BENDERS DECOMPOSITION

1. Benders decomposition is a widely used technique in mathematical optimization for breaking down a complex problem into smaller, more manageable subproblems.
2. It's particularly useful when dealing with mixed-integer linear programming (MILP) or mixed-integer nonlinear programming (MINLP) problems

Steps Involved to solve problems using Benders Decomposition:

1. **Problem Setup**:
    a. Identify a large-scale optimization problem that can be decomposed into smaller, more tractable subproblems.
    b. The problem should have a natural decomposition structure
2. **Main Problem (Master Problem):**
    a. The main problem is the original optimization problem you want to solve.
    b. It typically includes some or all of the original variables and constraints.
    c. However, it often omits certain constraints (the complicating constraints) that make the problem difficult to solve directly.
3. **Subproblems**:
    a. The subproblems are auxiliary optimization problems that help solve the main problem.
    b. Each subproblem typically involves a subset of the original variables and some of the complicating constraints omitted from the main problem.
    c. These subproblems are easier to solve compared to the original problem.

4. **Benders Cuts**:
   - Benders cuts are inequalities generated by solving the subproblems.
   - These cuts are added to the main problem to strengthen it and tighten the relaxation.
   - Benders cuts are designed to eliminate infeasible solutions and improve the lower bound of the main problem.

5. **Iterative Process**:
   - The Benders decomposition process is iterative. You start with an initial feasible solution or an empty solution and then alternate between solving the main problem and the subproblems.
      - In each iteration:
      - Solve the main problem to obtain a solution.
      - Solve the subproblems to generate Benders cuts.
      - Add the Benders cuts to the main problem and re-solve it.
      - Repeat until convergence (when no further improvements can be made).

6. **Termination**:
   - The process continues until the solution to the main problem converges to the optimal solution of the original problem.

SOLVING THE PROBLEM:(From Slides)

$$\min 9a + 10b + \text{‘}13c + 4d$$

$$s.t. 4a + 6b + 3c + d \geqslant 12$$

$$a + b + 2c + 2d \geqslant 7$$

$$a, b, c \geqslant 0; d \in \mathbb{Z}^{+}; d \leqslant 8$$

Let’s consider the problem below with two sets of variables x and y.

$$
\begin{aligned}
\text{minimize} \quad & \mathbf{c}^{\mathrm{T}}\mathbf{x} + \mathbf{d}^{\mathrm{T}}\mathbf{y} \\
\text{subject to} \quad & A\mathbf{x} + B\mathbf{y} \geq \mathbf{b} \\
& y \in Y \\
& \mathbf{x} \geq \mathbf{0}
\end{aligned}
$$

Assuming that y variables are the complicating variables. When fixing y, the problem can be written as:

$$\begin{array}{ll} \text{minimize} & \mathbf{c}^{\mathrm{T}}\mathbf{x} + \mathbf{d}^{\mathrm{T}}\bar{\mathbf{y}} \\ \text{subject to} & A\mathbf{x} \geq \mathbf{b} - B\bar{\mathbf{y}} \\ & \mathbf{x} \geq \mathbf{0} \end{array}$$

Benders SubProblem

$$\begin{array}{ll} \text{maximize} & (\mathbf{b} - B\bar{\mathbf{y}})^{\mathrm{T}}\mathbf{u} + \mathbf{d}^{\mathrm{T}}\bar{\mathbf{y}} \\ \text{subject to} & A^{\mathrm{T}}\mathbf{u} \leq \mathbf{c} \\ & \mathbf{u} \geq \mathbf{0} \end{array}$$

The model below is referred to as Benders' master problem.

$$\begin{array}{ll} \text{minimize} & \mathbf{z} \\ \text{subject to} & \{\text{cuts}\} \\ & \mathbf{y} \in Y \end{array}$$

Master Problem (MP):
The master problem includes some of the original variables and a relaxed version of the constraints.
In this case, we'll include **a, b, and c** as **continuous variables** and relax both constraints.

Objective:
Minimize $z(d)$
$d \le 8$

Subproblem (SP):
$$
\begin{aligned}
\max\ & 12y0 + 7y1 \\
\text{s.t.}\ & 4y0 + y1 \le 9 \\
& 6y0 + y1 \le 10 \\
& 3y0 + 2y1 \le 13
\end{aligned}
$$

In each iteration of the Benders decomposition algorithm:

1. Start solving the master problem to get a solution.
2. Then, solve the subproblem to generate Benders cuts.
3. The Benders cuts are added to the master problem to strengthen it.
4. We repeat this process iteratively until the master problem converges to the optimal solution of the original problem.

Further explanation with the help of code.
The problem is solved using Gurobi solver:

```python
import gurobipy as gp
from gurobipy import GRB
```

```python
# Create the master problem
master = gp.Model("Master Problem")
a = master.addVar(lb=0, vtype=GRB.CONTINUOUS, name="a")
b = master.addVar(lb=0, vtype=GRB.CONTINUOUS, name="b")
c = master.addVar(lb=0, vtype=GRB.CONTINUOUS, name="c")
d = master.addVar(lb=0, ub=8, vtype=GRB.INTEGER, name="d")
```

```python
# Create the subproblem
sub = gp.Model("Subproblem")
y0 = sub.addVar(lb=0, name="y0")
y1 = sub.addVar(lb=0, name="y1")

sub.setObjective(12 * y0 + 7 * y1, GRB.MAXIMIZE)
sub.addConstr(4 * y0 + y1 <= 9, "subconstraint1")  # Corresponding to the first constraint
sub.addConstr(6 * y0 + y1 <= 10, "subconstraint2")  # Corresponding to the second constraint
sub.addConstr(3 * y0 + 2 * y1 <= 13, "subconstraint2")  # Corresponding to the third constraint

converged = False
while not converged:
    # Solve the master problem
    master.optimize()

    # Solve the subproblem
    sub.optimize()

    if sub.objVal < 1e-6:
        converged = True

    # Update the coefficients of the Benders' cut constraint
    benders_cut_constr.setAttr(GRB.Attr.LB, 12 * y0.x + 7 * y1.x)
    #After solving the subproblem, you update the coefficients of the Benders' cut constraint in the master problem
```

```python
# Display the results
if master.status == GRB.OPTIMAL:
    print("Optimal Solution:")
    print(f"a = {a.x}")
    print(f"b = {b.x}")
    print(f"c = {c.x}")
    print(f"d = {d.x}")
    print(f"Objective Value: {master.objVal}")
else:
    print("No optimal solution found.")
```