

Aspectos computacionales de la resolución y obtención de Modelos de Ecuaciones Simultáneas

Jose Juan López Espín

dirigida por:
Domingo Giménez

diciembre de 2009



Esquema

- 1 Introducción
- 2 Descripción Teórica
- 3 MC2E basado en la descomposición de la inversa
- 4 Paralelización de los algoritmos en memoria distribuida
- 5 MC2E basado en la descomposición QR
- 6 Algoritmos QR con nodos ficticios
- 7 Obtención de M.E.S. mediante AG
- 8 Aplicaciones
- 9 Conclusiones y Trabajos futuros

Introducción

Motivación

- Los Modelos de Ecuaciones Simultáneas (M.E.S.) nacen como herramienta econométrica.
- Limitados por su gran necesidad computacional.
- Actualmente han comenzado a usarse en otras disciplinas.
- No conocemos la existencia de software en paralelo para la resolución de M.E.S. (aunque sí en secuencial).
- Solo se conocen estudios sobre algoritmos paralelos para M.E.S. de estimadores de información completa.
- No conocemos la existencia de estudios o software capaz de encontrar el mejor M.E.S. a partir de los datos de las variables.

Objetivos

Los **objetivos** del presente trabajo son:

- **Resolver Modelos de Ecuaciones Simultáneas** de manera más eficiente reduciendo el tiempo de ejecución y los recursos en memoria.
 - Utilizar descomposiciones matriciales (descomposición de la matriz inversa y la descomposición QR).
 - Desarrollar versiones paralelas en memoria compartida y distribuida.
- **Desarrollar una herramienta** que permita **encontrar el mejor modelo** posible a partir de un conjunto de datos de variables.
 - Técnicas Metaheurísticas.
 - Avance Rápido.
- **Aplicar** las herramientas desarrolladas a **problemas reales**.

Descripción Teórica y algoritmos básicos

¿Que es un M.E.S.?

El esquema de un modelo con N ecuaciones, N variables endógenas y K variables exógenas en forma matricial es:

$$BY^T + \Gamma X^T + u^T = 0$$

siendo:

$$Y = (y_1 \dots y_N) , X = (x_1 \dots x_K) , u = (u_1 \dots u_N)$$

$$B = \begin{pmatrix} \beta_{1,1} & \dots & \beta_{1,N} \\ & \dots & \\ \beta_{N,1} & \dots & \beta_{N,N} \end{pmatrix} , \Gamma = \begin{pmatrix} \gamma_{1,1} & \dots & \gamma_{1,K} \\ & \dots & \\ \gamma_{N,1} & \dots & \gamma_{N,K} \end{pmatrix}$$

matrices densas con valores reales, siendo $\beta_{i,i} = -1 \forall i = 1, \dots, N$.

El modelo estructural se puede expresar también en forma reducida:

$$Y = X\Pi + v, \text{ con } \Pi^T = -B^{-1}\Gamma , v^T = -B^{-1}u^T.$$

¿Que es un M.E.S.?

Si se descompone B de la forma $B = \text{diag}(-1, \dots, -1) + \tilde{B}$, la expresión anterior queda de la siguiente forma:

$$Y^T = \tilde{B}Y^T + \Gamma X^T + u^T$$

O en ecuaciones:

$$y_1 = \gamma_{1,1}x_1 + \dots + \gamma_{1,K}x_K + \beta_{1,2}y_2 + \beta_{1,3}y_3 + \dots + \beta_{1,N}y_N + u_1$$

$$y_2 = \gamma_{2,1}x_1 + \dots + \gamma_{2,K}x_K + \beta_{2,1}y_1 + \beta_{2,3}y_3 + \dots + \beta_{2,N}y_N + u_2$$

...

$$y_N = \gamma_{N,1}x_1 + \dots + \gamma_{N,K}x_K + \beta_{N,1}y_1 + \dots + \beta_{N,N-1}y_{N-1} + u_N$$

donde x_1, x_2, \dots, x_K son variables exógenas, y_1, y_2, \dots, y_N son variables endógenas, y u_1, u_2, \dots, u_N son variables de ruido blanco. Todas ellas son vectores de dimensión d , siendo d el tamaño de la muestra.

El problema de la identificación

Hay tres tipos de ecuaciones:

- subidentificadas: No se pueden resolver.
- sobreidentificadas: Las soluciones de los coeficientes no se obtienen de forma única.
- exactamente identificadas: Solución única.

Para estudiar la identificación de una ecuación se usan dos condiciones:

- Condición de Orden (condición necesaria aunque no suficiente):
Identificada si $n_i - 1 \leq K - k_i$, no identificada en caso contrario.
- Condición de Rango (condición necesaria y suficiente):
Identificada sí y solo sí puede construirse al menos una matriz no singular de tamaño $(N - 1) \times (N - 1)$ con los coeficientes de las variables excluidas de esa ecuación.

Mínimos Cuadrados Ordinarios (MCO)

Algoritmo 1 Algoritmo MCO_{bas}

Entrada: $X \in \mathbb{R}^{d \times K}$, $Y \in \mathbb{R}^{d \times N}$

Salida: $\beta \in \mathbb{R}^{K \times N}$, $\hat{Y} \in \mathbb{R}^{d \times N}$

- 1: Hacer $X^T X$ {coste $\rightarrow 2K^2d$ }
 - 2: Hacer $X^T Y$ {coste $\rightarrow 2NKd$ }
 - 3: Hacer $(X^T X)^{-1}(X^T Y)$ resolviendo el sistema $X^T Y = (X^T X)\beta$
{coste $\rightarrow \frac{2}{3}K^3 + 2K^2N$ }
 - 4: **Si** estimación=**Verdadero Entonces**
 - 5: Hacer $\hat{Y} = X\beta$ {coste $\rightarrow 2NKd$ }
 - 6: **Fin si**
-

Utilizar las librerías de Blas y Lapack.

Mínimos Cuadrados Indirectos (MCI)

- Solo se puede usar en ecuaciones exactamente identificadas ($K = k_i + n_i - 1$).
- Estima los valores de Π de la ecuación $Y = X\Pi + v$ mediante MCO.
- A continuación deduce los valores de la forma estructural ($0 = B_i Y^T + \Gamma_i X^T + u^T$) siguiendo la relación $-B_i \Pi = \Gamma_i$.

Algoritmo 2 Algoritmo MCI

Entrada: $X \in \mathbb{R}^{d \times K}$, $Y \in \mathbb{R}^{d \times N}$, $B \in \mathbb{R}^{N \times N}$ y $\Gamma \in \mathbb{R}^{N \times K}$

Salida: $B \in \mathbb{R}^{N \times N}$, $\Gamma \in \mathbb{R}^{N \times K}$

- 1: $\Pi^T = \text{MCO}_{bas}(Y, X, \text{estimación}=\text{Falso})$
 - 2: **Para** $i=1 \dots N$ **Hacer**
 - 3: **Si** ecuación i es exactamente identificada **Entonces**
 - 4: Resolver $-B_i \Pi = \Gamma_i$
 - 5: **Fin si**
 - 6: **Fin Para**
-

Mínimos Cuadrados en Dos Etapas (MC2E)

Estimar los valores de B y Γ en la expresión:

$$Y^T = \tilde{B}Y^T + \Gamma X^T + u^T$$

No es posible usar Mínimos Cuadrados (Y correlacionado con u).

Algoritmo 3 Algoritmo $MC2E_{bas}$

Entrada: $X \in \mathbb{R}^{d \times K}$, $Y \in \mathbb{R}^{d \times N}$, $B \in \mathbb{R}^{N \times N}$ y $\Gamma \in \mathbb{R}^{N \times K}$

Salida: $B \in \mathbb{R}^{N \times N}$, $\Gamma \in \mathbb{R}^{N \times K}$

- 1: Estimar las variables proxy $\hat{Y} = MCO_{bas}(Y, X, \text{estimación}=\mathbf{Verdadero})$
 - 2: Sustituir Y por \hat{Y} resultando $Y^T = \tilde{B}\hat{Y}^T + \Gamma X^T + u^T$
 - 3: **Para** $i = 1 \dots N$ **Hacer**
 - 4: Hacer $X_{e_i} = [X_i | \hat{Y}_i]$ {Usando B_i y Γ_i }
 - 5: $\beta_i = MCO_{bas}(y_i, X_{e_i}, \text{estimación}=\mathbf{Falso})$
 - 6: Sustituir los valores de β_i en las incógnitas de B_i y Γ_i
 - 7: **Fin Para**
-

MC2E basado en la descomposición de la inversa

MC2E basado en la descomposición de la inversa

La descomposición matricial que se va a usar está basada en el siguiente teorema:

Teorema

$$\begin{pmatrix} A & B \\ B^T & D \end{pmatrix}^{-1} = \begin{pmatrix} A^{-1} & 0 \\ 0 & 0 \end{pmatrix} + \begin{pmatrix} -E \\ Id \end{pmatrix} (D - B^T A^{-1} B)^{-1} (-E^T | Id)$$

donde $E = A^{-1}B$.

