Recent (and not so recent) Advances in Column Generation

Antonio Frangioni

Dipartimento di Informatica, Università di Pisa

Paris Dauphine

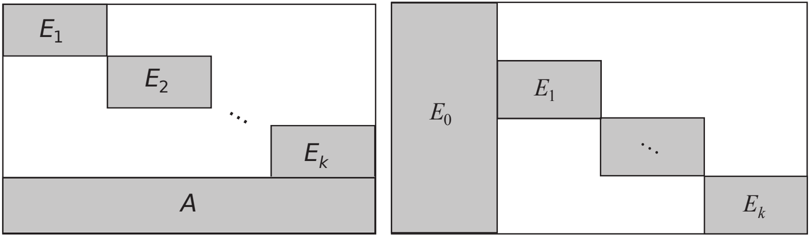
February 10th, 2016

Outline

1. Structured (Integer) Linear Programs
2. Dantzig-Wolfe decomposition, Column Generation
3. Lagrangian decomposition (Dantzig-Wolfe//Column Generation)
4. Stabilization
5. Dual-Optimal Cuts
6. Conclusions

Block-Structured Programs

- (Challenging) applications of Integer Linear Programming are large-scale: millions of variables/constraints
- Good news: all large-scale problems are block-structured
- Usually several nested forms of structure, but two main ones:



block-diagonal $\equiv$ complicating constraints

staircase-structured $\equiv$ complicating variables

- Relaxing constraints / fixing variables yields independent subproblems $\Longrightarrow$ much easier because of size and/or structure (integrality, ...)

Our working example: Multicommodity Network Design

- Graph $G = (N, A)$, a reasonably generic Multicommodity flow model

$$\min \textstyle\sum_{k \in K} \sum_{(i,j) \in A} d^k c^k_{ij} x^k_{ij} + \sum_{(i,j) \in A} f_{ij} y_{ij} \tag{1}$$

$$\sum_{(i,j) \in A} x^k_{ij} - \sum_{(j,i) \in A} x^k_{ji} = \begin{cases} -1 & \text{if } i = s^k \\ 1 & \text{if } i = t^k \\ 0 & \text{otherwise} \end{cases} \qquad i \in N \,,\; k \in K \tag{2}$$

$$\textstyle\sum_{k \in K} d^k x^k_{ij} \leq u_{ij} y_{ij} \qquad (i,j) \in A \tag{3}$$

$$x^k_{ij} \in [0,1] \;/\; \{0,1\} \qquad (i,j) \in A \,,\; k \in K \tag{4}$$

$$y_{ij} \in \mathbb{N} \;/\; \{0,1\} \;/\; \mathbb{R} \qquad (i,j) \in A \tag{5}$$

- $K \equiv$ commodities $\equiv (s^k \,,\, t^k \,,\, d^k)$ (not completely generic)
- Countless many relevant special cases:
  - $f \leq 0 \Longrightarrow$ splittable/nonsplittable multicommodity routing
  - $y_{ij} \in \{0,1\} \Longrightarrow$ almost all graph design problems
  - $\sum_{k \in K} d^k \leq u_{ij}$ + bipartite graph $\Longrightarrow$ uncapacitated facility location
  - multiple node/arc capacities by graph transformations
- Countless many generalizations (extra constraints, nonlinearities, ...)

Multicommodity flow applications

- Pervasive structure in most of combinatorial optimization
- Interesting links with many hard problems (e.g. Max-Cut)
- Very many practical applications: logistic, transportation, telecommunications, energy, ...
- Extremely hard to solve in general: many difficult problems in one
- Very different cases:
  - transportation: very large (often time-space $\Longrightarrow$ acyclic) networks, "few" commodities
  - telecommunications: "small" (undirected) networks, very many ($O(|N|^2)$) commodities
- Even continuous versions "hard": very-large-scale LPs
- Many sources of structure $\Longrightarrow$ the paradise of decomposition [1,2] because both complicating constraints and variables

[1] Ford, Fulkerson "A Suggested Computation for Maximal Multicommodity Network Flows" *Management Science* 1958
[2] Dantzig, Wolfe "The Decomposition Principle for Linear Programs" *Operations Research* 1960

(Very) Classical decomposition approaches

- Lagrangian relaxation [3] of linking constraints:
  - (3): $\Longrightarrow$ flow (shortest path) relaxation
  - (2): $\Longrightarrow$ knapsack relaxation
  - others possible
- Benders' decomposition [4] of linking variables:
  - design (y) variables are "naturally" linking
  - Benders' cuts are metric inequalities defining the multiflow feasibility
  - Linking variables can be artificially added if not present

$$d^k x_{ij}^k \leq u_{ij}^k \qquad , \qquad \sum_{k \in K} u_{ij}^k \leq u_{ij}$$

("resource decomposition") [5]

- We will talk of Lagrange but many things can be extended to Benders

---

[3] Geoffrion "Lagrangean relaxation for integer programming" *Mathematical Programming Study* 1974
[4] Benders "Partitioning procedures for solving mixed-variables programming problems" *Numerische Mathematik* 1962
[5] Kennington, Shalaby "An Effective Subgradient Procedure for Minimal Cost Multicomm. Flow Problems" *Man. Sci.* 1977

Dantzig-Wolfe decomposition Column Generation

A bird's view of "structure"

- The general structure of our models:

$$(\Pi) \qquad \max \{ \, cx \; : \; Ax = b \, , \; x \in X \, \}$$

where we know something about the combinatorial set X:
  - for sure we know some formulation, e.g., $X = \{ \, x \in \mathbb{Z}^n \; : \; Ex \leq d \, \}$
  - linearity used for simplicity, has some (but not paramount) impact, convexity is the issue
  - integrality a common (but not the only) nonconvex component
  - almost always $X = \bigotimes_{h \in \mathcal{K}} X^h$, i.e., $Ax = b$ are linking constraints (note that not always $K = \mathcal{K}$)

A bird's view of "structure"

- The general structure of our models:

$$(\Pi) \qquad \max \{ cx \ : \ Ax = b \ , \ x \in X \}$$

  where we know something about the combinatorial set X:
  - for sure we know some formulation, e.g., $X = \{ x \in \mathbb{Z}^n \ : \ Ex \leq d \}$
  - linearity used for simplicity, has some (but not paramount) impact, convexity is the issue
  - integrality a common (but not the only) nonconvex component
  - almost always $X = \bigotimes_{h \in \mathcal{K}} X^h$, i.e., $Ax = b$ are linking constraints (note that not always $K = \mathcal{K}$)

- What do we know? Perhaps the least possible requirement is we know how to (relatively) efficiently optimize upon X
  - clearly, $X = \bigotimes_{h \in \mathcal{K}} X^h$ helps a lot here
  - special case: we know the best possible convex formulation

$$\tilde{E}x \leq \tilde{d} \quad \text{such that} \quad \{ x \in \mathbb{R}^n \ : \ \tilde{E}x \leq \tilde{d} \} = conv(X)$$

    except, practically all good formulations are too large
  - sometimes $\tilde{E}x \leq \tilde{d}$ can be generated piecemeal, but not always

Dantzig-Wolfe reformulation

- The best possible (convex = solvable) relaxation

$$(\bar{\Pi}) \qquad \max \{ \, cx \; : \; Ax = b \, , \; x \in conv(X) \, \} \tag{6}$$

trivial if "by faces" representation $\tilde{E}$, $\tilde{d}$ known, workable, otherwise?

