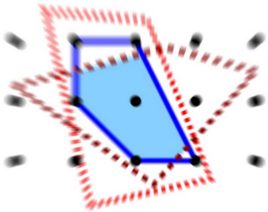
Welcome!


Sohaib Afifi
@sohaibafifi
Follow
To #paris for a week full of columns. ping @mluebbecke
Reply
Retweeted
Favorite
More
RETWEET
1
6:33 AM - 10 Mar 2014

School on Column Generation 2014

The Basics

Marco Lübbecke · Aachen, Germany



@mluebbecke OperationsResearchRWTH

Operations Research | RWTH AACHEN UNIVERSITY

What to read?

OPERATIONS RESEARCH
Vol. 53, No. 6, November–December 2005, pp. 1007–1023
ISSN 0030-364X | EISSN 1526-5463 | 05 | 5306 | 1007

informs
DOI 10.1287/opre.1050.0234

Selected Topics in Column Generation

Marco E. Lübbecke
Technische Universität Berlin, Institut für Mathematik, Sekr. MA 6-1, Straße des 17. Juni 136, D-10623 Berlin, Germany, m.luebbecke@math.tu-berlin.de

Jacques Desrosiers
HEC Montréal and GERAD, 3000, chemin de la Côte-Sainte-Catherine, Montréal, Québec, Canada H3T 2A7, jacques.desrosiers@hec.ca

Dantzig-Wolfe decomposition and column generation, devised for linear programs, is a success story in large-scale integer programming. We outline and relate the approaches, and survey mainly recent contributions, not yet found in textbooks. We emphasize the growing understanding of the dual point of view, which has brought considerable progress to the column generation theory and practice. It stimulated careful initializations, sophisticated solution techniques for the restricted master problem and subproblem, as well as better overall performance. Thus, the dual perspective is an ever recurring concept in our "selected topics."

Subject classifications: integer programming: column generation, Dantzig-Wolfe decomposition, Lagrangian relaxation, branch-and-bound; linear programming: large scale systems.
Area of review: Optimization.
History: Received December 2002; revision received March 2004; accepted October 2004.

BRANCH-PRICE-AND-CUT ALGORITHMS

JACQUES DESROSIERS
HEC Montréal and GERAD, Chemin de la Côte-Sainte-Catherine, Montréal Québec, Canada

MARCO E. LÜBBECKE
Fachbereich Mathematik, Technische Universität Darmstadt, Darmstadt, Germany

Decompositions and reformulations of mixed integer programs are classical approaches to obtaining stronger relaxations and reduce symmetry. These often entail the dynamic addition of variables (columns) and/or constraints (cutting planes) to the model. When the linear relaxation in each node of a branch-and-bound tree is solved by column generation, one speaks of branch-and-price. Optionally, as in standard branch-and-bound, cutting planes can be added in order to strengthen the relaxation, and this is problem and its extended reformulation, as first used in Desrosiers *et al.* [2].

There are very successful applications of branch-and-price in industry (see Desrosiers and Lübbecke [3], and also the section titled "Vehicle Routing and Scheduling" in this encyclopedia) and also to generic combinatorial optimization problems like bin packing and the cutting stock problem [4], graph coloring [5], machine scheduling [6], the p-median problem [7], the generalized assignment problem [8], and many others. The method today is an indispensable part of the integer programming toolbox.

COLUMN GENERATION

Consider the following *integer master problem*

$$\min \sum_{j\in J} c_j\lambda_j$$
$$\text{subject to } \sum_{j\in J} \mathbf{a}_j\lambda_j \leq \mathbf{b} \qquad (1)$$
$$\boldsymbol{\lambda} \in \mathbb{Z}_+^{|J|}.$$

Chapter 1

A PRIMER IN COLUMN GENERATION

Jacques Desrosiers
Marco E. Lübbecke

Abstract We give a didactic introduction to the use of the column generation technique in linear and in particular in integer programming. We touch on both, the relevant basic theory and more advanced ideas which help in solving large scale practical problems. Our discussion includes embedding Dantzig-Wolfe decomposition and Lagrangian relaxation within a branch-and-bound framework, deriving natural branching and cutting rules by means of a so-called compact formulation, and understanding and influencing the behavior of the dual variables during column generation. Most concepts are illustrated via a small example. We close with a discussion of the classical cutting stock problem and some suggestions for further reading.

COLUMN GENERATION

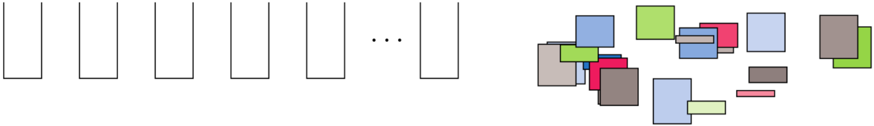
MARCO E. LÜBBECKE
Fachbereich Mathematik, Technische Universität Darmstadt, Darmstadt, Germany

Column generation is a classical technique to solve a mathematical program by iteratively adding the variables of the model [1]. Typically, only a tiny fraction of the variables is needed to prove optimality, which makes the technique interesting for problems with a huge number of variables. The method is often said in one sentence with Dantzig–Wolfe decomposition [2] (see also ***Dantzig–Wolfe Decomposition***), as it is

$$v(\text{MP}) := \min \sum_{j\in J} c_j\lambda_j$$
$$\text{subject to } \sum_{j\in J} \mathbf{a}_j\lambda_j \geq \mathbf{b}$$
$$\lambda_j \geq 0, \quad j \in J, \qquad (1)$$

with $|J| = n$ variables and m constraints. In many applications, n is exponential in m and working with Equation (1) explicitly is not an option because of its sheer size. Instead, consider the *restricted master problem* (RMP), which contains only a subset $J' \subseteq J$ of variables. An optimal solution $\boldsymbol{\lambda}^*$ to the RMP need not be optimal for the MP, of course. Denote an optimal dual solution to the RMP by $\boldsymbol{\pi}^*$. In the *pricing step* of the simplex method (see also ***Simplex Method and Complexity***),

Example: The Bin Packing Problem

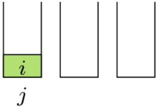


- n items of size a_i, $i = 1, \ldots, n$
- bins of capacity b
- pack all items into as few bins as possible

The Bin Packing Problem

$$
\begin{aligned}
\min \sum_{\text{bins } j} y_j & & \\
\sum_{\text{bins } j} x_{ij} &= 1 & \text{items } i \\
\sum_{\text{items } i} a_i x_{ij} &\leq b & \text{bins } j \\
x_{ij} &\leq y_j & i, j \\
x_{ij},\, y_j &\in \{0, 1\} & i, j
\end{aligned}
$$

$x_{ij} \sim$

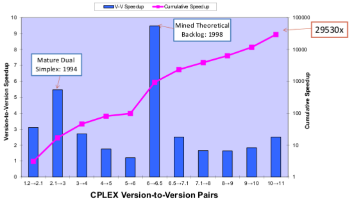


Algorithms for Solving Integer Programs

- very effective algorithms available
- industry strength implementations available (“solvers”)
- development 1991–2012
  - computer speedup: factor 2 000
  - algorithmic speedup: factor 500 000

$\Rightarrow$ can solve problems with 10^6 variables and 10^5 constraints

Speedups 1991-2007



Performance of this first Model

(poor)

Remarks on the first Model

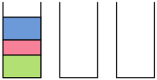
- single decision (variable) often does not matter much
- often “distributed decision making,” here: for every bin
- local decisions “similar” and often easy, here: knapsack
- globally coordinated/synchronized by a central authority

- a standard solver does not “see” this concert

A Model with “more meaningful” Variables

- $P_j \sim$ *all* possible patterns to fill bin j

$$
\begin{aligned}
\min \sum_{\text{bins } j} \sum_{p \in P_j} \lambda_{pj} & & \\
\sum_{\text{bins } j} \sum_{p \in P_j : i \in p} \lambda_{pj} &= 1 & \text{items } i \\
\sum_{p \in P_j} \lambda_{pj} &\leq 1 & \text{bins } j \\
\lambda_{pj} &\in \{0, 1\} & j = 1, \ldots, n,\ p \in P_j
\end{aligned}
$$

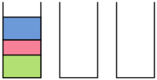
$\lambda_{pj} \sim$ 

j

The "most natural" Model?

- $P \sim$ *all* possible patterns

$$
\begin{array}{lllll}
\min & \displaystyle\sum_{p \in P} \lambda_p & & & \\
& \displaystyle\sum_{p \in P\, : i \in p} \lambda_p & = 1 & \text{items } i & \\
& \displaystyle\sum_{p \in P} \lambda_p & \leq n & \text{(not really needed)} & \\
& \lambda_p & \in \{0,1\} & & p \in P
\end{array}
$$

$\lambda_p \sim$ 

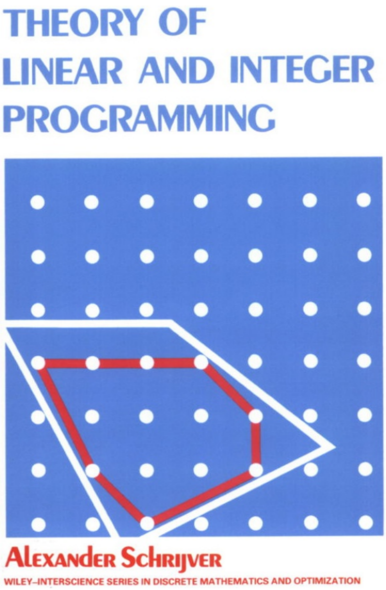
Performance of the “most natural” Model

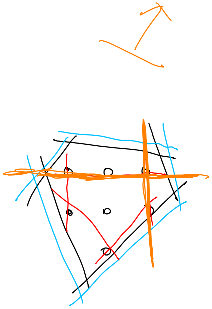
(very well)

