Column Generation: Just Say "No"... for the Wrong Variables

A. Frangioni[♮] E. Gorgone[♭]

[♮]Dipartimento di Informatica, Polo Universitario della Spezia – Università di Pisa
[♭]Dipartimento di Elettronica Informatica e Sistemistica – Università della Calabria

International Workshop on Column Generation 2012
Bromont, Québec, Canada
June 10 to 13, 2012

Outline

The Problem, Formulations

Fixed-Charge Multicommodity Min–Cost Flow Problem

- Graph $G = (N, A)$, commodities $K = \{(s^k, t^k, d^k)\}$
- Weak Formulation (WF): routing costs c^k_{ij}, design costs f_{ij}

$$
(\Omega) \quad \left|\begin{array}{lll}
\min & \displaystyle\sum_{k \in K} \sum_{(i,j) \in A} c^k_{ij} x^k_{ij} + \sum_{(i,j) \in A} f_{ij} y_{ij} & \\
& \displaystyle\sum_{(i,j) \in A} x^k_{ij} - \sum_{(j,i) \in A} x^k_{ji} = b^k_i & i \in N, \; k \in K \\
& \displaystyle\sum_{k \in K} x^k_{ij} \leq u_{ij} y_{ij} & (i,j) \in A \\
& 0 \leq x^k_{ij} \leq u^k_{ij} & (i,j) \in A, \; k \in K \\
& y_{ij} \in \{0, 1\} & (i,j) \in A
\end{array}\right.
$$

$b^k_i = d^k$ if $i = s^k$, $b^k_i = -d^k$ if $i = t^k$, and 0 otherwise

FC-MMCF: Strong Formulation (SF)

- WF : $|K|(|A|+1)$ variables, $|K||N|+|A|$ constraints (+ bounds)
- $\mathrm{SF} = \mathrm{WF} + |K||A|$ (redundant) individual capacity cuts

$$x_{ij}^k \leq u_{ij}^k y_{ij} \qquad (i,j) \in A,\ k \in K$$

(effective if $u_{ij}^k \ll u_{ij}$)

- Continuous relaxations $\overline{\mathrm{WF}}$, $\overline{\mathrm{SF}}$
- $\overline{\mathrm{SF}}$: rather difficult to solve when $|K|$ large (telecommunications) + rather weak bound
- Decent bound (to start with ...), but extremely difficult to solve (efficiently) ... even with row generation

Column Generation

Weak Formulation

- $\mathcal{S}$ = set of all (extreme) solutions $s = [\bar{x}^{1,s}, \ldots, \bar{x}^{k,s}, \bar{y}^{s}]$ of the pricing problem
- Dantzig–Wolfe (DW) reformulation of $\overline{\mathrm{WF}}$

$$
(\Delta_{\mathcal{S}}) \left|
\begin{array}{ll}
\min & \sum_{s \in \mathcal{S}} \left(\sum_{k \in K} \sum_{(i,j) \in A} c^k_{ij} \bar{x}^{k,s}_{ij} + \sum_{(i,j) \in A} f_{ij} \bar{y}^s_{ij} \right) \theta_s \\
& \sum_{s \in \mathcal{S}} \left(\sum_{k \in K} \bar{x}^{k,s}_{ij} - u_{ij} \bar{y}^s_{ij} \right) \theta_s \leq 0 \qquad (i,j) \in A \\
& \sum_{s \in \mathcal{S}} \theta_s = 1 \, , \quad \theta_s \geq 0 \qquad s \in \mathcal{S}
\end{array}
\right.
$$

- Exponential number of variables, only $|A| + 1$ (non-bound) constraints $\Rightarrow$ *column generation*
- (Small) subset $\mathcal{B}$ of $\mathcal{S}$, master problem with $\theta_s = 0$ for $s \in \mathcal{S} \setminus \mathcal{B}$
- In theory, $|\mathcal{B}| \leq |A| + 1$ could suffice

