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
(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
- Staircase-structured

Relaxing constraints / fixing variables yields independent subproblems

\[\Rightarrow\] 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

$$
\begin{align*}
\text{min} \left[ \sum_{k \in K} \sum_{(i,j) \in A} \mathbf{d}^k c_{ij}^k x_{ij}^k + \sum_{(i,j) \in A} f_{ij} y_{ij} \right] \\
\sum (i,j) \in A \quad x_{ij}^k - \sum (j,i) \in A \quad x_{ji}^k = \begin{cases} 
-1 & \text{if } i = s_k \\
1 & \text{if } i = t_k \\
0 & \text{otherwise}
\end{cases} \quad i \in N, k \in K \\
\sum_{k \in K} \mathbf{d}^k x_{ij}^k \leq u_{ij} y_{ij} \quad (i,j) \in A \\
x_{ij}^k \in [0, 1] / \{0, 1\} \quad (i,j) \in A, k \in K \\
y_{ij} \in \mathbb{N} / \{0, 1\} / \mathbb{R} \quad (i,j) \in A
\end{align*}
$$

- $K \equiv$ commodities $\equiv (s^k, t^k, d^k)$ (not completely generic)

- Countless many relevant special cases:
  - $f \leq 0 \implies$ splittable/nonsplittable multicommodity routing
  - $y_{ij} \in \{0, 1\} \implies$ almost all graph design problems
  - $\sum_{k \in K} \mathbf{d}^k \leq u_{ij} + \text{bipartite graph} \implies$ 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 $\rightarrow$ 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 $\rightarrow$ the paradise of decomposition** [1,2] because both complicating constraints and variables

---

(Very) Classical decomposition approaches

- **Lagrangian relaxation** [3] of linking constraints:
  - (3): $\Rightarrow$ flow (shortest path) relaxation
  - (2): $\Rightarrow$ 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
    \[
    d^k x^k_{ij} \leq u^k_{ij} \quad , \quad \sum_{k \in K} u^k_{ij} \leq u_{ij}
    \]
    (“resource decomposition”) [5]

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

---

Dantzig-Wolfe decomposition
Column Generation
A bird’s view of “structure”

- The general structure of our models:
  \[
  \max \{ cx : Ax = b, x \in X \}
  \]
  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:
  \[
  (\prod) \quad \max \{ \, cx : \, Ax = b \, , \, x \in X \, \} 
  \]
  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
    \[
    \tilde{E}x \leq \tilde{d} \quad \text{such that} \quad \{ \, x \in \mathbb{R}^n : \, \tilde{E}x \leq \tilde{d} \, \} = \text{conv}(X) 
    \]
    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

\[
\max \{ \; c x : \; A x = b , \; x \in \text{conv}(X) \; \} \tag{6}
\]

trivial if “by faces” representation \( \tilde{E}, \tilde{d} \) known, workable, otherwise?

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

\(|x_i| \leq 1 \forall i\), \(2^n \) faces, \(2^n \) vertices on \( n \)-cube

\(\sum |x_i| \leq 1 \), \(2^n \) faces, \(2^n \) vertices on \( n \)-co-cube

A. Frangioni (DI — UniPi)
Dantzig-Wolfe reformulation

- The best possible (convex = solvable) relaxation

\[
\max \{ \, cx : A x = b , \, x \in \text{conv}(X) \, \} \tag{6}
\]

trivial if “by faces” representation \( \tilde{E} \), \( \tilde{d} \) known, workable, otherwise?

- Temporarily assume \( X \) compact: represent \( \text{conv}(X) \) by points instead

\[
\text{conv}(X) = \{ \, x = \sum_{\bar{x} \in X} \bar{x} \theta_{\bar{x}} : \sum_{\bar{x} \in X} \theta_{\bar{x}} = 1 , \, \theta_{\bar{x}} \geq 0 \, \bar{x} \in X \, \}
\]

then reformulate \((\tilde{\Pi})\) in terms of the convex multipliers \( \theta \)

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

- Could this ever be a good idea? Actually, it could:

polyhedra may have few faces and many vertices . . . or vice-versa

- \( n \)-cube: \(|x_i| \leq 1 \ \forall \ i\) \( 2n \) faces \( 2^n \) vertices

- \( n \)-co-cube: \( \sum_i |x_i| \leq 1 \) \( 2^n \) faces \( 2n \) vertices
Dantzig-Wolfe decomposition

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

But, if we can efficiently optimize over $X$, we can generate vertices.