If nothing else, take away this Message

✎ in integer programming, a “good” model is crucial

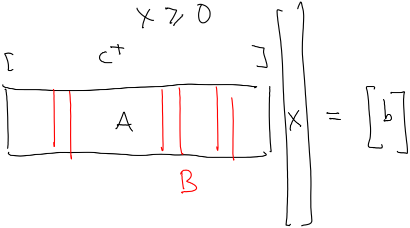
The Integer Programming Dilemma

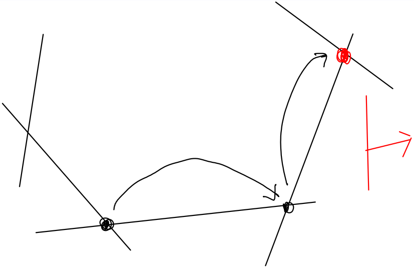




Simplex Method

$$\min \quad c^T x$$
$$Ax = b$$
$$x \geq 0$$





$$A_B x_B = b$$
$$x_N = 0$$

basic solution

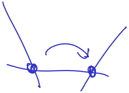
$\Updownarrow$

extreme point solutions

Simplex Method

$$A_B x_B + A_N x_N = b \quad \Big| \cdot A_B^{-1}$$

$$\rightarrow \quad x_B = \underbrace{A_B^{-1} b}_{\bar{b}} - \underbrace{\underbrace{A_B^{-1} A_N}\, \underbrace{x_N}_{=0}}_{=0}$$



$$c_B^T x_B + c_N^T x_N$$

$$c_B^T A_B^{-1} b - c_B^T A_B^{-1} A_N x_N + c_N^T x_N$$

$$c_B^T A_B^{-1} b + \underbrace{\Big(c_N^T - \overbrace{\boxed{c_B^T A_B^{-1}}}^{\pi} A_N \Big)}_{\bar{c}_N^T} x_N$$

„reduced cost"

optimal when $\bar{c}_N \geq 0$

Column Generation

$$\begin{aligned} \min \quad & c^T \lambda \\ \text{s.t.} \quad & A\lambda \geq b \\ & \lambda \geq 0 \end{aligned}$$

$\lambda_j \,,\ j \in J$

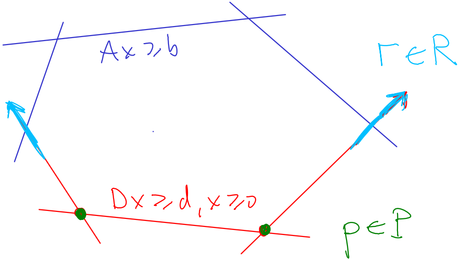
idea: instead of using J, start with subset $J' \subseteq J$

check $J \setminus J'$ for neg. red. cost.

$$\min_{j \in J \setminus J'} \left[c_j - \pi^T A_j \right]$$

Dantzig-Wolfe Reformulation of an LP (1960)

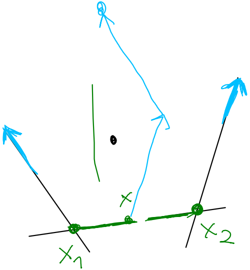
$$
\begin{aligned}
\min \quad & c^T x \\
\text{s.t.} \quad & Ax \geq b \\
& Dx \geq d \\
& x \geq 0
\end{aligned}
$$



$$x = \sum_{p \in P} \lambda_p x_p + \sum_{r \in R} \lambda_r x_r$$

$$\sum_{p \in P} \lambda_p = 1$$

$$\lambda_p, \quad \lambda_r \geq 0$$



$$x = \lambda_1 x_1 + \lambda_2 x_2$$

$$\lambda_1 + \lambda_2 = 1, \quad \lambda_1, \lambda_2 \geq 0$$

Dantzig-Wolfe Reformulation of an LP

$$\min \quad c^T\left[\sum_{p\in P}\lambda_p x_p + \sum_{r\in R}\lambda_r x_r\right]$$

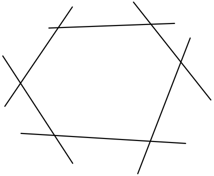
$$\text{s.t.} \quad A\left[\sum_{p\in P}\lambda_p x_p + \sum_{r\in R}\lambda_r x_r\right] \geq b$$

$$\sum_{p\in P}\lambda_p = 1$$ ← „convexity constraint"

$$\lambda_p, \quad \lambda_r \geq 0$$

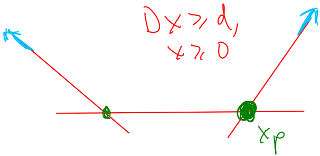
$$c^T x_p =: c_p$$

$$A x_p =: a_p$$



Dantzig-Wolfe Reformulation of an LP

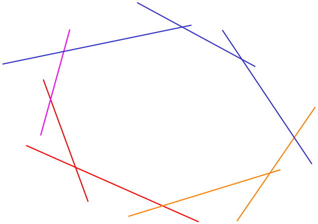
identifying an extr. point x_p of neg. red. cost.

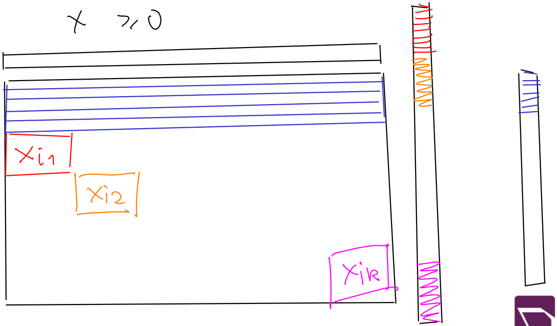


$$\min \; (c^T - \pi^T A)x$$
$$Dx \geq d$$
$$x \geq 0$$

Dantzig-Wolfe Reformulation of an LP

$$
\begin{aligned}
\min \quad & c^T x \\
\text{s.t.} \quad & A x \geq b \\
& D_1 x \geq d_1 \\
& D_2 x \geq d_2 \\
& \quad \vdots \\
& D_k x \geq d_k \\
& x \geq 0
\end{aligned}
$$

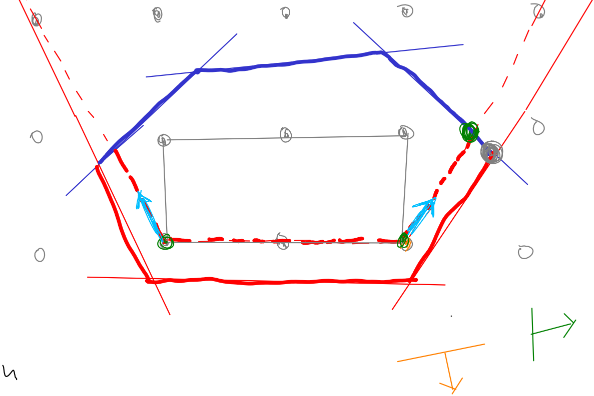




„block-diagonal structure"

Dantzig-Wolfe Reformulation of an IP

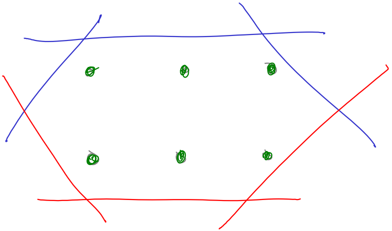
$$\min \quad c^T x$$
$$\text{s.t.} \quad Ax \geq b$$
$$Dx \geq d$$
$$x \in \mathbb{Z}^n_+$$



convexification

pricing problem: $\min \quad (c^T - \pi^T A)x$
$$Dx \geq d$$
$$x \in \mathbb{Z}^n_+$$

Dantzig-Wolfe Reformulation of an IP



„discretization"

$$x = \sum_{p \in P} \lambda_p x_p + \sum_{r \in R} \lambda_r x_r$$

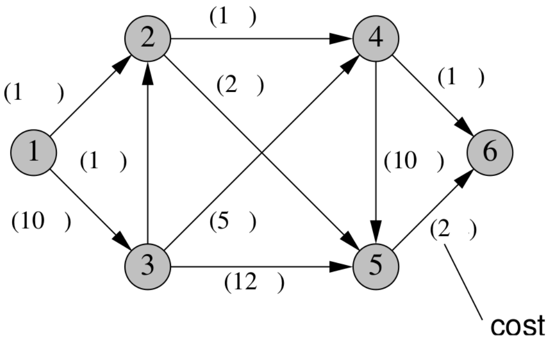
$$\sum_{p \in P} \lambda_p = 1$$

$$\lambda_p \in \{0,1\}$$

$$\lambda_r \in \mathbb{Z}_+^n$$

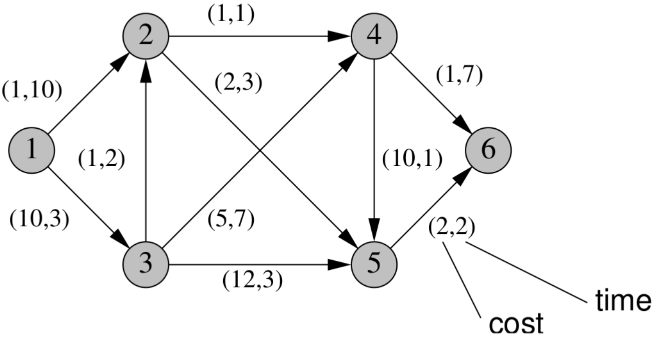
Example: Taken from the “Primer”

Find: shortest path from 1 to 6



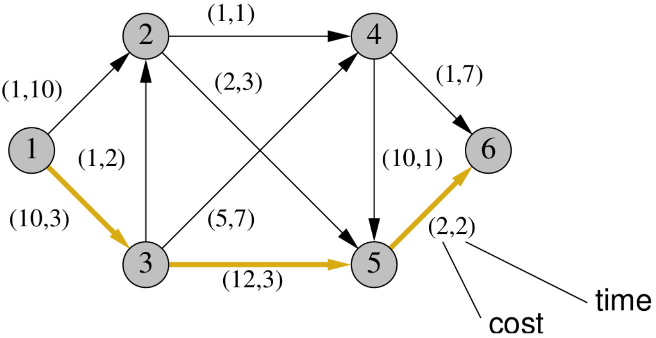
Example: Taken from the “Primer”

