Integer Programming
ISE 418

Lecture 21

Dr. Ted Ralphs
Suresh Bolusani
Variable Decomposition

- Recall the basic principle of decomposition: by relaxing/fixing the linking variables/constraints, we then get a model that is easier to solve.

- Here, we discuss methods of decomposing by fixing complicating variables.

- “Classical” decomposition arises from block structure in the constraint matrix.

\[
\begin{pmatrix}
A_{10} & A_{11} \\
A_{20} & A_{22} \\
\vdots & \ddots \\
A_{\gamma 0} & & A_{\kappa \kappa}
\end{pmatrix}
\]

- After fixing variables the problem becomes separable and the separability lends itself nicely to parallel implementation.

- However, there can be other reasons why problems become easier to solve upon fixing certain variables.
(Generalized) Benders’ Decomposition

- Most of what we’re referring to as *variable decomposition* methods are derivatives of an algorithm proposed by Benders.

- Benders’ original method was for the case of LPs, but the algorithm is easy to generalize.

- From a mathematical standpoint, Benders’ method amounts to projection of the problem into the space of a subset of the variables.

- The projection effectively amounts to a reformulation of the problem in terms of the value function of a restriction of the problem.
Benders’ Principle (Linear Programming)

\[ z_{LP} = \max_{(x,y) \in \mathbb{R}^n_+} \{ cx + dy \mid Ax \leq b, Dx + Gy \leq d \} \]

\[ = \max_{x \in \mathbb{R}^{n'}} \{ cx + \phi(d - Dx) \mid Ax \leq b \}, \]

where

\[ \phi(\beta) = \min \{ dy \mid G y \leq \beta \} \]

\[ s.t. \quad y \in \mathbb{R}^{n''}_+ \]

Basic Strategy:

- The function \( \phi \) is the value function of a linear program.
- We iteratively approximate it by generating \textit{dual functions}. 
Example

\[ z_{LP} = \max \quad -x - y \]

s.t. \[ -25x + 20y \leq 30 \]
      \[ x + 2y \leq 10 \]
      \[ 2x - y \leq 15 \]
      \[ -2x - 10y \leq -15 \]
      \[ x, y \in \mathbb{R} \]
Value Function Reformulation

\[ z_{LP} = \min_{x \in \mathbb{R}} -x + \phi_x(x), \]

where

\[ \phi_x(x) = \max -y \]

s.t. \quad 20y \leq 30 + 25x

\[ 2y \leq 10 - x \]

\[ -y \leq 15 - 2x \]

\[ -10y \leq -15 + 2x \]

\[ y \in \mathbb{R} \]

- Note that \( \phi_x(x) = \phi(d - Dx) \) and is not actually the value function itself.
- The reformulated problem can be interpreted precisely as the projection into the space of the first set of variables.
Benders’ Principle (Integer Programming)

\[
z_{LP} = \max_{(x,y) \in \mathbb{Z}_+^n} \{ cx + dy \mid Ax \leq b, Dx + Gy \leq d \}
\]

\[
= \max_{x \in \mathbb{Z}_+^{n'}} \{ cx + \phi(d - Dx) \mid Ax \leq b \},
\]

where

\[
\phi(\beta) = \min dy
\]

s.t. \( Gy \leq \beta \)

\( y \in \mathbb{Z}_+^{n''} \)

**Basic Strategy:**

- Here, \( \phi \) is the value function of an integer program.
- Here, we also iteratively generate an approximation by constructing a dual functions.
Example

\[ z_{IP} = \max (-x - y) \]

s.t. \[ -25x + 20y \leq 30 \]
     \[ x + 2y \leq 10 \]
     \[ 2x - y \leq 15 \]
     \[ -2x - 10y \leq -15 \]

\[ x, y \in \mathbb{Z} \]
Value Function Reformulation

\[ z_{IP} = \min_{x \in \mathbb{Z}} x + \phi(x), \]

where

\[ \phi_x(x) = \max \ -y \]

s.t. \[ 20y \leq 30 + 25x \]
\[ 2y \leq 10 - x \]
\[ -y \leq 15 - 2x \]
\[ -10y \leq -15 + 2x \]
\[ y \in \mathbb{Z} \]

- Note again that \( \phi_x(x) = \phi(d - Dx) \) and so is not the value function itself.
Generalized Benders

Benders’ Master Problem (iteration $k$)

$$\begin{align*}
\text{max } & \quad cx + z \\
\text{subject to } & \quad Ax \leq b \\
& \quad z \leq \phi_i(d - Dx), 1 \leq i \leq k \\
& \quad x \in \mathbb{Z}^{n'}
\end{align*}$$

Basic Scheme

- Solve master problem to obtain new candidate solution $x^k$ and lower bound.
- Solve subproblem by evaluating $\phi(d - dx^k)$ to obtain $\overline{\phi}_k$ (dual function and new upper bound).
- Terminate when upper bound equals lower bound.

