A New Approach for Finding a Base for the Splitting Preconditioner for Linear Systems from Interior Point Methods

Porfirio Suñagua S.
Linear Programming

Department of Mathematics FCPN-UMSA

Campinas, April 27-28, 2015

Content

Content

Content

Content

Content

Content

Content:

Penalized LP Problems

- Bounded LP Problem

$$\begin{aligned} \text{Min} \quad & c^T x \\ \text{s.t.} \quad & Ax = b \\ & Ex + v = u \\ & (x, v) \geq 0 \end{aligned}$$

- Sub-Problem with penalty parameter, $\rho > 0$

$$\begin{aligned} (P_\rho) \quad \text{Min} \quad & c^T x + \frac{\rho}{2}\left(\|b - Ax\|^2 + \|u - Ex - v\|^2\right) \\ \text{s.t.} \quad & (x, v) \geq 0 \end{aligned}$$

Such that, $\mathbf{x} \to \mathbf{x}^*$, $\rho \to \infty \Rightarrow \rho(\|b - Ax\|^2 + \|u - Ex - v\|^2) \to ?$.

- The objective function with logarithmic barrier parameter $\mu \geq 0$, is

$$\begin{aligned} \mathcal{P}(x, v) \quad = \quad & c^T x + \frac{\rho}{2}\left(\|b - Ax\|^2 + \|u - Ex - v\|^2\right) \\ & - \mu\left(\sum_{j=1}^{n} \log x_j + \sum_{j \in \mathcal{C}} \log v_j\right) \end{aligned}$$

Penalized LP Problems

- Bounded LP Problem

$$\begin{aligned} \text{Min} \quad & c^T x \\ \text{s.t.} \quad & Ax = b \\ & Ex + v = u \\ & (x, v) \geq 0 \end{aligned}$$

- Sub-Problem with penalty parameter, $\rho > 0$

$$(P_\rho) \quad \text{Min} \quad c^T x + \frac{\rho}{2}\left(\|b - Ax\|^2 + \|u - Ex - v\|^2\right)$$
$$\text{s.t.} \quad (x, v) \geq 0$$

Such that, $\mathbf{x} \to \mathbf{x}^*$, $\rho \to \infty \Rightarrow \rho(\|b - Ax\|^2 + \|u - Ex - v\|^2) \to ?$.

- The objective function with logarithmic barrier parameter $\mu \geq 0$, is

$$\begin{aligned} \mathcal{P}(x, v) = \; & c^T x + \frac{\rho}{2}\left(\|b - Ax\|^2 + \|u - Ex - v\|^2\right) \\ & - \mu \left(\sum_{j=1}^{n} \log x_j + \sum_{j \in \mathcal{C}} \log v_j \right) \end{aligned}$$

Penalized LP Problems

- Bounded LP Problem

$$\begin{aligned} \text{Min} \quad & c^T x \\ \text{s.t.} \quad & Ax = b \\ & Ex + v = u \\ & (x, v) \geq 0 \end{aligned}$$

- Sub-Problem with penalty parameter, $\rho > 0$

$$\begin{aligned} (P_\rho) \quad \text{Min} \quad & c^T x + \frac{\rho}{2}\left(\|b - Ax\|^2 + \|u - Ex - v\|^2\right) \\ \text{s.t.} \quad & (x, v) \geq 0 \end{aligned}$$

Such that, $\mathbf{x} \to \mathbf{x}^*$, $\rho \to \infty \Rightarrow \rho(\|b - Ax\|^2 + \|u - Ex - v\|^2) \to 0$.

- The objective function with logarithmic barrier parameter $\mu \geq 0$, is

$$\begin{aligned} \mathcal{P}(x, v) \quad = \quad & c^T x + \frac{\rho}{2}\left(\|b - Ax\|^2 + \|u - Ex - v\|^2\right) \\ & - \mu\left(\sum_{j=1}^{n} \log x_j + \sum_{j \in \mathcal{C}} \log v_j\right) \end{aligned}$$

Penalized LP Problems

- Bounded LP Problem

$$\begin{array}{ll} \text{Min} & c^T x \\ \text{s.t.} & Ax = b \\ & Ex + v = u \\ & (x, v) \geq 0 \end{array}$$

- Sub-Problem with penalty parameter, $\rho > 0$

$$\begin{array}{lll} (P_\rho) & \text{Min} & c^T x + \frac{\rho}{2}\left(\|b - Ax\|^2 + \|u - Ex - v\|^2\right) \\ & \text{s.t.} & (x, v) \geq 0 \end{array}$$

Such that, $\mathbf{x} \to \mathbf{x}^*$, $\rho \to \infty \Rightarrow \rho(\|b - Ax\|^2 + \|u - Ex - v\|^2) \to 0$.

