very small to avoid primal degeneration

file COPT / cplex.opt /
put  COPT putclose 'ScaInd -1' / 'LPMethod 1' / 'PreInd 0' / 'EpOpt 1e-9' / 'EpRHS 1e-9' / ;
Subproblem_Bd.OptFile = 1 ;
Stochastic FCTP solved by Benders decomposition (iii)

* parameter initialization

   \( LL (l) = \text{no} \);  
   \( \text{Theta.fx} = 0 \);  
   \( \text{Delta} (l) = 0 \);  
   \( \text{Z2.L} (l) = 0 \);  
   \( \text{PI_L(l,s,i,j)} = 0 \);  
   \( \text{Y_L (l, i,j)} = 0 \);

* Benders algorithm iterations

   \( \text{Loop (l } $(\text{abs}(1-Z_Lower/Z_Upper) > \text{BdTol})$,} \)

   * solving master problem

     \( \text{solve Master_Bd using MIP minimizing Z1;} \)

   * storing the master solution

     \( Y_L(l,i,j) = Y.l(i,j); \)

   * fixing first-stage variables and solving subproblem

     \( Y.fx (i,j) = Y.l(i,j); \)

   * solving subproblem

     \( \text{solve Subproblem_Bd using RMIP minimizing Z2;} \)

   * storing parameters to build a new Benders cut

     \( \text{if (Subproblem_Bd.ModelStat} = 4, } \)
     \( \text{Delta(l)} = 0; \)
     \( \text{Z2.L (l)} = \text{Subproblem_Bd.SumInfeas}; \)
     \( \text{else } \)
     \( \text{ updating lower and upper bound } \)
     \( Z_{Lower} = Z1.l; \)
     \( Z_{Upper} = \min(Z_{Upper}, Z1.l - \text{Theta.l} + Z2.l) \);  
     \( \text{Theta.lo} = -\infty; \)
     \( \text{Theta.up} = \infty; \)
     \( \text{Delta(l)} = 1; \)
     \( \text{Z2.L (l)} = \text{Subproblem_Bd.ObjVal}; \)
     
     \( \text{PI_L(l,s,i,j)} = \text{FlowLimit.m(s,i,j)}; \)
     \( Y.lo( i,j) = 0; \)
     \( Y.up( i,j) = 1; \)

   * increase the set of Benders cuts

     \( LL(l) = \text{yes}; \)

   * solving complete using MIP minimizing Z1:
### Stochastic FCTP. Bd Convergence

<table>
<thead>
<tr>
<th>Iteration</th>
<th>Lower Bound</th>
<th>Upper Bound</th>
</tr>
</thead>
<tbody>
<tr>
<td>1 to 52</td>
<td>-inf</td>
<td>+inf</td>
</tr>
<tr>
<td>53</td>
<td>250.0</td>
<td>498.9</td>
</tr>
<tr>
<td>54</td>
<td>359.5</td>
<td>498.9</td>
</tr>
<tr>
<td>55</td>
<td>381.5</td>
<td>461.5</td>
</tr>
<tr>
<td>56</td>
<td>442.9</td>
<td>461.3</td>
</tr>
<tr>
<td>57</td>
<td>449.3</td>
<td>461.3</td>
</tr>
<tr>
<td>58</td>
<td>449.3</td>
<td>461.3</td>
</tr>
<tr>
<td>59</td>
<td>449.3</td>
<td>461.3</td>
</tr>
<tr>
<td>60</td>
<td>461.3</td>
<td>461.3</td>
</tr>
</tbody>
</table>
Achtung! Achtung!

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

- Variable \( x_6 \) is degenerate (basic variable with value 0)

\[
\begin{align*}
\min z &= -3x_1 - 5x_2 \\
x_1 + x_3 &= 4 \\
2x_2 + x_4 &= 12 \\
3x_1 + 2x_2 + x_5 &= 18 \\
x_1 + x_2 + x_6 &= 8 \\
x_1, x_2, x_3, x_4, x_5, x_6 &\geq 0
\end{align*}
\]
When to use Benders decomposition?

- Variables $x_1$ complicate the solution of the problem
- Implicitly $n_1 \ll n_2$
- 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
6. **Nested Benders decomposition**

- General overview
- Applications in electric systems
- Two-stage and multistage programming
- Decomposition techniques
- Benders decomposition
- Nested Benders decomposition
- Dantzig-Wolfe decomposition
- Lagrangian relaxation
- Scenario tree
- Decomposition in two-stage and multistage stochastic programming
- Improvements in decomposition techniques
- Simulation in stochastic optimization
- Stochastic dual dynamic programming

**Nested Benders decomposition**
Nested Benders decomposition (i)

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

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

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

- At 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 $p-1$. 

```
  SUBPROBLEM 1
     Hydro output               Water value
                      ↓                      ↓
        SUBPROBLEM 2
     Hydro output               Water value
                      ↓                      ↓
        SUBPROBLEM 3
     Hydro output               Water value
                      ↓                      ↓
        SUBPROBLEM 4
```
Nested Benders decomposition. Cut deduction (i)

- Let it be this problem with 4 stages

\[
\begin{align*}
\min & \quad c_1^T x_1 + c_2^T x_2 + c_3^T x_3 + c_4^T x_4 \\
A_1 x_1 &= b_1 \\
B_1 x_1 + A_2 x_2 &= b_2 \\
B_2 x_2 + A_3 x_3 &= b_3 \\
B_3 x_3 + A_4 x_4 &= b_4 \\
x_1, x_2, x_3, x_4 &\geq 0
\end{align*}
\]

- Solve stage 4 for a proposal from stage 3

\[
\begin{align*}
\min & \quad c_4^T x_4 \\
A_4 x_4 &= b_4 - B_3 x_3^l : \pi_4 \\
x_4 &\geq 0
\end{align*}
\]
Nested Benders decomposition. Cut deduction (ii)

- Master problem of stage 3 will be

\[
\begin{align*}
\min & \quad c_3^T x_3 + \theta_4 \\
\text{s.t.} & \quad A_3 x_3 = b_3 - B_2 x_2^l : \pi_3 \\
& \quad \theta_4 + \pi_4^T B_3 x_3 \geq \pi_4^T b_4 : \eta_3 \\
& \quad x_3 \geq 0
\end{align*}
\]

- Objective function of dual problem is

\[
\max_{\pi_3, \eta_3} \pi_3^T (b_3 - B_2 x_2^l) + \eta_3^T (\pi_4^T b_4)
\]

- Master problem of simultaneous stages 3 and 4 will be

\[
\begin{align*}
\min & \quad c_3^T x_3 + c_4^T x_4 \\
\text{s.t.} & \quad A_4 x_3 = b_3 - B_2 x_2^l : \pi_3' \\
& \quad B_3 x_3 + A_4 x_4 = b_4 : \mu_3 \\
& \quad x_3, x_4 \geq 0
\end{align*}
\]

- Objective function of dual problem is

\[
\max_{\pi_3', \mu_3} \pi_3'^T (b_3 - B_2 x_2^l) + \mu_3^T b_4
\]
Nested Benders decomposition. Cut deduction (iii)

- Objective function of problem of stage 3 is a lower bound of the objective function of problem of stages 3 and 4 for each value of \( x_2 \):
  \[
  \pi^T_3 (b_3 - B_2 x_2) + \mu^T_3 b_4 \geq \pi^T_3 (b_3 - B_2 x_2) + \eta^T_3 (\pi^T_4 b_4)
  \]

- The cut to introduce in the stage 2 assuming stages 3 and 4 are a single subproblem would be:
  \[
  \theta_4 \geq \pi^T_3 (b_3 - B_2 x_2) + \mu^T_3 b_4 \geq \pi^T_3 (b_3 - B_2 x_2) + \eta^T_3 (\pi^T_4 b_4)
  \]

- The following cut is a valid cut:
  \[
  \theta_4 + \pi^T_3 B_2 x_2 \geq \pi^T_3 b_4 + \eta^T_3 (\pi^T_4 b_4)
  \]
Nested Benders decomposition. Cut deduction (iv)

- Master problem of stage 2 is

\[
\begin{align*}
\min \quad & c^T_2 x_2 + \theta_3 \\
\text{s.t.} \quad & A_2 x_2 = b_2 - B_2 x_1' : \pi_2 \\
& \theta_3 + \pi_3^T B_2 x_2 \geq \pi_3^T b_3 + \eta_3^T (\pi_4^T b_4) : \eta_2 \\
& x_2 \geq 0
\end{align*}
\]

