Decomposition Approaches: The Role of the Master Problem Formulation

Alberto Caprara[1] *Antonio Frangioni* [2] Tiziano Parriani[3]

1. DEIS, Università di Bologna
2. Dipartimento di Informatica, Università di Pisa
3. OptIt

19th Aussois Combinatorial Optimization Workshop

January 5, 2015

Outline

The Multicommodity flow model

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

$$
\begin{aligned}
&\min \textstyle\sum_{k \in K} \sum_{(i,j) \in A} c^k_{ij} x^k_{ij} && && (1)\\
&\textstyle\sum_{(i,j) \in A} x^k_{ij} - \sum_{(j,i) \in A} x^k_{ji} = b^k && i \in N \;,\; k \in K && (2)\\
&\textstyle\sum_{k \in K} x^k_{ij} \leq u_{ij} && (i,j) \in A && (3)\\
&0 \leq x^k_{ij} \leq u^k_{ij} && (i,j) \in A \;,\; k \in K && (4)
\end{aligned}
$$

- Multiple source/sink commodities with individual capacities
- Can assume w.l.o.g. only one source, but in principle need (4) then
- In many cases, $K = \{ (s^k , t^k , d^k) \}$, $u^k_{ij} = +\infty$ "naturally"
- Many generalizations (extra constraints, nonlinearities [1], ...)

[1] F., Galli, Scutellà "Delay-Constrained Shortest Paths: Approx. Algorithms and Second-Order Cone Models" *JOTA*, to appear

[2] F., Galli. Stea "Optimal Joint Path Computation and Rate Allocation Real-time Traffic" *The Computer Journal*, to appear

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, ...
- Very different cases:
  - transportation: very large (often time-space $\Longrightarrow$ acyclic) networks, "few" commodities
  - telecommunications: "small" (undirected) networks, very many ($O(|N|^2)$) commodities
  - ...
- "Easy" in theory but "hard" in practice: very-large-scale LPs
- The archetype of block-structured problems [3,4]

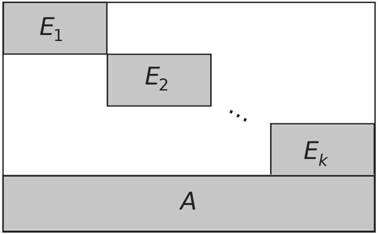
[3] Ford, Fulkerson "A Suggested Computation for Maximal Multicommodity Network Flows" *Management Science* 1958
[4] Dantzig, Wolfe "The Decomposition Principle for Linear Programs" *Operations Research* 1960

Block-Structured Linear Programs

- Block-structured LP:

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

$$X = \bigotimes_{k \in K} X^k = \{ \, x^k \; : \; E^k x^k \leq d^k \, \} \equiv Ax = b \text{ linking constraints}$$



- We know how to efficiently optimize upon X, for two reasons:
  - a bunch of (many, much) smaller problems instead of a large one
  - The X^k have structure: Min-Cost Flow (MCF) or shortest path (SPT)

- Many other applications (stochastic programs, ...)

Dantzig-Wolfe decomposition

- Dantzig-Wolfe reformulation (temporarily assume X compact): represent X by points instead

$$X = \{\ x = \textstyle\sum_{\bar{x} \in X}\ \bar{x}\theta_{\bar{x}}\ :\ \sum_{\bar{x} \in X}\ \theta_{\bar{x}} = 1\ ,\ \theta_{\bar{x}} \geq 0 \quad \bar{x} \in X\ \}$$

then reformulate (Π) in terms of the convex multipliers θ

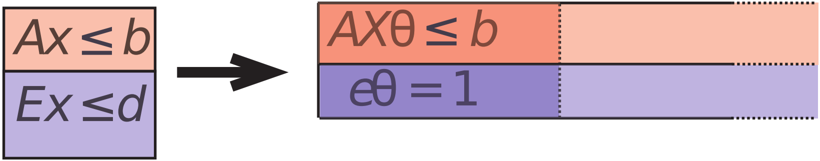
$$(\Pi) \qquad \begin{cases} \max \quad c\left(\sum_{\bar{x} \in X}\ \bar{x}\theta_{\bar{x}}\right) \\ \qquad A\left(\sum_{\bar{x} \in X}\ \bar{x}\theta_{\bar{x}}\right) \leq b \\ \qquad\qquad \sum_{\bar{x} \in X}\ \theta_{\bar{x}} = 1 \quad , \quad \theta_{\bar{x}} \geq 0 \qquad \bar{x} \in X \end{cases}$$

- 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 $\equiv$ Lagrangian relaxation

- Actually, only the vertices $V \subset X$ of 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 restriction of (Π) with $X \to \mathcal{B}$, i.e.,

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

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

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

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

The NDO Perspective: the Lagrangian dual

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

$$(\Delta_{\mathcal{B}}) \quad \begin{array}{rl} & \min \; \{ \; yb + v \;\; : \;\; v \geq (c - yA)x \quad x \in \mathcal{B} \; \} \\ = & \min \{ \; f_{\mathcal{B}}(y) = \max \{ \; cx + y(b - Ax) \;\; : \;\; x \in \mathcal{B} \; \} \; , \;\; y \geq 0 \; \} \end{array}$$