Dantzig-Wolfe reformulation

- The best possible (convex = solvable) relaxation

$$(\bar{\Pi}) \qquad \max \{ \; cx \; : \; Ax = b \; , \; x \in conv(X) \; \} \tag{6}$$

trivial if "by faces" representation $\tilde{E}$, $\tilde{d}$ known, workable, otherwise?

- Temporarily assume X compact: represent $conv(X)$ by points instead

$$conv(X) = \{ \; x = \sum_{\bar{x} \in X} \bar{x}\theta_{\bar{x}} \; : \; \sum_{\bar{x} \in X} \theta_{\bar{x}} = 1 \; , \; \theta_{\bar{x}} \geq 0 \quad \bar{x} \in X \; \}$$

then reformulate $(\bar{\Pi})$ in terms of the convex multipliers θ

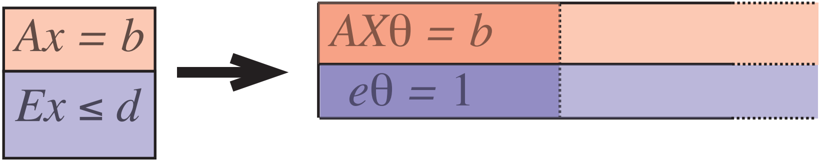
$$(\tilde{\Pi}) \qquad \begin{cases} \max & c \left(\sum_{\bar{x} \in X} \bar{x}\theta_{\bar{x}} \right) \\ & A \left(\sum_{\bar{x} \in X} \bar{x}\theta_{\bar{x}} \right) = b \\ & \sum_{\bar{x} \in X} \theta_{\bar{x}} = 1 \quad , \quad \theta_{\bar{x}} \geq 0 \qquad \bar{x} \in X \end{cases}$$

- Could this ever be a good idea? Actually, it could:
  polyhedra may have few faces and many vertices . . . or vice-versa

n-cube	$\lvert x_i \rvert \leq 1 \quad \forall \; i$	$2n$ faces	2^n vertices
n-co-cube	$\sum_i \lvert x_i \rvert \leq 1$	2^n faces	$2n$ vertices

Dantzig-Wolfe decomposition

- Actually, only the vertices $V \subseteq X$ of $conv(X)$ are required
- Except, most often the number of vertices is too large



- But, if we can efficiently optimize over X, we can generate vertices
- $\mathcal{B} \subset X$ (small), solve master problem $\equiv$ restriction of $(\bar{\Pi})$ with $X \to \mathcal{B}$

$$(\Pi_{\mathcal{B}}) \qquad \max \{ \, cx \; : \; Ax = b \, , \; x \in conv(\mathcal{B}) \, \}$$

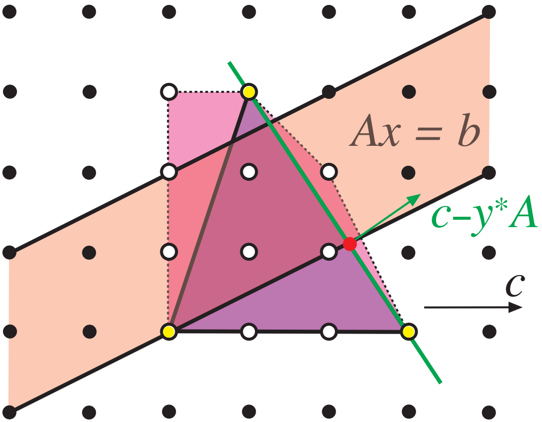
feed (partial) dual optimal solution y^* (of $Ax = b$) to pricing problem

$$(\Pi_{y^*}) \qquad \max \{ \, (c - y^* A)x \; : \; x \in X \, \} \quad [\, + y^* b \,] \tag{7}$$

(why?? you'll see later)

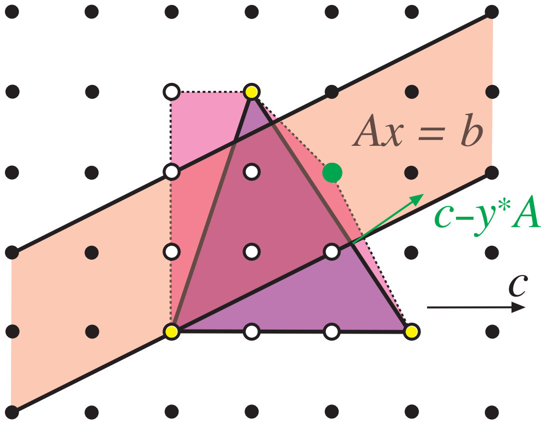
- Use primal optimal solution $\bar{x}$ of (Π_{y^*}) to enlarge $\mathcal{B}$

Geometry of Dantzig-Wolfe decomposition



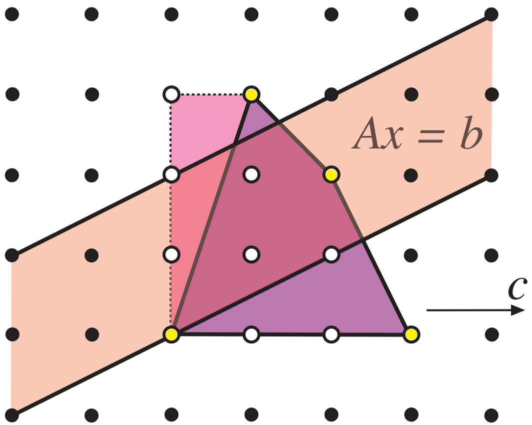
- $c - y^*A$ separates $conv(\mathcal{B}) \cap Ax = b$ from all $x \in X$ better than x^*

Geometry of Dantzig-Wolfe decomposition



- $c - y^*A$ separates $conv(\mathcal{B}) \cap Ax = b$ from all $x \in X$ better than x^*
- Thus, optimizing it allows finding new points (if any)

Geometry of Dantzig-Wolfe decomposition



- $c - y^* A$ separates $conv(\mathcal{B}) \cap Ax = b$ from all $x \in X$ better than x^*
- Thus, optimizing it allows finding new points (if any)
- Issue: $conv(\mathcal{B}) \cap Ax = b$ must be nonempty

Dantzig-Wolfe and Multicommodity flows (MMCF, $y = 1$)