- Objective function of dual problem is

\[
\max_{\pi_2, \theta_3} \pi_2^T (b_2 - B_2 x_1') + \eta_2^T \left[ \pi_3^T b_3 + \eta_3^T (\pi_4^T b_4) \right]
\]

- Master problem of simultaneous stages 2, 3 and 4 and the objective function of the dual is

\[
\begin{align*}
\min \quad & c^T_2 x_2 + c^T_3 x_3 + c^T_4 x_4 \\
\text{s.t.} \quad & A_2 x_2 = b_2 - B_2 x_1' : \pi_2' \\
& B_2 x_2 + A_3 x_3 = b_3 : \mu_2 \\
& B_3 x_3 + A_4 x_4 = b_4 : \mu_3 \\
& x_2, x_3, x_4 \geq 0
\end{align*}
\]

\[
\max_{\pi_2, \theta_3, \pi_2'} \pi_2'^T (b_2 - B_2 x_1') + \mu_2^T b_3 + \mu_3^T b_4
\]
Nested Benders decomposition. Cut deduction (v)

- The cut to introduce in the stage 1 would be
  \[ \theta_2 \geq \pi_2^T (b_2 - B_1 x_1) + \mu_2^T b_3 + \mu_3^T b_4 \geq \pi_2^T (b_2 - B_1 x_1) + \eta_2^T \left( \pi_3^T b_3 + \eta_3^T \left( \pi_4^T b_4 \right) \right) \]

- The following cut is a valid cut
  \[ \theta_2 + \pi_2^T B_1 x_1 \geq \pi_2^T b_2 + \eta_2^T \left( \pi_3^T b_3 + \eta_3^T \left( \pi_4^T b_4 \right) \right) \]

- Cuts for stages 2, 3 and 4 are
  \[ \theta_4 + \pi_4^T B_3 x_3 \geq \pi_4^T b_4 \]
  \[ \theta_3 + \pi_3^T B_2 x_2 \geq \pi_3^T b_3 + \eta_3^T \left( \pi_4^T b_4 \right) \]
  \[ \theta_2 + \pi_2^T B_1 x_1 \geq \pi_2^T b_2 + \eta_2^T \left( \pi_3^T b_3 + \eta_3^T \left( \pi_4^T b_4 \right) \right) \]

  \[ \theta_{p+1} + \pi_{p+1}^T B_p x_p \geq q_p = \pi_{p+1}^T b_{p+1} + \eta_{p+1}^T q_{p+1} \]
Nested Benders decomposition. Alternative cuts

- Writing the expressions of linearization around a point

\[
\theta_4 + \pi^T_4 B_3 x_3 \geq f^l_4 + \pi^T_4 B_3 x^l_3 \\
\theta_4 \geq f^l_4 + \pi^T_4 B_3 (x^l_3 - x_3)
\]

\[
\theta_3 \geq \pi^T_3 (b_3 - B_2 x_2) + \eta^T_3 (\pi^T b_1) = \\
= \pi^T_3 (b_3 - B_2 x_2 + B_2 x^l_2 - B_2 x^l_2) + \eta^T_3 (\pi^T b_1) = \\
= \pi^T_3 (b_3 - B_2 x^l_2) + \pi^T_3 B_2 (x^l_2 - x_2) + \eta^T_3 (\pi^T b_1) = \\
= f^l_3 + \pi^T_3 B_2 (x^l_2 - x_2)
\]

\[
\theta_{p+1} \geq f^l_{p+1} + \pi^T_{p+1} B_p (x^l_p - x_p) \\
\theta_{p+1} + \pi^T_{p+1} B_p x_p \geq f^l_{p+1} + \pi^T_{p+1} B_p x^l_p
\]
Nested Benders decomposition

- Generic problem to solve

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

- Problem converges when first stage does

\[
\begin{align*}
\min_{x_p, \theta_{p+1}} & \quad c^T_p x_p + \theta_{p+1} \\
A_p x_p &= b_p - B_{p-1}^i x_{p-1}^i \\
\pi^T_{p+1} B_p x_p + \theta_{p+1} &\geq f^l_{p+1} + \pi^T_{p+1} B_p x_p^j \\
x_p &\geq 0 \\
\theta_{p+1} &\equiv 0 \\
B_0 &\equiv 0 \\
\pi^j_{p+1} &\equiv 0 \\
\eta^j_{p+1} &\equiv 0
\end{align*}
\]
Convergence of a hydrothermal scheduling model

![Graph showing convergence of a hydrothermal scheduling model]
Dantzig-Wolfe decomposition
Dantzig-Wolfe decomposition

- Dantzig-Wolfe decomposition or
  - Dual decomposition (because dual information is sent),
  - Price decomposition (because master assigns prices),
  - Column generation (master increases the number of variables in each iteration),
  - Inner linearization (points are linear combination of the vertices)
- Splits PL-2 in master and subproblem
Dantzig-Wolfe decomposition

- First-stage constraints **complicate** the solution

- Implicitly $m_1 \ll m_2$
- It needs that o. f. and feasible region of the first-stage constraints will be convex
- Subproblems can be nonconvex (MIP, NLP)
- Applications:
  - Decentralized planning with central coordination
  - Unit commitment
DW Algorithm deduction (i)

- Two-stage linear programming PL-2 can be expressed as

\[
\begin{align*}
\min & \quad c_1^T x_1 \\
A_1 x_1 &= b_1 \\
x_1 &\in K
\end{align*}
\]

Being \( K \) the region defined as

\[
K = \{ x_1 \mid A_2 x_1 = b_2, x_1 \geq 0 \}
\]

- Every point of the polyhedron can be written as linear convex combination of their vertices

\[
K = \left\{ \sum_{l=1}^{v} x_l^l \lambda_l \mid \sum_{l=1}^{v} \lambda_l = 1, \lambda_l \geq 0 \right\}
\]
DW Algorithm deduction (ii)

• Therefore the complete (master) problem PL-2 can be expressed as

\[
\begin{align*}
\min_{\lambda} & \sum_{i=1}^{\nu} (c_i^T x_i) \lambda_i \\
\sum_{i=1}^{\nu} (A_i x_i) \lambda_i & = b_i : \pi_2 \\
\sum_{i=1}^{\nu} \lambda_i & = 1 : \mu \\
\lambda_i & \geq 0 \quad l = 1,\ldots,\nu
\end{align*}
\]

• Instead of computing all the vertices they are introduced iteratively

• The original complete problem is solved over a feasible region \(K\) bigger at each iteration, corresponding to the second set of constraints (the complicating constraints).

• Objective function decreases in each iteration when introducing a new variable, decreasing monotonically. It is an upper bound of that of the complete problem.
DW Algorithm deduction (iii)

- Optimality condition for the master problem: reduced costs of the non basic variables ≥ 0 in a minimization problem

\[ c_1^T x_1^* - \left( \pi_2^T \mu \right) \left( A_1 x_1^* / 1 \right) \geq 0 \]

\[ \theta_2 = \min_{x_1 \in K} \left( c_1^T - \pi_2^T A_1 \right) x_1 - \mu \]

\[ \theta_2 = \min_{x_1} \left( c_1^T - \pi_j^T A_1 \right) x_1 - \mu^j \]

\[ A_2 x_1 = b_2 \]

\[ x_1 \geq 0 \]

- Subproblem obtains in each iteration the vertex with lower reduced cost and it is incorporated into the master. When the objective function of the subproblem is positive or zero the optimum has been reached.