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

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

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

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

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

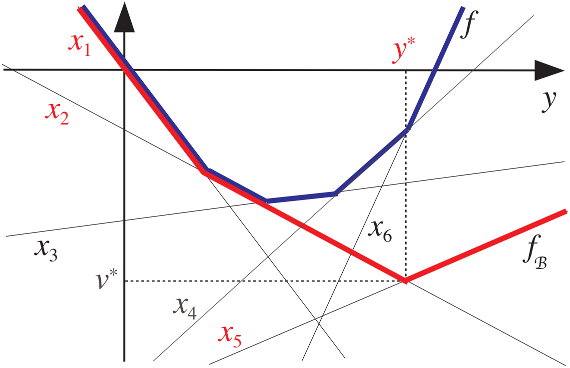
$$(\Delta) \quad \min \{ \; f(y) = \max \{ \; cx + y(b - Ax) \;\; : \;\; x \in X \; \} \; , \;\; y \geq 0 \; \}$$

  that is equivalent to (Π)

- Dantzig-Wolfe decomposition $\equiv$ Cutting Plane approach to (Δ) [5]

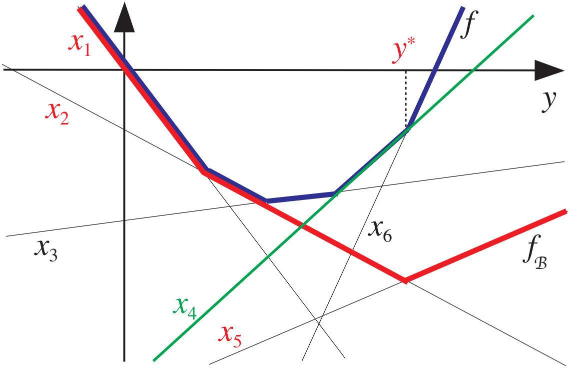
[5] Kelley "The Cutting-Plane Method for Solving Convex Programs" *Journal of the SIAM* 1960

Geometry of the Lagrangian dual



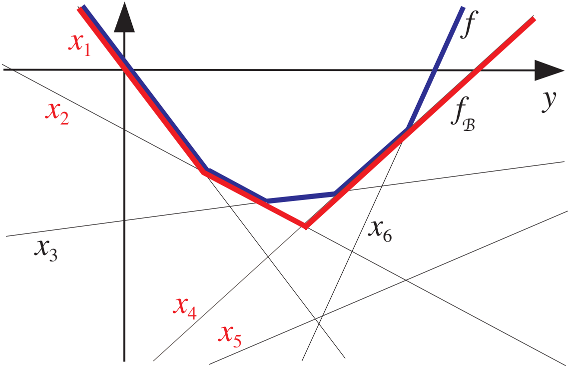
- $v^* = f_{\mathcal{B}}(y^*)$ lower bound on $v(\Pi_{\mathcal{B}})$

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

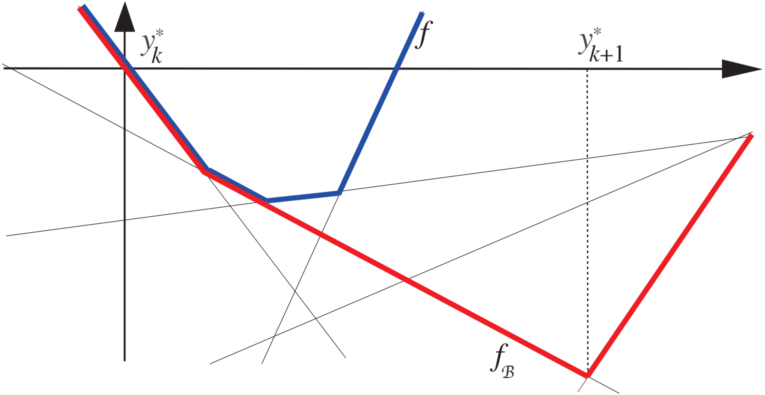


- $v^* = f_{\mathcal{B}}(y^*)$ lower bound on $v(\Pi_{\mathcal{B}})$
- Optimal solution $\bar{x}$ gives separator between (v^*, y^*) and *epi f*
- $(c\bar{x}, A\bar{x})$ = new row in $(\Delta_{\mathcal{B}})$ (subgradient of f at y^*)

Outline

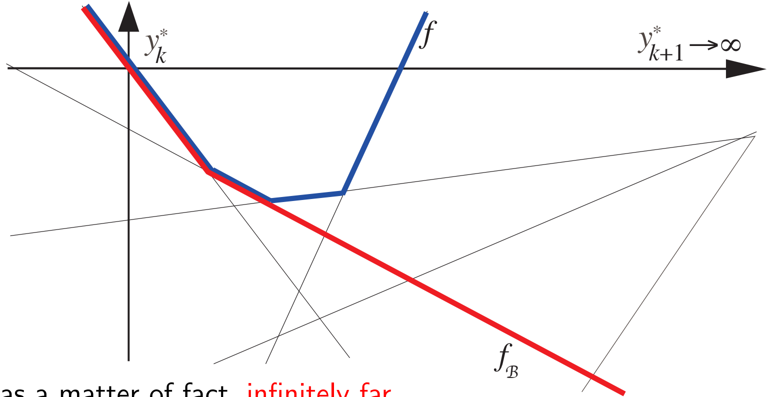
Issue with the approach: instability

- 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



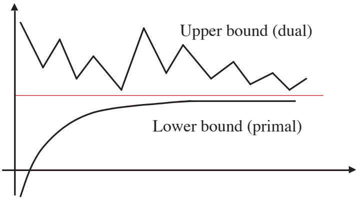
. . . as a matter of fact, infinitely far

