Large-scale optimization with BlockIP, a specialized IP solver for block-angular problems

J. Castro

Group of Numerical Optimization and Modelling
Department of Statistics and Operations Research
Universitat Politècnica de Catalunya
Barcelona

1st Brazilian Workshop on IPMs
27–28 April 2015

Supported by MTM2012-31440 grant

Outline

Outline

Block-angular problems

Modelling tool

- Multiperiod, multicommodity problems.
- Stochastic problems (two-stage, multi-stage optimization).
- Linking constraints.

Block-angular problems

Modelling tool

- Multiperiod, multicommodity problems.
- Stochastic problems (two-stage, multi-stage optimization).
- Linking constraints.

Applications

- Logistics
- Telecommunications
- Big-data.
- Energy

Block-angular problems

Modelling tool

- Multiperiod, multicommodity problems.
- Stochastic problems (two-stage, multi-stage optimization).
- Linking constraints.

Applications

- Logistics
- Telecommunications
- Big-data.
- Energy

Size

- Very large-scale problems

IPMs successful for very large-scale problems...

Index of /contrib/IPWS2008 - Mozilla Firefox

File Edit View History Bookmarks Tools Help

http://miplib.zib.de/contrib/IPWS2008/

Interior Point Workshop LP Instances

These instances have been collected for the Matheon Workshop on
"Perspectives in Interior Point Methods for Solving Linear Programs"
held at ZIB on January, 31st 2008.

They are all integer programming models with the integrality constraints dropped.

Name	Variables	Constraints	Non Zeros	Description
zib01	12,471,400	5,887,041	49,877,768	Group Channel Routing on a 3D Grid Graph (Chip-Bus-Routing)
zib02	37,709,944	9,049,868	146,280,582	Group Channel Routing on a 3D Grid Graph (different model)
zib03	29,128,799	19,731,970	104,422,573	Steiner-Tree-Packing on a 3D Grid Graph
zib04	37,423	7,433,543	69,004,977	Integrated WLAN Transmitter Selection and Channel Assignment
zib05	9,253,265	9,808	349,424,637	Duty Scheduling with base constraints

All instances with the exception of zib05 are already preprocessed by CPLEX 11.

As of March 16th, 2008, it was not possible to solve zib03 on a 256 GB machine with either CPLEX, MOSEK, or BPMPD.

The logs directory contains some log files from different solvers. The logs are not complete, and the times are *not* comparable.

In case you have questions or can solve zib03 to optimality please contact Thorsten Koch

Imprint

Name	Last modified	Size	Description
logs/	16-Mar-2008 19:37	-	
zib01.mps.gz	16-Mar-2008 11:56	255M	MPS format instance
zib02.mps.gz	15-Mar-2008 11:59	882M	MPS format instance
zib03.mps.gz	15-Mar-2008 12:00	620M	MPS format instance
zib04.mps.gz	25-Jan-2008 11:49	258M	MPS format instance
zib05.mps.gz	25-Jan-2008 11:43	1177M	MPS format instance

Apache/1.3.31 Server at miplib.zib.de Port 80

Done

... but some problems too-large for standard IPMs

Specialized vs standard IPMs

- Standard IPMs (CPLEX, XPRESS, MOSEK...) rely on Cholesky
- Specialized IPMs use PCG for systems of equations.
- Preconditioners are instrumental for efficiency.

... but some problems too-large for standard IPMs

Specialized vs standard IPMs

- Standard IPMs (CPLEX, XPRESS, MOSEK...) rely on Cholesky
- Specialized IPMs use PCG for systems of equations.
- Preconditioners are instrumental for efficiency.

Some preconditioners in IPMs

- Splitting preconditioners (Resende, Veiga 1993; Frangioni, Gentile 2004; Oliveira, Sorensen 2005; Bocanegra, Campos, Oliveira 2007)
- Constraints preconditioners (Keller, Gould, Wathen 2000; Gondzio et al. 2007; Gondzio 2012)
- Partial Cholesky (Bellavia et al. 2013)
- IPM converge even if systems solved approximately (Gondzio 2013)

Outline

Formulation of block-angular problems

For convex separable problems (f_i convex separable)

$$\begin{array}{ll} \min & \sum_{i=0}^{k} f_i(x^i) \\ \text{subject to} & \begin{bmatrix} N_1 & & & \\ & \ddots & & \\ & & N_k & \\ L_1 & \dots & L_k & I \end{bmatrix} \begin{bmatrix} x^1 \\ \vdots \\ x^k \\ x^0 \end{bmatrix} = \begin{bmatrix} b^1 \\ \vdots \\ b^k \\ b^0 \end{bmatrix} \\ & 0 \le x^i \le u^i \quad i = 0, \dots, k. \end{array}$$

