

Efecto de Pre-Ordenación en el Precondicionador Separador del Método de Puntos Interiores para resolver problemas de programación lineal

Porfirio Suñagua Salgado¹

¹ *Instituto de Investigación Matemática, Universidad Mayor de San Andrés, Bolivia*

Resumen

Una clase de métodos para resolver problemas de programación lineal de dimensiones grandes, es el método de puntos interiores de predictor-corrector de Mehrotra, en la cual, se debe resolver sistemas lineales simétricos mediante algoritmos iterativos donde la matriz de coeficientes del sistema es altamente mal condicionada cuando se aproxima al punto óptimo. Para mantener la eficiencia de los métodos, una de las estrategias es mejorar el condicionamiento de la matriz mencionada aplicando un precondicionador apropiado. De manera que, en las primeras iteraciones es suficiente aplicar un precondicionador basada en una descomposición incompleta de Cholesky, en tanto que, que en las iteraciones finales, cuando la matriz de coeficientes queda muy mal condicionada, se cambia al precondicionador separador basada en una base de columnas de la matriz de restricciones. Elegir una clase de descomposición incompleta de Cholesky, un criterio para hallar una base de columnas o un criterio de cambio de precondicionador generan diversos abordajes para resolver de manera eficiente y robusta problemas de programación lineal (PL) cuando la dimensión de la variable de decisión sea grande.

En el presente trabajo se quiere mostrar que si los diferentes criterios de pre-ordenación de columnas de la matriz de restricciones del problema de programación lineal, necesarias para ayudar a identificar una base de columnas mediante el proceso factorización rectangular LU son relevantes o no en el desempeño del precondicionador separador utilizada en el método de puntos interiores para resolver problemas de programación lineal especialmente de dimensiones de la variable de decisión grandes. Los criterios de pre-ordenación a abordar serán los basados en las normas euclidianas 1, 2 e infinito y otros dos criterios de ordenación por dispersidad de columnas dada por el número de entradas no nulas, y otro criterio de ordenación será que, en lo posible, las primeras columnas se ordenen en una forma de triangular superior. La comparación de los diferentes abordajes será realizada mediante la curva de desempeño de Dolan y Moré.

Palabras Clave: Programación lineal, método de puntos interiores, preconditionador separador.

1. INTRODUCCIÓN

Un problema de programación lineal (PL) es un problema matemático de maximización o minimización de una función lineal $z = c^T x$ sujeta a restricciones lineales dada por $Ax = b$ con $x \geq 0$, donde A es una matriz $m \times n$ con $m < n$, llamada matriz de restricciones, y los otros vectores tienen dimensiones apropiadas. La familia de métodos de puntos interiores para resolver problema de programación lineal (PL) se remonta a los años 1980, donde Karmakar [10] planteó un nuevo paradigma en encarar y resolver problemas PL, como una alternativa al método Simplex cuando la matriz de restricciones sea dispersa. Aunque se pueden aplicar ambos métodos para resolver cualquier problema de programación lineal, en las últimas décadas se han hecho muchas contribuciones en la eficiencia de los métodos de puntos interiores basados en el método predictor–corrector de Merhotra con múltiples correcciones propuesta por Gondzio [8]. En 2012, a tiempo de celebrar los 25 años del Método de Puntos Interiores, Gondzio [9], fue invitado para realizar una descripción y un análisis completo sobre el desarrollo de esta familia de métodos.

En el proceso de implementación del método de puntos interiores, como se muestra en Wright [16] y Oliveira y Sorensen [13], se tienen que resolver sistemas lineales donde la matriz de coeficientes del sistema, por lo general simétricos, resultan muy mal condicionados especialmente cuando se está cerca del punto óptimo. Si S denota esa matriz simétrica, un preconditionador de S , como se presenta en Golub y Van Loan [7], es una matriz no singular P , tal que, el número de condición de la matriz preconditionada $PSPT$ sea menor que de S . Uno de estos preconditionadores simples para una matriz definida positiva S puede ser $P = \text{diag}(S)^{-1}$. Sin embargo, para los métodos de puntos interiores, este preconditionador solo funciona en las iteraciones iniciales. Ciertamente, el mejor preconditionador sería $P = L^{-1}$, donde L es el factor de Cholesky que satisface $S = LL^T$. Más aún, para una dimensión de S grande, la descomposición en Cholesky puede resultar computacionalmente caro en términos de tiempo de cómputo como en tamaño de memoria ocupada en todo el proceso de factorización. A fin de economizar el recurso computacional, uno de los preconditionadores consideradas por los investigadores son basados en la factorización incompleta de Cholesky, como se presenta en Lin y Moré [11] y posiblemente más adecuada para el método de puntos interiores en Campos [2]. Las mismas permiten resolver sistemas lineales preconditionados resultantes del método de puntos interiores ya sea por métodos directos o mediante métodos numéricos como aquel método clásico de gradientes conjugados.

En método de puntos interiores, hay dos tipos de sistemas simétricos a resolver, uno de ellos es el *sistema normal de ecuaciones* y el otro es el llamado *sistema aumentado*. Para el primer caso se aplica un preconditionador basado en una factorización incompleta de Cholesky y para el segundo caso, consideramos un preconditionador llamado *separador* propuesta por Oliveira y Sorensen [13], que se construye a partir de una base de columnas de la matriz de restricciones, las mismas se identifican mediante un proceso

de factorización rectangular LU. Estrategias adicionales de mejoramiento de es preconditionador se encuentran en Bocanegra [1] y Suñagua y Oliveira [15]. Además, encontrar esa base de columnas de modo que el preconditionador separador sea eficiente, no es una tarea sencilla. A fin abaratar el costo computacional, se sugiere reordenar las columnas de la matriz de manera que en la búsqueda secuencial de columnas linealmente independientes las primera columnas cumplan ciertos criterios naturales de independencia lineal y el resto de las columnas potencialmente dependientes sean enviadas a las posiciones finales.