Supongamos que se va a resolver la ecuación i -ésima usando MC2E.

$$y_i = \gamma_0 + \gamma_1 x_{i_1} + \dots + \gamma_{k_i} x_{i_{k_i}} + \alpha_1 \hat{y}_{i_1} + \dots + \alpha_{n_i} \hat{y}_{i_{n_i}} + \epsilon$$

Las matriz usada al aplicar MCO en la ecuación es $[X_i | \hat{Y}_i]$ (con dimensión $d \times (n_i + k_i - 1)$), calculándose la expresión $([X_i | \hat{Y}_i]^T [X_i | \hat{Y}_i])^{-1} [X_i | \hat{Y}_i]^T y_i$.

MC2E basado en la descomposición de la inversa

Usando el teorema previo, la inversa de la matriz:

$$\left(\begin{array}{c|c} \begin{bmatrix} X_i^T \\ \hat{Y}_i^T \end{bmatrix} & [X_i | \hat{Y}_i] \end{array} \right)^{-1} = \begin{pmatrix} X_i^T X_i & X_i^T \hat{Y}_i \\ \hat{Y}_i^T X_i & \hat{Y}_i^T \hat{Y}_i \end{pmatrix}^{-1} = \begin{pmatrix} (X_i^T X_i)^{-1} & 0 \\ 0 & 0 \end{pmatrix} + \begin{pmatrix} -(X_i^T X_i)^{-1} X_i^T \hat{Y}_i \\ Id \end{pmatrix} (\hat{Y}_i^T \hat{Y}_i - Y_i^T X_i (X_i^T X_i)^{-1} X_i^T \hat{Y}_i)^{-1} (-Y_i^T X_i (X_i^T X_i)^{-1} | Id_{n_i-1})$$

- $X_i^T X_i$ se toma de $X^T X$, que se obtiene al calcular Π .
- $X_i^T \hat{Y}_i$ se toma de $X^T Y$, que se obtiene al calcular Π (ya que $X^T \hat{Y} = X^T Y$).
- $(X_i^T X_i)^{-1} [X_i^T \hat{Y}_i | Id_{k_i}]$ se tiene que calcular.
- $\hat{Y}_i^T X_i (X_i^T X_i)^{-1} X_i^T \hat{Y}_i$ se tiene que calcular.
- $\hat{Y}_i^T \hat{Y}_i$ se toma de $\hat{Y}^T \hat{Y}$, que es calculado antes.
- $\hat{Y}_i^T \hat{Y}_i - \hat{Y}_i^T X_i (X_i^T X_i)^{-1} X_i^T \hat{Y}_i$ se tiene que calcular.
- $(\hat{Y}_i^T \hat{Y}_i - \hat{Y}_i^T X_i (X_i^T X_i)^{-1} X_i^T \hat{Y}_i)^{-1} (-\hat{Y}_i^T X_i (X_i^T X_i)^{-1} | Id_{n_i-1})$ se tiene que calcular.
- Multiplicar $\begin{pmatrix} -(X_i^T X_i)^{-1} X_i^T \hat{Y}_i \\ Id_{n_i-1} \end{pmatrix}$ por la matriz resultante del paso anterior.
- Sumar la matriz resultante del paso anterior a $\begin{pmatrix} (X_i^T X_i)^{-1} & 0 \\ 0 & 0 \end{pmatrix}$.

MC2E basado en la descomposición de la inversa

Se ha calculado la inversa de la matriz en la expresión de MCO para la ecuación i -ésima $(([X_i|\hat{Y}_i]^T [X_i|\hat{Y}_i])^{-1} [X_i|\hat{Y}_i]^T y_i)$.

- La matriz $[X_i|\hat{Y}_i]^T y_i$ tiene que ser también calculada.
 - $X_i^T y_i$ se toma de $X^T Y$, que se obtiene al calcular Π .
 - $\hat{Y}_i^T y_i$ se toma de $\hat{Y}^T \hat{Y}$ puesto que $\hat{Y}^T \hat{Y} = \hat{Y}^T Y$.
- Se calcula la multiplicación de ambas matrices completando la expresión de MCO.

Proposición El número de flops realizados al resolver una ecuación de un M.E.S. por MCO_{bas} es mayor o igual que el número de flops realizados al resolver la misma ecuación por MCO_{inv} .

MC2E basado en la descomposición de la inversa

El algoritmo para MC2E basado en la descomposición de la inversa:

Algoritmo 4 Algoritmo $MC2E_{inv}$

Entrada: $X \in \mathbb{R}^{d \times K}$, $Y \in \mathbb{R}^{d \times N}$, $B \in \mathbb{R}^{N \times N}$ y $\Gamma \in \mathbb{R}^{N \times K}$

Salida: $B \in \mathbb{R}^{N \times N}$, $\Gamma \in \mathbb{R}^{N \times K}$

- 1: $\hat{Y} = MCO_{bas}(Y, X, \text{Verdadero})$ {guardando Π , $X^T X$ y $X^T Y$ }
 - 2: Hacer $\hat{Y}^T \hat{Y}$
 - 3: **Para** $j=1 \dots N$ **Hacer**
 - 4: $\beta = MCO_{inv}(y_j, \hat{Y}, X, X^T X, X^T Y, \hat{Y}^T \hat{Y})$
 - 5: Sustituir los valores de β en las incógnitas de B_j y Γ_j
 - 6: **Fin Para**
-

Cuando $k_i = 0$ o $n_i = 1$ no se puede aplicar esta segunda versión debido a que no se puede construir la matriz X_i o la matriz Y_i .

Estudio Experimental

Tiempo de ejecución (en segundos) y ratio entre el algoritmo $MC2E_{bas}$ y $MC2E_{inv}$:

N :	500	1000	1500	2000	2500
d :	500	500	1000	1000	1500
$MC2E_{bas}$	72.82	790.93	7031.96	19337.92	97874.21
$MC2E_{inv}$	12.91	198.16	1225.33	4192.10	10217.02
Ratio	5.64	3.99	5.74	4.61	9.58

Tiempo de ejecución (en segundos) y ratio comparativo entre el algoritmo MCI y el algoritmo $MC2E_{inv}$:

N :	500	1000	1500	2000	2500
d :	500	500	1000	1000	1500
$MC2E_{inv}$	22.76	289.12	1665.99	6165.98	14316.93
MCI	18.17	230.78	1073.21	3118.85	7181.54
Ratio	1.25	1.25	1.55	1.98	1.99

Algoritmo para la resolución de M.E.S.

Algoritmo 5 Algoritmo *MES*

Entrada: $X \in \mathbb{R}^{d \times K}$, $Y \in \mathbb{R}^{d \times N}$, $B \in \mathbb{R}^{N \times N}$ y $\Gamma \in \mathbb{R}^{N \times K}$

Salida: $B \in \mathbb{R}^{N \times N}$, $\Gamma \in \mathbb{R}^{N \times K}$

- 1: $\hat{Y} = MCO_{bas}(Y, X, \text{Verdadero})$ {guardando Π , $X^T X$ y $X^T Y$ }
 - 2: Hacer $\hat{Y}^T \hat{Y}$
 - 3: **Para** $i=1 \dots N$ **Hacer**
 - 4: **Si** la ecuación i es exactamente identificada **Entonces**
 - 5: Resolver $-B_i \Pi = \Gamma_i$ {Se utiliza *MCI*}
 - 6: **Si no**, **Si** la ecuación i es sobreidentificada **Entonces**
 - 7: **Si** $k_i > 0$ y $n_i > 1$ **Entonces**
 - 8: $\beta = MCO_{inv}(y_i, \hat{Y}, X, X^T X, X^T Y, \hat{Y}^T \hat{Y})$ {Se utiliza *MC2E_{inv}*}
 - 9: **Si no**
 - 10: $\beta = MCO_{bas}(y_i, X_e, \text{Falso})$ {Se utiliza *MC2E_{bas}*}
 - 11: **Fin si**
 - 12: Sustituir los valores de β en las incógnitas de B_i y Γ_i
 - 13: **Si no**
 - 14: no hacer nada {la ecuación i no está identificada}
 - 15: **Fin si**
 - 16: **Fin Para**
-

Paralelización de los algoritmos en memoria distribuida

Paralelización de MCO

Utilizar las librerías de ScaLAPACK y PBLAS.

Importante: Tiempo de comunicación entre procesadores.