Formulation of block-angular problems

For convex separable problems (f_i convex separable)

$$\begin{array}{ll} \min & \sum_{i=0}^{k} f_i(x^i) \\ \text{subject to} & \begin{bmatrix} N_1 & & & \\ & \ddots & & \\ & & N_k & \\ L_1 & \dots & L_k & I \end{bmatrix} \begin{bmatrix} x^1 \\ \vdots \\ x^k \\ x^0 \end{bmatrix} = \begin{bmatrix} b^1 \\ \vdots \\ b^k \\ b^0 \end{bmatrix} \\ & 0 \le x^i \le u^i \quad i = 0, \dots, k. \end{array}$$

Particular cases

- Linear: $f_i(x^i) = {c^i}^\top x^i$
- Quadratic: $f_i(x^i) = {c^i}^\top x^i + \frac{1}{2} {x^i}^\top Q_i x^i$, Q_i diagonal

Formulation of block-angular problems

For convex separable problems (f_i convex separable)

$$\min \quad \sum_{i=0}^{k} f_i(x^i)$$

$$\text{subject to} \quad \begin{bmatrix} N_1 & & & \\ & \ddots & & \\ & & N_k & \\ L_1 & \dots & L_k & I \end{bmatrix} \begin{bmatrix} x^1 \\ \vdots \\ x^k \\ x^0 \end{bmatrix} = \begin{bmatrix} b^1 \\ \vdots \\ b^k \\ b^0 \end{bmatrix}$$

$$0 \leq x^i \leq u^i \quad i = 0, \dots, k.$$

Particular cases

- Linear: $f_i(x^i) = {c^i}^\top x^i$
- Quadratic: $f_i(x^i) = {c^i}^\top x^i + \frac{1}{2} {x^i}^\top Q_i x^i$, Q_i diagonal

Approaches

- Dantzig-Wolfe, cutting planes
- But IPMs can also be used...

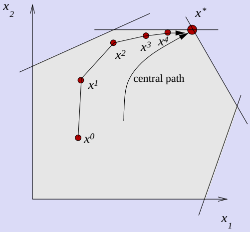
A path-following method

Convex optimization problem

$$(P)\quad \begin{array}{lll} \min & f(x) & \\ \text{s.to} & Ax=b & [\lambda] \\ & 0\le x\le u & [z,w] \end{array}$$

Central path defined by perturbed KKT-μ system

$$\begin{array}{rcl} A^\top\lambda+z-w-\nabla f(x) & = & 0 \\ Ax & = & b \\ (XZe,SWe) & = & (\mu e,\mu e)\quad \mu\in\mathbb{R}^+ \\ (z,w)>0 & & (x,s)>0 \quad s=u-x \end{array}$$



The linear algebra of IPMs

Augmented system

PCG-based IPMs usually solve the augmented system:

$$\begin{bmatrix} -\Theta^{-1} & A^\top \\ A & 0 \end{bmatrix}$$

Normal equations

BlockIP solves normal equations

$$(A\Theta A^\top)\Delta\lambda = g$$

where

$$\Theta = (ZX^{-1} + WS^{-1} + \nabla^2 f(x))^{-1}$$

is a diagonal matrix if problem is separable.

Solving normal equations

Exploiting structure of A and Θ

$$A = \begin{bmatrix} N_1 & & & \\ & \ddots & & \\ & & N_k & \\ L_1 & \dots & L_k & I \end{bmatrix} \qquad \Theta = \begin{bmatrix} \Theta_1 & & & \\ & \ddots & & \\ & & \Theta_k & \\ & & & \Theta_0 \end{bmatrix}$$

$$A\Theta A^\top = \left[\begin{array}{ccc|c} N_1\Theta_1 N_1^\top & & & N_1\Theta_1 L_1^\top \\ & \ddots & & \vdots \\ & & N_k\Theta_k N_k^\top & N_k\Theta_k L_k^\top \\ \hline L_1\Theta_1 N_1^\top & \dots & L_k\Theta_k N_k^\top & \Theta_0 + \sum_{i=1}^k L_i\Theta_i L_i^\top \end{array}\right] = \begin{bmatrix} B & C \\ C^\top & D \end{bmatrix}$$

Solving normal equations

Exploiting structure of A and Θ

$$A = \begin{bmatrix} N_1 & & & \\ & \ddots & & \\ & & N_k & \\ L_1 & \dots & L_k & I \end{bmatrix} \qquad \Theta = \begin{bmatrix} \Theta_1 & & & \\ & \ddots & & \\ & & \Theta_k & \\ & & & \Theta_0 \end{bmatrix}$$

$$A\Theta A^\top = \left[\begin{array}{ccc|c} N_1\Theta_1 N_1^\top & & & N_1\Theta_1 L_1^\top \\ & \ddots & & \vdots \\ & & N_k\Theta_k N_k^\top & N_k\Theta_k L_k^\top \\ \hline L_1\Theta_1 N_1^\top & \dots & L_k\Theta_k N_k^\top & \Theta_0 + \sum_{i=1}^k L_i\Theta_i L_i^\top \end{array}\right] = \begin{bmatrix} B & C \\ C^\top & D \end{bmatrix}$$

The Schur complement

$$\begin{bmatrix} B & C \\ C^\top & D \end{bmatrix} \begin{bmatrix} \Delta\lambda_1 \\ \Delta\lambda_2 \end{bmatrix} = \begin{bmatrix} g_1 \\ g_2 \end{bmatrix} \Longleftrightarrow \begin{array}{rl} (D - C^\top B^{-1} C)\Delta\lambda_2 = & (g_2 - C^\top B^{-1} g_1) \\ B\Delta\lambda_1 = & (g_1 - C\Delta y_2) \end{array}$$

- System with B solved by k Cholesky factorizations.
- Schur complement $S = D - C^\top B^{-1} C$ with large fill-in: system solved by PCG.