Weak Formulation – Pricing Problem

- Pricing problem $P(\alpha)$ of (WF)

$$
P(\alpha) \left|
\begin{array}{ll}
\min & \sum_{k \in K} \sum_{(i,j) \in A} (c_{ij}^k + \alpha_{ij}) x_{ij}^k + \sum_{(i,j) \in A} (f_{ij} - \alpha_{ij} u_{ij}) y_{ij} \\
& \sum_{(i,j) \in A} x_{ij}^k - \sum_{(j,i) \in A} x_{ji}^k = b_i^k \qquad i \in N \, , \; k \in K \\
& 0 \leq x_{ij}^k \leq u_{ij}^k \qquad (i,j) \in A \, , \; k \in K \\
& y_{ij} \in \{0,1\} \qquad (i,j) \in A
\end{array}
\right.
$$

- Decomposed into $|K|$ (SPT)s/(MCF)s + one trivial problem
- DW is equivalent to $\overline{\mathrm{WF}}$ (fully)

Strong Formulation

- Same thing for $\overline{\mathrm{SF}}$ ($\mathcal{S}$ exactly the same)

$$
(\Delta_{\mathcal{S}}) \left|
\begin{array}{lll}
\min & \sum_{s \in \mathcal{S}} \left(\sum_{k \in K} \sum_{(i,j) \in A} c^k_{ij} \bar{x}^{k,s}_{ij} + \sum_{(i,j) \in A} f_{ij} \bar{y}^s_{ij} \right) \theta_s & \\
& \sum_{s \in \mathcal{S}} \left(\sum_{k \in K} \bar{x}^{k,s}_{ij} - u_{ij} \bar{y}^s_{ij} \right) \theta_s \leq 0 & (i,j) \in A \\
& \sum_{s \in \mathcal{S}} \left(\bar{x}^{k,s}_{ij} - u^k_{ij} \bar{y}^s_{ij} \right) \theta_s \leq 0 & (i,j) \in A \, , \; k \in K \\
& \sum_{s \in \mathcal{S}} \theta_s = 1 \, , \quad \theta_s \geq 0 & s \in \mathcal{S}
\end{array}
\right.
$$

- ... but $|A|(|K| + 1) + 1$ (non-bound) constraints
- Turn-based row and column generation
- Now one expects $\mathcal{B} \approx |K||A|$ at least (although it may be less)

Strong Formulation – Pricing Problem

- Pricing problem $P(\alpha, \beta)$ of (SF)

$$
P(\alpha,\beta) \left|
\begin{array}{lll}
\min & \sum_{k \in K} \sum_{(i,j) \in A} (c^k_{ij} + \alpha_{ij} + \beta^k_{ij}) x^k_{ij} + & \\
 & \qquad \sum_{(i,j) \in A} (f_{ij} - \alpha_{ij} u_{ij} - \beta^k_{ij} u^k_{ij}) y_{ij} & \\
 & \sum_{(i,j) \in A} x^k_{ij} - \sum_{(j,i) \in A} x^k_{ji} = b^k_i & i \in N \, , \; k \in K \\
 & 0 \leq x^k_{ij} \leq u^k_{ij} & (i,j) \in A \, , \; k \in K \\
 & y_{ij} \in \{0,1\} & (i,j) \in A
\end{array}
\right.
$$

- Solvable exactly as the previous one
- DW is equivalent to $\overline{\mathrm{SF}}$

Disaggregated Dantzig–Wolfe

- $\mathcal{S}^k$ = extreme points of commodity k (paths), $\mathcal{S}^y$ = extreme points of fixed charges ($2^{|A|}$ vertices of the unary hypercube)
- Disaggregate formulation (of WF to keep it simple)