- The objective function with logarithmic barrier parameter $\mu \geq 0$, is

$$\begin{aligned} \mathcal{P}(x, v) &= c^T x + \frac{\rho}{2}\left(\|b - Ax\|^2 + \|u - Ex - v\|^2\right) \\ &\quad - \mu\left(\sum_{j=1}^{n} \log x_j + \sum_{j \in \mathcal{C}} \log v_j\right) \end{aligned}$$

Content:

Karush–Kuhn–Tucker conditions

- Perturbed KKT system for $\delta = 1/\rho$

$$\begin{aligned} Ax + \delta y &= b \\ Ex + v - \delta w &= u \\ A^T y + z - E^T w &= c \\ XZe &= \mu e \\ VWe &= \mu e \end{aligned} \tag{1}$$

- Newton system

$$\begin{aligned} Adx + \delta dy = r_p, &\quad r_p = b - Ax - \delta y \\ Edx + dv - \delta dw = r_u, &\quad r_u = u - Ex - v + \delta w \\ A^T dy + dz - E^T dw = r_d, &\quad r_d = c - A^T y - z + E^T w \\ Zdx + Xdz = r_s, &\quad r_s = -XZe + \mu e \\ Wdv + Vdw = r_r, &\quad r_r = -VWe + \mu e \end{aligned} \tag{2}$$

Karush–Kuhn–Tucker conditions

- Perturbed KKT system for $\delta = 1/\rho$

$$\begin{aligned} Ax + \delta y &= b \\ Ex + v - \delta w &= u \\ A^T y + z - E^T w &= c \\ XZe &= \mu e \\ VWe &= \mu e \end{aligned} \tag{1}$$

- Newton system

$$\begin{aligned} Adx + \delta dy = r_p, &\quad r_p = b - Ax - \delta y \\ Edx + dv - \delta dw = r_u, &\quad r_u = u - Ex - v + \delta w \\ A^T dy + dz - E^T dw = r_d, &\quad r_d = c - A^T y - z + E^T w \\ Zdx + Xdz = r_s, &\quad r_s = -XZe + \mu e \\ Wdv + Vdw = r_r, &\quad r_r = -VWe + \mu e \end{aligned} \tag{2}$$

Content:

Search directions

- $dz = X^{-1}(r_s - Zdx), \qquad dw = V^{-1}(r_r - Wdv)$
- $S = V + \delta W, \quad dv = S^{-1}V(r_u - Edx) + \delta S^{-1}r_r$
- Augmented System: $D = (X^{-1}Z + E^T S^{-1} WE)^{-1}$
  $r_1 = r_d - X^{-1}r_s + E^T S^{-1} r_r - E^T S^{-1} W r_u$

$$\begin{pmatrix} -D^{-1} & A^T \\ A & \delta I \end{pmatrix} \begin{pmatrix} dx \\ dy \end{pmatrix} = \begin{pmatrix} r_1 \\ r_p \end{pmatrix} \qquad (3)$$

- Normal Equations

$$(ADA^T + \delta I)dy = r_p + ADr_1 \qquad (4)$$

Search directions

- $dz = X^{-1}(r_s - Zdx), \qquad dw = V^{-1}(r_r - Wdv)$
- $S = V + \delta W, \quad dv = S^{-1}V(r_u - Edx) + \delta S^{-1} r_r$
- Augmented System: $D = (X^{-1}Z + E^T S^{-1} WE)^{-1}$
  $r_1 = r_d - X^{-1} r_s + E^T S^{-1} r_r - E^T S^{-1} W r_u$

$$\begin{pmatrix} -D^{-1} & A^T \\ A & \delta I \end{pmatrix} \begin{pmatrix} dx \\ dy \end{pmatrix} = \begin{pmatrix} r_1 \\ r_p \end{pmatrix} \tag{3}$$

- Normal Equations

$$(ADA^T + \delta I)dy = r_p + ADr_1 \tag{4}$$

Search directions

- $dz = X^{-1}(r_s - Zdx), \qquad dw = V^{-1}(r_r - Wdv)$
- $S = V + \delta W, \quad dv = S^{-1}V(r_u - Edx) + \delta S^{-1} r_r$
- Augmented System: $D = (X^{-1}Z + E^T S^{-1} W E)^{-1}$
  $r_1 = r_d - X^{-1} r_s + E^T S^{-1} r_r - E^T S^{-1} W r_u$

$$\begin{pmatrix} -D^{-1} & A^T \\ A & \delta I \end{pmatrix} \begin{pmatrix} dx \\ dy \end{pmatrix} = \begin{pmatrix} r_1 \\ r_p \end{pmatrix} \tag{3}$$

- Normal Equations

$$(ADA^T + \delta I)dy = r_p + ADr_1 \tag{4}$$

Search directions

- $dz = X^{-1}(r_s - Zdx), \qquad dw = V^{-1}(r_r - Wdv)$
- $S = V + \delta W, \quad dv = S^{-1}V(r_u - Edx) + \delta S^{-1} r_r$
- Augmented System: $D = (X^{-1}Z + E^T S^{-1} WE)^{-1}$
  $r_1 = r_d - X^{-1} r_s + E^T S^{-1} r_r - E^T S^{-1} W r_u$

$$\begin{pmatrix} -D^{-1} & A^T \\ A & \delta I \end{pmatrix} \begin{pmatrix} dx \\ dy \end{pmatrix} = \begin{pmatrix} r_1 \\ r_p \end{pmatrix} \tag{3}$$