The preconditioner

Based on P-regular splitting $S = D - (C^\top B^{-1} C)$ [SIOPT00,COAP07]

Spectral radius of $D^{-1}(C^\top B^{-1} C))$ satisfies $\rho(D^{-1}(C^\top B^{-1} C))) < 1$ and then

$$(D - C^\top B^{-1} C)^{-1} = \left(\sum_{i=0}^{\infty} (D^{-1}(C^\top B^{-1} C))^i \right) D^{-1}$$

The preconditioner

Based on P-regular splitting $S = D - (C^\top B^{-1} C)$ [SIOPT00,COAP07]

Spectral radius of $D^{-1}(C^\top B^{-1} C))$ satisfies $\rho(D^{-1}(C^\top B^{-1} C))) < 1$ and then

$$(D - C^\top B^{-1} C)^{-1} = \left(\sum_{i=0}^{\infty} (D^{-1}(C^\top B^{-1} C))^i \right) D^{-1}$$

Preconditioner M^{-1} obtained truncating the power series at term h

$$\begin{array}{lcll} M^{-1} & = & D^{-1} & \text{if } h = 0, \\ M^{-1} & = & (I + D^{-1}(C^\top B^{-1} C))D^{-1} & \text{if } h = 1. \end{array}$$

The preconditioner

Based on P-regular splitting $S = D - (C^\top B^{-1} C)$ [SIOPT00,COAP07]

Spectral radius of $D^{-1}(C^\top B^{-1} C))$ satisfies $\rho(D^{-1}(C^\top B^{-1} C))) < 1$ and then

$$(D - C^\top B^{-1} C)^{-1} = \left(\sum_{i=0}^{\infty} (D^{-1}(C^\top B^{-1} C))^i \right) D^{-1}$$

Preconditioner M^{-1} obtained truncating the power series at term h

$$\begin{array}{rcll} M^{-1} & = & D^{-1} & \text{if } h = 0, \\ M^{-1} & = & (I + D^{-1}(C^\top B^{-1} C))D^{-1} & \text{if } h = 1. \end{array}$$

Quality of preconditioner depends on

- $\rho < 1$: the farther from 1, the better the preconditioner.

Non-zero Hessians improve the preconditioner I

Proposition. Upper bound for ρ [MP11]

The spectral radius ρ of $D^{-1}(C^\top B^{-1} C)$ is bounded by

$$\rho \leq \max_{j \in \{1,\dots,l\}} \frac{\gamma_j}{\left(\frac{r_j}{v_j}\right)^2 \Theta_{0j} + \gamma_j} < 1,$$

where r is the eigenvector of $D^{-1}(C^\top B^{-1} C)$ associated to ρ; $\gamma_j, j = 1,\dots,l$, and $V = [V_1,\dots,V_l]$, are the eigenvalues and matrix of eigenvectors of $\sum_{i=1}^k L_i \Theta_i L_i^\top$, and $v = V^\top r$.

If $L_i = I$ the bound has the simple and computable form:

$$\rho \leq \max_{j \in \{1,\dots,l\}} \frac{\sum_{i=1}^k \Theta_{ij}}{\Theta_{0j} + \sum_{i=1}^k \Theta_{ij}} < 1.$$

Non-zero Hessians improve the preconditioner II

Proposition. PCG more efficient for quadratic or nonlinear problems

Under some mild conditions, the upper bound of ρ decreases for $\nabla^2 f(x) \succ 0$.

Proposition. PCG extremely efficient if Hessian is large

$$\lim_{\substack{\nabla^2 f_i(x) \to +\infty \\ i=1,\dots,k}} \rho = 0$$

Example: solution of a large (10 million variables, 210000 constraints) with quadratic objective function $x^\top Qx$, for different $Q = \beta I$

		CPLEX-11		Specialized IPM			
Instance	β	it.	CPU	it.	PCG	CPU	f^*
CTA-100-100-1000	0.01	7	29939	10	36	66	-2.6715e+08
CTA-100-100-1000	0.1	7	31328	9	40	61	-2.6715e+09
CTA-100-100-1000	1	8	33367	8	38	56	-2.6715e+10
CTA-100-100-1000	10	9	35220	7	37	51	-2.6715e+11