Lead $B \subset X$ (small), solve master problem $\equiv$ restriction of (\(\bar{\Pi}\)) with $X \rightarrow B$

\[
(\Pi_B) \quad \max \{ cx : Ax = b, x \in \text{conv}(B) \}
\]

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

\[
(\Pi_{y^*}) \quad \max \{ (c - y^*A)x : x \in X \} \quad [ + y^*b ]
\]  \hspace{1cm} (7)

(why?? you’ll see later)

Use primal optimal solution $\bar{x}$ of (\(\Pi_{y^*}\)) to enlarge $B$.
Geometry of Dantzig-Wolfe decomposition

$c - y^*A$ separates $\text{conv}(B) \cap Ax = b$ from all $x \in X$ better than $x^*$
Geometry of Dantzig-Wolfe decomposition

- $c - y^*A$ separates $\text{conv}(\mathcal{B}) \cap Ax = b$ 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 $\text{conv}(\mathcal{B}) \cap Ax = b$ from all $x \in X$ better than $x^*$
- Thus, optimizing it allows finding new points (if any)
- Issue: $\text{conv}(\mathcal{B}) \cap Ax = b$ must be nonempty
Dantzig-Wolfe and Multicommodity flows (MMCF, \( y = 1 \))

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

\[
\begin{align*}
\min & \quad \sum_{s \in S} \left( \sum_{k \in K} \sum_{(i,j) \in A} c_{ij}^k \tilde{x}_{ij}^{k,s} \right) \theta_s \\
\text{s.t.} & \quad \sum_{s \in S} \left( \sum_{k \in K} \tilde{x}_{ij}^{k,s} - u_{ij} \right) \theta_s \leq 0 \quad (i,j) \in A \\
& \quad \sum_{s \in S} \theta_s = 1, \quad \theta_s \geq 0 \quad s \in S
\end{align*}
\]

- 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: $X = X^1 \times X^2 \times \ldots \times X^{|K|} \Rightarrow \text{conv}(X) = \text{conv}(X^1) \times \text{conv}(X^2) \times \ldots \times \text{conv}(X^{|K|})$

- In practice: a different multiplier $\theta_k^s$ for each $\bar{x}^k,s$, with

$$\sum_{s \in S} \theta_k^s = 1 \quad k \in K$$

(clearly, previous case is $\theta_k^s = \theta_h^s, h \neq k$)

- Simple scaling leads to arc-path formulation:

$$p \in \mathcal{P}^k = \{ s^k-t^k \text{ paths } \}, \quad c_p \text{ cost}, \quad f_p(= d^k \theta_s^k) \text{ flow}, \quad \mathcal{P} = \bigcup_{k \in K} \mathcal{P}^k$$

$$\min \sum_{p \in \mathcal{P}} c_p f_p$$

$$\sum_{p \in \mathcal{P}} : (i,j) \in p \quad f_p \leq u_{ij} \quad (i,j) \in A$$

$$\sum_{p \in \mathcal{P}^k} f_p = d^k \quad k \in K$$

$$f_p \geq 0 \quad p \in \mathcal{P}$$

- 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 \((\Pi)\)

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

and the easy path is its continuous relaxation

\[(\bar{\Pi}) \quad \max \{ \, cx : Ax = b , \; Ex \leq d , \; x \in \mathbb{R}^n \, \}

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

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

\(\equiv\) 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 \(\implies\) “weak relaxation”

- (2) \(\equiv\) knapsack relaxation: subproblems are
  \[
  \min \sum_{(i,j) \in A} (d^k c_{ij} - \pi_i + \pi_j)x^k_{ij} + f_{ij}y_{ij} \\
  \sum_{k \in K} d^k x^k_{ij} \leq u_{ij}y_{ij} \\
  x^k_{ij} \in [0, 1] \quad k \in K \\
  y_{ij} \in \{0, 1\}
  \]
  \(\equiv\) continuous knapsack + 1 binary decision: surprisingly, do not have the integrality property (though definitely “easy”)

- Correspond to add the strong forcing constraints \(x^k_{ij} \leq y_{ij} \implies\) “strong relaxation”
Lagrangian Relaxation, a.k.a. Dantzig-Wolfe decomposition
Column Generation
Why does it work? The Lagrangian dual

- Dual of $(\Pi_B)$:

\[
\begin{align*}
(\Delta_B) & \quad \min \left\{ yb + v : v \geq (c - yA)x \quad x \in B \right\} \\
& = \min \left\{ f_B(y) = \max \left\{ cx + y(b - Ax) : x \in B \right\} \right\ }
\end{align*}
\]

(note: $x \in B$ “constraints index”)

