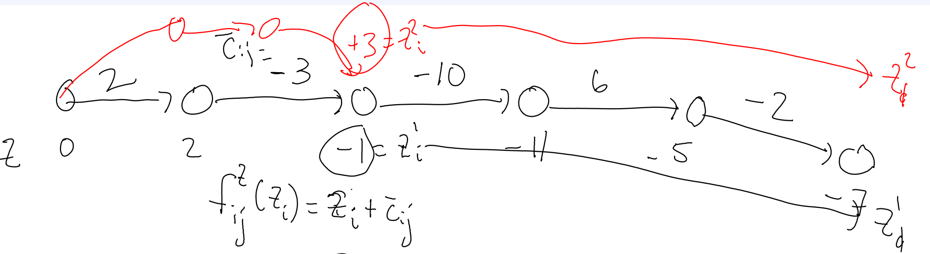
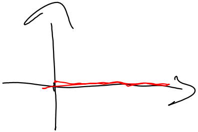
REFs

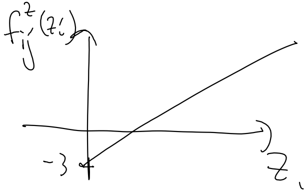


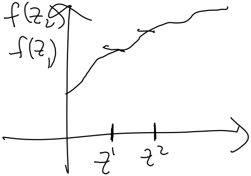
$$f_{ij}^z(z_i) = z_i + \bar{c}_{ij}$$
$$= z_i - 3$$

$$z_d^1 \le z_d^2$$

Reset function



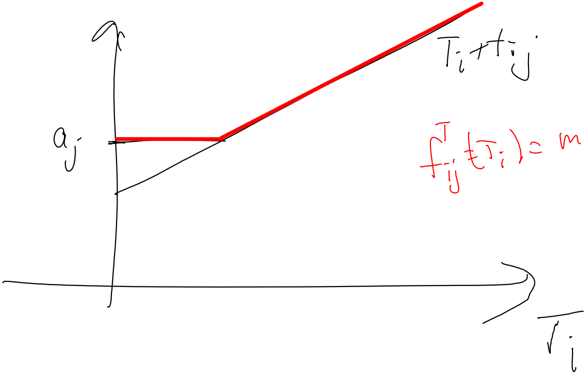




Dominance

$$z_i^1 \le z_i^2$$

REFs

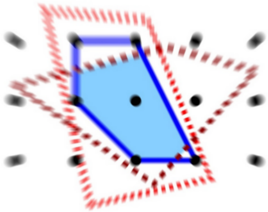


$$f^T_{ij}(T_i) = \max\{a_j, T_i + t_{ij}\}$$

School on Column Generation 2014

Dual, Extended, Branch, Cut

Marco Lübbecke · Aachen, Germany



@mluebbecke OperationsResearchRWTH

Plagiarism!


11
Souvenirs of PARIS
GIFTS & GADGETS - TEE SHIRTS - SWEAT SHIRTS
NATURALIA
11

T-Shirt 2.0

$$
\begin{array}{lrcll}
\min & \displaystyle\sum_{p\in P} c_p \lambda_p + \sum_{r\in R} c_r \lambda_r & & & \\
\text{s. t.} & \displaystyle\sum_{p\in P} \mathbf{a}_p \lambda_p + \sum_{r\in R} \mathbf{a}_r \lambda_r & \geq & \mathbf{b} & [\pi] \\
& \displaystyle\sum_{p\in P} \lambda_p & = & 1 & [\pi_0] \\
& \lambda & \geq & \mathbf{0} & \\
& \mathbf{x} & = & \displaystyle\sum_{p\in P} \mathbf{x}_p \lambda_p + \sum_{r\in R} \mathbf{x}_r \lambda_r & \\
& \mathbf{x} & \in & \mathbb{Z}_+^n &
\end{array}
$$

Reminder: Notation

Original problem:

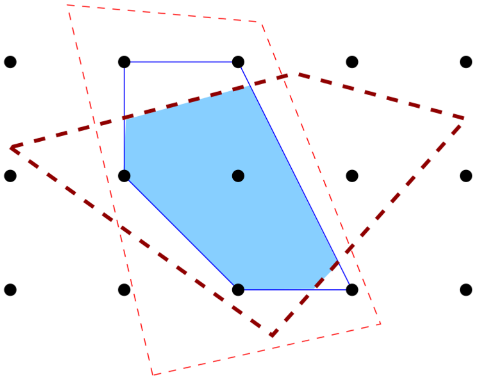
$$\begin{array}{rrcl} \min & \mathbf{cx} & & \\ \text{subject to} & A\mathbf{x} & \geq & \mathbf{b} \\ & \mathbf{x} & \in & X = \{\mathbf{x} \in \mathbb{Z}_+^n \times \mathbb{Q}_+^q \mid D\mathbf{x} \geq \mathbf{d}\} \end{array}$$

Master problem:

$$\begin{array}{rrlcl} z_{\mathrm{MP}} & := & \min & \sum_{j \in J} c_j \lambda_j & & \\ & & \text{subject to} & \sum_{j \in J} \mathbf{a}_j \lambda_j & \geq & \mathbf{b} \\ & & & \lambda_j & \geq & 0, \quad j \in J = P \cup R \end{array}$$

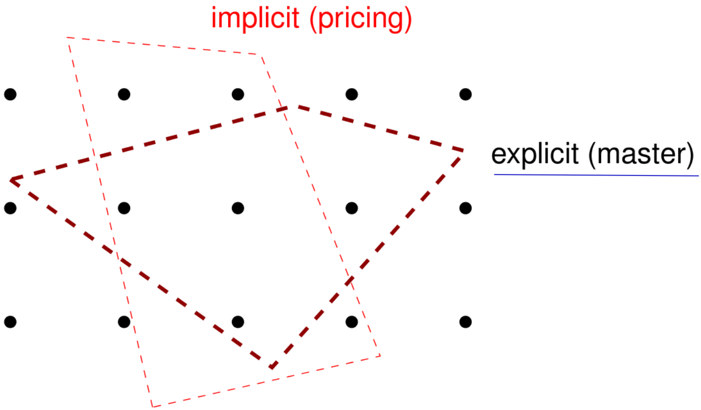
Pricing: $\min_{j \in J}\{c_j - \boldsymbol{\pi}\mathbf{a}_j\} = \min_{\mathbf{x} \in X}\{c(\mathbf{x}) - \boldsymbol{\pi} a(\mathbf{x})\}$

What we did



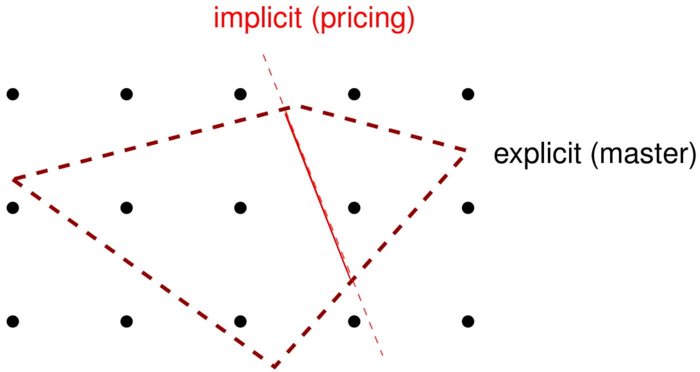
What we did

$$\mathcal{Q} := \quad \{\mathbf{x} \in \mathbb{Q}^n \mid D\mathbf{x} \geq \mathbf{d}\} \;\cap\; \underline{\{\mathbf{x} \in \mathbb{Q}^n \mid A\mathbf{x} \geq \mathbf{b}\}}$$



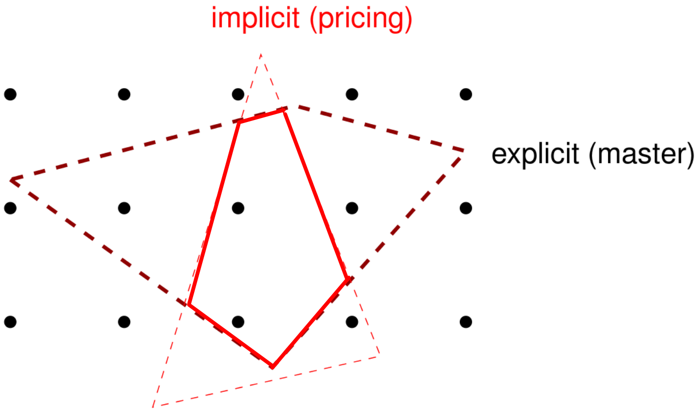
What we did

$$\mathcal{Q} := \quad \{\mathbf{x} \in \mathbb{Q}^n \mid D\mathbf{x} \geq \mathbf{d}\} \;\cap\; \{\mathbf{x} \in \mathbb{Q}^n \mid A\mathbf{x} \geq \mathbf{b}\}$$



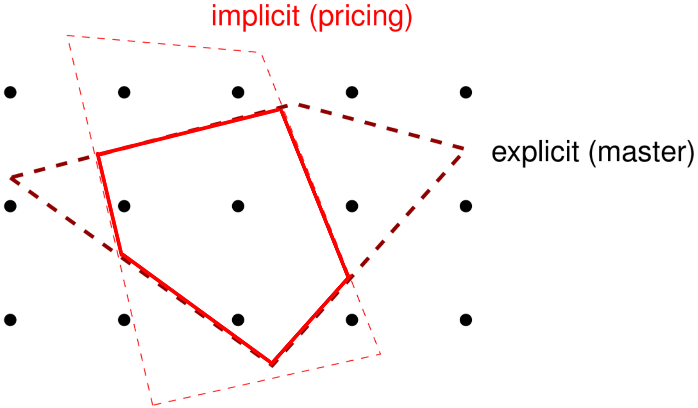
What we did

$$\mathcal{Q} := \quad \{\mathbf{x} \in \mathbb{Q}^n \mid D\mathbf{x} \geq \mathbf{d}\} \cap \{\mathbf{x} \in \mathbb{Q}^n \mid A\mathbf{x} \geq \mathbf{b}\}$$