- $y = 1 \equiv$ (MMCF) $\equiv$ continuous problem
- Dantzig-Wolfe reformulation of mutual capacity constraints (3)
- X decomposes in $|K|$ shortest paths (MCFs) $\equiv$ easy (flow relaxation)
- $\mathcal{S} = \{$ (extreme) flows $s = [\bar{x}^{1,s}, \dots, \bar{x}^{k,s}]\,\}$

$$
\begin{array}{lll}
\min & \sum_{s \in \mathcal{S}} \big(\sum_{k \in K} \sum_{(i,j) \in A} c^k_{ij} \bar{x}^{k,s}_{ij} \big) \theta_s & \\
 & \sum_{s \in \mathcal{S}} \big(\sum_{k \in K} \bar{x}^{k,s}_{ij} - u_{ij} \big) \theta_s \leq 0 & (i,j) \in A \\
 & \sum_{s \in \mathcal{S}} \theta_s = 1 \quad , \quad \theta_s \geq 0 & s \in \mathcal{S}
\end{array}
$$

- Gives the same value (bound) as ordinary (arc-flow) formulation
- Competitive with Cplex? 20 years ago maybe, hardly today

Disaggregated Dantzig-Wolfe Reformulation for MMCF

- Another possibility: $X = X^1 \times X^2 \times \ldots \times X^{|K|} \Longrightarrow$ $conv(X) = conv(X^1) \times conv(X^2) \times \ldots \times conv(X^{|K|})$
- In practice: a different multiplier θ^k_s for each $\bar{x}^{k,s}$, with

$$\textstyle\sum_{s \in \mathcal{S}} \theta^k_s = 1 \qquad k \in K$$

(clearly, previous case is $\theta^k_s = \theta^h_s , h \neq k$)

- Simple scaling leads to arc-path formulation:

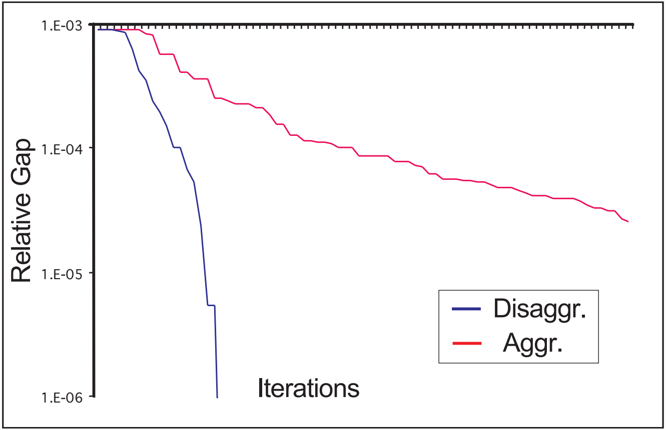
$p \in \mathcal{P}^k = \{ \; s^k\text{–}t^k \text{ paths } \}$, c_p cost, $f_p (= d^k \theta^k_s)$ flow, $\mathcal{P} = \cup_{k \in K} \mathcal{P}^k$

$$\begin{array}{lll}
\min & \sum_{p \in \mathcal{P}} c_p f_p & \\
& \sum_{p \in \mathcal{P} \,:\, (i,j) \in p} f_p \leq u_{ij} & (i,j) \in A \\
& \sum_{p \in \mathcal{P}^k} f_p = d^k & k \in K \\
& f_p \geq 0 & p \in \mathcal{P}
\end{array}$$

- Disaggregated formulation: more columns but sparser, more rows

Disaggregated Dantzig-Wolfe decomposition

- Master problem size $\approx$ time increases, but convergence speed increases a lot $\equiv$ consistent improvement



- Competitive with Cplex? Maybe with a lot of care, but you don't really want to do this for a bit of extra speed

When do you want to do that?

- You are solving the integer problem (Π)

$$(\Pi) \qquad \max \{ \, cx \; : \; Ax = b \, , \; x \in X \subset x \in \mathbb{Z}^n \, \}$$

and the easy path is its continuous relaxation

$$(\underline{\Pi}) \qquad \max \{ \, cx \; : \; Ax = b \, , \; Ex \leq d \, , \; x \in \mathbb{R}^n \, \}$$

- You do DW/CG (perhaps, only) if $(\bar{\Pi})$ is better than $(\underline{\Pi})$ (6) $\equiv$

$$conv(X) = \{ \, x \in \mathbb{R}^n \; : \; \tilde{E}x \leq \tilde{d} \, \} \subset \{ \, x \in \mathbb{R}^n \; : \; Ex \leq d \, \}$$

$\equiv$ the subproblem (7) is not too easy

- "No free lunch" principle: you get nothing for nothing
- The integrality property is your enemy!

Example: FC-MMCF

- You can (sometimes) decide who to treat as linking constraints
- (3) $\equiv$ flow relaxation: subproblems are shortest paths +

$$\min\{\ (f_{ij} - \lambda_{ij})y_{ij}\ :\ y_{ij} \in \{0,1\}\ \}$$

both are integral $\Longrightarrow$ "weak relaxation"

- (2) $\equiv$ knapsack relaxation: subproblems are

$$\begin{aligned} \min \textstyle\sum_{(i,j)\in A}(d^k c_{ij} - \pi_i + \pi_j)x^k_{ij} + f_{ij}y_{ij} & \\ \textstyle\sum_{k\in K} d^k x^k_{ij} \leq u_{ij}y_{ij} & \\ x^k_{ij} \in [0,1] & \qquad k \in K \\ y_{ij} \in \{0,1\} & \end{aligned}$$

$\equiv$ continuous knapsack + 1 binary decision: surprisingly, do not have the integrality property (though definitely "easy")

- Correspond to add the strong forcing constraints $x^k_{ij} \leq y_{ij} \Longrightarrow$ "strong relaxation"

Lagrangian Relaxation, a.k.a. Dantzig-Wolfe decomposition Column Generation

Why does it work? The Lagrangian dual

- Dual of $(\Pi_{\mathcal{B}})$:

$$(\Delta_{\mathcal{B}}) \quad \begin{array}{rl} & \min \left\{ \, yb + v \;:\; v \geq (c - yA)x \quad x \in \mathcal{B} \, \right\} \\ = & \min \left\{ \, f_{\mathcal{B}}(y) = \max \left\{ \, cx + y(b - Ax) \;:\; x \in \mathcal{B} \, \right\} \, \right\} \end{array}$$

(note: $x \in \mathcal{B}$ "constraints index")

- $f_{\mathcal{B}}$ = lower approximation of "true" Lagrangian function (recall (7))

$$f(y) = \max \left\{ \, cx + y(b - Ax) \;:\; x \in X \, \right\}$$

"easy" computability of $f(y)$ the only requirement

- Thus, $(\Delta_{\mathcal{B}})$ outer approximation of the Lagrangian dual

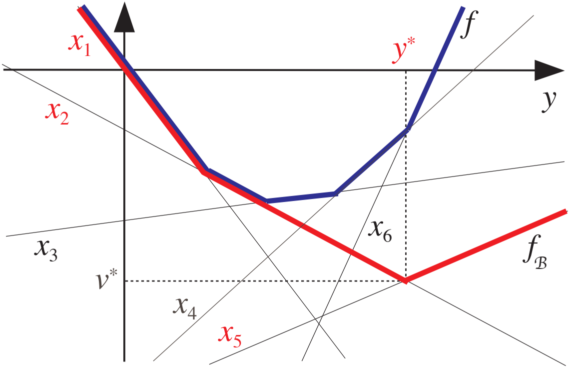
$$(\Delta) \qquad \min \left\{ \, f(y) = \max \left\{ \, cx + y(b - Ax) \;:\; x \in X \, \right\} \, \right\} \tag{8}$$

that is equivalent to $(\bar{\Pi})$, $(\tilde{\Pi})$

- Dantzig-Wolfe decomposition $\equiv$ Cutting Plane approach to (Δ) [6]