- Normal Equations

$$(ADA^T + \delta I)dy = r_p + ADr_1 \tag{4}$$

Search directions

- $dz = X^{-1}(r_s - Zdx), \qquad dw = V^{-1}(r_r - Wdv)$
- $S = V + \delta W, \quad dv = S^{-1}V(r_u - Edx) + \delta S^{-1}r_r$
- Augmented System: $D = (X^{-1}Z + E^T S^{-1} WE)^{-1}$
  $r_1 = r_d - X^{-1}r_s + E^T S^{-1} r_r - E^T S^{-1} W r_u$

$$\begin{pmatrix} -D^{-1} & A^T \\ A & \delta I \end{pmatrix} \begin{pmatrix} dx \\ dy \end{pmatrix} = \begin{pmatrix} r_1 \\ r_p \end{pmatrix} \tag{3}$$

- Normal Equations

Improved ill-conditioning of ADA^T

$$(ADA^T + \delta I)dy = r_p + ADr_1 \tag{4}$$

Search directions

- $dz = X^{-1}(r_s - Zdx), \qquad dw = V^{-1}(r_r - Wdv)$
- $S = V + \delta W, \quad dv = S^{-1}V(r_u - Edx) + \delta S^{-1} r_r$
- Augmented System: $D = (X^{-1}Z + E^T S^{-1} W E)^{-1}$

  $r_1 = r_d - X^{-1} r_s + E^T S^{-1} r_r - E^T S^{-1} W r_u$

$$\begin{pmatrix} -D^{-1} & A^T \\ A & \delta I \end{pmatrix} \begin{pmatrix} dx \\ dy \end{pmatrix} = \begin{pmatrix} r_1 \\ r_p \end{pmatrix} \tag{3}$$

- Normal Equations

Improved ill-conditioning of ADA^T

$$(ADA^T + \delta I)dy = r_p + ADr_1 \tag{4}$$

Cholesky's method, or

Cojugate Gradient Methods

Search directions

- $dz = X^{-1}(r_s - Zdx), \qquad dw = V^{-1}(r_r - Wdv)$
- $S = V + \delta W, \quad dv = S^{-1}V(r_u - Edx) + \delta S^{-1} r_r$
- Augmented System: $D = (X^{-1}Z + E^T S^{-1} WE)^{-1}$
  $r_1 = r_d - X^{-1} r_s + E^T S^{-1} r_r - E^T S^{-1} W r_u$

Preconditioning

$$\begin{pmatrix} -D^{-1} & A^T \\ A & \delta I \end{pmatrix} \begin{pmatrix} dx \\ dy \end{pmatrix} = \begin{pmatrix} r_1 \\ r_p \end{pmatrix} \tag{3}$$

- Normal Equations

$$(ADA^T + \delta I) dy = r_p + ADr_1 \tag{4}$$

Preconditioning

Cojugate Gradient Methods
(Preconditioned)

Search directions

- $dz = X^{-1}(r_s - Zdx), \qquad dw = V^{-1}(r_r - Wdv)$
- $S = V + \delta W, \quad dv = S^{-1}V(r_u - Edx) + \delta S^{-1} r_r$
- Augmented System: $D = (X^{-1}Z + E^T S^{-1} W E)^{-1}$
  $r_1 = r_d - X^{-1} r_s + E^T S^{-1} r_r - E^T S^{-1} W r_u$

Splitting Preconditioner

$$\begin{pmatrix} -D^{-1} & A^T \\ A & \delta I \end{pmatrix} \begin{pmatrix} dx \\ dy \end{pmatrix} = \begin{pmatrix} r_1 \\ r_p \end{pmatrix} \tag{3}$$

- Normal Equations

$$(ADA^T + \delta I) dy = r_p + A D r_1 \tag{4}$$

Controlled Cholesky Preconditioner(CCF)

Cojugate Gradient Methods (Preconditioned)

Content

Content:

Splitting Preconditioner

- Given a base of columns B of the matrix A, we split $A = [B, N]$, $D = \text{diag}(D_B, D_N)$. A class of splitting preconditioner suggested by Oliveira and Sorensen [14] for augmented system is

$$M^{-1} = \begin{pmatrix} D_B^{\frac{1}{2}} & 0 & D_B^{-\frac{1}{2}} B^{-1} \\ 0 & D_N^{\frac{1}{2}} & 0 \\ D_B^{\frac{1}{2}} & 0 & 0 \end{pmatrix} \tag{5}$$

- Desirable property

$$D_B^{-\frac{1}{2}} B^{-1} (ADA^T + \delta I) B^{-T} D_B^{-\frac{1}{2}} = I + \delta W_0 W_0^T + WW^T \approx I$$

si $W = D_B^{-\frac{1}{2}} B^{-1} N D_N^{\frac{1}{2}} \approx 0$, where $W_0 = D_B^{-\frac{1}{2}} B^{-1}$.

- In order to not calculate the ADA^T product we can apply preconditioned conjugate gradients method.