$$
(\Delta_{\mathcal{S}}) \left|
\begin{array}{lll}
\min & \sum_{k \in K} \sum_{(i,j) \in A} c_{ij}^k \Big(\sum_{s \in \mathcal{S}^k} \bar{x}_{ij}^{k,s} \theta_s \Big) + \sum_{(i,j) \in A} f_{ij} \Big(\sum_{s \in \mathcal{S}} \bar{y}_{ij}^s \theta_s \Big) & \\
& \sum_{k \in K} \Big(\sum_{s \in \mathcal{S}^k} \bar{x}_{ij}^{k,s} \theta_s \Big) \leq u_{ij} \Big(\sum_{s \in \mathcal{S}} \bar{y}_{ij}^s \theta_s \Big) & (i,j) \in A \\
& \sum_{s \in \mathcal{S}^k} \theta_s = 1 \,, \quad \theta_s \geq 0 & s \in \mathcal{S}^k \,, \; k \in K \\
& \sum_{s \in \mathcal{S}^y} \theta_s = 1 \,, \quad \theta_s \geq 0 & s \in \mathcal{S}^y
\end{array}
\right.
$$

- Slightly more constraints, but sparser + much sparser columns
- $|K| + 1$ *times* more columns ... but $|\mathcal{B}| \approx 2|A|$ in the *best* case (SF)
- Usually (much) more efficient

Even more disaggregated Dantzig–Wolfe

- The unitary hypercube is a cartesian product: why not $\mathcal{Y} = \bigotimes_{(i,j)\in A} \left(\mathcal{Y}^{ij} = \{0,1\} \right)$?
- $y_{ij} \;\longrightarrow\; 0 \cdot \theta^{ij,0} + 1 \cdot \theta^{ij,1} \;,\; \theta^{ij,0} + \theta^{ij,1} = 1 \;,\; \theta^{ij,0} \geq 0 \;,\; \theta^{ij,1} \geq 0.$

  $\boxed{y_{ij} \in [\, 0 \;,\; 1 \,]}$ (no, ... really?!)
- Arc-path formulation with original arc design variables

$$
\begin{array}{lll}
\min & \displaystyle\sum_{p\in\mathcal{P}=\cup_{k\in K}\mathcal{P}^k} c_p f_p + \sum_{(i,j)\in A} f_{ij} y_{ij} & \\
& \displaystyle\sum_{p\in\mathcal{P}\,:\,(i,j)\in p} f_p \leq u_{ij} y_{ij} & (i,j) \in A \\
& \displaystyle\sum_{p\in\mathcal{P}^k} f_p = d^k & k \in K \\
& f_p \geq 0 & p \in \mathcal{P} = \cup_{k\in K}\mathcal{P}^k \\
& y_{ij} \in [0,1] & (i,j) \in A
\end{array}
$$

Even more disaggregated Dantzig–Wolfe(cont.d)

- The standard DW approaches use a blatantly "wrong" representation of $\mathcal{Y}$: $2^{|A|}$ variables and one constraint of the form

$$y_{ij} \longrightarrow \sum_{s \in \mathcal{S}^y} \bar{y}_{ij}^s \theta_s \ , \ \sum_{s \in \mathcal{S}^y} \theta_s = 1 \ , \ \theta_s \geq 0 \quad s \in \mathcal{S}^y$$

versus $2|A|$ variables and $|A|$ constraints of the form

$$y_{ij} \longrightarrow 0 \cdot \theta^{ij,0} + 1 \cdot \theta^{ij,1} \ , \ \theta^{ij,0} + \theta^{ij,1} = 1 \ , \ \theta^{ij,0} \geq 0 \ , \ \theta^{ij,1} \geq 0$$

- This is a special case, in fact it corresponds to decompose the unitary hypercube $\mathcal{Y}$ in A components. What could you do when this trick doesn't work? (e.g. adding $\sum_{(i,j) \in A} y_{ij} \leq \mathrm{C}$) $\Rightarrow$ "Easy" components approach
- Our approach does not "column generate" the $\mathcal{Y}$ set and uses a full decription of it

Stabilized Column Generation

Stabilized Dantzig–Wolfe

- Drawbacks of DW approach: possible unboundedness, instability and slow in pratice (see [Frangioni, 2002])
- Using a compact notation rewrite the problem as

$$(\Omega) \quad \max_u \{ c(u) : \; Au = b, \; u \in U \} \, ,$$

  with c concave and U convex

- To overcome these convergence problems $\Rightarrow$ Stabilized Dantzig–Wolfe (see [Frangioni, 2009]) = Generalized Augmented Lagrangian of Ω

$$(\Delta_{\mathcal{B},\bar{x},t}) \quad \max_u \{ c(u) + \bar{x}(b - Au) - D_t^*(Au - b) : u \in \mathrm{co}\mathcal{B} \}$$