En este trabajo se quiere mostrar que si los diferentes criterios de pre-ordenación de columnas de la matriz de restricciones del problema de programación lineal, necesarias para ayudar a identificar una base de columnas mediante el proceso factorización rectangular LU son relevantes o no en el desempeño del preconditionador separador utilizada en el método de puntos interiores para resolver problemas de programación lineal. Los criterios de pre-ordenación a abordar serán los basados en las normas euclidianas 1, 2 e infinito y otros dos criterios de ordenación por dispersidad de columnas dada por el número de entradas no nulas, y otro criterio de ordenación será que, en lo posible, las primeras columnas se ordenen en una forma de triangular superior. La comparación de los diferentes abordajes será realizada mediante la curva de desempeño de Dolan y Moré.

2. PROBLEMAS DE PROGRAMACIÓN LINEAL

Consideramos el problema primal (P) de programación lineal en su forma estándar y su correspondiente problema dual¹ (D)

$$\begin{array}{ll}
 (P) & \text{mín } c^T x \\
 & \text{s. a. } Ax = b \\
 & \quad x \geq 0 \\
 (D) & \text{máx } b^T y \\
 & \text{s. a. } A^T y + z = c \\
 & \quad y \text{ libre, } z \geq 0.
 \end{array}$$

Estos dos problemas están intrínsecamente relacionados por el teorema fundamental de programación lineal, como por el teorema de dualidad, y otros resultados que relacionan variables primales y duales. Uno de los resultados en el punto óptimo es el llamado teorema de complementariedad que presentamos a continuación, la cual permite escribir mas adelante las condiciones de optimalidad de un problema lineal.

Teorema 2.1 (Complementariedad). *Sean x y (y, z) soluciones factibles de los problemas (P) y (D) respectivamente. Una condición necesaria y suficiente para que ambas soluciones sean óptimas es que, si $x_j > 0$, entonces $z_j = 0$, si $z_j > 0$, entonces $x_j = 0$, la cual se puede escribir como $x_j \cdot z_j = 0$, para todo $j = 1, 2, 3, \dots, n$.*

Demostración. La condición $x_j \cdot z_j = 0, \forall j$ se puede escribir también como $z^T x = 0$. Como $z = c - A^T y$, entonces, $z^T x = (c - A^T y)^T x = c^T x - y^T Ax = c^T x - y^T b = c^T x - b^T y = 0 \Leftrightarrow c^T x = b^T y$ \square

¹En general, el problema dual en PL se define como el problema $\max_y \theta(y)$, donde $\theta(y) = \inf_{x \geq 0} \mathcal{L}(x, y)$, y $\mathcal{L}(x, y) = c^T x + y^T (b - Ax)$ es el Lagrangiano del problema primal.

Cualquier solución del problema primal y dual obviamente debe satisfacer las restricciones de sus correspondientes problemas, luego desde la condición de complementariedad, tenemos las condiciones de optimalidad, llamada también, condiciones de Karush-Kuhn-Tucker (KKT).

Así, las condiciones de optimalidad para los problemas (P) y (D) están dados por

$$(1) \quad \begin{array}{ll} Ax = b, & x \geq 0 & \text{Primal factible} \\ A^T y + z = c, & z \geq 0 & \text{Dual factible} \\ XZe = 0 & & \text{Complementariedad} \end{array}$$

donde e es el vector de unos, X es la matriz diagonal de x y Z es la matriz diagonal de z .

2.1. Métodos de resolución. En la Figura 1, se presenta el esquema donde un problema PL puede ser resuelto por el método clásico Simplex o bien por los Métodos de Puntos Interiores (MPI). En cada iteración de ambos métodos se necesita resolver sistemas lineales ya sea por métodos directos o métodos numéricos según las características y dimensión del problema. Cuando la dimensión del problema es grande, los métodos de resolución de sistemas lineales no funcionan bien si la matriz de coeficientes del sistema está mal condicionada.

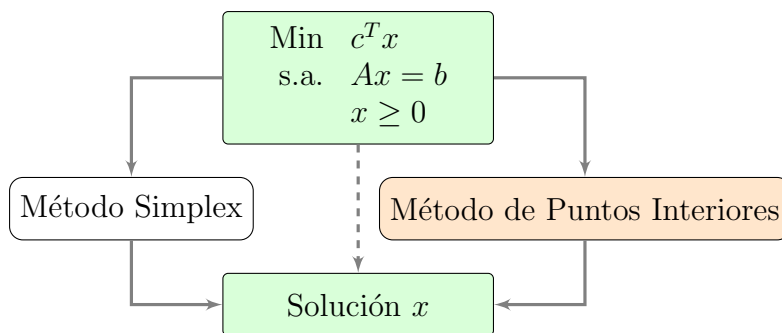


FIGURA 1. Métodos de solución.

Teóricamente, tanto el Método Simplex, como el Método de Puntos Interiores tienen fórmulas explícitas. Sin embargo, el mayor costo computacional de estos métodos consiste en la resolución de sistemas lineales. Para problemas PL de dimensiones grandes, es decir, para m y n grandes, digamos $m > 50000$ y $n > 100000$, se necesita sofisticar los métodos con algoritmos iterativos. Aunque esas dimensiones pueden estar relacionados con la tecnología computacional.

En la Figura 2, de acuerdo al teorema fundamental de programación lineal, se muestra que a partir de una solución inicial factible, el método Simplex recorre los vértices vecinos hasta encontrar el óptimo o detenerse por algún criterio de parada. Mientras que a partir de un punto inicial interior, el método de puntos interiores, mediante direcciones de tipo Newton, recorre el interior del conjunto restricciones de desigualdad hasta converger al punto óptimo o detenerse por algún criterio de parada.

