DM872 – Spring 2019

Math Optimization at Work

Advanced Methods for MILP

Marco Chiarandini

Department of Mathematics & Computer Science
University of Southern Denmark

[*Partly based on slides by David Pisinger, DIKU (now DTU)*]

Outline

Outline

Outline

Relaxation

In branch and bound we find upper bounds by relaxing the problem

Relaxation

$$\max_{s \in P} g(s) \geq \left\{ \begin{array}{l} \max_{s \in P} f(s) \\ \max_{s \in S} g(s) \end{array} \right\} \geq \max_{s \in S} f(s)$$

- P: candidate solutions;
- $S \subseteq P$ feasible solutions;
- $g(x) \geq f(x)$

Which constraints should be relaxed?

- Quality of bound (tightness of relaxation)
- Remaining problem can be solved efficiently
- Proper multipliers can be found efficiently
- Constraints difficult to formulate mathematically
- Constraints which are too expensive to write up

Different relaxations

- LP-relaxation
- Deleting constraint
- Lagrange relaxation
- Surrogate relaxation
- Semidefinite relaxation

Tighter

Best surrogate relaxation

Best Lagrangian relaxation

LP relaxation

Relaxations are often used in combination.

Tightness of relaxation

$$
\begin{aligned}
\max\ & cx \\
\text{s.t. }\ & Ax \le b \\
& Dx \le d \\
& x \in \mathbb{Z}_+^n
\end{aligned}
$$

LP-relaxation:

$$\max\{cx \ : \ x \in \operatorname{conv}(Ax \le b, Dx \le d, x \in \mathbb{Z}_+)\}$$

$\rightsquigarrow$ Lagrangian Relaxation:

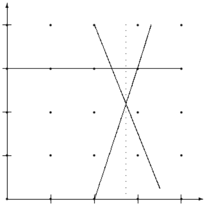
$$
\begin{aligned}
\max\ & z_{LR}(\lambda) = cx - \lambda(Dx - d) \\
\text{s.t. }\ & Ax \le b \\
& x \in \mathbb{Z}_+^n
\end{aligned}
$$

LP-relaxation:

$$\max\{cx \ : \ Dx \le d, x \in \operatorname{conv}(Ax \le b, x \in \mathbb{Z}_+)\}$$

Surrogate relaxation, example

$$
\begin{array}{lrcrcl}
\text{maximize} & 4x_1 & + & x_2 & & \\
\text{subject to} & 3x_1 & - & x_2 & \leq & 6 \\
 & & & x_2 & \leq & 3 \\
 & 5x_1 & + & 2x_2 & \leq & 18 \\
 & x_1, & & x_2 & \geq & 0, \text{ integer}
\end{array}
$$



IP solution $(x_1, x_2) = (2, 3)$ with $z_{IP} = 11$
LP solution $(x_1, x_2) = (\frac{30}{11}, \frac{24}{11})$ with $z_{LP} = \frac{144}{11} = 13.1$

First and third constraint complicating, surrogate relax using multipliers $\lambda_1 = 2$, and $\lambda_3 = 1$

$$
\begin{array}{lrcrcl}
\text{maximize} & 4x_1 & + & x_2 & & \\
\text{subject to} & & & x_2 & \leq & 3 \\
 & 11x_1 & & & \leq & 30 \\
 & x_1, & & x_2 & \geq & 0, \text{ integer}
\end{array}
$$

Solution $(x_1, x_2) = (2, 3)$ with $z_{SR} = 4 \cdot 2 + 3 = 11$
Upper bound

Surrogate relaxation

Integer Programming Problem: $\max\{cx \mid Ax \leq b, Dx \leq d, x \in \mathbb{Z}^n_+\}$
Relax complicating constraints $Dx \leq d$.
$\rightsquigarrow$ Surrogate Relax $Dx \leq d$ using multipliers $\lambda \geq 0$, i.e., add together constraints using weights λ

$$\begin{aligned} z_{SR}(\lambda) = \max\ & cx \\ \text{s.t.}\ & Ax \leq b \\ & \lambda Dx \leq \lambda d \\ & x \in \mathbb{Z}^n_+ \end{aligned}$$