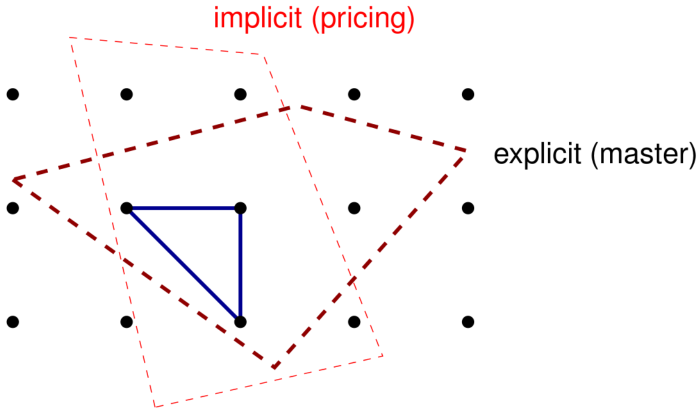
What we did

$$\mathcal{Q} := \quad \{\mathbf{x} \in \mathbb{Q}^n \mid D\mathbf{x} \geq \mathbf{d}\} \;\cap\; \{\mathbf{x} \in \mathbb{Q}^n \mid A\mathbf{x} \geq \mathbf{b}\}$$



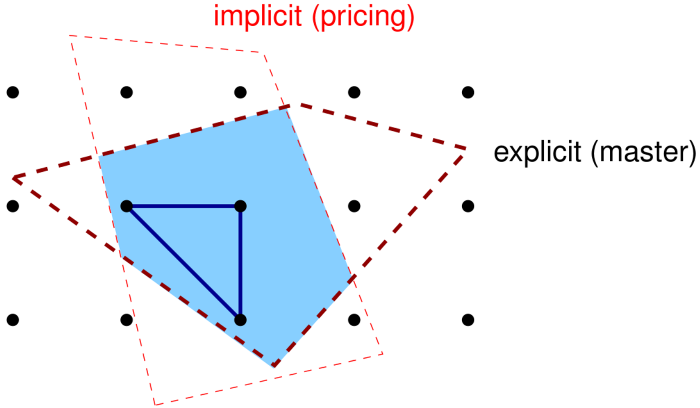
What we did

$$\mathcal{Q} := \quad \{\mathbf{x} \in \mathbb{Q}^n \mid D\mathbf{x} \geq \mathbf{d}\} \;\cap\; \{\mathbf{x} \in \mathbb{Q}^n \mid A\mathbf{x} \geq \mathbf{b}\}$$



What we did

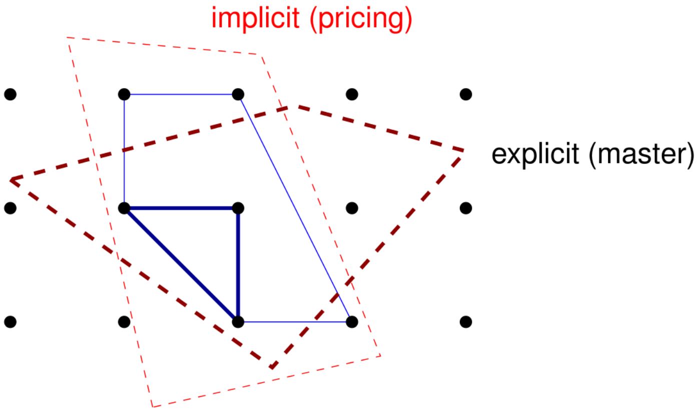
$$\mathcal{Q} := \{\mathbf{x} \in \mathbb{Q}^n \mid D\mathbf{x} \geq \mathbf{d}\} \cap \{\mathbf{x} \in \mathbb{Q}^n \mid A\mathbf{x} \geq \mathbf{b}\}$$



Not better than “standard” LP relaxation

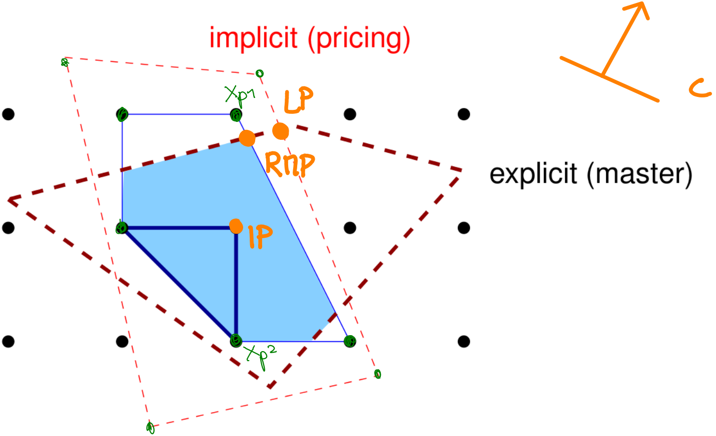
What we did

$$\mathcal{Q} := \{\mathbf{x} \in \mathbb{Q}^n \mid D\mathbf{x} \geq \mathbf{d}\} \cap \{\mathbf{x} \in \mathbb{Q}^n \mid A\mathbf{x} \geq \mathbf{b}\}$$
$$\mathcal{P} := \operatorname{conv}\{\mathbf{x} \in \mathbb{Z}^n \mid D\mathbf{x} \geq \mathbf{d}\}$$



What we did

$$\mathcal{Q} := \{\mathbf{x} \in \mathbb{Q}^n \mid D\mathbf{x} \geq \mathbf{d}\} \cap \{\mathbf{x} \in \mathbb{Q}^n \mid A\mathbf{x} \geq \mathbf{b}\}$$
$$\mathcal{P} := \mathrm{conv}\{\mathbf{x} \in \mathbb{Z}^n \mid D\mathbf{x} \geq \mathbf{d}\}$$



Better than LP relaxation iff $\mathcal{P} \subsetneq \mathcal{Q}$

Dual Interpretation of Column Generation

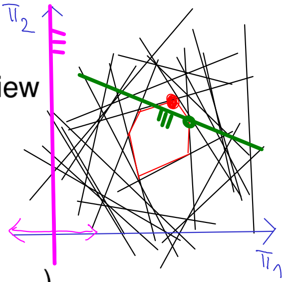
$$\bar{c}_p = c_p - \pi^T a_p \overset{!}{\geq} 0$$

keep in mind

Pricing vars in the primal is separating rows in the dual

Each variable "cuts away" parts of the infeasible dual region

Kelley (1961), Cheney & Goldstein (1959)



We first want to understand this dual point of view and then draw a few conclusions:

- Using Lagrangian relaxation
- Dual variable stabilization
- Consequences for the primal (init, pricing, . . .)

The Road ahead

Relation to Lagrangian Relaxation

The classical choice for producing dual bounds for integer programs is *Lagrangian Relaxation*

„linking constraints"

We penalize the violation of constraints $Ax \geq \mathbf{b}$ in the objective function via *Lagrangian multipliers* $\pi \geq \mathbf{0}$:

$$L(\boldsymbol{\pi}) := \min_{\mathbf{x} \in X} \; \mathbf{c}\mathbf{x} - \boldsymbol{\pi}(A\mathbf{x} - \mathbf{b})$$

$<$

$= \{x \mid Dx \geq d;\ x \in \mathbb{Z}^n_+\}$

Relation to Lagrangian Relaxation

The classical choice for producing dual bounds for integer programs is *Lagrangian Relaxation*

We penalize the violation of constraints $A\mathbf{x} \geq \mathbf{b}$ in the objective function via *Lagrangian multipliers* $\pi \geq \mathbf{0}$:

$$L(\boldsymbol{\pi}) := \min_{\mathbf{x} \in X} \ \mathbf{c}\mathbf{x} - \boldsymbol{\pi}(A\mathbf{x} - \mathbf{b})$$

This *Lagrangian subproblem* gives a *lower bound*:

$$L(\boldsymbol{\pi}) \leq \min\{\mathbf{c}\mathbf{x} - \boldsymbol{\pi}(\underbrace{A\mathbf{x} - \mathbf{b}}_{\geq 0}) \mid A\mathbf{x} \geq \mathbf{b}, \mathbf{x} \in X\} \leq z_{\mathrm{MP}}$$

Relation to Lagrangian Relaxation

The classical choice for producing dual bounds for integer programs is *Lagrangian Relaxation*

We penalize the violation of constraints $A\mathbf{x} \geq \mathbf{b}$ in the objective function via *Lagrangian multipliers* $\pi \geq \mathbf{0}$:

$$L(\boldsymbol{\pi}) := \min_{\mathbf{x} \in X} \; \mathbf{cx} - \boldsymbol{\pi}(A\mathbf{x} - \mathbf{b})$$

This *Lagrangian subproblem* gives a *lower bound*:

$$L(\boldsymbol{\pi}) \leq \min\{\mathbf{cx} - \boldsymbol{\pi}(\underbrace{A\mathbf{x} - \mathbf{b}}_{\geq 0}) \mid A\mathbf{x} \geq \mathbf{b}, \mathbf{x} \in X\} \leq z_{\mathrm{MP}}$$

We are interested in the best such bound:

$$\mathcal{L} := \max_{\boldsymbol{\pi} \geq \mathbf{0}} \; L(\boldsymbol{\pi}) \qquad \text{“\textit{Lagrangian dual}”}$$