Algoritmo 6 Algoritmo $PMCO_{bas}$

Entrada: $X \in \mathbb{R}^{d \times K}$ e $Y \in \mathbb{R}^{d \times N}$ (distribuidas al estilo de ScaLAPACK)

Salida: $\beta \in \mathbb{R}^{K \times N}$ e $\hat{Y} \in \mathbb{R}^{d \times N}$ (distribuidas al estilo de ScaLAPACK)

- 1: Hacer $X^T X$ {Multiplicación en paralelo}
 - 2: Hacer $X^T Y$ {Multiplicación en paralelo}
 - 3: Hacer $(X^T X)^{-1}(X^T Y)$ resolviendo el sistema $X^T Y = (X^T X)\beta$ en paralelo
 - 4: **Si** estimación=**Verdadero Entonces**
 - 5: Hacer $\hat{Y} = X\beta$ {Multiplicación en paralelo}
 - 6: **Fin si**
-

Paralelización de MCI

Algoritmo 7 Algoritmo *PMCI*

Entrada: $X \in \mathbb{R}^{d \times K}$ e $Y \in \mathbb{R}^{d \times N}$ (distribuidas al estilo de ScaLAPACK), $B \in \mathbb{R}^{N \times N}$ y $\Gamma \in \mathbb{R}^{N \times K}$ (distribuidas $\frac{N}{p}$ filas por procesador)

Salida: $B \in \mathbb{R}^{N \times N}$ y $\Gamma \in \mathbb{R}^{N \times K}$ (distribuidas $\frac{N}{p}$ filas por procesador)

- 1: $\Pi^T = PMCO_{bas}(Y, X, \text{Falso})$
 - 2: Distribuir Π en todos los procesadores
 - 3: EN PARALELO cada procesador $q = 0, \dots, p - 1$ HACE
 - 4: **Para** $j=1 \dots \frac{N}{p}$ **Hacer**
 - 5: $i = q + (j - 1)p + 1$
 - 6: **Si** ecuación i es exactamente identificada **Entonces**
 - 7: Resolver $-B_i \Pi = \Gamma_i$
 - 8: **Fin si**
 - 9: **Fin Para**
 - 10: FIN PARALELO
-

Paralelización de MC2E

Algoritmo 8 Algoritmo $PMC2E_{inv}$

Entrada: $X \in \mathbb{R}^{d \times K}$ e $Y \in \mathbb{R}^{d \times N}$ (distribuidas al estilo de ScaLAPACK), $B \in \mathbb{R}^{N \times N}$ y $\Gamma \in \mathbb{R}^{N \times K}$ (distribuidas $\frac{N}{p}$ filas por procesador)

Salida: $B \in \mathbb{R}^{N \times N}$ y $\Gamma \in \mathbb{R}^{N \times K}$ (distribuidas $\frac{N}{p}$ filas por procesador)

- 1: $\hat{Y} = PMCO_{bas}(Y, X, \mathbf{Verdadero})$ {guardando Π , $X^T X$ y $X^T Y$ }
 - 2: Hacer $\hat{Y}^T \hat{Y}$ {multiplicaciones en paralelo}
 - 3: Distribuir Π , $X^T X$, $X^T Y$, \hat{Y} , $\hat{Y}^T \hat{Y}$ a todos los procesadores
 - 4: EN PARALELO cada procesador $q = 0, \dots, p - 1$ HACE
 - 5: **Para** $j=1 \dots \frac{N}{p}$ **Hacer**
 - 6: $i = q + (j - 1)p + 1$
 - 7: $\beta_i = MCO_{inv}(y_i, \hat{Y}, X, X^T X, X^T Y, \hat{Y}^T \hat{Y})$
 - 8: Sustituir los valores de β_i en las incógnitas de B_i y Γ_i
 - 9: **Fin Para**
 - 10: FIN PARALELO
-

Algoritmo paralelo para la resolución de M.E.S.

Algoritmo 9 Algoritmo PMES

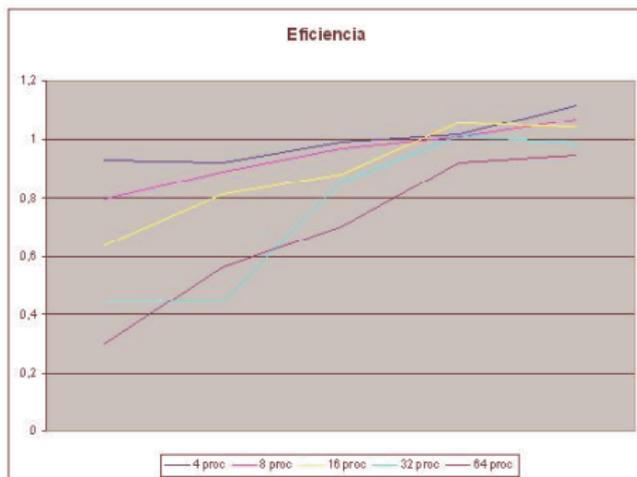
Entrada: $X \in \mathbb{R}^{d \times K}$ e $Y \in \mathbb{R}^{d \times N}$ (distribuidas al estilo de ScaLAPACK), $B \in \mathbb{R}^{N \times N}$ y $\Gamma \in \mathbb{R}^{N \times K}$ (distribuidas $\frac{N}{p}$ filas por procesador)

Salida: $B \in \mathbb{R}^{N \times N}$ y $\Gamma \in \mathbb{R}^{N \times K}$ (distribuidas $\frac{N}{p}$ filas por procesador)

- 1: $\hat{Y} = PMCO_{bas}(Y, X, \text{Verdadero})$ {guardando $\Pi, X^T X$ y $X^T Y$ }
- 2: $\hat{Y}^T \hat{Y}$ {multiplicación en paralelo}
- 3: Distribuir $\Pi, X^T X, X^T Y, \hat{Y}, \hat{Y}^T \hat{Y}$ en todos los procesadores
- 4: EN PARALELO cada procesador $q = 0, \dots, p - 1$ HACE
- 5: **Para** $j=1 \dots \frac{N}{p}$ **Hacer**
- 6: $i = q + (j - 1)p + 1$
- 7: **Si** la ecuación i es exactamente identificada **Entonces**
- 8: Resolver $-B_i \Pi = \Gamma_i$ {Se utiliza MCI}
- 9: **Si no, Si** la ecuación i es sobreidentificada **Entonces**
- 10: **Si** $k_i > 0$ y $n_i > 1$ **Entonces**
- 11: $\beta_i = MCO_{inv}(y_i, \hat{Y}, X, X^T X, X^T Y, \hat{Y}^T \hat{Y})$ {Se utiliza MC2E_{inv}}
- 12: **Si no**
- 13: $\beta_i = MCO_{bas}(y_i, X_e, \text{Falso})$ {Se utiliza MC2E_{bas}}
- 14: **Fin si**
- 15: Sustituir los valores de β_i en las incógnitas de B_i y Γ_i
- 16: **Si no**
- 17: no hacer nada {la ecuación i no está identificada}
- 18: **Fin si**
- 19: **Fin Para**
- 20: **FIN PARALELO**

Estudio experimental

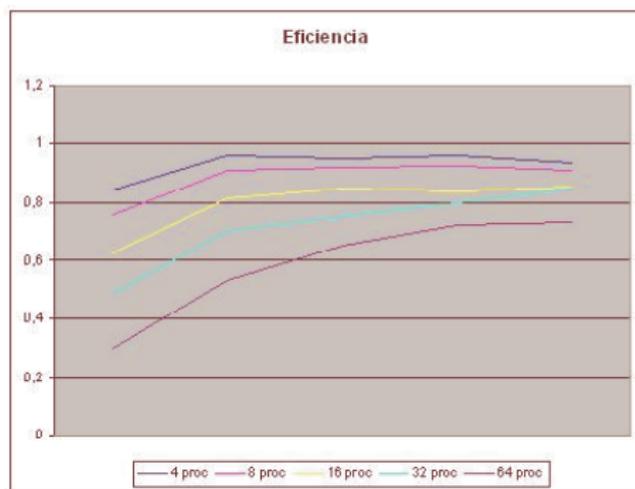
Eficiencia y speed-up del algoritmo *PMCI*.