Splitting Preconditioner

- Given a base of columns B of the matrix A, we split $A = [B, N]$, $D = \text{diag}(D_B, D_N)$. A class of splitting preconditioner suggested by Oliveira and Sorensen [14] for augmented system is

$$M^{-1} = \begin{pmatrix} D_B^{\frac{1}{2}} & 0 & D_B^{-\frac{1}{2}} B^{-1} \\ 0 & D_N^{\frac{1}{2}} & 0 \\ D_B^{\frac{1}{2}} & 0 & 0 \end{pmatrix} \quad (5)$$

- Desirable property

$$D_B^{-\frac{1}{2}} B^{-1} (ADA^T + \delta I) B^{-T} D_B^{-\frac{1}{2}} = I + \delta W_0 W_0^T + WW^T \approx I$$

si $W = D_B^{-\frac{1}{2}} B^{-1} N D_N^{\frac{1}{2}} \approx 0$, where $W_0 = D_B^{-\frac{1}{2}} B^{-1}$.

- In order to not calculate the ADA^T product we can apply preconditioned conjugate gradients method.

Splitting Preconditioner

- Given a base of columns B of the matrix A, we split $A = [B, N]$, $D = \text{diag}(D_B, D_N)$. A class of splitting preconditioner suggested by Oliveira and Sorensen [14] for augmented system is

$$M^{-1} = \begin{pmatrix} D_B^{\frac{1}{2}} & 0 & D_B^{-\frac{1}{2}} B^{-1} \\ 0 & D_N^{\frac{1}{2}} & 0 \\ D_B^{\frac{1}{2}} & 0 & 0 \end{pmatrix} \quad (5)$$

- Desirable property

$$D_B^{-\frac{1}{2}} B^{-1} (ADA^T + \delta I) B^{-T} D_B^{-\frac{1}{2}} = I + \delta W_0 W_0^T + WW^T \approx I$$

si $W = D_B^{-\frac{1}{2}} B^{-1} N D_N^{\frac{1}{2}} \approx 0$, where $W_0 = D_B^{-\frac{1}{2}} B^{-1}$.

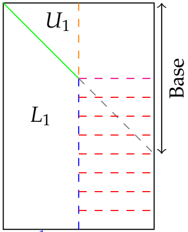
- In order to not calculate the ADA^T product we can apply preconditioned conjugate gradients method.

Content:

Rectangular Factorization

LUstd

$D^{\frac{1}{2}} A^T =$

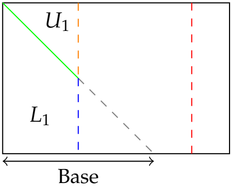


$PD^{\frac{1}{2}} A^T Q = LU$

- Fill-in: The columns of $D^{\frac{1}{2}} A^T$ is pre-ordered by AMD
- LU rectangular multifrontal factorization is applied with threshold $\sigma = 0.1$.
- Vector base: $B = p(1:m)$, where p is the permutation vector of P.

LUret

$AP_0 =$



$PAP_0 Q = LU$

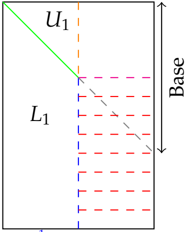
P_0 : Pre-orderer Matrix

- The columns of A is pre-ordered by norm-2 of $AD^{\frac{1}{2}}$ and LI criteria
- LU rectangular factorization is applied to pre-ordered A. Restart if filling.
- Vector Base $B = q(1:m)$, where q is the permutation vector of Q.

Rectangular Factorization

LUstd

$D^{\frac{1}{2}}A^T =$

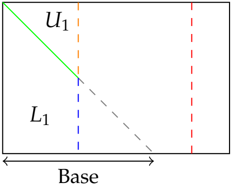


$PD^{\frac{1}{2}}A^TQ = LU$

- Fill-in: The columns of $D^{\frac{1}{2}}A^T$ is pre-ordered by AMD
- LU rectangular multifrontal factorization is applied with threshold $\sigma = 0.1$.
- Vector base: $B = p(1 : m)$, where p is the permutation vector of P.

LUret

$AP_0 =$



$PAP_0Q = LU$

P_0 : Pre-orderer Matrix

- The columns of A is pre-ordered by norm-2 of $AD^{\frac{1}{2}}$ and LI criteria
- LU rectangular factorization is applied to pre-ordered A. Restart if filling.
- Vector Base $B = q(1 : m)$, where q is the permutation vector of Q.

Content:

Hybrid Approach

- Phase I: P : Controlled Cholesky Factorization

$$P^{-1}(ADA^T + \delta I)P^{-T}P^T dy = P^{-1}r. \tag{6}$$

- Phase II: Solve the preconditioned augmented system with splitting preconditioner M^{-1}, is equivalent to solve the preconditioned Normal Equations with preconditioner $P^{-1} = D_B^{-\frac{1}{2}}B^{-1}$.

$$D_B^{-\frac{1}{2}}B^{-1}(ADA^T + \delta I)B^{-T}D_B^{-\frac{1}{2}}d\tilde{y} = D_B^{-\frac{1}{2}}B^{-1}\left(r_p + ADr_1\right) \tag{7}$$

then, $dy = B^{-T}D_B^{-\frac{1}{2}}d\tilde{y}$, and $dx = D(A^T dy - r_1)$.

