Optimization problems with Decomposable structure

Jalal Kazempour
Technical University of Denmark (DTU)
Learning Objectives

After this session the participants are expected to be able to:

• Explain the need for decomposition

• Identify whether each optimization problem is decomposable or not (if so, how?)
Decomposition

Main idea

Original (non-decomposed) optimization problem with decomposable structure

Decomposition

Decomposed optimization problem 1
Decomposed optimization problem 2

\[ \ldots \]

Decomposed optimization problem \( n \)

Each decomposed problem is easier-to-solve than the original (non-decomposed) problem!

Jalal Kazempour
Motivation

• Why do we need decomposition in power systems?
Decomposition

Motivation

• Why do we need decomposition in power systems?
  ▪ Operation problems (e.g., unit commitment)

  ▪ Planning problems (e.g., expansion)
Why do we need decomposition in power systems?

- Operation problems (e.g., unit commitment)
  - Need to be computationally tractable
  - Need to be solved in a specific time period

- Planning problems (e.g., expansion)
  - Need to be computationally tractable
Optimization problems with decomposable structure:

- Problems with complicating variable(s):

- Problems with complicating constraint(s):
Optimization problems with decomposable structure:

- Problems with complicating variable(s):
  The original problem is decomposed if the complicating variables are fixed to given values!

- Problems with complicating constraint(s):
  The original problem is decomposed if the complicating constraints are relaxed (removed)!
Decomposition

Decomposable structure

Optimization problems with decomposable structure:

- **Problems with complicating variable(s):**
  
The original problem is decomposed if the complicating variables are fixed to given values!

- **Problems with complicating constraint(s):**
  
The original problem is decomposed if the complicating constraints are relaxed (removed)!

In the literature, “complicating” variables/constraints are also called as “coupling” variables/constraint!

Jalal Kazempour
Decomposition

Features of decomposition techniques

- Iterative solution techniques

- Original optimization problem decomposes to:
  - A single master problem (not always!)
  - A set of subproblems
Decomposable Structures

Optimization problems with complicating constraint(s)

Example: A linear programming (LP) as the original problem
Decomposable Structures

Optimization problems with complicating constraint(s)

Example: A linear programming (LP) as the original problem

Minimize

\[ A_1 x_1 + A_2 x_2 + A_3 x_3 + B_1 y_1 + B_2 y_2 + C_1 z_1 + C_2 z_2 + C_3 z_3 \]

Subject to

\[ E_{11} x_1 + E_{12} x_2 + E_{13} x_3 \geq F_1 \]
\[ E_{21} x_1 + E_{22} x_2 + E_{23} x_3 \geq F_2 \]
\[ E_{31} y_1 + E_{32} y_2 \geq F_3 \]
\[ E_{41} z_1 + E_{42} z_2 + E_{43} z_3 \geq F_4 \]
\[ E_{51} z_1 + E_{52} z_2 + E_{53} z_3 \geq F_5 \]
\[ E_{61} x_1 + E_{62} x_2 + E_{63} x_3 + E_{64} y_1 + E_{65} y_2 + E_{66} z_1 + E_{67} z_2 + E_{68} z_3 \geq F_6 \]
Decomposable Structures

Optimization problems with complicating constraint(s)

Example: A linear programming (LP) as the original problem

Minimize
\[
A_1 x_1 + A_2 x_2 + A_3 x_3 + B_1 y_1 + B_2 y_2 + C_1 z_1 + C_2 z_2 + C_3 z_3
\]

Subject to
\[
E_{11} x_1 + E_{12} x_2 + E_{13} x_3 \geq F_1 \\
E_{21} x_1 + E_{22} x_2 + E_{23} x_3 \geq F_2 \\
E_{31} y_1 + E_{32} y_2 \geq F_3 \\
E_{41} z_1 + E_{42} z_2 + E_{43} z_3 \geq F_4 \\
E_{51} z_1 + E_{52} z_2 + E_{53} z_3 \geq F_5 \\
E_{61} x_1 + E_{62} x_2 + E_{63} x_3 + E_{64} y_1 + E_{65} y_2 + E_{66} z_1 + E_{67} z_2 + E_{68} z_3 \geq F_6
\]

Jalal Kazempour
Decomposable Structures

Optimization problems with complicating constraint(s)

Example: A linear programming (LP) as the original problem

Minimize \( A_1 x_1 + A_2 x_2 + A_3 x_3 + B_1 y_1 + B_2 y_2 + C_1 z_1 + C_2 z_2 + C_3 z_3 \)

Subject to

Constraints including only variables \( x \)
\[
\begin{align*}
E_{11} x_1 + E_{12} x_2 + E_{13} x_3 & \geq F_1 \\
E_{21} x_1 + E_{22} x_2 + E_{23} x_3 & \geq F_2
\end{align*}
\]

Constraint including only variables \( y \)
\[E_{31} y_1 + E_{32} y_2 \geq F_3\]

Constraints including only variables \( z \)
\[
\begin{align*}
E_{41} z_1 + E_{42} z_2 + E_{43} z_3 & \geq F_4 \\
E_{51} z_1 + E_{52} z_2 + E_{53} z_3 & \geq F_5
\end{align*}
\]

\[
\begin{align*}
E_{61} x_1 + E_{62} x_2 + E_{63} x_3 + E_{64} y_1 + E_{65} y_2 + E_{66} z_1 + E_{67} z_2 + E_{68} z_3 & \geq F_6
\end{align*}
\]
Decomposable Structures

Optimization problems with complicating constraint(s)

Example: A linear programming (LP) as the original problem

Minimize
\[
A_1 x_1 + A_2 x_2 + A_3 x_3 + B_1 y_1 + B_2 y_2 + C_1 z_1 + C_2 z_2 + C_3 z_3
\]

Subject to

Constraints including only variables \( x \)
\[
\begin{align*}
E_{11} x_1 + E_{12} x_2 + E_{13} x_3 & \geq F_1 \\
E_{21} x_1 + E_{22} x_2 + E_{23} x_3 & \geq F_2 \\
E_{31} y_1 + E_{32} y_2 & \geq F_3
\end{align*}
\]

Constraints including only variables \( y \)
\[
E_{41} z_1 + E_{42} z_2 + E_{43} z_3 \geq F_4
\]
\[
E_{51} z_1 + E_{52} z_2 + E_{53} z_3 \geq F_5
\]

Constraints including only variables \( z \)
\[
E_{61} x_1 + E_{62} x_2 + E_{63} x_3 + E_{64} y_1 + E_{65} y_2 + E_{66} z_1 + E_{67} z_2 + E_{68} z_3 \geq F_6
\]
Decomposable Structures

Optimization problems with complicating constraint(s)

Example: A linear programming (LP) as the original problem

Minimize

\[ A_1 x_1 + A_2 x_2 + A_3 x_3 + B_1 y_1 + B_2 y_2 + C_1 z_1 + C_2 z_2 + C_3 z_3 \]

Subject to

Constraints including only variables \( x \)

\[ E_{11} x_1 + E_{12} x_2 + E_{13} x_3 \geq F_1 \]
\[ E_{21} x_1 + E_{22} x_2 + E_{23} x_3 \geq F_2 \]

Constraint including only variables \( y \)

\[ E_{31} y_1 + E_{32} y_2 \geq F_3 \]

Constraints including only variables \( z \)

\[ E_{41} z_1 + E_{42} z_2 + E_{43} z_3 \geq F_4 \]
\[ E_{51} z_1 + E_{52} z_2 + E_{53} z_3 \geq F_5 \]

\[ E_{61} x_1 + E_{62} x_2 + E_{63} x_3 + E_{64} y_1 + E_{65} y_2 + E_{66} z_1 + E_{67} z_2 + E_{68} z_3 \geq F_6 \]

This is complicating constraint: If relaxed (removed), then the original problem decomposes to three smaller optimization problems (subproblems)!

Jalal Kazempour
Decomposable Structures

*Optimization problems with complicating constraint(s)*

**Example:** A linear programming (LP) as the original problem

**Subproblem 1:**

Minimize \( A_1 x_1 + A_2 x_2 + A_3 x_3 \)

Subject to

Constraints including only variables \( x \)

\[
E_{11} x_1 + E_{12} x_2 + E_{13} x_3 \geq F_1 \\
E_{21} x_1 + E_{22} x_2 + E_{23} x_3 \geq F_2
\]
Decomposable Structures

Optimization problems with complicating constraint(s)

Example: A linear programming (LP) as the original problem

Subproblem 2:

Minimize $B_1 y_1 + B_2 y_2$

Subject to

Constraint including only variables $y$

$E_{31} y_1 + E_{32} y_2 \geq F_3$

Jalal Kazempour
Decomposable Structures

Optimization problems with complicating constraint(s)

Example: A linear programming (LP) as the original problem

Subproblem 3:

Minimize \( C_1 z_1 + C_2 z_2 + C_3 z_3 \)

Subject to

Constraints including only variables \( z \):

\[
E_{41} z_1 + E_{42} z_2 + E_{43} z_3 \geq F_4
\]
\[
E_{51} z_1 + E_{52} z_2 + E_{53} z_3 \geq F_5
\]

Jalal Kazempour
Decomposable Structures

Optimization problems with complicating constraint(s)

