DM872

Math Optimization at Work

Dantzig-Wolfe Decomposition and Delayed Column Generation

Marco Chiarandini

Department of Mathematics & Computer Science
University of Southern Denmark

Outline

1. Solving the Linear Master Problem

2. Solving the Master Problem: Branch and Price

Outline

Solving the Linear Master Problem

Integer Programming Problem with block structure:

$$
\begin{aligned}
z_{MP} = \max \quad & c^1 \sum_{t=1}^{T_1} \lambda_{1,t} x^{1,t} + c^2 \sum_{t=1}^{T_2} \lambda_{2,t} x^{2,t} + \ldots + c^K \sum_{t=1}^{T_K} \lambda_{K,t} x^{K,t} \\
& A^1 \Big(\sum_{t=1}^{T_1} \lambda_{1,t} x^{1,t}\Big) + A^2 \Big(\sum_{t=1}^{T_2} \lambda_{2,t} x^{2,t}\Big) + \ldots + A^K \Big(\sum_{t=1}^{T_K} \lambda_{K,t} x^{K,t}\Big) = b \\
& \sum_{t=1}^{T_k} \lambda_{k,t} = 1 \qquad k = 1, \ldots, K \\
& \lambda_{k,t} \in \{0,1\} \qquad t \in T_k, k = 1, \ldots, K
\end{aligned}
$$

Let's consider the case $K = 1$

$$
\begin{aligned}
z_{MP} = \max \quad & \sum_{t=1}^{T} (cx^t) \lambda_t \\
& \sum_{t=1}^{T} (Ax^t) \lambda_t = b \\
& \sum_{t=1}^{T} \lambda_t = 1 \\
& \lambda_t \in \{0,1\} \qquad t \in T
\end{aligned}
$$

$$
\begin{aligned}
z_{LMP} = \max \quad & \sum_{t=1}^{T} (cx^t) \lambda_t \\
& \sum_{t=1}^{T} (Ax^t) \lambda_t = b \\
& \sum_{t=1}^{T} \lambda_t = 1 \\
& \lambda_t \geq 0 \qquad t \in T
\end{aligned}
$$

Restricted LMP and Dual

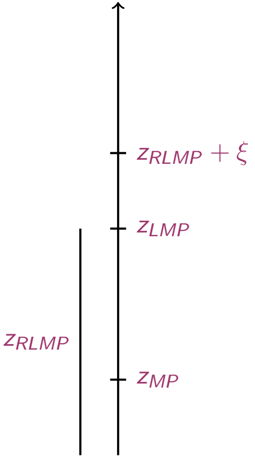
$$
\begin{aligned}
z_{LMP} = \max \ & \sum_{t=1}^{T} (cx^t)\lambda_t \\
& \sum_{t=1}^{T} (Ax^t)\lambda_t = b \\
& \sum_{t=1}^{T} \lambda_t = 1 \\
& \lambda_t \geq 0 \qquad\qquad t \in T
\end{aligned}
$$

$$
\begin{aligned}
z_{RLMP} = \max \ & \sum_{t=1}^{p} (cx^t)\lambda_t \\
& \sum_{t=1}^{p} (Ax^t)\lambda_t = b \\
& \sum_{t=1}^{p} \lambda_t = 1 \\
& \lambda_t \geq 0 \qquad\qquad t = 1, \ldots, p
\end{aligned}
$$

$$
\begin{aligned}
z_{DLMP} = \min \ & \pi b + \pi_0 \\
& \pi A^T x^t + \pi_0 \geq cx^t, \ \ t = 1, \ldots, T \\
& \pi \in \mathbb{R}^m \\
& \pi_0 \in \mathbb{R}
\end{aligned}
$$

$$
\begin{aligned}
z_{DRLMP} = \min \ & \pi b + \pi_0 \\
& \pi A^T x^t + \pi_0 \geq cx^t, \ \ t = 1, \ldots, p \\
& \pi \in \mathbb{R}^m \\
& \pi_0 \in \mathbb{R}
\end{aligned}
$$

Column Generation Process and Dual Bound



- $z_{LMP} \geq z_{MP}$ because linear relaxation
- $z_{LMP} \geq z_{RLMP}$ because of simplex theory (some columns missing)

while True **do**

  solve RLMP and get z_{RLMP}, λ^* and (π^*, π_0^*);

  solve subproblem (pricing or constraint violation)

  $\xi = \max\{cx^t - \pi^* A^T x^t - \pi_0^* \mid x^t \in X\}$ and get x^*;