- $(\Pi_{\mathcal{B}})$ 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

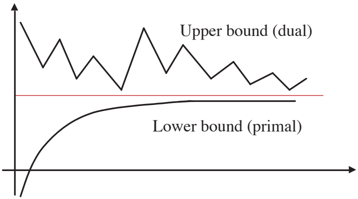
$\Longrightarrow$ tailing off, slow convergence



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! ...

Stabilized Dantzig-Wolfe

…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 [6]

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

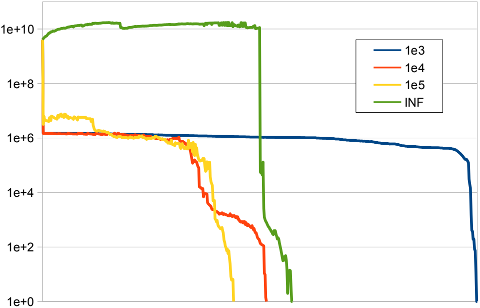
- Corresponding stabilized primal master problem

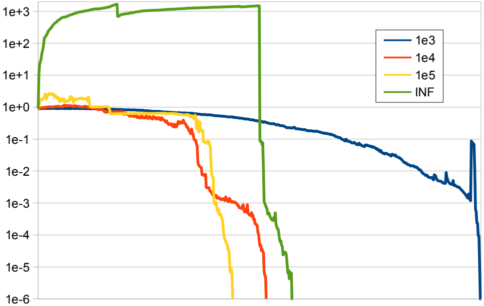
$$(\Pi_{\mathcal{B},\bar{y},t}) \quad \max \left\{ \, cx + \bar{y}z - t\|\, z \,\|_1 \;:\; z \geq Ax - b \,,\; z \geq 0 \,,\; x \in conv(\mathcal{B}) \,\right\}$$

  i.e., just Dantzig-Wolfe with slacks
- When stuck and $z^* = [Ax^* - b]_+ \neq 0$, either move $\bar{y}$ or enlarge t
- Minor modifications to the master problem
- How should one choose t?
- Does this really work?

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

Computational results of the boxstep method (pds18)

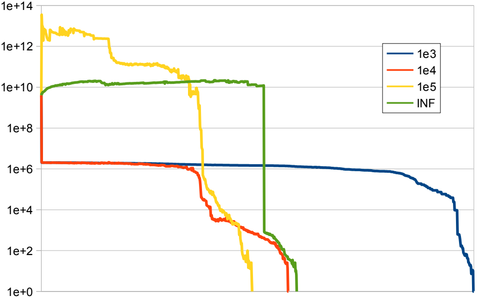




- Left = distance from final dual optimum

  right = relative gap with optimal value
- All cases show a "combinatorial tail" where convergence is very quick
- $t = 1e3$: "smooth but slow" until the combinatorial tail kicks in
- $t = \infty$: apparently trashing along until some magic threshold is hit
- "intermediate" t work best

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
- A different stabilizing term would help? Already

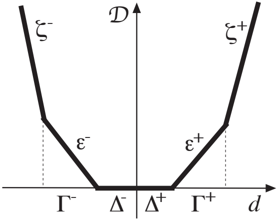
$$(\Delta_{\mathcal{B},\bar{y},t}) \qquad \min \left\{ f_{\mathcal{B}}(\,\bar{y} + d\,) + \tfrac{1}{2t} \|\, d \,\|_2^2 \right\}$$

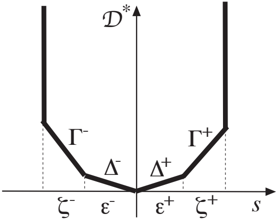
does [7,8], or even a more generic $\mathcal{D}(\,d\,) \Longrightarrow \mathcal{D}^*(\,d\,)$ in the primal [9]

[7] Lemaréchal "Bundle Methods in Nonsmooth Optimization" in *Nonsmooth Optimization* vol. 3, Pergamon Press, 1978
[8] Briant, Lemaréchal, et. al. "Comparison of bundle and classical column generation" *Mathematical Programming* 2006
[9] F. "Generalized Bundle Methods" *SIAM Journal on Optimization* 2002

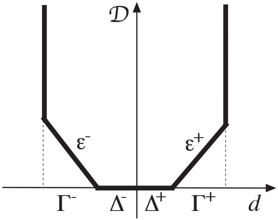
A 5-piecewise-linear function

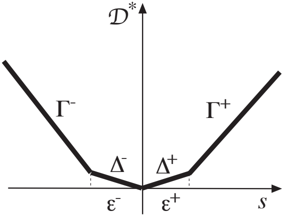
Trust region on $\bar{y}$ + small penalty close + much larger penalty farther [10]





Slightly simplified version: only 3 pieces.