Quadratic regularizations improve the preconditioner

Standard barrier, proximal-point and quadratic regularization

- $B(x,\mu) \triangleq f(x) + \mu\left(-\sum_{i=1}^{n} \ln x_i - \sum_{i=1}^{n} \ln(u_i - x_i)\right)$
- $B_P(x,\mu) \triangleq f(x) + \frac{1}{2}(x-\bar{x})^\top Q_P (x-\bar{x}) + \mu\left(-\sum_{i=1}^{n} \ln x_i - \sum_{i=1}^{n} \ln(u_i - x_i)\right)$
- $B_Q(x,\mu) \triangleq f(x) + \mu\left(\frac{1}{2}x^\top Q_R x - \sum_{i=1}^{n} \ln x_i - \sum_{i=1}^{n} \ln(u_i - x_i)\right)$

Regularization only affects to Θ matrices

$$\Theta = (ZX^{-1} + WS^{-1} + \nabla^2 f(x))^{-1} \qquad \text{for } B$$
$$\Theta = (Q_P + ZX^{-1} + WS^{-1} + \nabla^2 f(x))^{-1} \qquad \text{for } B_P$$
$$\Theta = (\mu Q_R + ZX^{-1} + WS^{-1} + \nabla^2 f(x))^{-1} \qquad \text{for } B_Q$$

- μQ_R vanishes as we approach the solution, B_Q being equivalent to B.
- B_Q thus preferred to B_P.

Spectral radius ρ can be estimated from Ritz values

… but this was discussed in a previous talk …

Outline

The BlockIP solver: some features

- Efficient implementation of the IPM for block-angular problems.
- For LO, QO, or CO problems.
- Problems in standard or general form.
- Uses Ng-Peyton Sparse Cholesky package (room for improvement).
- Fully written in C++, about 14000 lines of code.
- Many options: computation Ritz values, quadratic regularizations,...
- Comes with different types of matrices: General, oriented and non-oriented Network, Identity, Diagonal, $[I \;\; I]$, $[D_1 \;\; D_2]$.
  - Easy addition of other types of matrices.
  - Extension to Matrix-Free paradigm.

How to input a problem? 1. Callable library

The most efficient option

Example

```
...
// declare N (block constraints matrix) as a Matrix for BlockIP
MatrixBlockIP N;
// declare arc source and destination vectors
int *srcN, *dstN;
// N is created as network matrix
N.create_network_matrix(numArcs, numNodes, srcN, dstN);
// fill srcN and dstN; srcN and dstN allocated by create_network_matrix()
...
// declare L (linking constraints matrix) as a Matrix for BlockIP
MatrixBlockIP L;
// L is created as an identity matrix
L.create_identity_matrix(numArcs);

BlockIP bip; // declare BlockIP problem

double *cost, *qcost, *ub, *rhs;
// creation of BlockIP problem
bip.create_problem(BlockIP::QUADRATIC, cost, qcost, NULL, NULL, ub, rhs,
                   numBlocks, true, &N, true, &L);
// fill cost, qcost, ub, rhs ...
```

How to input a problem? 2. Input file in BlockIP format

Efficient format: vectors and sparse matrices

Example

```
#typeobj 0=linear 1=quadratic 2=nonlinear
1
#number of blocks
2
#sameN 1=yes 0=no
1
#Matrix:  first line m,n,nnz; next nnz lines i,j,a
3 5 7
1 1 1
1 2 1
1 3 1
2 1 -1
2 4 1
3 2 -1
3 5 1
...
```

How to input a problem? 3. Input file in Structured MPS

MPS extension for block-angular problems developed for BlockIP

Example

```
ROWS
E Block1:Cons1
...
E LinkCons1
...
COLUMNS
Block1:Var1 obj 1 Block1:Cons1 1
...
Slack1 LinkCons1 1
```

How to input a problem? 4. SML (Grothey et al. 2009)

- AMPL extension for structured problems.
- SML extended to separable nonlinear problems for BlockIP.

Example (multicommodity transportation problem)

```
block Prod{p in PROD}:
  var Trans {ORIG, DEST} >= 0; # units to be shipped
  minimize total_cost:
         sum {i in ORIG, j in DEST} cost[p,i,j] * Trans[i,j];
  subject to Supply {i in ORIG}:
         sum {j in DEST} Trans[i,j] = supply[p,i];
  subject to Demand {j in DEST}:
         sum {i in ORIG} Trans[i,j] = demand[p,j];
end block;
subject to Multi {i in ORIG, j in DEST}:
         sum {p in PROD} Prod[p].Trans[i,j] <= limit[i,j];
```

Outline

Outline

1. Block-angular structures and large-scale problems
2. IPM for block-angular problems
3. The BlockIP solver
4. Some applications
   - Statistical tabular data confidentiality problems
   - Multi-period facility location problems
   - Generation of random networks
   - Other applications

Minimum Distance Controlled Tabular Adjustment

Statistical table

- Vector $a \in \mathbb{R}^n$ of n cells.
- Satisfies constraints: $Aa = b, l_a \leq a \leq u_a$.

Goal: to find cell perturbations $x \in \mathbb{R}^n$ such that

- Minimizes $||x||_\ell$ for some distance ℓ
- Satisfies $A(x + a) = b,\ l_a \leq x + a \leq u_a \Longleftrightarrow Ax = 0,\ l \leq x \leq u$
- Satisfies protection requirements: $\alpha_i \leq x_i \leq \beta_i \quad i \in \mathcal{S} \subseteq \{1, \dots, n\}$, $0 \notin [\alpha_i, \beta_i]$.