Find: Resource constrained shortest path from 1 to 6
Total traversal time must not exceed 14 units



Example: Taken from the "Primer"

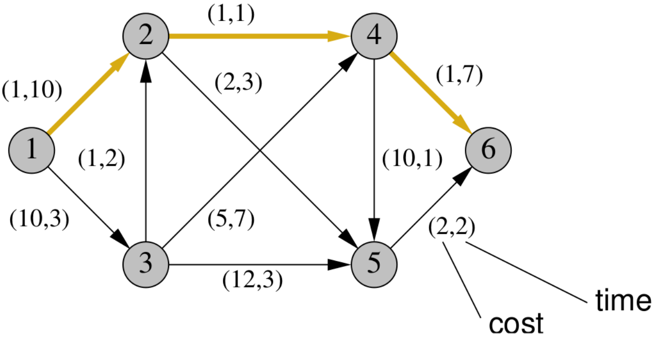
Find: Resource constrained shortest path from 1 to 6
Total traversal time must not exceed 14 units



Path 1-3-5-6 is quick but expensive: cost 24, time 8

Example: Taken from the “Primer”

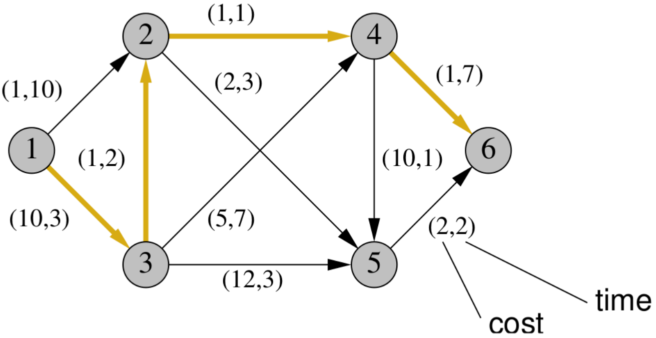
Find: Resource constrained shortest path from 1 to 6
Total traversal time must not exceed 14 units



Path 1-2-4-6 is cheap but too slow: cost 3, time 18

Example: Taken from the “Primer”

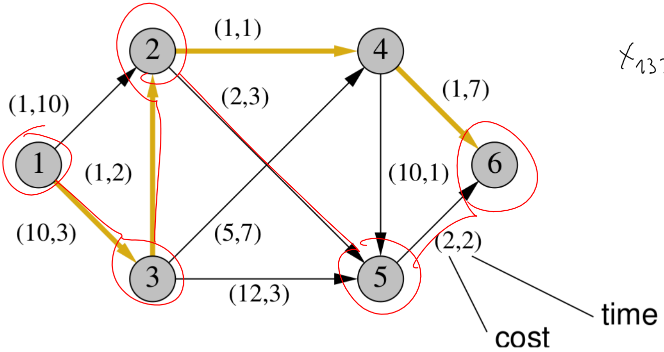
Find: Resource constrained shortest path from 1 to 6
Total traversal time must not exceed 14 units



Path 1-3-2-4-6 is optimal: cost 13, time 13

Example: Taken from the “Primer”

Find: Resource constrained shortest path from 1 to 6
Total traversal time must not exceed 14 units



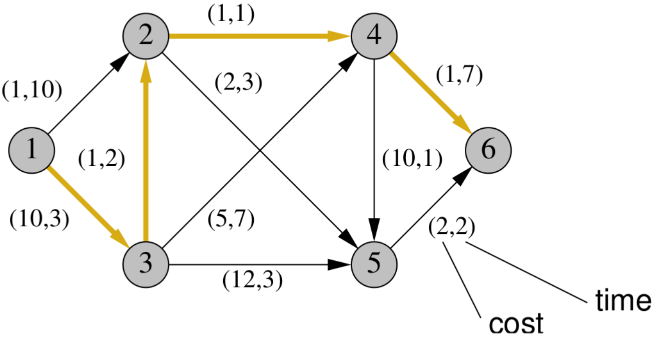
$$x_{13256} = \begin{pmatrix} x_{12} \\ x_{13} \\ x_{32} \\ x_{24} \\ x_{25} \\ x_{34} \\ x_{35} \\ x_{45} \\ x_{46} \\ x_{56} \end{pmatrix} = \begin{pmatrix} 0 \\ 1 \\ 1 \\ 0 \\ 1 \\ 0 \\ 0 \\ 0 \\ 0 \\ 1 \end{pmatrix}$$

Path 1-3-2-4-6 is optimal: cost 13, time 13

How do we find out?

Example: Taken from the “Primer”

Find: Resource constrained shortest path from 1 to 6
Total traversal time must not exceed 14 units



Path 1-3-2-4-6 is optimal: cost 13, time 13
How do we find out? *This is why we are here...*

Integer Program (IP)

The problem is (weakly) $\mathcal{NP}$-hard
This justifies a solution approach via integer programs

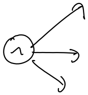
Close enough to “real-world” problems to see almost the whole column generation mechanism at work

Integer Program (IP)

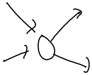
c_{ij} cost on arc (i,j), t_{ij} time to traverse (i,j)

$$z^\star := \min \sum_{(i,j)\in A} c_{ij}x_{ij}$$

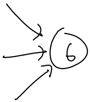
$$\text{subject to} \quad \sum_{j:(1,j)\in A} x_{1j} = 1$$



$$\sum_{j:(i,j)\in A} x_{ij} - \sum_{j:(j,i)\in A} x_{ji} = 0 \qquad i = 2,3,4,5$$



$$\sum_{i:(i,6)\in A} x_{i6} = 1$$



$$\sum_{(i,j)\in A} t_{ij}x_{ij} \le 14$$

$$x_{ij} \in \{0,1\} \quad (i,j)\in A$$

Operations Research

RWTH AACHEN UNIVERSITY

Integer Program (IP)

c_{ij} cost on arc (i,j), t_{ij} time to traverse (i,j)

$$z^\star := \min \sum_{(i,j)\in A} c_{ij}x_{ij}$$

$$\text{subject to} \sum_{j:(1,j)\in A} x_{1j} = 1$$

$$\sum_{j:(i,j)\in A} x_{ij} - \sum_{j:(j,i)\in A} x_{ji} = 0 \qquad i = 2,3,4,5$$

$$\sum_{i:(i,6)\in A} x_{i6} = 1$$

$$\sum_{(i,j)\in A} t_{ij}x_{ij} \leq 14$$

$$x_{ij} \in \{0,1\} \quad (i,j) \in A$$

Could be solved by branch-and-bound (B&B)

Integer Program (IP)

c_{ij} cost on arc (i,j), t_{ij} time to traverse (i,j)

$$z^\star := \min \sum_{(i,j)\in A} c_{ij}x_{ij}$$

subject to
$$\sum_{j:(1,j)\in A} x_{1j} = 1$$

$$\sum_{j:(i,j)\in A} x_{ij} - \sum_{j:(j,i)\in A} x_{ji} = 0 \qquad i = 2,3,4,5$$

$$\sum_{i:(i,6)\in A} x_{i6} = 1$$

$$\sum_{(i,j)\in A} \cancel{t_{ij}x_{ij}} \leq 14$$

$$x_{ij} \in \{0,1\} \quad (i,j)\in A$$

Instead: Exploit embedded shortest path problem

Paths vs. Arcs Formulation

What remains

$$\sum_{j:(1,j)\in A} x_{1j} = 1$$

$$\sum_{j:(i,j)\in A} x_{ij} - \sum_{j:(j,i)\in A} x_{ji} = 0 \qquad i = 2,3,4,5$$

$$\sum_{i:(i,6)\in A} x_{i6} = 1$$

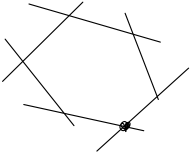
$$x_{ij} \in \{0,1\} \quad (i,j) \in A$$

defines a (particular) network flow problem

Fact: Every flow defined on arcs decomposes into flows on paths (and cycles) Ahuja, Magnanti & Orlin (1993)

Paths vs. Arcs Formulation

The convex hull of



$$\sum_{j:(1,j)\in A} x_{1j} = 1$$

$$\sum_{j:(i,j)\in A} x_{ij} - \sum_{j:(j,i)\in A} x_{ji} = 0 \qquad i = 2,3,4,5$$

$$\sum_{i:(i,6)\in A} x_{i6} = 1$$

$$x_{ij} \in \{0,1\} \quad (i,j) \in A$$

defines a polyhedron with integer vertices

Fact: Every (fractional) flow can be represented as convex combination of path (and cycle) flows

Paths vs. Arcs Formulation

Fact: Every (fractional) flow can be represented as convex combination of path (and cycle) flows

$$x_{ij} = \sum_{p \in P} x_{pij} \lambda_p \quad (i,j) \in A$$

$$\sum_{p \in P} \lambda_p = 1$$

$$\lambda_p \geq 0 \quad p \in P$$

P is the set of *all* paths from node 1 to node 6

$x_{pij} = 1$ iff edge (i,j) on path p, otherwise $x_{pij} = 0$

$\sum_{p \in P} \lambda_p = 1$ is called *convexity constraint*

Master Problem

Now substitute for x_{ij} in our original IP...

$$\begin{aligned} \sum_{p \in P} \lambda_p &= 1 && \\ \lambda_p &\geq 0 && p \in P \\ \sum_{p \in P} x_{pij} \lambda_p &= x_{ij} && (i,j) \in A \end{aligned}$$

Master Problem

Now substitute for x_{ij} in our original IP...