Disaggregated stabilized Dantzig–Wolfe

Consider the following decomposable problem:

$$(\Omega) \quad \max \left\{ \sum_{k \in K} c^k(u^k) \,:\, \sum_{k \in K} A^k u^k = b \,,\; u^k \in U^k \,,\; k \in K \right\}$$

Then, the corresponding disaggregated master problem is

$$(\Delta_{\mathcal{B},\bar{x},t}) \quad \max \Big\{ \textstyle\sum_{k \in K} c^k(u^k) + \bar{x}(b - \sum_{k \in K} A^k u^k) - D_t^*(\sum_{k \in K} A^k u^k - b) \,:\, u \in \mathrm{co}\mathcal{B} \,,\; k \in K \Big\}$$

Stabilized Column Generation

+ "Easy" Components

What's special here?

Consider the following structured optimization problem:

$$(\Omega) \quad \max_{u_1, u_2} \{ c_1 u_1 + c_2(u_2) : \; u_1 \in U^1 \, , \; G(u_2) \leq g \, , \; A_1 u_1 + A_2 u_2 = b \}$$

where

$$\max_{u_1} \{ (c_1 - \bar{x} A_1) u_1 : \; u_1 \in U^1 \}$$

is "difficult for some reason" (efficient but "totally obscure" black box), while

$$\max_{u_2} \{ c_2(u_2) - (\bar{x} A_2) u_2 : \; G(u_2) \leq g \}$$

"easy" because a compact convex formulation is known

- Usual approach: disregard differences. Our approach: treat "easy" components specially
- Easy: insert explicit "full" description of second component in the master problem
- Master problem larger (at start), but with "perfect" information

In formulae

The master problem: abstract form

$$
(\Omega_{\mathcal{B},\bar{x},t}) \left| \begin{array}{ll} \max\limits_{z,u_1,u_2} & c_1 u_1 + c_2(u_2) + \bar{x}z - D_t^*(-z) \\ & z = b - A_1 u_1 - A_2 u_2 \\ & u_1 \in \mathrm{co}\mathcal{B} \,,\; u_2 \in U^2 \end{array} \right.
$$

and implementable form

$$
(\Delta_{\mathcal{B},\bar{x},t}) \left| \begin{array}{ll} \max\limits_{z,\theta,u_2} & c_1 \big(\sum_{\bar{u}_1 \in \mathcal{B}} \bar{u}_1 \theta_{\bar{u}_1} \big) + c_2(u_2) + \bar{x}z - D_t^*(-z) \\ & z = b - A_1 \big(\sum_{\bar{u}_1 \in \mathcal{B}} \bar{u}_1 \theta_{\bar{u}_1} \big) - A_2 u_2 \\ & \theta \in \Theta \,,\; G(u_2) \leq g \end{array} \right.
$$

Extensions

- Multiple easy/hard components: trivial
- Constrained case: $x \in X = \{ x : \; Hx \leq h \}$

$$(\Delta_{\mathcal{B},\bar{x},t}) = \left\{ \begin{array}{ll} \max\limits_{\substack{z,\theta \\ u_2,w}} & c_1\big(\sum_{\bar{u}_1 \in \mathcal{B}} \bar{u}_1 \theta_{\bar{u}_1}\big) + c_2(u_2) + wh + \bar{x}z - D_t^*(-z) \\ & z = b + wH - A_1\big(\sum_{\bar{u}_1 \in \mathcal{B}} \bar{u}_1 \theta_{\bar{u}_1}\big) - A_2 u_2 \\ & \theta \in \Theta \,,\; G(u_2) \leq g \,,\; w \geq 0 \end{array} \right.$$