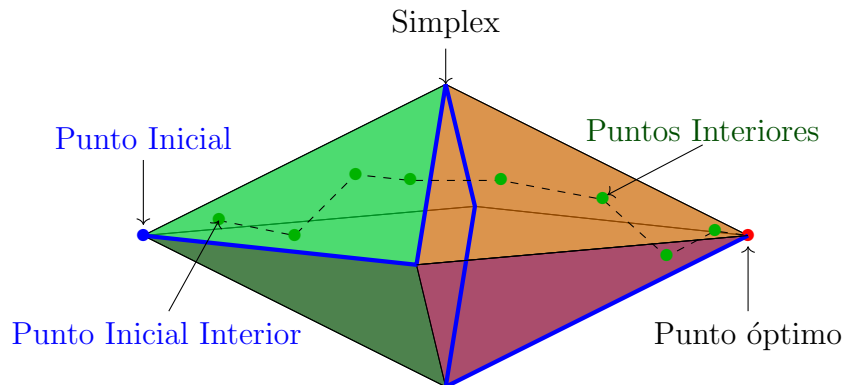


FIGURA 2. Simplex versus MPI.

2.2. Método de Newton. Dada una función vectorial multivariada, $\mathbf{F} : D_F \subset \mathbb{R}^n \rightarrow \mathbb{R}^n$, se quiere resolver la ecuación $\mathbf{F}(\mathbf{x}) = \mathbf{0}$. Por el Teorema de Taylor $\mathbf{F}(\mathbf{x}) \approx \mathbf{F}(\mathbf{x}^k) + \mathbf{J}(\mathbf{x}^k)(\mathbf{x} - \mathbf{x}^k)$, donde $\mathbf{J}(\mathbf{x}) = \nabla F(x)$ es la matriz Jacobiana de F en x . Una vez determinada la dirección de Newton \mathbf{d}^k , la ecuación de iteración, está dada por

$$\mathbf{x}^{k+1} = \mathbf{x}^k + \mathbf{d}^k$$

donde, $\mathbf{d}^k = -\mathbf{J}(\mathbf{x}^k)^{-1}\mathbf{F}(\mathbf{x}^k)$. En la práctica es mejor resolver el sistema lineal $\mathbf{J}(\mathbf{x}^k)\mathbf{d}^k = -\mathbf{F}(\mathbf{x}^k)$.

Ejemplo 2.1 (Newton univariado). Consideramos la ecuación $F(x) = \frac{1}{4}(x^2 - 1) = 0$. En este caso, la dirección de Newton se determina de

$$\frac{1}{2}x^k d^k = -\frac{1}{4}((x^k)^2 - 1) \Rightarrow d^k = -\frac{1}{2} \left(x^k - \frac{1}{x^k} \right).$$

Luego, las iteraciones de Newton, resulta

$$x^{k+1} = x^k + d^k = \frac{1}{2} \left(x^k + \frac{1}{x^k} \right).$$

La representación gráfica de las iteraciones se muestra en la Figura 3.

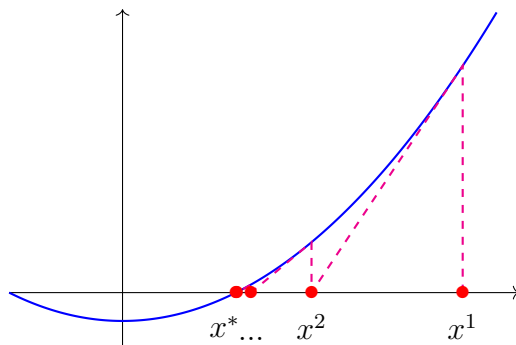


FIGURA 3. Newton univariado

Ejemplo 2.2. Consideremos el problema de minimización de la función de Rosenbrock

$$f(x, y) = (1 - x)^2 + 100(y - x^2)^2,$$

luego, aplicamos el método de Newton a la condición necesaria del óptimo de primer orden

$$\mathbf{F}(x, y) = \nabla f(x, y) = 0.$$

Para deducir las ecuaciones de iteración de Newton, la matriz Jacobiana, resulta

$$\mathbf{J}(x, y) = \nabla \mathbf{F}(x, y) = \nabla^2 f(x, y),$$

luego, la dirección de Newton y las iteraciones correspondientes para hallar las raíces de la condición de optimalidad, resultan

$$\begin{aligned} \mathbf{J}(x^k, y^k) \mathbf{d}^k &= -\mathbf{F}(x^k, y^k), \\ x^{k+1} &= x^k + d_1^k, \quad y^{k+1} = y^k + d_2^k, \end{aligned}$$

donde, d_1^k y d_2^k son las coordenadas de la dirección \mathbf{d}^k . En la Figura 4 se muestran las iteraciones para el caso bidimensional. Además, se conoce que el método de Newton tiene convergencia cuadrática cerca de una raíz simple. Para raíces múltiples, la convergencia puede ser solo superlineal, Mehrotra [12].

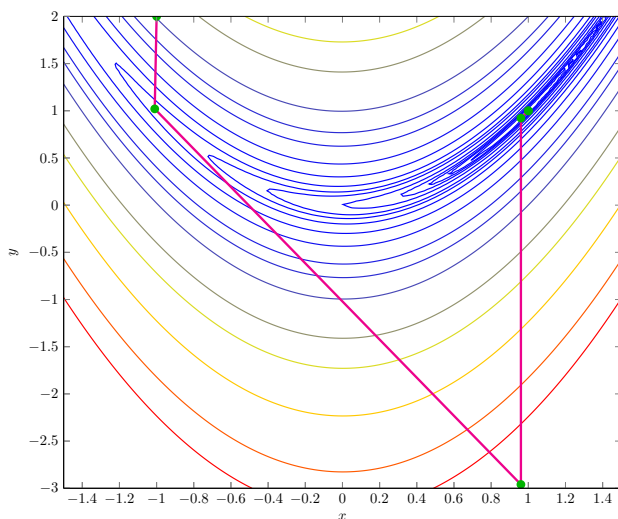


FIGURA 4. Punto crítico de la función de Rosenbrock.