Example: A linear programming (LP) as the original problem

Decomposable matrix with complicating constraint(s)

\[
\begin{align*}
A^T & \quad B^T & \quad C^T \\
\text{Subject to} & & \\
E[1] & & \\
E[2] & & \\
E[3] & & \\
\end{align*}
\]

\[
\begin{align*}
x & \\
y & \\
z & \\
F[1] & \\
F[2] & \\
F[3] & \\
F[4]
\end{align*}
\]

Complicating constraint

Jalal Kazempour
Decomposable Structures

Optimization problems with complicating variable(s)

Example: A linear programming (LP) as the original problem
Decomposable Structures

Optimization problems with complicating variable(s)

Example: A linear programming (LP) as the original problem

Minimize

\[ A_1 x_1 + A_2 x_2 + B_1 y_1 + B_2 y_2 + C_1 z_1 + C_2 z_2 + D_1 \beta \]

Subject to

\[ E_{11} x_1 + E_{12} x_2 + E_{13} \beta \geq F_1 \]
\[ E_{21} x_1 + E_{22} x_2 + E_{23} \beta \geq F_2 \]
\[ E_{31} y_1 + E_{32} y_2 + E_{33} \beta \geq F_3 \]
\[ E_{41} z_1 + E_{42} z_2 + E_{43} \beta \geq F_4 \]
\[ E_{51} z_1 + E_{52} z_2 + E_{53} \beta \geq F_5 \]
Decomposable Structures

 Optimization problems with complicating variable(s)

Example: A linear programming (LP) as the original problem

Minimize \[ A_1 x_1 + A_2 x_2 + B_1 y_1 + B_2 y_2 + C_1 z_1 + C_2 z_2 + D_1 \beta \]

Subject to

\[ E_{11} x_1 + E_{12} x_2 + E_{13} \beta \geq F_1 \]
\[ E_{21} x_1 + E_{22} x_2 + E_{23} \beta \geq F_2 \]
\[ E_{31} y_1 + E_{32} y_2 + E_{33} \beta \geq F_3 \]
\[ E_{41} z_1 + E_{42} z_2 + E_{43} \beta \geq F_4 \]
\[ E_{51} z_1 + E_{52} z_2 + E_{53} \beta \geq F_5 \]
Decomposable Structures

*Optimization problems with complicating variable(s)*

**Example**: A linear programming (LP) as the original problem

Minimize  \[
A_1 x_1 + A_2 x_2 + B_1 y_1 + B_2 y_2 + C_1 z_1 + C_2 z_2 + D_1 \beta
\]

Subject to

\[
E_{11} x_1 + E_{12} x_2 + E_{13} \beta \geq F_1
\]
\[
E_{21} x_1 + E_{22} x_2 + E_{23} \beta \geq F_2
\]
\[
E_{31} y_1 + E_{32} y_2 + E_{33} \beta \geq F_3
\]
\[
E_{41} z_1 + E_{42} z_2 + E_{43} \beta \geq F_4
\]
\[
E_{51} z_1 + E_{52} z_2 + E_{53} \beta \geq F_5
\]

\(\beta\) is a **complicating variable**, i.e., if it is fixed to a given value (\(\beta^{\text{fixed}}\)), then the original problem decomposes to 3 smaller problems (subproblems)!

Jalal Kazempour
Decomposable Structures

Optimization problems with complicating variable(s)

Example: A linear programming (LP) as the original problem

Minimize

\[
A_1 x_1 + A_2 x_2 + B_1 y_1 + B_2 y_2 + C_1 z_1 + C_2 z_2 + D_1 \beta^{\text{fixed}}
\]

Subject to

\[
\begin{align*}
E_{11} x_1 + E_{12} x_2 & \geq F_1 - E_{13} \beta^{\text{fixed}} \\
E_{21} x_1 + E_{22} x_2 & \geq F_2 - E_{23} \beta^{\text{fixed}} \\
E_{31} y_1 + E_{32} y_2 & \geq F_3 - E_{33} \beta^{\text{fixed}} \\
\end{align*}
\]

\[
\begin{align*}
E_{41} z_1 + E_{42} z_2 & \geq F_4 - E_{43} \beta^{\text{fixed}} \\
E_{51} z_1 + E_{52} z_2 & \geq F_5 - E_{53} \beta^{\text{fixed}}
\end{align*}
\]

\(\beta^{\text{fixed}}\) is a complicating variable, i.e., if it is fixed to a given value \(\beta^{\text{fixed}}\), then the original problem decomposes to 3 smaller problems (subproblems)!

Jalal Kazempour
Decomposable Structures

*Optimization problems with complicating variable(s)*

**Example**: A linear programming (LP) as the original problem

**Subproblem 1:**

Minimize \[ A_1 x_1 + A_2 x_2 \]

Subject to

**Constraints including only variables x**

\[
\begin{align*}
E_{11} x_1 + E_{12} x_2 & \geq F_1 - E_{13} \beta_{\text{fixed}} \\
E_{21} x_1 + E_{22} x_2 & \geq F_2 - E_{23} \beta_{\text{fixed}}
\end{align*}
\]

Jalal Kazempour
Example: A linear programming (LP) as the original problem

Subproblem 2:

Minimize $B_1 y_1 + B_2 y_2$

Subject to

Constraint including only variables $y$

$E_{31} y_1 + E_{32} y_2 \geq F_3 - E_{33} \beta^{\text{fixed}}$

Jalal Kazempour
Decomposable Structures

*Optimization problems with complicating variable(s)*

**Example:** A linear programming (LP) as the original problem

**Subproblem 3:**

Minimize \( C_1 z_1 + C_2 z_2 \)

Subject to

Constraints including only variables \( z \)

\[
E_{41} z_1 + E_{42} z_2 \geq F_4 - E_{43} \beta^{fixed}
\]

\[
E_{51} z_1 + E_{52} z_2 \geq F_5 - E_{53} \beta^{fixed}
\]

Jalal Kazempour
**Decomposable Structures**

*Optimization problems with complicating variable(s)*

**Example**: A linear programming (LP) as the original problem

**Decomposable matrix with complicating variable(s)**

<table>
<thead>
<tr>
<th>$A^T$</th>
<th>$B^T$</th>
<th>$C^T$</th>
<th>$D^T$</th>
</tr>
</thead>
<tbody>
<tr>
<td>Subject to</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>$E^{[1]}$</td>
<td>$E^{[4]}$</td>
<td></td>
<td></td>
</tr>
<tr>
<td>$E^{[2]}$</td>
<td>$E^{[5]}$</td>
<td></td>
<td></td>
</tr>
<tr>
<td>$E^{[3]}$</td>
<td>$E^{[6]}$</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

\[
\begin{array}{c}
x \\
y \\
z \\
\beta \\
\end{array} \quad \times \quad \begin{array}{c}
E^{[1]} \\
E^{[2]} \\
E^{[3]} \\
E^{[4]} \\
E^{[5]} \\
E^{[6]} \\
\end{array} = \begin{array}{c}
F^{[1]} \\
F^{[2]} \\
F^{[3]} \\
\end{array}
\]

Complicating variable

Jalal Kazempour
Exercise:

Seven examples are available in the papers on your table. Please check them in the next 10 minutes, and identify whether they are decomposable problems or not (if so, how?). Then, check your results with your neighbors around the table.
Example 1

Identify whether the following problem is decomposable or not (if so, how!)

Minimize \( -4x_1 - y_1 - 6z_1 \)

subject to

\( x_1 - x_2 = 1 \)
\( x_1 + x_3 = 3 \)
\( y_1 - y_2 = 1 \)
\( y_1 + y_3 = 2 \)
\( z_1 - z_2 = 1 \)
\( z_1 + z_3 = 2 \)
\( 3x_1 + 2y_1 + 4z_1 + w_1 = 17 \)
\( x_1, x_2, x_3, y_1, y_2, y_3, z_1, z_2, z_3, w_1 \geq 0 \)
Example 1

Identify whether the following problem is decomposable or not (if so, how!)

Minimize

\[ -4x_1 - y_1 - 6z_1 \]

subject to

\[ x_1 - x_2 = 1 \]
\[ x_1 + x_3 = 3 \]
\[ y_1 - y_2 = 1 \]
\[ y_1 + y_3 = 2 \]
\[ z_1 - z_2 = 1 \]
\[ z_1 + z_3 = 2 \]

\[ 3x_1 + 2y_1 + 4z_1 + w_1 = 17 \]

\[ x_1, x_2, x_3, y_1, y_2, y_3, z_1, z_2, z_3, w_1 \geq 0 \]

Complicating constraint: if relaxed, then the original problem decomposes to 3 subproblems
Example 2

Identify whether the following problem is decomposable or not (if so, how)!

