

Program of the day:

- Dantzig-Wolfe reformulation of an IP
- Master problem
- Subproblem
- Linear Programming – canonical form, reduced costs, etc. (Taha chapter 7)
- Example: Cutting stock problem

1

Dantzig-Wolfe Decomposition

Motivation

- If you have a large or difficult problem: split it up into smaller pieces

Applications

- Facility location problems
- Cutting Stock problems
- Air-crew Scheduling
- Vehicle Routing Problems
- ...

Two currently most promising directions for MIP:

- Branch-and-price
- Branch-and-cut

2

Dantzig-Wolfe Decomposition

The problem is split into a master problem and a subproblem

- Tighter bounds
- Better control of subproblem
- Model may become (very) large

Delayed column generation

Write up the decomposed model gradually as needed


- Generate a few solutions to the subproblems
- Solve the master problem to LP-optimality
- Use the dual information to find most promising solutions to the subproblem
- Extend the master problem with the new subproblem solutions.


3

Motivation: Cutting stock problem

- Infinite number of raw stocks, having length L
- Cut m piece types i , each having width w_i demand b_i
- Satisfy demands using least possible raw stocks

Example:

- $w_1 = 5, b_1 = 7$ 

- $w_2 = 3, b_2 = 3$ 

- Raw length $L = 22$



Some possible cuts



4

Dantzig-Wolfe decomposition

Substituting X^k in original model getting *Master Problem*

$$\begin{aligned} \max \quad & c^1 \left(\sum_{t \in T_1} \lambda_{1,t} x^{1,t} \right) + c^2 \left(\sum_{t \in T_2} \lambda_{2,t} x^{2,t} \right) + \dots + c^K \left(\sum_{t \in T_K} \lambda_{K,t} x^{K,t} \right) \\ \text{s.t.} \quad & A^1 \left(\sum_{t \in T_1} \lambda_{1,t} x^{1,t} \right) + A^2 \left(\sum_{t \in T_2} \lambda_{2,t} x^{2,t} \right) + \dots + A^K \left(\sum_{t \in T_K} \lambda_{K,t} x^{K,t} \right) = b \\ & \sum_{t \in T_k} \lambda_{k,t} = 1 \quad k = 1, \dots, K \\ & \lambda_{k,t} \in \{0, 1\}, \quad t \in T_k \quad k = 1, \dots, K \end{aligned}$$

9

Strength of linear master model

Solving LP-relaxation of master problem, is equivalent to (Wolsey Prop 11.1)

$$\begin{aligned} \max \quad & c^1 x^1 + c^2 x^2 + \dots + c^K x^K \\ \text{s.t.} \quad & A^1 x^1 + A^2 x^2 + \dots + A^K x^K = b \\ & x^1 \in \text{conv}(X^1) \quad x^2 \in \text{conv}(X^2) \quad \dots \quad x^K \in \text{conv}(X^K) \end{aligned}$$

Proof: Consider LP-relaxation

$$\begin{aligned} \max \quad & c^1 \left(\sum_{t \in T_1} \lambda_{1,t} x^{1,t} \right) + c^2 \left(\sum_{t \in T_2} \lambda_{2,t} x^{2,t} \right) + \dots + c^K \left(\sum_{t \in T_K} \lambda_{K,t} x^{K,t} \right) \\ \text{s.t.} \quad & A^1 \left(\sum_{t \in T_1} \lambda_{1,t} x^{1,t} \right) + A^2 \left(\sum_{t \in T_2} \lambda_{2,t} x^{2,t} \right) + \dots + A^K \left(\sum_{t \in T_K} \lambda_{K,t} x^{K,t} \right) = b \\ & \sum_{t \in T_k} \lambda_{k,t} = 1 \quad k = 1, \dots, K \\ & \lambda_{k,t} \geq 0, \quad t \in T_k \quad k = 1, \dots, K \end{aligned}$$

Informally speaking we have

- joint constraint is solved to LP-optimality
- block constraints are solved to IP-optimality

10

Strength of Lagrangian relaxation

- z^{LPM} be LP-solution value of master problem
- z^{LD} be solution value of Lagrangian dual problem

(Theorem 11.2)

$$z^{LPM} = z^{LD}$$

Proof: Lagrangian relaxing joint constraint in

$$\begin{aligned} \max \quad & c^1 x^1 + c^2 x^2 + \dots + c^K x^K \\ \text{s.t.} \quad & A^1 x^1 + A^2 x^2 + \dots + A^K x^K = b \\ & D^1 x^1 + \dots \leq d_1 \\ & \quad \quad \quad + D^2 x^2 \leq d_2 \\ & \quad \quad \quad \dots \leq \vdots \\ & \quad \quad \quad D^K x^K \leq d_K \\ & x^1 \in \mathbb{Z}_+^{n_1} \quad x^2 \in \mathbb{Z}_+^{n_2} \quad \dots \quad x^K \in \mathbb{Z}_+^{n_K} \end{aligned}$$