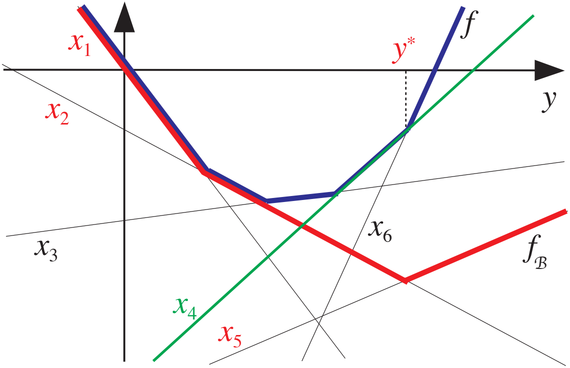
[6] Kelley "The Cutting-Plane Method for Solving Convex Programs" *Journal of the SIAM* 1960

Geometry of the Lagrangian dual



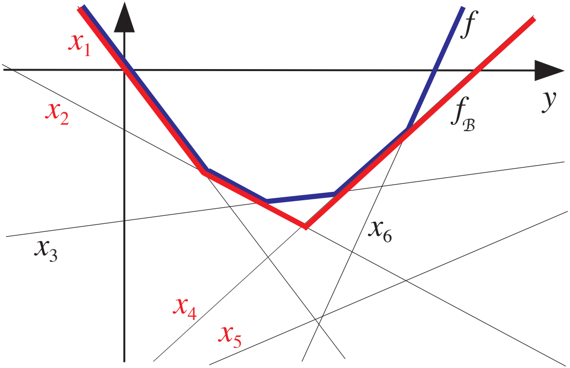
- $v^* = f_{\mathcal{B}}(y^*)$ lower bound on $v(\Pi_{\mathcal{B}})$

Geometry of the Lagrangian dual



- $v^* = f_{\mathcal{B}}(y^*)$ lower bound on $v(\Pi_{\mathcal{B}})$
- Optimal solution $\bar{x}$ gives separator between (v^*, y^*) and *epi f*

Geometry of the Lagrangian dual



- $v^* = f_{\mathcal{B}}(y^*)$ lower bound on $v(\Pi_{\mathcal{B}})$
- Optimal solution $\bar{x}$ gives separator between (v^*, y^*) and *epi f*
- $(c\bar{x}, A\bar{x})$ = new row in $(\Delta_{\mathcal{B}})$ (subgradient of f at y^*)

How to construct the DW reformulation?

- ...do the Lagrangian dual, then simplify
- Exercise: Cutting Stock. $I = \{1, \ldots, n\}$ pieces to cut of length d_i, $R = \{1, \ldots, m\}$ rolls of length L
- Kantorovich formulation (very weak)

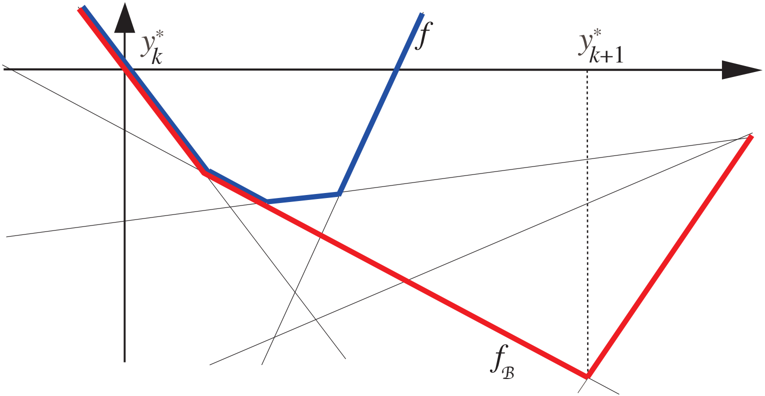
$$
\begin{array}{lr}
\min \sum_{r \in R} y_r & \\
\quad \sum_{r \in R} x_{ir} = 1 & i \in I \\
\quad \sum_{i \in I} d_i x_{ir} \leq L y_r & r \in R \\
\quad x_{ir} \in \{0, 1\} & i \in I \;\;,\;\; r \in R \\
\quad y_r \in \{0, 1\} & r \in R
\end{array}
$$

- Which Lagrangian relaxation? DW Reformulation? A vous ...

Stabilization

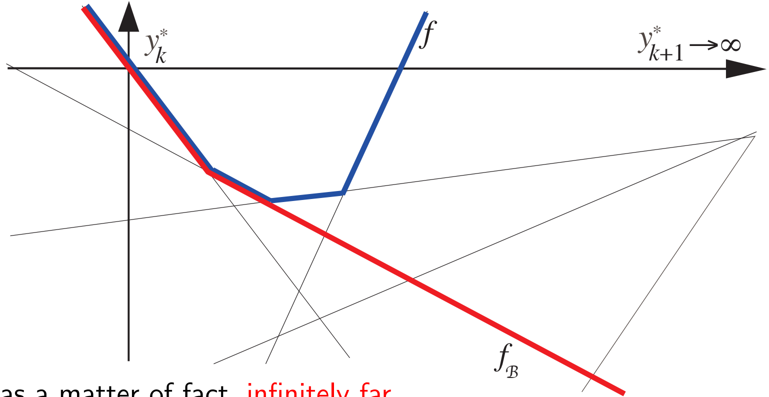
Issue with the approach: instability

- y^*_{k+1} can be very far from y^*_k, where $f_{\mathcal{B}}$ is a "bad model" of f



Issue with the approach: instability

- y^*_{k+1} can be very far from y^*_k, where $f_{\mathcal{B}}$ is a "bad model" of f



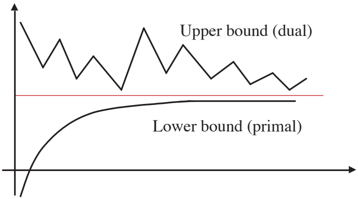
...as a matter of fact, infinitely far

- $(\Pi_{\mathcal{B}})$ empty $\equiv$ $(\Delta_{\mathcal{B}})$ unbounded $\Rightarrow$ Phase 0 / Phase 1 approach
- More in general: $\{y^*_k\}$ is unstable, has no locality properties $\equiv$ convergence speed does not improve near the optimum

The effects of instability

- What does it mean?
  - a good (even perfect) estimate of dual optimum is useless!
  - frequent oscillations of dual values
  - “bad quality” of generated columns

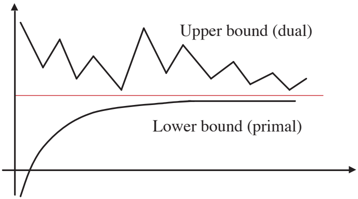
$\Longrightarrow$ tailing off, slow convergence



The effects of instability

- What does it mean?
  - a good (even perfect) estimate of dual optimum is useless!
  - frequent oscillations of dual values
  - "bad quality" of generated columns

$\Longrightarrow$ tailing off, slow convergence



- The solution is pretty obvious: stabilize it
- Gedankenexperiment: starting from known dual optimum, constrain duals in a box of given width

width	time		iter.		columns	
∞	4178.4	%	509	%	37579	%
200.0	835.5	20.0	119	23.4	9368	24.9
20.0	117.9	2.8	35	6.9	2789	7.4
2.0	52.0	1.2	20	3.9	1430	3.8
0.2	47.5	1.1	19	3.7	1333	3.5

Works wonders! . . .