Maximize \[4x_1 + 3x_2 + 2x_3 + 3x_4 + 2x_5 + 3x_6\]

subject to

\[x_1 - 2x_2 + 2x_6 \leq 3\]
\[2x_1 + x_2 + x_6 \leq 3\]
\[-2x_1 + 3x_2 + x_6 \leq 7\]
\[x_3 + 3x_6 \leq 4\]
\[2x_3 - x_6 \leq 3\]
\[x_4 \leq 1\]
\[2x_4 - 4x_5 + 3x_6 \leq 5\]
\[3x_4 + x_5 - x_6 \leq 4\]
Example 2

Identify whether the following problem is decomposable or not (if so, how!)

\[
\begin{align*}
\text{Maximize} & \quad 4x_1 + 3x_2 + 2x_3 + 3x_4 + 2x_5 + 3x_6 \\
\text{subject to} & \\
-x_1 - 2x_2 + 2x_6 & \leq 3 \\
2x_1 + x_2 + x_6 & \leq 3 \\
-2x_1 + 3x_2 + x_6 & \leq 7 \\
x_3 + 3x_6 & \leq 4 \\
2x_3 - x_6 & \leq 3 \\
x_4 & \leq 1 \\
2x_4 - 4x_5 + 3x_6 & \leq 5 \\
3x_4 + x_5 - x_6 & \leq 4 
\end{align*}
\]

\(x_6\) is a complicating variable:
if fixed to a given value, then the original problem decomposes to 3 subproblems
Example 3

Identify whether the following problem is decomposable or not (if so, how!)

Single-node single-hour optimal power flow (OPF) problem (total cost minimization) with 3 conventional generators and a single inelastic load:

Minimize \[ 10g_1 + 25g_2 + 30g_3 \]

subject to

\[ 0 \leq g_1 \leq 100 \]
\[ 0 \leq g_2 \leq 150 \]
\[ 0 \leq g_3 \leq 200 \]
\[ g_1 + g_2 + g_3 = 350 \]
Example 3

Identify whether the following problem is decomposable or not (if so, how!)

Single-node single-hour optimal power flow (OPF) problem (total cost minimization) with 3 conventional generators and a single inelastic load:

Minimize \[ 10g_1 + 25g_2 + 30g_3 \]
subject to
\[ 0 \leq g_1 \leq 100 \]
\[ 0 \leq g_2 \leq 150 \]
\[ 0 \leq g_3 \leq 200 \]
\[ g_1 + g_2 + g_3 = 350 \]

Complicating constraint: if relaxed, then the original problem decomposes to 3 subproblems, one per generator

Jalal Kazempour
Example 4

Identify whether the following problem is decomposable or not (if so, how)!

Single-node single-hour optimal power flow (OPF) problem (social welfare maximization) with 3 conventional generators and a single elastic load:

Maximize \(40d_1 - 10g_1 - 25g_2 - 30g_3\)

subject to

\[0 \leq g_1 \leq 100\]
\[0 \leq g_2 \leq 150\]
\[0 \leq g_3 \leq 200\]
\[0 \leq d_1 \leq 350\]
\[g_1 + g_2 + g_3 = d_1\]

Jalal Kazempour
Example 4

Identify whether the following problem is decomposable or not (if so, how!)

Single-node single-hour optimal power flow (OPF) problem (social welfare maximization) with 3 conventional generators and a single elastic load:

Maximize \( 40d_1 - 10g_1 - 25g_2 - 30g_3 \)

subject to

\[ 0 \leq g_1 \leq 100 \]
\[ 0 \leq g_2 \leq 150 \]
\[ 0 \leq g_3 \leq 200 \]
\[ 0 \leq d_1 \leq 350 \]

\[ g_1 + g_2 + g_3 = d_1 \]

Complicating constraint:
if relaxed, then the original problem decomposes to 4 subproblems, one per agent (generator and demand)
Example 5

Identify whether the following problem is decomposable or not (if so, how)?

Single-node multi-hour (Index: \( h \)) optimal power flow (OPF) problem (social welfare maximization) with 3 conventional generators and a single elastic load:

\[
\text{Maximize } \sum_h [40d_{h,1} - 10g_{h,1} - 25g_{h,2} - 30g_{h,3}]
\]

subject to

\[
\begin{align*}
0 & \leq g_{h,1} \leq 100 & \forall h \\
0 & \leq g_{h,2} \leq 150 & \forall h \\
0 & \leq g_{h,3} \leq 200 & \forall h \\
0 & \leq d_{h,1} \leq D_{h,1} & \forall h \\
g_{h,1} + g_{h,2} + g_{h,3} & = d_{h,1} & \forall h
\end{align*}
\]

Jalal Kazempour
Example 5

Identify whether the following problem is decomposable or not (if so, how!)

Single-node multi-hour (Index: $h$) optimal power flow (OPF) problem (social welfare maximization) with 3 conventional generators and a single elastic load:

Maximize $\sum_h \left[ 40d_{h,1} - 10g_{h,1} - 25g_{h,2} - 30g_{h,3} \right]$

subject to

$0 \leq g_{h,1} \leq 100 \quad \forall h$

$0 \leq g_{h,2} \leq 150 \quad \forall h$

$0 \leq g_{h,3} \leq 200 \quad \forall h$

$0 \leq d_{h,1} \leq D_{h,1} \quad \forall h$

$g_{h,1} + g_{h,2} + g_{h,3} = d_{h,1} \quad \forall h$

Complicating constraints:

if relaxed, then the original problem decomposes to a set of subproblems, one per agent per hour
Example 6

Identify whether the following problem is decomposable or not (if so, how)!

Single-node multi-hour (Index: \( h \)) optimal power flow (OPF) problem (social welfare maximization) with 3 conventional generators and a single elastic load (enforcing inter-temporal ramping constraints for one of generators):

Maximize \( \sum_{h}[40d_{h,1} - 10g_{h,1} - 25g_{h,2} - 30g_{h,3}] \)

subject to

\[
\begin{align*}
0 & \leq g_{h,1} \leq 100 & \forall h \\
0 & \leq g_{h,2} \leq 150 & \forall h \\
0 & \leq g_{h,3} \leq 200 & \forall h \\
0 & \leq d_{h,1} \leq D_{h,1} & \forall h \\
-25 & \leq [g_{h,1} - g_{(h-1),1}] \leq 25 & \forall h \\
g_{h,1} + g_{h,2} + g_{h,3} & = d_{h,1} & \forall h
\end{align*}
\]

Jalal Kazempour
Example 6

Identify whether the following problem is decomposable or not (if so, how):

Single-node multi-hour (Index: $h$) optimal power flow (OPF) problem (social welfare maximization) with 3 conventional generators and a single elastic load (enforcing inter-temporal ramping constraints for one of generators):

Maximize \( \sum_{h} [40d_{h,1} - 10g_{h,1} - 25g_{h,2} - 30g_{h,3}] \)

subject to

\[
\begin{align*}
0 & \leq g_{h,1} \leq 100 \quad \forall h \\
0 & \leq g_{h,2} \leq 150 \quad \forall h \\
0 & \leq g_{h,3} \leq 200 \quad \forall h \\
0 & \leq d_{h,1} \leq D_{h,1} \quad \forall h
\end{align*}
\]

Complicating constraints:
if relaxed, then the original problem decomposes to a set of subproblems, one per agent per hour

\[-25 \leq [g_{h,1} - g_{(h-1),1}] \leq 25 \quad \forall h\]

\[g_{h,1} + g_{h,2} + g_{h,3} = d_{h,1} \quad \forall h\]

Jalal Kazempour
Example 7

Identify whether the following problem is decomposable or not (if so, how!)

Single-node single-year (static) generation expansion problem (GEP), considering two existing generators (units 1 and 2) and one candidate generator (unit 3)

Minimize \[ 15000x_3 + \sum_h \left[ 10g_{h,1} + 25g_{h,2} + 30g_{h,3} \right] \]

subject to

\[ 0 \leq g_{h,1} \leq 100 \quad \forall h \]
\[ 0 \leq g_{h,2} \leq 150 \quad \forall h \]
\[ 0 \leq g_{h,3} \leq x_3 \quad \forall h \]
\[ g_{h,1} + g_{h,2} + g_{h,3} = D_{h,1} \quad \forall h \]
\[ x_3 \geq 0 \]
Example 7

Identify whether the following problem is decomposable or not (if so, how!)

Single-node single-year (static) generation expansion problem (GEP), considering two existing generators (units 1 and 2) and one candidate generator (unit 3)

Minimize \( 15000x_3 + \sum_{h} [10g_{h,1} + 25g_{h,2} + 30g_{h,3}] \)

subject to

\( 0 \leq g_{h,1} \leq 100 \quad \forall h \)

\( 0 \leq g_{h,2} \leq 150 \quad \forall h \)

\( 0 \leq g_{h,3} \leq x_3 \quad \forall h \)

\( g_{h,1} + g_{h,2} + g_{h,3} = D_{h,1} \quad \forall h \)

\( x_3 \geq 0 \)

\( x_3 \) is a complicating variable: if fixed to a given value, then the original problem decomposes to a set of subproblems, one per hour

Jalal Kazempour
Thanks for your attention!

Email: seykaz@elektro.dtu.dk
Decomposition techniques for optimization problems with complicating constraints

Jalal Kazempour
Technical University of Denmark (DTU)
Learning Objectives

After this session the participants are expected to be able to:

- Explain the functioning of Lagrangian relaxation (LR), augmented Lagrangian relaxation (ALR), and alternating direction method of multipliers (ADMM)
Decomposition Techniques

Applicable to optimization problems with complicating constraints

• Lagrangian relaxation (LR)
  In the literature, this technique has been also known as standard or conventional LR!

• Augmented Lagrangian relaxation (ALR)
  • Auxiliary problem principle (APP)
  • Alternating direction method of multipliers (ADMM)

• Dantzig-Wolfe decomposition (DWD)
• ...