Hybrid Approach

- Phase I: P : Controlled Cholesky Factorization

$$P^{-1}(ADA^T + \delta I)P^{-T}P^T dy = P^{-1}r. \tag{6}$$

- Phase II: Solve the preconditioned augmented system with splitting preconditioner M^{-1}, is equivalent to solve the preconditioned Normal Equations with preconditioner $P^{-1} = D_B^{-\frac{1}{2}} B^{-1}$.

$$D_B^{-\frac{1}{2}} B^{-1}(ADA^T + \delta I)B^{-T} D_B^{-\frac{1}{2}} d\tilde{y} = D_B^{-\frac{1}{2}} B^{-1}\left(r_p + ADr_1\right) \tag{7}$$

then, $dy = B^{-T} D_B^{-\frac{1}{2}} d\tilde{y}$, and $dx = D(A^T dy - r_1)$.

Content

Content:

NETLIB, QAP, Kennington and KBAPAH Problems

Total of Linear Programming problems considered: 189

Medium Problems			Large Problems
BL	fit1p	pds-30	ken18
BL2	fit2p	pilot87	kra30a[2]
chr22b	GE	qap12	kra30b[2]
chr25a	greenbeb	qap15	nug07-3rd
CO5	ken13	rou20	nug20[2]
CO9	NL	scr20	pds-70
CQ9	nug06-3rd	stocfor3	pds-80
cre-b	nug12		pds-90
cre-d	nug15		pds-100
dfl001	osa-60		ste36a[2]
els19	pds-10		ste36b[2]
ex09	pds-20		ste36c[2]

[2] It may require a huge amount of computational memory

CG Iterations in Phase II of LUstd and LUret

The more bad-conditioned is the coefficient matrix, the number of iterations of conjugate gradient method will be more higher.

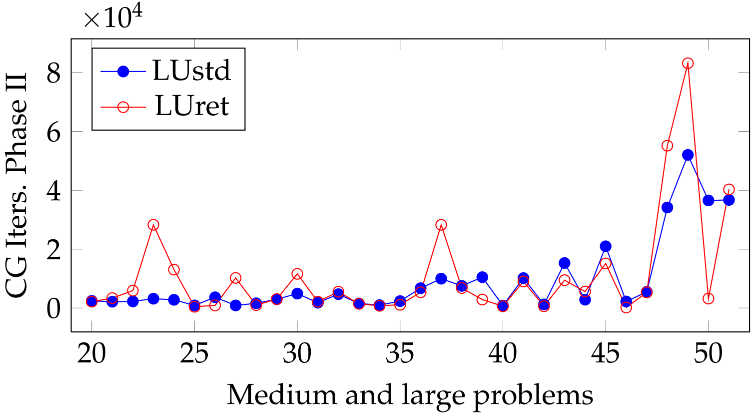


Figure 1: Conjugate Gradient Iterations in Phase II of LUstd and LUret
Average LUstd = 5961.61, Average LUret = 7435.37.

LU Factorization time % of LUstd and LUret

The more bad-conditioned is the coefficient matrix, the time % of *LU* factorization will be more higher.

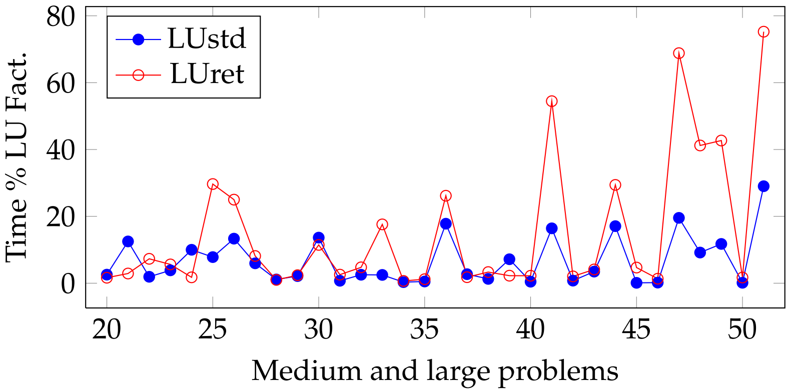


Figure 2: Time % *LU* Fact. of LUstd and LUret
Average LUstd = 7.2%, Average LUret = 10.5%

Content:

Performance of *PCx* with LUstd

To compare the modified *PCx* performance with the new approach, we consider the four variants below

Four approach

1. Lustd with penalization parameter (the new approach).
2. Lustd without penalization parameter.
3. LUret with penalization parameter.
4. Original LUret of Oliveira and Sorensen [14].

Once calculated processing times for the four approaches, will build the performance profile suggested by Dolan and Moré.

Performance of *PCx* with LUstd

To compare the modified *PCx* performance with the new approach, we consider the four variants below

Four approach

1. Lustd with penalization parameter (the new approach).
2. Lustd without penalization parameter.
3. LUret with penalization parameter.
4. Original LUret of Oliveira and Sorensen [14].

Once calculated processing times for the four approaches, will build the performance profile suggested by Dolan and Moré.