- $f_B = \text{lower approximation}$ 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, $(\Delta_B)$ outer approximation of the Lagrangian dual

\[
(\Delta) \quad \min \left\{ f(y) = \max \left\{ cx + y(b - Ax) : x \in X \right\} \right\}
\]

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

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

Geometry of the Lagrangian dual

\[ v^* = f_B(y^*) \text{ lower bound on } v(\Pi_B) \]
Geometry of the Lagrangian dual

- $v^* = f_B(y^*)$ lower bound on $v(\Pi_B)$
- Optimal solution $\bar{x}$ gives separator between $(v^*, y^*)$ and $\text{epi } f$
Geometry of the Lagrangian dual

\[ v^* = f_B(y^*) \] lower bound on \( v(\Pi_B) \)

Optimal solution \( \bar{x} \) gives separator between \( (v^*, y^*) \) and \( epi f \)

\( (c\bar{x}, A\bar{x}) = \) new row in \( (\Delta_B) \) (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)

\[
\begin{align*}
\min & \sum_{r \in R} y_r \\
\text{s.t.} & \sum_{r \in R} x_{ir} = 1 \quad i \in I \\
& \sum_{i \in I} d_i x_{ir} \leq L y_r \quad r \in R \\
& x_{ir} \in \{0, 1\} \quad i \in I, \ r \in R \\
& y_r \in \{0, 1\} \quad r \in R
\end{align*}
\]

- Which Lagrangian relaxation? DW Reformulation? A vous ...
Stabilization
Issue with the approach: instability

- $y^*_{k+1}$ can be very far from $y^*_k$, where $f_B$ is a “bad model” of $f$
Issue with the approach: instability

- $y_{k+1}^*$ can be very far from $y_k^*$, where $f_B$ is a “bad model” of $f$

...as a matter of fact, infinitely far

- $(\Pi_B)$ empty $\equiv (\Delta_B)$ 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

→ 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

⇒ 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

<table>
<thead>
<tr>
<th>width</th>
<th>time</th>
<th>iter.</th>
<th>columns</th>
</tr>
</thead>
<tbody>
<tr>
<td>∞</td>
<td>4178.4%</td>
<td>509%</td>
<td>37579%</td>
</tr>
<tr>
<td>200.0</td>
<td>835.5%</td>
<td>119%</td>
<td>9368%</td>
</tr>
<tr>
<td>20.0</td>
<td>117.9%</td>
<td>35%</td>
<td>2789%</td>
</tr>
<tr>
<td>2.0</td>
<td>52.0%</td>
<td>20%</td>
<td>1430%</td>
</tr>
<tr>
<td>0.2</td>
<td>47.5%</td>
<td>19%</td>
<td>1333%</td>
</tr>
</tbody>
</table>

Works wonders! ...
Stabilizing column generation

...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]

$$(\Delta_{\mathcal{B},\bar{y},t}) \quad \min \{ f_{\mathcal{B}}(\bar{y} + d) : \|d\|_{\infty} \leq t \} \quad (9)$$

- Corresponding stabilized primal master problem

$$(\Pi_{\mathcal{B},\bar{y},t}) \quad \max \{ cx + \bar{y}z - t\|z\|_{1} : z = b - Ax, \ x \in \text{conv}(\mathcal{B}) \} \quad (10)$$

e.g., 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?

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
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 = ∞: apparently trashing along until some magic threshold is hit

“intermediate” t work best, but pattern not clear

• $t = 1e5$: 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]

$$(\Delta_B, \tilde{y}, t) \min \left\{ f_B(\tilde{y} + d) + \frac{1}{2t} \| d \|_2^2 \right\}$$

• “Because it’s not LP” $\implies$ a different duality need be used

Generalized stabilization

- General stabilizing term $\mathcal{D}$, stabilized dual problem
  \[
  (\Delta \bar{y}, \mathcal{D}) \quad \phi_{\mathcal{D}}(\bar{y}) = \min \left\{ f(\bar{y} + d) + \mathcal{D}(d) \right\}
  \]  
  (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 $(\Delta \bar{y}, \mathcal{D})$
  \[
  (\Pi \bar{y}, \mathcal{D}) \quad \min \left\{ f^*(z) - z\bar{y} + \mathcal{D}^*(-z) \right\}
  \]  
  (12)

- For our dual $f$, a generalized augmented Lagrangian
  \[
  \max \left\{ cx + \bar{y}(b - Ax) - \mathcal{D}^*(Ax - b) \mid x \in \text{conv}(X) \right\}
  \]  
  (13)

- Note: a “primal” exists even for a non-dual $f$:
  \[
  (\Pi) \quad \max\{-f^*(z) : z = 0\}
  \]
  $v(\Pi) = -f^*(0) = v(\Delta)$, and $f(\bar{y}) = \max\{\bar{y}z - f^*(z)\}$, i.e., the Lagrangian relaxation of $(\Pi)$ w.r.t. $z = 0$
What a stabilizing term may look like?