$$
\begin{aligned}
z^\star = \min \sum_{p \in P} \Big(\sum_{(i,j) \in A} c_{ij} x_{pij} \Big) \lambda_p & & \\
\text{subject to } \sum_{p \in P} \Big(\sum_{(i,j) \in A} t_{ij} x_{pij} \Big) \lambda_p \leq 14 & & \\
\sum_{p \in P} \lambda_p = 1 & & \\
\lambda_p \geq 0 & \quad & p \in P \\
\sum_{p \in P} x_{pij} \lambda_p = x_{ij} & \quad & (i,j) \in A \\
x_{ij} \in \{0,1\} & \quad & (i,j) \in A
\end{aligned}
$$

Master Problem

Now substitute for x_{ij} in our original IP...

$$z^\star = \min \sum_{p\in P} \Big(\sum_{(i,j)\in A} c_{ij} x_{pij} \Big) \lambda_p$$

$$\text{subject to} \quad \sum_{p\in P} \Big(\sum_{(i,j)\in A} t_{ij} x_{pij} \Big) \lambda_p \leq 14$$

$$\sum_{p\in P} \lambda_p = 1$$

$$\lambda_p \geq 0 \qquad p \in P$$

$$\sum_{p\in P} x_{pij} \lambda_p = x_{ij} \qquad (i,j) \in A$$

$$x_{ij} \in \{0,1\} \quad (i,j) \in A$$

✎ keep in mind

Integrality must hold for original x_{ij} variables

Master Problem: Linear Relaxation

...and relax integrality constraint of x_{ij}

$$\bar{z}^\star = \min \sum_{p \in P} \Big(\sum_{(i,j) \in A} c_{ij} x_{pij} \Big) \lambda_p$$

subject to
$$\sum_{p \in P} \Big(\sum_{(i,j) \in A} t_{ij} x_{pij} \Big) \lambda_p \leq 14$$

$$\sum_{p \in P} \lambda_p = 1$$

$$\lambda_p \geq 0 \qquad p \in P$$

$$\sum_{p \in P} x_{pij} \lambda_p = x_{ij} \qquad (i,j) \in A$$

$$x_{ij} \geq 0 \qquad (i,j) \in A$$

Master Problem: Linear Relaxation

...and relax integrality constraint of x_{ij}

$$
\begin{aligned}
\bar{z}^\star = \min \sum_{p \in P} \Big(\sum_{(i,j) \in A} c_{ij} x_{pij} \Big) \lambda_p & \\
\text{subject to } \sum_{p \in P} \Big(\sum_{(i,j) \in A} t_{ij} x_{pij} \Big) \lambda_p \leq 14 & \\
\sum_{p \in P} \lambda_p = 1 & \\
\lambda_p \geq 0 & \qquad p \in P
\end{aligned}
$$

We can remove the link between x_{ij} and λ_p variables

Master Problem: Linear Relaxation

... and relax integrality constraint of x_{ij}

$$\bar{z}^\star = \min \sum_{p \in P} \Big(\sum_{(i,j) \in A} c_{ij} x_{pij} \Big) \lambda_p$$

$$\text{subject to} \quad \sum_{p \in P} \Big(\sum_{(i,j) \in A} t_{ij} x_{pij} \Big) \lambda_p \leq 14$$

$$\sum_{p \in P} \lambda_p = 1$$

$$\lambda_p \geq 0 \qquad p \in P$$

Where is "we are looking for a path"?

Master Problem: Linear Relaxation

That is: We would like to solve the linear program

$$\bar{z}^\star = \min \sum_{p \in P} \Big(\sum_{(i,j) \in A} c_{ij} x_{pij} \Big) \lambda_p$$

$$\text{subject to} \quad \sum_{p \in P} \Big(\sum_{(i,j) \in A} t_{ij} x_{pij} \Big) \lambda_p \leq 14$$

$$\sum_{p \in P} \lambda_p = 1$$

$$\lambda_p \geq 0 \qquad p \in P$$

Problem: Here we only have 9 paths; in general we will have gazillions of them (sometimes guyzillions) . . .

Master Problem: Linear Relaxation

That is: We would like to solve the linear program

$$\bar{z}^\star = \min \sum_{p \in P} \Big(\sum_{(i,j) \in A} c_{ij} x_{pij} \Big) \lambda_p$$

$$\text{subject to} \quad \sum_{p \in P} \Big(\sum_{(i,j) \in A} t_{ij} x_{pij} \Big) \lambda_p \leq 14$$

$$\sum_{p \in P} \lambda_p = 1$$

$$\lambda_p \geq 0 \qquad p \in P$$

Problem: Here we only have 9 paths; in general we will have gazillions of them (sometimes guyzillions) . . .

Master Problem: Linear Relaxation

Writing out all coefficients we need a tiny font size…

$$
\begin{array}{lrrrrrrrrrl}
\min & 3\lambda_{1246} + & 14\lambda_{12456} + & 5\lambda_{1256} + & 13\lambda_{13246} + & 24\lambda_{132456} + & 15\lambda_{13256} + & 16\lambda_{1346} + & 27\lambda_{13456} + & 24\lambda_{1356} & \\
\text{s.t.} & 18\lambda_{1246} + & 14\lambda_{12456} + & 15\lambda_{1256} + & 13\lambda_{13246} + & 9\lambda_{132456} + & 10\lambda_{13256} + & 17\lambda_{1346} + & 13\lambda_{13456} + & 8\lambda_{1356} & \leq 14 \\
& \lambda_{1246} + & \lambda_{12456} + & \lambda_{1256} + & \lambda_{13246} + & \lambda_{132456} + & \lambda_{13256} + & \lambda_{1346} + & \lambda_{13456} + & \lambda_{1356} & = 1 \\
& \lambda_{1246}\,, & \lambda_{12456}\,, & \lambda_{1256}\,, & \lambda_{13246}\,, & \lambda_{132456}\,, & \lambda_{13256}\,, & \lambda_{1346}\,, & \lambda_{13456}\,, & \lambda_{1356} & \geq 0
\end{array}
$$

Idea: Work only with a (small) subset of variables

This (much) smaller LP is called the *restricted master problem* (RMP)

Add more variables only when needed…

Master Problem: Linear Relaxation

Writing out all coefficients we need a tiny font size...

$$\begin{array}{lrrrl}
\min & 5\lambda_{1256} + 13\lambda_{13246} & + 15\lambda_{13256} & & \\
\text{s.t.} & 15\lambda_{1256} + 13\lambda_{13246} & + 10\lambda_{13256} & \leq & 14 \\
 & \lambda_{1256} + \lambda_{13246} & + \lambda_{13256} & = & 1 \\
 & \lambda_{1256}\ ,\ \lambda_{13246} & ,\ \lambda_{13256} & \geq & 0
\end{array}$$

Idea: Work only with a (small) subset of variables

This (much) smaller LP is called the *restricted master problem* (RMP)

Add more variables only when needed...

Master Problem: Linear Relaxation

Writing out all coefficients we need a tiny font size. . .

c_{13256} a_{13256}

$$
\begin{array}{llllr}
\min & 5\lambda_{1256} + 13\lambda_{13246} & \cdots & + 15\lambda_{13256} & \\
\text{s.t.} & 15\lambda_{1256} + 13\lambda_{13246} & \cdots & + 10\lambda_{13256} & \leq 14 \\
& \lambda_{1256} + \lambda_{13246} & \cdots & + 1\lambda_{13256} & = 1 \\
& \lambda_{1256}, \lambda_{13246} & \cdots & , \lambda_{13256} & \geq 0
\end{array}
$$

Idea: Work only with a (small) subset of variables

This (much) smaller LP is called the *restricted master problem (RMP)*

Add more variables only when needed. . .

Restricted Master Problem (RMP)

How do we *generate* such a *"column"* we don't know?

$$
\begin{array}{llcccccc}
\bar{z} = \min & \dots & + & 24\lambda_{132456} & + & \dots & & \\
\text{s.t.} & \dots & + & 9\lambda_{132456} & + & \dots & \leq & 14 \\
 & \dots & + & 1\lambda_{132456} & + & \dots & = & 1 \\
 & \dots & & 1\lambda_{132456} & & \dots & \geq & 0
\end{array}
$$

Restricted Master Problem (RMP)

How do we *generate* such a *"column"* we don't know?

$$
\begin{array}{llllllll}
 & & & & & & & \text{duals} \\
 & & & & & & & \downarrow \\
\bar{z} = \min & \dots & + & 24\lambda_{132456} & + & \dots & & \\
\text{s.t.} & \dots & + & 9\lambda_{132456} & + & \dots & \leq 14 & \pi_1 \\
 & \dots & + & 1\lambda_{132456} & + & \dots & = 1 & \pi_0 \\
 & \dots & & 1\lambda_{132456} & & \dots & \geq 0 &
\end{array}
$$

In simplex method: Variable to enter the basis must have *negative reduced cost*:

$$
\begin{aligned}
\bar{c}_{132456} &= 24 - (\pi_1, \pi_0)^t \cdot \begin{pmatrix} 9 \\ 1 \end{pmatrix} \\
&= 24 - 9\pi_1 - 1\pi_0 \overset{!}{<} 0
\end{aligned}
$$

Pricing subproblem

Checking whether there is a variable λ_p with

$$\bar{c}_p = \sum_{(i,j\in A)} c_{ij} x_{pij} - (\sum_{(i,j\in A)} t_{ij} x_{pij})\pi_1 - \pi_0 \quad < \quad 0$$

is an optimization problem, the *pricing subproblem*:

$\bar{c}^\star = \min \sum_{(i,j)\in A} (c_{ij} - \pi_1 t_{ij}) x_{ij} - \pi_0$
s. t. the x_{ij} encode a *feasible column*

If $\bar{c}^\star \geq 0$ then there is *no* improving variable;
otherwise we found a column to add to the RMP

Pricing subproblem

Checking whether there is a variable λ_p with

$$\bar{c}_p = \sum_{(i,j \in A)} c_{ij} x_{pij} - \Big(\sum_{(i,j \in A)} t_{ij} x_{pij} \Big) \pi_1 - \pi_0 \quad < \quad 0$$

is an optimization problem, the *pricing subproblem*:

$\bar{c}^\star = \min \sum_{(i,j) \in A} (c_{ij} - \pi_1 t_{ij}) x_{ij} - \pi_0$

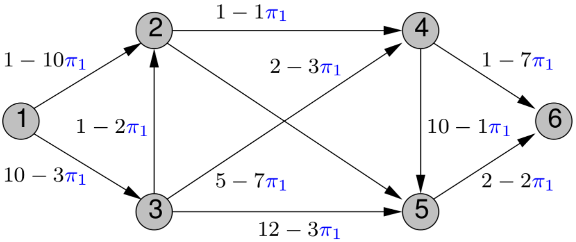
s. t.

$$\begin{aligned}
\sum_{j:(1,j) \in A} x_{1j} &= 1 \\
\sum_{j:(i,j) \in A} x_{ij} - \sum_{j:(j,i) \in A} x_{ji} &= 0 \quad i = 2,3,4,5 \\
\sum_{i:(i,6) \in A} x_{i6} &= 1 \\
x_{ij} &\geq 0 \qquad (i,j) \in A
\end{aligned}$$

If $\bar{c}^\star \geq 0$ then there is *no* improving variable;
otherwise we found a column to add to the RMP.

