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)

$$z_{\mathrm{LP}} = \max_{(x,y)\in\mathbb{R}^n_+} \{cx + dy \mid Ax \le b, Dx + Gy \le d\}$$
$$= \max_{x\in\mathbb{R}^{n'}} \{cx + \phi(d - Dx) \mid Ax \le b\},$$

where

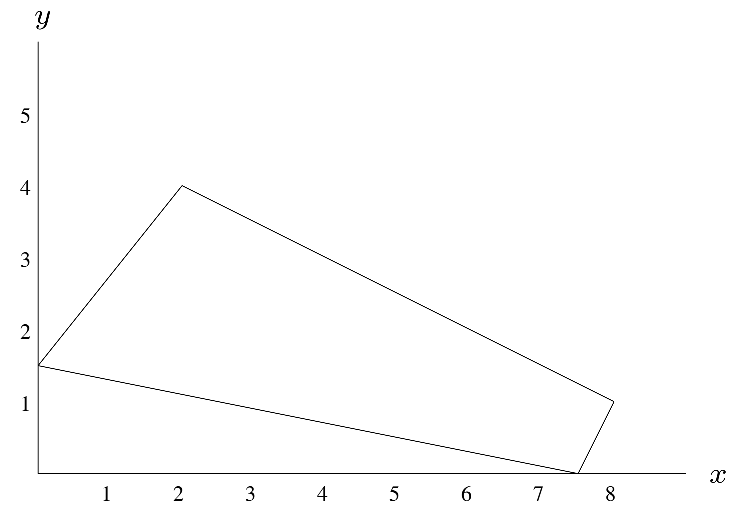
$$\phi(\beta) = \min dy$$
$$\text{s.t. } Gy \le \beta$$
$$y \in \mathbb{R}^{n''}_+$$

Basic Strategy:

- The function ϕ is the value function of a linear program.
- We iteratively approximate it by generating *dual functions*.

Example

$$
\begin{aligned}
z_{LP} \quad = \quad \max \quad & -x - y \\
\text{s.t.} \quad & -25x + 20y \le 30 \\
& x + 2y \le 10 \\
& 2x - y \le 15 \\
& -2x - 10y \le -15 \\
& x, y \in \mathbb{R}
\end{aligned}
$$

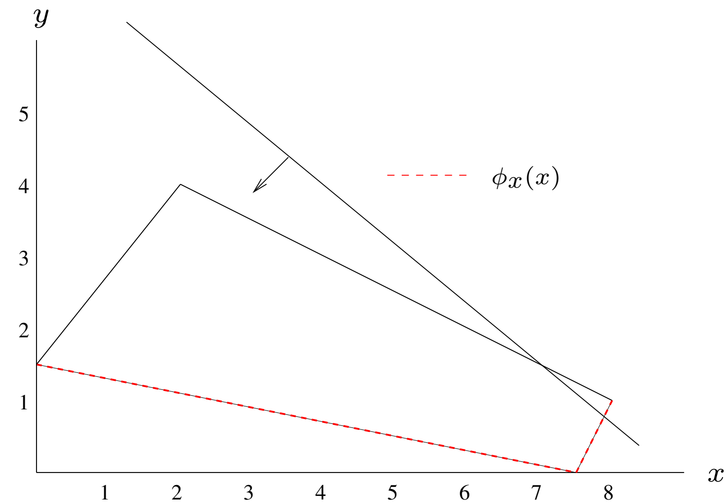


Value Function Reformulation

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

where

$$\begin{aligned} \phi_x(x) = \max \quad & -y \\ \text{s.t.} \quad & 20y \leq 30 + 25x \\ & 2y \leq 10 - x \\ & -y \leq 15 - 2x \\ & -10y \leq -15 + 2x \\ & y \in \mathbb{R} \end{aligned}$$



- Note that $\phi_x(x) = \phi(d - Dx)$ and isnot 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)

$$z_{\text{LP}} = \max_{(x,y)\in\mathbb{Z}^n_+} \{cx + dy \mid Ax \le b, Dx + Gy \le d\}$$

$$= \max_{x\in\mathbb{Z}^{n'}_+} \{cx + \phi(d - Dx) | Ax \le b\},$$

where

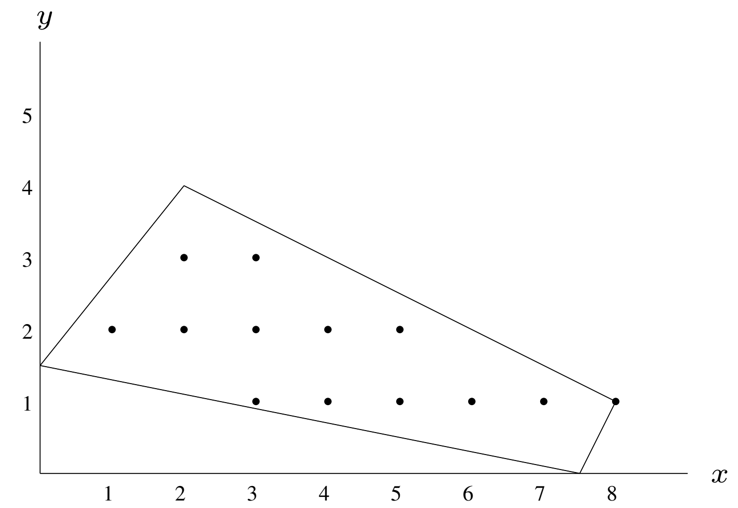
$$\begin{aligned}\phi(\beta) = \ & \min dy \\ & \text{s.t. } Gy \le \beta \\ & y \in \mathbb{Z}^{n''}_+\end{aligned}$$