Jalal Kazempour
Decomposition Techniques

Applicable to optimization problems with complicating constraints

• Lagrangian relaxation (LR)
  In the literature, this technique has been also known as standard or conventional LR!

• Augmented Lagrangian relaxation (ALR)
  ❏ Auxiliary problem principle (APP)
  ❏ Alternating direction method of multipliers (ADMM)

• Dantzig-Wolfe decomposition (DWD)

• ...

Jalal Kazempour

Will not be covered in this course!
Optimization problems with complicating constraints

Some examples related to power systems

• Single-node optimal power flow (OPF) problem (investigated in the previous lecture)
  ✓ Complicating constraints: balance equalities and ramping limits of generators
  ✓ If relaxed, original problem decomposes by agent (and hour)
Optimization problems with complicating constraints

Some examples related to power systems

- Single-node optimal power flow (OPF) problem (investigated in the previous lecture)
  - Complicating constraints: balance equalities and ramping limits of generators
  - If relaxed, original problem decomposes by agent (and hour)

- Multi-regional OPF (or unit commitment) problem, e.g., in case of pan-European electricity market
  - Complicating constraints: tie-line constraints (power flow, and tie-line capacity)
  - If relaxed, original problem decomposes by region. This way, operator of each region only solves its own OPF problem (so-called distributed OPF problem)

Jalal Kazempour
Lagrangian Relaxation (LR)

Background

• The theory of LR (and also ALR) was firstly developed for problems with \textit{continuous} variables, and functions (objective function and constraints) with \textit{first derivatives continuous}.

• However, the theory has been used in problems with \textit{binary} variables (like unit commitment problems) with success.

Jalal Kazempour
Lagrangian Relaxation (LR)

Background

• LR works efficiently if the number of complicating constraints is relatively low, and it is OK to have binary variables in the formulation.

• LR was extensively used in the 90’s to solve unit commitment problems (complicating constrains are just balance constraints and ramping constraints).
Lagrangian Relaxation (LR)

Background

Key point

In case of LR:

In addition to convexity, the objective function of the original problem (not decomposed problem) needs to be smooth (continuous first derivatives), e.g., quadratic. **If this objective function is linear, the LR procedure does not converge!**

- Alternative solution technique for problems with linear objective function is ALR.

Jalal Kazempour
Lagrangian Relaxation (LR)

Background

For unit commitment (and also OPF) problems:

- LR (for problems with quadratic objective function)
- ALR (for problems with either quadratic or linear objective function)

Both have been extensively and very successfully used in the literature, though unit commitment problem is fully non-convex (due to binary variables).

Jalal Kazempour
Lagrangian Relaxation (LR)

Mathematical procedure

Original (non-decomposed) problem:

Minimize \( \sum_{i=1}^{I} f_i(x_i) \)

Subject to

\( g_i(x_i) = A_i \quad \forall i \)
\( h_i(x_i) \leq B_i \quad \forall i \)
\( \sum_{i=1}^{I} c_i(x_i) = M \)
\( \sum_{i=1}^{I} d_i(x_i) \leq N \)

Jalal Kazempour
Lagrangian Relaxation (LR)

Mathematical procedure

Original (non-decomposed) problem:

Minimize
\[ \sum_{i=1}^{l} f_i(x_i) \]

Subject to
\[ g_i(x_i) = A_i \quad \forall i \]
\[ h_i(x_i) \leq B_i \quad \forall i \]
\[ \sum_{i=1}^{l} c_i(x_i) = M \]
\[ \sum_{i=1}^{l} d_i(x_i) \leq N \]
Lagrangian Relaxation (LR)

Mathematical procedure

Original (non-decomposed) problem:

Minimize \( \sum_{i=1}^{l} f_i(x_i) \)

Subject to

\( g_i(x_i) = A_i \quad \forall i \)
\( h_i(x_i) \leq B_i \quad \forall i \)

\[ \left\{ \begin{array}{l}
\sum_{i=1}^{l} c_i(x_i) = M \\
\sum_{i=1}^{l} d_i(x_i) \leq N
\end{array} \right\} \]

Dual variables
(Lagrangian multipliers)

Complicating constraints

Jalal Kazempour
In the optimal point, the original problem is equivalent to:

Minimize \[ \sum_{i=1}^{l} f_i(x_i) + \lambda \left[ M - \sum_{i=1}^{l} c_i(x_i) \right] + \mu \left[ N - \sum_{i=1}^{l} d_i(x_i) \right] \]

Subject to

\[ g_i(x_i) = A_i \quad \forall i \]
\[ h_i(x_i) \leq B_i \quad \forall i \]
In the optimal point, the original problem is equivalent to:

\[
\begin{align*}
\text{Minimize} & \quad \sum_{i=1}^{I} f_i(x_i) + \lambda \left[ M - \sum_{i=1}^{I} c_i(x_i) \right] + \mu \left[ N - \sum_{i=1}^{I} d_i(x_i) \right] \\
\text{Subject to} & \quad g_i(x_i) = A_i \quad \forall i \\
& \quad h_i(x_i) \leq B_i \quad \forall i 
\end{align*}
\]

Is the original problem decomposable now?
In the optimal point, the original problem is equivalent to:

Minimize $\sum_{i=1}^{I} f_i(x_i) + \lambda \left[ M - \sum_{i=1}^{I} c_i(x_i) \right] + \mu \left[ N - \sum_{i=1}^{I} d_i(x_i) \right]$ 

Subject to 

$g_i(x_i) = A_i \quad \forall i$ 
$h_i(x_i) \leq B_i \quad \forall i$

Is the original problem decomposable now? **Not yet!**
Lagrangian Relaxation (LR)
Mathematical procedure

In the optimal point, the original problem is equivalent to:

\[
\text{Minimize } \sum_{i=1}^{I} f_i(x_i) + \lambda \left[ M - \sum_{i=1}^{I} c_i(x_i) \right] + \mu \left[ N - \sum_{i=1}^{I} d_i(x_i) \right]
\]

Subject to

\[
\begin{align*}
 g_i(x_i) &= A_i & \forall i \\
 h_i(x_i) &\leq B_i & \forall i
\end{align*}
\]

Is the original problem decomposable now? Not yet!

• Let's relax the equivalent problem above by fixing dual variables \((\lambda \text{ and } \mu)\) to given values, i.e., \(\bar{\lambda}\) and \(\bar{\mu}\).
Lagrangian Relaxation (LR)

Mathematical procedure

In the optimal point, the original problem is equivalent to:

\[
\begin{align*}
\text{Minimize} & \quad \sum_{i=1}^{l} f_i(x_i) + \bar{\lambda} \left[ M - \sum_{i=1}^{l} c_i(x_i) \right] + \bar{\mu} \left[ N - \sum_{i=1}^{l} d_i(x_i) \right] \\
\text{Subject to} & \\
g_i(x_i) & = A_i \quad \forall i \\
h_i(x_i) & \leq B_i \quad \forall i
\end{align*}
\]

Is the original problem decomposable now?

Jalal Kazempour
Lagrangian Relaxation (LR)

Mathematical procedure

In the optimal point, the original problem is equivalent to:

Minimize \[ \sum_{i=1}^{l} f_i(x_i) + \lambda \left[ M - \sum_{i=1}^{l} c_i(x_i) \right] + \mu \left[ N - \sum_{i=1}^{l} d_i(x_i) \right] \]

Subject to

\[ g_i(x_i) = A_i \quad \forall i \]
\[ h_i(x_i) \leq B_i \quad \forall i \]

Is the original problem decomposable now? Yes, one per \( i \):

\[ \begin{cases} \text{Minimize} & f_i(x_i) + \lambda c_i(x_i) + \mu d_i(x_i) \\ \text{Subject to} & g_i(x_i) = A_i \\ & h_i(x_i) \leq B_i \end{cases} \quad \forall i \]
Lagrangian Relaxation (LR)

Mathematical procedure

LR is an iterative approach with a systematic way to update the values of fixed dual variables ($\bar{\lambda}$ and $\bar{\mu}$) in each iteration.

Available techniques in the literature to update $\bar{\lambda}$ and $\bar{\mu}$:

1. Subgradient method
2. Cutting plane method
3. Bundle method
4. Trust region method
5. ...

Jalal Kazempour
Lagrangian Relaxation (LR)

Mathematical procedure

LR is an iterative approach with a systematic way to update the values of fixed dual variables (\( \bar{\lambda} \) and \( \bar{\mu} \)) in each iteration.

Available techniques in the literature to update \( \bar{\lambda} \) and \( \bar{\mu} \):

1. Subgradient method
2. Cutting plane method
3. Bundle method
4. Trust region method
5. ...

Will not be covered in this course!

Jalal Kazempour
Minimize \( x^2 + y^2 \quad x \geq 0, y \geq 0 \)

Subject to \(-x - y \leq -4 \quad (\mu)\)

Note: Objective function includes quadratic terms, so LR works!
Lagrangian Relaxation (LR)

Numerical example

Minimize \( x^2 + y^2 \)
subject to \(-x - y \leq -4\) \((\mu)\)

Note: Objective function includes quadratic terms, so LR works!

Subproblem 1:
Minimize \( x^2 - \bar{\mu}x \)
subject to \(x \geq 0\)

Subproblem 2:
Minimize \( y^2 - \bar{\mu}y \)
subject to \(y \geq 0\)

Jalal Kazempour
Lagrangian Relaxation (LR)

Numerical example

Minimize $x^2 + y^2$

subject to $-x - y \leq -4$ \hspace{1cm} (\mu)

Note: Objective function includes quadratic terms, so LR works!