Surrogate Dual Problem

$$z_{SD} = \min_{\lambda \geq 0} z_{SR}(\lambda)$$

LP Relaxation:

$$\max\left\{cx\ :\ x \in \mathrm{conv}(Ax \leq b, \lambda Dx \leq \lambda d, x \in \mathbb{Z}^n_+)\right\}$$

best surrogate relaxation (i.e., best λ multipliers) is tighter than best Lagrangian relax.

Relaxation strategies

Which constraints should be relaxed

- "the complicating ones"
- remaining problem is polynomially solvable
  (e.g. min spanning tree, assignment problem, linear programming)
- remaining problem is totally unimodular
  (e.g. network problems)
- remaining problem is NP-hard but good techniques exist
  (e.g. knapsack)
- constraints which cannot be expressed in MIP terms
  (e.g. cutting)
- constraints which are too extensive to express
  (e.g. subtour elimination in TSP)

Subgradient optimization Lagrange multipliers

$$\begin{aligned} \max \ & z = cx \\ \text{s.t.}\ & Ax \leq b \\ & Dx \leq d \\ & x \in \mathbb{Z}_+^n \end{aligned}$$

Lagrange Relaxation, multipliers $\lambda \geq 0$

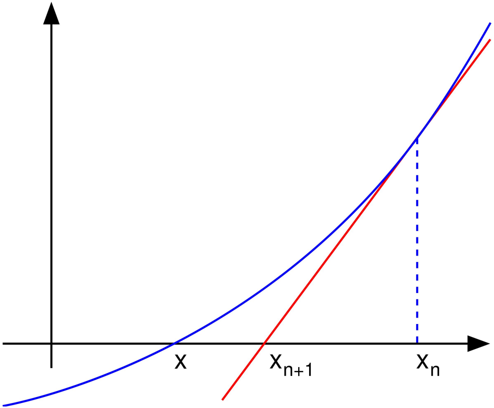
$$\begin{aligned} z_{LR}(\lambda) = \max \ & cx - \lambda(Dx - d) \\ \text{s.t.}\ & Ax \leq b \\ & x \in \mathbb{Z}_+^n \end{aligned}$$

Lagrange Dual Problem

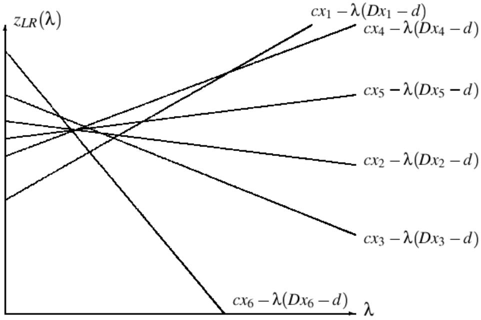
$$z_{LD} = \min_{\lambda \geq 0} z_{LR}(\lambda)$$

- We do not need best multipliers in B&B algorithm

Subgradient optimization, motivation



Netwon-like method to minimize a function in one variable



Lagrange function $z_{LR}(\lambda)$ is piecewise linear and convex

Digression: Gradient methods

Gradient methods are iterative approaches:

- find a descent direction with respect to the objective function f
- move x in that direction by a step size

The descent direction can be computed by various methods, such as gradient descent, Newton-Raphson method and others. The step size can be computed either exactly or loosely by solving a line search problem.