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

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

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

- 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
- 5-pieces better than 3-pieces, 5-then-3 even better

Outline

The Arc-Path Formulation of Multicommodity Flows

- Assume each $k \in K$ is a O-D pair s^k–t^k with demand d^k (natural in many cases, can be forced somewhat in general)
- Arc-path formulation of Multicommodity Flows:
  $p \in \mathcal{P}^k = \{\ s^k$–$t^k$ paths $\}$, c_p cost, f_p flow, $\mathcal{P} = \cup_{k \in K} \mathcal{P}^k$

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

  Fewer constraints but exponentially many variables: oddly familiar?
- In fact, just a disaggregated version of the Dantzig-Wolfe formulation
- General principle: $X = X^1 \times X^2 \times \ldots \times X^{|K|} \Longrightarrow$ $conv(X) = conv(X^1) \times conv(X^2) \times \ldots \times conv(X^{|K|})$

Dantzig-Wolfe and Multicommodity flows

- Standard D-W: $\mathcal{S} = \{$ (extreme) flows $s = [\bar{x}^{1,s}, \ldots, \bar{x}^{k,s}]\}$

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

- Disaggregated D-W: a different multiplier θ^k_s for each $\bar{x}^{k,s}$, with

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

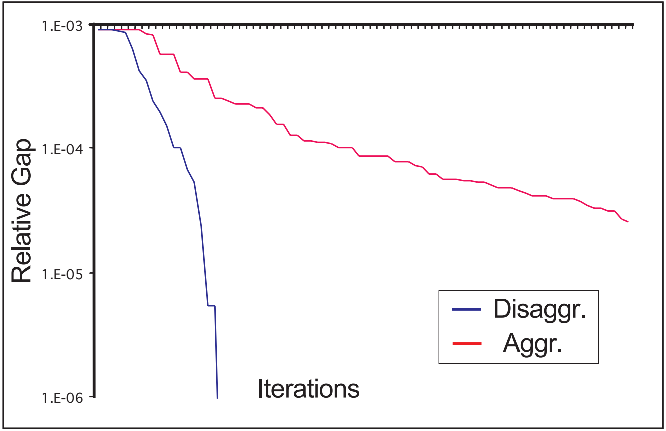
(clearly, previous case is $\theta^k_s = \theta^h_s, h \neq k$) $\Longrightarrow$ better value

- In NDO-speak: sum of models is better than model of the sum
- Simple scaling leads to arc-path formulation: $f_p = d^k \theta^k_s$
- Many more columns but sparser, (a few) more rows
- Master problem size ($\approx$ time, or not) increases, but convergence speed increases a lot $\equiv$ consistent improvement [11]

[11] Jones, Lustig, Farwolden, Powell "Multicommodity Network Flows: The Impact of Formulation on Decomposition" *Mathematical Programming* 1993

Disaggregated decomposition

- Easily extended to any decomposable $X \equiv$ sum-function [12]



- Stabilized versions immediate
- Is there anything more to say?

[12] Borghetti, F., Lacalandra, Nucci "Lagrangian Heuristics Based on Disaggregated Bundle Methods for Hydrothermal Unit Commitment" *IEEE Transactions on Power Systems* 2003

More Disaggregated Versions

- Aggregation is arbitrary, then why "all or nothing"?
- Partition $C = (C_1, C_2, \dots, C_h)$ of K
- Partially aggregated model $f_{\mathcal{B}}^{C}$ with h $(+1)$ components $f_{\mathcal{B}}^{i}$, each the sum over one C_i
- Basically, $\theta_s^k = \theta_s^h$ for each $(h, k) \in C_i \times C_i$
- Exploring the trade-off between master problem size $\Longrightarrow$ time and iterations, subproblem time remains the same
- Aggregation index $\eta \in [0, 1]$:

  $$h = |C| = \max\left\{ \lceil\, (1 - \eta)|K| \,\rceil , 1 \right\}$$

  $0 =$ fully disaggregated, $1 =$ fully aggregated
- How to choose the commodities in each C_i? In general open problem, here just group commodities with "close original names"

Even More Disaggregated Versions

- But what is a commodity, anyway?
  - Modeler's view: a product, origin-destination, stream of packets, ...
  - Algorithm's view: all that can be bunched together
- Commodity-independent data $\equiv$ bunch commodities with same origin
- Why is that? Because you can solve a unique SPT for all of them (which is because SPT has a funny auto-separability property)
- From a modeling viewpoint, there is no reason to (can always recover the original solution, less variables)
- This impact how the master problem is formulated [11] ...