Minimum Distance Controlled Tabular Adjustment

Statistical table

- Vector $a \in \mathbb{R}^n$ of n cells.
- Satisfies constraints: $Aa = b, l_a \leq a \leq u_a$.

Goal: to find cell perturbations $x \in \mathbb{R}^n$ such that

- Minimizes $||x||_\ell$ for some distance ℓ
- Satisfies $A(x+a) = b,\ l_a \leq x+a \leq u_a \Longleftrightarrow Ax = 0,\ l \leq x \leq u$
- Satisfies protection requirements: $\alpha_i \leq x_i \leq \beta_i \quad i \in \mathscr{S} \subseteq \{1,\ldots,n\}$, $0 \notin [\alpha_i, \beta_i]$.

Optimization problem

$$\begin{array}{ll} \min\limits_x & ||x||_\ell \\ \text{s. to} & Ax = 0 \\ & l \leq x \leq u \\ & \alpha_i \leq x_i \leq \beta_i \quad i \in \mathscr{S} \end{array}$$

Block-angular structure of 3D tables: cube/box of data

Example: Profession × County × Sex

A 2D table for each sex, plus a third table for totals

Males

	C1	C2	C3	Sums
P1	230	210	250	690
P2	180	190	210	580
P3	150	140	160	450
Sums	560	540	620	1720

Females

	C1	C2	C3	Sums
P1	184	168	200	552
P2	144	152	168	464
P3	120	112	128	360
Sums	448	432	496	1376

Sums

	C1	C2	C3	Sums
P1	414	378	450	1242
P2	324	342	378	1044
P3	270	252	288	810
Sums	1008	972	1116	3096

Different problems for three distances

Linear Problem: $\nabla^2 f(x) = 0$, twice the number of variables

$$\|x\|_{\ell_1} = \sum_{i=1}^{n} |x_i| = \sum_{i=1}^{n} (x_i^+ + x_i^-)$$

Quadratic Problem: $\nabla^2 f(x) = 2I$

$$\|x\|_{\ell_2}^2 = \sum_{i=1}^{n} x_i^2$$

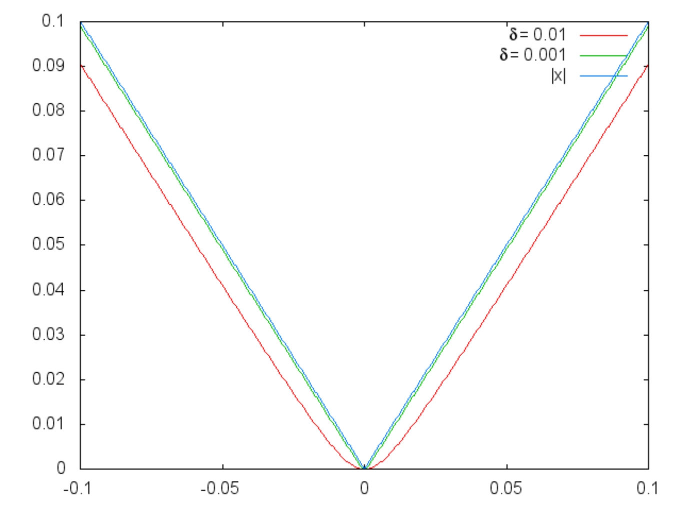
Nonlinear Problem: $\nabla^2 f(x) \succ 0$

$$\|x\|_{\ell_1} = \sum_{i=1}^{n} |x_i| \approx \sum_{i=1}^{n} \phi_\delta(x_i)$$

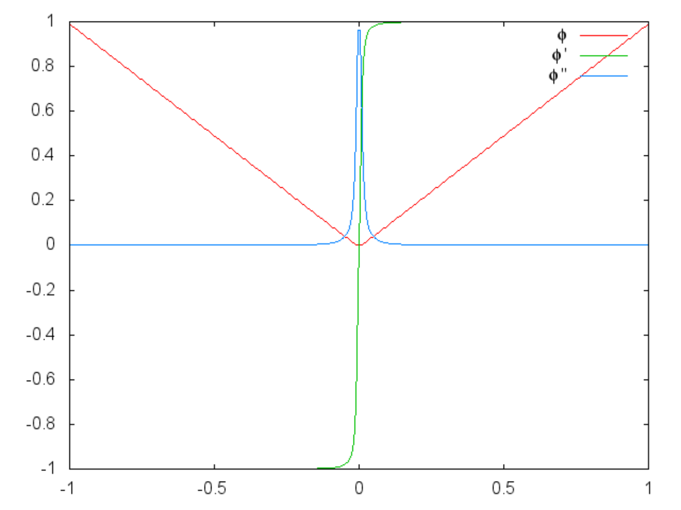
Pseudo-Huber function ϕ_δ approximates absolute value (Fountoulakis, Gondzio 2014):