- Global lower bound $l \leq \min f$: (c_2 and G linear)

$$(\Delta_{\mathcal{B},\bar{x},t}) = \left\{ \begin{array}{ll} \max\limits_{\substack{z,\theta' \\ u'_2,\rho}} & c_1\big(\sum\limits_{\bar{u}_1 \in \mathcal{B}} \bar{u}_1 \theta'_{\bar{u}_1}\big) + c_2 u'_2 - l(1-\rho) + \bar{x}z - D_t^*(-z) \\ & z = \rho b - A_1\big(\sum_{\bar{u}_1 \in \mathcal{B}} \bar{u}_1 \theta'_{\bar{u}_1}\big) - A_2 u'_2 \\ & \sum_{\bar{u}_1 \in \mathcal{B}} \theta'_{\bar{u}_1} = \rho \\ & \theta \geq 0 \,,\; G u'_2 \leq \rho g \,,\; \rho \in [0,1] \end{array} \right.$$

$(\theta = \theta'/\rho \,,\; u_2 = u'_2/\rho)$

Computational results

Computational results – FC–MMCF

- Which type of master problems do we solve?
  - a *fully aggregated* (FA) version
  - a *partly disaggregated with easy y* (PDE) version
  - a *disaggregated* (DD) version
  - a *disaggregated with easy y* (DE) version
- 48 randomly generated instances:

group	1	2	3	4	5	6	7	8	9	10	11	12
$\|N\|$	20	20	20	20	30	30	30	30	50	50	50	50
$\|A\|$	300	300	300	300	600	600	600	600	1200	1200	1200	1200
$\|K\|$	100	200	400	800	100	200	400	800	100	200	400	800

Description of the instances

- Stopping criterion: $c(u^*) + \bar{x}(b - Au^*) - D^*_{\bar{t}}(Au^* - b) \leq \varepsilon f(\bar{x})$

Computational results – weak formulation

DE			PDE				DD				FA-1				FA-2			
time	f	iter	time	f	iter	gap	time	f	iter	gap	time	f	iter	gap	time	f	iter	gap
0.04	0.00	5	0.03	0.01	6		557	2.54	6200	1e-7	979	3.97	9105	1e-3	7.64	0.75	2383	1e-7
0.08	0.01	6	0.08	0.01	12		772	2.94	3153	6e-3	1000	4.43	4772	3e-2	14.24	1.37	1931	6e-9
0.25	0.01	7	0.57	0.12	52	1e-7	739	2.79	1365	2e-7	862	10.57	5579	3e-3	12.66	1.99	1117	5e-7
0.64	0.03	7	1.06	0.23	50	3e-7	1000	2.27	482	9e-3	1000	14.49	3201	8e-3	42.38	7.74	1714	7e-7
0.10	0.01	7	0.30	0.03	39		665	4.92	5799	4e-3	945	6.15	7538	8e-3	4.12	0.50	834	3e-7
0.25	0.02	10	1.81	0.21	122		498	3.37	1899	7e-8	808	9.76	5599	3e-3	6.36	1.06	664	1e-6
0.45	0.04	8	20.56	1.93	483	2e-7	1000	1.81	415	2e-2	1000	2.58	638	5e-2	134.49	15.00	3795	6e-7
1.10	0.08	9	5.17	1.09	120	1e-7	1000	3.48	378	2e-2	1000	10.08	1134	4e-2	126.29	26.19	2905	8e-7
0.34	0.02	11	34.80	0.78	449	5e-9	1000	1.39	746	5e-3	1000	2.23	1205	4e-2	28.92	2.77	1630	1e-6
0.42	0.05	9	2.39	0.26	89		1000	6.23	1647	3e-2	1000	8.51	2343	5e-2	32.77	5.26	1414	8e-7
0.99	0.10	11	16.03	2.34	271	1e-7	1000	6.18	717	2e-2	1000	11.31	1321	4e-2	80.05	16.48	1848	8e-7
2.19	0.18	10	124.38	13.95	811	6e-7	1000	5.05	278	2e-2	1000	14.63	838	6e-2	233.40	50.47	2851	8e-7