Using result next page

$$\begin{aligned} \max \quad & c^1 x^1 + c^2 x^2 + \dots + c^K x^K \\ \text{s.t.} \quad & A^1 x^1 + A^2 x^2 + \dots + A^K x^K = b \\ & x^1 \in \text{conv}(X^1) \quad x^2 \in \text{conv}(X^2) \quad \dots \quad x^K \in \text{conv}(X^K) \end{aligned}$$

11

Strength of Lagrangian Relaxation (section 10.2)

Integer Programming Problem

$$\begin{aligned} \text{maximize} \quad & cx \\ \text{subject to} \quad & Ax \leq b \\ & Dx \leq d \\ & x_j \in \mathbb{Z}_+, \quad j = 1, \dots, n \end{aligned}$$

Lagrange Relaxation, multipliers $\lambda \geq 0$

$$\begin{aligned} \text{maximize} \quad & z_{LR}(\lambda) = cx - \lambda(Dx - d) \\ \text{subject to} \quad & Ax \leq b \\ & x_j \in \mathbb{Z}_+, \quad j = 1, \dots, n \end{aligned}$$

for best multiplier $\lambda \geq 0$



$$\boxed{\max \left\{ cx : Dx \leq d, x \in \text{conv}(Ax \leq b, x \in \mathbb{Z}_+) \right\}}$$

12

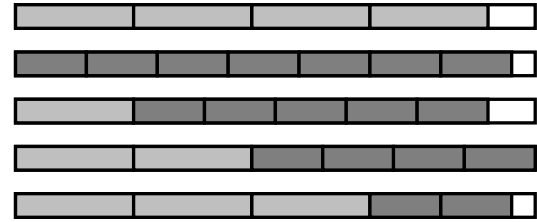
Delayed column generation, linear master

- Master problem can (and will) contain many columns
- To find bound, solve LP-relaxation of master
- Delayed column generation gradually writes up master

Delayed column generation, linear master

- $w_1 = 5, b_1 = 7$ 
- $w_2 = 3, b_2 = 3$ 
- Raw length $L = 22$

Some possible cuts



In matrix form

$$A = \begin{pmatrix} 4 & 0 & 1 & 2 & 3 & \dots \\ 0 & 7 & 5 & 4 & 2 & \dots \end{pmatrix}$$

LP-problem

$$\begin{aligned} \min \quad & cx \\ \text{s.t.} \quad & Ax = b \\ & x \geq 0 \end{aligned}$$

where

- $b = (7, 3)$,
- $x = (x_1, x_2, x_3, x_4, x_5, \dots)$
- $c = (1, 1, 1, 1, 1, \dots)$.

Simplex in matrix form (Taha section 7.1)

$$\begin{aligned} \text{maximize} \quad & cx \\ \text{subject to} \quad & Ax = b \\ & x \geq 0, \end{aligned}$$

Reformulation in matrix form

$$\begin{pmatrix} 1 & -c \\ 0 & A \end{pmatrix} \begin{pmatrix} z \\ x \end{pmatrix} = \begin{pmatrix} 0 \\ b \end{pmatrix}$$

Assume

- x_B set of basis variables
- c_B coefficients in c corresponding to basis
- A_B coefficients in A corresponding to basis

By multiplying with

$$\begin{pmatrix} 1 & c_B A_B^{-1} \\ 0 & A_B^{-1} \end{pmatrix}$$

get equivalent form

$$\begin{pmatrix} 1 & c_B A_B^{-1} A - c \\ 0 & A_B^{-1} A \end{pmatrix} \begin{pmatrix} z \\ x \end{pmatrix} = \begin{pmatrix} c_B A_B^{-1} b \\ A_B^{-1} b \end{pmatrix}$$

Table page 295 in Taha

basis	x_j	solution
z	$c_B A_B^{-1} A_j - c_j$	$c_B A_B^{-1} b$
x_B	$A_B^{-1} A_j$	$A_B^{-1} b$

Simplex in matrix form

Basis variables and non-basis variables

$$x_B = (x_1, x_2) \quad x_N = (x_3, x_4, x_5, \dots)$$

split matrix

$$A_B = \begin{pmatrix} 4 & 0 \\ 0 & 7 \end{pmatrix} \quad A_N = \begin{pmatrix} 1 & 2 & 3 & \dots \\ 5 & 4 & 2 & \dots \end{pmatrix}$$

reformulated problem

$$\begin{aligned} \min \quad & c_B x_B + c_N x_N \\ \text{s.t.} \quad & A_B x_B + A_N x_N = b \\ & x \geq 0 \end{aligned}$$