2.3. Primal–Dual seguidor de camino. Para el método Primal–Dual Seguidor de Camino, asumiendo $x > 0$, consideramos la función Lagrangeana con barrera logarítmica de parámetro $\mu > 0$

$$\mathcal{L}(x, y) = c^T x + y^T (b - Ax) - \mu \sum_{i=1}^n \ln x_i.$$

Luego, las condiciones necesarias de primer orden serán

$$\begin{aligned}\frac{\partial \mathcal{L}}{\partial x} &= c - A^T y - \mu X^{-1} e = 0, \\ \frac{\partial \mathcal{L}}{\partial y} &= b - Ax = 0,\end{aligned}$$

donde, X es la matriz diagonal de x , luego llamando, $z = Ze = \mu X^{-1} e$, entonces, $XZe = \mu e$, donde e es el vector de unos, tenemos las condiciones de optimalidad perturbada del problema dado

$$(2) \quad \begin{aligned}Ax &= b, \\ A^T y + z &= c, \\ XZe &= \mu e.\end{aligned}$$

Para aplicar el método de Newton a (2), hallamos la matriz Jacobiana y escribimos el sistema lineal para las direcciones de Newton

$$\begin{pmatrix} A & 0 & 0 \\ 0 & A^T & I \\ Z & 0 & X \end{pmatrix} \begin{pmatrix} dx \\ dy \\ dz \end{pmatrix} = \begin{pmatrix} r_p \\ r_d \\ r_c \end{pmatrix} \quad \begin{aligned} r_p &= b - Ax \\ r_d &= c - A^T y - z \\ r_c &= -XZe + \mu e.\end{aligned}$$

Para resolver para las direcciones de Newton, escribimos el sistema de ecuaciones por bloques

$$(3) \quad Adx = r_p$$

$$(4) \quad A^T dy + dz = r_d$$

$$(5) \quad Zdx + Xdz = r_c.$$

De (5), tenemos, $dz = X^{-1}(r_c - Zdx)$. Para $D^{-1} = X^{-1}Z$, sustituyendo en (4), obtenemos

$$\begin{aligned}A^T dy + X^{-1}r_c - X^{-1}Zdx &= r_d &\Rightarrow & (X^{-1}Z)dx = A^T dy - r_d + X^{-1}r_c \\ & &\Rightarrow & dx = D(A^T dy - r_d + X^{-1}r_c).\end{aligned}$$

Ahora sustituyendo en el bloque de ecuaciones (3), tenemos

$$\begin{aligned}AD(A^T dy - r_d + X^{-1}r_c) &= r_p \\ ADA^T dy &= r_p + AD(r_d - X^{-1}r_c) \\ (ADA^T)dy &= r_p + AD(r_d - X^{-1}r_c).\end{aligned}$$

Así, el sistema aumentado y el sistema normal de ecuaciones para las direcciones dx y dy , para $D = Z^{-1}X$, resultan

$$\left. \begin{aligned} -D^{-1}dx + A^T dy &= r_d - X^{-1}r_c \\ Adx &= r_p \end{aligned} \right\}, \quad ADA^T dy = r_p + AD(r_d - X^{-1}r_c).$$

El sistema aumentado es mal condicionado, debido a que la matriz D es mal escalada a razón de que cerca del óptimo, a fin de satisfacer la condición de complementariedad, el producto de la diagonal de D , dada por $x_j^{-1}z_j$, produce valores numéricamente muy

grandes o muy pequeños. En tanto que, el sistema normal, todavía es mas mal condicionada, pues las entradas mal escaladas de D , empeoran en número de condición del producto ADA^T .

A fin de analizar las ventajas o bondades del sistema de ecuaciones normales (6) o del sistema aumentado (7), en el Cuadro 1 muestra una comparativa de resolver uno de los sistemas en base a algunas características de cada uno de ellos.

$$(6) \quad ADA^T dy = r,$$

$$(7) \quad \begin{pmatrix} -D^{-1} & A^T \\ A & 0 \end{pmatrix} \begin{pmatrix} dx \\ dy \end{pmatrix} = \begin{pmatrix} r_1 \\ r_p \end{pmatrix},$$

donde, $r = r_p + ADr_1$ y $r_1 = r_d - X^{-1}r_c$.

CUADRO 1. Propiedades de los sistemas lineales.

Ecuaciones Normales	Sistema Aumentado
Características de las matrices	
Definida Positiva	Indefinida
Simétrica	Simétrica
dimensión m	dimensión $m + n$
Perdida de la estructura dispersa	Utiliza matrices originales
Sólo D cambia en la matriz	Sólo D cambia en la matriz
Muy mal-condicionado	mal-condicionado
Métodos directos	
Cholesky	Bunch-Parlett
Métodos numéricos	
Gradientes Conjugados (Prec.)	Métodos iterativos

Cabe señalar que algunos métodos iterativos pueden ser Gauss-Seidel, MINRES (minimum residual method), GMRES (generalized minimum residual method), cuyos detalles los podemos encontrar en Paiges y Saunders [14] y ampliamente desarrolladas en Golub y Van Loan [7].

3. PRECONDICIONADOR

Definición 3.1. Dada una matriz \mathbf{A} , una matriz P no singular se llama preconditionador si la matriz $P\mathbf{A}$ tiene menor número de condición que de \mathbf{A} . Si \mathbf{A} es simétrica, para mantener la simetría, la matriz preconditionada será $P\mathbf{A}P^T$.