  $z_{MP} \leq z_{LMP} \leq z_{RLMP} + \xi$ hence, valid dual bound on z_{MP};

  if $\xi = 0$ **then**

    $z_{LMP} = z_{RLMP}$ and stop column generation process

  if $\xi > 0$ **then**

    stop if $\pi^*(Ax^* - b) = 0$;

  else

    add column $(cx^*, Ax^*, 1)$;

Column Generation Process and Dual Bounds

Initialization Solving RLMP gives an optimal primal solution λ^* and an optimal dual soluton $(\pi^*, \pi_0^*) \in \mathbb{R}^m \times \mathbb{R}^1$

Primal Feasibility λ^* is also *feasible* solution for LMP and so $z_{LRMP} = \sum_{t=1}^{p}(cx^t)\lambda_t^* = \pi^* b + \pi_0^* \leq z_{LMP}$

Optimality Check for LMP Is λ^* and (π^*, π_0^*) optimal for LMP? To answer solve subproblem:

$$\xi = \max_x\{(c - \pi^* A)x - \pi_0^* \mid x \in X\}$$

with optimal solution x^*.

Dual Bound from the subproblem:
for all $x \in X$, $\xi \geq (c - \pi^* A)x - \pi_0^* \iff$
for all $x \in X$, $(c - \pi^* A)x - \pi_0^* - \xi \leq 0$
Hence, $(\pi^*, \pi_0^* + \xi)$ is dual *feasible* for LMP

$$z_{LMP} \leq \pi^* b + \pi_0^* + \xi$$

Column Generation Process and Dual Bounds

Stopping Criterion If $\xi = 0$ then (π^*, π_0^*) is dual *feasible* for LMP and so $\pi^* b + \pi_0^* \geq z_{LMP}$ and λ^* is *optimal* for LMP.

Generating a New Column If $\xi > 0$ add column $(cx^*, Ax^*, 1)^T$

Alternative Stopping Criterion If $\xi > 0$ and subproblem solution x^* satisfies linking constraints and $\pi^*(Ax^* - b) = 0$ then x^* is optimal for LMP. Indeed

$$\xi = (c - \pi^* A)x^* - \pi_0^* \implies cx^* = \pi^* Ax^* + \pi_0^* + \xi = \pi^* b + \pi_0^* + \xi$$

Finding an Initial Feasible Solution introduce m artificial variables and add or one artificial variable and column $(b, 1)^T$. Large cost for the artificial variables.

Valid dual bounds in delayed CG

Linear relaxation of the reduced master problem:

$$z_{LRMP} = \max\left\{c\lambda \mid \bar{A}\lambda \leq b, \lambda \geq 0\right\}$$

Note: $z_{LRMP} \not\geq z_{LMP}$ (LMP Lin. relax. master problem)

However, during colum generation we have access to a dual bound so that we can terminate the process when a desired solution quality is reached.

When we know that

$$\sum_{j \in J} \lambda_j \leq \kappa \qquad J \text{ is the unrestricted set of columns}$$

for an optimal solution of the master, we cannot improve z_{RMP} by more than κ times the largest reduced cost obtained by the Pricing Problem (PP):

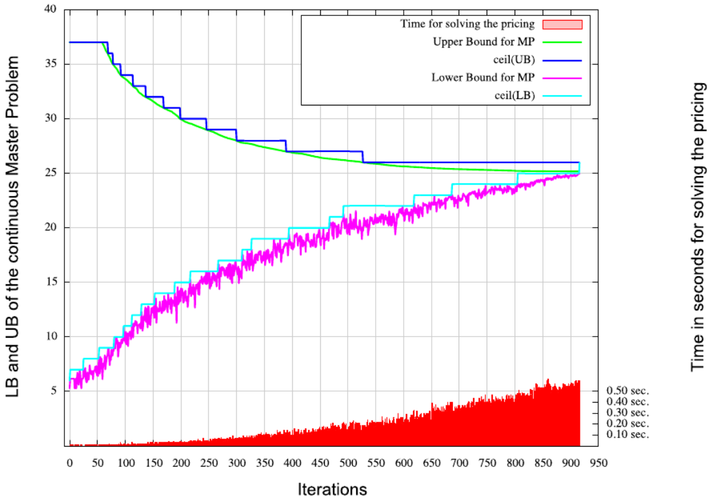
$$z_{LRMP} + \kappa z_{PP} \geq z_{LMP}$$

(It can be shown that this bound coincides with the Lagrangian dual bound.)

- with convexity constraints $\sum_{j \in J} \lambda_j \leq 1$ then $\kappa = 1$
- when $\boldsymbol{c} = \mathbf{1}$ we can set $\kappa = z_{LMP}$ and derive the better dual bound $\frac{z_{LRMP}}{1 - z_{PP}} \geq z_{LMP}$

Convergence in CG

In general the dual bound is not monotone during the iterations, for a **problem of minimum**:



[plot by Stefano Gualandi, Milan University]

Questions

- Will the process terminate?