...if only we knew the dual optimum! (which we don't)

- Current point $\bar{y}$, box of size $t > 0$ around it
- Stabilized dual master problem [7]

$$(\Delta_{\mathcal{B},\bar{y},t}) \qquad \min \left\{ f_{\mathcal{B}}(\,\bar{y} + d\,) \;:\; \|\,d\,\|_\infty \le t \,\right\} \tag{9}$$

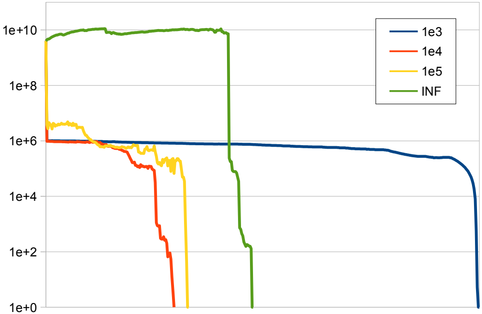
- Corresponding stabilized primal master problem

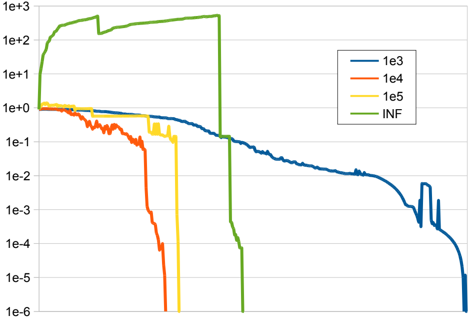
$$(\Pi_{\mathcal{B},\bar{y},t}) \qquad \max \left\{ cx + \bar{y}z - t\|\,z\,\|_1 \;:\; z = b - Ax \;,\; x \in conv(\mathcal{B}) \,\right\} \tag{10}$$

  i.e., just Dantzig-Wolfe with slacks
- When stuck and $z^* = b - Ax^* \neq 0$, either move $\bar{y}$ or enlarge t
- Uses just LP tools, relatively minor modifications
- How should one choose t?
- Does this really work?

[7] Marsten, Hogan, Blankenship "The Boxstep Method for Large-scale Optimization" *Operations Research* 1975

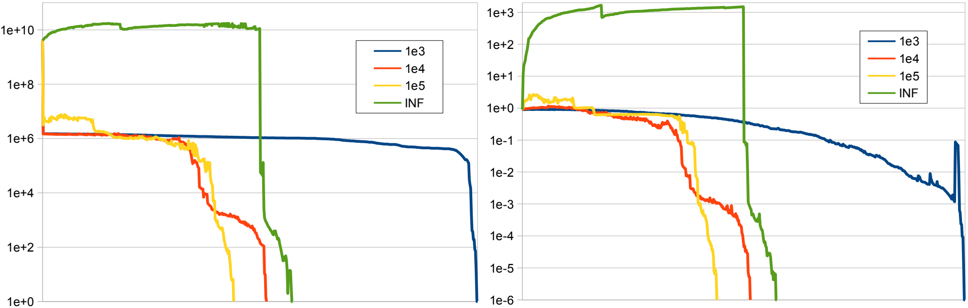
Computational results of the boxstep method (pds7)





- Pure multicommodity flow instance (no y)
- Left $=$ distance from final dual optimum
  right $=$ relative gap with optimal value
- Stabilized with (fixed) different t, un-stabilized ($t = \infty$)
- One can clearly over-stabilize

Computational results of the boxstep method (pds18)

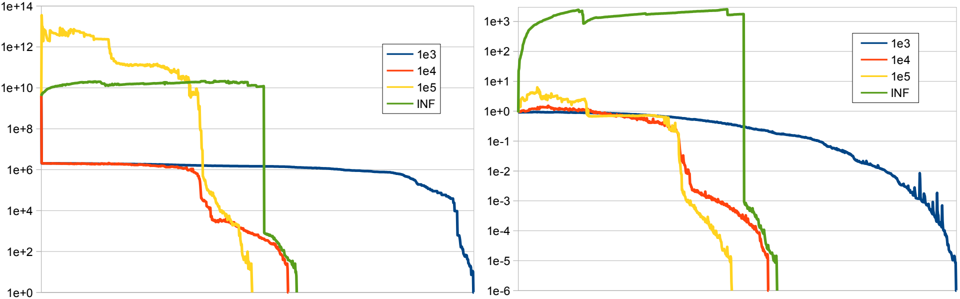


- All cases show a "combinatorial tail" where convergence is very quick
- $t = 1e3$: "smooth but slow" until the combinatorial tail kicks in
  a short-step approach not unlike subgradient methods [8]
- $t = \infty$: apparently trashing along until some magic threshold is hit
- "intermediate" t work best, but pattern not clear

---

[8] Camerini, Fratta, Maffioli "On Improving Relaxation Methods by Modified Gradient Techniques" *Math. Prog. Study* 1975

Computational results of the boxstep method (pds30)



- $t = 1e5$: initially even worse than $t = \infty$ but ends up faster
- Clearly, some on-line tuning of t would be appropriate
- Perhaps a different stabilizing term would help? Why not [9]

$$(\Delta_{\mathcal{B},\bar{y},t}) \qquad \min \left\{ f_{\mathcal{B}}(\, \bar{y} + d \,) + \tfrac{1}{2t} \| \, d \, \|_2^2 \right\}$$

- "Because it's not LP" $\Longrightarrow$ a different duality need be used

[9] Lemaréchal "Bundle Methods in Nonsmooth Optimization" in *Nonsmooth Optimization* vol. 3, Pergamon Press, 1978

Generalized stabilization

- General stabilizing term $\mathcal{D}$, stabilized dual problem

$$(\Delta_{\bar{y},\mathcal{D}}) \qquad \phi_{\mathcal{D}}(\,\bar{y}\,) = \min\left\{\, f(\,\bar{y} + d\,) + \mathcal{D}(\,d\,)\,\right\} \tag{11}$$

- With proper $\mathcal{D}$, $\phi_{\mathcal{D}}$ has same minima as f but is "smoother" (Moreau–Yosida regularization)
- Stabilized primal problem = Fenchel's dual of $(\Delta_{\bar{y},\mathcal{D}})$

$$(\Pi_{\bar{y},\mathcal{D}}) \qquad \min\left\{\, f^*(z) - z\bar{y} + \mathcal{D}^*(-z)\,\right\} \tag{12}$$

- For our dual f, a generalized augmented Lagrangian

$$\max\left\{\, cx + \bar{y}(b - Ax) - \mathcal{D}^*(Ax - b) \;:\; x \in conv(X)\,\right\} \tag{13}$$

- Note: a "primal" exists even for a non-dual f:

$$(\Pi) \qquad \max\{\, -f^*(z) \;:\; z = 0\,\}$$

  $v(\Pi) = -f^*(0) = v(\Delta)$, and $f(\bar{y}) = \max\{\, \bar{y}z - f^*(z)\,\}$, i.e., the Lagrangian relaxation of (Π) w.r.t. $z = 0$

What a stabilizing term may look like?