Operations Research | RWTH AACHEN UNIVERSITY

Lagrangian Dual Problem

We are interested in the best such bound:

$$\mathcal{L} := \max_{\boldsymbol{\pi} \geq \mathbf{0}} L(\boldsymbol{\pi}) \; = \; \max_{\boldsymbol{\pi} \geq \mathbf{0}} \min_{\mathbf{x} \in X} \mathbf{c}\mathbf{x} - \boldsymbol{\pi}(A\mathbf{x} - \mathbf{b})$$

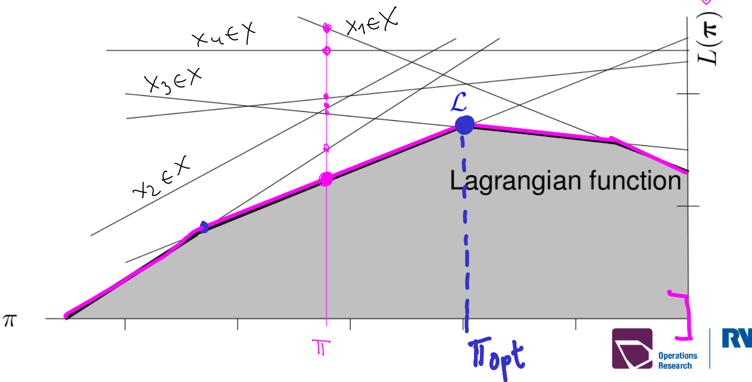
optimization is over $\boldsymbol{\pi}$; problem in the dual space
before we worked with $\mathbf{x}$ in the primal space

Lagrangian Dual Problem

We are interested in the best such bound:

$$\mathcal{L} := \max_{\boldsymbol{\pi} \geq \mathbf{0}} L(\boldsymbol{\pi}) = \max_{\boldsymbol{\pi} \geq \mathbf{0}} \min_{\mathbf{x} \in X} \mathbf{cx} - \boldsymbol{\pi}(A\mathbf{x} - \mathbf{b})$$

optimization is over $\boldsymbol{\pi}$; problem in the dual space
before we worked with $\mathbf{x}$ in the primal space



Lagrangian Dual Problem

We are interested in the best such bound:

$$\mathcal{L} := \max_{\boldsymbol{\pi} \geq \mathbf{0}} L(\boldsymbol{\pi}) = \max_{\boldsymbol{\pi} \geq \mathbf{0}} \min_{\mathbf{x} \in X} \mathbf{cx} - \boldsymbol{\pi}(A\mathbf{x} - \mathbf{b})$$

$$= \underbrace{(c - \pi A)}_{} x + \pi b$$

optimization is over $\boldsymbol{\pi}$; problem in the dual space
before we worked with $\mathbf{x}$ in the primal space

The *Lagrangian function* $L(\boldsymbol{\pi}), \boldsymbol{\pi} \geq \mathbf{0}$ is

- lower envelope of a family of functions linear in $\boldsymbol{\pi}$
- *concave* function of $\boldsymbol{\pi}$ and *piecewise linear*
- *breakpoints* where optimal solution of $L(\boldsymbol{\pi})$ not unique
- not differentiable, only *sub-differentiable*
- Most commonly solved via *subgradient algorithms*

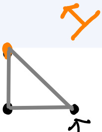
Re-Writing the Lagrangian Dual Problem

Remember: Replacing X by $\mathrm{conv}(X)$ does not change z_{MP}

Given multipliers $\boldsymbol{\pi}$, the Lagrangian function is

$$L(\boldsymbol{\pi}) = \begin{cases} -\infty & \text{if } (\mathbf{c} - \boldsymbol{\pi} A)\mathbf{x}_r < 0 \text{ for an } r \in R \\ \mathbf{c}\mathbf{x}_p - \boldsymbol{\pi}(A\mathbf{x}_p - \mathbf{b}) & \text{for some } p \in P \text{ otherwise.} \end{cases}$$

Re-Writing the Lagrangian Dual Problem



Remember: Replacing X by $\operatorname{conv}(X)$ does not change z_{MP}

Given multipliers $\boldsymbol{\pi}$, the Lagrangian function is

$$L(\boldsymbol{\pi}) = \begin{cases} -\infty & \text{if } (\mathbf{c} - \boldsymbol{\pi} A)\mathbf{x}_r < 0 \text{ for an } r \in R \\ \mathbf{c}\mathbf{x}_p - \boldsymbol{\pi}(A\mathbf{x}_p - \mathbf{b}) & \text{for some } p \in P \text{ otherwise.} \end{cases}$$

extreme points P

Assuming z_{MP} to be finite, we avoid unboundedness by re-writing the Lagrangian dual problem as

$$\max_{\boldsymbol{\pi} \geq \mathbf{0}} \min_{p \in P} \mathbf{c}\mathbf{x}_p - \boldsymbol{\pi}(A\mathbf{x}_p - \mathbf{b}) \quad \text{such that} \quad (\mathbf{c} - \boldsymbol{\pi} A)\mathbf{x}_r \geq 0, \forall r \in R$$

Written as Linear Program

$$\max_{\boldsymbol{\pi} \geq \mathbf{0}} \min_{p \in P} \mathbf{c}\mathbf{x}_p - \boldsymbol{\pi}(A\mathbf{x}_p - \mathbf{b}) \text{ such that } (\mathbf{c} - \boldsymbol{\pi}A)\mathbf{x}_r \geq 0, \forall r \in R$$

is a linear program with many constraints

$$\begin{array}{rlcll} \mathcal{L} = \max & \pi_0 & & & \\ \text{subject to} & \boldsymbol{\pi}(A\mathbf{x}_p - \mathbf{b}) + \pi_0 & \leq & \mathbf{c}\mathbf{x}_p, & p \in P \quad \leftarrow \\ & \boldsymbol{\pi}A\mathbf{x}_r & \leq & \mathbf{c}\mathbf{x}_r, & r \in R \\ & \boldsymbol{\pi} & \geq & \mathbf{0} & \end{array}$$

Written as Linear Program

$$\max_{\boldsymbol{\pi}\geq\mathbf{0}} \min_{p\in P} \mathbf{c}\mathbf{x}_p - \boldsymbol{\pi}(A\mathbf{x}_p - \mathbf{b}) \text{ such that } (\mathbf{c}-\boldsymbol{\pi}A)\mathbf{x}_r \geq 0, \forall r \in R$$

is a linear program with many constraints

$$\begin{array}{rlcll}
\mathcal{L} = \max & \pi_0 & & & \\
\text{subject to} & \boldsymbol{\pi}(A\mathbf{x}_p - \mathbf{b}) + \pi_0 & \leq & \mathbf{c}\mathbf{x}_p, & p \in P \\
& \boldsymbol{\pi}A\mathbf{x}_r & \leq & \mathbf{c}\mathbf{x}_r, & r \in R \\
& \boldsymbol{\pi} & \geq & \mathbf{0} &
\end{array}$$

The dual is (our convexification master formulation!)

$$\begin{array}{rlcl}
\mathcal{L} = \min & \sum_{p\in P} \mathbf{c}\mathbf{x}_p\lambda_p + \sum_{r\in R} \mathbf{c}\mathbf{x}_r\lambda_r & & \\
\text{subject to} & \sum_{p\in P} A\mathbf{x}_p\lambda_p + \sum_{r\in R} A\mathbf{x}_r\lambda_r & \geq & \mathbf{b}\sum_{p\in P}\lambda_p \\
& \sum_{p\in P}\lambda_p & = & 1 \\
& \boldsymbol{\lambda} & \geq & \mathbf{0}
\end{array}$$

Equivalence of Bounds

For a given vector $\boldsymbol{\pi}$ of multipliers and a constant π_0,

$$L(\boldsymbol{\pi}) = (\boldsymbol{\pi}\mathbf{b} + \pi_0) + \min_{\mathbf{x}\in \mathrm{conv}(X)} (\mathbf{c} - \boldsymbol{\pi}A)\mathbf{x} - \pi_0 \quad = \quad z_{\mathrm{RMP}} + z_{\mathrm{SP}},$$

✎ keep in mind

The Dantzig-Wolfe bound equals the Lagrangian bound

For optimal $\mathbf{x}$ and $\boldsymbol{\pi}$ we have $z_{\mathrm{MP}} = \mathbf{c}\mathbf{x} = L(\boldsymbol{\pi})$.

"Dantzig-Wolfe decomposition and Lagrangian relaxation are dual to each other"

Stealing from Lagrange

There is a lot of literature on Lagrangian relaxation

- we do not (always) need optimal dual solutions, an approximation obtained from a subgradient algorithm may suffice
- a dual heuristic may help
- hybrids: a few simplex iterations, then a subgradient algorithm to improve the dual solution
- or the other way round: start with bundle or volume, then simplex
- ...