Results for the weak formulation

- $\varepsilon = 1e - 6$ [relative accuracy] , maximum running time $=$ 1000 seconds
- FA-1, PDE, DD and DE are solved with a general-purpose LP solver by Cplex, while in FA-2 the specialized quadratic solver by [Frangioni, 1996] is used

Computational results – weak formulation (cont.d)

Cplex						DE		FA-2	
primal	dual	barrier	p.net.	d.net.	auto	1e-6	1e-12	1e-6	1e-12
0.30	0.13	8.73	0.18	0.23	0.36	0.04	0.04	7.64	7.74
0.89	0.90	21.25	0.58	1.95	2.40	0.08	0.08	14.24	14.37
3.04	10.22	76.24	2.24	16.32	25.44	0.25	0.26	12.66	13.13
8.21	16.56	151.14	4.62	27.58	44.79	0.64	0.64	42.38	49.18
1.09	4.98	42.57	0.74	6.88	10.62	0.10	0.10	4.12	4.19
3.28	24.68	135.57	2.77	29.46	69.86	0.25	0.26	6.36	7.94
53.25	22.58	417.10	8.96	51.45	55.86	0.45	0.45	134.49	137.41
18.74	67.24	1115.22	10.56	99.96	177.40	1.10	1.10	126.29	163.88
19.98	84.33	303.29	3.92	112.71	187.37	0.34	0.35	28.92	42.71
7.89	82.64	583.52	18.60	259.65	309.74	0.42	0.42	32.77	40.60
38.09	230.79	1952.75	15.85	325.33	690.30	0.99	0.99	80.05	108.94
586.07	459.49	3586.63	51.71	738.23	1266.87	2.19	2.19	233.40	1789.08

Comparison with Cplex for the weak formulation

- Cplex accuracy is $1e-12$, except for barrier which is $1e-10$
- For Lagrangian approaches the relative accuracy is set to both $1e-$

Computational results – strong formulation (cont.d)

1e−6					1e−8			1e−10			1e−12			
time	f	add	iter	gap	time	iter	gap	time	iter	gap	time	f	add	iter
31.69	0.05	0.96	77	1e-7	57.73	143	4e-9	62.07	170	3e-11	63.78	0.11	1.10	181
47.53	0.04	2.04	30	3e-7	51.22	33	2e-9	51.37	33		51.38	0.05	2.06	33
28.98	0.07	2.70	24	2e-7	29.15	25		29.15	25		29.16	0.07	2.74	25
65.31	0.14	6.58	20	3e-8	65.67	21		65.68	21		65.69	0.15	6.61	21
25.93	0.04	0.89	47	8e-8	28.28	51	3e-9	32.00	57		32.00	0.06	0.93	57
27.97	0.09	1.48	36	4e-7	55.43	51	4e-10	56.01	52	1e-11	56.28	0.12	1.60	52
20.80	0.09	1.80	21	2e-8	20.84	21	2e-9	25.69	24		25.69	0.11	1.84	24
132.60	0.24	10.03	23	8e-8	132.74	23		132.76	23		132.78	0.24	10.09	23
2.47	0.06	0.48	26	2e-10	2.47	26	2e-10	2.57	27	3e-12	2.66	0.06	0.49	27
245.91	0.26	4.18	59	1e-7	295.56	72	4e-9	333.22	84	2e-11	337.38	0.39	4.54	86
283.71	0.43	7.24	39	7e-8	442.56	55	2e-9	506.83	63	5e-12	507.52	0.71	7.78	63
241.84	0.52	11.85	24	2e-11	241.88	24	2e-11	241.92	24	2e-11	253.59	0.55	11.98	25