Uno de las aplicaciones de un preconditionador es en la resolución de sistemas lineales preconditionadas. Es decir, para resolver el sistema $\mathbf{A}x = r$, preconditionamos y resolvemos.

$$(P\mathbf{A}P^T)P^{-T}x = Pr \quad \Leftrightarrow \quad \tilde{A}\bar{x} = \tilde{r},$$

donde, $\tilde{A} = P\mathbf{A}P^T$, $\tilde{r} = Pr$ y $\bar{x} = P^{-T}x$, luego la solución del sistema original será $x = P^T\bar{x}$. Así, un preconditionador P debería tener la propiedad deseable, $P\mathbf{A}P^T \approx I$, lo que es equivalente a que sus autovalores $\lambda_{P\mathbf{A}P^T} \approx 1$.

Observación 3.1. El mejor preconditionador, pero caro de ser calculado, en términos de tiempo de ejecución y cantidad de memoria requerida, para una matriz simétrica definida positiva $\mathbf{A} > 0$, es basado en el factor de descomposición de Cholesky $\mathbf{A} = LL^T$. En este caso se toma, $P = L^{-1}$ luego $P\mathbf{A}P^T = L^{-1}LL^TL^{-T} = I$.

Observación 3.2 (Buen preconditionador). Un buen preconditionador, no puede ser muy caro de ser calculado, y la matriz preconditionada debe ser razonablemente bien condicionada. Un equilibrio de estos dos criterios define un buen preconditionador.

En la resolución de sistemas preconditionados de dimensiones grandes

$$\tilde{A}\bar{x} = \tilde{r}$$

generalmente para encontrar la solución se utilizan métodos iterativos basados en productos de matriz por vector $\tilde{A}v = (P\mathbf{A}P^T)v = P(\mathbf{A}(P^T v))$, sin necesidad de obtener explícitamente el producto $P\mathbf{A}P^T$.

3.1. Precondicionador Separador. Dado el sistema aumentado

$$\begin{pmatrix} -D^{-1} & A^T \\ A & 0 \end{pmatrix} \begin{pmatrix} dx \\ dy \end{pmatrix} = \begin{pmatrix} r_1 \\ r_p \end{pmatrix}$$

consideramos una partición $A = [B, N]$ y $D = \text{diag}(D_B, D_N)$, donde B es una base de columnas de A . Un preconditionador eficiente para el sistema aumentado, es el llamado *precondicionador separador* propuesto por Oliveira y Sorensen [13]

$$(8) \quad M^{-1} = \begin{pmatrix} D_B^{\frac{1}{2}} & 0 & D_B^{-\frac{1}{2}}B^{-1} \\ 0 & D_N^{\frac{1}{2}} & 0 \\ D_B^{\frac{1}{2}} & 0 & 0 \end{pmatrix}$$

tal que B tenga la propiedad deseable

$$D_B^{-\frac{1}{2}}B^{-1}ADA^TB^{-T}D_B^{-\frac{1}{2}} = I + WW^T \approx I$$

o bien de manera equivalente, si $W = D_B^{-\frac{1}{2}}B^{-1}ND_N^{\frac{1}{2}} \approx 0$.

3.2. Abordaje híbrida. En las iteraciones iniciales del método de puntos interiores, llamado **Fase I**; cuando la matriz ADA^T no es mal condicionada se usa el preconditionador basado en una descomposición incompleta de Cholesky, llamada la factorización controlada de Cholesky, propuesta por Campos [2], con el cual se resuelve el sistema normal de ecuaciones.

$$(9) \quad P^{-1}(ADA^T)P^{-T}P^T dy = P^{-1}r.$$

En la **Fase II** cuando la matriz de coeficientes ADA^T queda muy mal condicionada se cambia para resolver el sistema aumentado usando el preconditionador separador M^{-1} dada en (8). Así como está detallada en Oliveira y Sorensen [13], esto equivalente a resolver el sistema normal de ecuaciones con el preconditionador $P^{-1} = D_B^{-\frac{1}{2}}B^{-1}$.

$$(10) \quad D_B^{-\frac{1}{2}}B^{-1}(ADA^T)B^{-T}D_B^{-\frac{1}{2}}d\tilde{y} = D_B^{-\frac{1}{2}}B^{-1}(r_p + ADr_1)$$

luego, $dy = B^{-T} D_B^{-\frac{1}{2}} d\tilde{y}$, $dx = D(A^T dy - r_1)$ y $dz = X^{-1}(r_c - Zdx)$.

3.3. Base del preconditionador para el abordaje LUret. Dada una matriz $A : m \times n$ con $m < n$, se necesita identificar m columnas linealmente independientes B de los n disponibles, tal que, $A = [B, N]$ y $W = D_B^{-\frac{1}{2}} B^{-1} N D_N^{\frac{1}{2}} \approx 0$, donde D está dado en el método de puntos interiores. En la Figura 5 las líneas rojas ilustran a esas m columnas que se quiere que sean linealmente independientes.

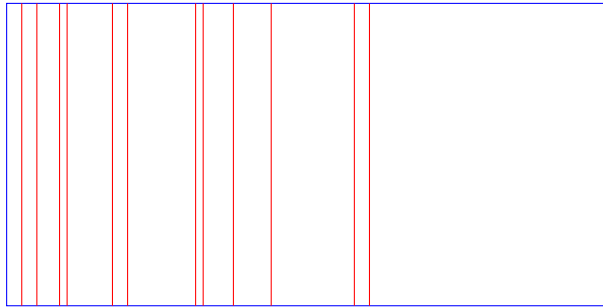


FIGURA 5. Selección de m columnas base de A .

En principio, a fin de facilitar la selección de una base, a priori se inicia con un reordenamiento inicial de columnas basadas en norma-2, norma-1, o bien otro criterio de independencia lineal entre las columnas de la matriz $AD^{1/2}$. En Oliveira y Sorensen [13] realizan un estudio de potenciales columnas linealmente independientes y dependientes desde un grafo asociado a la matriz de restricciones, por el cual las columnas potencialmente dependientes se intercambia a las posiciones finales de manera que tengan pocas chances de ser elegidos durante el proceso de factorización LU aplicada a la matriz rectangular A , por esta razón, este proceso lo llamaron LUret.