Second stage
DW Relaxed Master and Subproblem

- DW Relaxed Master

\[
\min_{\lambda} \sum_{l=1}^{j} (c_l^T x_l^j) \lambda_l \\
\sum_{l=1}^{j} (A_l x_l^j) \lambda_l = b_l : \pi_2 \\
\sum_{l=1}^{j} \lambda_l = 1 : \mu \\
\lambda_l \geq 0 \quad l = 1, \ldots, j
\]

- DW Subproblem

\[
\theta_2 = \min_{x_1} \left( c_1^T - \pi_2^T A_4 \right) x_1 - \mu^j \\
A_2 x_1 = b_2 \\
x_1 \geq 0
\]
DW Relaxed Master and Subproblem

• DW Relaxed Master:
  – Linear combination of the first-stage solutions and constraints (inner linearization)
  – Generates economic signals (dual variables)
  – Adds a new variable at each iteration
  – May have feasibility problems in the first iterations
  – Must be convex

• DW Subproblem:
  – Objective function is reduced costs
  – Second-stage constraints
  – Plans decentralizedly internalizing the economic signals
  – Each iteration changes the objective function
  – Can be nonconvex
**DW Optimal solution**

- The optimal solution is the linear combination of the solutions of all the iterations

\[
x_1^* = \sum_{l=1}^{j} x_1^l \lambda_l
\]

\[
z_1^* = \sum_{l=1}^{j} (c_1^T x_1^l) \lambda_l
\]
Comparison between Bd and DW

- Original linear problem PL-2: \((m_1 + m_2) \times (n_1 + n_2)\)
- For each Bd iteration
  - Increases in 1 the number of constraints of the master problem
  - Modifies the RHS of the constraints of the subproblem
  - Bd Relaxed Master: \((m_1 + j) \times (n_1 + 1)\)
  - Bd Subproblem: \(m_2 \times n_2\)
- For each DW iteration
  - Increases in 1 the number of variables of the master problem
  - Modifies the objective function of the subproblem
  - DW Relaxed Master: \((m_1 + 1) \times j\)
  - DW Subproblem: \(m_2 \times n_1\)
**DW Algorithm deduction (i)**

- **Original problem**
  \[
  \min_{x_1} c_1^T x_1 \\
  A_1 x_1 = b_1 : \pi_2'^T \\
  x_1 \in K
  \]

- **Lagrangian**
  \[
  L(x_1, \pi_2') = c_1^T x_1 + \pi_2'^T (A_1 x_1 - b_1)
  \]

- **The dual function** (concave) will be an upper bound

  \[
  \theta_2(\pi_2') = \min_{x_1 \in K} L(x_1, \pi_2') = c_1^T x_1 + \pi_2'^T (A_1 x_1 - b_1)
  = -\pi_2'^T b_1 + \min_{x_1 \in K} (c_1^T + \pi_2'^T A_1) x_1
  \]

  or

  \[
  \pi_2 = -\pi_2'
  \]

  \[
  \theta_2(\pi_2) = \min_{x_1 \in K} (c_1^T - \pi_2'^T A_1) x_1 + \pi_2'^T b_1
  \]

  \[
  A_2 x_1 = b_2 \\
  x_1 \geq 0
  \]

**SUBPROBLEM**
**DW Algorithm deduction (ii)**

- The optimum of the dual function is reached in one of the vertices

\[
\theta_2(\pi_2) = \pi_2^T b_1 + \min_{x_1} (c_1^T - \pi_2^T A_1) x_1^l \\
\quad l = 1, \ldots, \nu
\]

- It can be expressed as

\[
\begin{align*}
\theta_2(\pi_2) &\leq \pi_2^T b_1 + (c_1^T - \pi_2^T A_1) x_1^1 \\
\theta_2(\pi_2) &\leq \pi_2^T b_1 + (c_1^T - \pi_2^T A_1) x_1^2 \\
\quad \vdots \\
\theta_2(\pi_2) &\leq \pi_2^T b_1 + (c_1^T - \pi_2^T A_1) x_1^\nu
\end{align*}
\]

- Taking the dual we obtain the complete master problem

\[
\begin{align*}
\max_{\theta_2, \pi_2} &\quad \theta_2 \\
\text{s.t.} &\quad \theta_2 + (A_1 x_1^1 - b_1) \pi_2 \leq c_1^T x_1^1 : \lambda_1 \\
&\quad \theta_2 + (A_1 x_1^2 - b_1) \pi_2 \leq c_1^T x_1^2 : \lambda_2 \\
&\quad \vdots \\
&\quad \theta_2 + (A_1 x_1^\nu - b_1) \pi_2 \leq c_1^T x_1^\nu : \lambda_\nu
\end{align*}
\]
Lagrangian relaxation

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

14. Decomposition in two-stage and multistage stochastic optimization

15. Lagrangian relaxation

16. Scenario tree

17. Decomposition in two-stage and multistage stochastic

18. Lagrangian relaxation
LR Algorithm deduction (i)


- We take the optimum of the dual function

\[
\begin{align*}
\max_{\theta_2, \pi_2} & \quad \theta_2 \\
\theta_2 + (A_2 x_1^1 - b_1)^T \pi_2 & \leq c_1^T x_1^1 : \lambda_1 \\
\theta_2 + (A_2 x_1^2 - b_1)^T \pi_2 & \leq c_2^T x_1^2 : \lambda_2 \\
\vdots \\
\theta_2 + (A_2 x_1^\nu - b_1)^T \pi_2 & \leq c_1^T x_1^\nu : \lambda_\nu
\end{align*}
\]

or doing the variable change

\[
\theta_2 = b_1^T \pi_2 + \mu
\]
LR Algorithm deduction (ii)

- **LR Relaxed Master problem**

\[
\begin{align*}
\max_{\theta_2} & \quad \theta_2 + (A_i x^l - b_i)^T \pi_2 & \leq c_i^T x^l : \lambda & l = 1, \ldots, j \\
\max_{\pi_2, \mu} & \quad b_i^T \pi_2 + \mu & \leq c_i^T x^l : \lambda & l = 1, \ldots, j
\end{align*}
\]

- Constraints are called **dual cuts or Lagrangian optimality cuts**.

- The formulation is the method of **Kelly’s cutting planes**.

- To avoid unbounded subproblems, **bounding cuts are introduced**

\[
\begin{align*}
\max_{\theta_2} & \quad \delta^l \theta_2 + (A_i x^l - \delta^l b_i)^T \pi_2 & \leq c_i^T x^l : \lambda & l = 1, \ldots, j \\
\max_{\pi_2, \mu} & \quad (A_i x^l)^T \pi_2 + \delta^l \mu & \leq c_i^T x^l : \lambda & l = 1, \ldots, j
\end{align*}
\]

- \( \delta^l = 1 \) dual cuts, \( \delta_1^l = 0 \) bounding cuts
LR Algorithm deduction (iii)

• LR Subproblem

\[
\theta_2 = \min_{x_1} (c^T_1 - \pi_2^TA_1)x_1 + \pi_2^Tb_1
\]

\[
A_2x_1 = b_2
\]

\[
x_1 \geq 0
\]

\[
\theta_2 = \min_{x_1} (c^T_1 - \pi_2^TA_1)x_1 - \mu^T
\]

\[
A_2x_1 = b_2
\]

\[
x_1 \geq 0
\]
LR Relaxed Master and Subproblem

- **LR Relaxed Master:**
  - Dual of DW Relaxed Master
  - Generates economic signals (primal variables)
  - Adds a new constraint at each iteration (dual cut)
  - Objective function decreasing monotonically
  - Outer linearization of the subproblem

- **LR Subproblem:**
  - Objective function is reduced costs or total costs minus recourse price
  - Plans decentralizedly internalizing the economic signals
  - Second-stage constraints
  - Each iteration changes the objective function
  - Can be nonconvex
Comparison between DW and LR

- Original linear problem PL-2: \((m_1 + m_2) \times n_1\)
- For each LR iteration
  - Increases in 1 the number of constraints of the master problem
  - Dual simplex method
  - LR Relaxed Master: \(j \times (m_1 + 1)\)
  - Modifies the objective function of the LR subproblem
  - Primal simplex method
  - LR Subproblem: \(m_2 \times n_1\)
- Comparison with DW
  - DW: dual variables are dual for the master (degeneracy)
  - LR: dual variables are primal for the master
LR Optimal solution

- LR optimal solution is not necessarily feasible in the complicating constraints $A_i x_i = b_i$
  - Introduce a penalty in the o.f. of the subproblem (augmented langrangian). Besides, differentiability is achieved
  - Postprocessing the optimal solution
Subgradient method

- Use NLP techniques to obtain the Lagrange multipliers
  \[ \pi^{j+1}_2 = \pi^j_2 + \alpha_j p_j \]
- Search direction: gradient of the Lagrangian wrt \( \pi^j_2 \)
  \[ p_j = A_i x^j_i - b_i \]
- Updating the step length \( \alpha_j \)
  \[ \alpha_j = \frac{\beta_j \left[ \theta_2(\pi^j_2) - c^*_i x^*_1 \right]}{\sum_m \left( \sum_n a^*_m x^j_i - b_m \right)^2} \]
- Decreasing sequence 1, \( \frac{1}{2} \), \( \frac{1}{4} \), 1/8
Subgradient method