Dual Bounds: Monday's Example Revisited

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

Master Solution	z_{RMP}	π_0	π_1	z_{SP}	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			

Dual Bounds: Monday's Example Revisited

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

Master Solution	z_{RMP}	π_0	π_1	z_{SP}	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
$\lambda_{1246} = \lambda_{13256} = 0.5$	$\boxed{9.0}$	30.00	-1.50	$\boxed{-2.5}$	1256	5	15
$\lambda_{13256} = 0.2, \lambda_{1256} = 0.8$	$\boxed{7.0}$	35.00	-2.00	0			

We could have stopped *before* the last pricing (which can be costly!):

$z_{\text{RMP}} + z_{\text{SP}} = 9.0 - 2.5 = 6.5$ and $\lceil 6.5 \rceil = \boxed{7.0}$

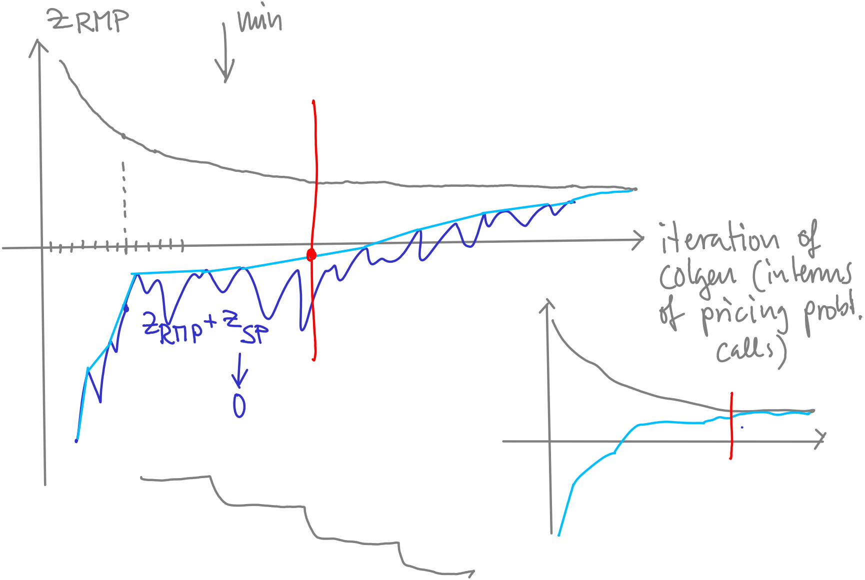
When the optimum is integer one can stop as soon as $\lceil LB \rceil = UB$

for a lower bound LB and an upper bound UB on z_{MP}

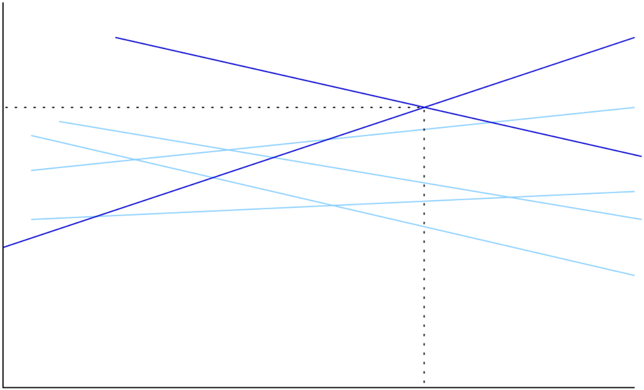
Operations Research

RWTH AACHEN UNIVERSITY

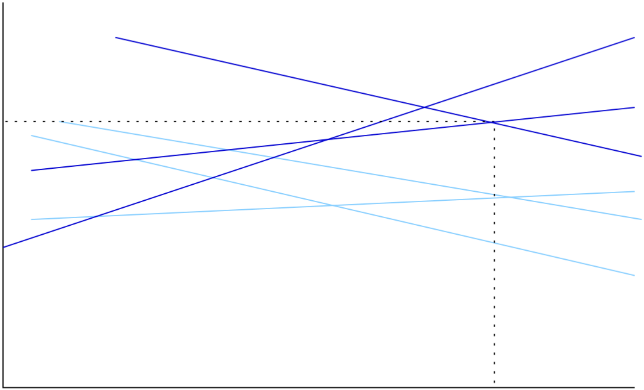
Use Lagrangian Bound to branch early



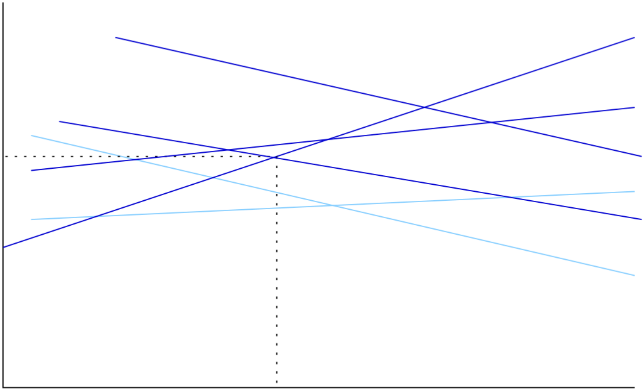
What Happens in the Dual Space I



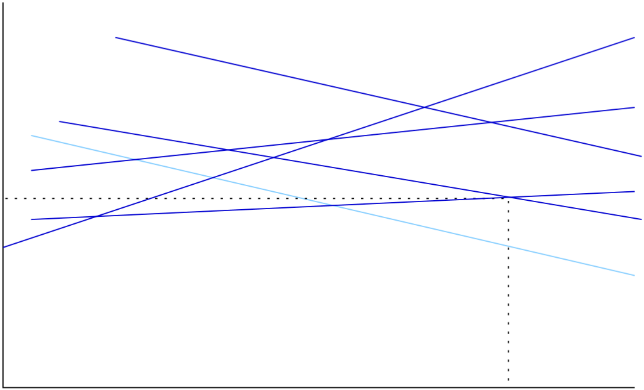
What Happens in the Dual Space I



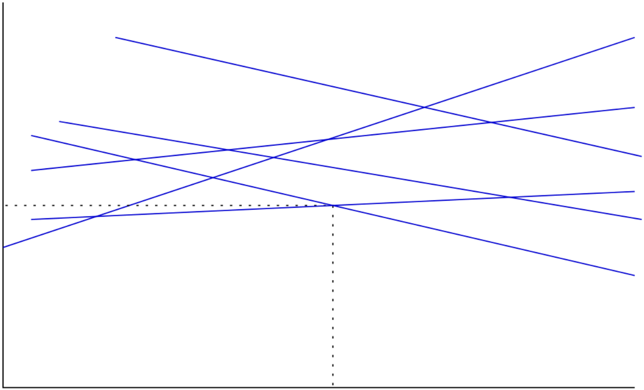
What Happens in the Dual Space I



What Happens in the Dual Space I



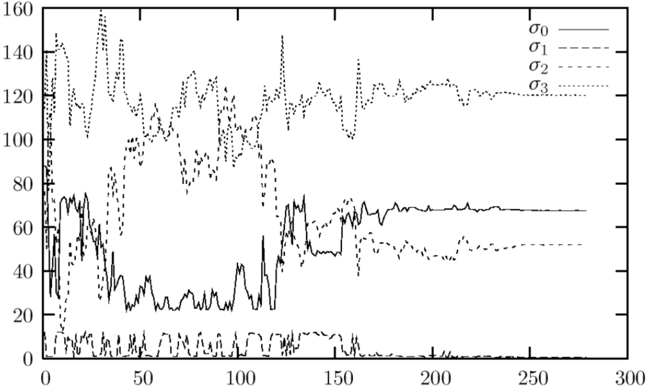
What Happens in the Dual Space I



The dual solution “jumps around” during the iterations

What Happens in the Dual Space I

You can observe this yourself when you plot the values of dual variables over the iterations



What Happens in the Dual Space I

This behavior is a major efficiency issue!

In the beginning, generated variables are “irrelevant” because the initial dual information is meaningless (“*heading-in*”)

Conversely, quality of columns is best in the end, but they may be hard to generate, and proving optimality may take time (“*tailing-off*”)

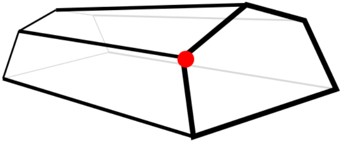
What Happens in the Dual Space II

When the RMP is solved with a simplex method one ends up in a *vertex* of the optimal face F of the dual polyhedron.

What Happens in the Dual Space II

When the RMP is solved with a simplex method one ends up in a *vertex* of the optimal face F of the dual polyhedron.

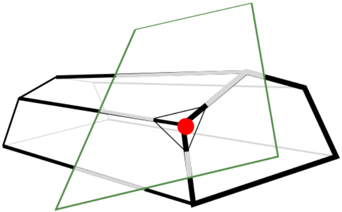
This is kind of an *arbitrary* representation of F
A column may “cut away” only that vertex (and a bit more)



What Happens in the Dual Space II

When the RMP is solved with a simplex method one ends up in a *vertex* of the optimal face F of the dual polyhedron.

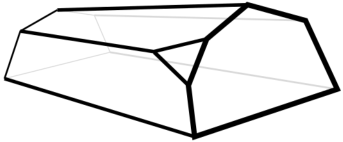
This is kind of an *arbitrary* representation of F
A column may "cut away" only that vertex (and a bit more)



What Happens in the Dual Space II

When the RMP is solved with a simplex method one ends up in a *vertex* of the optimal face F of the dual polyhedron.

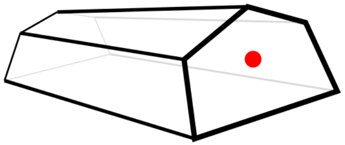
This is kind of an *arbitrary* representation of F
A column may "cut away" only that vertex (and a bit more)



What Happens in the Dual Space II

When the RMP is solved with a simplex method one ends up in a *vertex* of the optimal face F of the dual polyhedron.

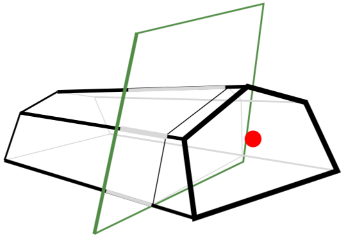
This is kind of an *arbitrary* representation of F
A “better” representation could yield a “deeper” cut



What Happens in the Dual Space II

When the RMP is solved with a simplex method one ends up in a *vertex* of the optimal face F of the dual polyhedron.

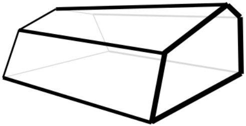
This is kind of an *arbitrary* representation of F
A "better" representation could yield a "deeper" cut



What Happens in the Dual Space II

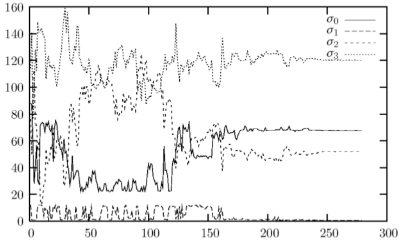
When the RMP is solved with a simplex method one ends up in a *vertex* of the optimal face F of the dual polyhedron.