Example: gradient descent

1. Set iteration counter $t = 0$, and make an initial guess x_0 for the minimum
2. Repeat:
3. Compute a descent direction $\Delta_t = \nabla(f(x_t))$
4. Choose α_t to minimize $f(x_t - \alpha\Delta_t)$ over $\alpha \in \mathbb{R}_+$
5. Update $x_{t+1} = x_t - \alpha_t\Delta_t$, and $t = t + 1$
6. Until $\|\nabla f(x_k)\| < tolerance$

Step 4 can be solved 'loosely' by taking a fixed small enough value $\alpha > 0$

Newton-Raphson method

[from Wikipedia]

Find zeros of a real-valued derivable function

$$x : f(x) = 0 .$$

- Start with a guess x_0
- Repeat:
  Move to a better approximation

  $$x_{n+1} = x_n - \frac{f(x_n)}{f'(x_n)}$$

  until a sufficiently accurate value is reached.

Geometrically, $(x_n, 0)$ is the intersection with the x-axis of a line tangent to f at $(x_n, f(x_n))$.

$$f'(x_n) = \frac{\Delta y}{\Delta x} = \frac{f(x_n) - 0}{x_n - x_{n+1}} .$$

Subgradient

Generalization of gradients to non-differentiable functions.

Definition

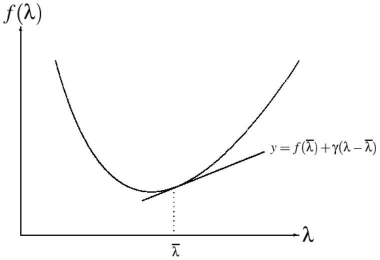
An m-vector γ is subgradient of $f(\lambda)$ at $\bar{\lambda}$ if

$$f(\lambda) \geq f(\bar{\lambda}) + \gamma(\lambda - \bar{\lambda})$$

The inequality says that the hyperplane

$$y = f(\bar{\lambda}) + \gamma(\lambda - \bar{\lambda})$$

is tangent to $y = f(\lambda)$ at $\lambda = \bar{\lambda}$ and supports $f(\lambda)$ from below



Proposition Given a choice of nonnegative multipliers $\bar{\lambda}$. If x' is an optimal solution to $z_{LR}(\lambda)$ then

$$\gamma = d - Dx'$$

is a subgradient of $z_{LR}(\lambda)$ at $\lambda = \bar{\lambda}$.

Proof We wish to prove that from the subgradient definition:

$$\max_{Ax \leq b} (cx - \lambda(Dx - d)) \geq \max_{Ax \leq b} \left(cx - \bar{\lambda}(Dx - d)\right) + \gamma(\lambda - \bar{\lambda})$$

Using that x' is an opt. solution to $f(\bar{\lambda})$ and the definition of γ we get:

$$\begin{aligned}\max_{Ax \leq b} (cx - \lambda(Dx - d)) &\geq (cx' - \bar{\lambda}(Dx' - d)) + (d - Dx')(\lambda - \bar{\lambda}) \\ &= cx' - \lambda(Dx' - d)\end{aligned}$$

Intuition

Lagrange dual:

$$\begin{aligned} \min\ & z_{LR}(\lambda) = cx - \lambda(Dx - d) \\ \text{s.t.}\ & Ax \leq b \\ & x \in \mathbb{Z}_+^n \end{aligned}$$

Gradient in x' is

$$\gamma = d - Dx'$$

Subgradient Iteration

Recursion

$$\lambda^{k+1} = \max\left\{\lambda^k - \theta\gamma^k, 0\right\}$$

where $\theta > 0$ is step-size

If $\gamma > 0$ and θ is sufficiently small $z_{LR}(\lambda)$ will decrease.

- Small θ slow convergence
- Large θ unstable

Held and Karp

Initially

$$\lambda^{(0)} = \{0, \ldots, 0\}$$

compute the new multipliers by recursion

$$\lambda_i^{(k+1)} := \begin{cases} \lambda_i^{(k)} & \text{if } |\gamma_i| \leq \varepsilon \\ \max(\lambda_i^{(k)} - \theta\gamma_i, 0) & \text{if } |\gamma_i| > \varepsilon \end{cases}$$

where γ is subgradient.