N :	500	1000	1500	2000	2500
d :	500	500	1000	1000	1500
p	Speed-up				
4	3.72	3.69	3.97	4.06	4.46
8	6.37	7.15	7.77	8.07	8.54
16	10.22	13.02	14.11	16.93	16.67
32	14.44	14.29	27.42	32.40	31.55
64	19.27	35.96	44.92	59.03	60.79

Estudio experimental

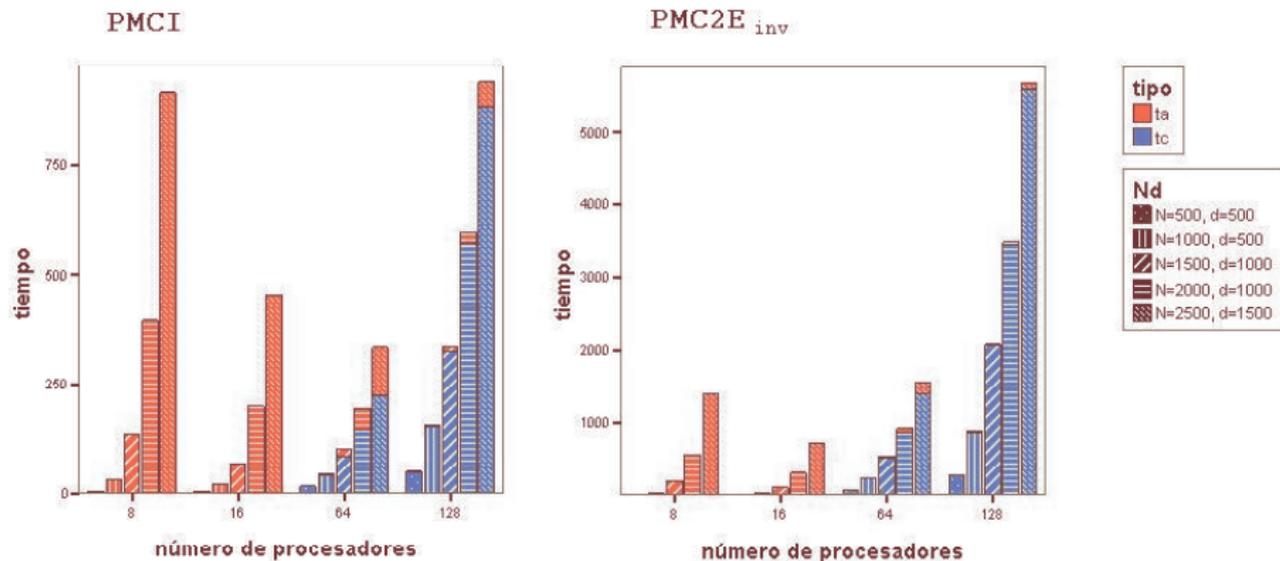
Eficiencia y speed-up del algoritmo $PMC2E_{inv}$.



N :	500	1000	1500	2000	2500
d :	500	500	1000	1000	1500
p	Speed-up				
4	3.39	3.85	3.82	3.85	3.75
8	6.07	7.30	7.34	7.42	7.28
16	10.00	13.06	13.61	13.44	13.69
32	15.55	22.45	24.05	25.61	27.25
64	18.96	33.94	41.61	46.42	46.98

Estudio experimental

Gráfico comparativo de los tiempos de comunicación (t_c) y de operaciones aritméticas (t_a) de los algoritmos $PMCI$ y $PMC2E_{inv}$:



MC2E basado en la descomposición QR

MC2E mediante reflexiones de Householder

- Hacer la descomposición QR de X ($X = Q \begin{pmatrix} R_1 \\ 0 \end{pmatrix}$ mediante reflexiones de Householder).
- Calcular \hat{Y}

$$\begin{aligned} \hat{Y} &= X(X^T X)^{-1} X^T Y = QR(R^T Q^T QR)^{-1} R^T Q^T Y = \\ &= Q \begin{pmatrix} R_1 \\ 0 \end{pmatrix} (R_1^{-1} R_1^{-T}) [R_1^T | 0] Q^T Y = Q \begin{pmatrix} Id_K \\ 0 \end{pmatrix} [Id_K | 0] Q^T Y \\ &= Q \begin{pmatrix} Id_K & 0 \\ 0 & 0 \end{pmatrix} Q^T Y = Q \begin{pmatrix} Id_K & 0 \\ 0 & 0 \end{pmatrix} \begin{pmatrix} \tilde{Y}_1 \\ \tilde{Y}_2 \end{pmatrix} = Q \begin{pmatrix} \tilde{Y}_1 \\ 0 \end{pmatrix} \end{aligned}$$

- En cada ecuación i :
 - Hacer la descomposición QR de la matriz $X_{e_i} = Q_i R_i$ mediante reflexiones de Householder.
 - Calcular $\hat{\beta}_i = (X_{e_i}^T X_{e_i})^{-1} X_{e_i}^T y_i = [R_{i,1}^{-1} | 0] Q_i^T y_i = [R_{i,1}^{-1} | 0] \tilde{y}_i = R_{i,1}^{-1} \tilde{y}_{i,1}$

Algoritmo para MC2E mediante reflexiones de Householder

Algoritmo 10 Algoritmo $MC2E_H$

Entrada: $X \in \mathbb{R}^{d \times K}$, $Y \in \mathbb{R}^{d \times N}$, $B \in \mathbb{R}^{N \times N}$ y $\Gamma \in \mathbb{R}^{N \times K}$

Salida: $B \in \mathbb{R}^{N \times N}$, $\Gamma \in \mathbb{R}^{N \times K}$

- 1: Obtener Q y R (desc. QR de X) {coste $\rightarrow \frac{4}{3}K^2(3d - K)$ }
 - 2: $\tilde{Y} = Q^T Y$ {coste $\rightarrow 2NK(2d - K)$ }
 - 3: $\hat{Y} = Q \begin{pmatrix} \tilde{Y}_1 \\ 0 \end{pmatrix}$ {coste $\rightarrow 2NK^2$ }
 - 4: **Para** $i=1 \dots N$ **Hacer**
 - 5: Obtener Q_i y R_i (desc. QR de X_{e_i}) {coste $\rightarrow \frac{4}{3}(n_i + k_i - 1)^2(3d - (n_i + k_i - 1))$ }
 - 6: $\tilde{y}_i = Q_i^T y_i$ {coste $\rightarrow 2(n_i + k_i - 1)(2d - (n_i + k_i - 1))$ }
 - 7: $\beta_i = R_{i,1}^{-1} \tilde{y}_{i,1}$ {coste $\rightarrow (n_i + k_i - 1)^2$ }
 - 8: Sustituir los valores de β_i en las incógnitas de B_i y Γ_i
 - 9: **Fin Para**
-

Estudio Experimental

Tiempo de ejecución (en segundos) y ratio comparativo entre el algoritmo $MC2E_{bas}$, $MC2E_{inv}$ y $MC2E_H$.

$N:$	500	1000	1500	2000	2500
$d:$	500	500	1000	1000	1500
Alg.	Tiempo	Tiempo	Tiempo	Tiempo	Tiempo
$MC2E_{bas}$	72.82	790.93	7031.96	19337.92	97874.21
$MC2E_{inv}$	12.91	198.16	1225.33	4192.10	10217.02
$MC2E_H$	26.53	257.10	2363.64	5457.78	16825.37
	Ratio				
$MC2E_{bas}/MC2E_{inv}$	5.64	3.99	5.74	4.61	9.58
$MC2E_H/MC2E_{inv}$	2.05	1.30	1.93	1.30	1.65

MC2E mediante rotaciones de Givens

Se detectan las siguientes deficiencias en el algoritmo anterior:

- No se aprovecha la descomposición QR de X en la descomposición en cada ecuación.
- El cálculo de la matriz \hat{Y} exige aplicar dos veces los reflectores de Householder a la matriz Y .

Se propone: Obtener la descomposición QR de cada X_{e_i} mediante rotaciones de Givens haciendo ceros los elementos no nulos por debajo de la diagonal principal.

- Hacer la descomposición QR de $[X|\hat{Y}]$ de la cual se obtendrán las descomposiciones QR de las X_{e_i} .
- Hacer la descomposición QR de X mediante reflexiones de Householder.

- Calcular $Q^T[X|\hat{Y}] = [Q^T X | Q^T \hat{Y}] = [R | Q^T \hat{Y}] = \begin{pmatrix} R_1 & \tilde{Y}_1 \\ 0 & 0 \end{pmatrix}$

MC2E mediante rotaciones de Givens II

Resolver la ecuación i -ésima $y_i = X_i b_i + Y_i \Gamma_i + \epsilon_i$ cuya matriz asociada es $X_{e_i} = [X_i | \hat{Y}_i]$.

La matriz se puede expresar de la forma $X_{e_i} = [X | \hat{Y}] S_i$ donde S_i es una matriz de selección.

$$Q^T X_{e_i} = Q^T [X | \hat{Y}] S_i = [R | Q^T \hat{Y}] S_i = \begin{pmatrix} R_1 & \tilde{Y}_1 \\ 0 & 0 \end{pmatrix} S_i = \begin{pmatrix} R_{i,1} & \tilde{Y}_{i,1} \\ 0 & 0 \end{pmatrix},$$

siendo $R_{i,1}$ e $\tilde{Y}_{i,1}$ las K primeras filas de la matriz resultante de la multiplicación.