Basic Strategy:

- Here, ϕ is the value function of an integer program.
- Here, we also iteratively generate an approximation by constructing a dual functions.

Example

$$
\begin{aligned}
z_{IP} \quad = \quad \max \quad & -x - y \\
\text{s.t.} \quad & -25x + 20y \leq 30 \\
& x + 2y \leq 10 \\
& 2x - y \leq 15 \\
& -2x - 10y \leq -15 \\
& x, y \in \mathbb{Z}
\end{aligned}
$$

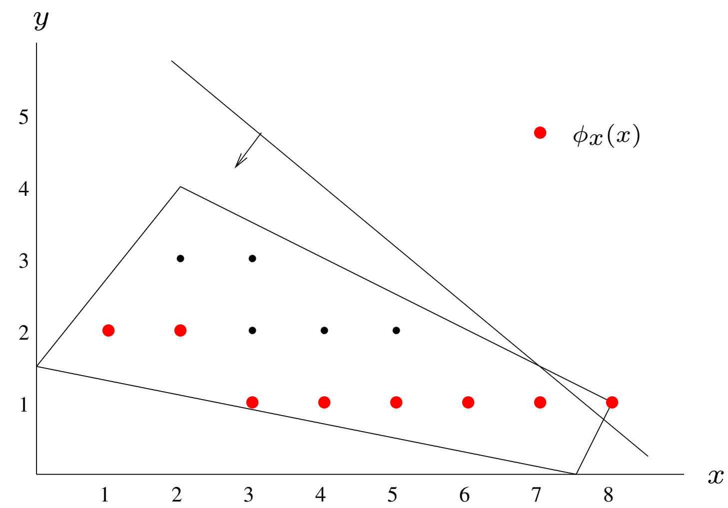


Value Function Reformulation

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

where

$$\begin{aligned} \phi_x(x) = \max \quad & -y \\ \text{s.t.} \quad & 20y \leq 30 + 25x \\ & 2y \leq 10 - x \\ & -y \leq 15 - 2x \\ & -10y \leq -15 + 2x \\ & y \in \mathbb{Z} \end{aligned}$$



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

Generalized Benders

Benders' Master Problem (iteration k)

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

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

- $\overline{\phi}_k$ 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 ϕ 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, ϕ is a piece-wise linear concave function.
- The classical Benders' algorithm approximates ϕ 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

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

Master problem:

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

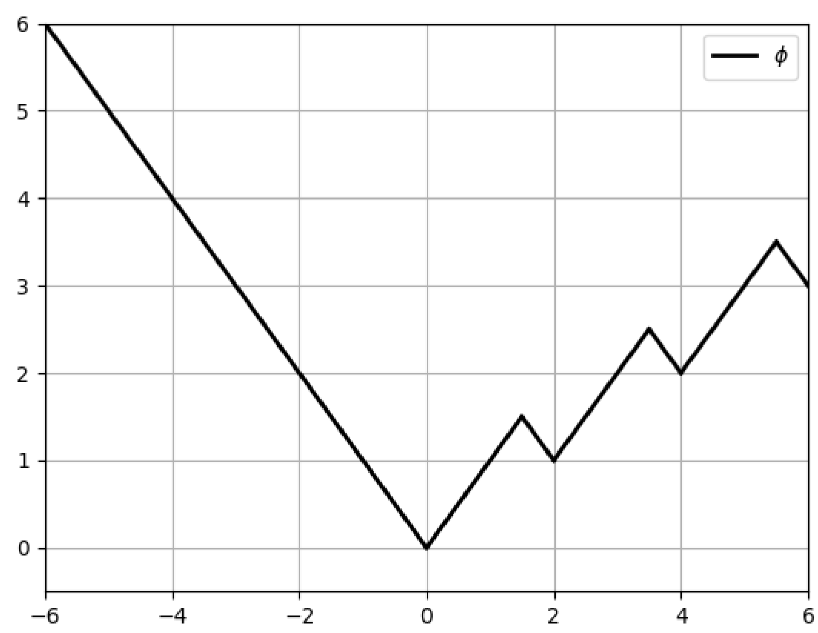
Subproblem ($\beta = x_1$):

$$
\begin{aligned}
\phi(\beta) = \min \;& y_1 + y_2 + y_3 \\
\text{s.t.} \;& 2y_1 - y_2 + y_3 = \beta \\
& y_1 \in \mathbb{Z}_+ \\
& y_2, y_3 \in \mathbb{R}_+
\end{aligned}
$$