The step size θ is defined by

$$\theta = \mu \frac{\overline{z} - \underline{z}}{\sum_i \gamma_i^2}$$

where μ is an appropriate constant.
E.g. $\mu = 1$ and halved if upper bound not decreased in 20 iterations

Lagrange relaxation and LP

For an LP-problem where we Lagrange relax all constraints

- Dual variables are best choice of Lagrange multipliers
- Lagrange relaxation and LP "relaxation" give same bound

Gives a clue to solve LP-problems without Simplex

- Iterative algorithms
- Polynomial algorithms

Outline

Dantzig-Wolfe Decomposition

Motivation: Large difficult IP models

$\Longrightarrow$ split them up into smaller pieces

Applications

- Cutting Stock problems
- Multicommodity Flow problems
- Facility Location problems
- Capacitated Multi-item Lot-sizing problem
- Air-crew and Manpower Scheduling
- Vehicle Routing Problems
- Scheduling

Leads to methods also known as:

- Branch-and-price (column generation + branch and bound)
- Branch-and-cut-and-price (column generation + branch and bound + cutting planes)

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.

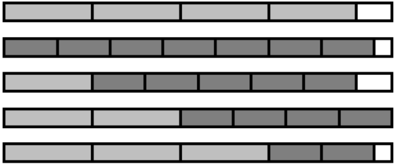
Motivation: Cutting stock problem

- Infinite number of raw stocks, having length L.
- Cut m piece types i, each having width w_i and 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



Formulation 1

$$
\begin{array}{ll}
\text{minimize} & u_1 + u_2 + u_3 + u_4 + u_5 \\
\text{subject to} & 5x_{11} + 3x_{12} \leq 22u_1 \\
& 5x_{21} + 3x_{22} \leq 22u_2 \\
& 5x_{31} + 3x_{32} \leq 22u_3 \\
& 5x_{41} + 3x_{42} \leq 22u_4 \\
& 5x_{51} + 3x_{52} \leq 22u_5 \\
& x_{11} + x_{21} + x_{31} + x_{41} + x_{51} \geq 7 \\
& x_{12} + x_{22} + x_{32} + x_{42} + x_{52} \geq 3 \\
& u_j \in \{0, 1\} \\
& x_{ij} \in \mathbb{Z}_+
\end{array}
$$

LP-relaxation gives solution value $z = 2$ with

$$u_1 = u_2 = 1, x_{11} = 2.6, x_{12} = 3, x_{21} = 4.4$$

Block structure

$$
\boxed{
\begin{array}{lccccccc}
\text{min} & u_1 & +u_2 & +u_3 & +u_4 & +u_5 & \\
\text{s.t} & x_{11} & +x_{21} & +x_{31} & +x_{41} & +x_{51} & \geq 7 \\
& x_{12} & +x_{22} & +x_{32} & +x_{42} & +x_{51} & \geq 3 \\
& 5x_{11} + 3x_{12} - 22u_1 & & & & & \leq 0 \\
& & 5x_{21} + 3x_{22} - 22u_2 & & & & \leq 0 \\
& & & 5x_{31} + 3x_{32} - 22u_3 & & & \leq 0 \\
& & & & 5x_{41} + 3x_{42} - 22u_4 & & \leq 0 \\
& & & & & 5x_{51} + 3x_{52} - 22u_5 & \leq 0
\end{array}
}
$$

Formulation 2

The matrix A contains all different cutting patterns
All (undominated) 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-relaxation gives solution value $z = 2.125$ with

$$\lambda_1 = 1.375, \lambda_4 = 0.75$$

Due to integer property a lower bound is $\lceil 2.125 \rceil = 3$.
Optimal solution value is $z^* = 3$.

Round up LP-solution getting heuristic solution $z_H = 3$.