Simplex algorithm sets $x_N = 0$ and solves $A_B x_B = b$ getting

$$x_B = A_B^{-1} b$$



corresponding objective function

$$z = c_B A_B^{-1} b + x_N (c_N - c_B A_B^{-1} A_N)$$

Since dual variables $y = c_B A_B^{-1}$ we have

$$z = yb + x_N (c_N - yA_N)$$

Delayed column generation (example)

- $w_1 = 5, b_1 = 7$ 
- $w_2 = 3, b_2 = 3$ 
- Row length $L = 22$

Initially we choose only the trivial cutting patterns

$$A = \begin{pmatrix} 4 & 0 \\ 0 & 7 \end{pmatrix}$$

Solve LP-problem

$$\begin{aligned} \min \quad & cx \\ \text{s.t.} \quad & Ax = b \\ & x \geq 0 \end{aligned}$$

i.e.

$$\begin{pmatrix} 4 & 0 \\ 0 & 7 \end{pmatrix} \begin{pmatrix} x_1 \\ x_2 \end{pmatrix} = \begin{pmatrix} 7 \\ 3 \end{pmatrix}$$

with solution $x_1 = \frac{7}{4}$ and $x_2 = \frac{3}{7}$.

The dual variables are $y = c_B A_B^{-1}$ i.e.

$$(1 \ 1) \begin{pmatrix} \frac{1}{4} & 0 \\ 0 & \frac{1}{7} \end{pmatrix} = \begin{pmatrix} \frac{1}{4} \\ \frac{1}{7} \end{pmatrix}$$

17

Small example (continued)

Find entering variable

$$A = \begin{pmatrix} 1 & 2 & 3 & \dots \\ 5 & 4 & 2 & \dots \end{pmatrix} \quad \begin{matrix} \frac{1}{4} \leftarrow y_1 \\ \frac{1}{7} \leftarrow y_2 \end{matrix}$$

$$c_N - yA_N = \left(1 - \frac{27}{28} \quad 1 - \frac{30}{28} \quad 1 - \frac{29}{28} \quad \dots \right)$$

We could also solve optimization problem

$$\begin{aligned} \min \quad & 1 - \frac{1}{4}x_1 - \frac{1}{7}x_2 \\ \text{s.t.} \quad & 5x_1 + 3x_2 \leq 22 \\ & x \geq 0, \text{ integer} \end{aligned}$$

which is equivalent to knapsack problem

$$\begin{aligned} \max \quad & \frac{1}{4}x_1 + \frac{1}{7}x_2 \\ \text{s.t.} \quad & 5x_1 + 3x_2 \leq 22 \\ & x \geq 0, \text{ integer} \end{aligned}$$

This problem has optimal solution $x_1 = 2, x_2 = 4$.

Reduced cost of entering variable

$$1 - 2\frac{1}{4} - 4\frac{1}{7} = 1 - \frac{30}{28} = -\frac{1}{14} < 0$$

18

Small example (continued)

Add new cutting pattern to A getting

$$A = \begin{pmatrix} 4 & 0 & 2 \\ 0 & 7 & 4 \end{pmatrix}$$

Solve problem to LP-optimality, getting primal solution

$$x_1 = \frac{11}{8}, x_3 = \frac{3}{4}$$

and dual variables

$$y_1 = \frac{1}{4}, y_2 = \frac{1}{8}$$

Note, we do not need to care about “leaving variable”

To find entering variable, solve

$$\begin{aligned} \max \quad & \frac{1}{4}x_1 + \frac{1}{8}x_2 \\ \text{s.t.} \quad & 5x_1 + 3x_2 \leq 22 \\ & x \geq 0, \text{ integer} \end{aligned}$$

This problem has optimal solution $x_1 = 4, x_2 = 0$.

Reduced cost of entering variable

$$1 - 4\frac{1}{4} - 0\frac{1}{8} = 0$$

Terminate with $x_1 = \frac{11}{8}, x_3 = \frac{3}{4}$, and $z_{LP} = \frac{17}{8} = 2.125$.

19

Questions

- Will the process terminate?

Always improving objective value. Only a finite number of basis solutions.

- Can we repeat the same pattern?

No, since the objective functions is improved. We know the best solution among existing columns. If we generate an already existing column, then we will not improve the objective.