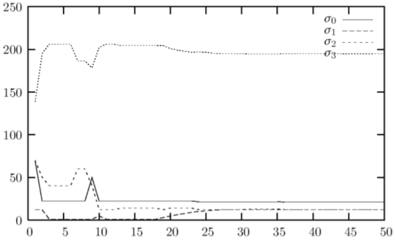
This is kind of an *arbitrary* representation of F
A “better” representation could yield a “deeper” cut



Dual Variable Stabilization

What if we could reduce the dual oscillation effect?





Observe the number of iterations!

Interior Point Stabilization

keep in mind

Try getting a dual solution interior to the optimal dual face

Interior Point Stabilization

✎ keep in mind

Try getting a dual solution interior to the optimal dual face

How can we do that?

- solve RMP with an interior point method, barrier, ACCPM,...
  appears to work very well
- obtain several vertex solutions and use their convex combination:
  Solve RMP; fix objective function to optimum; try several random $\mathbf{c}$ and $-\mathbf{c}$ Rousseau, Gendreau & Feillet (2007)
  probably very expensive when RMPs are large
- ...

An Experiment: A Box Method

Assume *good* dual multipliers $\bar{\pi}$ are available initially
We impose a *box constraint* $\bar{\boldsymbol{\pi}} - \varepsilon \leq \boldsymbol{\pi} \leq \bar{\boldsymbol{\pi}} + \varepsilon$

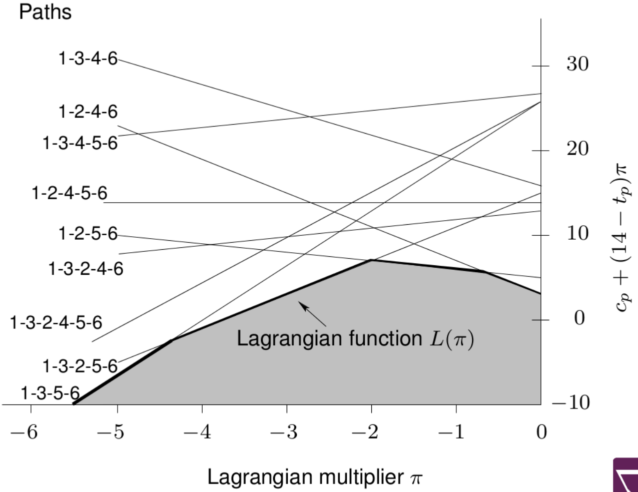
which is added as columns to the RMP,
in our example (from Monday) e.g., $\bar{\pi} = -2$, $\varepsilon = 0.1$

$$
\begin{array}{llrcrcrcr}
z_{\text{RMP}} = \min & 100y_0 & + & 1.9s_1 & + & 2.1s_2 & & \\
\text{s.t.} & y_0 & + & s_1 & - & s_2 & \leq & 14 \\
 & y_0 & & & & & = & 1 \\
 & y_0 & , & s_1 & , & s_2 & \geq & 0
\end{array}
$$

Master Solution	z_{RMP}	π_0	π_1	z_{SP}	p	c_p	t_p	UB	LB
$y_0 = 1, s_2 = 14$	73.4	100.0	-1.9	-66.5	1256	5	15	$--$	6.9
$\lambda_{1256} = 1, s_1 = 1$	7.1	36.5	-2.1	-0.5	13256	15	10	7.1	6.6
$\lambda_{13256} = 0.2, \lambda_{1256} = 0.8$	7.0	35.0	-2.0	0				7	7

Geometric Interpretation

p	1246	1256	12456	13246	13256	132456	1346	13456	1356
line	$3-4\pi$	$5-\pi$	14	$13+\pi$	$15+4\pi$	$24+5\pi$	$16-3\pi$	$27+\pi$	$24+6\pi$



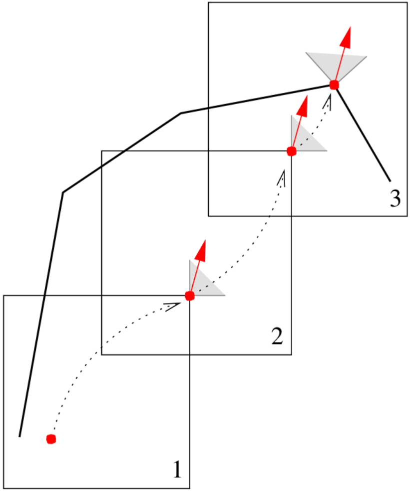
Stabilization: Boxstep method

Jacques can tell you more about "perfect dual information"

For the moment: we don't have it!

What if we trusted our current dual information, e.g., the solution which produced the currently best dual bound, and put a box around such a "stability center?"

Stabilization: Boxstep method



- When dual optimal solution hits boundary: Relocate box
- Otherwise: done

Marsten, Hogan & Blankenship (1975)

But a hard-coded box is not very flexible

Stabilization: Polyhedral Penalty Terms

primal:

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

Stabilization: Polyhedral Penalty Terms

primal:

$$\begin{array}{llrcl} \min & \mathbf{c}\boldsymbol{\lambda} & & & \\ \text{s.t.} & & A\boldsymbol{\lambda} - \mathbf{y}_- + \mathbf{y}_+ & = & \mathbf{b} \\ & & \mathbf{y}_- & \leq & \boldsymbol{\varepsilon}_- \\ & & \mathbf{y}_+ & \leq & \boldsymbol{\varepsilon}_+ \\ & & \boldsymbol{\lambda}, \mathbf{y}_-, \mathbf{y}_+ & \geq & \mathbf{0} \end{array}$$

Slack/surplus variables $\mathbf{y}_-$, $\mathbf{y}_+$ perturb $\mathbf{b}$ by $\boldsymbol{\varepsilon} \in [-\boldsymbol{\varepsilon}_-, \boldsymbol{\varepsilon}_+]$

This reduces primal degeneracy!

Stabilization: Polyhedral Penalty Terms

$$
\text{primal:} \quad \begin{array}{rrcl}
\min & \mathbf{c}\boldsymbol{\lambda} - \boldsymbol{\delta}_-\mathbf{y}_- + \boldsymbol{\delta}_+\mathbf{y}_+ & & \\
\text{s.t.} & A\boldsymbol{\lambda} - \mathbf{y}_- + \mathbf{y}_+ & = & \mathbf{b} \\
& \mathbf{y}_- & \leq & \boldsymbol{\varepsilon}_- \\
& \mathbf{y}_+ & \leq & \boldsymbol{\varepsilon}_+ \\
& \boldsymbol{\lambda}, \mathbf{y}_-, \mathbf{y}_+ & \geq & \mathbf{0}
\end{array}
$$

Slack/surplus variables $\mathbf{y}_-$, $\mathbf{y}_+$ perturb $\mathbf{b}$ by $\boldsymbol{\varepsilon} \in [-\boldsymbol{\varepsilon}_-, \boldsymbol{\varepsilon}_+]$

This reduces primal degeneracy!

Stabilization: Polyhedral Penalty Terms

$$\begin{array}{llrcll}
\text{primal:} & \min & \mathbf{c}\boldsymbol{\lambda} - \boldsymbol{\delta}_{-}\mathbf{y}_{-} + \boldsymbol{\delta}_{+}\mathbf{y}_{+} & & & \\
& \text{s.t.} & A\boldsymbol{\lambda} - \mathbf{y}_{-} + \mathbf{y}_{+} & = & \mathbf{b} & [\boldsymbol{\pi}] \\
& & \mathbf{y}_{-} & \leq & \boldsymbol{\varepsilon}_{-} & [\mathbf{w}_{-}] \\
& & \mathbf{y}_{+} & \leq & \boldsymbol{\varepsilon}_{+} & [\mathbf{w}_{+}] \\
& & \boldsymbol{\lambda}, \mathbf{y}_{-}, \mathbf{y}_{+} & \geq & \mathbf{0} &
\end{array}$$

Slack/surplus variables $\mathbf{y}_{-}$, $\mathbf{y}_{+}$ perturb $\mathbf{b}$ by $\boldsymbol{\varepsilon} \in [-\boldsymbol{\varepsilon}_{-}, \boldsymbol{\varepsilon}_{+}]$

This reduces primal degeneracy!

$$\begin{array}{llrcl}
\text{dual:} & \max & \boldsymbol{\pi}\mathbf{b} - \boldsymbol{\varepsilon}_{-}\mathbf{w}_{-} - \boldsymbol{\varepsilon}_{+}\mathbf{w}_{+} & & \\
& \text{s.t.} & \boldsymbol{\pi}A & \leq & \mathbf{c} \\
& & -\boldsymbol{\pi} - \mathbf{w}_{-} & \leq & -\boldsymbol{\delta}_{-} \\
& & \boldsymbol{\pi} - \mathbf{w}_{+} & \leq & \boldsymbol{\delta}_{+} \\
& & \mathbf{w}_{-}, \mathbf{w}_{+} & \geq & \mathbf{0}
\end{array}$$

Stabilization: Polyhedral Penalty Terms

primal:

$$\begin{array}{lrcll} \min & \mathbf{c}\boldsymbol{\lambda} - \boldsymbol{\delta}_-\mathbf{y}_- + \boldsymbol{\delta}_+\mathbf{y}_+ & & & \\ \text{s.t.} & A\boldsymbol{\lambda} - \mathbf{y}_- + \mathbf{y}_+ & = & \mathbf{b} & [\boldsymbol{\pi}] \\ & \mathbf{y}_- & \leq & \boldsymbol{\varepsilon}_- & [\mathbf{w}_-] \\ & \mathbf{y}_+ & \leq & \boldsymbol{\varepsilon}_+ & [\mathbf{w}_+] \\ & \boldsymbol{\lambda}, \mathbf{y}_-, \mathbf{y}_+ & \geq & \mathbf{0} & \end{array}$$