- General properties (i) and ii) hold for \( D \iff \text{they hold for } D^* \)
  
  i) \( D \geq 0 \) convex, \( D(0) = 0 \)
  
  ii) \( S_\delta(D) \) compact and full-dimensional \( \forall \delta > 0 \)
  
  iii) \( D \) differentiable in \( 0 \) \( \iff \) \( D^* \) strictly convex in \( 0 \)
  
  iv) \( D \) is “steep enough” \( \implies (\Delta \bar{y}, D) \) 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 \( f_B \))

\[
-z^* \in \partial D(d^*) \ , \ d^* \in \partial D^*(-z^*) \ , \ z^* \in \partial f(\bar{x} + d^*)
\]

\[
\bar{y} + d^* \in \partial f^*(z^*) \ , \ D(d^*) + D^*(-z^*) = -z^*d^*
\]

\[
f(\bar{y} + d^*) + f^*(z^*) = z^*(\bar{y} + d^*)
\]  

(14)

- Optimal solution \( d^* \): \( f(\bar{y} + d^*) < f(\bar{y}) \) \( (d^* = 0 \implies \bar{y} \text{ optimum}) \)

- \( \bar{y} := \bar{y} + d^* \), properly change \( D \) (or not), iterate:
  solving \( (\Delta) \) by a (generalized) Proximal Point method [10]

Approximated generalized stabilization

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

\[
\begin{align*}
(\Delta \mathcal{B}, \bar{y}, \mathcal{D}) & \min \left\{ f_{\mathcal{B}}(\bar{y} + d) + \mathcal{D}(d) \right\} \\
(\Pi \mathcal{B}, \bar{y}, \mathcal{D}) & \max \left\{ cx + \bar{y}(b - Ax) - \mathcal{D}^*(Ax - b) : x \in \text{conv}(\mathcal{B}) \right\}
\end{align*}
\]  
(15)

- Evaluate \(f(\bar{y} + d^*)\), update \(\mathcal{B}\), iterate \(\implies\) solve \((\Delta \bar{y}, \mathcal{D}) / (\Pi \bar{y}, \mathcal{D})\)

- Remind: \(d^*\) has to ensure descent of \(f\), but we compute \(f(\bar{y} + d^*)\)

\(\implies\) 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)

Classical stabilizing terms

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


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

[14] \[ D = l_{B_\infty}(t) \]
\[ D^* = t \| \cdot \|_1 \]

[7]
A 5-piecewise-linear function

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

Slightly simplified version: only 3 pieces.

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

\[ D(u) = \sum_{i=1}^{m} D_i(u_i) \]

where

\[ D_i(u_i) = \begin{cases} 
-(\zeta_i^- + \varepsilon_i^-)(u_i + \Gamma_i^-) - \zeta_i^- \Delta_i^- & \text{if } -\infty \leq u_i \leq -\Gamma_i^- - \Delta_i^- \\
-\varepsilon_i^- (u_i - \Delta_i^-) & \text{if } -\Gamma_i^- - \Delta_i^- \leq u_i \leq -\Delta_i^- \\
0 & \text{if } -\Delta_i^- \leq u_i \leq \Delta_i^+ \\
+\varepsilon_i^+(u_i - \Delta_i^+) & \text{if } \Delta_i^+ \leq u_i \leq \Delta_i^+ + \Gamma_i^+ \\
+(\varepsilon_i^+ + \zeta_i^+)(u_i - \Gamma_i^+) - \zeta_i^+ \Delta_i^+ & \text{if } \Delta_i^+ + \Gamma_i^+ \leq u_i \leq +\infty 
\end{cases} \]

\[ D^*(z) = \sum_{i=1}^{m} D^*_i(z_i) \]

where

\[ D^*_i(z_i) = \begin{cases} 
+\infty & \text{if } z_i < -(\zeta_i^- + \varepsilon_i^-) \\
-(\Gamma_i^- + \Delta_i^-)z_i - \Gamma_i^+ \varepsilon_i^- & \text{if } -\zeta_i^- - \varepsilon_i^- \leq z_i \leq -\varepsilon_i^- \\
-\Delta_i^- z_i & \text{if } -\varepsilon_i^- \leq z_i \leq 0 \\
+\Delta_i^+ z_i & \text{if } 0 \leq z_i \leq \varepsilon_i^- \\
+(\Gamma_i^+ + \Delta_i^+)z_i - \Gamma_i^+ \varepsilon_i^+ & \text{if } \varepsilon_i^+ \leq z_i \leq (\zeta_i^+ + \varepsilon_i^+) \\
+\infty & \text{if } z_i > (\zeta_i^+ + \varepsilon_i^+) 
\end{cases} \]