Results for DE with varying precision

- Four accuracy settings: $1e-6$, $1e-8$, $1e-10$ and $1e-12$ for DE approach

Computational results – strong formulation (cont.d)

opt				$20 \cdot \lvert K \rvert$				Rmv $= 20$				Sep $= 10$			
time	add	it	gap	time	add	it	gap	time	add	it	gap	time	add	it	gap
31.69	0.96	77	1e-7	289.41	2.27	841	7e-7	104.60	1.20	218	2e-7	72.96	1.35	194	1e-6
47.53	2.04	30	3e-7	3000.76	7.67	1585	3e-4	1564.82	4.99	803	4e-5	363.67	4.12	159	3e-7
28.98	2.70	24	2e-7	1125.93	6.73	726	4e-7	2585.05	7.82	796	1e-6	141.61	5.51	65	1e-6
65.31	6.58	20	3e-8	81.33	6.68	20	3e-8	17415.68	28.00	2121	8e-5	669.34	18.82	78	5e-7

Effect of different algorithmic parameters settings on DE

- Opt: maximum size of bundle is set to $50\dot{\lvert}K\rvert$, constraints violation is checked at every iteration (Sep $= 1$) and constraints whose Lagrangian multiplier is zero are never removed (Rmv $= 0$)

Computational results – strong formulation (cont.d)

Cplex				DE		FA-2					FA-V				
primal	dual	net.	barr.	1e-6	1e-12	time	f	add	it	gap	time	f	add	it	gap
12	10	11	15	31.69	63.78	410	12	7	14880	9e-7	2.50	0.47	0.45	875	9e-3
64	53	61	71	47.53	51.38	1855	19	16	11141	3e-6	5.83	1.15	1.15	842	2e-2
139	114	132	157	28.98	29.16	1254	32	20	9035	1e-6	11.91	2.28	2.24	796	3e-2
559	456	531	587	65.31	65.69	1732	100	67	12940	1e-6	25.76	5.07	4.96	760	4e-2
46	39	43	60	25.93	32.00	322	12	10	10320	1e-6	5.53	0.88	1.13	871	8e-3
147	132	144	209	27.97	56.28	294	15	9	5300	1e-6	11.88	2.13	2.38	831	9e-3
509	301	478	648	20.80	25.69	5033	169	155	27231	1e-6	25.91	4.50	5.37	794	3e-3
2329	1930	2302	2590	132.60	132.78	3122	192	169	14547	1e-6	51.35	8.58	10.63	760	4e-2
196	131	156	304	2.47	2.66	344	20	12	7169	1e-6	11.61	1.99	2.30	827	3e-3
926	708	862	1174	245.91	337.38	2256	111	118	17034	2e-5	28.50	4.95	6.08	869	1e-2
2706	2167	2542	3272	283.71	507.52	5475	192	249	15061	3e-6	57.86	9.23	13.00	817	2e-2
11156	8908	11675	11683	241.84	253.59	11863	349	413	13953	1e-6	108.75	16.78	24.07	765	2e-2

Comparison with Cplex and Volume for the strong formulation

- Fa–V: a fully disaggregated model solved with the volume algorithm
- Sep is set to 100 and 10, respectively, for FA–2 and FA–V

Conclusions

Conclusions

- The new approach is based on decomposable structure of some NDO problems, in particular the combinatorial ones solved via Lagrangian relaxation
- The structure of the "easy" components is highly beneficial, as the comparison between FA-2, PDE, DD and DE shows. The "easy" versions far outperform the "difficult" ones
- The DE approach significantly outperforms Cplex in most cases
- The exploitation of the "easy" structure can be beneficial in other applications (e.g. Unit Commitment)

References

- [Frangioni, 1996] A. Frangioni. Solving Semidefinite Quadratic problems Within Nonsmooth Optimization Algortithms. *Computer & Operations Research*, 21:1099–1118, 1996
- [Frangioni, 2002] A. Frangioni. Generalized Bundle Methods *SIAM Journal on Optimization*, 13(1):117–156, 2002
- [Frangioni, 2009] H. Ben Amor, J. Desrosiers and A. Frangioni *On the Choice of Explicit Stabilizing Terms in Column Generation* Discrete Applied Mathematics 157(6):1167– 1184, 2009
- [Frangioni, 2011] A. Frangioni, E. Gorgone. Generalized Bundle Methods for Sum–Functions with "Easy" Components: Applications to Multicomodity Network Design *submitted to Math. Prog.*