Las diferentes estrategias de selección de columnas durante la factorización LU generan diferentes abordajes. En la literatura, como en Suñagua y Oliveira [15], también en Gondzio [9] se encuentran abordajes de mejoras de número de condición de la matriz de coeficientes de sistemas lineales a través de parámetros de penalización externa o de regularización.

4. ABORDAJES DE REORDENAMIENTO Y EXPERIMENTOS NUMÉRICOS

En el presente trabajo, se ha propuesto mostrar el efecto de las preordenaciones de columnas de $AD^{1/2}$ por la norma-1, norma-2, norma infinita de \mathbb{R}^m antes de aplicar el proceso Luret. Además, proponemos otros dos criterios heurísticos de independencia lineal basada el primero en la dispersidad de columnas, es decir, por el número de entradas no nulas en cada columna de la misma matriz anterior, de manera que, las columnas mas dispersas sean los primeros candidatos para ser seleccionados por LUret en la base y las columnas densas tengan menos chance de ser seleccionado, pues ellos tendrían mayor posibilidad de ser linealmente dependiente con otras columnas. Otro criterio está basado en que una matriz triangular superior tiene a sus columnas linealmente independientes. Así, preordenamos las columnas de manera que se tenga una forma triangular superior.

Para conseguir este propósito, identificamos el índice de la fila cuya entrada es el último no nulo en la columna, de manera que, los de índices menores ocuparán los primeros lugares.

Para comparar los criterios propuestos, se ha incorporado los cinco criterios de preordenación al programa de código abierto PCx, originalmente implementado en lenguaje C por Czyzyk et al [5], y posteriormente modificado por Oliveira y Sorensen [13] para incorporar el proceso de LUret y Bocanegra [1] que mejora la aplicación híbrida de preconditionadores entre una Cholesky incompleta y el preconditionador separador. Para este efecto, se han considerado 189 problemas de programación lineal tomados de los repositorios NETLIB, Kennington y QAP. En el Cuadro 2 tenemos algunos problemas selectos agrupados en pequeños, medianos y grandes junto a sus características como en número de variables (columnas), el número de restricciones (filas), el número de entradas no nulas de la matriz de restricciones (nnz) y el valor óptimo de la función objetivo calculado con una de las estrategias. La clasificación de problemas en esos grupos no es un asunto bien definido, sin embargo, se pueden tomar como indicadores a filas, columnas, nnz y sobre todo la estructura de dispersidad de A y también de ADA^T , las cuales se reflejan en el tiempo total de resolución del problema. En el Cuadro 3, presentamos el tiempo de procesamiento total en segundos para los problemas seleccionados donde el tiempo finito indica que el proceso terminó exitosamente, caso contrario, marcamos con un * para indicar que ese tiempo es desconocido. En las cinco columnas correspondiente a los cinco abordajes, se puede visualizar que los tiempos de procesamiento no difieren significativamente de uno del otro para el mismo problema. Esta situación sugiere que ninguno de los abordajes se destaca considerablemente. Finalmente, indicar que el equipo computacional utilizado para el procesamiento de los problemas fue Dell XPS L502x Intel Core i7-2670QM 2.20 GHz con 8GB de RAM y arquitectura x64 con sistema operativo Debian de la familia Linux.

Cuadro 2: Problemas selectos de cada grupo

Problema	filas	columnas	nnz	objetivo
Problemas pequeños				
afiro	27	32	102	-464.75314
kb2	43	41	313	-1749.90013
scsd8	397	2750	8584	905
grow22	440	946	8252	-160834336.5
ken-07	2426	3602	5847	-679520443.3
nug07	602	931	3318	148
maros	846	1443	6634	-58063.74335
qap8	912	1632	5936	203.5
25fv47	821	1571	10538	5501.84589
osa-07	1118	23949	89316	535722.5173
Problemas medianos				
maros-r7	3136	9408	100486	1497185.167
continua en la siguiente página				

Cuadro 2 – continua de la página anterior

Problema	filas	columnas	nnz	objetivo
ken11	14694	21349	38157	-6972382262
pds-10	16558	48763	103725	26727094927
BL	6325	8018	33963	22627325.12
nug12	3192	8856	33528	522.89435
qap12	3192	8856	33528	522.89435
scr20	5079	12180	61780	-383154.2237
stocfor3	16675	15695	62960	-39976.78394
ken13	28632	42659	82191	-10257394789
pds-30	49944	154998	333260	21385445780
Problemas grandes				
nug15	6330	22275	85470	1040.99404
qap15	6330	22275	85470	1040.99404
rou20	7359	33840	152980	-1180918.292
pds-70	114944	382311	822526	12241162846
ken18	105127	154699	297886	-52217025252
pds-80	129181	426278	916852	11469077420
pds-90	142823	466671	1002902	11087561659
pds-100	156243	505360	1060567	10928230039
ste36a	27683	109656	512640	-42417.74742
ste36b	27683	109656	512640	-240276.9708

Cuadro 3: Tiempos de procesamiento de los problemas selectos en segundos con PCx para cada una de las estrategias.