- Search direction: gradient of the Lagrangian wrt $x^*_i$

$$p_j = c^T_1 - \pi^{jT}_2 A_1$$

- Updating mechanism

$$\pi^{j+1}_2 = \pi^j_2 + \alpha_j p_j$$

- Step length

$$\alpha_j = \frac{\beta_j [c^T_1 x_1 - \pi^{jT}_2 (A x_1 - b_1) - c^T_1 x^*_1]}{\sum_j (p_j)^2}$$

- $\beta_j$ decreasing sequence 1, $\frac{1}{2}$, $\frac{1}{4}$, $\frac{1}{8}$
**LR Algorithm**

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

2. Solving the LR Relaxed Master problem

\[
\begin{align*}
\max_{b_1, \pi_2} & \quad \theta_2 \\
\delta i \theta_2 + (A_i x_i - \delta_i b_i)^T \pi_2 & \leq c_i^T x_i : \lambda_i \quad l = 1, \ldots, j
\end{align*}
\]

Obtain the solution \([\bar{z}]_2\)

3. Bounding LR Subproblem solution

\[
\theta'_2(\pi_2) = \min_{x_i} (c_i^T - \pi_2^T A_i)x_i
\]

\[
\begin{align*}
A_i x_i & \leq 0 \\
x_i & \geq 0
\end{align*}
\]

If \([\theta'_2(\pi_2)] \geq 0\) go to step 4.
If not, build a bounding cut. \([A_i x_i]^T \pi_2 \leq c_i^T x_i\]

4. Solving the LR Subproblem

\[
\begin{align*}
\theta_2 & = \min_{x_i} (c_i^T - \pi_2^T A_i)x_i + \pi_2^T b_i \\
A_i x_i & = b_i \\
x_i & \geq 0
\end{align*}
\]

\[
\begin{align*}
\theta_2 & = \min_{x_i} (c_i^T - \pi_2^T A_i)x_i - \mu' \\
A_i x_i & = b_i \\
x_i & \geq 0
\end{align*}
\]

Obtain \([x'_1]\) and build a dual cut. \([\theta_2 + (A_i x_i' - b_i)^T \pi_2 \leq c_i^T x_i]\)

5. If stopping rule \(d(\pi_2^j - \pi_2^{j-1}) < \varepsilon\) is satisfied stops. If not go to step 2.
Fixed-charge transportation problem

- Complete problem

\[
\begin{align*}
\min_{x_{ij}, y_{ij}} & \sum_{ij} \left( c_{ij} x_{ij} + f_{ij} y_{ij} \right) \\
\sum_{j} x_{ij} & \leq a_i \quad \forall i \\
\sum_{i} x_{ij} & \geq b_j \quad \forall j \\
x_{ij} & \leq M_{ij} y_{ij} \quad \forall ij \\
x_{ij} & \geq 0, y_{ij} \in \{0,1\}
\end{align*}
\]
Fixed-charge transportation problem

\[
\begin{align*}
\min_{x_{ij}, y_{ij}} & \sum_{ij} \left( c_{ij} x_{ij} + f_{ij} y_{ij} \right) + \lambda_{ij}^k \left( x_{ij} - M_{ij} y_{ij} \right) \\
\sum_{j} x_{ij} & \leq a_i \\
\sum_{i} x_{ij} & \geq b_j \\
x_{ij} & \geq 0, y_{ij} \in \{0,1\}
\end{align*}
\]

\[
\begin{align*}
\min_{x_{ij}, y_{ij}} & \sum_{ij} \left( c_{ij} + \lambda_{ij}^k \right) x_{ij} + \left( f_{ij} - \lambda_{ij}^k M_{ij} \right) y_{ij} \\
\sum_{j} x_{ij} & \leq a_i \\
\sum_{i} x_{ij} & \geq b_j \\
x_{ij} & \geq 0 \\
x_{ij} & \geq 0, y_{ij} \in \{0,1\}
\end{align*}
\]

\[
\begin{align*}
\min_{y_{ij}} & \sum_{ij} \left( f_{ij} - \lambda_{ij}^k M_{ij} \right) y_{ij} \\
y_{ij} & \in \{0,1\}
\end{align*}
\]

- LR Subproblem
  - Separable
Fixed-charge transportation problem

- LR Master Problem

\[
\begin{align*}
\max_{\theta_2, \lambda_{ij}} & \quad \theta_2 \\
\theta_2 & \leq \sum_{ij} \left( c_{ij} x_{ij}^k + f_{ij} y_{ij}^k \right) + \lambda_{ij} \left( x_{ij}^k - M_{ij} y_{ij}^k \right) \quad \forall k \\
\lambda_{ij} & \geq 0
\end{align*}
\]
Fixed-charge transportation problem. LR Convergence
Unit Commitment. LR convergence

Objective Function vs. Iterations

- Upper Bound
- Lower Bound
- Difference

0 1 2 3 4 5 6

1 32 63 94 125 156 187 218 249 280 311 342 373 404 435 466 497 528 559 590 621 652 683 714 745 776 807 838

Escuela Técnica Superior de Ingeniería ICAI
Stochastic Optimization 2017-2018
1. General overview
2. Applications in electric systems
3. Two-stage and multistage programming
4. Decomposition techniques
5. Benders decomposition
6. Nested Benders decomposition
7. Dantzig-Wolfe decomposition
8. Lagrangian relaxation
9. Scenario tree
10. Decomposition in two-stage and multistage stochastic programming
11. Improvements in decomposition techniques
12. Simulation in stochastic optimization
13. Stochastic dual dynamic programming
Objectives

• To understand
  – How uncertainty is represented
  – What is a scenario tree
  – How it is generated
Stochasticity or uncertainty

- **Origin**
  - Future information (e.g., prices or future demand)
  - Lack of reliable data
  - Measurement errors

- **In electric energy systems planning**
  - Demand (yearly, seasonal or daily variation, load growth)
  - Hydro inflows
  - Availability of generation or network elements
  - Electricity or fuel prices

- **Uncertainty relevance for each time scale**
Long term load forecasting methods (i)

- **Final use models**
  - Explain the direct use of electricity for the different users
  - Require lot of data and are dependent on their quality

- **Econometric models**
  - Use economic data to explain the demand variation
Short term load forecasting methods (ii)

- **Regression models**
  - Determine the relation of the demand with factors as humidity, temperature, day of the week
- **Time series analysis**
  - Detect the intrinsic structure of the demand: trend, seasonal and daily variation
- **Artificial neural networks**
  - Perform an nonlinear adjust of the demand as a function of previous factors
- **Fuzzy logic**
  - Introduce qualitative aspects by means of fuzzy numbers
Monthly WG capacity factor

A saying:
Marzo ventoso y abril lluvioso sacan a mayo florido y hermoso (March winds and April showers bring forth May flowers)

WG Operation hours at full capacity

Source: REE
Yearly WG Capacity factor

Source: REE
Stochastic hydro inflows

- Natural hydro inflows (clearly the most important factor in the Spanish electric system)

- Changes in reservoir volumes are significant because of:
  - stochasticity in hydro inflows
  - chronological pattern of inflows and
  - capacity of the reservoir with respect to the inflows

### Year | Hydro energy | Index | % of being exceeded
--- | --- | --- | ---
1990 | 20.3 | 0.57 | 98%
1991 | 25.4 | 0.84 | 76%
1992 | 19.5 | 0.64 | 95%
1993 | 22.8 | 0.75 | 85%
1994 | 24.2 | 0.80 | 29%
1995 | 21.7 | 0.72 | 88%
1996 | 39.4 | 1.30 | 17%
1997 | 35.6 | 1.19 | 27%
1998 | 27.1 | 0.90 | 64%
1999 | 19.8 | 0.67 | 92%
2000 | 26.2 | 0.90 | 64%
2001 | 32.9 | 1.13 | 32%
2002 | 20.9 | 0.72 | 87%
2003 | 33.2 | 1.15 | 30%
2004 | 22.7 | 0.79 | 80%
2005 | 12.9 | 0.45 | 100%
2006 | 23.3 | 0.82 | 74%
2007 | 18.4 | 0.65 | 92%
2008 | 18.9 | 0.67 | 90%
2009 | 22.3 | 0.79 | 76%
2010 | 37.4 | 1.34 | 13%
2011 | 22.6 | 0.81 | 74%
2012 | 12.7 | 0.46 | 100%
2013 | 32.6 | 1.18 | 25%
2014 | 32.4 | 1.17 | 26%
2015 | 18.9 | 0.69 | 88%

Source: REE
How uncertain are hydro inflows in Spain?