Decomposition

If model has "block" structure

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

Lagrangian relaxation

Objective becomes

$$
\begin{array}{l}
c^1x^1 + c^2x^2 + \ldots + c^Kx^K \\
-\lambda\left(A^1x^1 + A^2x^2 + \ldots + A^Kx^K - b\right)
\end{array}
$$

Decomposed into

$$
\begin{array}{lcccccl}
\max & c^1x^1 - \lambda A^1x^1 + & c^2x^2 - \lambda A^2x^2 + & \ldots + & c^Kx^K - \lambda A^Kx^K + & b & \\
\text{s.t.} & D^1x^1 & + & & & & \leq d_1 \\
& & + & D^2x^2 & & & \leq d_2 \\
& & & \ldots & & & \leq \vdots \\
& & & & D^Kx^K & & \leq d_K \\
& x^1 \in \mathbb{Z}_+^{n_1} & & x^2 \in \mathbb{Z}_+^{n_2} & \ldots & x^K \in \mathbb{Z}_+^{n_K} &
\end{array}
$$

Model is separable

Dantzig-Wolfe decomposition

If model has "block" structure

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

Describe each set X^k, $k = 1, \ldots, K$

$$
\begin{array}{lcccccc}
\max & c^1x^1 & + & c^2x^2 & + \ldots + & c^Kx^K & \\
\text{s.t.} & A^1x^1 & + & A^2x^2 & + \ldots + & A^Kx^K & = b \\
 & x^1 \in X^1 & & x^2 \in X^2 & \ldots & x^K \in X^K &
\end{array}
$$

where $X^k = \{x^k \in \mathbb{Z}_+^{n_k} : D^kx^k \leq d_k\}$

Assuming that X^k has finite number of points $\{x^{k,t}\}$ $t \in T_k$

$$
X^k = \left\{ \begin{array}{ll} x^k \in \mathbb{R}^{n_k} : & x^k = \sum_{t \in T_k} \lambda_{k,t} x^{k,t}, \\ & \sum_{t \in T_k} \lambda_{k,t} = 1, \\ & \lambda_{k,t} \in \{0,1\}, t \in T_k \end{array} \right\}
$$

Dantzig-Wolfe decomposition

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

$$\max c^1\Big(\sum_{t\in T_1}\lambda_{1,t}x^{1,t}\Big)+c^2\Big(\sum_{t\in T_2}\lambda_{2,t}x^{2,t}\Big)+\ldots+c^K\Big(\sum_{t\in T_K}\lambda_{K,t}x^{K,t}\Big)$$

$$\text{s.t.}\quad A^1\Big(\sum_{t\in T_1}\lambda_{1,t}x^{1,t}\Big)+A^2\Big(\sum_{t\in T_2}\lambda_{2,t}x^{2,t}\Big)+\ldots+A^K\Big(\sum_{t\in T_K}\lambda_{K,t}x^{K,t}\Big)=b$$

$$\sum_{t\in T_k}\lambda_{k,t}=1 \qquad k=1,\ldots,K$$

$$\lambda_{k,t}\in\{0,1\}, \qquad t\in T_k \quad k=1,\ldots,K$$

Strength of linear master model

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

$$\begin{array}{llllllll} \max & c^1x^1 & + & c^2x^2 & +\ldots+ & c^kx^k & \\ \text{s.t.} & A^1x^1 & + & A^2x^2 & +\ldots+ & A^kx^k & =b \\ & x^1 \in \operatorname{conv}(X^1) & & x^2 \in \operatorname{conv}(X^2) & \ldots & x^k \in \operatorname{conv}(X^k) & \end{array}$$

Proof: Consider LP-relaxation

$$\max c^1(\sum_{t\in T_1} \lambda_{1,t}x^{1,t}) + c^2(\sum_{t\in T_2} \lambda_{2,t}x^{2,t}) + \ldots + c^K(\sum_{t\in T_K} \lambda_{K,t}x^{K,t})$$

$$\text{s.t.}\ \ A^1(\sum_{t\in T_1} \lambda_{1,t}x^{1,t}) + A^2(\sum_{t\in T_2} \lambda_{2,t}x^{2,t}) + \ldots + A^K(\sum_{t\in T_K} \lambda_{K,t}x^{K,t}) = b$$