Subproblem 1:
Minimize $x^2 - \bar{\mu}x$

Subproblem 2:
Minimize $y^2 - \bar{\mu}y$

Updating fixed dual variable ($\bar{\mu}$) using subgradient method:

- Solve subproblems 1 and 2 in iteration \( v \), and obtain the values \( x^{(v)} \) and \( y^{(v)} \)
- \( \bar{\mu}^{(v+1)} \leftarrow \bar{\mu}^{(v)} + \frac{1}{a+by} \frac{-x^{(v)} - y^{(v)} + 4}{|x^{(v)} - y^{(v)} + 4|} \)
- \( a \) and \( b \) are positive constants, e.g., \( a = 1 \) and \( b = 0.1 \).

Jalal Kazempour
Algorithm:

• **Step 0: Initialization**
  Set \( v = 1 \) and \( \bar{\mu}^{(1)} = \bar{\mu}^{\text{initial}} \)

• **Step 1: Solve subproblems 1 and 2, and obtain** \( x^{(v)} \) **and** \( y^{(v)} \)

• **Step 2: Update fixed dual variable, i.e.,** \( \bar{\mu}^{(v+1)} \)

• **Step 3: Convergence check**
  If \( \frac{\|\bar{\mu}^{(v+1)} - \bar{\mu}^{(v-1)}\|}{\|\bar{\mu}^{(v)}\|} \leq \epsilon \), then the optimal solution with a level of accuracy \( \epsilon \) is obtained, otherwise \( v \leftarrow v + 1 \) and go Step 1
Lagrangian Relaxation (LR)

Numerical example

Exercise:

The printed version of GAMS code of the LR example is available on your table. Please check it in the next 15 minutes, then explain it to your neighbors around the table.

My colleagues and I will answer your questions!

This code has been prepared by Lejla Halilbasic and Christos Ordoudis.

Jalal Kazempour
Augmented Lagrangian Relaxation (ALR)

Recall:
ALR works for problems with either quadratic objective function (like LR) or linear one (unlike LR)

Main difference of ALR with respect to LR:
An additional penalty term within the subproblems
Augmented Lagrangian Relaxation (ALR)

Numerical Example

Minimize $x^2 + y^2$

subject to $-x - y = -4$ ($\lambda$)
Augmented Lagrangian Relaxation (ALR)

Numerical Example

Minimize \( x^2 + y^2 \)
\[ x \geq 0, y \geq 0 \]

Subject to \( -x - y = -4 \) \((\lambda)\)

Equivalent to:

Minimize \( x^2 + y^2 + \lambda(-x - y + 4) + \frac{y}{2} \| -x - y + 4 \|^2 \)

Additional penalty term with respect to LR, \( \gamma \) is a positive constant.

Jalal Kazempour
Augmented Lagrangian Relaxation (ALR)

Numerical Example

Minimize \( x^2 + y^2 \)
\[ x \geq 0, y \geq 0 \]

Subject to \( -x - y = -4 \) (\( \lambda \))

Additional penalty term with respect to LR, \( \gamma \) is a positive constant

Equivalent to:

Minimize \( x^2 + y^2 + \lambda(-x - y + 4) + \frac{\gamma}{2} \| -x - y + 4 \|^2 \)
\[ x \geq 0, y \geq 0 \]

Question:

• Similar to LR, assume dual variable \( \lambda \) is fixed to given value \( \bar{\lambda} \). Is the problem above decomposable for given \( \bar{\lambda} \)?

Jalal Kazempour
Augmented Lagrangian Relaxation (ALR)

Numerical Example

Minimize \( x^2 + y^2 \)
\[ \text{subject to } -x - y = -4 \ (\lambda) \]

Equivalent to:

Minimize \( x^2 + y^2 + \lambda(-x - y + 4) + \frac{\gamma}{2} \| -x - y + 4 \|^2 \)

Question:

- Similar to LR, assume dual variable \( \lambda \) is fixed to given value \( \lambda \). Is the problem above decomposable for given \( \lambda \)? No, due to product of \( x \) and \( y \) in the penalty term!

Jalal Kazempour
Augmented Lagrangian Relaxation (ALR)

Numerical Example

Available alternatives to solve ALR:
- Auxiliary problem principle (APP)
- Alternating direction method of multipliers (ADMM)
Available alternatives to solve ALR:

- Auxiliary problem principle (APP): *will not be covered in this course*
- **Alternating direction method of multipliers (ADMM)**

Jalal Kazempour
Augmented Lagrangian Relaxation (ALR)

Numerical Example

Available alternatives to solve ALR:
- Auxiliary problem principle (APP): *will not be covered in this course*
- **Alternating direction method of multipliers (ADMM)**

Note:
ADMM directly fixes variables to their values in the previous iteration, and decomposes the ALR to subproblems.
Augmented Lagrangian Relaxation (ALR)

Numerical Example

Available alternatives to solve ALR:
• Auxiliary problem principle (APP): *will not be covered in this course*
• **Alternating direction method of multipliers (ADMM)**

**Note:**
ADMM directly fixes variables to their values in the previous iteration, and decomposes the ALR to subproblems.

\[
\begin{align*}
\text{Minimize} & \quad x^2 + y^2 + \lambda(-x - y + 4) + \frac{\gamma}{2} \| -x - y + 4 \|^2 \\
\text{subject to} & \quad x, y \geq 0
\end{align*}
\]

The problem above in iteration \(v\) can be decomposed to two subproblems:

\[
\begin{align*}
\text{Minimize} & \quad x^2(v) + \lambda^{(v-1)}(-x^{(v)} + 2) + \frac{\gamma}{2} \| -x^{(v)} - y^{(v-1)} + 4 \|^2 \\
& \text{subject to} \quad x^{(v)} \geq 0 \\
\text{Minimize} & \quad y^2(v) + \lambda^{(v-1)}(-y^{(v)} + 2) + \frac{\gamma}{2} \| -y^{(v)} - x^{(v-1)} + 4 \|^2 \\
& \text{subject to} \quad y^{(v)} \geq 0
\end{align*}
\]

where \(\lambda^{(v)} \leftarrow \lambda^{(v-1)} + \gamma(-x^{(v)} - y^{(v)} + 4)\)
Augmented Lagrangian Relaxation (ALR)

Numerical Example

Exercise:

The printed version of GAMS code of the ALR/ADMM example is available on your table. Please check it in the next 15 minutes, then explain it to your neighbors around the table.

My colleagues and I will answer your questions!

This code has been prepared by Lejla Halilbasic and Christos Ordoudis.

Jalal Kazempour
References


Additional References


Thanks for your attention!

Email: seykaz@elektro.dtu.dk
Decomposition techniques for optimization problems with complicating variables

Jalal Kazempour
Technical University of Denmark (DTU)
Learning Objectives

After this session the participants are expected to be able to:

• Explain the functioning of Benders’ decomposition
• Explain the application of Benders’ decomposition to multi-stage stochastic problems
Multi-stage Problems

**Deterministic** multi-stage problem (e.g., deterministic unit commitment problem)

**Stochastic** multi-stage problem (e.g., stochastic unit commitment problem)

Jalal Kazempour
Two-stage Deterministic Problem

\[
\begin{align*}
\min_{x_1, x_2} \quad & c_1 x_1 + c_2 x_2 \\
\text{subject to} \quad & A_1 x_1 \geq b_1 \\
& E_1 x_1 + A_2 x_2 \geq b_2
\end{align*}
\]
Two-stage Deterministic Problem

\[ \begin{align*}
\text{min} & \quad c_1 x_1 + c_2 x_2 \\
\text{subject to} & \quad A_1 x_1 \geq b_1 \\
& \quad E_1 x_1 + A_2 x_2 \geq b_2
\end{align*} \]
Two-stage Deterministic Problem

First-stage problem:

$$\min_{x_1, x_2} c_1 x_1 + c_2 x_2$$

subject to

$$A_1 x_1 \geq b_1$$

$$E_1 x_1 + A_2 x_2 \geq b_2$$

Second-stage problem:

$$\alpha_1(x_1) = \min_{x_2} c_2 x_2$$

subject to

$$A_2 x_2 \geq b_2 - E_1 x_1$$

$$\alpha_1(x_1) : \text{the second-stage cost as a function of the first-stage decisions } x_1$$

(future cost function)

Jalal Kazempour
Two-stage Deterministic Problem

First-stage problem:

\[
\min_{x_1} c_1 x_1 + \alpha_1(x_1)
\]
subject to
\[
A_1 x_1 \geq b_1
\]

Second-stage problem:

\[
\alpha_1(x_1) = \min_{x_2} c_2 x_2
\]
subject to
\[
A_2 x_2 \geq b_2 - E_1 x_1
\]

Note: \( x_1 \) appears in the second-stage problem!