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

- Can we repeat the same pattern, assuming the simplex is not cycling?

  No, since the objective function is improving. We know the best solution among existing columns. If we generate an already existing column, then we will not improve the objective. However, we may be in the case of degenerate tableaux.

Consider the following LP problem

$$\begin{aligned}
\text{maximize} \quad & x_1 + 2x_2 + 4x_3 + 8x_4 + 16x_5 \\
\text{st} \quad & x_1 + 2x_2 + 3x_3 + 4x_4 + 5x_5 \le 5 \\
& 7x_1 + 5x_2 - 3x_3 - 2x_4 \le 0 \\
& x_1, x_2, x_3, x_4, x_5 \ge 0
\end{aligned}$$

```
|--------+--------+--------+--------+--------+--------+--------+--------+--------+
|     x1 |     x2 |     x3 |     x4 |     x5 |     x6 |     x7 |     -z |      b |
|--------+--------+--------+--------+--------+--------+--------+--------+--------+
|      0 |   9/35 |  24/35 |  32/35 |      1 |    1/5 |  -1/35 |      0 |      1 |
|      1 |    5/7 |   -3/7 |   -4/7 |      0 |      0 |    1/7 |      0 |      0 |
|--------+--------+--------+--------+--------+--------+--------+--------+--------+
|      0 | -99/35 |-229/35 |-212/35 |      0 |  -16/5 |  11/35 |      1 |    -16 |
|--------+--------+--------+--------+--------+--------+--------+--------+--------+
```

Let x_6, x_7 be the two slack variables.

- $[x_3, x_5]$ and $[x_5, x_7]$ are both optimal bases and they both give the same optimal solution
- the the tableau is degenerate: a $\bar{b}$ term is null and one of the basic variables has value 0, entering variable stays at zero.
- a tableau implies a unique vertex but we may have the same vertex represented by different tableaux like here (hence a vertex does not imply a unique tableau). tableau $\Longrightarrow$ vertex; tableu $\not\Longleftarrow$ vertex
- the optimal dual solutions are two different (we need different multipliers): $[3.21.87], [3.2, 0]$, respectively.

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

Outline

Mixed Integer Linear Programs

- The primary use of column generation is in this context
  (in LP simplex is better)
- column generation re-formulations often give much stronger bounds than the original LP relaxation
- The use of column generation in this context gives rise to what is known as branch-and-price
- 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.

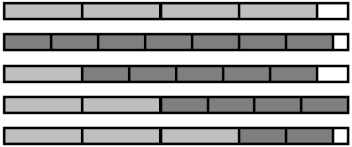
Branch-and-price

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

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

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

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

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
- Note, it can lead to infeasibility

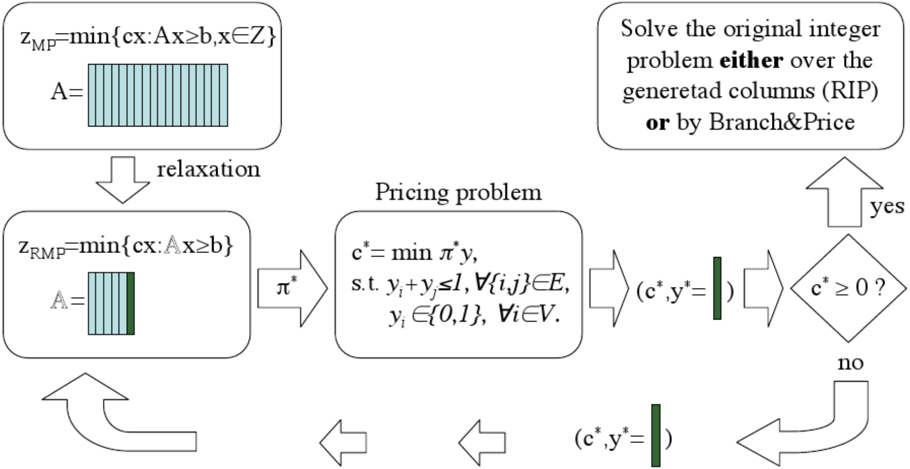
Branching constraints

- branch on original variables or on column variables
- disadvantages of branching on column variables: B&B tree unbalanced and subproblem difficult to solve

Solving the LP master at a node

The constraints introduced for branching (and other cutting planes) change the master problem or the subproblem. Where they should be considered is a design choice.

Summary



[illustration by Stefano Gualandi, Milan Un.]
(the pricing problem is for a GCP)