- Mean values
  24,768 GWh
  2,827 MW
Output: stochastic reservoir levels
Natural hydro inflows: (monthly) historical series

![Graph showing natural hydro inflows](image)

- **Caudal [m³/s]**
- **1970-1997 media mínima máxima**
Natural hydro inflows: (monthly) historical series

Natural hydro inflows: (monthly) historical series
Probability density function (pdf) $f(x)$
Cumulative distribution function (cdf) $F(x)$
Water inflows

- Several **measurement points** in **main different river basins**
- **Partial spatial correlation** among them
- **Temporal correlation** in each one
- No establish method for obtaining a unique multivariate probability tree
Inflows decrease in Iberian rivers in the last 60 years

How uncertain are hydro inflows in Brazil?

- Hydro inflows in Teres-Pires
- Hydro inflows in Ita
How uncertain are hydro inflows in Colombia?
Alternatives to model stochastic parameters

- Discrete probability function (i.e., scenario tree)
- Continuous or historic probability function that generates the tree by sampling (simulating) in each time period
Scenario or probability tree

- **Tree**: represents how the stochasticity is revealed over time, i.e., the different states of the random parameters and simultaneously the non anticipative decisions over time.

  **Correlation** among parameters should be taken into account

- **Scenario**: any path going from the root to the leaves

- The scenarios that share the information until a certain period do the same into the tree (non anticipative decisions)
Scenario tree

- **Nodes**: where decisions are taken.
- **Scenarios**: realizations of the random process.
Scenario tree example

In each node a decision is made and afterwards stochastic parameters are revealed.

<table>
<thead>
<tr>
<th>Period 1</th>
<th>Period 2</th>
</tr>
</thead>
<tbody>
<tr>
<td>Wet</td>
<td>Inflow: 35 m³/s Prob: 0.60</td>
</tr>
<tr>
<td>Dry</td>
<td>Inflow: 25 m³/s Prob: 0.40</td>
</tr>
<tr>
<td>Wet</td>
<td>Inflow: 20 m³/s Prob: 0.35</td>
</tr>
<tr>
<td>Dry</td>
<td>Inflow: 10 m³/s Prob: 0.65</td>
</tr>
<tr>
<td>Wet</td>
<td>Inflow: 20 m³/s Prob: 0.45</td>
</tr>
<tr>
<td>Dry</td>
<td>Inflow: 25 m³/s Prob: 0.55</td>
</tr>
<tr>
<td>Wet</td>
<td>Inflow: 35 m³/s Prob: 0.60</td>
</tr>
</tbody>
</table>

Inflow: 25 m³/s Prob: 0.55

Inflow: 35 m³/s Prob: 0.60

Inflow: 25 m³/s Prob: 0.40

Inflow: 20 m³/s Prob: 0.35

Inflow: 20 m³/s Prob: 0.45

Inflow: 10 m³/s Prob: 0.65
Recombining scenario tree example

In each node a decision is made and afterwards stochastic parameters are revealed.
Recombining scenario tree example

In each node a decision is made and afterwards stochastic parameters are revealed.
Recombining scenario tree

- The inflows depend on the scenarios in each period.
  - In the previous tree in period 2 there are four scenarios, 30, 25, 20 and 10 m³/s.
  - In the previous recombining tree, in period 2 there are only two scenarios, 30 and 15 m³/s.
Efect of the uncertainty representation

- Tree based in
  - Historical series (usually in a reduced number)
  - Synthetic series
- Tree generation based on
  - Results of the stochastic optimization in the first stage
  - Statistical properties (moments, distances) of the original series and the scenario tree
- Tree options
  - Recombining
  - Not recombining
Scenario tree trade-off

- **Big scenario tree and simplified optimization model**
  - Where do we branch the tree?

- **Small scenario tree and realistic optimization model**
Where do we branch?

- Where there are huge variety of stochastic values
  - Winter and spring in hydro inflows
- Short-term future will affect more than long-term future
  - If the scope of the model is from January to December
    branching in winter and spring will be more relevant than
    branching in autumn
Scenario tree generation (i)

- **Univariate** series *(one inflow)*
  - Distance from the cluster centroid to each series from a period to the last one

- **Multivariate** series *(several inflows)*
  - Distance from the multidimensional cluster centroid to each series of each variable from a period to the last one
Scenario tree generation (ii)

• There is no established method to obtain a unique scenario tree

• A multivariate scenario tree is obtained by neural gas clustering technique that simultaneously takes into account the main stochastic series and their spatial and temporal dependencies.

• **Contamination:** very extreme scenarios can be artificially introduced with a very low probability

• Number of scenarios generated enough for observing parameter variability
Common approach for tree generation

- Process divided into **two phases**:
  - **Generation** of a scenario tree.
    Neural gas method.
  - **Reduction** of a scenario tree.
    Using probabilistic distances.
Centroids have the **minimum distance** to their corresponding points
Their **probability** is proportional to the number of points included in the centroid
Scenario tree generation

- **Idea**
  - Minimize the distance of the scenario tree to the original series
  - Predefined maximum tree structure (2x2x2x1x1x1x1x1x1x1x1, for example)
  - Extension of clustering technique to consider many inflows and many periods

Neural gas algorithm (i)

- **Soft competitive learning method**
  - All the scenarios/centroids are adapted for each new series introduced
  - Decreasing adapting rate

- Iterative adaptation of the scenario/centroid as a function of how close is to a new series randomly chosen

- **Modifications** to this method
  - Initialization: considers the tree structure of the centroids
  - Adaptation: the modification of each node is the average of the corresponding for belonging to each scenario
Neural gas algorithm (ii)

1. Initialize the tree \( \{\omega^k\} \) with randomly chosen series.
2. Choose randomly a new series \( \omega \).
3. Compute the distance of the scenario tree to the series:
   \[
   d^k = \|\omega - \omega^k\| \quad \text{for} \quad k = 1, 2, \ldots, K
   \]
4. Sort by increasing order these distances and store in \( \omega^k \) the order of each scenario in this sequence.
5. Compute the modification of each node:
   \[
   \Delta \alpha^n = \varepsilon(j) \sum_{h \in \omega^k} h \cdot (\omega - \omega^k) \\
   = \varepsilon(j) \left( \sum_{h \in \omega^k} h \cdot (\omega - \omega^k) \right)
   \]
6. If maximum number of iterations has not been reached go to 2.
Natural inflows (I)

- Data series for almost 30 years.
- **Weekly data** in m$^3$/s.
- Corresponding to **8 measurement points** in 3 basins.
- Organized in **natural hydrological years**, from October to September.

**Maximum structure** of the tree:

- 16 scenarios
- Branches in stages 5, 9, 13 and 17
Natural inflows (II)

- Quantization error: distance from the series to the branches of the tree they belong to
Natural inflows (III)

- Relative quantization error with number of scenarios

![Graph showing relative quantization error with number of scenarios]
Natural inflows (V)

- **Data series for one hydro inflow:**

![Graph showing natural inflows]
Natural inflows (VI)

- Initial scenario tree for one hydro inflow:
Natural inflows (VII)

- Reduced scenario tree for one hydro inflow:
Natural inflows: scenario tree
Weekly load, a 4-scenario tree example

Historical series (green)

Scenario tree (black)
Weekly load, a 32-scenario tree example
Scenario tree for hydro inflows in Iceland

Hal (historical series)

Sig (historical series)

Sul (historical series)

Hal (scenario tree)

Sig (scenario tree)

Sul (scenario tree)
Decomposition in two-stage and multistage stochastic programming

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

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

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

• If \( A_2^\omega \) doesn’t depend on \( \omega \) it is called fixed resource

• Structure of the constraint matrix

\[
\begin{pmatrix}
A_1 & 0 \\
B_1^1 & A_2^1 \\
B_1^2 & A_2^2 \\
B_1^3 & A_2^3
\end{pmatrix}
\]
Deterministic equivalent problem (DEP)