Es necesario obtener la descomposición QR de X_{e_i} , por lo que hay que encontrar una matriz ortogonal que multiplicada a la anterior dé como resultado una matriz triangular superior.

$$\tilde{Q}_i^T = \left(\prod_{n=1}^{n_i-1} \prod_{j=1}^{K-n} G_{K-j-k_i, K-j-k_i+1}^{(i)} \right) \left(\prod_{n=1}^{k_i} \prod_{j=1}^{\lambda_{i,n}-n} G_{\lambda_{i,n}-j, \lambda_{i,n}-j+1}^{(n)} \right)$$

MC2E mediante rotaciones de Givens III

La siguiente figura muestra la secuencia de eliminación de elementos en el proceso de retriangularización mediante rotaciones de Givens de una matriz $X \in \mathbb{R}^{7 \times 5}$. La entrada i ($i=1, \dots, 6$) indica el elemento reducido a cero en la i -ésima rotación, multiplicando a la izquierda por la matriz

$$\tilde{Q}_i^T = G_{5,6}^{(5)} G_{5,7}^{(5)} G_{4,5}^{(4)} G_{4,6}^{(4)} G_{3,4}^{(3)} G_{2,3}^{(2)}.$$

$$\begin{pmatrix} \bullet & \bullet & \bullet & \bullet & \bullet \\ & \bullet & \bullet & \bullet & \bullet \\ & 1 & \bullet & \bullet & \bullet \\ & & 2 & \bullet & \bullet \\ & & & 4 & \bullet \\ & & & 3 & 6 \\ & & & & 5 \end{pmatrix}$$

Algoritmo para MC2E mediante rotaciones de Givens

Algoritmo 11 Algoritmo $MC2E_G$

Entrada: $X \in \mathbb{R}^{d \times K}$, $Y \in \mathbb{R}^{d \times N}$, $B \in \mathbb{R}^{N \times N}$ y $\Gamma \in \mathbb{R}^{N \times K}$

Salida: $B \in \mathbb{R}^{N \times N}$, $\Gamma \in \mathbb{R}^{N \times K}$

- 1: Obtener Q y R (Householder) {coste $\rightarrow \frac{4}{3}K^2(3d - K)$ }
 - 2: $\tilde{Y} = Q^T Y$ {coste $\rightarrow 2NK(2d - K)$ }
 - 3: Crear la matriz $[R_1 | \tilde{Y}_1]$
 - 4: **Para** $i=1 \dots N$ **Hacer**
 - 5: Retriangularizar la matriz $[R_{i,1} | \tilde{Y}_{i,1}]$ {coste $\rightarrow C_i(\lambda_i, k_i, n_i)$ }
 - 6: $\tilde{y}_i = \tilde{Q}_i^T \tilde{Y}_i$ {aplicando las rotaciones de Givens calculadas en el paso anterior} {coste $\rightarrow 6 \left(\sum_{j=1}^{k_i} (\lambda_{i,j} - j) + \sum_{j=1}^{n_i-1} (K - j - k_i) \right)$ }
 - 7: $\hat{\beta}_i = R_{i,1}^{-1} \tilde{y}_{i,1}$ {coste $\rightarrow (n_i + k_i - 1)^2$ }
 - 8: Sustituir los valores de β_i en las incógnitas de B_i y Γ_i
 - 9: **Fin Para**
-

Comparativa entre $MC2E_H$ y $MC2E_G$

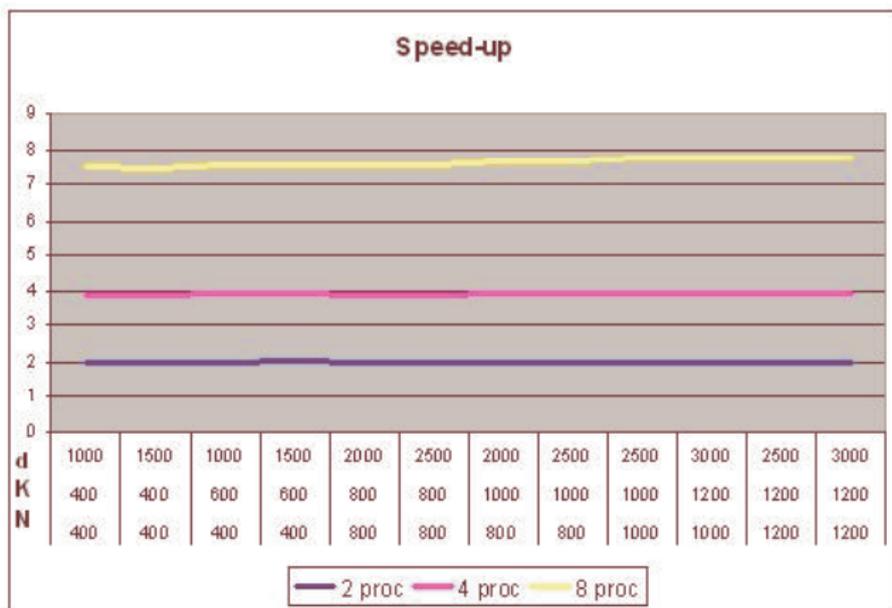
Tiempo de ejecución (en segundos) y ratio, cuando se varía el tamaño del problema. Se muestra el promedio (número grande) y la desviación típica (subíndice) de 5 medidas para el mismo tamaño del problema.

Tam. del problema			Tiempo	Tiempo	Ratio
N	K	d	Householder	Givens	
100	100	500	0.50 _{0.02}	0.15 _{0.00}	3.35 _{0.06}
100	100	1000	1.06 _{0.03}	0.16 _{0.00}	6.63 _{0.21}
100	200	500	1.39 _{0.06}	0.67 _{0.02}	2.09 _{0.09}
100	200	1000	3.23 _{0.11}	0.70 _{0.00}	4.60 _{0.13}
200	200	500	3.39 _{0.08}	1.90 _{0.03}	1.79 _{0.01}
200	200	1000	7.85 _{0.32}	1.94 _{0.05}	4.05 _{0.07}
200	400	500	10.25 _{0.14}	9.16 _{0.13}	1.12 _{0.02}
200	400	1000	23.28 _{0.20}	9.31 _{0.16}	2.50 _{0.05}
400	400	1000	56.60 _{2.47}	28.05 _{0.41}	2.02 _{0.07}
400	400	1500	98.85 _{2.84}	28.62 _{0.31}	3.45 _{0.08}
400	600	1000	108.68 _{3.94}	74.55 _{1.18}	1.46 _{0.03}
400	600	1500	200.76 _{5.57}	75.13 _{0.91}	2.67 _{0.05}
800	800	2000	1319.30 _{12.58}	438.14 _{2.24}	3.01 _{0.01}
800	800	2500	1889.01 _{20.66}	440.61 _{3.57}	4.29 _{0.02}
800	1000	2000	2067.94 _{16.62}	741.28 _{4.60}	2.79 _{0.01}
800	1000	2500	2894.90 _{28.64}	741.77 _{5.43}	3.90 _{0.02}

Paralelización de $MC2E_G$ **Algoritmo 12** Algoritmo $PMC2E_G$ **Entrada:** $X \in \mathbb{R}^{d \times K}$, $Y \in \mathbb{R}^{d \times N}$, $B \in \mathbb{R}^{N \times N}$ y $\Gamma \in \mathbb{R}^{N \times K}$ **Salida:** $B \in \mathbb{R}^{N \times N}$ y $\Gamma \in \mathbb{R}^{N \times K}$

- 1: Obtener Q y R (Householder de X en par.) {coste $\rightarrow \frac{4K^2}{3p}(3d - K)$ }
- 2: Calcular $\tilde{Y} = Q^T Y$ {Aplicar los reflectores de Householder en paralelo a la matriz} {coste $\rightarrow \frac{2NK}{p}(2d - K)$ }
- 3: Crear la matriz $[R_1 | \tilde{Y}_1]$
- 4: EN PARALELO cada procesador $q = 0, \dots, p - 1$ HACE
- 5: **Para** $j=1 \dots \frac{N}{p}$ **Hacer**
- 6: $i = q + (j - 1)p + 1$
- 7: Retriangularizar la matriz $[R_{i,1} | \tilde{Y}_{i,1}]$ {coste $\rightarrow C_i(\lambda_i, k_i, n_i)$ }
- 8: Calcular $\tilde{y}_i = \tilde{Q}_i^T \tilde{Y}_i$ {apl. las rot. de Givens anteriores} {coste $\rightarrow 6 \left(\sum_{j=1}^{k_i} (\lambda_{i,j} - j) + \sum_{j=1}^{n_i-1} (K - j - k_i) \right)$ }
- 9: Calcular $\hat{\beta}_i = R_{i,1}^{-1} \tilde{y}_{i,1}$ {coste $\rightarrow (n_i + k_i - 1)^2$ }
- 10: Sustituir los valores de β_i en las incógnitas de B_i y Γ_i
- 11: **Fin Para**
- 12: **FIN PARALELO**

Estudio experimental de $PMC2E_G$



Algoritmos QR con nodos ficticios

Reducir el coste de $MC2E_G$

Para reducir el número de operaciones en el algoritmo $MC2E_G$ tendremos en cuenta dos consideraciones:

- Resolver las ecuaciones en orden que reduzca el número de operaciones.
- Creación de matrices intermedias para reducir el número de retriangularizaciones a usar para hallar la descomposición QR en cada ecuación.

Se propone: Creación de un árbol cuyos nodos corresponden a conjuntos de variables.