\( \alpha_1(x_1) \): the second-stage cost as a function of the first-stage decisions \( x_1 \) (future cost function)

Jalal Kazempour
Two-stage Deterministic Problem

One potential solution approach

First-stage problem:

\[ \min_{x_1} c_1 x_1 + \alpha_1(x_1) \]

subject to

\[ A_1 x_1 \geq b_1 \]

Second-stage problem:

\[ \alpha_1(x_1) = \min_{x_2} c_2 x_2 \]

subject to

\[ A_2 x_2 \geq b_2 - E_1 x_1 \]
Two-stage Deterministic Problem

One potential solution approach

- **Step 1)** Discretize $x_1$ into a set of trial values $\{\hat{x}_{1i}, i = 1, \ldots, n\}$
- **Step 2)** Solve the second-stage problem for each of the trial values
- **Step 3)** Construct future cost function $\alpha_1(x_1)$. Intermediate values of $\alpha_1(x_1)$ are obtained by interpolation from the neighboring discretized values.
- **Step 4)** Solve the first-stage problem using the future cost function constructed.

**First-stage problem:**

$$\min_{x_1} c_1 x_1 + \alpha_1(x_1)$$

subject to

$$A_1 x_1 \geq b_1$$

**Second-stage problem:**

$$\alpha_1(x_1) = \min_{x_2} c_2 x_2$$

subject to

$$A_2 x_2 \geq b_2 - E_1 x_1$$
Two-stage Deterministic Problem

One potential solution approach

• **Step 1)** Discretize \( x_1 \) into a set of trial values \( \{\hat{x}_{1i}, i = 1, ..., n\} \)
• **Step 2)** Solve the second-stage problem for each of the trial values
• **Step 3)** Construct future cost function \( \alpha_1(x_1) \). Intermediate values of \( \alpha_1(x_1) \) are obtained by interpolation from the neighboring discretized values.
• **Step 4)** Solve the first-stage problem using the future cost function constructed.

The future cost function \( \alpha_1(x_1) \) constructed

Jalal Kazempour
Two-stage Deterministic Problem

One potential solution approach

- **Step 1)** Discretize $x_1$ into a set of trial values \( \{\hat{x}_{1i}, i = 1, \ldots, n\} \)
- **Step 2)** Solve the second-stage problem for each of the trial values
- **Step 3)** Construct future cost function \( \alpha_1(x_1) \). Intermediate values of \( \alpha_1(x_1) \) are obtained by interpolation from the neighboring discretized values.
- **Step 4)** Solve the first-stage problem using the future cost function constructed.

What is the name of this technique?
Two-stage Deterministic Problem

One potential solution approach

- **Step 1)** Discretize \( x_1 \) into a set of trial values \( \{\hat{x}_{1i}, i = 1, ..., n\} \)
- **Step 2)** Solve the second-stage problem for each of the trial values
- **Step 3)** Construct future cost function \( \alpha_1(x_1) \). Intermediate values of \( \alpha_1(x_1) \) are obtained by interpolation from the neighboring discretized values.
- **Step 4)** Solve the first-stage problem using the future cost function constructed.

The future cost function \( \alpha_1(x_1) \) constructed

What is the name of this technique?

**Dynamic programming (DP)**
Two-stage Deterministic Problem

One potential solution approach

The future cost function $\alpha_1(x_1)$ constructed

What is the main drawback of dynamic programming (DP)?
Two-stage Deterministic Problem

One potential solution approach

What is the main drawback of dynamic programming (DP)?

DP needs to discretize the decision variables $x_1$, which results in computational issues!

For example, 10 decision variables and 4 discretized value for each variable leads to $4^{10}$ discrete values!
Two-stage Deterministic Problem

Alternative solution approach
Two-stage Deterministic Problem

Alternative solution approach

**Dual** dynamic programming (DDP) instead of DP!

**Advantage:**
To approximate the future cost function $\alpha_1(x_1)$ by analytical functions rather than a set of discrete values!
Two-stage Deterministic Problem

DDP functioning procedure

$$\alpha_1(x_1) = \min_{x_2} c_2 x_2$$

subject to

$$A_2 x_2 \geq b_2 - E_1 x_1 : \pi$$
Two-stage Deterministic Problem

**DDP functioning procedure**

\[
\alpha_1(x_1) = \min_{x_2} c_2 x_2 \\
\text{subject to} \\
A_2 x_2 \geq b_2 - E_1 x_1 : \pi
\]

Dual of the second-stage problem:

\[
\max_{\pi} \pi (b_2 - E_1 x_1) \\
\text{subject to} \\
\pi A_2 \geq c_2
\]
Two-stage Deterministic Problem

**DDP functioning procedure**

\[
\alpha_1(x_1) = \min_{x_2} \ c_2 x_2
\]

subject to

\[
A_2 x_2 \geq b_2 - E_1 x_1 : \pi
\]

Dual of the second-stage problem:

\[
\max_{\pi} \ \pi(b_2 - E_1 x_1)
\]

subject to

\[
\pi A_2 \geq c_2
\]

In the optimal solution

(\text{strong duality theorem})

\[
\alpha_1(x_1) = \pi^*(b_2 - E_1 x_1)
\]
Two-stage Deterministic Problem

**DDP functioning procedure**

\[
\alpha_1(x_1) = \min_{x_2} c_2 x_2
\]

subject to

\[
A_2 x_2 \geq b_2 - E_1 x_1 : \pi
\]

**Interpretation:** there is a linear relation between \(x_1\) and the future cost function \(\alpha_1(x_1)\) if the sensitivity \(\pi^*\) is known!

Dual of the second-stage problem:

\[
\max_{\pi} \pi(b_2 - E_1 x_1)
\]

subject to

\[
\pi A_2 \geq c_2
\]

In the optimal solution (strong duality theorem)

\[
\alpha_1(x_1) = \pi^*(b_2 - E_1 x_1)
\]
Two-stage Deterministic Problem

DDP functioning procedure

\[
\alpha_1(x_1) = \pi^*(b_2 - E_1 x_1)
\]
Two-stage Deterministic Problem

DDP functioning procedure

\[ \alpha_1(x_1) = \pi^*(b_2 - E_1x_1) \]

Assume \( \pi^1, \pi^2, ..., \pi^n \) are possible values for \( \pi^* \). Then, \( \alpha_1(x_1) \) can be characterized as follows:

\[
\begin{align*}
\alpha_1(x_1) &= \min_{\alpha, x_1} \alpha \\
\text{subject to} \\
\alpha &\geq \pi^1(b_2 - E_1x_1) \\
\alpha &\geq \pi^2(b_2 - E_1x_1) \\
&\quad \ldots \\
\alpha &\geq \pi^n(b_2 - E_1x_1)
\end{align*}
\]
Two-stage Deterministic Problem

**DDP functioning procedure**

\[ \alpha_1(x_1) = \pi^*(b_2 - E_1x_1) \]

Assume \( \pi^1, \pi^2, \ldots, \pi^n \) are possible values for \( \pi^* \). Then, \( \alpha_1(x_1) \) can be characterized as follows:

\[ \alpha_1(x_1) = \min_{\alpha, x_1} \alpha \]

subject to

\[ \alpha \geq \pi^1(b_2 - E_1x_1) \]
\[ \alpha \geq \pi^2(b_2 - E_1x_1) \]
\[ \quad \ldots \]
\[ \alpha \geq \pi^n(b_2 - E_1x_1) \]

Let’s interpret this optimization problem!
Two-stage Deterministic Problem

**DDP functioning procedure**

\[
\alpha_1(x_1) = \pi^*(b_2 - E_1 x_1)
\]

Assume \(\pi^1, \pi^2, \ldots, \pi^n\) are possible values for \(\pi^*\). Then, \(\alpha_1(x_1)\) can be characterized as follows:

\[
\alpha_1(x_1) = \min_{\alpha, x_1} \alpha
\]

subject to

\[
\alpha \geq \pi^1(b_2 - E_1 x_1)
\]

\[
\alpha \geq \pi^2(b_2 - E_1 x_1)
\]

\[
\vdots
\]

\[
\alpha \geq \pi^n(b_2 - E_1 x_1)
\]

**Note:** this means that we can construct a piecewise linear function for \(\alpha_1(x_1)\) problem (analytically but approximately) without need to discretize \(x_1\)!
Two-stage Deterministic Problem

**DDP functioning procedure**

\[ \alpha_1(x_1) = \pi^*(b_2 - E_1 x_1) \]

Assume \( \pi^1, \pi^2, ..., \pi^n \) are possible values for \( \pi^* \). Then, \( \alpha_1(x_1) \) can be characterized as follows:

\[
\begin{align*}
\alpha_1(x_1) &= \min_{\alpha, x_1} \alpha \\
&\text{subject to} \\
\alpha &\geq \pi^1(b_2 - E_1 x_1) \\
\alpha &\geq \pi^2(b_2 - E_1 x_1) \\
&\quad \vdots \\
\alpha &\geq \pi^n(b_2 - E_1 x_1)
\end{align*}
\]

Recall the first-stage problem:

\[
\begin{align*}
\min_{x_1} & \quad c_1 x_1 + \alpha_1(x_1) \\
&\quad \text{subject to} \\
&\quad A_1 x_1 \geq b_1
\end{align*}
\]
Two-stage Deterministic Problem

**DDP functioning procedure**

\[ \alpha_1(x_1) = \pi^*(b_2 - E_1x_1) \]