Slack/surplus variables $\mathbf{y}_-$, $\mathbf{y}_+$ perturb $\mathbf{b}$ by $\boldsymbol{\varepsilon} \in [-\boldsymbol{\varepsilon}_-, \boldsymbol{\varepsilon}_+]$

This reduces primal degeneracy!

dual:

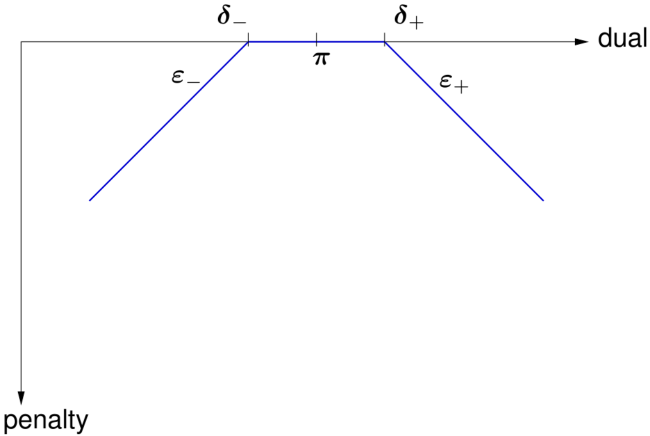
$$\begin{array}{lrcl} \max & \boldsymbol{\pi}\mathbf{b} - \boldsymbol{\varepsilon}_-\mathbf{w}_- - \boldsymbol{\varepsilon}_+\mathbf{w}_+ & & \\ \text{s.t.} & \boldsymbol{\pi}A & \leq & \mathbf{c} \\ & -\boldsymbol{\pi} - \mathbf{w}_- & \leq & -\boldsymbol{\delta}_- \\ & \boldsymbol{\pi} - \mathbf{w}_+ & \leq & \boldsymbol{\delta}_+ \\ & \mathbf{w}_-, \mathbf{w}_+ & \geq & \mathbf{0} \end{array}$$

Restricts dual variables to $\boldsymbol{\pi} \in [\boldsymbol{\delta}_- - \mathbf{w}_-, \boldsymbol{\delta}_+ + \mathbf{w}_+]$

$\Longleftrightarrow$ deviation of $\boldsymbol{\pi}$ from $[\boldsymbol{\delta}_-, \boldsymbol{\delta}_+]$ is allowed but penalized

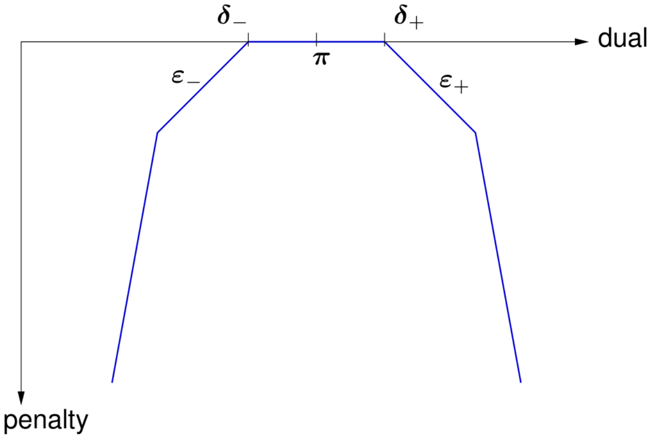
Stabilization: Polyhedral Penalty Terms

The penalty function



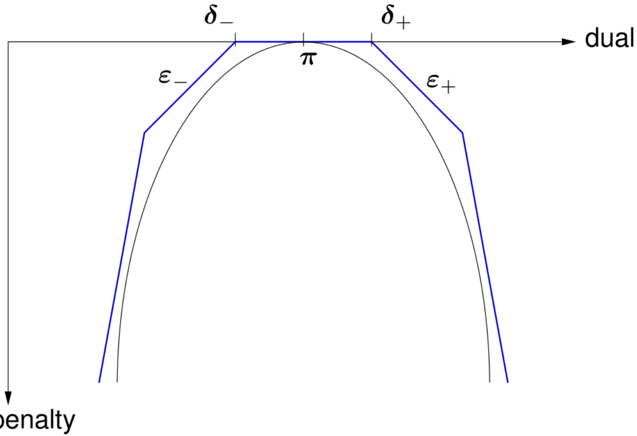
Stabilization: Polyhedral Penalty Terms

The penalty function can be easily extended to five pieces (apparently works best) Ben Amor, Desrosiers & Frangioni (2009)



Stabilization: Polyhedral Penalty Terms

The penalty function can be easily extended to five pieces (apparently works best) Ben Amor, Desrosiers & Frangioni (2009)



Doesn't this almost look like a quadratic function?

Stabilization: Quadratic Penalty Term

Bundle methods

- maximize a concave function (as the Langrangian)
- penalize the Euclidean distance to a stability center

Extensive comparison to “classical stabilization:”
no clear winner

Briant, Lemaréchal, Meurdesoif, Michel, Perrot & Vanderbeck (2008)

But it’s so easy to solve the RMP with a bundle method!

Stabilization: Convex Combinations

Combine current dual π with a stability center $\hat{\pi}$
Call pricing with $\alpha\hat{\boldsymbol{\pi}} + (1-\alpha)\boldsymbol{\pi}$ for $0 \leq \alpha \leq 1$

- a column is added only when it is negative w.r.t. to π
- otherwise the dual bound improves
- terminate when the duality gap is below a threshold

The duality gap reduces in each iteration provably by a factor of $(1-\alpha)^{-1}$ Pessoa, Uchoa, Poggi de Aragão & Rodrigues (2008)

- only a single parameter to tune
- less flexible than stabilization with penalty terms
- recently improved to “auto-tune”

Dual Cutting Planes

Complementary concept: Valid inequalities in the dual
Easiest example: change set partitioning to set covering constraints (essentially halves the dual space)

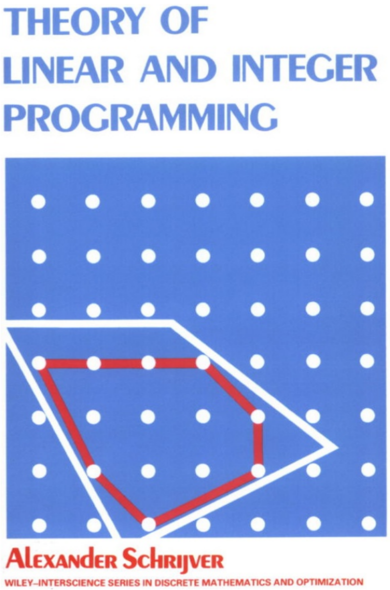
Add $\boldsymbol{\pi} E \leq \mathbf{e}$ which are valid for the dual optimal face:

RMP becomes $\min\{\mathbf{c}\boldsymbol{\lambda} + \mathbf{e}\mathbf{y} \mid A\boldsymbol{\lambda} + E\mathbf{y} \geq \mathbf{b},\ \boldsymbol{\lambda}, \mathbf{y} \geq \mathbf{0}\}$

Ben Amor, Desrosiers & Valério de Carvalho (2006)

Jacques can say more on that

The Integer Programming Dilemma



A Glimpse of Extended Formulations

$Q = \{(\mathbf{x}, \boldsymbol{\lambda}) \in \mathbb{R}^n \times \mathbb{R}^\ell \mid A\mathbf{x} + L\boldsymbol{\lambda} \leq \mathbf{b}\}$ is called an *extended formulation* of a polyhedron $O \subseteq \mathbb{R}^n$ if

$$O = \mathrm{proj}_{\mathbf{x}}(Q) := \{\mathbf{x} \in \mathbb{R}^n \mid \exists \boldsymbol{\lambda} \in \mathbb{R}^\ell : (\mathbf{x}, \boldsymbol{\lambda}) \in Q\}$$

Q is an extended formulation of the integer set X if
$X = \mathrm{proj}_{\mathbf{x}}(Q) \cap \mathbb{Z}^n_+$

This is of interest in combinatorial optimization, since extended formulations are typically stronger than the original.

A Glimpse of Extended Formulations

$Q = \{(\mathbf{x}, \boldsymbol{\lambda}) \in \mathbb{R}^n \times \mathbb{R}^\ell \mid A\mathbf{x} + L\boldsymbol{\lambda} \leq \mathbf{b}\}$ is called an *extended formulation* of a polyhedron $O \subseteq \mathbb{R}^n$ if

$$O = \mathrm{proj}_{\mathbf{x}}(Q) := \{\mathbf{x} \in \mathbb{R}^n \mid \exists \boldsymbol{\lambda} \in \mathbb{R}^\ell : (\mathbf{x}, \boldsymbol{\lambda}) \in Q\}$$

Q is an extended formulation of the integer set X if
$X = \mathrm{proj}_{\mathbf{x}}(Q) \cap \mathbb{Z}^n_+$

This is of interest in combinatorial optimization, since extended formulations are typically stronger than the original. Can we believe this?