- Las hojas corresponden a las ecuaciones.
- Los nodos intermedios son nodos ficticios.
- El tiempo de obtención del árbol más el tiempo de la resolución de las ecuaciones utilizando el árbol no sea superior al tiempo de resolver las ecuaciones retriangularizando directamente.
- **Algoritmo heurístico** para hallar el AMC basado en un conjunto de reglas.

Construcción del Árbol de Mínimo Coste (AMC) I

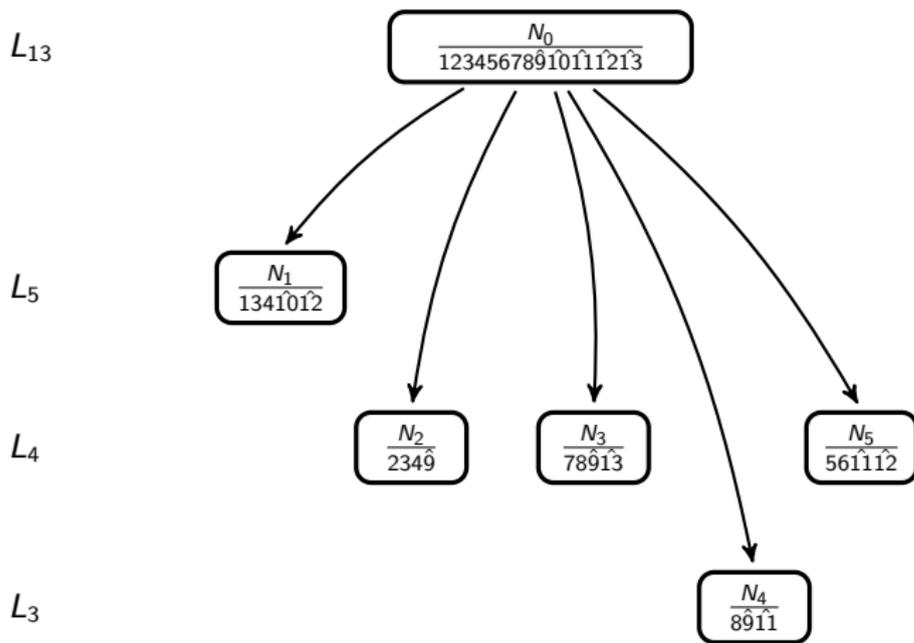
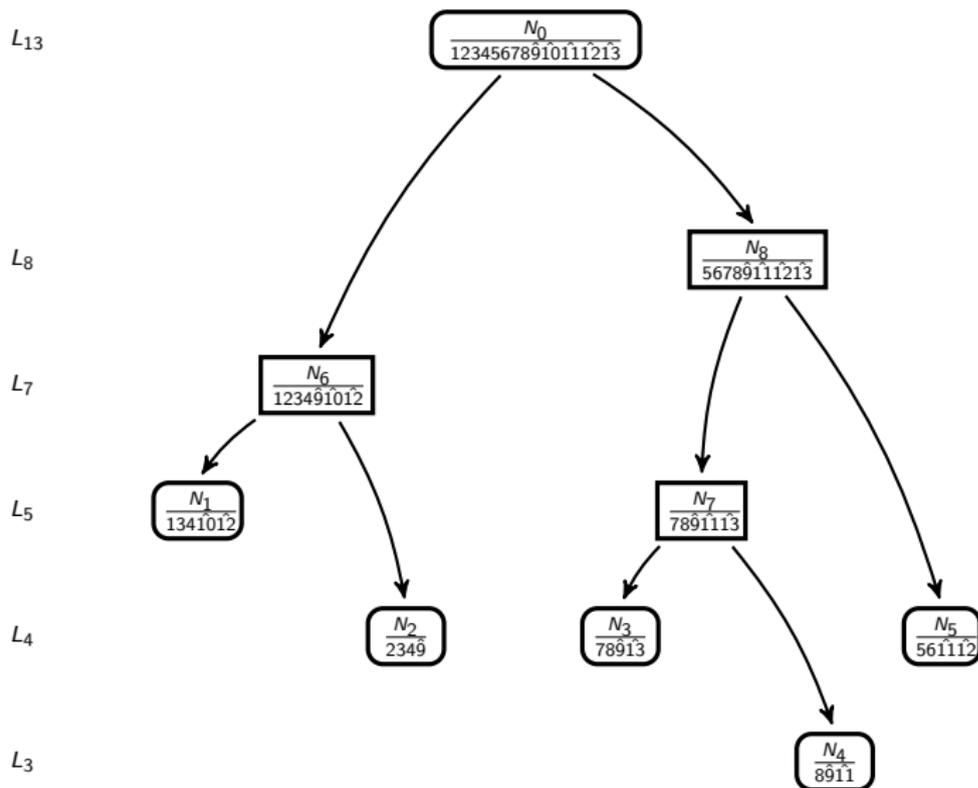


Figura: Árbol de resolución de un M.E.S. con $N=5$ y $K=8$ donde se representan los nodos ecuaciones y el nodo raíz por niveles (mostrados a la izquierda).

Construcción del Árbol de Mínimo Coste (AMC) II



Comparativa entre $MC2E_G$ y $MC2E_{NF}$ I

Tiempos de ejecución (en segundos) de los algoritmos $MC2E_G$, AMC y $MC2E_{NF}$, cuando se varía el tamaño del problema. Se muestran los ratios de resolución del sistema y del total de tiempos de los algoritmos.

			Tiempos			Ratio de los tiempos	
N	K	d	$MC2E_G$	AMC	$MC2E_{NF}$	$\frac{MC2E_G}{MC2E_{NF}}$	$\frac{MC2E_G}{AMC+MC2E_{NF}}$
400	400	1000	27.87	94.12	7.71	3.61	0.27
400	400	1500	28.03	94.46	7.99	3.51	0.27
400	600	1000	75.82	151.82	19.80	3.83	0.44
400	600	1500	76.40	148.98	18.73	4.08	0.46
800	800	2000	435.11	2204.85	102.39	4.25	0.19
800	800	2500	439.88	2205.01	103.70	4.24	0.19
800	1000	2000	741.75	3130.04	167.41	4.43	0.22
800	1000	2500	736.48	3148.18	170.98	4.31	0.22

Comparativa entre $MC2E_G$ y $MC2E_{NF}$ II

N	K	d	AMC	Resol. sis. $MC2E_{NF}$	Resol. sis. $MC2E_G$	Ratio de resol. sis.	Ratio Total
20	1000	2000	0.18 _{0.01}	1.20 _{0.18}	1.45 _{0.22}	1.22 _{0.12}	1.01 _{0.02}
20	2000	3000	0.63 _{0.06}	9.58 _{1.56}	12.32 _{2.15}	1.29 _{0.13}	1.06 _{0.03}
20	3000	4000	1.60 _{0.30}	39.13 _{2.85}	49.14 _{4.73}	1.26 _{0.06}	1.07 _{0.02}
20	4000	5000	2.72 _{0.40}	72.50 _{20.75}	114.95 _{17.57}	1.65 _{0.31}	1.16 _{0.05}
40	1000	2000	0.59 _{0.07}	2.65 _{0.42}	3.39 _{0.35}	1.29 _{0.10}	1.02 _{0.03}
40	2000	3000	2.19 _{0.43}	18.74 _{3.45}	23.99 _{2.87}	1.30 _{0.12}	1.06 _{0.04}
40	3000	4000	4.59 _{0.31}	57.66 _{15.39}	83.49 _{11.66}	1.49 _{0.21}	1.15 _{0.05}
40	4000	5000	9.07 _{2.05}	142.94 _{32.17}	229.08 _{22.03}	1.65 _{0.27}	1.24 _{0.08}
80	1000	2000	2.78 _{0.38}	5.24 _{0.72}	7.37 _{0.63}	1.42 _{0.14}	0.95 _{0.03}
80	2000	3000	10.04 _{2.66}	36.34 _{4.52}	51.93 _{4.63}	1.43 _{0.07}	1.08 _{0.05}
80	3000	4000	16.21 _{3.69}	119.47 _{26.79}	189.90 _{21.42}	1.63 _{0.24}	1.25 _{0.06}
80	4000	5000	34.11 _{6.74}	288.99 _{7.47}	466.51 _{17.56}	1.61 _{0.07}	1.29 _{0.03}
160	1000	2000	14.75 _{5.19}	11.89 _{1.09}	17.09 _{1.21}	1.44 _{0.07}	0.71 _{0.13}
160	2000	3000	38.13 _{12.11}	74.52 _{8.15}	110.15 _{10.60}	1.48 _{0.10}	0.99 _{0.10}
160	3000	4000	80.38 _{12.09}	233.54 _{2.83}	405.32 _{49.14}	1.74 _{0.20}	1.23 _{0.15}
160	4000	5000	136.56 _{27.46}	592.51 _{45.85}	1008.26 _{60.27}	1.71 _{0.18}	1.31 _{0.10}
320	1000	2000	162.46 _{10.94}	28.14 _{1.42}	42.87 _{3.36}	1.52 _{0.05}	0.24 _{0.01}
320	2000	3000	420.65 _{144.57}	172.06 _{9.90}	250.99 _{11.15}	1.46 _{0.05}	0.47 _{0.12}
320	3000	4000	556.28 _{59.27}	511.22 _{26.35}	892.16 _{76.46}	1.75 _{0.15}	0.85 _{0.09}
320	4000	5000	842.07 _{73.73}	1206.26 _{71.12}	2242.92 _{149.72}	1.87 _{0.18}	1.09 _{0.09}

Estudio experimental de $PMC2E_{NF}$ I

Tiempos de ejecución (en segundos) y speed-up correspondientes al algoritmo $PMC2E_{NF}$ sin el cálculo del AMC (no se contempla el tiempo del algoritmo AMC) para ambos tipos de recorrido del árbol, cuando se varía el tamaño del problema.