$$\phi_\delta(x_i) = \sqrt{\delta^2 + x_i^2} - \delta \qquad \delta \approx 0$$

Plots of and $|x|$ and ϕ for some δ



Plots of ϕ, ϕ' and ϕ'' for $\delta = 0.01$



Results for ℓ_1

	Dimensions		BlockIP	CPLEX 12.5
Instance	constraints	variables	CPU	CPU
25-25-25	1850	31875	4	1
25-25-50	3075	63125	12	2
25-50-25	3100	63750	19	2
25-50-50	4950	126250	61	10
50-25-25	3100	63750	28	1
50-25-50	4950	126250	1	7
50-50-25	4975	127500	33	9
50-50-50	7450	252500	16	41
100-100-100	29900	2010000	8	986
100-100-200	49800	4010000	25	2262
200-100-200	79800	8020000	49	8789
200-200-200	119800	16040000	144	64521
500-500-50	299950	25250000	424	19595
500-50-500	299500	25025000	227	17415

Results for ℓ_2

	Dimensions		BlockIP	CPLEX 12.5
Instance	constraints	variables	CPU	CPU
25-25-25	1850	16250	0.0	0.8
25-25-50	3075	31875	0.1	1.4
25-50-25	3100	32500	0.1	1.2
25-50-50	4950	63750	0.1	5.8
50-25-25	3100	32500	0.1	1.2
50-25-50	4950	63750	0.1	4.2
50-50-25	4975	65000	0.1	5.1
50-50-50	7450	127500	0.2	19
100-100-100	29900	1010000	3	874
100-100-200	49800	2010000	6	1802
200-100-200	79800	4020000	11	7319
200-200-200	119800	8040000	29	65467
500-500-50	299950	12750000	91	15437
500-50-500	299500	12525000	28	14784

Results for pseudo-Huber in small instances

Pseudo-Huber more efficient since $\nabla^2 f \succ 0$

	Dimensions		BlockIP		BlockIP ℓ_1	
Instance	const.	variables	CPU	PCG	CPU	PCG
25-25-25	1850	16250	1	3285	4	16475
25-25-50	3075	31875	2	2940	12	22430
25-50-25	3100	32500	2	2525	19	34863
25-50-50	4950	63750	5	4658	61	57641
50-25-25	3100	32500	2	2404	28	53667
50-25-50	4950	63750	4	4392	1	526
50-50-25	4975	65000	4	3298	33	28669
50-50-50	7450	127500	6	1831	16	5523

- Other state-of-the-art convex solvers could not solve these instances.
- Larger instances neither could be solved with BlockIP.

Outline

The multi-period facility location problem [WP-15]

Data: sets and parameters

- T: Set of time periods in the planning horizon, $k = |T|$.
- I: Set of candidate locations for facilities, $n = |I|$.
- J: Set of customers, $m = |J|$.

- f_i^t: Cost for operating a facility at location i at period t.
- c_{ij}^t: Unitary transportation cost from facility i to customer j at period t.
- h_j^t: Unitary shortage cost at customer j at period t.

- d_j^t: Demand of customer j at period t.
- q_i: Capacity of a facility located at i.
- p^t: Maximum number of facilities operating at period t.

Variables

- $y_i^t \in \{0,1\}$: if 1 a facility is operating at i during period t; 0 otherwise. Design variables
- x_{ij}^t: Amount shipped from facility i to customer j at period t.
- z_j^t: Shortage of customer j at period t.

The multi-period facility location problem: formulation

Formulation

$$
\begin{aligned}
\min \quad & \sum_{t\in T}\left(\sum_{i\in I} f_i^t y_i^t + \sum_{i\in I}\sum_{j\in J} c_{ij}^t x_{ij} + \sum_{j\in J} h_j^t z_j^t\right), & \\
\text{subject to} \quad & \sum_{i\in I} x_{ij}^t + z_j^t = d_j^t, & t\in T, j\in J, \\
& \sum_{j\in J} x_{ij}^t \le q_i y_i^t, & t\in T, i\in I, \\
& \sum_{i\in I} y_i^t \le p^t, & t\in T, \\
& y_i^t \le y_i^{t+1}, & t\in T\setminus\{k\}, i\in I, \\
& y_i^t \in \{0,1\}, & t\in T, i\in I, \\
& x_{ij}^t \ge 0, & t\in T, i\in I, j\in J \\
& z_j^t \ge 0, & t\in T, j\in J.
\end{aligned}
$$

Equivalent formulation

Projection on y-space: master problem

$$\begin{aligned}
\min_{y} \quad & \sum_{t\in T}\sum_{i\in I} f_i^t y_i^t + Q(y), \\
\text{subject to} \quad & \sum_{i\in I} y_i^t \le p^t, && t\in T, \\
& y_i^t \le y_i^{t+1}, && t\in T\setminus\{k\}, i\in I, \\
& y_i^t \in \{0,1\}, && t\in T, i\in I.
\end{aligned}$$