- General properties (i) and ii) hold for $\mathcal{D} \iff$ they hold for $\mathcal{D}^*$)
  i) $\mathcal{D} \geq 0$ convex, $\mathcal{D}(0) = 0$
  ii) $S_\delta(\mathcal{D})$ compact and full-dimensional $\forall \delta > 0$
  iii) $\mathcal{D}$ differentiable in 0 $\iff$ $\mathcal{D}^*$ strictly convex in 0
  iv) $\mathcal{D}$ is "steep enough" $\Longrightarrow$ $(\Delta_{\bar{y},\mathcal{D}})$ is always bounded
- Actually, iii) and iv) can be relaxed somewhat, albeit at a cost
- A slew of useful consequences (for any f, hence also $f_{\mathcal{B}}$)

$$
\begin{gathered}
-z^* \in \partial\mathcal{D}(d^*) \;\;,\;\; d^* \in \partial\mathcal{D}^*(-z^*) \;\;,\;\; z^* \in \partial f(\bar{x} + d^*) \\
\bar{y} + d^* \in \partial f^*(z^*) \;\;,\;\; \mathcal{D}(d^*) + \mathcal{D}^*(-z^*) = -z^* d^* \\
f(\bar{y} + d^*) + f^*(z^*) = z^*(\bar{y} + d^*)
\end{gathered}
\tag{14}
$$

- Optimal solution d^*: $f(\bar{y} + d^*) < f(\bar{y})$ ($d^* = 0 \Longrightarrow \bar{y}$ optimum)
- $\bar{y} := \bar{y} + d^*$, properly change $\mathcal{D}$ (or not), iterate:
  solving (Δ) by a (generalized) Proximal Point method [10]

[10] Rockafellar "Monotone Operators and the Proximal Point Algorithm" *SIAM J. on Control and Optimization*, 1975

Approximated generalized stabilization

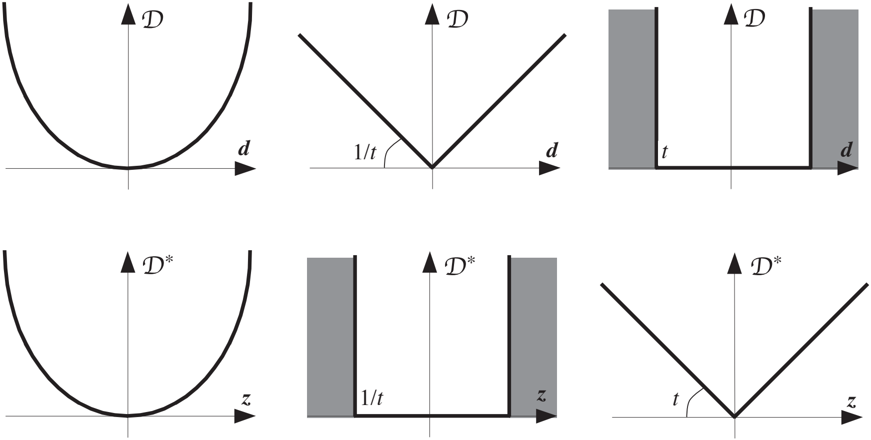
- All nice and well, except $(\Delta_{\bar{y},\mathcal{D}})$ is as difficult to solve as (Δ)
- However, we can solve $(\Delta_{\bar{y},\mathcal{D}})$ by column generation
- Stabilized master problems

$$
\begin{array}{ll}
(\Delta_{\mathcal{B},\bar{y},\mathcal{D}}) & \min \left\{ f_{\mathcal{B}}(\, \bar{y} + d \,) + \mathcal{D}(\, d \,) \right\} \\
(\Pi_{\mathcal{B},\bar{y},\mathcal{D}}) & \max \left\{ cx + \bar{y}(b - Ax) - \mathcal{D}^*(Ax - b) \; : \; x \in conv(\mathcal{B}) \right\}
\end{array}
\tag{15}
$$

- Evaluate $f(\bar{y} + d^*)$, update $\mathcal{B}$, iterate $\Longrightarrow$ solve $(\Delta_{\bar{y},\mathcal{D}})$ / $(\Pi_{\bar{y},\mathcal{D}})$
- Remind: d^* has to ensure descent of f, but we compute $f(\bar{y} + d^*)$ $\Longrightarrow$ early termination: stop as soon as $f(\bar{y} + d^*) \ll f(\bar{y})$
- Overall convergent method for $(\Delta)/(\Pi)$ [11], particularly nifty trick: aggregation. With proper $\mathcal{D}$ ($S_0(\mathcal{D}) = \{0\}$) $\mathcal{B} = \{\, x^* \,\}$ converges (rather slowly [12]: "poorman bundle" $\approx$ volume [13] $\equiv$ subgradient)

[11] F. "Generalized Bundle Methods" *SIAM Journal on Optimization* 2002
[12] Briant, Lemaréchal, et. al. "Comparison of bundle and classical column generation" *Mathematical Programming* 2006
[13] Bahiense, Maculan, Sagastizábal "The volume algorithm revisited: relation with bundle methods" *Math. Prog.* 2002

Classical stabilizing terms



[8] $\mathcal{D} = \frac{1}{2t}\|\cdot\|_2^2$, $\mathcal{D}^* = \frac{1}{2}t\|\cdot\|_2^2$

[14] $\mathcal{D} = \frac{1}{t}\|\cdot\|_1$, $\mathcal{D}^* = I_{B_\infty(1/t)}$

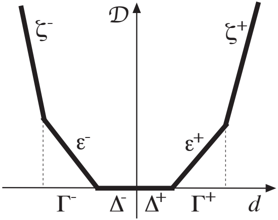
[7] $\mathcal{D} = I_{B_\infty(t)}$, $\mathcal{D}^* = t\|\cdot\|_1$

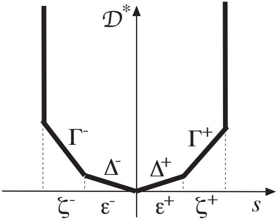
---

[14] Kim, Chang, Lee "A descent method with LP subproblems for nondifferentiable convex optimization" *Math. Prog.* 1995

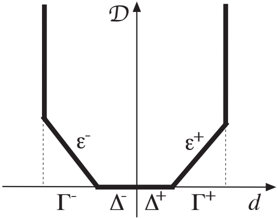
A 5-piecewise-linear function

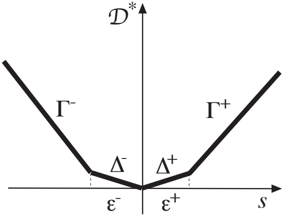
Trust region on $\bar{y}$ + small penalty close + much larger penalty farther [18]





Slightly simplified version: only 3 pieces.