Performance of *PCx* with LUstd

To compare the modified *PCx* performance with the new approach, we consider the four variants below

Four approach

1. Lustd with penalization parameter (the new approach).
2. Lustd without penalization parameter.
3. LUret with penalization parameter.
4. Original LUret of Oliveira and Sorensen [14].

Once calculated processing times for the four approaches, will build the performance profile suggested by Dolan and Moré.

Performance of *PCx* with LUstd

To compare the modified *PCx* performance with the new approach, we consider the four variants below

Four approach

1. Lustd with penalization parameter (the new approach).
2. Lustd without penalization parameter.
3. LUret with penalization parameter.
4. Original LUret of Oliveira and Sorensen [14].

Once calculated processing times for the four approaches, will build the performance profile suggested by Dolan and Moré.

Performance of *PCx* with LUstd

To compare the modified *PCx* performance with the new approach, we consider the four variants below

Four approach

1. Lustd with penalization parameter (the new approach).
2. Lustd without penalization parameter.
3. LUret with penalization parameter.
4. Original LUret of Oliveira and Sorensen [14].

Once calculated processing times for the four approaches, will build the performance profile suggested by Dolan and Moré.

Processing time

Problem	LUret	LUret+Pen	LUstd-Pen	LUstd+Pen
pds-70	1350.78	1440.13	1460.42	1431.93
ken18	1181.18	1196.53	1573.41	1438.93
pds-80	1705.75	1860.69	1744.39	1872.91
pds-90	2140.66	2106.1	2143.03	2126.16
pds-100	2896.42	3099.77	3088.34	3153.13
kra30a			6693.08	6698.49
ste36a	14072.58	14563.07	7298.44	6911.57
ste36b			9497.97	8305.62
kra30b			6537.62	9326.45
ste36c			11286.5	9986.89
nug20			12245.43	10651.28
Remaining	2191.73	6060.74	3693.04	3591.50
Total	25539.1	30327.03	67261.67	65494.86
In Common	25473.5	26249.59	19119.04	18726.05

Table 1: Processing time in seconds

Summary of times of processing

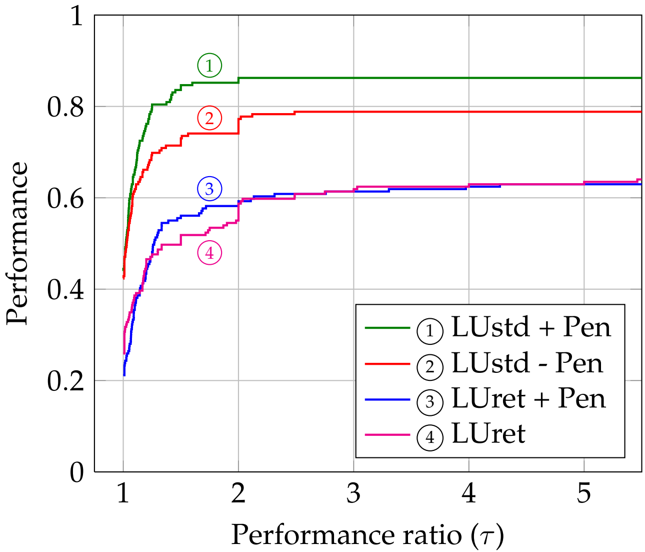
	LUret	LUret+Pen	LUstd-Pen	LUstd+Pen
Problems	189	189	189	189
Solved	121	119	149	163
	64.02%	62.96%	78.84%	86.24%
Unsolved	68[3]	70[3]	40	26
	35.98%	37.04%	21.16%	13.76%
Total (seg)	25539.1	30327.03	67261.67	65494.86
109 Prob.	25473.5	26249.59	19119.04	18726.05
121 Prob.	25539.1			18793.51

[3] 19 problems ended with some processing errors

Table 2: Summary of times of processing
Difference: $25539.1 - 18793.51 = 6745.59 \approx 1h52m$.
LUstd is more efficient than LUret

Content:

Performance Profile of LUstd y LUret



Content

Mixed barrier-penalty algorithm

$$(NLP) \quad \min f(x) \qquad (8)$$
$$\text{s.t. } x \in \Omega_1, x \in \Omega_2, x \in \Omega_3.$$

$$(BP_{\rho,\mu}) \quad \min f(x) + \rho\mathcal{P}(x) + \mu B(x) \qquad (9)$$
$$\text{s.t. } x \in \text{Int}(\Omega_2), x \in \Omega_3.$$

Algorithm

Given $x_0 \in \mathbb{R}^n$, $\rho_0 > 0$, $\mu_0 > 0$, and $k = 0$

1. Test the optimality of x_k for (8) and stop if satisfied
2. Compute $x(\rho_k, \mu_k)$ as global minimizer of (9)
3. Take $x_{k+1} = x(\rho_k, \mu_k)$, $\rho_{k+1} > \rho_k$, $0 < \mu_{k+1} < \mu_k$, $k = k + 1$ and return to step 1.