Pricing Subproblem

In our case this is a shortest path problem in



This is the original graph with modified costs

Column Generation Algorithm

An algorithmic summary of what we know so far:

0. formulate RMP (we talk about this later)
1. solve RMP to obtain optimal dual solution
2. solve pricing problem to determine whether a variable of negative reduced cost exists
3. if so, add it to RMP, and iterate (goto step 1)
4. otherwise: a primal solution to RMP is optimal for the master problem as well

✎ keep in mind

we need dual variables for the termination criterion

Example cont'd: Solving the LP Relaxation

Problem: How to start?
Initially, we have no feasible solution to the RMP!

One possibility: "Big M approach"
Introduce artificial variable y_0 with "large" cost,
say $M = 100$:

$$
\begin{array}{llcr}
\bar{z} = \min & 100y_0 & & \\
\text{s.t.} & y_0 & \leq & 14 \\
 & y_0 & = & 1 \\
 & y_0 & \geq & 0
\end{array}
$$

Example cont'd: Solving the LP Relaxation

$$
\begin{array}{llrl}
\bar{z} = \min & 100 y_0 & & \\
\text{s.t.} & y_0 & \leq 14 & \pi_1 \\
 & y_0 & = 1 & \pi_0 \\
 & y_0 & \geq 0 &
\end{array}
$$

Master Solution	$\bar{z}$	π_0	π_1	$\bar{c}^\star$	p	c_p	t_p

Example cont'd: Solving the LP Relaxation

$$\begin{array}{llrl} \bar{z} = \min & 100y_0 & & \\ \text{s.t.} & y_0 & \leq 14 & \pi_1 \\ & y_0 & = 1 & \pi_0 \\ & y_0 & \geq 0 & \end{array}$$

Master Solution	$\bar{z}$	π_0	π_1	$\bar{c}^\star$	p	c_p	t_p
$y_0 = 1$	100.0	100.00	0.00				

Example cont'd: Solving the LP Relaxation

$$
\begin{array}{llrl}
\bar{z} = \min & 100y_0 & & \\
\text{s.t.} & y_0 & \leq 14 & \pi_1 \\
 & y_0 & = 1 & \pi_0 \\
 & y_0 & \geq 0 &
\end{array}
$$

Master Solution	$\bar{z}$	π_0	π_1	$\bar{c}^\star$	p	c_p	t_p
$y_0 = 1$	100.0	100.00	0.00	-97.0	1246	3	18

Example cont'd: Solving the LP Relaxation

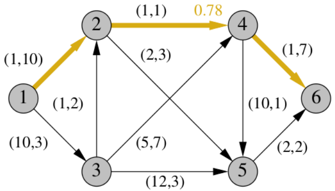
$$\begin{array}{llrcrcll}
\bar{z} = \min & 100y_0 & + & 3\lambda_{1246} & & & \\
\text{s.t.} & y_0 & + & 18\lambda_{1246} & \leq & 14 & {\color{blue}\pi_1} \\
 & y_0 & + & \lambda_{1246} & = & 1 & {\color{magenta}\pi_0} \\
 & y_0 & , & \lambda_{1246} & \geq & 0 &
\end{array}$$

Master Solution	$\bar{z}$	π_0	π_1	$\bar{c}^\star$	p	c_p	t_p
$y_0 = 1$	100.0	100.00	0.00	-97.0	1246	3	18

Example cont'd: Solving the LP Relaxation

$$
\begin{array}{lrcrl}
\bar{z} = \min & 100y_0 & + & 3\lambda_{1246} & & \\
\text{s.t.} & y_0 & + & 18\lambda_{1246} & \leq 14 & {\color{blue}\pi_1} \\
 & y_0 & + & \lambda_{1246} & = 1 & {\color{magenta}\pi_0} \\
 & y_0 & , & \lambda_{1246} & \geq 0 &
\end{array}
$$

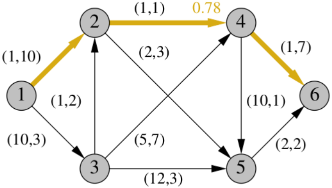
Master Solution	$\bar{z}$	π_0	π_1	$\bar{c}^\star$	p	c_p	t_p
$y_0 = 1$	100.0	100.00	0.00	-97.0	1246	3	18
$y_0 = 0.22, \lambda_{1246} = 0.78$	24.6	100.00	-5.39				



Example cont'd: Solving the LP Relaxation

$$\begin{array}{llllll} \bar{z} = \min & 100y_0 + 3\lambda_{1246} + 24\lambda_{1356} & & & \\ \text{s.t.} & y_0 + 18\lambda_{1246} + 8\lambda_{1356} & \leq & 14 & \pi_1 \\ & y_0 + \lambda_{1246} + \lambda_{1356} & = & 1 & \pi_0 \\ & y_0 , \lambda_{1246} , \lambda_{1356} & \geq & 0 & \end{array}$$

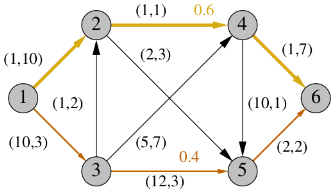
Master Solution	$\bar{z}$	π_0	π_1	$\bar{c}^\star$	p	c_p	t_p
$y_0 = 1$	100.0	100.00	0.00	-97.0	1246	3	18
$y_0 = 0.22, \lambda_{1246} = 0.78$	24.6	100.00	-5.39	-32.9	1356	24	8



Example cont'd: Solving the LP Relaxation

$$
\begin{array}{llcrcrl}
\bar{z} = \min & 100y_0 & + & 3\lambda_{1246} & + & 24\lambda_{1356} & \\
\text{s.t.} & y_0 & + & 18\lambda_{1246} & + & 8\lambda_{1356} & \leq 14 \quad \pi_1 \\
& y_0 & + & \lambda_{1246} & + & \lambda_{1356} & = 1 \quad \pi_0 \\
& y_0 & , & \lambda_{1246} & , & \lambda_{1356} & \geq 0
\end{array}
$$

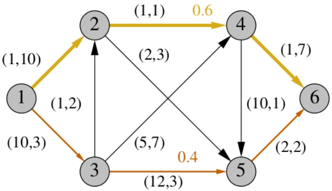
Master Solution	$\bar{z}$	π_0	π_1	$\bar{c}^\star$	p	c_p	t_p
$y_0 = 1$	100.0	100.00	0.00	-97.0	1246	3	18
$y_0 = 0.22, \lambda_{1246} = 0.78$	24.6	100.00	-5.39	-32.9	1356	24	8
$\lambda_{1246} = 0.6, \lambda_{1356} = 0.4$	11.4	40.80	-2.10				



Example cont'd: Solving the LP Relaxation

$$
\begin{array}{llcccccccl}
\bar{z} = \min & 100y_0 & + & 3\lambda_{1246} & + & 24\lambda_{1356} & + & 15\lambda_{13256} & & \\
\text{s.t.} & y_0 & + & 18\lambda_{1246} & + & 8\lambda_{1356} & + & 10\lambda_{13256} & \leq 14 & \pi_1 \\
 & y_0 & + & \lambda_{1246} & + & \lambda_{1356} & + & \lambda_{13256} & = 1 & \pi_0 \\
 & y_0 & , & \lambda_{1246} & , & \lambda_{1356} & , & \lambda_{13256} & \geq 0 &
\end{array}
$$

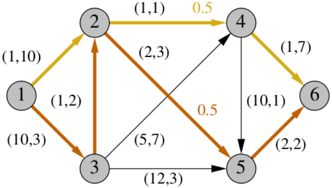
Master Solution	$\bar{z}$	π_0	π_1	$\bar{c}^\star$	p	c_p	t_p
$y_0 = 1$	100.0	100.00	0.00	-97.0	1246	3	18
$y_0 = 0.22, \lambda_{1246} = 0.78$	24.6	100.00	-5.39	-32.9	1356	24	8
$\lambda_{1246} = 0.6, \lambda_{1356} = 0.4$	11.4	40.80	-2.10	-4.8	13256	15	10



Example cont'd: Solving the LP Relaxation

$$\begin{array}{llllllllll}
\bar{z} = \min & 100y_0 & + & 3\lambda_{1246} & + & 24\lambda_{1356} & + & 15\lambda_{13256} & & \\
\text{s.t.} & y_0 & + & 18\lambda_{1246} & + & 8\lambda_{1356} & + & 10\lambda_{13256} & \leq 14 & \pi_1 \\
 & y_0 & + & \lambda_{1246} & + & \lambda_{1356} & + & \lambda_{13256} & = 1 & \pi_0 \\
 & y_0 & , & \lambda_{1246} & , & \lambda_{1356} & , & \lambda_{13256} & \geq 0 &
\end{array}$$