A Glimpse of Extended Formulations

$Q = \{(\mathbf{x}, \boldsymbol{\lambda}) \in \mathbb{R}^n \times \mathbb{R}^\ell \mid A\mathbf{x} + L\boldsymbol{\lambda} \leq \mathbf{b}\}$ is called an *extended formulation* of a polyhedron $O \subseteq \mathbb{R}^n$ if

$$O = \mathrm{proj}_{\mathbf{x}}(Q) := \{\mathbf{x} \in \mathbb{R}^n \mid \exists \boldsymbol{\lambda} \in \mathbb{R}^\ell : (\mathbf{x}, \boldsymbol{\lambda}) \in Q\}$$

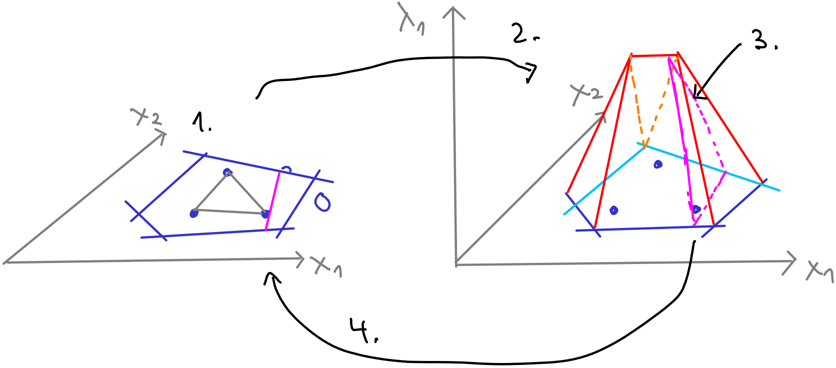
Q is an extended formulation of the integer set X if
$X = \mathrm{proj}_{\mathbf{x}}(Q) \cap \mathbb{Z}^n_+$

This is of interest in combinatorial optimization, since extended formulations are typically stronger than the original. Can we believe this?

Of course! DW reformulation is an example

$$\ldots \quad \mathbf{x} = \sum_{p \in P} \mathbf{x}_p \lambda_p \quad \ldots$$

Extended Formulations: Sketch



1. formulate original model
2. extend, i.e., re-formulate in higher-dim. space
3. find valid inequalities
4. project these back to original space, hopefully tightening the relaxation

Example: Multicommodity Flow Extension

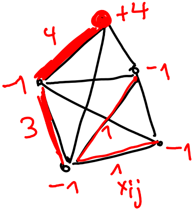
$$\min \sum_{ij \in E} c_{ij} y_{ij}$$

Problem: Minimum spanning tree in $G = (V, E)$

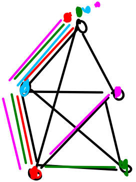
Flow based formulation: Send one flow unit from root to all other nodes, $|V| - 1$ units in total

x_{ij} amount of flow on edge $(i, j) \in E$
y_{ij} indicate whether $(i, j) \in E$ is in the spanning tree

A crucial constraint: $x_{ij} \leq (|V| - 1) \cdot y_{ij}$ (big-M!)



$x_{ij} \leq 4 \cdot y_{ij}$



$x_{ij}^{\bullet} \leq y_{ij}$
$x_{ij}^{\bullet} \leq y_{ij}$
$x_{ij}^{\bullet} \leq y_{ij}$
$x_{ij}^{\bullet} \leq y_{ij}$

$x_{ij} = x_{ij}^{\bullet} + x_{ij}^{\bullet} + x_{ij}^{\bullet} + x_{ij}^{\bullet}$

Example: Multicommodity Flow Extension

Problem: Minimum spanning tree in $G = (V, E)$

Flow based formulation: Send one flow unit from root to all other nodes, $|V| - 1$ units in total

x_{ij} amount of flow on edge $(i, j) \in E$
y_{ij} indicate whether $(i, j) \in E$ is in the spanning tree

A crucial constraint: $x_{ij} \leq (|V| - 1) \cdot y_{ij}$ (big-M!)

Extended formulation: A separate commodity for each vertex, that is, $x_{ij} = \sum_k x_{ij}^k, \; k = 1, \ldots, |V|$

The above constraint becomes $x_{ij}^k \leq y_{ij}$ which obviously better reflects the integrality of the y_{ij} variables

Example: Time Extension

Machine scheduling: Jobs j with processing times p_j
Classical: Start time variables s_j

Problematic: Sequencing constraints hard to formulate

$s_i + p_i \leq s_j$ xor $s_j + p_j \leq s_i$ usually with “big M”

Example: Time Extension

$t = 1, \dots, T$

Machine scheduling: Jobs j with processing times p_j
Classical: Start time variables s_j

Problematic: Sequencing constraints hard to formulate

$s_i + p_i \leq s_j$ xor $s_j + p_j \leq s_i$ usually with "big M"

Better: $s_j^t \in \{0, 1\}$ job j starts at time t or not

No jobs in parallel: $\sum_j \sum_{u=t-p_j+1}^{t} s_j^t \leq 1 \ \forall t$

much stronger!

Similar: hop-extended formulation for TSP;
capacity-extended formulation for the VRP; etc.

Example: Dynamic Programming Extension

Dynamic programs $\sim$ shortest path problem on a (possibly huge) network (e.g., Guy yesterday)

nodes $\sim$ states, arcs $\sim$ state transitions

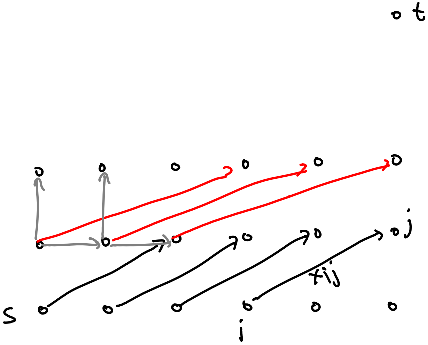
Standard flow formulation gives a linear program
(the dual of which describes the DP recursion)

The variables of this LP give an extended formulation

These variables describe state transitions,
they are a rich source for defining cutting planes

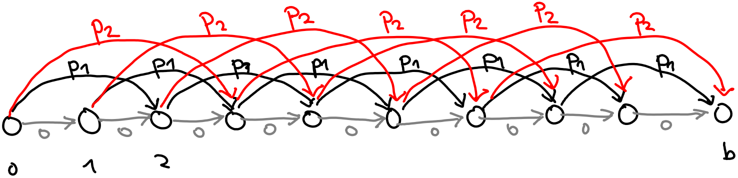
Example: Knapsack

$$\sum_{i=1}^{n} a_i x_i \leq b$$



$a_1 = 2$

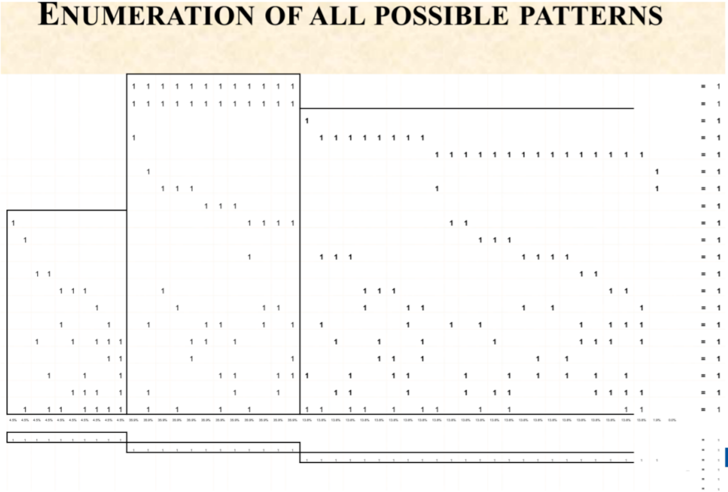
$a_2 = 3$



Example: “Partial Convexification”

Remember Jacques’ “secret ballot” problem:

The original formulation was hopelessly fractional, the reformulated variables *carried more information*



Extended Formulations: Learn More!

✎ keep in mind

Extended formulations can "put part of the integrality into the new variables' meaning"

Knowledge in this area will help you with B&C and B&P!

Read the survey by Vanderbeck & Wolsey (2010)

Reminder: DW Decomposition

Original problem:

$$\begin{array}{rrcl} \min & \mathbf{c}\mathbf{x} & & \\ \text{subject to} & A\mathbf{x} & \leq & \mathbf{b} \leftarrow \\ & \mathbf{x} & \in & X \end{array}$$

$$X = \{\mathbf{x} \in \mathbb{Z}_+^n \times \mathbb{Q}_+^q \mid D\mathbf{x} \leq \mathbf{d}\}$$

Reminder: Convexification

Reformulation of $\text{conv}(X)$:

$$
\begin{array}{rrcl}
\min & \sum_{p\in P} c_p\lambda_p + \sum_{r\in R} c_r\lambda_r & & \\
\text{subject to} & \sum_{p\in P} \mathbf{a}_p\lambda_p + \sum_{r\in R} \mathbf{a}_r\lambda_r & \leq & \mathbf{b} \\
 & \sum_{p\in P} \lambda_p & = & 1 \\
 & \boldsymbol{\lambda} & \geq & \mathbf{0} \\
 & \mathbf{x} & = & \sum_{p\in P} \mathbf{x}_p\lambda_p + \sum_{r\in R} \mathbf{x}_r\lambda_r \\
 & \mathbf{x} & \in & \mathbb{Z}_+^n
\end{array}
$$

Pricing: $\min\{\mathbf{cx} - \boldsymbol{\pi} A\mathbf{x} - \pi_0 \mid \mathbf{x} \in X\}$

Reminder: Discretization

Reformulation of X itself:

$$\mathbf{x} = \sum_{p \in P} \mathbf{x}_p \lambda_p + \sum_{r \in R} \mathbf{x}_r \lambda_r, \ \sum_{p \in P} \lambda_p = 1, \ \boldsymbol{\lambda} \in \mathbb{Z}_+^{|P|+|R|}$$

This yields:

$$\begin{array}{rrcl} \min & \sum_{p \in P} c_p \lambda_p + \sum_{r \in R} c_r \lambda_r & & \\ \text{subject to} & \sum_{p \in P} \mathbf{a}_p \lambda_p + \sum_{r \in R} \mathbf{a}_r \lambda_r & \leq & \mathbf{b} \\ & \sum_{p \in P} \lambda_p & = & 1 \\ & \boldsymbol{\lambda} & \in & \{0,1\}^{|P|} \times \mathbb{Z}_+^{|R|} \end{array}$$

Pricing: as before, however, needs to be able to generate points in the interior of X

Cutting Planes on Original Variables

In convexification, integrality holds on original variables

$$\begin{array}{rrcl} \min & \mathbf{c}\mathbf{x} & & \\ \text{subject to} & A\mathbf{x} & \leq & \mathbf{b} \\ & F\mathbf{x} & \leq & \mathbf{f} \qquad \text{valid inequalities with duals } \alpha \\ & \mathbf{x} & \in & X \end{array}$$

DW reformulation yields (added to the master):

$$\sum_{p \in P} \mathbf{f}_p \lambda_p + \sum_{r \in R} \mathbf{f}_r \lambda_r \leq \mathbf{f} \qquad \text{with } \mathbf{f}_j = F\mathbf{x}_j,\ j \in P \cup R$$

Pricing: $\min\{\mathbf{c}\mathbf{x} - \pi A\mathbf{x} - \alpha F\mathbf{x} - \pi_0 \mid \mathbf{x} \in X\}$

✎ keep in mind

Only the subproblem's objective function is changed!