Even More Disaggregated Versions

- But what is a commodity, anyway?
  - Modeler's view: a product, origin-destination, stream of packets, ...
  - Algorithm's view: all that can be bunched together
- Commodity-independent data $\equiv$ bunch commodities with same origin
- Why is that? Because you can solve a unique SPT for all of them (which is because SPT has a funny auto-separability property)
- From a modeling viewpoint, there is no reason to (can always recover the original solution, less variables)
- This impact how the master problem is formulated [11] ... or not: the Master Problem can be freely reformulated
- Aggregation index $\eta \in [-1, 0]$: $\overline{K}$ the number of OD pairs,

$$h = |C| = \max \left\{ \left\lceil \, -\eta|\overline{K}| \, \right\rceil \, , \, |K| \, \right\}$$

  $-1 =$ ODP formulation, $0 =$ DSP formulation [11]
- Again, commodities in a C_i just have "close destination node names"

Dealing With Multiple-origin Commodities

- What about commodities that have many origins (PSP in [11])?
- Can always assume one origin (add a super-origin) ...

Dealing With Multiple-origin Commodities

- What about commodities that have many origins (PSP in [11])?
- Can always assume one origin (add a super-origin) ...
  but must add commodity-specific capacities on super-origin arcs
- Data no longer commodity-independent, subproblems no longer SPTs
  $\Longrightarrow$ cannot disaggregate by origin ...

Dealing With Multiple-origin Commodities

- What about commodities that have many origins (PSP in [11])?
- Can always assume one origin (add a super-origin) . . .
  but must add commodity-specific capacities on super-origin arcs
- Data no longer commodity-independent, subproblems no longer SPTs
  $\Longrightarrow$ cannot disaggregate by origin . . . or you can:
  "just" consider individual capacities as complicating constraints
- This is still a reformulation of the master problem,
  has (almost) nothing to do with the original problem formulation
- Obvious trade-off: simpler subproblems, harder master problem
  (possibly many more rows, more columns but sparser ones)
- TTBoOK haven't been computationally explored to far

(Preliminary, $\eta \geq 0$) Computational Results

- Generalized Bundle code using $D_t^* = \|\cdot\|_1$ (boxstep)
- Latest Cplex as Master Problem Solver
- Efficient implementation: overhead due to subgradient handling significant
- Limited effect of stabilization (not much need)
- (Reasonably) efficient subproblem solution with `MCFClass`

  `http://www.di.unipi.it/optimize/Software/MCF.html`

- Many instances, some old, some new, from

  `http://www.di.unipi.it/optimize/Data/MMCF.html`

- Results for $\eta < 0$ still brewing, but these significant enough already

Computational Results: Planar & Grid Instances

	0		0.2		0.4		0.6		0.8		1	
	time	it.	time	it.	time	it.	time	it.	time	it.	time	it.
grid7	2.5	12	2.91	13	2.39	15	2.12	14	2.62	18	285.76	1169
grid8	18.52	18	18.33	19	21.05	20	25.61	23	42.36	33	***	3848
grid9	36.04	15	45.94	16	60.54	18	85.99	20	189.92	32	***	2862
grid10	54.51	15	61.40	16	77.96	17	104.18	18	233.07	24	***	3848
grid12	61.64	11	61.24	10	65.44	11	71.81	11	148.89	13	***	2862
grid14	433.64	11	388.76	11	289.13	12	230.66	11	259.22	12	***	2862
planar100	2.16	14	1.96	13	1.42	13	2.36	13	2.74	15	25.49	1400
planar150	25.75	17	29.11	17	28.77	17	30.44	19	35.49	23	***	68896
planar300	21.34	13	22.86	14	23.54	14	24.12	15	24.71	14	1292.09	2967
planar500	15.27	11	14.75	11	13.91	11	12.71	12	10.84	11	197.62	317

- Large, nasty instances (you'll see later)
- *** = out of time limit (6400 seconds): all for $\eta = 1$, clearly worst
- Results somewhat erratic, but clearly $\eta = 0$ not always best

Computational Results: Goto & Mnetgen Instances

	0		0.2		0.4		0.6		0.8		1	
	time	it.	time	it.	time	it.	time	it.	time	it.	time	it.
Goto6-100	1.05	25	1.33	30	1.39	35	1.67	44	1.40	69	16.09	926
Goto6-400	1.45	15	1.59	17	1.76	19	2.40	22	5.79	32	60.53	1272
Goto6-800	2.41	12	2.54	14	2.85	15	3.62	17	9.24	25	134.42	1709
Goto8-10	2.96	75	4.57	104	6.14	137	7.68	164	18.12	301	45.29	722
Goto8-100	3.43	21	4.86	27	4.98	31	5.58	45	13.73	79	388.32	2114
Goto8-400	5.88	16	8.13	18	11.03	20	14.68	23	24.86	36	582.66	2690
Goto8-800	3.12	11	3.30	12	4.53	13	6.34	15	10.32	20	82.93	729
128-32	17.66	57	27.64	76	23.54	91	31.09	128	32.92	222	294.03	2753
128-32	57.23	46	66.04	59	63.66	70	79.97	92	108.53	169	1337.79	5296
128-64	95.45	34	125.27	43	126.71	50	147.25	65	174.81	108	1750.57	3741
128-128	5.68	109	5.73	109	8.08	158	12.34	209	24.09	437	25.22	449
256-8	31.65	140	45.55	183	77.50	252	94.51	276	289.69	635	1020.79	1826
256-16	146.37	148	181.38	219	244.79	271	404.15	381	885.73	704	1856.84	2175
256-32	400.59	117	510.74	163	640.14	200	1081.34	299	1666.35	480	2740.50	2615
256-64	563.66	86	744.93	113	1108.17	143	1624.06	196	1834.86	293	2670.98	1821