Subproblem $Q(y)$

$$\begin{aligned}
Q(y) = \quad \min_{x} \quad & \sum_{t\in T}\left(\sum_{j\in J}\sum_{i\in I} c_{ij}^t x_{ij}^t + \sum_{j\in J} h_j^t z_j^t\right), \\
\text{subject to} \quad & \sum_{i\in I} x_{ij}^t + z_j^t = d_j^t, && t\in T, j\in J, \\
& \sum_{j\in J} x_{ij}^t \le q_i y_i^t, && t\in T, i\in I, \\
& x_{ij}^t \ge 0, && t\in T, i\in I, j\in J, \\
& z_j^t \ge 0, && t\in T, j\in J.
\end{aligned}$$

Solution of equivalent formulation

- $Q(y)$ is convex (and piecewise linear if y was continuous).
- $Q(y)$ can thus be lower-approximated by cutting planes.
- Cutting planes obtained by solving subproblem $Q(y^l)$ at some point y^l.
- Generated cutting planes are iteratively added to the master problem.
- This is basically: Benders decomposition.
- Three properties of subproblem:
  - Separable in $k = |T|$ subproblems, one for time period.
  - Each of the k subproblems has block-angular structure, BlockIP can be used.
  - No need to optimally solve each subproblem: inexact cuts. BlockIP can thus be effectively used, avoiding last expensive IPM iterations with PCG.

Solution of large-scale facility location instances

- World-wide problems: 100s locations, 100000s of customers.

Optimality tolerance 10^{-5} for the subproblems.

						BlockIP			BarOpt		
n	m	k	const.	bin.var.	cont. var.	iter.	gap	CPU	iter.	gap	CPU
100	100000	1	100101	100	10100000	3	0.0000	25.39	3	0.0000	42.01
100	100000	2	200302	200	20200000	4	0.0000	63.36	4	0.0000	129.91
100	100000	3	300503	300	30300000	6	0.0000	166.86	6	0.0000	336.20
100	500000	1	500101	100	50500000	2	0.0003	552.66	2	0.0011	474.67
100	500000	2	1000302	200	101000000	3	0.0000	2534.44	3	0.0004	1391.85
100	500000	3	1500503	300	151500000	4	0.0071	5524.19	4	0.0004	3932.30
100	1000000	1	1000101	100	101000000	2	0.0001	1292.37	2	0.0002	1221.85
100	1000000	2	2000302	200	202000000	2	0.0002	3124.20		†	
100	1000000	3	3000503	300	303000000	3	0.0000	11218.75		†	
200	100000	1	100201	200	20100000	3	0.0000	25.45	3	0.0000	102.69
200	100000	2	200602	400	40200000	4	0.0000	63.59	4	0.0000	310.33
200	100000	3	301003	600	60300000	6	0.0000	167.76	6	0.0000	787.70
200	500000	1	500201	200	100500000	3	0.0000	1402.13	3	0.0000	1064.93
200	500000	2	1000602	400	201000000	5	0.0000	5814.12		‡	
200	500000	3	1501003	600	301500000	6	0.0000	8652.29		†	
200	1000000	1	1000201	200	201000000		*			†	
200	1000000	2	2000602	400	402000000	4	0.0001	14514.18		†	
200	1000000	3	3001003	600	603000000	6	0.0001	40744.86		†	

* Preconditioned conjugate gradient solver failed

† CPLEX ran out of memory (required more than 144 Gigabytes of RAM)

‡ Execution aborted

Solution of large-scale facility location instances

- World-wide problems: 100s locations, 100000s of customers.

Optimality tolerance 10^{-3} for the subproblems.

						BlockIP			CPLEX		
n	m	k	const.	bin.var.	cont. var.	iter.	gap	CPU	iter.	gap	CPU
100	100000	1	100101	100	10100000	3	0.0010	17.35	3	0.0002	40.14
100	100000	2	200302	200	20200000	4	0.0007	32.04	4	0.0001	117.45
100	100000	3	300503	300	30300000	5	0.0009	63.31	6	0.0035	373.97
100	500000	1	500101	100	50500000	2	-0.0040	69.17		*	
100	500000	2	1000302	200	101000000	3	0.0006	272.39		*	
100	500000	3	1500503	300	151500000	5	0.0007	760.93		*	
100	1000000	1	1000101	100	101000000	2	-0.0002	123.53	2	0.0008	655.72
100	1000000	2	2000302	200	202000000	2	-0.0010	312.23		†	
100	1000000	3	3000503	300	303000000	3	0.0009	891.81		†	
200	100000	1	100201	200	20100000	3	0.0010	17.39	3	0.0002	100.54
200	100000	2	200602	400	40200000	4	0.0007	32.09	4	0.0001	297.03
200	100000	3	301003	600	60300000	5	0.0009	63.55	6	0.0035	912.05
200	500000	1	500201	200	100500000	3	0.0010	110.02	3	0.0000	1146.60
200	500000	2	1000602	400	201000000	4	0.0010	309.68		‡	
200	500000	3	1501003	600	301500000	6	0.0010	868.43		†	
200	1000000	1	1000201	200	201000000	3	0.0009	729.79		†	
200	1000000	2	2000602	400	402000000	4	0.0010	1109.64		†	
200	1000000	3	3001003	600	603000000	6	0.0009	3254.21		†	

* Repeated solution in master, Benders would not converge

† CPLEX ran out of memory (required more than 144 Gigabytes of RAM)

‡ CPLEX aborted

Outline

Generation of random networks with constraints [EJOR15]

Scope of application

- Important application in the analysis of social networks and complex systems.
- The goal is to generate a sample of (hundreds, thousands of) graphs with some particular properties solving some binary problems by randomly modifying the cost vector.
- Constraints matrices are totally unimodular for many problems.