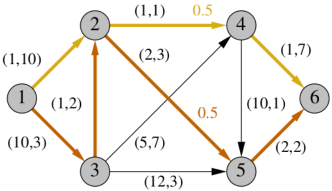
Master Solution	$\bar{z}$	π_0	π_1	$\bar{c}^\star$	p	c_p	t_p
$y_0 = 1$	100.0	100.00	0.00	-97.0	1246	3	18
$y_0 = 0.22, \lambda_{1246} = 0.78$	24.6	100.00	-5.39	-32.9	1356	24	8
$\lambda_{1246} = 0.6, \lambda_{1356} = 0.4$	11.4	40.80	-2.10	-4.8	13256	15	10
$\lambda_{1246} = \lambda_{13256} = 0.5$	9.0	30.00	-1.50				



Example cont'd: Solving the LP Relaxation

$$
\begin{array}{llcccccccccc}
\bar{z} = \min & 100y_0 & + & 3\lambda_{1246} & + & 24\lambda_{1356} & + & 15\lambda_{13256} & + & 5\lambda_{1256} & & \\
\text{s.t.} & y_0 & + & 18\lambda_{1246} & + & 8\lambda_{1356} & + & 10\lambda_{13256} & + & 15\lambda_{1256} & \leq & 14 \quad \pi_1 \\
& y_0 & + & \lambda_{1246} & + & \lambda_{1356} & + & \lambda_{13256} & + & \lambda_{1256} & = & 1 \quad \pi_0 \\
& y_0 & , & \lambda_{1246} & , & \lambda_{1356} & , & \lambda_{13256} & , & \lambda_{1256} & \geq & 0
\end{array}
$$

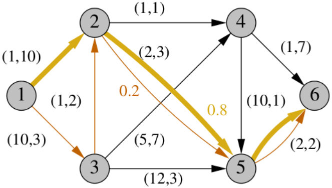
Master Solution	$\bar{z}$	π_0	π_1	$\bar{c}^\star$	p	c_p	t_p
$y_0 = 1$	100.0	100.00	0.00	-97.0	1246	3	18
$y_0 = 0.22, \lambda_{1246} = 0.78$	24.6	100.00	-5.39	-32.9	1356	24	8
$\lambda_{1246} = 0.6, \lambda_{1356} = 0.4$	11.4	40.80	-2.10	-4.8	13256	15	10
$\lambda_{1246} = \lambda_{13256} = 0.5$	9.0	30.00	-1.50	-2.5	1256	5	15



Example cont'd: Solving the LP Relaxation

$$
\begin{array}{lrcrcrcrcrcl}
\bar{z} = \min & 100y_0 & + & 3\lambda_{1246} & + & 24\lambda_{1356} & + & 15\lambda_{13256} & + & 5\lambda_{1256} & & \\
\text{s.t.} & 0\,y_0 & + & 18\lambda_{1246} & + & 8\lambda_{1356} & + & 10\lambda_{13256} & + & 15\lambda_{1256} & \leq & 14 \quad \pi_1 \\
 & y_0 & + & \lambda_{1246} & + & \lambda_{1356} & + & \lambda_{13256} & + & \lambda_{1256} & = & 1 \quad \pi_0 \\
 & y_0 & , & \lambda_{1246} & , & \lambda_{1356} & , & \lambda_{13256} & , & \lambda_{1256} & \geq & 0
\end{array}
$$

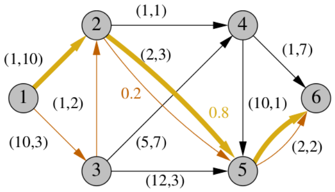
Master Solution	$\bar{z}$	π_0	π_1	$\bar{c}^\star$	p	c_p	t_p
$y_0 = 1$	100.0	100.00	0.00	-97.0	1246	3	18
$y_0 = 0.22, \lambda_{1246} = 0.78$	24.6	100.00	-5.39	-32.9	1356	24	8
$\lambda_{1246} = 0.6, \lambda_{1356} = 0.4$	11.4	40.80	-2.10	-4.8	13256	15	10
$\lambda_{1246} = \lambda_{13256} = 0.5$	9.0	30.00	-1.50	-2.5	1256	5	15
$\lambda_{13256} = 0.2, \lambda_{1256} = 0.8$	7.0	35.00	-2.00	0			

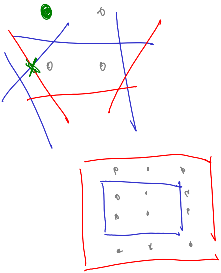


Arc flows: $x_{12} = 0.8,\; x_{13} = x_{32} = 0.2,\; x_{25} = x_{56} = 1$

Example cont'd: Solving the LP Relaxation

Remarks about the optimal RMP solution





- We use 0.2 times path 13256
- We use 0.8 times path 1256, which is *infeasible*

- A lower bound on the optimal integer solution objective value is $\bar{z} = 7.0$
- We would have obtained the corresponding "arc flow" solution (with the same lower bound) by solving the LP relaxation of the original IP

Integer Solutions

Beware! So far we *only* solved the LP relaxation

Integer Solutions

We will now apply a branch-and-bound (B&B) algorithm (we solved the root node so far)

In *every* B&B node we have to

- apply the reformulation again,
- and solve the RMP by column generation

Branching on Fractional Paths

It seems obvious to branch on $\lambda_{13256} = 0.2$ or $\lambda_{1256} = 0.8$

Generally, this is a bad idea. Why?

Branch $\lambda_{13256} = 1.0$ *enforces* this path
$\Rightarrow$ this reduces the problem
The lower bound from the LP relaxation may well improve

Branch $\lambda_{13256} = 0.0$ *forbids* this path
$\Rightarrow$ so what? why would the other paths care?
The lower bound from the LP most likely remains unchanged

keep in mind

Branching on master variables may give irrelevant decisions

Branching on Fractional Paths

It seems obvious to branch on $\lambda_{13256} = 0.2$ or $\lambda_{1256} = 0.8$

Generally, this is a bad idea. Why?

Branch $\lambda_{13256} = 1.0$ *enforces* this path
$\Rightarrow$ this reduces the problem
The lower bound from the LP relaxation may well improve

Branch $\lambda_{13256} = 0.0$ *forbids* this path
$\Rightarrow$ so what? why would the other paths care?
The lower bound from the LP most likely remains unchanged

keep in mind

Branching on master variables may give irrelevant decisions

Branching on Fractional Paths

It seems obvious to branch on $\lambda_{13256} = 0.2$ or $\lambda_{1256} = 0.8$

Generally, this is a bad idea. Why?

Branch $\lambda_{13256} = 1.0$ *enforces* this path
$\Rightarrow$ this reduces the problem
The lower bound from the LP relaxation may well improve

Branch $\lambda_{13256} = 0.0$ *forbids* this path
$\Rightarrow$ so what? why would the other paths care?
The lower bound from the LP most likely remains unchanged

✎ keep in mind

Branching on master variables may give irrelevant decisions

Branching on Fractional Paths

It seems obvious to branch on $\lambda_{13256} = 0.2$ or $\lambda_{1256} = 0.8$

Generally, this is a bad idea. Why?

Branch $\lambda_{13256} = 1.0$ *enforces* this path
$\Rightarrow$ this reduces the problem
The lower bound from the LP relaxation may well improve

Branch $\lambda_{13256} = 0.0$ *forbids* this path
$\Rightarrow$ so what? why would the other paths care?
The lower bound from the LP most likely remains unchanged

✎ keep in mind

Branching on master variables may give irrelevant decisions

Branching on Fractional Paths

It seems obvious to branch on $\lambda_{13256} = 0.2$ or $\lambda_{1256} = 0.8$

Generally, this is a bad idea. Why?

Branch $\lambda_{13256} = 1.0$ *enforces* this path
$\Rightarrow$ this reduces the problem
The lower bound from the LP relaxation may well improve

Branch $\lambda_{13256} = 0.0$ *forbids* this path
$\Rightarrow$ so what? why would the other paths care?
The lower bound from the LP most likely remains unchanged

✎ keep in mind

Branching on master variables may give irrelevant decisions

Branching on Fractional Paths

It seems obvious to branch on $\lambda_{13256} = 0.2$ or $\lambda_{1256} = 0.8$

Generally, this is a bad idea. Why?

Branch $\lambda_{13256} = 1.0$ *enforces* this path
$\Rightarrow$ this reduces the problem
The lower bound from the LP relaxation may well improve

Branch $\lambda_{13256} = 0.0$ *forbids* this path
$\Rightarrow$ so what? why would the other paths care?
The lower bound from the LP most likely remains unchanged

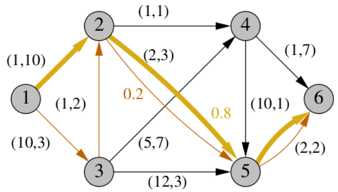
Possibly more involved:
How would we take care of this in the pricing?
(The forbidden path must not be re-generated!)

keep in mind

Branching on master variables may give irrelevant decisions and may need modification in the pricing

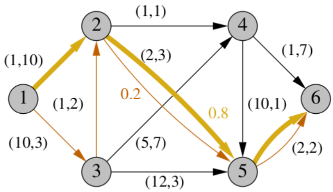
Branching on Fractional Arcs

What else can we do? Next “obvious” is to branch on
fractional *arc* variables, like $x_{12} = 0.8$



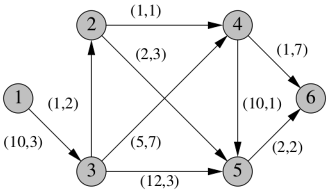
Branching on Fractional Arcs

What else can we do? Next "obvious" is to branch on fractional *arc* variables, like $x_{12} = 0.8$



Branching on Fractional Arcs

What else can we do? Next "obvious" is to branch on fractional *arc* variables, like $x_{12} = 0.8$



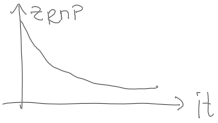
Branch $x_{12} = 0$

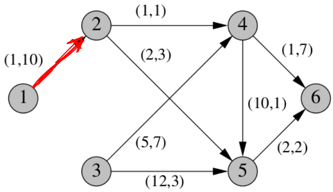
In subproblem: Arc $(1, 2)$ is removed from graph