Cutting Planes on Original Variables

In convexification, integrality holds on original variables

$$\begin{array}{rrcll} \min & \mathbf{c}\mathbf{x} & & & \\ \text{subject to} & A\mathbf{x} & \leq & \mathbf{b} & \\ & F\mathbf{x} & \leq & \mathbf{f} & \quad \text{valid inequalities with duals } \boldsymbol{\alpha} \\ & \mathbf{x} & \in & X & \end{array}$$

DW reformulation yields (added to the master):

$$\sum_{p \in P} \mathbf{f}_p \lambda_p + \sum_{r \in R} \mathbf{f}_r \lambda_r \leq \mathbf{f} \qquad \text{with } \mathbf{f}_j = F\mathbf{x}_j,\, j \in P \cup R$$

Pricing: $\min\{\mathbf{c}\mathbf{x} - \boldsymbol{\pi} A\mathbf{x} - \boldsymbol{\alpha} F\mathbf{x} - \pi_0 \mid \mathbf{x} \in X\}$

✎ keep in mind

Only the subproblem's objective function is changed!

HEN
SITY

Cutting Planes on Original Variables

In convexification, integrality holds on original variables

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

$$\begin{array}{rrcl} \min & \mathbf{c}\mathbf{x} & & \\ \text{subject to} & A\mathbf{x} & \leq & \mathbf{b} \\ & F\mathbf{x} & \leq & \mathbf{f} \\ & \mathbf{x} & \in & X \end{array}$$

valid inequalities with duals $\boldsymbol{\alpha}$

DW reformulation yields (added to the master):

$$\sum_{p \in P} \mathbf{f}_p \lambda_p + \sum_{r \in R} \mathbf{f}_r \lambda_r \leq \mathbf{f} \qquad \text{with } \mathbf{f}_j = F\mathbf{x}_j,\ j \in P \cup R$$

Pricing: $\min\{\mathbf{c}\mathbf{x} - \boldsymbol{\pi} A\mathbf{x} - \boldsymbol{\alpha} F\mathbf{x} - \pi_0 \mid \mathbf{x} \in X\}$

✎ keep in mind

Only the subproblem's objective function is changed!

Enforcing the Cut in the Pricing Problem

Alternatively (or additionally): Add cut to the subproblem, i.e., let $X_F = \{\mathbf{x} \in X \mid F\mathbf{x} \leq \mathbf{f}\}$

Pricing: $\min\{\mathbf{cx} - \boldsymbol{\pi} A\mathbf{x} - \pi_0 \mid \mathbf{x} \in X_F\}$

✎ keep in mind

The cut is convexified and may give a stronger dual bound

✎ keep in mind

This may destroy the pricing problem's structure,
but Jacques has something to say

Cutting Planes on Original Variables

If we could make our favorite IP solver separate generic cuts on original variables. . .

. . . partially works with SCIP Gamrath & Lübbecke (2010)

But: we do not have a basis of the fractional RMP solution (ask Jonas)

Cutting Planes on Master Variables

- Not every valid inequality in $\boldsymbol{\lambda}$ vars can be formulated in $\mathbf{x}$ vars
  $\rightarrow$ only those which arrive via DW decomposition
- We would like to exploit the advantages of the extended formulation
- Many applications have integer *master* variables
  $\rightarrow$ in particular when using discretization

Cutting Planes on Master Variables

Let's try to get a grip on master problem cuts

$$\begin{array}{rrcll}
\min & \sum_{p\in P} c_p\lambda_p + \sum_{r\in R} c_r\lambda_r & & & \\
\text{subject to} & \sum_{p\in P} \mathbf{a}_p\lambda_p + \sum_{r\in R} \mathbf{a}_r\lambda_r & \leq & \mathbf{b} & \\
 & \sum_{p\in P} \mathbf{g}_p\lambda_p + \sum_{r\in R} \mathbf{g}_r\lambda_r & \leq & \mathbf{g} & \text{with duals } \beta \\
 & \sum_{p\in P} \lambda_p & = & 1 & \\
 & \boldsymbol{\lambda} & \in & \mathbb{Z}_+^{|P|+|R|} &
\end{array}$$

How to respect the dual variables in the pricing problem?
= how do we know a new variable's coefficient in a cut?

Cutting Planes on Master Variables

Let's try to get a grip on master problem cuts

$$\begin{array}{rllll} \min & \sum_{p\in P} c_p\lambda_p + \sum_{r\in R} c_r\lambda_r & & & \\ \text{subject to} & \sum_{p\in P} \mathbf{a}_p\lambda_p + \sum_{r\in R} \mathbf{a}_r\lambda_r & \leq & \mathbf{b} & \\ & \sum_{p\in P} \mathbf{g}_p\lambda_p + \sum_{r\in R} \mathbf{g}_r\lambda_r & \leq & \mathbf{g} & \text{with duals } \beta \\ & & \vdots & & \end{array}$$

Pricing: $\min\{\mathbf{cx} - \boldsymbol{\pi} A\mathbf{x} - \beta g(A\mathbf{x}) - \pi_0 \mid \mathbf{x} \in X\}$

let cut coefficients $\mathbf{g}_j$ depend on columns $\mathbf{a}_j$

Cutting Planes on Master Variables

Let's try to get a grip on master problem cuts

$$\begin{array}{rlcl}
\min & \sum_{p\in P} c_p\lambda_p + \sum_{r\in R} c_r\lambda_r & & \\
\text{subject to} & \sum_{p\in P} \mathbf{a}_p\lambda_p + \sum_{r\in R} \mathbf{a}_r\lambda_r & \leq & \mathbf{b} \\
& \sum_{p\in P} \mathbf{g}_p\lambda_p + \sum_{r\in R} \mathbf{g}_r\lambda_r & \leq & \mathbf{g} \qquad \text{with duals } \beta \\
& & \vdots &
\end{array}$$

Pricing: $\min\{\mathbf{cx} - \boldsymbol{\pi} A\mathbf{x} - \beta g(A\mathbf{x}) - \pi_0 \mid \mathbf{x} \in X\}$

let cut coefficients $\mathbf{g}_j$ depend on columns $\mathbf{a}_j$

if there were original vars $\mathbf{y} = g(A\mathbf{x})$ ...

Cutting Planes on Master Variables

Pricing: $\min\{\mathbf{cx} - \boldsymbol{\pi} A\mathbf{x} - \beta g(A\mathbf{x}) - \pi_0 \mid \mathbf{x} \in X\}$

where cut coefficients $\mathbf{g}_j$ depend on $\mathbf{a}_j$

if there were original vars $\mathbf{y} = g(A\mathbf{x})$ ...

Cutting Planes on Master Variables

Pricing: $\min\{\mathbf{cx} - \boldsymbol{\pi} A\mathbf{x} - \beta g(A\mathbf{x}) - \pi_0 \mid \mathbf{x} \in X\}$

where cut coefficients $\mathbf{g}_j$ depend on $\mathbf{a}_j$

if there were original vars $\mathbf{y} = g(A\mathbf{x})$ ...

... we could derive the cuts by DW decomposition from

$$\begin{array}{rrcl} \min & \mathbf{cx} & & \\ \text{subject to} & A\mathbf{x} & \leq & \mathbf{b} \\ & \mathbf{y} & \leq & \mathbf{g} \\ & \mathbf{x} & \in & X \\ & \mathbf{y} & = & g(A\mathbf{x}) \end{array}$$

$\mathbf{y} \leq \mathbf{g}$ remains in the master, $\mathbf{y} = g(A\mathbf{x})$ goes in the pricing

Cutting Planes on Master Variables

Pricing: $\min\{\mathbf{cx} - \boldsymbol{\pi} A\mathbf{x} - \beta g(A\mathbf{x}) - \pi_0 \mid \mathbf{x} \in X\}$

where cut coefficients $\mathbf{g}_j$ depend on $\mathbf{a}_j$

if there were original vars $\mathbf{y} = g(A\mathbf{x})$ ...

... we could derive the cuts by DW decomposition from

$$\begin{array}{rrcl} \min & \mathbf{cx} & & \\ \text{subject to} & A\mathbf{x} & \leq & \mathbf{b} \\ & \mathbf{y} & \leq & \mathbf{g} \\ & \mathbf{x} & \in & X \\ & \mathbf{y} & = & g(A\mathbf{x}) \end{array}$$

$\mathbf{y} \leq \mathbf{g}$ remains in the master, $\mathbf{y} = g(A\mathbf{x})$ goes in the pricing

Et voilà! An extended original formulation!