[18] Ben Amor, Desrosiers, F. "On the choice of explicit stabilizing terms in column generation" *Discrete Applied Math.* 2009

A 5-piecewise-linear function (cont.d)

$\mathcal{D}(u) = \sum_{i=1}^{m} \mathcal{D}_i(u_i)$ where

$$\mathcal{D}_i(u_i) = \begin{cases} -(\zeta_i^- + \varepsilon_i^-)(u_i + \Gamma_i^-) - \zeta_i^- \Delta_i^- & \text{if} \quad -\infty \le u_i \le -\Gamma_i^- - \Delta_i^- \\ -\varepsilon_i^-(u_i - \Delta_i^-) & \text{if} \quad -\Gamma_i^- - \Delta_i^- \le u_i \le -\Delta_i^- \\ 0 & \text{if} \quad -\Delta_i^- \le u_i \le \Delta_i^+ \\ +\varepsilon_i^+(u_i - \Delta_i^+) & \text{if} \quad \Delta_i^+ \le u_i \le \Delta_i^+ + \Gamma_i^+ \\ +(\varepsilon_i^+ + \zeta_i^+)(u_i - \Gamma_i^+) - \zeta_i^+ \Delta_i^+ & \text{if} \quad \Delta_i^+ + \Gamma_i^+ \le u_i \le +\infty \end{cases}$$

$\mathcal{D}^*(z) = \sum_{i=1}^{m} \mathcal{D}_i^*(z_i)$ where

$$\mathcal{D}_i^*(z_i) = \begin{cases} +\infty & \text{if} \quad z_i < -(\zeta_i^- + \varepsilon_i^-) \\ -(\Gamma_i^- + \Delta_i^-)z_i - \Gamma_i^- \varepsilon_i^- & \text{if} \quad -\zeta_i^- - \varepsilon_i^- \le z_i \le -\varepsilon_i^- \\ -\Delta_i^- z_i & \text{if} \quad -\varepsilon_i^- \le z_i \le 0 \\ +\Delta_i^+ z_i & \text{if} \quad 0 \le z_i \le \varepsilon_i^- \\ +(\Gamma_i^+ + \Delta_i^+)z_i - \Gamma_i^+ \varepsilon_i^+ & \text{if} \quad \varepsilon_i^+ \le z_i \le (\zeta_i^+ + \varepsilon_i^+) \\ +\infty & \text{if} \quad z_i > (\zeta_i^+ + \varepsilon_i^+) \end{cases}$$

Interval widths $\Longleftrightarrow$ penalties

A 5-piecewise-linear master problem

$$
(\Pi_{\mathcal{B},\bar{y},\mathcal{D}}) \quad \left\{ \begin{array}{ll} \max & c\left(\sum_{\bar{x}\in\mathcal{B}} \bar{x}\theta_{\bar{x}} \right) - \bar{y}\,(s^- + w^- - w^+ - s^+) \\ & \qquad + \gamma^- s^- + \delta^- w^- + \delta^+ w^+ + \gamma^+ s^+ \\ & A\left(\sum_{\bar{x}\in\mathcal{B}} \bar{x}\theta_{\bar{x}} \right) + s^- + w^- - w^+ - s^+ = b \\ & \sum_{\bar{x}\in\mathcal{B}} \theta_{\bar{x}} = 1 \;\; , \;\; \theta_{\bar{x}} \;\; \geq 0 \quad \bar{x} \in \mathcal{B} \\ & 0 \leq s^- \leq \zeta^- \;\; , \;\; 0 \leq s^+ \leq \zeta^+ \\ & 0 \leq w^- \leq \varepsilon^- \;\; , \;\; 0 \leq w^+ \leq \varepsilon^+ \end{array} \right.
$$

A 5-piecewise-linear master problem

$$
(\Pi_{\mathcal{B},\bar{y},\mathcal{D}}) \quad \left\{
\begin{array}{ll}
\max & c\left(\sum_{\bar{x}\in\mathcal{B}} \bar{x}\theta_{\bar{x}} \right) - \bar{y}\,(s^- + w^- - w^+ - s^+) \\
 & \qquad\qquad + \gamma^- s^- + \delta^- w^- + \delta^+ w^+ + \gamma^+ s^+ \\
 & A\left(\sum_{\bar{x}\in\mathcal{B}} \bar{x}\theta_{\bar{x}} \right) + s^- + w^- - w^+ - s^+ = b \\
 & \sum_{\bar{x}\in\mathcal{B}} \theta_{\bar{x}} = 1 \;,\; \theta_{\bar{x}} \geq 0 \quad \bar{x} \in \mathcal{B} \\
 & 0 \leq s^- \leq \zeta^- \;,\; 0 \leq s^+ \leq \zeta^+ \\
 & 0 \leq w^- \leq \varepsilon^- \;,\; 0 \leq w^+ \leq \varepsilon^+
\end{array}
\right.
$$

- Same constraints as $(\Pi_{\mathcal{B}})$, 4 slack variables for each constraint
- Many parameters: *widths* $\Gamma^{\pm}$ and $\Delta^{\pm}$, *penalties* $\zeta^{\pm}$ and $\varepsilon^{\pm}$, different roles for small and large penalties
- Large penalties $\zeta^{\pm}$ easily make $(\Delta_{\mathcal{B},\bar{y},\mathcal{D}})$ bounded $\Longrightarrow$ no Phase 0
- 3-pieces: either large penalty $\Longrightarrow$ small moves, or
  small penalty $\Longrightarrow$ instability

Some computational results

- Comparing unstabilized, 5-piecewise and 3-piecewise penalty functions
- State-of-the-art `GenCol` code, large-scale, difficult MDVS instances (only root relaxation times)

		p1	p2	p3	p4	p5	p6	p7	p8	p9	p10
time	CG	139	177	235	159	3138	3966	3704	1742	3685	3065
	PP-3	80	84	103	70	1173	819	1440	1143	1787	2283
	PP-5	33	36	38	28	482	335	946	572	1065	2037
iter	CG	117	149	200	165	408	524	296	186	246	247
	PP-3	82	92	104	75	181	129	134	145	144	189
	PP-5	47	47	49	45	93	64	98	83	86	150
mpt	CG	88	125	165	105	1679	2004	1955	925	1984	1743
	PP-3	44	47	60	42	572	399	740	543	858	1351
	PP-5	13	16	17	10	189	128	428	257	542	1326

- 5-pieces better than 3-pieces, 5-then-3 even better
- Quadratic more "stable", but optimized 5-pieces always better (quadratic has far less parameters, easier but less flexible)
- All this with fixed parameters, on-line adjustment possible (?)