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

\[
\begin{align*}
\min_{x_1, x_2, x_3} & \quad c^T_1 x_1 + p^{\omega_1} c_2^{\omega_1} x_2^{\omega_1} + p^{\omega_2} c_2^{\omega_2} x_2^{\omega_2} + p^{\omega_3} c_2^{\omega_3} x_2^{\omega_3} \\
A_1 x_1 & = b_1 \\
B_1^{\omega_1} x_1 + A_2^{\omega_1} x_2^{\omega_1} & = b_2^{\omega_1} \\
B_1^{\omega_2} x_1 + A_2^{\omega_2} x_2^{\omega_2} & = b_2^{\omega_2} \\
B_1^{\omega_3} x_1 + A_2^{\omega_3} x_2^{\omega_3} & = b_2^{\omega_3} \\
x_1, x_2^{\omega_1}, x_2^{\omega_2}, x_2^{\omega_3}, & \geq 0
\end{align*}
\]

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

- **Bd Relaxed Master monotcut** and multicut

\[
\begin{align*}
\min_{x_1, x_2} & \quad c_1^T x_1 + \theta_2 \\
A_1 x_1 &= b_1 \\
\sum_{\omega \in \Omega} p^\omega \pi_2^\omega &= b_2^\omega \\
x_1 &\geq 0
\end{align*}
\]

\[
\begin{align*}
\min_{x_1, x_2} & \quad c_1^T x_1 + \sum_{\omega \in \Omega} p^\omega \theta_2^\omega \\
A_1 x_1 &= b_1 \\
\sum_{\omega \in \Omega} p^\omega \pi_2^\omega &= b_2^\omega \\
x_1 &\geq 0
\end{align*}
\]

- **Bd Subproblem**

\[
\begin{align*}
\min_{x_2^\omega} & \quad c_2^\omega x_2^\omega \\
A_2^\omega x_2^\omega &= b_2^\omega - B_1^\omega x_1^i : \pi_2^\omega \\
x_2^\omega &\geq 0
\end{align*}
\]

One subproblem for each scenario
Monocut vs. multicut

- Monocut \((m_1 + j) \times (n_1 + 1)\)
  Multicuts \((m_1 + j\Omega) \times (n_1 + \Omega)\)
- Multicuts convenient when \(m_2\) is large and \(\Omega\) no much larger than \(n_1\). Requires less Benders iterations but more cumbersome
- Multicuts approximates independently each scenario. Monocut approximates the weighted sum of scenarios
Risk measures.
Conditional Value at Risk (CVaR). Value at Risk (VaR)

\[ CVaR = \max_{\text{VaR}, y^\omega} \left( \text{VaR} - \frac{1}{\beta} \sum_\omega p^\omega y^\omega \right) \]
\[ y^\omega \geq \text{VaR} - x^\omega, \forall \omega : q^\omega \]
\[ y^\omega \geq 0, \forall \omega \]

and its dual problem

\[ CVaR = \min_{q^\omega} \sum_\omega x^\omega q^\omega \]
\[ \sum_\omega q^\omega = 1 : \text{VaR} \]
\[ q^\omega \leq \frac{p^\omega}{\beta}, \forall \omega : -y^\omega \]
\[ q^\omega \geq 0 \]

**Variables**
- \( y^\omega \) profit value below VaR
- \( q^\omega \) modified probability of scenario \( \omega \)

**Parameters**
- \( x^\omega \) UC objective function (profit)
- \( p^\omega \) probability of each scenario \( \omega \)
- \( \beta \) probability (5 %)
Risk constrained profit-based UC

\[
\begin{align*}
\max_{x^\omega, y^\omega, CVaR, VaR} & \quad (1 - \mu) \sum_{\omega} p^\omega x^\omega + \mu CVaR \\
CVaR & = VaR - \frac{1}{\beta} \sum_{\omega} p^\omega y^\omega \\
y^\omega & \geq VaR - x^\omega, \forall \omega \\
y^\omega & \geq 0, \forall \omega \\
x^\omega & \in K
\end{align*}
\]

\[
K = \left\{ x^\omega = \sum_l x^{l\omega} \lambda^l \mid \sum_l \lambda^l = 1, \lambda^l \geq 0, \forall l \right\}
\]

\[K\] UC constraints
\[
\mu \text{ risk weight factor}
\]

Complicating constraints because link all the scenarios

\[x^\omega\] can be expressed as a linear combination of the vertices
Risk constrained profit-based UC

- DW Complete master problem

\[
\max_{\lambda^l, \gamma^\omega, \text{VAR}} (1 - \mu) \sum_{\omega} p^\omega \sum_l x^{l\omega} \lambda^l + \mu (\text{VAR} - \frac{1}{\beta} \sum_{\omega} p^\omega y^\omega)
\]

\[
y^\omega \geq \text{VAR} - \sum_l x^{l\omega} \lambda^l, \forall \omega : \pi^\omega
\]

\[
\sum_l \lambda^l = 1 : \theta
\]

\[
\lambda^l \geq 0, \forall l
\]

\[
y^\omega \geq 0, \forall \omega
\]

\[
\max \left( (1 - \mu) \sum_{\omega} p^\omega x^{l\omega} - \frac{\mu}{\beta} p^\omega \mu \right) \left( \begin{array}{c} \lambda^l \\ \gamma^\omega \\ \text{VAR} \end{array} \right)
\]

\[
\left( \begin{array}{ccc} x^{l\omega} & 1 & -1 \\ 1 & 0 & 0 \end{array} \right) \left( \begin{array}{c} \lambda^l \\ \gamma^\omega \\ \text{VAR} \end{array} \right) \geq \left( \begin{array}{c} 0 \\ 1 \end{array} \right)
\]

\[
\lambda^l \geq 0, \forall l
\]

\[
y^\omega \geq 0, \forall \omega
\]

\[\pi^\omega, \theta \text{ dual variables}\]
**Risk constrained profit-based UC**

- Taking the dual

\[
\min \theta \\
\begin{pmatrix}
x^{l\omega} & 1 \\
1 & 0
\end{pmatrix}
\begin{pmatrix}
\pi^\omega \\
\theta
\end{pmatrix} \geq \begin{pmatrix}
(1 - \mu) \sum \omega p^\omega x^{l\omega} \\
- \frac{\mu}{\beta} p^\omega
\end{pmatrix}
\]

\[\pi^\omega \leq 0, \forall \omega\]

\[
\min_{\pi^\omega, \theta} \theta \\
\theta \geq (1 - \mu) \sum \omega p^\omega x^{l\omega} - \sum \omega x^{l\omega} \pi^\omega \\
\pi^\omega \geq - \frac{\mu}{\beta} p^\omega \\
d - \sum \omega \pi^\omega = \mu \\
\pi^\omega \leq 0, \forall \omega
\]
Risk constrained profit-based UC

- LR Relaxed master problem

  If we define \( q^\omega = -\frac{\pi^\omega}{\mu} \)

\[
\begin{align*}
\min_{q^\omega, \theta} & \quad \theta \geq (1 - \mu) \sum_\omega p^\omega x_j^\omega + \mu \sum_\omega x_j^\omega q^\omega, \forall j \\
& \quad 0 \leq q^\omega \leq \frac{p^\omega}{\beta}, \forall \omega \\
& \quad \sum_\omega q^\omega = 1
\end{align*}
\]

- LR subproblem

\[
\begin{align*}
\max_{x^\omega} (1 - \mu) \sum_\omega p^\omega x^\omega + \mu \sum_\omega q^\omega x^\omega \\
& \quad x^\omega \in K
\end{align*}
\]
Multistage stochastic linear programming PLE-P

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

\[
\min_{x_{p}^{p}} \sum_{p=1}^{P} \sum_{\omega_p \in \Omega} p_{\omega_p}^{\omega_p} c_{p}^{\omega_p} T x_{p}^{\omega_p} \\
B_{p-1}^{\omega_{p-1}} x_{p-1}^{\omega_{p-1}} + A_{p}^{\omega_{p}} x_{p}^{\omega_{p}} = b_{p}^{\omega_{p}} \quad p = 1, \ldots, P \\
x_{p}^{\omega_{p}} \geq 0 \\
B_{0}^{\omega_{p}} \equiv 0
\]

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

\[
\begin{array}{c c c c}
A_1 & A_1 & A_1 & A_1 \\
B_1 & B_2 & A_2 & A_2 \\
B_1 & B_2 & A_2 & A_2 \\
& B_2 & A_2 & A_2 \\
& & B_3 & A_3 \\
& & & B_4 \\
& & & & B_5 \\
& & & & & B_6 \\
\end{array}
\]
Multistage stochastic problem. Nested Benders decomposition
Decomposition of PLE-P

- NBd Relaxed Master monocut

\[
\begin{align*}
\min_{x_p^\omega, \theta_{p+1}^\omega} & \quad c_p^\omega x_p^\omega + \theta_{p+1}^\omega \\
A_p^\omega x_p^\omega & = b_p^\omega - B_p^\omega x_{p-1}^\omega \\
\sum_{k \in d(\omega_p)} p^\omega \pi_{p+1}^{klT} B_p^\omega x_p^\omega + \theta_{p+1}^\omega & \geq q_{p+1}^\omega = \sum_{k \in d(\omega_p)} p^\omega \left( \pi_p^{MT} b_{p+1}^k + \eta_{p+1}^{MT} q_{p+1}^k \right) \\
x_p^\omega & \geq 0
\end{align*}
\]

where \(a(\omega_p)\) is ancestor and \(d(\omega_p)\) is descendant of a given subproblem
Stochastic multistage decomposition

Step 0
Set \( I_0^\xi = J_0^\xi = 0 \). Set \( \theta_0^\xi \equiv 0 \) at the initial iteration.