Assume \( \pi^1, \pi^2, \ldots, \pi^n \) are possible values for \( \pi^* \). Then, \( \alpha_1(x_1) \) can be characterized as follows:

\[ \begin{align*}
\alpha_1(x_1) &= \min_{\alpha, x_1} \alpha \\
&\text{subject to} \\
&\alpha \geq \pi^1(b_2 - E_1x_1) \\
&\alpha \geq \pi^2(b_2 - E_1x_1) \\
&\alpha \geq \pi^1(b_2 - E_1x_1) \\
&\alpha \geq \pi^n(b_2 - E_1x_1)
\end{align*} \]

Recall the first-stage problem:

\[ \begin{align*}
&\min_{x_1} c_1x_1 + \alpha_1(x_1) \\
&\text{subject to} \\
&A_1x_1 \geq b_1
\end{align*} \]

Let’s merge them!
Two-stage Deterministic Problem

DDP functioning procedure

Updated first-stage problem including the piecewise linear function $\alpha_1(x_1)$

$$\begin{align*}
\min_{\alpha, x_1} & \quad c_1 x_1 + \alpha \\
\text{subject to} & \quad A_1 x_1 \geq b_1 \\
& \quad \alpha \geq \pi^1(b_2 - E_1 x_1) \\
& \quad \alpha \geq \pi^2(b_2 - E_1 x_1) \\
& \quad \vdots \\
& \quad \vdots \\
& \quad \alpha \geq \pi^n(b_2 - E_1 x_1)
\end{align*}$$
Two-stage Deterministic Problem

DDP functioning procedure

How to generate possible values for $\pi^*$, i.e., $\pi^1, \pi^2, ..., \pi^n$?
How to generate possible values for $\pi^*$, i.e., $\pi^1, \pi^2, ..., \pi^n$?

**Option 1:**

1. Generate trial values for $x_1$
   
   $(\hat{x}_{1i}, i = 1, ..., n)$

2. Solve the second-stage problem for each trial, and obtain $\pi^1, \pi^2, ..., \pi^n$

3. Solve the first-stage problem
Two-stage Deterministic Problem

DDP functioning procedure

How to generate possible values for $\pi^*$, i.e., $\pi^1, \pi^2, \ldots, \pi^n$?

Option 1:

1. Generate trial values for $x_1$
   
   $(\hat{x}_{1i}, i = 1, \ldots, n)$

2. Solve the second-stage problem for each trial, and obtain $\pi^1, \pi^2, \ldots, \pi^n$

3. Solve the first-stage problem

Do you recommend this option?
Two-stage Deterministic Problem

DDP functioning procedure

How to generate possible values for $\pi^*$, i.e., $\pi^1, \pi^2, ..., \pi^n$?

Option 2 (systematic iterative approach):
Two-stage Deterministic Problem

**DDP functioning procedure**

How to generate possible values for $\pi^*$, i.e., $\pi^1, \pi^2, \ldots, \pi^n$?

**Option 2 (systematic iterative approach):**

1. **Iteration $i = 1$**
   - Generate a trial value for $x_1$

2. Solve the second-stage problem for given $x_1$ and obtain $x_2$ and $\pi_i$:
   
   $$\min_{x_2} c_2 x_2$$
   subject to
   $$A_2 x_2 \geq b_2 - E_1 x_1 : \pi_i$$

3. Solve the first-stage problem for given $\pi_i$ and update $x_1$:
   $$\min_{\alpha, x_1} c_1 x_1 + \alpha$$
   subject to
   $$A_1 x_1 \geq b_1$$
   $$\alpha \geq \pi_k (b_2 - E_1 x_1) \quad \forall k = 1, \ldots, i$$

4. **Check** $\alpha = c_2 x_2$
   - **No** $i = i + 1$
   - **Yes** Optimal results

**Update $x_1$**

Jalal Kazempour
Two-stage Deterministic Problem

DDP functioning procedure

How to generate possible values for $\pi^*$, i.e., $\pi^1, \pi^2, \ldots, \pi^n$?

Option 2 (systematic iterative approach):

Master problem

- **Iteration $i = 1$**
  - Generate a trial value for $x_1$
  - **Subproblem**
    - Solve the second-stage problem for given $x_1$ and obtain $x_2$ and $\pi_i$:
      $$\min_{x_2} c_2 x_2$$
      subject to
      $$A_2 x_2 \geq b_2 - E_1 x_1 : \pi_i$$
  - Updated $x_1$

- **Solve the first-stage problem for given $\pi_i$ and update $x_1$**:
  $$\min_{\alpha, x_1} c_1 x_1 + \alpha$$
  subject to
  $$A_1 x_1 \geq b_1$$
  $$\alpha \geq \pi_k (b_2 - E_1 x_1) \quad \forall k = 1, \ldots, i$$

- **Check** $\alpha = c_2 x_2$
  - **No**
    - $i = i + 1$
    - **Yes**
      - Optimal results

Jalal Kazempour
How to generate possible values for $\pi^*$, i.e., $\pi^1, \pi^2, ..., \pi^n$?

Option 2 (systematic iterative approach):

This approach is indeed **Benders’ decomposition**!
Important note:

We can guarantee obtaining the global optimal solution by Benders’ decomposition, if the objective function of the original (non-decomposed) problem is convex with respect to the complicating variable!
Two-stage Deterministic Problem

Simple Example

\[
\text{minimize} \quad z = -y - x/4
\]
\[
\begin{align*}
    y \quad -x & \leq 5 \\
y \quad -\frac{1}{2}x & \leq \frac{15}{2} \\
y \quad +\frac{1}{2}x & \leq \frac{35}{2} \\
-y \quad +x & \leq 10 \\
    0 \quad \leq x & \leq 16 \\
y \quad \geq 0.
\end{align*}
\]
Two-stage Deterministic Problem

Simple Example

Let’s consider $y$ as the complicating variable!

minimize $z = -y - x/4$

$x, y$

$y - x \leq 5$

$y - \frac{1}{2}x \leq \frac{15}{2}$

$y + \frac{1}{2}x \leq \frac{35}{2}$

$-y + x \leq 10$

$0 \leq x \leq 16$

$y \geq 0$. 

Jalal Kazempour
Two-stage Deterministic Problem

Simple Example

\begin{align*}
\text{minimize} & \quad -y^{(i)} \\
\text{subject to} & \quad y^{(i)} \leq 5 + x^{\text{fixed}}(i) : \pi^{(i)} \\
& \quad y^{(i)} \leq \frac{15}{2} + \frac{x^{\text{fixed}}(i)}{2} : \mu^{(i)} \\
& \quad y^{(i)} \leq \frac{35}{2} - \frac{x^{\text{fixed}}(i)}{2} : \sigma^{(i)} \\
& \quad y^{(i)} \leq 10 - x^{\text{fixed}}(i) : \gamma^{(i)} \\
& \quad y^{(i)} \geq 0
\end{align*}

Subproblem:

\begin{align*}
\text{minimize} & \quad -\frac{x^{(i)}}{4} + \alpha^{(i)} \\
\text{subject to} & \quad 0 \leq x^{(i)} \leq 16 \\
& \quad \alpha^{(i)} \geq \alpha^{\text{down}} \\
& \quad \alpha^{(i)} \geq \pi^{(k)}[5 + x^{(i)}] + \mu^{(k)}\left[\frac{15}{2} + \frac{x^{(i)}}{2}\right] + \sigma^{(k)}\left[\frac{35}{2} - \frac{x^{(i)}}{2}\right] \\
& \quad + \gamma^{(k)}[10 - x^{(i)}] \quad \forall k = 1, \ldots, i - 1
\end{align*}

Master problem:

\begin{align*}
& \quad i : \text{ current Benders' iteration} \\
& \quad k : \text{ set of previous iterations}
\end{align*}
Two-stage Deterministic Problem

Simple Example

Subproblem:

minimize \(- y^{(i)}\)
subject to
\(y^{(i)} \leq 5 + x^{\text{fixed}}^{(i)} : \pi^{(i)}\)
\(y^{(i)} \leq \frac{15}{2} + \frac{x^{\text{fixed}}^{(i)}}{2} : \mu^{(i)}\)
\(y^{(i)} \leq \frac{35}{2} - \frac{x^{\text{fixed}}^{(i)}}{2} : \sigma^{(i)}\)
\(y^{(i)} \leq 10 - x^{\text{fixed}}^{(i)} : \gamma^{(i)}\)
\(y^{(i)} \geq 0\)

\(i:\) current Benders’ iteration
\(k:\) set of previous iterations

Master problem:

minimize \(- \frac{x^{(i)}}{4} + \alpha^{(i)}\)
subject to
\(0 \leq x^{(i)} \leq 16\)
\(\alpha^{(i)} \geq \alpha^{\text{down}}\)
\(\alpha^{(i)} \geq \pi^{(k)}[5 + x^{(i)}] + \mu^{(k)}\left[\frac{15}{2} + \frac{x^{(i)}}{2}\right] + \sigma^{(k)}\left[\frac{35}{2} - \frac{x^{(i)}}{2}\right]\)
\(+ \gamma^{(k)}[10 - x^{(i)}] \ \forall k = 1, \ldots, i - 1\)
Two-stage Deterministic Problem

Simple Example

**Subproblem:**

\[
\begin{align*}
\text{minimize } & -y^{(i)} \\
\text{subject to } & y^{(i)} \leq 5 + x^{\text{fixed}}^{(i)} : \pi^{(i)} \\
& y^{(i)} \leq \frac{15}{2} + \frac{x^{\text{fixed}}^{(i)}}{2} : \mu^{(i)} \\
& y^{(i)} \leq \frac{35}{2} - \frac{x^{\text{fixed}}^{(i)}}{2} : \sigma^{(i)} \\
& y^{(i)} \leq 10 - x^{\text{fixed}}^{(i)} : \gamma^{(i)} \\
& y^{(i)} \geq 0
\end{align*}
\]

**Master problem:**

\[
\begin{align*}
\text{minimize } & -\frac{x^{(i)}}{4} + \alpha^{(i)} \\
\text{subject to } & 0 \leq x^{(i)} \leq 16 \\
& \alpha^{(i)} \geq \alpha^{\text{down}} \\
& \alpha^{(i)} \geq \pi^{(k)} \left[5 + x^{(i)}\right] + \mu^{(k)} \left[\frac{15}{2} + \frac{x^{(i)}}{2}\right] + \sigma^{(k)} \left[\frac{35}{2} - \frac{x^{(i)}}{2}\right] \\
& + \gamma^{(k)} \left[10 - x^{(i)}\right] \forall k = 1, \ldots, i - 1
\end{align*}
\]

\(i\): current Benders’ iteration
\(k\): set of previous iterations

Note: In subproblem, symbols following colon are dual variables (sensitivities).