- . . . although in some cases $\eta = 0$ can be (almost) uniformly best

Computational Results: Waxman & Rmnet Instances

	0		0.2		0.4		0.6		0.8		1	
	time	it.	time	it.	time	it.	time	it.	time	it.	time	it.
W-50	1.43	3	0.17	3	0.11	3	0.07	3	0.03	3	0.04	9
W-100-6	1.53	2	0.20	2	0.13	2	0.09	2	0.04	2	0.06	10
W-100-10	1.34	3	0.38	3	0.32	3	0.27	3	0.22	3	0.70	15
W-100	1.50	2	2.10	2	1.37	2	0.98	2	0.72	2	1.06	7
W-150-6	2.44	2	2.30	2	1.81	2	1.20	2	0.63	2	4.54	44
W-150-10	1.23	3	0.83	3	0.66	3	0.60	3	0.14	2	0.48	4
W-150	3.23	3	4.74	3	3.17	3	2.70	3	0.67	3	4.49	9
4-8-11-100	0.56	5	1.31	5	0.83	5	0.58	5	0.40	5	0.31	8
4-8-12-200	1.31	5	2.07	5	1.64	5	1.18	5	1.06	5	0.45	6
4-8-13-200	5.88	7	11.11	7	9.31	8	6.54	8	6.00	9	9.70	62
4-8-14-400	55.62	7	75.70	7	39.81	8	27.77	8	15.59	9	19.89	62
7-6-11-100	1.00	6	2.27	6	2.42	6	2.29	6	1.22	7	5.38	54
7-6-12-500	1.80	5	3.08	5	3.62	5	3.23	5	1.80	5	1.64	8
7-6-13-500	4.56	5	8.85	5	7.34	5	5.86	5	4.48	6	11.96	30
7-6-14-1000	30.29	5	35.54	5	27.04	5	24.58	5	12.57	6	30.26	38

- . . . or (almost) uniformly worst (save for $\eta = 1$)
- but often strange things happen ($\eta = 1$ can even be best)

Outline

Structured Decomposition

- Came out for a different (still multicommodity) problem [13]
- D-W $\equiv$ reformulation of X always in the same form ...

Structured Decomposition

- Came out for a different (still multicommodity) problem [13]
- D-W $\equiv$ reformulation of X always in the same form ...
  or not, as we have already seen. But we can do better:
  - Assumption 1: alternative Formulation of "easy" set
  $$X = \{\ x = C\theta \ :\ \Gamma\theta \leq \gamma\ \}$$
  - Assumption 2: $\mathcal{B}$ subset of rows and columns, padding with zeroes
  $$\Gamma_{\mathcal{B}}\bar{\theta}_{\mathcal{B}} \leq \gamma_{\mathcal{B}} \implies \Gamma\left[\ \bar{\theta}_{\mathcal{B}}\ ,\ 0\ \right] \leq \gamma$$
  $$\implies X_{\mathcal{B}} = \left\{\ x = C_{\mathcal{B}}\theta_{\mathcal{B}} \ :\ \Gamma_{\mathcal{B}}\theta_{\mathcal{B}} \leq \gamma_{\mathcal{B}}\ \right\} \subseteq X$$
  - Assumption 3: easy update of rows and columns
  Given $\mathcal{B}$, $\bar{x} \in X$, $\bar{x} \notin X_{\mathcal{B}}$, it is "easy" to find $\mathcal{B}' \supset \mathcal{B}$
  ($\implies \Gamma_{\mathcal{B}'}$, $\gamma_{\mathcal{B}'}$) such that $\exists\ \mathcal{B}'' \supseteq \mathcal{B}'$ such that $\bar{x} \in X_{\mathcal{B}''}$.

[13] F., Gendron "0-1 reformulations of the multicommodity capacitated network design problem" *Discrete Applied Math.* 2009

The Structured Dantzig-Wolfe Algorithm

- Structured master problem $\equiv$ structured model

$$(\Pi_{\mathcal{B}}) \qquad \max\left\{\, cx \ : \ Ax \leq b \,,\ x = C_{\mathcal{B}}\theta_{\mathcal{B}} \,,\ \ \Gamma_{\mathcal{B}}\theta_{\mathcal{B}} \leq \gamma_{\mathcal{B}} \,\right\}$$

$$f_{\mathcal{B}}(y) \ = \ \max\{\, (c - yA)x + xb \ : \ x = C_{\mathcal{B}}\theta_{\mathcal{B}} \,,\ \Gamma_{\mathcal{B}}\theta_{\mathcal{B}} \leq \gamma_{\mathcal{B}} \,\}$$