In RMP: Variables λ_{1246} and λ_{1256} must be eliminated; re-optimization will give $y_0 > 0$, i.e., infeasible RMP

Branching on Fractional Arcs

What else can we do? Next "obvious" is to branch on fractional *arc* variables, like $x_{12} = 0.8$





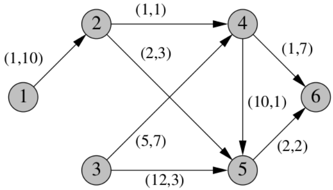
Branch $x_{12} = 1$

In subproblem: Arcs $(1,3)$ and $(3,2)$ are removed

In RMP: Variables corresponding to paths containing these arcs must be eliminated from RMP

Branching on Fractional Arcs

What else can we do? Next “obvious” is to branch on fractional *arc* variables, like $x_{12} = 0.8$

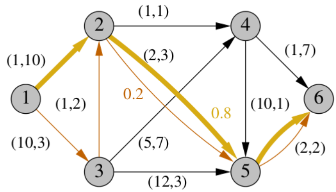


✎ keep in mind

Try branching on *original* variables

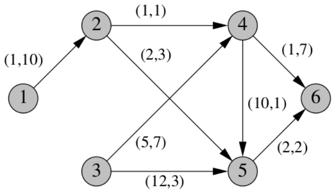
Branching: Further Ideas

We can try to impose “richer” constraints, e.g., branch on $x_{13} + x_{32} = 0.4$



Branching: Further Ideas

We can try to impose "richer" constraints, e.g., branch on $x_{13} + x_{32} = 0.4$



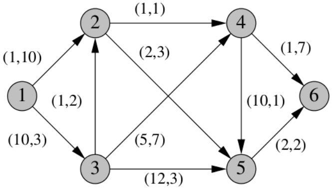
Branch $x_{13} + x_{32} = 0$

In subproblem: Arcs $(1,3)$ and $(3,2)$ are removed

In RMP: Variables corresponding to paths containing these arcs must be eliminated from RMP

Branching: Further Ideas

We can try to impose "richer" constraints, e.g., branch on $x_{13} + x_{32} = 0.4$

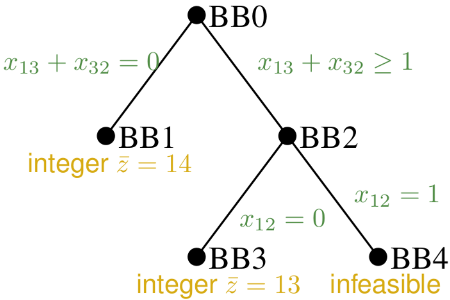


Branch $x_{13} + x_{32} \geq 1$

In subproblem: no changes

In RMP: This constraint must be expressed in terms of λ_p variables *via the reformulation process*

Example cont'd: Branch-and-Bound Tree

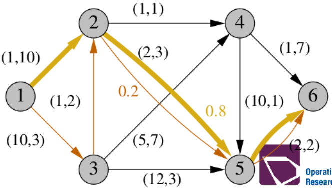


Example cont'd: Branch-and-Bound Tree

$$
\begin{array}{lrcrcrcrcrcrl}
\min & 100y_0 & + & 3\lambda_{1246} & + & 24\lambda_{1356} & + & 15\lambda_{13256} & + & 5\lambda_{1256} & & & \\
\text{s.t.} & & & 18\lambda_{1246} & + & 8\lambda_{1356} & + & 10\lambda_{13256} & + & 15\lambda_{1256} & \le & 14 & {\color{blue}\pi_1} \\
& y_0 & + & \lambda_{1246} & + & \lambda_{1356} & + & \lambda_{13256} & + & \lambda_{1256} & = & 1 & {\color{magenta}\pi_0} \\
& y_0, & & \lambda_{1246}, & & \lambda_{1356}, & & \lambda_{13256}, & & \lambda_{1256} & \ge & 0 &
\end{array}
$$

BB0 Master Solution	$\bar{z}$	π_0	π_1	π_2	$\bar{c}^\star$	p	c_p	t_p
$\lambda_{13256} = 0.2, \lambda_{1256} = 0.8$	7.0	35.00		-2.00	0			

BB0



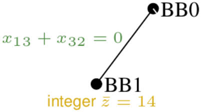
Operations Research

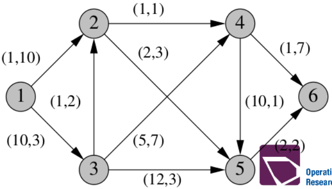
RWTH AACHEN UNIVERSITY

Example cont'd: Branch-and-Bound Tree

$$
\begin{array}{llcrcrcrcrcll}
\min & 100y_0 & + & 3\lambda_{1246} & + & 24\lambda_{1356} & + & 15\lambda_{13256} & + & 5\lambda_{1256} & & & \\
\text{s.t.} & & & 18\lambda_{1246} & + & 8\lambda_{1356} & + & 10\lambda_{13256} & + & 15\lambda_{1256} & \leq & 14 & \color{blue}{\pi_1} \\
 & & & & & & & & & & & & \\
 & y_0 & + & \lambda_{1246} & + & \lambda_{1356} & + & \lambda_{13256} & + & \lambda_{1256} & = & 1 & \color{magenta}{\pi_0} \\
 & y_0, & & \lambda_{1246}, & & \lambda_{1356}, & & \lambda_{13256}, & & \lambda_{1256} & \geq & 0 &
\end{array}
$$

BB1	Master Solution	$\bar{z}$	π_0	π_1	π_2	$\bar{c}^\star$	p	c_p	t_p





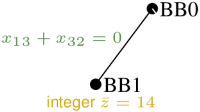
Operations Research

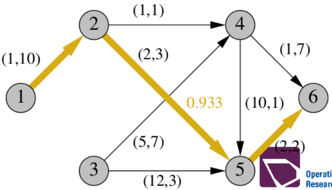
RWTH AACHEN UNIVERSITY

Example cont'd: Branch-and-Bound Tree

$$
\begin{array}{llrcrcrcrl}
\min & 100y_0 & + & 3\lambda_{1246} & + & 5\lambda_{1256} & & & \\
\text{s.t.} & & & 18\lambda_{1246} & + & 15\lambda_{1256} & \leq & 14 & {\color{blue}\pi_1} \\
 & & & & & & & & \\
 & y_0 & + & \lambda_{1246} & + & \lambda_{1256} & = & 1 & {\color{magenta}\pi_0} \\
 & y_0, & & \lambda_{1246}, & & \lambda_{1256} & \geq & 0 &
\end{array}
$$

BB1	Master Solution	$\bar{z}$	π_0	π_1	π_2	$\bar{c}^\star$	p	c_p	t_p
	$y_0 = 0.067, \lambda_{1256} = 0.933$	11.3	100	-6.33		0			





Operations Research

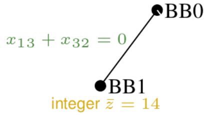
RWTH AACHEN UNIVERSITY

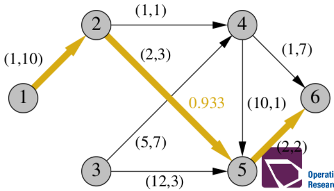
Example cont'd: Branch-and-Bound Tree

$$
\begin{array}{lrcrcrcrl}
\min & 100y_0 & + & 3\lambda_{1246} & + & 5\lambda_{1256} & & & \\
\text{s.t.} & & & 18\lambda_{1246} & + & 15\lambda_{1256} & \leq & 14 & \color{blue}{\pi_1} \\
 & y_0 & + & \lambda_{1246} & + & \lambda_{1256} & = & 1 & \color{magenta}{\pi_0} \\
 & y_0, & & \lambda_{1246}, & & \lambda_{1256} & \geq & 0 &
\end{array}
$$

BB1 Master Solution	$\bar{z}$	π_0	π_1	π_2	$\bar{c}^\star$	p	c_p	t_p
$y_0 = 0.067, \lambda_{1256} = 0.933$	11.3	100	-6.33		0			

M increased to 1000

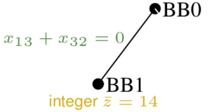


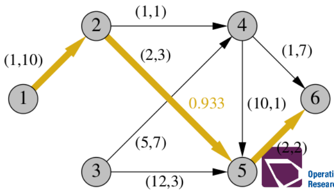


Example cont'd: Branch-and-Bound Tree

$$
\begin{array}{llllllll}
\min & 100y_0 & + & 3\lambda_{1246} & + & 14\lambda_{12456} & + & 5\lambda_{1256} & & & \\
\text{s.t.} & & & 18\lambda_{1246} & + & 14\lambda_{12456} & + & 15\lambda_{1256} & \leq & 14 & \pi_1 \\
& y_0 & + & \lambda_{1246} & + & \lambda_{12456} & + & \lambda_{1256} & = & 1 & \pi_0 \\
& y_0, & & \lambda_{1246}, & & \lambda_{12456}, & & \lambda_{1256} & \geq & 0 &
\end{array}
$$

BB1 Master Solution	$\bar{z}$	π_0	π_1	π_2	$\bar{c}^\star$	p	c_p	t_p
$y_0 = 0.067, \lambda_{1256} = 0.933$	11.3	100	-6.33		0			
M increased to1000								
$y_0 = 0.067, \lambda_{1256} = 0.933$	71.3	1000	-66.33		-57.3	12456	14	14