20

The Cutting Stock Problem (general model)

- Infinite number of raw stocks, having length L
- Cut m piece types i , each having width w_i demand b_i
- Satisfy demands using least possible raw stocks

IP-problem:

$$\begin{aligned} \min \quad & \sum_{j=1}^n x_j \\ \text{s.t.} \quad & \sum_{i=1}^m w_i a_{ij} \leq Lx_j & j = 1, \dots, n \\ & \sum_{j=1}^n a_{ij} \geq b_i, & i = 1, \dots, m \\ & a_{ij} \geq 0, \text{ integer} & i = 1, \dots, m, j = 1, \dots, n \\ & x_j \in \{0, 1\} & j = 1, \dots, n \end{aligned}$$

21

Cutting Stock: Better relaxation

- Write up all different cutting patterns
- Solve the LP-relaxation

$$\begin{aligned} \min \quad & \sum_{j=1}^n x_j \\ \text{s.t.} \quad & \sum_{j=1}^n a_{ij} x_j \geq b_i, & i = 1, \dots, m \\ & x_j \in \{0, 1\} & j = 1, \dots, n \end{aligned}$$

where

- n number of cutting patterns (large)
- a_{ij} is number of pieces type i in pattern j
- x_j is 1 if pattern j is used

22

Cutting Stock: Delayed column generation

Choosing variable to enter basis

$$z = yb + x_N(c_N - yA_N)$$

where $y = c_B A_B^{-1}$.

For every column a_j in A , the coefficient of variable x_j is

$$1 - \sum_{i=1}^m y_i a_{ij}$$

Thus to find the most negative coefficient we solve

$$\begin{aligned} z^S = \min \quad & 1 - \sum_{i=1}^m y_i x_i \\ \text{s.t.} \quad & \sum_{i=1}^m w_i x_i \leq L \\ & x \geq 0, \text{ integer} \end{aligned}$$

Delayed column generation

- Columns in A are generated on the fly
- The process is greedy
- Terminate when no variable can enter basis ($z^S \geq 0$)
- Hopefully a small set of columns need to be generated

23

Branch and price

Terminology

- Master Problem
- Restricted Master Problem
- Subproblem or Pricing Problem
- Branch and cut: Branch-and-bound algorithm using cuts to strengthen bounds.
- Branch and price: Branch-and-bound algorithm using column generation to derive bounds.
- One says that discarded columns are “priced out”.

24

Branch-and-price

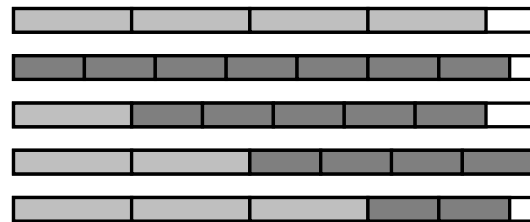
- LP-solution of master problem may have fractional solutions
- Branch-and-bound for getting IP-solution
- In each node solve LP-relaxation of master
- Subproblem may change when we add constraints to master problem
- Branching strategy should make subproblem easy to solve

25

Branch-and-price, example

The matrix A contains all different cutting patterns

$$A = \begin{pmatrix} 4 & 0 & 1 & 2 & 3 \\ 0 & 7 & 5 & 4 & 2 \end{pmatrix}$$



Problem

$$\begin{aligned} & \text{minimize } \lambda_1 + \lambda_2 + \lambda_3 + \lambda_4 + \lambda_5 \\ & \text{subject to } 4\lambda_1 + 0\lambda_2 + 1\lambda_3 + 2\lambda_4 + 3\lambda_5 \geq 7 \\ & \quad \quad \quad 0\lambda_1 + 7\lambda_2 + 5\lambda_3 + 4\lambda_4 + 2\lambda_5 \geq 3 \\ & \quad \quad \quad \lambda_j \in \mathbb{Z}_+ \end{aligned}$$

LP-solution $\lambda_1 = 1.375, \lambda_4 = 0.75$

Branch on $\lambda_1 = 0, \lambda_1 = 1, \lambda_1 = 2$

- Column generation may not generate pattern (4,0)
- Pricing problem is knapsack problem with pattern forbidden

26

Branch-and-price, example

Better branching strategy

- Branch 1: item i and item j are cut from *same* pattern
- Branch 2: item i and item j are cut from *different* patterns

Pricing problem

- Branch 1: “glue together” the two items
- Branch 2: multiple-choice knapsack problem

Will the branching terminate?

- Sooner or later we will have defined exactly which items go into same pattern, i.e. all patterns

27

Tailing off effect

Column generation may converge slowly in the end

- We do not need exact solution, just lower bound
- Solving master problem for subset of columns does not give valid lower bound (why?)
- Instead we may use Lagrangian relaxation of joint constraint
- “guess” lagrangian multipliers equal to dual variables from master problem

Heuristic solution

- Restricted master problem will only contain a subset of the columns
- We may solve restricted master problem to IP-optimality
- Restricted master is a “set-covering-like” problem which is not too difficult to solve

28