$\langle$ initialize $\mathcal{B}$ $\rangle$;
`repeat`
  $\langle$ solve $(\Pi_{\mathcal{B}})$ for x^*, y^* (duals of $Ax \leq b$); $v^* = cx^*$ $\rangle$;
  $\bar{x} = \text{argmin}\ \{\, (c - y^*A)x \ : \ x \in X \,\}$;
  $\langle$ update $\mathcal{B}$ as in Assumption 3 $\rangle$;
`until` $v^* < c\bar{x} + y^*(b - A\bar{x})$

- Finitely terminates with an optimal solution, even if (proper) removal from $\mathcal{B}$ is allowed, X is non compact and $\mathcal{B} = \emptyset$ at start (Phase 0)
- The subproblem to be solved is identical to that of DW
- Requires ($\Longrightarrow$ exploits) extra information on the structure
- Master problem with any structure, possibly much larger

Stabilizing the Structured Dantzig-Wolfe Algorithm

- Exactly the same as stabilizing DW: stabilized master problem

$$(\Delta_{\mathcal{B},\bar{y},\mathcal{D}}) \qquad \min \{ f_{\mathcal{B}}(\bar{y} + d) + \mathcal{D}(d) \}$$

except $f_{\mathcal{B}}$ is a different model of f (not the cutting plane one)

- Even simpler from the primal viewpoint [14]:

$$\max \{ cx + \bar{y}z - \mathcal{D}^*(z) \,:\, z \geq Ax - b \,,\, z \geq 0 \,,\, x = C_{\mathcal{B}}\theta_{\mathcal{B}} \,,\, \Gamma_{\mathcal{B}}\theta_{\mathcal{B}} \leq \gamma_{\mathcal{B}} \}$$

- With proper choice of $\mathcal{D}$, still a(sparsely structured) Linear Program
- Dual optimal variables of "$z \geq Ax - b$" still give d^*, ...
- How to move $\bar{y}$, handle t, handle $\mathcal{B}$: basically as in [9], actually even somewhat simpler because $\mathcal{B}$ is inherently finite
- Funnily, aggregation $\mathcal{B} = \mathcal{B} \cup \{ x^* \}$ is also possible, up to

$$\mathcal{B} = \{ x^* \} \equiv \text{"poorman" method}$$

although clearly contrary to the spirit of S^{2}DW

[14] F., Gendron "A Stabilized Structured Dantzig-Wolfe Decomposition Method" *Mathematical Programming* 2013

Structured Decomposition for Multicommodity Flows

- All nice and well, but how can we come up with a $x = C\theta$?
- Surprisingly simple: use the node-arc formulation
- Start with "empty graph", find paths: if a node/arc is missing, add it
- Intermediate formulation between node-arc and arc-path
- Would seem to generalize to many other network-structured problems
- Current implementation heavily relies on Cplex preprocessor
  it may be preferable to do the path splitting by hand
- Current implementation is not stabilized at all

(Preliminary) Computational results

- Ad-hoc code (including in general Bundle non trivial, but possible) [15]
- No stabilization (but probably none needed)
- Still using Cplex as main driving force
- Comparing also against direct use of Cplex (tuned)
- Exactly the same subproblem solver (`FiOracle`)
- Surely can be improved a lot (e.g. explicit graph operations)
- Same instances, same machine

[15] F., Gorgone "Bundle methods for sum-functions with "easy" components: applications to multicommodity network design" *Mathematical Programming* 2014

Computational Results: Planar & Grid Instances

	0		*		SDW		Cplex
	time	it.	time	it.	time	it.	time
grid7	2.5	12	2.12	14	1.29	9	54.73
grid8	18.52	18	18.33	19	23.81	12	1745.65
grid9	36.04	15	36.04	15	193.53	12	***
grid10	54.51	15	54.51	15	596.83	13	***
grid12	61.64	11	61.24	10	881.37	11	***
grid14	433.64	11	230.66	11	6086.84	11	***
planar100	2.16	14	1.42	13	2.66	8	43.90
planar150	25.75	17	25.75	17	183.94	11	4239.98
planar300	21.34	13	21.34	13	112.87	9	5127.74
planar500	15.27	11	10.84	11	25.16	7	***

- *** = out of time limit (6400 seconds): Cplex clearly worst
- SDW seldom competitive here, although much better than Cplex
- $\eta = 0$ not a bad choice overall, but not necessarily best

Computational Results: Goto & Mnetgen Instances

	0 = *		SDW		Cplex
	time	it.	time	it.	time
Goto6-100	1.05	25	0.60	11	0.67
Goto6-400	1.45	15	2.42	14	14.22
Goto6-800	2.41	12	5.54	15	64.09
Goto8-10	2.96	75	0.11	8	0.11
Goto8-100	3.43	21	1.45	14	5.63
Goto8-400	5.88	16	11.12	17	105.13
Goto8-800	3.12	11	17.23	18	326.01
128-32	17.66	57	3.90	6	0.32
128-32	57.23	46	15.08	6	0.87
128-64	95.45	34	32.66	7	1.61
128-128	5.68	109	0.25	5	0.05
256-8	31.65	140	0.80	6	0.07
256-16	146.37	148	4.97	6	0.28
256-32	400.59	117	23.95	6	1.07
256-64	563.66	86	61.45	7	1.69