N	K	d	1 proc.	2 proc.	Sp	4 proc.	Sp	8 proc.	Sp
Recorrido en profundidad									
400	400	1000	7.71	4.01	1.92	2.37	3.25	1.61	4.79
400	600	1000	19.80	9.89	2.00	5.26	3.76	3.22	6.15
800	800	2000	102.39	52.19	1.96	28.98	3.53	15.60	6.56
800	1000	2000	167.41	87.80	1.91	46.16	3.63	22.99	7.28
Recorrido en amplitud									
400	400	1000	7.71	4.13	1.87	2.4	3.21	1.45	5.32
400	600	1000	18.18	9.33	1.95	5.44	3.34	3.33	5.46
800	800	2000	102.39	52.41	1.95	30.27	3.38	17.5	5.85
800	1000	2000	169.51	90.99	1.86	48.15	3.52	24.13	7.02

Estudio experimental de $PMC2E_{NF}$ II

Tiempos de ejecución (en segundos) y speed-up correspondientes al algoritmo $PMC2E_{NF}$ sin el cálculo del AMC (no se contempla el tiempo de la llamada al algoritmo AMC) para ambos tipos de recorrido del árbol, cuando se varía el tamaño del problema.

N	K	d	1 proc.	2 proc.	Sp	4 proc.	Sp	8 proc.	Sp
Recorrido en profundidad									
40	4000	5000	194.44	129.33	1.50	113.83	1.71	80.05	2.43
80	4000	5000	566.09	431.87	1.31	363.29	1.56	228.66	2.48
160	4000	5000	1329.29	779.17	1.71	521.41	2.55	350.01	3.80
320	4000	5000	2571.66	1482.43	1.73	976.61	2.63	599.32	4.29
Recorrido en amplitud									
40	4000	5000	194.44	128.51	1.51	110.10	1.77	77.80	2.50
80	4000	5000	566.09	328.96	1.72	259.97	2.18	189.38	2.99
160	4000	5000	1329.29	750.13	1.77	425.08	3.13	275.89	4.82
320	4000	5000	2510.03	1339.88	1.87	790.25	3.18	474.77	5.29

Obtención de M.E.S. mediante AG

Obtención de M.E.S. mediante AG

Tradicionalmente:

- Los Modelos de Ecuaciones Simultáneas han sido desarrollados por personas con una extensa experiencia en el problema particular que representa el modelo.
- Cuando el número de variables es muy grande o la relación entre algunas de ellas no es muy clara, la construcción del modelo se hace difícil y subjetiva

Idea: desarrollar un algoritmo el cual, una vez dadas las variables endógenas y exógenas, encuentre el mejor M.E.S. posible (acorde con un criterio de comparación de modelos).

Problema: El espacio de posibles soluciones es muy extenso (el número total de posibles soluciones es $2^{N(N+K)}$).

Solución: Algoritmos Genéticos.

Parámetros de criterio para la comparación de M.E.S.

Las expresiones de AIC y BIC son:

$$AIC = d \cdot \ln |\hat{\Sigma}_e| + 2 \sum_{i=1}^N (n_i + k_i - 1) + N(N + 1) \quad (1)$$

$$BIC = d \cdot \ln |\hat{\Sigma}_e| + \ln d \cdot \left(\sum_{i=1}^N (n_i + k_i - 1) + \frac{N}{2}(N + 1) \right) \quad (2)$$

donde $|\hat{\Sigma}_e|$ es el determinante de la matriz de covarianzas de los errores e_i , $i = 1, \dots, N$, donde e_i es la diferencia entre Y_i y la estimación de Y_i dada en la ecuación i .

Esquema general de un algoritmo genético

Algoritmo 13 Esquema de un algoritmo genético

- 1: *Inicializar*(S)
 - 2: **Mientras** *No CondicionesdeFin*(S) **Hacer**
 - 3: *Evaluar*(S) {función de fitness}
 - 4: $SS1 = \text{Seleccionar}$ los elementos mejores de S
 - 5: $SS2 = \text{Cruce}(SS1)$
 - 6: **Si** Hay que mutar **Entonces**
 - 7: Seleccionar aleatoriamente $c \in SS1$
 - 8: *Mutar*(c)
 - 9: Introducir c en $SS1$
 - 10: **Fin si**
 - 11: $S = SS1 \cup SS2$
 - 12: **Fin Mientras**
-

Definición de un cromosoma

- Un cromosoma se representa por una matriz con N filas y $N + K$ columnas.
- En cada fila se representa una ecuación mediante unos y ceros.
- Si la variable j aparece en la ecuación i , el valor para la posición (i, j) en el cromosoma es uno, y cero en caso contrario.
- Las primeras N columnas del cromosoma representan las variables endógenas y las otras K las variables exógenas.

$$Y_1 = \beta_{1,2} Y_2 + \gamma_{1,1} X_1 + \gamma_{1,2} X_2 + u_1$$

$$Y_2 = \beta_{2,1} Y_1 + \gamma_{2,3} X_3 + u_2$$

es representado por el cromosoma:

11110
11001

donde las primeras dos columnas corresponden a las variables endógenas, y las otras tres columnas a las exógenas.

Cromosoma válido

Las condiciones necesarias para que un cromosoma sea válido son:

- C1: El modelo tiene que tener al menos una ecuación.
- C2: Si el elemento (i, i) es cero, la columna i -ésima debe tener todo ceros. Esto significa que cuando la ecuación i -ésima no está en el sistema la variable i -ésima no formará parte de otras ecuaciones.
- C3: Cada ecuación en el modelo debe tener al menos dos variables, es decir, si el elemento (i, i) es uno, entonces existe j con $1 \leq j \leq N + K$ y $j \neq i$ tal que el elemento (i, j) es uno.
- C4 (Condición de Rango): La ecuación i -ésima está identificada sí y solo sí es posible encontrar una matriz de dimensión $(N - 1) \times (N - 1)$ de rango completo cuyos elementos son los coeficientes de la forma estructural $\beta_{1,2}, \beta_{1,3}, \dots, \beta_{N,N-1}, \gamma_{1,1}, \dots, \gamma_{N,K}$ que no aparecen en la ecuación i -ésima.

Cruce de cromosomas

- Cruce por elementos (*Cruce1*). Es el cruce básico utilizado en los algoritmos genéticos.
- Cruce por ecuaciones (*Cruce2*). Aporta soluciones a la población que son combinación de ecuaciones de modelos que ya están en la población.
- Cruce dentro de una ecuación (*Cruce3*). Genera dos descendientes que son casi igual que los ascendientes salvo en una sola ecuación.

El siguiente ejemplo muestra los tres métodos de cruce (para $N = 3$ y $K = 5$). Los valores de e , v_1 y v_2 se han tomado aleatoriamente.

Ascendente		Cruce1 $e = 10$	
<i>asc1</i>	<i>asc2</i>	<i>desc1</i>	<i>desc2</i>
11110110	10100100	11110110	10100110
11110101	01110100	11110100	01110101
01110110	11110110	11110110	01110110
Cruce2 $e = 1$		Cruce3 $e = 2, v_1 = 2, v_2 = 3$	
<i>desc1</i>	<i>desc2</i>	<i>desc1</i>	<i>desc2</i>
11110110	10100100	11110110	10100100
01110100	11110101	11110100	01110101
11110110	01110110	01110110	11110110

Mutación

Invertir un elemento del cromosoma elegido aleatoriamente.

Problema: Un cromosoma mutado puede no sobrevivir lo suficiente para tener descendencia (aún estando cerca del óptimo).

Idea: Explorar la vecindad de cada cromosoma mutado e insertar en la población el mejor.

Solución: Avance Rápido.