$$\sum_{t\in T_k} \lambda_{k,t} = 1 \qquad\qquad k = 1,\ldots,K$$

$$\lambda_{k,t} \geq 0, \qquad t \in T_k \qquad k = 1,\ldots,K$$

Informally speaking we have

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

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

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

Using result next page

$$\begin{array}{llllllll}
\max & c^1x^1 & + & c^2x^2 & +\ldots+ & c^kx^k & & \\
\text{s.t.} & A^1x^1 & + & A^2x^2 & +\ldots+ & A^kx^k & = & b \\
& x^1 \in \text{conv}(X^1) & & x^2 \in \text{conv}(X^2) & \ldots & x^k \in \text{conv}(X^k) & &
\end{array}$$

Strength of Lagrangian Relaxation (section 10.2)

Integer Programming Problem

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

Lagrange Relaxation, multipliers $\lambda \ge 0$

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

for best multiplier $\lambda \ge 0$

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

Outline

Delayed Column Generation

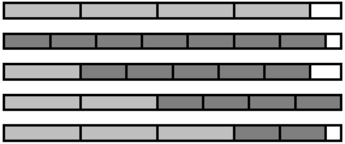
Delayed column generation, linear master

- 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

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

LP-problem

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

where

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

Revised Simplex Method

- $\max\{cx \mid Ax \leq b, x \geq 0\}$
- $B = \{1 \ldots m\}$ basic variables
- $N = \{m+1 \ldots m+n\}$ non-basic variables (will be set to lower bound 0)
- $A_B = [A_1 \ldots A_m]$
- $A_N = [A_{m+1} \ldots A_{m+n}]$

Standard form

$$\left[\begin{array}{c:c:c:c} A_B & A_N & 0 & b \\ \hdashline c_B & c_N & 1 & 0 \end{array}\right]$$

$$
\begin{aligned}
Ax &= A_N x_N + A_B x_B = b \\
A_B x_B &= b - A_N x_N \\
x_B &= A_B^{-1} b - A_B^{-1} A_N x_N
\end{aligned}
$$

basic feasible solution:

- $X_N = 0$
- A_B lin. indep.
- $X_B \geq 0$

$$
\begin{aligned}
z = c^T x &= c_B^T (A_B^{-1} b - A_B^{-1} A_N x_N) + c_N^T x_N = \\
&= c_B^T A_B^{-1} b + (c_N^T - c_B^T A_B^{-1} A_N) x_N
\end{aligned}
$$

Canonical form

$$
\left[\begin{array}{c:c:c:c}
I & A_B^{-1} A_N & 0 & A_B^{-1} b \\
\hdashline
0 & c_N^T - c_B^T A_B^{-1} A_N & 1 & -c_B^T A_B^{-1} b
\end{array}\right]
$$

In scalar form: the objective function is obtained by multiplying and subtracting constraints by means of multipliers π: $\pi = c_B^T A_B^{-1}$ (the dual variables)

$$z = \sum_{j=1}^{m} \left[c_j - \sum_{i=1}^{m} \pi_i a_{ij} \right] x_j + \sum_{j=m+1}^{m+n} \left[c_j - \sum_{i=1}^{m} \pi_i a_{ij} \right] x_j + \sum_{i=1}^{m} \pi_i b_i$$