A particular case: undirected edge-colored graphs

- Generate a graph with different types of edges (different colors), and given number of edges per color.
- Extra constraints can be added, for instance the degree of the nodes, etc.

Formulation of edge-colored generation problem

Parameters and variables

- n: number of nodes.
- C: set of colors.
- d_c: number of edges per color $c \in C$.
- Set of possible undirected edges: $H = \{(i,j) : 1 \le i \le n-1, i < j \le n\}$.
- Decision variables: $x_{ij} \in \{0,1\}, (i,j) \in H$.

Optimization problem

$$\begin{array}{lll} \min & \sum_{(i,j)\in H} \sum_{c\in C} w_{ij}^c x_{ij}^c & \\ & \sum_{c\in C} x_{ij}^c \le 1 & (i,j) \in H \\ & \sum_{(i,j)\in H} x_{ij}^c = d_c, & c \in C \\ & x_{ij}^c \in \{0,1\} & (i,j) \in H, c \in C \end{array}$$

- The constraints of linear relaxation are TUM, so it can be solved as LP.
- This and others random graph problem are shown to have a block-angular structure.

Solution of some edge-colored network generation instances

- Medium-size problems: up to 450 nodes and 450 colors.

CPU time and iterations (in parentheses) of three algorithms.

n	C	var.	constr.	Dual Simplex CPLEX 12.5	Barrier CPLEX 12.5	BlockIP
50	50	61250	100	0.7 (733)	0.2 (13)	0.2 (34)
150	50	558750	200	6.4 (934)	5.3 (26)	3.4 (43)
450	50	5051250	500	126.3 (2462)	90.8 (46)	45.7 (56)
50	150	183750	200	5.0 (1435)	0.9 (14)	0.9 (40)
150	150	1676250	300	58.3 (2503)	18.7 (30)	12.6 (52)
450	150	15153750	600	1175.5 (5243)	384.6 (53)	147.2 (58)
50	450	551250	500	5.2 (1956)	2.9 (15)	3.3 (45)
150	450	5028750	600	378.8 (4822)	64.9 (35)	37.7 (51)
450	450	45461250	900	10926.3 (13047)	1287.4 (52)	409.6 (54)

Outline

Other applications under consideration

Routing in telecommunications networks

- Nonoriented multicommodity network.
- Many OD pairs
- Nonlinear Kleinrock delay function
- Already implemented: good results. Work in progress.

Transportation assignment problem in urban networks

- Similar to routing in telecommunications networks.
- Many OD pairs
- Nonlinear BPR (Bureau of Public Roads) function.
- To be tested soon.

Conclusions

- IP solver for block-angular problems.
- Shown to be very efficient for some applications.
- Many future applications to be tried.

Available for research purposes from
`www-eio.upc.edu/~jcastro/BlockIP.html`

Some references about the IPM and applications

- J. Castro, S. Nasini, F. Saldanha, A cutting-plane approach for large-scale capacitated multi-period facility location using a specialized interior-point method, working paper, to be submitted, 2015.
- J. Castro, S. Nasini, Mathematical programming approaches for classes of random network, *European Journal of Operational Research*, doi:10.1016/j.ejor.2015.03.021, 2015.
- J. Castro, A. Frangioni, C. Gentile, Perspective reformulations of the CTA problem with L2 distances, *Operations Research*, 62 891–909, 2014.
- S. Bocanegra, J. Castro, A.R.L. Oliveira, Improving an interior-point approach for large block-angular problems by hybrid preconditioners, *European Journal of Operational Research*, 231 263–273, 2013.
- J. Castro, J. Cuesta, Improving an interior-point algorithm for multicommodity flows by quadratic regularizations, *Networks*, 59 117–131, 2012.
- J. Castro, Recent advances in optimization techniques for statistical tabular data protection, *European Journal of Operational Research*, 216 257-269, 2012.
- J. Castro, J. Cuesta, Quadratic regularizations in an interior-point method for primal block-angular problems, *Mathematical Programming*, 130 415-445, 2011.
- J. Castro, J. Cuesta, Existence, uniqueness and convergence of the regularized primal-dual central path, *Operations Research Letters*, 38 366–371, 2010.
- J. Castro. An interior-point approach for primal block-angular problems, *Computational Optimization and Applications*, 36 195–219, 2007.
- J. Castro. A specialized interior-point algorithm for multicommodity network flows, *SIAM Journal on Optimization*, 10 852–877, 2000.

Thanks for your attention