Mixed barrier-penalty algorithm

$$(NLP) \quad \min f(x) \quad \text{s.t. } x \in \Omega_1, x \in \Omega_2, x \in \Omega_3. \tag{8}$$

$$(BP_{\rho,\mu}) \quad \min f(x) + \rho\mathcal{P}(x) + \mu B(x) \quad \text{s.t. } x \in \text{Int}(\Omega_2), x \in \Omega_3. \tag{9}$$

Algorithm

Given $x_0 \in \mathbb{R}^n$, $\rho_0 > 0$, $\mu_0 > 0$, and $k = 0$

1. Test the optimality of x_k for (8) and stop if satisfied
2. Compute $x(\rho_k, \mu_k)$ as global minimizer of (9)
3. Take $x_{k+1} = x(\rho_k, \mu_k)$, $\rho_{k+1} > \rho_k$, $0 < \mu_{k+1} < \mu_k$, $k = k + 1$ and return to step 1.

Mixed barrier-penalty algorithm

$$(NLP) \quad \min f(x) \quad \text{s.t. } x \in \Omega_1, x \in \Omega_2, x \in \Omega_3. \tag{8}$$

$$(BP_{\rho,\mu}) \quad \min f(x) + \rho\mathcal{P}(x) + \mu B(x) \quad \text{s.t. } x \in \text{Int}(\Omega_2), x \in \Omega_3. \tag{9}$$

Algorithm

Given $x_0 \in \mathbb{R}^n$, $\rho_0 > 0$, $\mu_0 > 0$, and $k = 0$

1. Test the optimality of x_k for (8) and stop if satisfied
2. Compute $x(\rho_k, \mu_k)$ as global minimizer of (9)
3. Take $x_{k+1} = x(\rho_k, \mu_k)$, $\rho_{k+1} > \rho_k$, $0 < \mu_{k+1} < \mu_k$, $k = k + 1$ and return to step 1.

Mixed barrier-penalty algorithm

$$(NLP) \quad \min f(x) \quad \text{s.t. } x \in \Omega_1, x \in \Omega_2, x \in \Omega_3. \tag{8}$$

$$(BP_{\rho,\mu}) \quad \min f(x) + \rho\mathcal{P}(x) + \mu B(x) \quad \text{s.t. } x \in \text{Int}(\Omega_2), x \in \Omega_3. \tag{9}$$

Algorithm

Given $x_0 \in \mathbb{R}^n$, $\rho_0 > 0$, $\mu_0 > 0$, and $k = 0$

1. Test the optimality of x_k for (8) and stop if satisfied
2. Compute $x(\rho_k, \mu_k)$ as global minimizer of (9)
3. Take $x_{k+1} = x(\rho_k, \mu_k)$, $\rho_{k+1} > \rho_k$, $0 < \mu_{k+1} < \mu_k$, $k = k + 1$ and return to step 1.

Convergence Theorem

Global convergence for mixed method

Let $\{x_k\}$ be a sequence of global minimizers of the mixed penalized problem generated by above mixed Algorithm in which $\rho_k \to +\infty$ and $\mu_k \to 0$. Then any limit point of sequence is a global minimizer of the (NLP) problem.

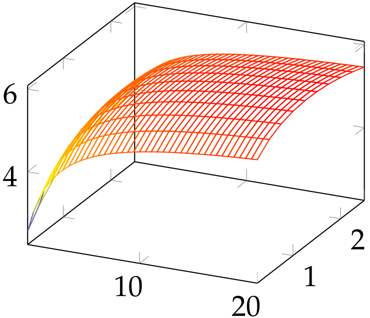


Figure 3: $\theta(\rho, \mu)$, $0 < \mu < 2.5$, $0 < \rho < 20$

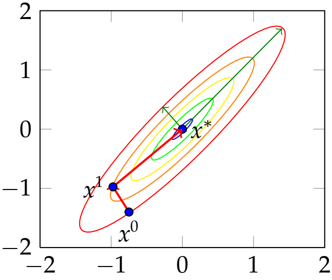
Content

Content:

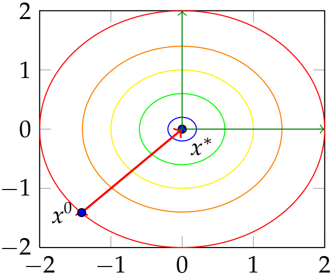
Better Condition number

1. Number of iterations of the conjugated gradients method in second phase: for 51 problems in common

   Average LUstd = 5961.61, Average LUret = 7435.37. ↓ 19.81%



(a) Ill-conditioned



(b) Well conditioned

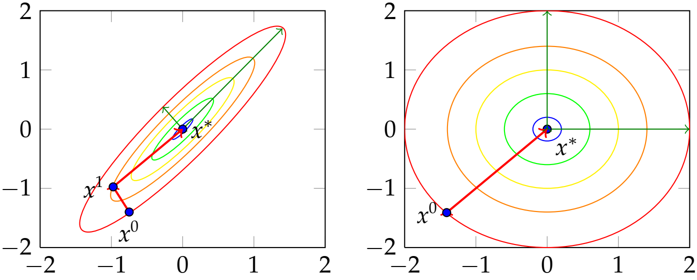
2. Time percentage of the *LU* factorización