Interval widths \iff penalties
A 5-piecewise-linear master problem

\[
\begin{align*}
\text{max} & \quad c \left( \sum_{\bar{x} \in B} \bar{x} \theta_{\bar{x}} \right) - \bar{y} (s^- + w^- - w^+ - s^+) \\
& \quad + \gamma^- s^- + \delta^- w^- + \delta^+ w^+ + \gamma^+ s^+ \\
& \quad A \left( \sum_{\bar{x} \in B} \bar{x} \theta_{\bar{x}} \right) + s^- + w^- - w^+ - s^+ = b \\
& \quad \sum_{\bar{x} \in B} \theta_{\bar{x}} = 1, \quad \theta_{\bar{x}} \geq 0 \quad \bar{x} \in B \\
& \quad 0 \leq s^- \leq \zeta^-, \quad 0 \leq s^+ \leq \zeta^+ \\
& \quad 0 \leq w^- \leq \varepsilon^-, \quad 0 \leq w^+ \leq \varepsilon^+
\end{align*}
\]
A 5-piecewise-linear master problem

\begin{align*}
\text{max} & \quad c ( \sum_{\bar{x} \in B} \bar{x} \theta_{\bar{x}} ) - \bar{y} (s^- + w^- - w^+ - s^+) \\
& + \gamma^- s^- + \delta^- w^- + \delta^+ w^+ + \gamma^+ s^+ \\
A ( \sum_{\bar{x} \in B} \bar{x} \theta_{\bar{x}} ) + s^- + w^- - w^+ - s^+ &= b \\
\sum_{\bar{x} \in B} \theta_{\bar{x}} &= 1, \quad \theta_{\bar{x}} \geq 0 \quad \bar{x} \in B \\
0 &\leq s^- \leq \zeta^- , \quad 0 \leq s^+ \leq \zeta^+ \\
0 &\leq w^- \leq \epsilon^- , \quad 0 \leq w^+ \leq \epsilon^+ \\
\end{align*}

- Same constraints as (\(\Pi_B\)), 4 slack variables for each constraint
- Many parameters: widths \(\Gamma^\pm\) and \(\Delta^\pm\), penalties \(\zeta^\pm\) and \(\epsilon^\pm\), different roles for small and large penalties
- Large penalties \(\zeta^\pm\) easily make \((\Delta_B,\bar{y},D)\) bounded \(\implies\) no Phase 0
- 3-pieces: either large penalty \(\implies\) small moves, or small penalty \(\implies\) 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)

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

- 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 $X$: $x \in X \implies x + \lambda r \in X$ for infinitely large $\lambda$

- $(c - yA)r > 0 \Rightarrow f(y) = +\infty \Rightarrow \text{constraint } cr \leq y(Ar)$ in dual space

\[(\Delta) \min \{ f(y) : y \in Y \} \]

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**:
  - $A\tilde{x} \rightarrow [ A\tilde{x}, 1 ]$, $b \rightarrow [ b, 1 ]$
  - Ignoring the special role of $v$ (just another $y$)
  - Advantage: everything is a constraint
On unboundedness and early termination

- A ray $r$ of $X$: $x \in X \implies x + \lambda r \in X$ for infinitely large $\lambda$

- $(c - yA)r > 0 \implies f(y) = +\infty \implies$ constraint $cr \leq y(Ar)$ in dual space

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

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:
  - $A\tilde{x} \rightarrow [A\tilde{x}, 1], \quad b \rightarrow [b, 1]$;
  - Ignoring the special role of $\nu$ (just another $y$)
  - Advantage: everything is a constraint

This is a bad idea!
On unboundedness and early termination

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

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

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:
  - $A\bar{x} \rightarrow [A\bar{x}, 1]$, $b \rightarrow [b, 1]$;
  - Ignoring the special role of $\nu$ (just another $y$)
  - Advantage: everything is a constraint

  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

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

- 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
  $\implies$ 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 $\tilde{y}$, to be updated: it’d be nice if we could do without

- Simple observation: dual constraints = primal variables $\implies$ 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]

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
  - Large master problem time
  - “Unstructured” master problems ⇒ general-purpose solvers
  - “Complicated” ⇒ 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]

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