Where do we get $\overline{\phi}_k$?
Constructing the Dual Function

• $\bar{\phi}_k$ is a dual function that we construct by evaluating $\phi(d - Dx^k)$.

• We have “projected” out the second set of variables.

• Each evaluation of $\phi$ yields information that we can use to build up our approximation.

• The algorithm gives us a natural way of sampling the domain.

• In the LP case, $\phi$ is a piece-wise linear concave function.

• The classical Benders’ algorithm approximates $\phi$ as the minimum of linear dual functions, which arise as the dual solutions to the LP.

• Hence, Benders can be seen as a cutting plane method in this case.
An MILP Example

\[
\begin{align*}
\text{min} & \quad -x_1 + y_1 + y_2 + y_3 \\
\text{s.t.} & \quad -x_1 + 2y_1 - y_2 + y_3 = 0 \\
& \quad x_1 \in [0, 3] \\
& \quad x_1, y_1 \in \mathbb{Z}_+ \\
& \quad y_2, y_3 \in \mathbb{R}_+
\end{align*}
\]

Master problem:

\[
\begin{align*}
\text{min} & \quad -x_1 + \theta \\
\text{s.t.} & \quad \theta \geq \phi(x_1) \\
& \quad x_1 \in [0, 3] \\
& \quad x_1 \in \mathbb{Z}_+ \\
& \quad \theta \text{ free}
\end{align*}
\]

Subproblem \((\beta = x_1)\):

\[
\begin{align*}
\phi(\beta) = \min & \quad y_1 + y_2 + y_3 \\
\text{s.t.} & \quad 2y_1 - y_2 + y_3 = \beta \\
& \quad y_1 \in \mathbb{Z}_+ \\
& \quad y_2, y_3 \in \mathbb{R}_+
\end{align*}
\]
An MILP Example

Subproblem:

\[ \phi(\beta) = \min \ y_1 + y_2 + y_3 \]

s.t. 2y_1 - y_2 + y_3 = \beta

y_1 \in \mathbb{Z}_+

y_2, y_3 \in \mathbb{R}_+
Example

Iteration 1:

\[
\begin{align*}
\text{min} & \quad -x_1 \\
\text{s.t.} & \quad x_1 \in [0, 3] \\
& \quad x_1 \in \mathbb{Z}_+ \\
\end{align*}
\]

\[
\begin{align*}
\phi(\beta = x_1^1) & = \min y_1 + y_2 + y_3 \\
\text{s.t.} & \quad 2y_1 - y_2 + y_3 = 3 \\
& \quad y_1 \in \mathbb{Z}_+ \\
& \quad y_2, y_3 \in \mathbb{R}_+ \\
\end{align*}
\]

\[x_1^1 = 3, \theta^1 = -\infty\]
Example

Iteration 2:

\[
\begin{align*}
\min & \quad -x_1 + \theta \\
\text{s.t.} & \quad \theta \geq \min \{x_1 - 1, -x_1 + 6\} \\
& \quad x_1 \in [0, 3] \\
& \quad x_1 \in \mathbb{Z}_+ \\
\end{align*}
\]

\[
\begin{align*}
\phi(\beta = x_1^2) &= \min \; y_1 + y_2 + y_3 \\
\text{s.t.} & \quad 2y_1 - y_2 + y_3 = 0 \\
& \quad y_1 \in \mathbb{Z}_+ \\
& \quad y_2, y_3 \in \mathbb{R}_+ \\
\end{align*}
\]

\[x_1^2 = 0, \; \theta^2 = -1\]
Example

Iteration 3:

\[
\min -x_1 + \theta \\
\text{s.t.} \quad \theta \geq \min \{x_1 - 1, -x_1 + 6\} \\
\theta \geq -x_1 \\
x_1 \in [0, 3] \\
x_1 \in \mathbb{Z}_+
\]

\[
\phi(\beta = x_1^3) = \min y_1 + y_2 + y_3 \\
\text{s.t.} \quad 2y_1 - y_2 + y_3 = 1 \\
y_1 \in \mathbb{Z}_+ \\
y_2, y_3 \in \mathbb{R}_+
\]

\[x_1^3 = 1, \theta^3 = 0\]
Example

Iteration 4:

\[
\begin{align*}
\text{min } & -x_1 + \theta \\
\text{s.t. } & \theta \geq \min\{x_1 - 1, -x_1 + 6\} \\
& \theta \geq -x_1 \\
& \theta \geq \min\{x_1, -x_1 + 3\} \\
& x_1 \in [0, 3] \\
& x_1 \in \mathbb{Z}_+ \\
\end{align*}
\]

\[
\begin{align*}
\phi(\beta = x_1^4) = & \min y_1 + y_2 + y_3 \\
\text{s.t. } & 2y_1 - y_2 + y_3 = 3 \\
& y_1 \in \mathbb{Z}_+ \\
& y_2, y_3 \in \mathbb{R}_+ \\
\end{align*}
\]

\[
x_1^4 = 3, \theta^4 = 2
\]