Step 1
Forward pass:
Repeat for \( t = 1, \ldots, T \)
\[ \quad \]
Repeat for each node \( \xi_t \) of stage \( t \)
\[ \quad \]
Solve \( (RP_t^\xi) \)
\[ \quad \]
If feasible: obtain solution \( x_t^\xi \)
\[ \quad \]
If \( t = 1 \) obtain lower bound \( z = v(RP_1^\xi) \)
\[ \quad \]
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 \( \bar{z} - \hat{z} < \text{tol} \) stop, \( x_t^\xi \) is optimal solution, else go to Step 4.

Step 4
Backward pass:
Repeat for \( t = T', \ldots, 1 \)
\[ \quad \]
Repeat for each node \( \xi_t \) of stage \( t \)
\[ \quad \]
Solve \( (RP_t^\xi) \)
\[ \quad \]
If feasible: obtain objective \( \theta_t^{\xi,i} = v(RP_t^\xi) \) and dual values \( \pi_t^{\xi,i} \)
\[ \quad \]
Augment \( I_t^\xi = I_t^\xi + 1 \)
\[ \quad \]
If infeasible: obtain sum of infeasibilities \( \bar{\theta}_t^{\xi,i} \) and dual values \( \bar{\pi}_t^{\xi,i} \)
\[ \quad \]
Augment \( J_t^\xi = J_t^\xi + 1 \)
\[ \quad \]
Go to step 1.
Improvements in decomposition techniques

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

- On the **problem solution**:
  - Master problem solution time (solution time can be long because of size, integer variables or a large number of cuts)
  - Subproblem solution time (solution time can be long because of a large number of scenarios or a large number of constraints)

- On the **decomposition algorithm**
  - Decrease the number of iterations

- **Practical benefit achieved has to be assessed on a case-by-case basis**
  - Most of them involve trade-offs that must be assessed individually
Optimization technique for any problem

- Optimization method used for the problems
  Problems solved many times with modifications.
  In Benders decomposition the master problem adds constraints and the subproblem changes the constraint RHS.
  - Simplex dual method is the initial candidate. Try primal simplex method or interior point method.
  - Use the best solver (CPLEX, GUROBI, XPRESS)
  - Solver tuning: use of non default values of solver parameters
  - Warm start: use of a previous basis (controlled by option BRATIO in GAMS) for the same GAMS model
  - Use of initial point taken from the deterministic equivalent problem for a scenario.
Optimization technique for **MIP master problem (i)**

- **Master problem relaxations**
  - Binary variables complicate the resolution, so solving the LP relaxation is much quicker
  - It yields valid Benders cuts
  - Progressive discretization of variables to improve convergence

---

Optimization technique for master problem (ii)

- **Sub-optimal MIP master problem solution**
  - Early terminations of the master problem might improve convergence
  - Using any feasible solution
  - Rounding linearized solutions
    (feasibility must be checked)

- **Advanced start**: provide with initial promising solutions that might generate preliminary cuts for the first iterations prior to initiating a formal Benders algorithm

- **Box-step method**: limit the difference between consecutive proposals

- **Introduction of additional constraints**:
  - Ideally they are constraints that should be met at the optimum, so they only eliminate not useful zones of the feasible region
    - From expert opinion
    - Data mining
Optimization technique for MIP master problem (iii)

- Use of a more suitable technique
  - E.g. Constraint programming and logic-based methods
    - In many cases, complex problems include many logical constraints that make use of auxiliary binary variables
    - This greatly complicates the problem
    - There are techniques that have been specially developed for these problems, where the logical constraints are included explicitly (e.g. LOGMIP)
  
- Alternative strategies to find master proposals
  - Metaheuristic techniques can be applied to find near-optimal solutions

- Local branching: divide the feasible region of the master problem into several sub-areas and searches in a neighborhood
  - If a better solution is found, the upper bound will be improved
  - If not, the explored solutions will result in an improved lower bound (more cuts)
Alternative optimization technique for subproblem

- **Bunching**: if the second-stage scenarios are similar we can solve only one scenario and calculate the others using the calculated sensitivities

- **GAMS GUSS** (Gather-Update-Solve-Scatter)
  - Use of sensitivity analysis for solving many similar problems

- Application of **specific solution algorithms** or even a series of **increasingly accurate versions** (problem formulation) of the subproblem (as long as they have increasing values of the o.f.) (e.g., lossless power flow for the first iterations and then with losses)

- Sub-optimal subproblem solutions (**Zakeri’s cuts**)
  - Any unfeasible solution in the subproblem will give a valid cut (can use IPM)
Reduction of master Benders cuts (i)

- Extracting non-dominated cuts / Pareto-optimal cuts
  - A cut or constraint dominates another if any evaluation of first stage decisions is larger than or equal to the previous one
- Removing inactive cuts
  - Dynamically defining the master problem so that only the cuts that are likely to be active constraints are taken into account
Generation of master Benders cuts (ii)

- Generating covering cuts
  - Generating cuts so that they carry the maximum amount of information possible
  - They include the maximum possible number of 1st stage variables

- Minimal Infeasible Subsystems (MIS) can be used to modify the way feasibility cuts are calculated
  - Instead of minimizing the sum of infeasibilities the problem minimizes the number of equations that are infeasible
  - This enables faster convergence in some cases
  - Conversely, if most of the solutions are infeasible, it is possible to keep a maximum feasible set to derive optimality cuts to better guide the search
Generation of master Benders cuts (ii)

- Type of cut formulation
  - Linear or nonlinear type (linearization around a point)

- Partitioning the scenario tree for subproblem solution
  - Cut aggregation: monocut or multicut
    - More cuts $\Rightarrow$ more information to the master $\Rightarrow$ less iterations.
    - More variables $\Rightarrow$ more constraints in the master
  - Divide the scenario tree for reducing total solution time, dealing with:
    - Decide size of subproblems
      - 3 small subproblems vs. 1 large subproblem vs. 2 medium subproblem
    - Number of cuts
    - Number of Benders iterations
Scenario tree for subproblem solution (i)

- **Tree partition or node aggregation** *(multicoordination)*
  - Decompose as less as possible
    - Subtrees defined by the tree breaks
  - 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
Scenario tree for subproblem solution (ii)

- **Node** and **scenario** partition
Scenario tree for subproblem solution (iii)

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

Original tree (asymmetric)  Subtrees from leaves to root

Scenario tree for subproblem solution for two-stage problem (iv)

- **Subtree partition**
  - Two type of scenarios: wind scenarios and system state scenarios.
    - **Second-stage scenarios** can be arranged differently
  - In general, the most efficient arrangement cannot be known beforehand (tradeoff between the accuracy of the cuts and solution time for the master problem)

(a) Benders’ scenario decomposition by both wind scenarios and system states
(b) Benders’ scenario decomposition by system states
(c) Benders’ scenario decomposition by wind scenarios

ws: wind scenario
ss: system state
Scenario tree for subproblem solution for two-stage problem (v)

- **Scenario aggregation**: the second-stage scenario corresponding to the scenarios with the highest impact on the final design are added to the master problem
  - The master problem proposes solutions that are closer to the optimum
  - Convergence speed can be increased
  - In this case, the most probable states are added to the master
    - Base case system state (no failures)
    - High wind

\[ \text{ws: wind scenario} \]
\[ \text{ss: system state} \]
Scenario 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**)

---

Benders decomposition in grid computing

- Distributed computing

- GAMS grid
  - Use of multiple cores of a computer
Benders decomposition algorithm extensions

- To integer subproblems:
  - Decomposition for multistage problems with integer variables in each stage. Convex hull calculation
Further improvements

- Decrease number of subproblem evaluations
  - *Simulation in stochastic optimization*

- Approximate recourse function by an analytic expression to avoid subproblem evaluation
  - *Multivariate nonlinear regression*
1. General overview
2. Applications in electric systems
3. Two-stage and multistage programming
4. Decomposition techniques
5. Benders decomposition
6. Nested Benders decomposition
7. Dantzig-Wolfe decomposition
8. Lagrangian relaxation
9. Scenario tree
10. Decomposition in two-stage and multistage stochastic programming
11. Improvements in decomposition techniques
12. Simulation in stochastic optimization
13. Stochastic dual dynamic programming

Simulation in stochastic optimization
Why do we need simulation?