Note: The last constraint of master problem generate “cuts”.
Two-stage Deterministic Problem

Simple Example

Algorithm:

• Step 0: Initialization
  Set $i = 1$, $x^{\text{fixed (1)}} = x^{\text{initial}}$, and lower bound (LB) = $-\infty$

• Step 1: Solve subproblem(s): obtain the values of all dual variables (sensitivities), and the value of objective function, which is upper bound (UB)

• Step 2: Convergence check
  If $|UB - LB| \leq \epsilon$, then the optimal solution with a level of accuracy $\epsilon$ is obtained, otherwise $i \leftarrow i + 1$

• Step 3: Solve master problem: obtain the updated $x^{(i)}$ and the value of $\alpha^{(i)}$ as the updated LB, and go Step 1 with the updated fixed $x$

Jalal Kazempour
A more compact form of Benders’ decomposition:

**Subproblem:**

\[
\begin{align*}
\text{minimize} & \quad y^{(i)} - x^{(i)} \\
\text{subject to} & \quad y^{(i)} - x^{(i)} \leq 5 \\
& \quad y^{(i)} - \frac{x^{(i)}}{2} \leq \frac{15}{2} \\
& \quad y^{(i)} + \frac{x^{(i)}}{2} \leq \frac{35}{2} \\
& \quad y^{(i)} + x^{(i)} \leq 10 \\
& \quad y^{(i)} \geq 0 \\
& \quad x^{(i)} = x^{\text{fixed (i)}} : \rho^{(i)}
\end{align*}
\]

**Master problem:**

\[
\begin{align*}
\text{minimize} & \quad \frac{x^{(i)}}{4} + \alpha^{(i)} \\
\text{subject to} & \quad 0 \leq x^{(i)} \leq 16 \\
& \quad \alpha^{(i)} \geq \alpha^{\text{down}} \\
& \quad \alpha^{(i)} \geq -y^{(k)} + \rho^{(k)} [x^{(i)} - x^{(k)}] \quad \forall k = 1, \ldots, i - 1
\end{align*}
\]
Exercise:

The printed version of GAMS code of Benders’ decomposition example is available on your table. Please check it in the next 10 minutes, then explain it to your neighbors around the table.

My colleagues and I will answer your questions!

This code has been prepared by Lejla Halilbasic, Christos Ordoudis, and Jalal Kazempour.
Two-stage **Stochastic** Problem

*Two-stage (day-ahead and real-time) stochastic OPF problem*
Two-stage Stochastic Problem

Two-stage (day-ahead and real-time) stochastic OPF problem

- Day-ahead (DA) market
- Real-time (RT) market

DA decisions

- RT decisions for scenario 1
- RT decisions for scenario 2
- RT decisions for scenario n

Jalal Kazempour
Two-stage Stochastic Problem

Two-stage (day-ahead and real-time) stochastic OPF problem

Day-ahead (DA) decisions:

Real-time (RT) decisions:
Two-stage Stochastic Problem

Two-stage (day-ahead and real-time) stochastic OPF problem

Day-ahead (DA) decisions:
- Power schedule [MW] of each generator ($\forall g: 1, \ldots, G$), which is $P_g$.

**Note:** this variable is **scenario-independent**, in the sense that it should be adapted to all foreseen scenarios (here-and-now decisions).

Real-time (RT) decisions:
Two-stage Stochastic Problem

Two-stage (day-ahead and real-time) stochastic OPF problem

Day-ahead (DA) decisions:
- Power schedule [MW] of each generator ($\forall g: 1, \ldots, G$), which is $P_g$.

Note: this variable is scenario-independent, in the sense that it should be adapted to all foreseen scenarios (here-and-now decisions).

Real-time (RT) decisions:

(For given DA decisions)
- Reserve deployment [MW] of each generator $g$ under each foreseen scenario ($\forall s: 1, \ldots, S$), which is $r_{g,s}$.
- Load shedding, wind curtailment, etc (all indexed by $s$).

Note: this variable is scenario-dependent (wait-and-see decisions).
Two-stage Stochastic Problem

Two-stage (day-ahead and real-time) stochastic OPF problem

- Is there any complicating variable in this two-stage stochastic problem?

- DA decisions
  - RT decisions for scenario 1
  - RT decisions for scenario 2
    - RT decisions for scenario \( n \)
Two-stage Stochastic Problem

Two-stage (day-ahead and real-time) stochastic OPF problem

• Is there any complicating variable in this two-stage stochastic problem?

DA decisions

RT decisions for scenario 1
RT decisions for scenario 2
RT decisions for scenario n

Fix DA decisions to given values!
Two-stage Stochastic Problem

Two-stage (day-ahead and real-time) stochastic OPF problem

- Is there any **complicating variable** in this two-stage stochastic problem?

DA decisions

- RT decisions for scenario 1
- RT decisions for scenario 2
- ...
- RT decisions for scenario \( n \)

Fix DA decisions to given values!
Two-stage Stochastic Problem

Two-stage (day-ahead and real-time) stochastic OPF problem

- Is there any complicating variable in this two-stage stochastic problem?

DA decisions

Fix DA decisions to given values!

Then, how many subproblems will you have?

Jalal Kazempour
Two-stage Stochastic Problem

Two-stage (day-ahead and real-time) stochastic OPF problem

- Is there any complicating variable in this two-stage stochastic problem?

DA decisions

[Diagram showing DA decisions leading to RT decisions for scenario 1, RT decisions for scenario 2, ..., RT decisions for scenario n]

Fix DA decisions to given values!

Then, how many subproblems will you have? One per scenario!
Two-stage Stochastic Problem

Two-stage (day-ahead and real-time) stochastic OPF problem

Master problem

Day-ahead (DA) decision-making problem (DA-OPF)

Real-time (RT) decision-making problems (RT-OPF), one per scenario

Subproblems (one per scenario)

Fixed DA variables

Sensitivities (dual variables)
The application of Benders’ decomposition to two-stage stochastic problems is also referred in the literature as L-shaped decomposition!
Some more thoughts

What to do in case of multi-stage (e.g., 4-stage) stochastic problem?
Some more thoughts

What to do in case of multi-stage (e.g., 4-stage) stochastic problem?

- **Step 1 in each iteration**: Solve a single two-stage stochastic problem
- **Master problem**: Stage 1
- **Subproblems**: Stage 2
Some more thoughts

What to do in case of multi-stage (e.g., 4-stage) stochastic problem?

- **Step 2 in each iteration**: Solve 3 two-stage stochastic problems (separately)
- **Master problems**: Stage 2
- **Subproblems**: Stage 3
Some more thoughts

What to do in case of multi-stage (e.g., 4-stage) stochastic problem?

• **Step 3 in each iteration**: Solve 9 two-stage stochastic problems (separately)
• **Master problems**: Stage 3
• **Subproblems**: Stage 4
Some more thoughts

What to do in case of multi-stage (e.g., 4-stage) stochastic problem?

Each multi-stage stochastic problem is a collection of **nested** two-stage stochastic problems.
Some more thoughts

What to do in case of multi-stage (e.g., 4-stage) stochastic problem?

Each multi-stage stochastic problem is a collection of **nested** two-stage stochastic problems

Techniques to be used:
- Nested Benders’ decomposition
- Stochastic dual dynamic programming (SDDP)

Some more thoughts

What to do in a case in which subproblems include **binary (0/1)** variables?
Some more thoughts

What to do in a case in which subproblems include **binary (0/1)** variables?

The trouble?

Sensitivities (dual variables) cannot be obtained in a discrete feasible region! But, we need those sensitivities to generate cuts in the master problem!
Some more thoughts

What to do in a case in which subproblems include **binary (0/1)** variables?

**Solution:**

- Generate cuts in the master problem based on the values obtained for **“primal” variables, and not “dual” variables** of the subproblems!
Some more thoughts

What to do in a case in which subproblems include binary (0/1) variables?

**Solution technique:**

- Primal Benders’ decomposition (cutting-plane method)

  This technique has recently been used in power systems applications for two-stage robust optimization problems


Additional References


Thanks for your attention!

Email: seykaz@elektro.dtu.dk