Each basic variable has cost null in the objective function

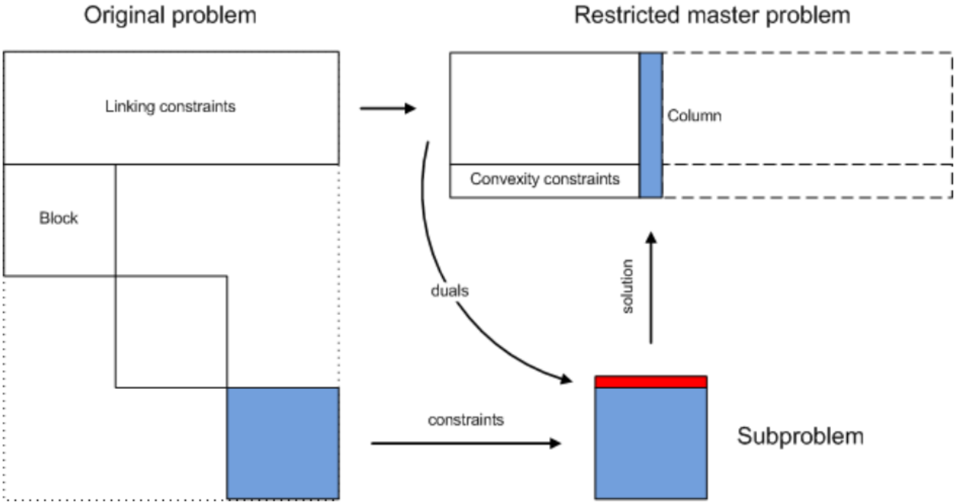
$$c_j - \sum_{i=1}^{m} \pi_i a_{ij} = 0 \qquad j = 1, \dots, m$$

Reduced costs of non-basic variables:

$$\bar{c}_j = c_j - \sum_{i=1}^{m} \pi_i a_{ij} \qquad j = m+1, \dots, m+n$$

If basis is optimal then $\bar{c}_j \leq 0$ for all $j = m+1, \dots, m+n$.

Dantzig Wolfe Decomposition with Column Generation



Delayed column generation (example)

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

Initially we choose only the trivial cutting patterns

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

Solve LP-problem

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

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.

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

Small example (continued)

Find entering variable

$$\begin{array}{rcl} A & = & \begin{pmatrix} 1 & 2 & 3 & \cdots \\ 5 & 4 & 2 & \cdots \end{pmatrix} \qquad \begin{array}{l} \frac{1}{4} \leftarrow y_1 \\ \frac{1}{7} \leftarrow y_2 \end{array} \\ c_N - yA_N & = & \begin{pmatrix} 1 - \frac{27}{28} & 1 - \frac{30}{28} & 1 - \frac{29}{28} & \cdots \end{pmatrix} \end{array}$$

We could also solve optimization problem

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

which is equivalent to knapsack problem

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

This problem has optimal solution $x_1 = 2, x_2 = 4$.
Reduced cost of entering variable

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

Small example (continued)

Add new cutting pattern to A getting

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

Solve problem to LP-optimality, getting primal solution

$$x_1 = \frac{5}{8}, x_3 = \frac{3}{2}$$

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

$$1 - 4\frac{1}{4} - 0\frac{1}{7} = 0$$

Terminate with $x_1 = \frac{5}{8}$, $x_3 = \frac{3}{2}$, and $z_{LP} = \frac{17}{8} = 2.125$.

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 function is improved. We know the best solution among existing columns. If we generate an already existing column, then we will not improve the objective.

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

Dual Bounds

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

$$\sum_{j \in J} \lambda_j \leq \kappa$$

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):

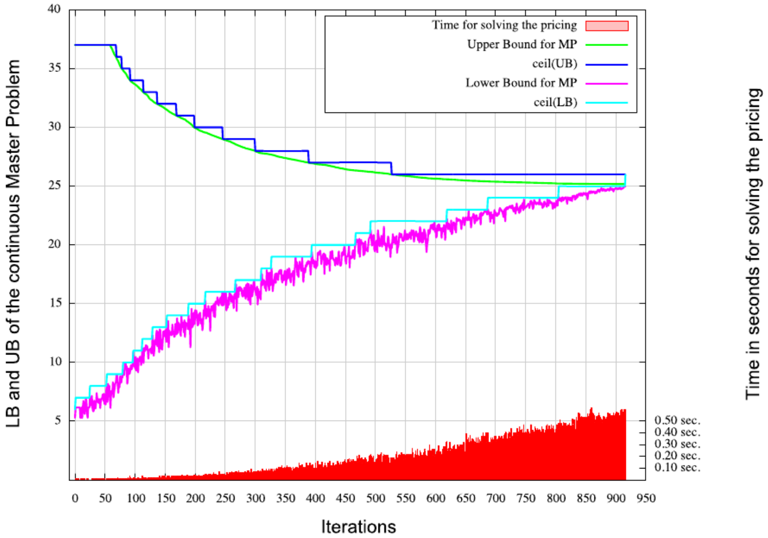
$$z_{LRMP} + \kappa z_{PP} \leq z_{LMP}$$

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

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

Convergence in CG

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



Row and Column Generation

In problems with many rows we can generate them like done in column generation.