- It is used when the number of values of random parameters is too high.
- Computation of expectation in the recourse function (multicut) 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).
Types of sampling

- **External sampling**
  - Samples are taken to reduce the scenario tree size and then we solve the stochastic optimization problem
  - In nested Benders decomposition we take samples in the forward pass (when making proposals to the descendants)

- **Internal sampling**
  - Samples are taken at the same time that we solve the stochastic optimization problem
  - In a two-stage planning problem, the expected value for the second state is substituted by the sample mean of the second stage
Monte Carlo simulation

- If parameters are independent: the joint probability density function if the product of all the probability density functions
- Computation of sample mean, mean variance, confidence interval.
- Stop sampling when confidence interval of the sample mean of the second stage objective function lower than a certain threshold.
- Quadratic behavior (multiply by 4 the number of samples to half the confidence interval)
- Events with low probability but large values of the objective function cause large variances. Therefore, many samples are needed
Stages in general Monte Carlo simulation

1. Random variable generation
2. Simulation or parameter sampling
3. (Variance Reduction Techniques)
4. Results collection
5. Stop the Monte Carlo sampling process
Monte Carlo simulation. Assumptions

- **Expected cuts are no longer supporting hyperplanes** of the convex recourse function, they may intersect

- **Assumptions** in the sampling process
  - Error in Benders cuts are in the RHS, not in slopes, and their variance is the same for the objective function
  - Master problem has the same basis independently of the cut RHS
  - Benders cuts of different iterations are statistically independent
Monte Carlo simulation. Simple convergence criterion

- **Lower and upper bounds are random parameters**
  - Lower upper bound is the upper bound of the smaller mean of all the iterations
  - Upper lower bound is the last lower bound
  - Variance of each bound associated to the objective function of the subproblems

- **Convergence criterion** for a number of samples: confidence interval of the difference of bounds contains the 0.

- **Confidence interval of the optimal solution** defined by the lower limit of interval of the lower bound and the upper limit of interval of the upper bound. It has to be smaller than a threshold. If not increase the number of samples.
Variance reduction techniques VRT (i)

- Reduce the size of the mean confidence interval without perturbing its value for the same number of samples or, alternatively, achieve the desired precision with lower sampling effort.
- Usually, it is impossible to know beforehand which if going to be the variance reduction or even if it is going to be reduced. You must experiment considering the system under analysis.
- The use of VRT can be understood as a way to take advantage from the system information.
- They imply a computational cost to do some preliminary computation or complementary computation during the simulation process.
Variance reduction techniques VRT (ii)

- **Common random numbers or correlated sampling or comparative simulation or matched pairs**
  - Samples are done for different system configurations with the same set of random numbers being used, each number for the same random variable in the different samples.

- **Antithetic variables**
  - The basic idea is to introduce a negative correlation between two consecutive samples.
  - It consists in the use of complementary random numbers in two consecutive samples.
Variance reduction techniques VRT (iii)

- **Control variable**
  - The basic idea is to use the results of a simpler model to predict or explain part of the variance of the mean to estimate.
  - A preliminary computation is needed to estimate the mean of the control variable.
  - This computation has to be very quick with respect to the one of the variable to estimate.
Variance reduction techniques VRT (iv)

- **Importance sampling**
  - Replaces the original random variable by another one with the same mean and lower variance.
  - The sampling probability density function is modified to center it in the interesting zone.
  - Avoids sampling frequent values but not interesting.

- **Stratified sampling**
  - Intuitive idea similar to previous one but in discrete version.
  - Take more samples of the random variables in the interesting zones.
  - Variance is reduced by concentrating the sampling effort in the relevant strata.
Stochastic dual dynamic programming

1. General overview
2. Applications in electric systems
3. Two-stage and multistage programming
4. Decomposition techniques
5. Benders decomposition
6. Nested Benders decomposition
7. Dantzig-Wolfe decomposition
8. Lagrangian relaxation
9. Scenario tree
10. Decomposition in two-stage and multistage stochastic programming
11. Improvements in decomposition techniques
12. Simulation in stochastic optimization
13. Stochastic dual dynamic programming
Stochastic Dual Dynamic Programming (SDDP)

- Nested Benders decomposition with
  - **Forward pass:** scenario sampling instead of solving all the scenarios (sample average approximation, SAA)
  - **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
    - If the stopping criterion is 1% for a confidence interval with a 95% of confidence level. The algorithm stops with we are sure with a confidence level of 95% that the relative difference between upper and lower bounds is lower than 1%


## Stochastic Dual Dynamic Programming (SDDP)

**Step 0**
Set $I_{0}^{\xi} = J_{0}^{\xi} = 0$. Set $\theta_{0}^{\xi} \equiv 0$ at the initial iteration.

**Step 1**
Simulate $N$ scenarios $(h_{n}^{\xi})_{n}$, $n : 1, ..., N$, $t = 1, ..., T$.

**Forward pass:**
- Repeat for $n : 1, ..., N$.
  - Repeat for $t = 1, ..., T$.
    - Solve $(RP_{t}^{\xi})$ with r. hand side value $(h_{t}^{\xi})_{n}$ and obtain solution $(x_{t}^{\xi})_{n}$.
    - If $t = 1$, obtain lower bound $\underline{z} = v(RP_{1}^{\xi})$.
    - 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. $\bar{z} = \frac{1}{N} \sum_{i=1}^{I} c_{i} (x_{i}^{\xi})_{n}$.

**Step 3**
(stopping rule)
If $\bar{z} - \underline{z} < tol$, stop, $x_{t}^{\xi}$ is optimal solution, else go to Step 4.

**Step 4**
Backward pass
- Repeat for $t = T, ..., 1$.
  - Repeat for each node $\xi_{i}$ of stage $t$.
    - Repeat for each proposal obtained in forward pass, modifying the right hand side value of subproblem $(RP_{t}^{\xi})$.
    - Solve $(RP_{t}^{\xi})$.
    - If feasible: obtain objective $\theta_{t}^{\xi,i} = v(RP_{t}^{\xi})$ and dual values $\pi_{t}^{\xi,i}$.
      - Augment $I_{i}^{\xi} = I_{i}^{\xi} + 1$.
      - If infeasible: obtain sum of infeasibilities $\bar{\theta}_{t}^{\xi,i}$ and dual values $\bar{\pi}_{t}^{\xi,i}$.
        - Augment $J_{i}^{\xi} = J_{i}^{\xi} + 1$.
  - Go to step 1.
Stochastic convergence in SDDP (i). Case 1

Stop the algorithm if the lower bound enters into the confidence interval of the upper bound for a confidence level of 5%.

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

Average upper bound of the last 100 iterations.
Stochastic convergence in SDDP (ii)
Stochastic convergence in SDDP (ii). Case 2
## Comparison of decomposition methods

<table>
<thead>
<tr>
<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>
</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>
</tr>
<tr>
<td>Solution of the deterministic equivalent problem</td>
<td>Impossible to solve the deterministic equivalent problem</td>
</tr>
<tr>
<td>Multicut</td>
<td>Multicut</td>
</tr>
<tr>
<td>Flexible node aggregation (tree partition)</td>
<td>Rigid node aggregation (tree partition) conditioned by the branching periods</td>
</tr>
<tr>
<td>In forward pass all the scenarios are solved</td>
<td>In forward pass only one scenario is solved</td>
</tr>
<tr>
<td>Deterministic stopping criterion</td>
<td>Stochastic stopping criterion</td>
</tr>
<tr>
<td>Exponential time increase with number of scenarios</td>
<td>Linear time increase with number of scenarios</td>
</tr>
</tbody>
</table>
Prof. Andres Ramos

https://www.iit.comillas.edu/aramos/
Andres.Ramos@comillas.edu
arght@mit.edu