Avance Rápido

Algoritmo 14 Algoritmo AR

Entrada: Un cromosoma c de la población

Salida: El cromosoma c mutado

```

1: Mientras Número de ecuaciones generadas  $\leq$  NEG Hacer
2:   Generar  $1 \leq e \leq N$  y  $1 \leq v \leq N + K$  aleatoriamente
3:    $c_1 = \text{Mutar}(c)$  {Invertir el elemento  $(e, v)$  del cromosoma  $c$ }
4:    $c_{\text{mejor}} = c_1$  y Condición de Fin=Falso
5:   Mientras Condición de Fin=Falso Hacer
6:     Para  $v=1 \dots N + K$  Hacer
7:        $c_2 = \text{Mutar}(c_1)$  {Invertir el elemento  $(e, v)$  del cromosoma  $c_1$ }
8:       Si  $c_2$  es válido Y Evaluación( $c_2$ )  $<$  Evaluación( $c_{\text{mejor}}$ ) Entonces
9:          $c_{\text{mejor}} = c_2$ 
10:      Fin si
11:     Fin Para
12:     Si  $c_{\text{mejor}} = c_1$  Entonces
13:       Condición de Fin=Verdadero
14:     Si no
15:        $c_1 = c_{\text{mejor}}$ 
16:     Fin si
17:   Fin Mientras
18: Fin Mientras

```

Estudio experimental

Comparación entre tres métodos diferentes de cruce variando el tamaño del problema. En cada caso, t es el tiempo de ejecución (en segundos), $iter$ es el número de iteraciones que el algoritmo necesita para finalizar, y AIC es el valor (por 10^{-3}) de la evaluación del mejor cromosoma encontrado. El tamaño de la población es 100 y $MaxBest=15$.

N	K	d	Cruce1			Cruce2			Cruce3		
			t	$iter$	AIC	t	$iter$	AIC	t	$iter$	AIC
10	15	50	3.03	48	2683	5.11	97	2732	0.67	20	2833
15	20	50	8.00	62	4548	6.73	53	4540	1.94	40	4709
30	40	100	58.33	50	21937	87.54	72	22120	9.47	17	22765
40	50	100	325.87	111	30956	294.19	102	31262	64.41	24	32975

Estudio experimental

Comparación de las soluciones encontradas por el algoritmo genético basado en AIC o BIC. El cruce utilizado es el *cruce3*, $d = 100$, $MaxBest=15$ y $MaxIter=150$. $Error_{AIC}$ y $Error_{BIC}$ son los errores de la solución utilizando AIC y BIC, respectivamente, en la función de evaluación.

N	K	σ	$PopSize = 100$		$PopSize = 500$	
			$Error_{AIC}$	$Error_{BIC}$	$Error_{AIC}$	$Error_{BIC}$
30	40	0	1.47 _{0.72}	1.24 _{0.65}	1.31 _{0.31}	1.33 _{0.54}
30	40	0.01	1.17 _{0.32}	0.99 _{0.39}	0.88 _{0.28}	0.87 _{0.36}
30	40	0.1	1.06 _{0.32}	0.92 _{0.42}	0.91 _{0.35}	0.95 _{0.31}
40	50	0	2.29 _{0.52}	2.01 _{0.43}	2.29 _{0.64}	2.28 _{0.78}
40	50	0.01	1.64 _{0.46}	1.58 _{0.40}	1.59 _{0.49}	1.62 _{0.27}
40	50	0.1	1.64 _{0.37}	1.54 _{0.34}	1.56 _{0.38}	1.31 _{0.19}

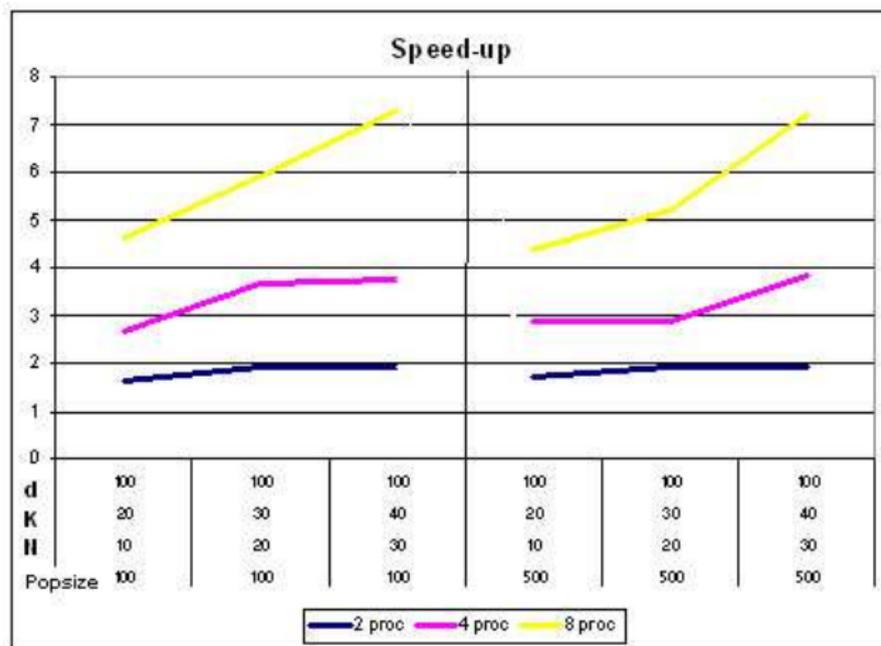
Estudio experimental

Comparación entre la solución encontrada por el algoritmo genético usando y sin usar avance rápido en la mutación. $PopSize=100$, $d=100$, el cruce usado es *cruce3*, $MaxBest=30$ y $MaxIter=150$

Modo	NEG	$N = 10, K = 20$		$N = 20, K = 30$	
		AIC	t	AIC	t
sin AR	-	2138.93 _{495.92}	5.10 _{1.41}	4658.06 _{722.33}	15.41 _{5.20}
con AR	1	2143.54 _{399.53}	9.79 _{3.92}	4710.53 _{580.39}	49.14 _{21.72}
	$\lfloor \frac{N}{4} \rfloor$	1491.13 _{488.72}	12.62 _{2.54}	3072.98 _{1135.88}	102.23 _{45.53}
	$\lfloor \frac{N}{2} \rfloor$	-680.61 _{642.75}	27.48 _{8.89}	811.65 _{784.60}	227.35 _{83.93}
	N	-3586.46 _{453.69}	34.17 _{20.74}	-4920.00 _{466.22}	449.78 _{165.68}

Estudio experimental

Speed-up del algoritmo híbrido en memoria compartida. El cruce usado es *cruce3*, $NEG = \lfloor \frac{N}{2} \rfloor$ y $MaxBest=MaxIter=150$.



Aplicaciones

Aplicaciones

Se utilizan los algoritmos desarrollados en el capítulo anterior para dos estudios reales:

- Encontrar un M.E.S. a partir de los datos de las variables que aparecen en uno de los modelos clásicos de econometría, el Modelo Keynesiano Simple.
- Obtener un M.E.S. eficiente para modelar un conjunto de variables que son usadas en la predicción de preeclampsia.

Conclusiones y Trabajos futuros

Conclusiones

- Se han desarrollado algoritmos para los estimadores MCI y MC2E.
- Se han mejorado los algoritmos utilizando descomposiciones matriciales (descomposición de la inversa y QR).
- El algoritmo $MC2E_G$ que utiliza rotaciones de Givens mejora a $MC2E_H$ basado únicamente en reflexiones de Householder.
- Se ha desarrollado un algoritmo que introduce nodos ficticios y que en determinados problemas mejora a $MC2E_G$.
- Se han desarrollado versiones en paralelo en memoria compartida y distribuida obteniéndose speed-ups satisfactorios.
- Se ha desarrollado un algoritmo híbrido de obtención de M.E.S. a partir de los datos de las variables utilizando algoritmos genéticos y avance rápido.
- Se han utilizado los algoritmos desarrollados en dos problemas reales.

Trabajos Futuros

- **Resolución de M.E.S.** Desarrollar versiones en paralelo tanto en memoria compartida como distribuida de todos los algoritmos secuenciales, también versiones híbridas y en sistemas heterogéneos.
- **Árbol de Mínimo Coste.** Mejora del algoritmo heurístico que obtiene dicho árbol.
- **Metaheurísticas.** Desarrollar algoritmos utilizando otras metaheurísticas (Scatter Search, GRASP, Búsqueda tabú, etc.) para su comparación tanto en el tiempo de ejecución como en la bondad de la solución encontrada.
- **Búsqueda exhaustiva.** Estudiar la posibilidad de desarrollar algoritmos de búsqueda exhaustiva encontrando propiedades que reduzcan el espacio de búsqueda, o añadiendo restricciones dadas por los expertos.
- **Modelos de Ecuaciones Simultáneas No Lineales.** Estudio de algoritmos paralelos para este tipo de modelos de la misma forma que se ha hecho para los M.E.S. lineales.
- **Otras técnicas econométricas.** Estudio de algoritmos paralelos para otras técnicas econométricas como, por ejemplo, los modelos VAR, Modelos de Datos Panel, etc.