Cutting plane methods where the pricing problem is the separation problem.

Combining the two: column generation cannot ignore the missing rows. Existing approaches are problem specific.

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
- Often column generation referred to as branch-and-price

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.

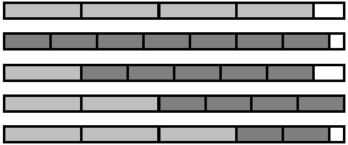
Branch-and-price

- 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

Branch-and-price, example

The matrix A contains all different cutting patterns

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



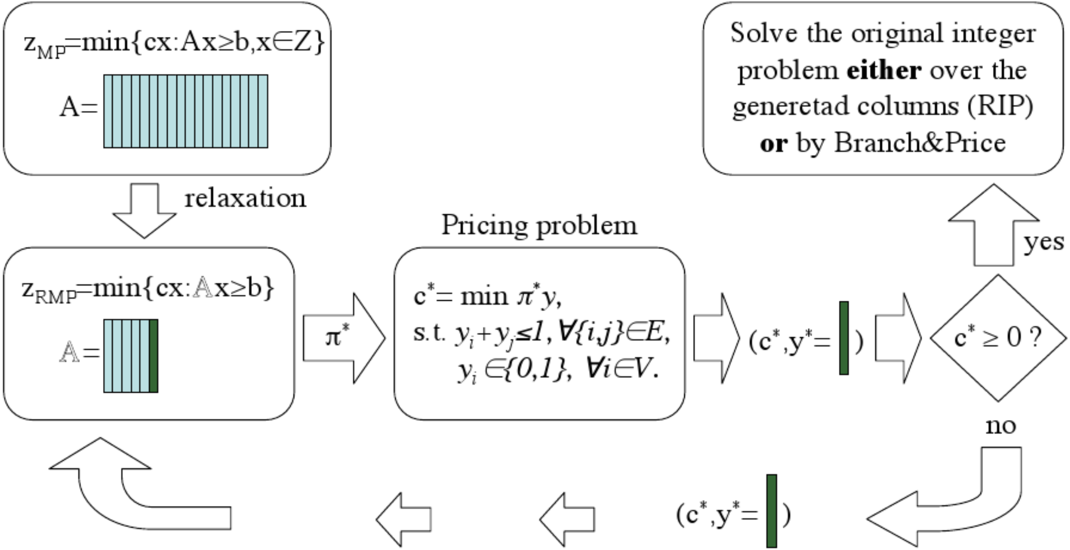
Problem

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

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



z_{MP}=min{cx:Ax≥b,x∈Z}
A=
relaxation
z_{RMP}=min{cx:𝔸x≥b}
𝔸=
π^*
Pricing problem
$c^* = \min \pi^* y$,
s.t. $y_i + y_j \leq 1, \forall \{i,j\} \in E$,
$y_i \in \{0,1\}, \forall i \in V$.
(c^*,y^*=)
$c^* \geq 0$?
yes
no
Solve the original integer problem **either** over the generetad columns (RIP) **or** by Branch&Price
(c^*,y^*=)

Heuristic solution (eg, in sec. 12.6)

- 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

References

Fisher M.L. (2004). **The Lagrangian relaxation method for solving integer programming problems**. *Management Science*, 50(12), pp. 1861–1871. This article originally appeared in Management Science, January 1981, Volume 27, Number 1, pp. 1-18, published by The Institute of Management Sciences.

Lübbecke (2010). **Column Generation**. John Wiley & Sons, Inc.