   Average LUstd = 7.2%, Average LUret = 10.5%. ↓ 28.57%

Better Condition number

1. Number of iterations of the conjugated gradients method in second phase: for 51 problems in common

   Average LUstd = 5961.61, Average LUret = 7435.37. ↓ 19.81%



(c) Ill-conditioned (d) Well conditioned

2. Time percentage of the *LU* factorización

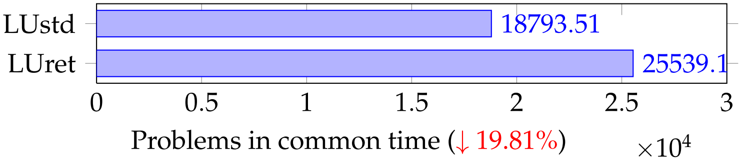
   Average LUstd = 7.2%, Average LUret = 10.5%. ↓ 28.57%

Content:

Efficiency and robustness

1. The more efficient and robust method is one that quickly reaches one higher probability.

$$LUstd : Prob(\tau \leq 2) = 0.86, \quad LUret : Prob(\tau \leq 5.5) = 0.64$$



Thus, LUstd is more efficient and more robust than LUret.

Bibliography I

- G. Al-Jeiroudi, J. Gondzio e J. Hall, *Preconditioning Indefite System in Interior Point Methods for Large Scale Linear Optimization*, Optimization Methods and Software, 2008, volume 23 págs. 345–363.
- Patrick R. Amestoy e Chiara Puglisi, *An unsymmetrized multifrontal LU factorization*, SIAM Journal on Matrix Analysis and Applications, 2002, volume 24, numero 2, pág. 553–569.
- L. Bergamaschi, J. Gondzio e G. Zilli, *Preconditioning Indefite System in Interior Point Methods for Optimization*, Computational Optimization and Application, 2004, volume 28 págs. 149–171, kluwer Academic Publishers – Netherlands.
- Luca Bergamaschi, Jacek Gondzio e Giovanni Zilli, *Preconditioning indefinite systems in interior point methods for optimization*, Computational Optimization and Applications, 2004, volume 28, nº 2 págs. 149–171.
- Frederico Ferreira Campos e John S. Rollet, *Controlled Cholesky factorisation for preconditioning the conjugate gradient method.*, Oxford University Computing Laboratory, Numerical Analysis Group, 1995.
- J. Chai e K. Toh, *Preconditioning and Iterative Solution of Symmetric Indefinite Linear System Arising from Interior Point Methods for Linear Programming*, Computational Optimization and Applications, 2007, volume 36 págs. 221–247.
- Joseph Czyzyk, Sanjay Mehrotra, Michael Wagner e Stephen J Wright, *PCx user guide (Version 1.1)*, Optimization Technology Center, Northwestern University, 1997.
- Timothy A. Davis e Iain S. Duff, *An unsymmetric-pattern multifrontal method for sparse LU factorization*, SIAM Journal on Matrix Analysis and Applications, 1997, volume 18, nº 1 págs. 140–158.
- J. Gondzio, *Interior Point Method 25 year later*, European Journal of Operational Research, 2012, Elsevier. volume 218, nº 3 págs. 587–601.

Bibliography II

David G. Luenberger, *Linear and Nonlinear Programming*, 2ª ed., Springer, Stanford University USA, 2005.

Sanjay Mehrotra, *Quadratic convergence in a primal-dual method*, Mathematics of Operations Research, 1993, volume 18, nº 3 págs. 741–751.

Carl Meyer, *Matrix analysis and applied linear algebra book and solutions manual*, volume 2, Society for Industrial and Applied Mathematics, 2000.

Ašić, Miroslav D and Kovačević-Vujčić, Vera V, *Ill-conditionedness and interior-point methods*, Publikacija Elektrotehničkog fakulteta-serija: matematika, 2000, No. 11, pp. 53–58.

A. R. L. Oliveira e D. C. Sorensen, *A New Class of Preconditioners for Large-Scale Linear Systems from Interior Point Methods for Linear Programming*, Linear Álgebra and its Applications, 2005, volume 394 págs. 1–24, England http://www.ime.unicamp.br/∼aurelio/artigos/split.pdf.

J.A. Tomlin e M. Sauders, *Solving regularized linear Programs using barrier method and KKT system.*, Rel. Téc. SOL 96-4, Standfor University, Standfor CA 94305, 1996, USA.

Velazco, MI and Oliveira, Aurelio RL and Campos, FF, *A note on hybrid preconditioners for large-scale normal equations arising from interior-point methods*, Journal Optimization Methods & Software, volume 25, nº2, págs 321–332, 2010, Taylor & Francis.

Y. Zhang e R.A. Tapia, *Superlinear and Quadratic Convergence of Primal-Dual Interior-Point Methods for Linear Programming Revisited*, Journal of Optimization Theory and Applications, 1992, volume 73, nº 2 págs. 229–242.

Acknowledgement

THANK YOU

For your Attention

Porfirio Suñagua Salgado

UNICAMP