On unboundedness and early termination

- A ray r of X: $x \in X \Longrightarrow x + \lambda r \in X$ for infinitely large λ
- $(c - yA)r > 0 \Rightarrow f(y) = +\infty \Rightarrow$ constraint $cr \leq y(Ar)$ in dual space

$$(\Delta) \quad \min\{\ f(y)\ :\ y \in Y\ \}$$

  where facets of Y are dynamically generated like ordinary columns (constraint = column with a 0 in the convexity constraint)
- One might even hide the convexity constraint:
  - $A\bar{x} \rightarrow \left[\ A\bar{x}\ ,\ 1\ \right] \quad , \quad b \rightarrow \left[\ b\ ,\ 1\ \right]$;
  - Ignoring the special role of v (just another y)
  - Advantage: everything is a constraint

On unboundedness and early termination

- A ray r of X: $x \in X \Longrightarrow x + \lambda r \in X$ for infinitely large λ
- $(c - yA)r > 0 \Rightarrow f(y) = +\infty \Rightarrow$ constraint $cr \leq y(Ar)$ in dual space

$$(\Delta) \quad \min\{\ f(y)\ :\ y \in Y\ \}$$

  where facets of Y are dynamically generated like ordinary columns (constraint = column with a 0 in the convexity constraint)
- One might even hide the convexity constraint:
  - $A\bar{x} \rightarrow [\ A\bar{x}\ ,\ 1\ ] \quad , \quad b \rightarrow [\ b\ ,\ 1\ ]$;
  - Ignoring the special role of v (just another y)
  - Advantage: everything is a constraint

$\boxed{\text{This is a bad idea!}}$

On unboundedness and early termination

- A ray r of X: $x \in X \Longrightarrow x + \lambda r \in X$ for infinitely large λ
- $(c - yA)r > 0 \Rightarrow f(y) = +\infty \Rightarrow$ constraint $cr \leq y(Ar)$ in dual space

$$(\Delta) \quad \min\{ \, f(y) \; : \; y \in Y \, \}$$

  where facets of Y are dynamically generated like ordinary columns (constraint = column with a 0 in the convexity constraint)

- One might even hide the convexity constraint:
  - $A\bar{x} \to \left[\, A\bar{x} \, , \, 1 \, \right] \quad , \quad b \to \left[\, b \, , \, 1 \, \right]$;
  - Ignoring the special role of v (just another y)
  - Advantage: everything is a constraint

$\boxed{\text{This is a bad idea!}}$

- Approximate stabilization = testing for decrease in f-value, but when a ray is generated, $f(\bar{y} + d^*) = +\infty$
- Convexity constraints are good: invent them if they are not there

Bundle vs. Proximal Point

- Same computational setting as before
- Comparing the same stabilization (5-piecewise) with (BP) or without (PP) early termination

		p1	p2	p3	p4	p5	p6	p7	p8	p9	p10
time	CG	139	177	235	159	3138	3966	3704	1742	3685	3065
	PP	33	36	38	28	482	335	946	572	1065	2037
	BP	26	28	35	21	295	257	639	352	545	1505
iter	CG	117	149	200	165	408	524	296	186	246	247
	PP	47	47	49	45	93	64	98	83	86	150
	BP	37	43	44	36	57	53	59	49	51	101
mpt	CG	88	125	165	105	1679	2004	1955	925	1984	1743
	PP	13	16	17	10	189	128	428	257	542	1326
	BP	10	14	15	10	100	70	329	206	334	983

- Stabilization works well, approximate stabilization works better

Dual-Optimal Cuts

Dual-Optimal Cuts

- Stabilizing = restricting the dual space
- The above approaches need stability center $\bar{y}$, to be updated: it'd be nice if we could do without
- Simple observation: dual constraints = primal variables
  $\Longrightarrow$ need to add even more variables to the primal

  ... in such a way that not all dual optimal solution are cut

Dual-Optimal Cuts

- Stabilizing = restricting the dual space
- The above approaches need stability center $\bar{y}$, to be updated: it'd be nice if we could do without
- Simple observation: dual constraints = primal variables
  $\Longrightarrow$ need to add even more variables to the primal

  . . . in such a way that not all dual optimal solution are cut
- Actually quite simple:
  the new variables must not add new primal solutions [19]

---

[19] Ben Amor, Desrosiers, Valério de Carvalho "Dual-optimal inequalities for stabilized column generation" *Op. Res.* 2006

Dual-Optimal Cuts for Multicommodity flows

- $\mathcal{C}$ = directed circuits with one reversed arc (aggregated flow)
- Constraints become

$$\sum_{p \in \mathcal{P}\,:\,(i,j) \in p} f_p + \sum_{c \in \mathcal{C}\,:\,(i,j) \in c} \pm f_c \leq u_{ij}$$

where "$-$" if (i,j) is reversed in c; hence, one also needs

$$0 \leq \sum_{p \in \mathcal{P}\,:\,(i,j) \in p} f_p + \sum_{c \in \mathcal{C}\,:\,(i,j) \in c} \pm f_c$$

- Any feasible solution to the extended model can be converted into a feasible solution to the original model

Dual-Optimal Cuts for Multicommodity flows

- $\mathcal{C}$ = directed circuits with one reversed arc (aggregated flow)
- Constraints become

$$\sum_{p \in \mathcal{P} \,:\, (i,j) \in p} f_p + \sum_{c \in \mathcal{C} \,:\, (i,j) \in c} \pm f_c \leq u_{ij}$$

where "$-$" if (i,j) is reversed in c; hence, one also needs

$$0 \leq \sum_{p \in \mathcal{P} \,:\, (i,j) \in p} f_p + \sum_{c \in \mathcal{C} \,:\, (i,j) \in c} \pm f_c$$

- Any feasible solution to the extended model can be converted into a feasible solution to the original model
- $|\mathcal{C}| \in O(n^2)$ if G is planar, all-pairs SPT pricing otherwise
- Promising initial results
- Other applications: Cutting Stock (different cuts)

Conclusions

Conclusions and (a lot of) future work

- Multicommodity flows is a veritable mine of structures
- DW decomposition is a very old idea, very well-understood
- Yet, by-the-book decomposition is often not effective enough
- Many possible ideas to improve on the standard approach
- Substantial issue: what works is "large" master problems so that "combinatorial tail" kicks in very quickly $\Longrightarrow$
  - Large master problem time
  - "Unstructured" master problems $\Longrightarrow$ general-purpose solvers
  - "Complicated" $\Longrightarrow$ costly stabilizing functions (at least $\|\cdot\|_2^2$)
  - Need to find modern equivalent of [23] to exploit the structure of an unstructured problem (perhaps less contradictory than is sounds [32])
- Huge challenge: make these techniques mainstream
- A new hope: automatic reformulation techniques [33]

[32] Kiwiel "An alternating linearization bundle method for ... and nonlinear multicommodity flow problems" *Math. Prog.* 2013

[33] F., Perez Sanchez "Transforming mathematical models using declarative reformulation rules" *LNCS* 6683, 2011