An MILP Example

Subproblem:

$$
\begin{aligned}
\phi(\beta) = \min \ & y_1 + y_2 + y_3 \\
\text{s.t.} \ & 2y_1 - y_2 + y_3 = \beta \\
& y_1 \in \mathbb{Z}_+ \\
& y_2, y_3 \in \mathbb{R}_+
\end{aligned}
$$



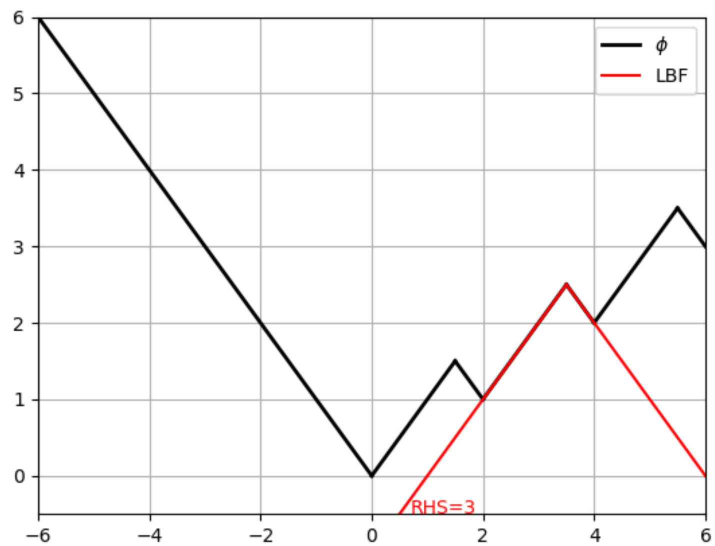
Example

Iteration 1:

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

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

$x_1^1 = 3, \theta^1 = -\infty$



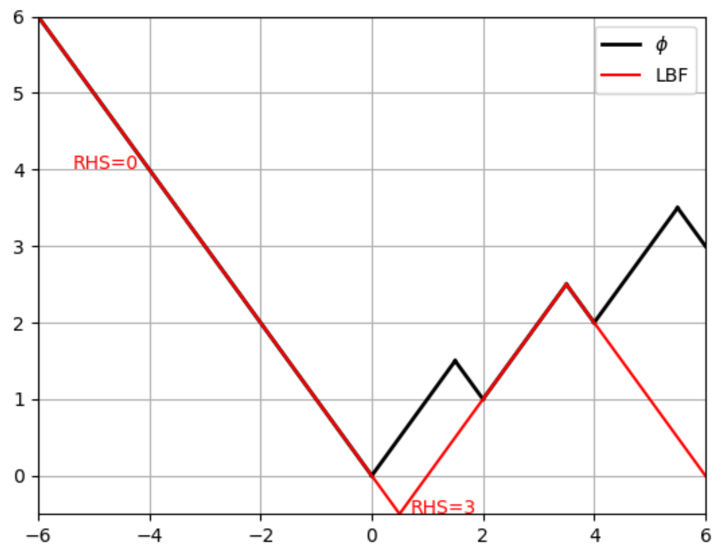
Example

Iteration 2:

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

$$\begin{aligned} \phi(\beta = x_1^2) = \min\ & y_1 + y_2 + y_3 \\ \text{s.t.}\ & 2y_1 - y_2 + y_3 = 0 \\ & y_1 \in \mathbb{Z}_+ \\ & y_2, y_3 \in \mathbb{R}_+ \end{aligned}$$

$x_1^2 = 0, \theta^2 = -1$



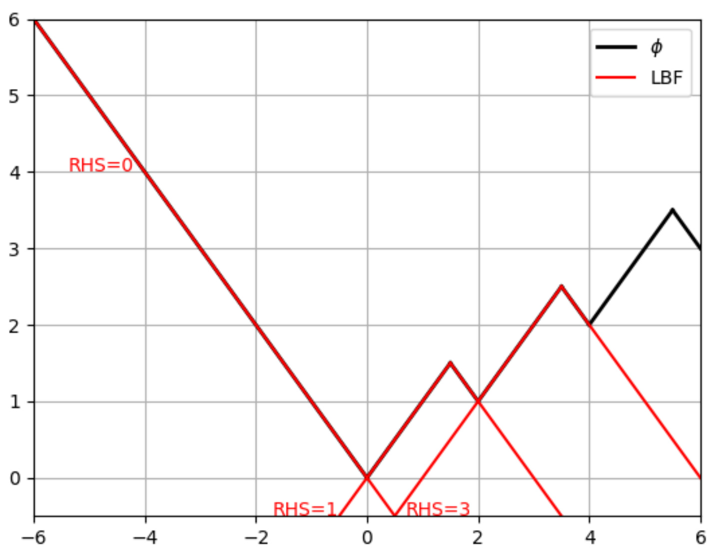
Example

Iteration 3:

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

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

$x_1^3 = 1, \theta^3 = 0$



Example

Iteration 4:

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

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

$x_1^4 = 3, \theta^4 = 2$