- SDW is not often the best, but it is never the worst

Computational Results: Waxman [& Rmnet] Instances

	0		$0.8 = *$		SDW		Cplex
	time	it.	time	it.	time	it.	time
W-50	1.43	3	0.03	3	0.32	7	1.12
W-100-6	1.53	2	0.04	2	0.39	7	1.20
W-100-10	1.34	3	0.22	3	1.11	6	3.14
W-100	1.50	2	0.72	2	0.86	2	22.49
W-150-6	2.44	2	0.63	2	2.93	6	33.82
W-150-10	1.23	3	0.14	2	3.54	4	10.38
W-150	3.23	3	0.67	3	2.14	3	52.21

- Er . . . Rmnet not ready yet, sorry (preliminary I said)
- When few paths (= iterations) are required, SDW can't help much
- Still better than using Cplex directly, though
- Often better than standard decomposition with non-optimal η

Conclusions and (a lot of) future work

- After 50+ years, Multicommodity flows still inspirational to NDO

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

Conclusions and (a lot of) future work

- After 50+ years, Multicommodity flows still inspirational to NDO
- DW decomposition $\equiv$ CP is a very old idea, very well-understood; yet, by-the-book decomposition is not effective enough

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

Conclusions and (a lot of) future work

- After 50+ years, Multicommodity flows still inspirational to NDO
- DW decomposition $\equiv$ CP is a very old idea, very well-understood; yet, by-the-book decomposition is not effective enough
- Many possible ideas to improve on the standard approach, almost all of them based on reformulating the MP one way or other

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

Conclusions and (a lot of) future work

- After 50+ years, Multicommodity flows still inspirational to NDO
- DW decomposition $\equiv$ CP is a very old idea, very well-understood; yet, by-the-book decomposition is not effective enough
- Many possible ideas to improve on the standard approach, almost all of them based on reformulating the MP one way or other
- Substantial issue: what works best is "large" MPs
  so that "combinatorial tail" kicks in very quickly $\Longrightarrow$

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

Conclusions and (a lot of) future work

- After 50+ years, Multicommodity flows still inspirational to NDO
- DW decomposition $\equiv$ CP is a very old idea, very well-understood; yet, by-the-book decomposition is not effective enough
- Many possible ideas to improve on the standard approach, almost all of them based on reformulating the MP one way or other
- Substantial issue: what works best is "large" MPs so that "combinatorial tail" kicks in very quickly $\Longrightarrow$
  - Large MP time

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

Conclusions and (a lot of) future work

- After 50+ years, Multicommodity flows still inspirational to NDO
- DW decomposition $\equiv$ CP is a very old idea, very well-understood; yet, by-the-book decomposition is not effective enough
- Many possible ideas to improve on the standard approach, almost all of them based on reformulating the MP one way or other
- Substantial issue: what works best is "large" MPs
  so that "combinatorial tail" kicks in very quickly $\Longrightarrow$
  - Large MP time
  - "Unstructured" MPs $\Longrightarrow$ general-purpose solvers

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

Conclusions and (a lot of) future work

- After 50+ years, Multicommodity flows still inspirational to NDO
- DW decomposition $\equiv$ CP is a very old idea, very well-understood; yet, by-the-book decomposition is not effective enough
- Many possible ideas to improve on the standard approach, almost all of them based on reformulating the MP one way or other
- Substantial issue: what works best is "large" MPs so that "combinatorial tail" kicks in very quickly $\Longrightarrow$
  - Large MP time
  - "Unstructured" MPs $\Longrightarrow$ general-purpose solvers
  - Hard to find the right trade-off between iterations and MP time

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

Conclusions and (a lot of) future work

- After 50+ years, Multicommodity flows still inspirational to NDO
- DW decomposition $\equiv$ CP is a very old idea, very well-understood; yet, by-the-book decomposition is not effective enough
- Many possible ideas to improve on the standard approach, almost all of them based on reformulating the MP one way or other
- Substantial issue: what works best is "large" MPs so that "combinatorial tail" kicks in very quickly $\Longrightarrow$
  - Large MP time
  - "Unstructured" MPs $\Longrightarrow$ general-purpose solvers
  - Hard to find the right trade-off between iterations and MP time
  - Need to exploit the structure of an unstructured problem (perhaps less contradictory than is sounds [16])

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

Conclusions and (a lot of) future work

- After 50+ years, Multicommodity flows still inspirational to NDO
- DW decomposition $\equiv$ CP is a very old idea, very well-understood; yet, by-the-book decomposition is not effective enough
- Many possible ideas to improve on the standard approach, almost all of them based on reformulating the MP one way or other
- Substantial issue: what works best is "large" MPs so that "combinatorial tail" kicks in very quickly $\Longrightarrow$
  - Large MP time
  - "Unstructured" MPs $\Longrightarrow$ general-purpose solvers
  - Hard to find the right trade-off between iterations and MP time
  - Need to exploit the structure of an unstructured problem (perhaps less contradictory than is sounds [16])
- Lesson to NDO: think outside the (black) box, all structure that is there has to be exploited

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

Visit Pisa in September!

Come to AIRO 2015