Problema	Norma-1	Norma-2	Norma- ∞	Dispersidad	Triangular
Problemas pequeños					
afiro	0.001	0.001	0.001	0.001	0.001
kb2	0.001	0.01	0.01	0.01	0.001
scsd8	0.06	0.06	0.06	0.06	0.06
grow22	0.1	0.09	0.09	0.1	0.09
ken-07	0.1	0.1	0.11	0.11	0.1
nug07	0.26	0.25	0.24	0.27	*
maros	1.38	1.64	2.46	0.92	1.29
nug08	0.59	0.65	0.62	0.59	0.64
25fv47	1.33	3.12	1.74	1.27	3.18
osa-07	0.88	0.83	0.83	0.83	0.83
Problemas medianos					
maros-r7	10.41	10.37	10.32	10.44	10.39
ken11	11.77	11.85	12.08	11.8	11.78

continua en la siguiente página

Cuadro 3 – continua de la página anterior

Problema	Norma-1	Norma-2	Norma- ∞	Dispersidad	Triangular
pds-10	18.32	18.56	18.83	18.91	18.55
BL	20.43	21.02	26.08	21.74	23.73
nug12	86.36	71.43	69.37	85.78	115.99
qap12	87.67	88.73	101.59	88.23	99.67
scr20	63.13	66.99	97.45	72.15	97.78
stocfor3	98.92	99.93	107.86	98.32	96.66
ken13	103.61	102.79	105.05	106.48	104.65
pds-30	231.57	233.07	232.47	232.05	239.97
Problemas grandes					
nug15	*	1271.89	1334.9	*	*
qap15	1076.52	1555.89	*	1069.85	1463.65
rou20	478.32	864.55	933.23	477.72	881.71
pds-70	1261.15	1268.72	1278.05	1270.07	1268.33
ken18	1076.87	1091.22	1144.11	1088.68	1141.44
pds-80	1545.21	1553.13	1564.66	1555.46	1547.07
pds-90	1946.55	1968.04	1980.2	1957.78	1977.66
pds-100	2676.54	2811.19	2711.74	2686.52	2686.42
ste36a	6646.81	10936.42	17107.3	6669.83	*
ste36b	*	12392.43	*	*	*

4.1. Curva de desempeño. Para comparar los cinco abordajes, consideramos los tiempos totales de computo para todos los problemas considerados, sin embargo para una efectiva comparación, utilizamos solo los problemas que convergen por lo menos con uno de los abordajes, en concreto, nos quedó 131 problemas cuyos conteo se muestra en el Cuadro 4.

CUADRO 4. Número de problemas convergentes con los diferentes criterios

norma-1	norma-2	norma infinita	dispersidad	triangular superior
125	125	123	125	123

Para obtener la curva de desempeño consideramos un conjunto \mathcal{P} de n_p problemas y n_s métodos o abordajes. Definimos

$t_{p,s}$ = el tiempo de procesamiento del problema p por el método s .

Para los problemas que no convergen, asignamos a $t_{p,s}$ un valor r_M que sea mayor todos los tiempos de computo calculado para todos los problemas resueltos. Teóricamente, $t_{p,s}$ debería tener el valor infinito para esos problemas, pero será suficiente tomar

$$r_M = 2 \max\{t_{p,s} < +\infty : 1 \leq p \leq n_p, 1 \leq s \leq n_s\}^2.$$

²http://www.mcs.anl.gov/~more/dfo/matlab/perf_profile.m

Para comparar los tiempos de un mismo problema resuelto con diversos métodos, Dolan e Moré [6] considera la *razón de desempeño*, dada por

$$r_{p,s} = \frac{t_{p,s}}{\min\{t_{p,s} : 1 \leq s \leq n_s\}}.$$

El método más eficiente para un problema es aquel que tiene razón uno, y para los demás métodos la razón es mayor que uno. Luego, definimos el *desempeño del método s* como una probabilidad empírica construida a partir de la serie de tiempos de razón de desempeño $\{r_{p,s} : 1 \leq p \leq n_p\}$. Es decir,

$$(11) \quad \rho_s(\tau) = \frac{1}{n_p} \#\{p \in \mathcal{P} : r_{p,s} \leq \tau\},$$

entonces, $\rho_s(\tau)$ representa la probabilidad del método s en que la razón de desempeño $r_{p,s}$ sea menor o igual que un mejor razón posible τ . Así, los métodos que alcancen una probabilidad alta para menores valores de τ , son considerados eficientes. Además, si la probabilidad llegara cerca de uno para valores no muy lejos de uno, sería considerado como el método más robusto.

En la Figura 6 se muestra la curva de desempeño de Dolan y Moré [6], donde ninguno de los abordajes se diferencia de manera significativa de los otros, aunque el criterio basado en la norma-1 se comporta mejor que los otros. Sin embargo, en Castro, Heredia y Oliveira [4], Casacio et al [3] y Bocanegra [1] prefieren utilizar el criterio de pre-ordenación por la norma-2 seguramente por la facilidad de interpretación geométrica. Sin embargo, por el presente trabajo, es irrelevante usar cualquiera de las normas equivalentes de \mathbb{R}^n u cualquiera de los criterios extras propuestos. Los detalles de cálculo de la curva de desempeño se puede ver en Suñagua y Oliveira [15].

5. CONCLUSIONES

Por lo expuesto en las secciones precedentes una de las abordajes eficientes para resolver problemas de programación lineal es la estrategia híbrida que consiste en aplicar, en la primera fase, el preconditionador basado en la descomposición incompleta controlada de Cholesky de Campos y Rollet [2], y en la segunda fase, cambiar al preconditionador separador basado en identificar una base de columnas de $AD^{1/2}$ bajo el algoritmo de factorización LU rectangular previa una ordenación de columnas de la misma matriz basada en norma-1 o norma-2, como fue desarrollada en Suñagua y Oliveira [15].

La curva de desempeño de la Figura 6 basada en Dolan y Moré [6], muestra que todos los criterios planteados de pre-ordenación pueden ser aplicados, sin embargo, las de normas 1 y 2 son las que toman en cuenta todas las variaciones de las columnas de la matriz $AD^{1/2}$, así cerca del punto óptimo, la matriz preconditionada del sistema normal $ADA^T = AD^{1/2}D^{1/2}A^T$ mejora su número de condición gracias al preconditionador separador.