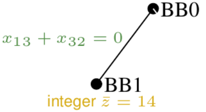
Operations Research

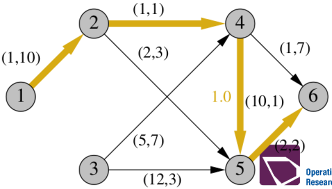
RWTH AACHEN UNIVERSITY

Example cont'd: Branch-and-Bound Tree

$$
\begin{array}{llcccccccl}
\min & 100y_0 & + & 3\lambda_{1246} & + & 14\lambda_{12456} & + & 5\lambda_{1256} & & \\
\text{s.t.} & & & 18\lambda_{1246} & + & 14\lambda_{12456} & + & 15\lambda_{1256} & \leq & 14 \quad \pi_1 \\
 & y_0 & + & \lambda_{1246} & + & \lambda_{12456} & + & \lambda_{1256} & = & 1 \quad \pi_0 \\
 & y_0, & & \lambda_{1246}, & & \lambda_{12456}, & & \lambda_{1256} & \geq & 0
\end{array}
$$

BB1 Master Solution	$\bar{z}$	π_0	π_1	π_2	$\bar{c}^\star$	p	c_p	t_p
$y_0 = 0.067, \lambda_{1256} = 0.933$	11.3	100	-6.33		0			
M increased to 1000								
$y_0 = 0.067, \lambda_{1256} = 0.933$	71.3	1000	-66.33		-57.3	12456	14	14
$\lambda_{12456} = 1$	$\mathbf{14}$	1000	-70.43		0			

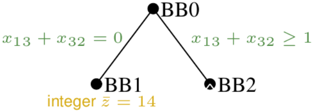


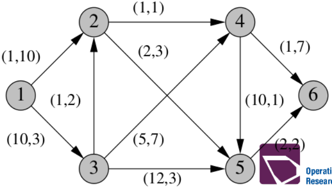


Example cont'd: Branch-and-Bound Tree

$$
\begin{array}{llcrcrcrcrcrl}
\min & 100y_0 & + & 3\lambda_{1246} & + & 24\lambda_{1356} & + & 15\lambda_{13256} & + & 5\lambda_{1256} & & & \\
\text{s.t.} & & & 18\lambda_{1246} & + & 8\lambda_{1356} & + & 10\lambda_{13256} & + & 15\lambda_{1256} & \leq & 14 & \pi_1 \\
 & y_0 & + & \lambda_{1246} & + & \lambda_{1356} & + & \lambda_{13256} & + & \lambda_{1256} & = & 1 & \pi_0 \\
 & y_0, & & \lambda_{1246}, & & \lambda_{1356}, & & \lambda_{13256}, & & \lambda_{1256} & \geq & 0 &
\end{array}
$$

BB2	Master Solution	$\bar{z}$	π_0	π_1	π_2	$\bar{c}^\star$	p	c_p	t_p

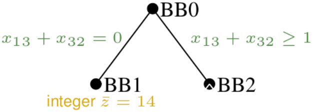


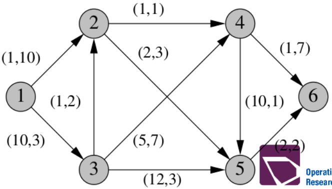


Example cont'd: Branch-and-Bound Tree

$$
\begin{array}{llllllllllll}
\min & 100y_0 & + & 3\lambda_{1246} & + & 24\lambda_{1356} & + & 15\lambda_{13256} & + & 5\lambda_{1256} & & \\
\text{s.t.} & & & 18\lambda_{1246} & + & 8\lambda_{1356} & + & 10\lambda_{13256} & + & 15\lambda_{1256} & \leq 14 & \color{blue}{\pi_1} \\
 & & & & & \lambda_{1356} & + & 2\lambda_{13256} & & & \geq 1 & \color{orange}{\pi_2} \\
 & y_0 & + & \lambda_{1246} & + & \lambda_{1356} & + & \lambda_{13256} & + & \lambda_{1256} & = 1 & \color{magenta}{\pi_0} \\
 & y_0, & & \lambda_{1246}, & & \lambda_{1356}, & & \lambda_{13256}, & & \lambda_{1256} & \geq 0 &
\end{array}
$$

BB2	Master Solution	$\bar{z}$	π_0	π_1	π_2	$\bar{c}^\star$	p	c_p	t_p



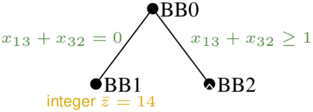


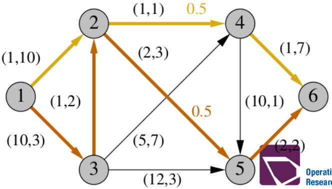
Example cont'd: Branch-and-Bound Tree

$$
\begin{array}{llllllllllll}
\min & 100y_0 & + & 3\lambda_{1246} & + & 24\lambda_{1356} & + & 15\lambda_{13256} & + & 5\lambda_{1256} & & \\
\text{s.t.} & & & 18\lambda_{1246} & + & 8\lambda_{1356} & + & 10\lambda_{13256} & + & 15\lambda_{1256} & \leq 14 & \pi_1 \\
 & & & & & \lambda_{1356} & + & 2\lambda_{13256} & & & \geq 1 & \pi_2 \\
 & y_0 & + & \lambda_{1246} & + & \lambda_{1356} & + & \lambda_{13256} & + & \lambda_{1256} & = 1 & \pi_0 \\
 & y_0, & & \lambda_{1246}, & & \lambda_{1356}, & & \lambda_{13256}, & & \lambda_{1256} & \geq 0 &
\end{array}
$$

BB2 Master Solution	$\bar{z}$	π_0	π_1	π_2	$\bar{c}^\star$	p	c_p	t_p
$\lambda_{1246} = \lambda_{13256} = 0.5$	9	15	-0.67	3.33	0			

Arc flows: $x_{12} = x_{13} = x_{24} = x_{25} = x_{32} = x_{46} = x_{56} = 0.5$



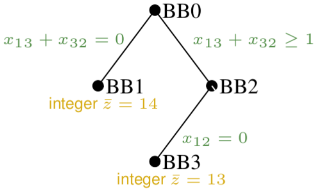


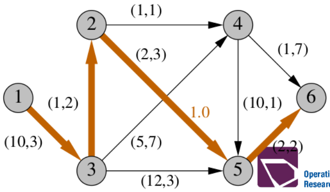
Operations Research
RWTH AACHEN UNIVERSITY

Example cont'd: Branch-and-Bound Tree

$$
\begin{array}{llllllll}
\min & 100y_0 & + 24\lambda_{1356} & + \lambda_{13246} & + 15\lambda_{13256} & & & \\
\text{s.t.} & & + 8\lambda_{1356} & + \lambda_{13246} & + 10\lambda_{13256} & \leq & 14 & \pi_1 \\
 & & \lambda_{1356} & & + 2\lambda_{13256} & \geq & 1 & \pi_2 \\
 & y_0 & + \lambda_{1356} & + \lambda_{13246} & + \lambda_{13256} & = & 1 & \pi_0 \\
 & y_0, & \lambda_{1356}, & \lambda_{13246}, & \lambda_{13256}, & \geq & 0 &
\end{array}
$$

BB3 Master Solution	$\bar{z}$	π_0	π_1	π_2	$\bar{c}^\star$	p	c_p	t_p
$\lambda_{13256} = 1$	**15**	15	0	0	-2	13246	13	13

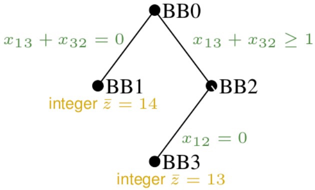


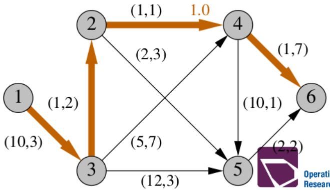


Example cont'd: Branch-and-Bound Tree

$$\begin{array}{lrrrrrrrrl}
\min & 100y_0 & + & 24\lambda_{1356} & + & \lambda_{13246} & + & 15\lambda_{13256} & & \\
\text{s.t.} & & + & 8\lambda_{1356} & + & \lambda_{13246} & + & 10\lambda_{13256} & \leq 14 & \color{blue}{\pi_1} \\
 & & & \lambda_{1356} & & & + & 2\lambda_{13256} & \geq 1 & \color{orange}{\pi_2} \\
 & y_0 & + & \lambda_{1356} & + & \lambda_{13246} & + & \lambda_{13256} & = 1 & \color{magenta}{\pi_0} \\
 & y_0, & & \lambda_{1356}, & & \lambda_{13246}, & & \lambda_{13256}, & \geq 0 &
\end{array}$$

BB3 Master Solution	$\bar{z}$	π_0	π_1	π_2	$\bar{c}^\star$	p	c_p	t_p
$\lambda_{13256} = 1$	**15**	15	0	0	-2	13246	13	13
$\lambda_{13246} = 1$	**13**	13	0	0	0			





Example cont'd: Branch-and-Bound Tree

$$
\begin{array}{lrcrcrcrcrcrcl}
\min & 100y_0 & + & 3\lambda_{1246} & + & 24\lambda_{1356} & + & \lambda_{13246} & + & 15\lambda_{13256} & + & 5\lambda_{1256} & & \\
\text{s.t.} & & & 18\lambda_{1246} & + & 8\lambda_{1356} & + & \lambda_{13246} & + & 10\lambda_{13256} & + & 15\lambda_{1256} & \leq & 14 \quad \pi_1 \\
 & & & & & \lambda_{1356} & & & + & 2\lambda_{13256} & & & \geq & 1 \quad \pi_2 \\
 & y_0 & + & \lambda_{1246} & + & \lambda_{1356} & + & \lambda_{13246} & + & \lambda_{13256} & + & \lambda_{1256} & = & 1 \quad \pi_0 \\
 & y_0, & & \lambda_{1246}, & & \lambda_{1356}, & & \lambda_{13246}, & & \lambda_{13256}, & & \lambda_{1256} & \geq & 0
\end{array}
$$

BB4 Master Solution	$\bar{z}$	π_0	π_1	π_2	$\bar{c}^\star$	p	c_p	t_p
$y_0 = 0.067, \lambda_{1256} = 0.933$	111.3	100	-6.33	100	0			
Infeasible arc flows