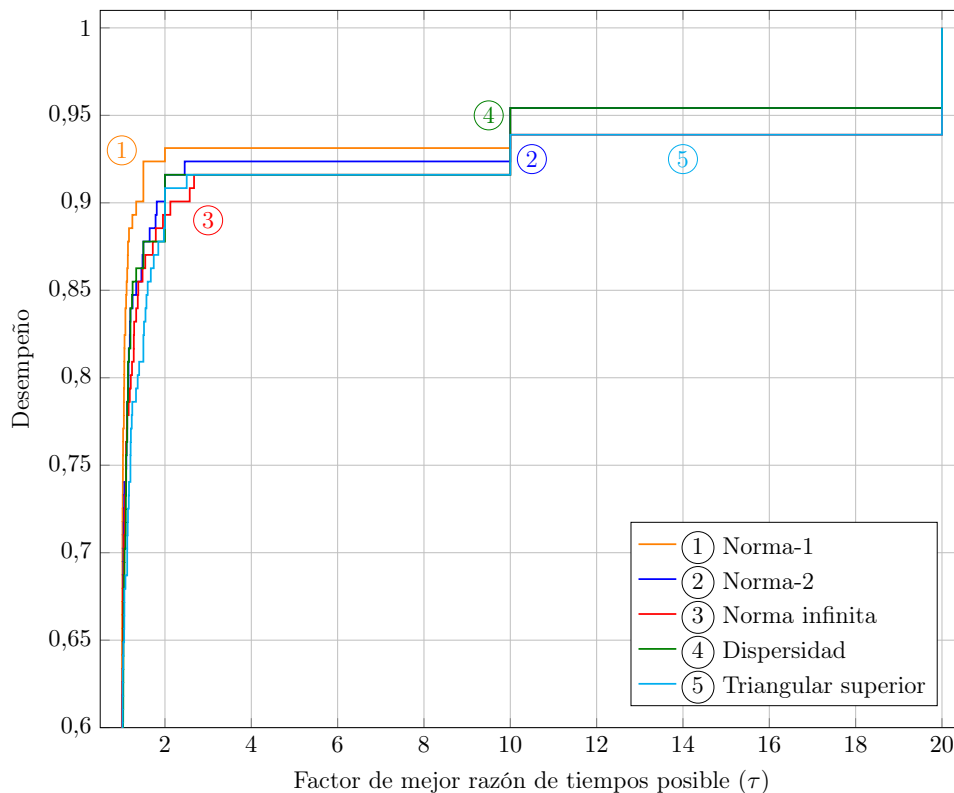


FIGURA 6. Curva de desempeño de los diferentes métodos.

REFERENCIAS

- [1] Silvana Bocanegra, FF. Campos, and Aurelio R.L. Oliveira. Using a hybrid preconditioner for solving large-scale linear systems arising from interior point methods. *Computational Optimization and Applications*, 36(2-3):149–164, 2007.
- [2] Frederico Ferreira Campos and John S Rollet. *Controlled Cholesky factorization for preconditioning the conjugate gradient method*. Oxford University Computing Laboratory, Numerical Analysis Group, 1995.
- [3] Luciana Casacio, Christiano Lyra, Aurelio Ribeiro Leite Oliveira, and Cecilia Orellana Castro. Improving the preconditioning of linear systems from interior point methods. *Computers & Operations Research*, 85:129–138, 2017.
- [4] Cecilia Orellana Castro, Manolo Rodriguez Heredia, and Aurelio RL Oliveira. Recycling basic columns of the splitting preconditioner in interior point methods. *Computational Optimization and Applications*, pages 1–30, 2023.
- [5] Joseph Czyzyk, Sanjay Mehrotra, Michael Wagner, and Stephen J Wright. Pcx user guide (version 1.1). *Optimization Technology Center, Northwestern University*, 1997.
- [6] Elizabeth D Dolan and Jorge J More. Benchmarking optimization software with performance profiles. *Mathematical programming*, 91(2):201–213, 2002.
- [7] Gene H Golub and Charles F Van Loan. *Matrix computations*. JHU Press, 2013.
- [8] Jacek Gondzio. Multiple centrality corrections in a primal-dual method for linear programming. *Computational Optimization and Applications*, 6(2):137–156, 1996.
- [9] Jacek Gondzio. Interior point methods 25 years later. *European Journal of Operational Research*, 218(3):587–601, 2012.

- [10] N.K. Karmarkar. A new polynomial-time algorithm for linear programming. *Combinatorica*, 4:373–395, 1984.
- [11] Chih-Jen Lin and Jorge J Moré. Incomplete cholesky factorizations with limited memory. *SIAM Journal on Scientific Computing*, 21(1):24–45, 1999.
- [12] Sanjay Mehrotra. Quadratic convergence in a primal-dual method. *Mathematics of Operations Research*, 18(3):741–751, 1993.
- [13] Aurelio R.L. Oliveira and Danny C. Sorensen. A new class of preconditioners for large-scale linear systems from interior point methods for linear programming. *Linear Algebra and its applications*, 394:1–24, 2005. England.
- [14] Christopher C Paige and Michael A Saunders. Solution of sparse indefinite systems of linear equations. *SIAM Journal on Numerical Analysis*, 12(4):617–629, 1975.
- [15] Porfirio Suñagua and Aurelio RL Oliveira. A new approach for finding a basis for the splitting preconditioner for linear systems from interior point methods. *Computational Optimization and Applications*, 67(1):111–127, 2017.
- [16] Margaret H. Wright. Interior methods for constrained optimization. *Acta Numérica*, 1:341–407, 1992.

P. SUÑAGUA, Instituto de Investigación Matemática, Universidad Mayor de San Andrés.
Calle 27 de Cota Cota, Campus Universitario, Edificio de Ciencias Puras y Naturales, 1er Piso. La Paz– Bolivia
E-mail:psunagua@umsa.bo