

DESEMPEÑO COMPUTACIONAL DE ESTRATEGIAS HÍBRIDAS PARA LA SOLUCIÓN DE PROBLEMAS CUADRÁTICOS NO CONVEXOS CON RESTRICCIONES DE CAJA

Ridelio Miranda Pérez*, Sira Allende Alonso**, Boris Pérez Cañedo*, Gemayqzel Bouza Allende**

*Universidad de Cienfuegos, Cuba.

**Universidad de la Habana, Cuba.

ABSTRACT

In this paper the minimization of a non-convex quadratic function under box constraints is considered. For solving this problem, we propose a hybrid strategy combining a cut and continuation method with a Branch and Bound procedure, based on double non-negative relaxation. We compare the results of the new approach with the Branch and Bound method and with the preceding hybrid algorithm. The influence of characteristics of the problem in the algorithm's behavior is explored. Computational experiments include problems of higher dimension than the ones reported in the referred literature and show that the new strategy improves the results concern to number of explored nodes and computation time.

KEYWORDS: Non-convex quadratic programming, Parametric optimization, Branch-and-Bound, Double non negative relaxation, Semidefinite programming.

MSC: 90C20, 90C26.

RESUMEN

Se considera el problema de minimizar una función cuadrática no convexa sujeta a restricciones de la caja. Para resolver este problema proponemos una estrategia híbrida que combina un método de corte y continuación con un procedimiento de ramificación y acotación, basado en relajación doblemente no negativa. Comparamos el desempeño del nuevo enfoque con el método de ramificación y acotación y con el algoritmo híbrido precedente. Se investiga además cómo influyen las características del problema en el comportamiento del algoritmo. Los experimentos computacionales incluyen problemas de dimensión superior a los reportados en la literatura referenciada y muestran que la nueva estrategia mejora los resultados relativos al número de nodos explorados y al tiempo de cómputo.

PALABRAS CLAVE: programación cuadrática no convexa, optimización paramétrica, ramificación y acotación, relajación no negativa doble, programación semi-definida.

1. INTRODUCCIÓN

Consideramos el problema de programación cuadrática con restricciones de caja definido por

$$[QPB] \quad f_{minglob} = \min\{f(x) = \frac{1}{2}x^T Qx + q^T x \mid x \in B\} \quad (1)$$

$$B = \{x \in \mathbb{R}^n \mid 0 \leq x \leq e\} \quad (2)$$

donde Q denota una matriz simétrica de orden (n) ($Q \in S^n$) no-semidefinida positiva ($Q \not\geq 0$), $q \in \mathbb{R}^n$ el vector de los coeficientes lineales, y $e = [1; 1; \dots; 1]^T \in \mathbb{R}^n$ el vector unitario.

*rmiranda@ucf.edu.cu

QPB es uno de los problemas NP-duros [12] más simples de la programación no lineal y en consecuencia ha llamado la atención de muchos investigadores. Las técnicas de optimización global implementadas que mejores resultados reportan para su solución son aquellas que utilizan algún esquema de ramificación y acotación (*B&B*). Los aportes más significativos en este campo corresponden a [2], [6] y [13]. En estos trabajos se implementa un procedimiento de ramificación finito basado en el cumplimiento forzado de las condiciones KKT de complementariedad. La diferencia fundamental entre ellos radica en el tipo de relajación empleada para el cálculo de las cotas inferiores.

Por otra parte, los resultados reportados en [9] apoyan la aplicación del enfoque basado en corte y continuación (*C&C*) propuesto en [7]. En esencia, este procedimiento consiste en la ejecución de un proceso iterativo, que incluye en cada iteración: el cálculo de un mínimo local x^* de [QPB], y el empleo de un método de continuación y saltos, para determinar una nueva solución factible $\bar{x} \in B$ con $f(\bar{x}) \leq f(x^*) - \varepsilon$, ($\varepsilon > 0$ suficientemente pequeño).

En un trabajo más reciente [10] se propone combinar el método de corte y continuación con el método de ramificación y acotación de Burer-Vandenbussche [2] para obtener así un algoritmo híbrido más eficiente. Este nuevo procedimiento permite la determinación de cotas superiores más fuertes, al tiempo que evita el tratamiento de las singularidades difíciles detectadas por *C&C*. La experimentación numérica realizada muestra que el nuevo algoritmo supera con creces los resultados alcanzados por los dos algoritmos que le dan origen.

En este trabajo se presenta una variación del algoritmo híbrido citado, considerando aquí el empleo del método *B&B* de Burer-Chen [6] en lugar del utilizado en [10]. Con vistas a examinar el efecto que provoca este cambio en el desempeño del algoritmo se conduce luego un conjunto de experimentos computacionales, en los que se comparan los resultados del mismo bajo una y otra estrategia de ramificación-acotación (entre sí) y con el método *B&B* de Burer-Chen.

El artículo está organizado en la siguiente forma. En la sección 2 se presentan formulaciones equivalentes para [QPB], a partir de las cuales se discuten las características del método de ramificación y acotación seleccionado para el algoritmo. En la sección 3 se resumen los aspectos más significativos del método de corte y continuación, y se formaliza la nueva versión del algoritmo híbrido propuesto. La sección 4 se dedica al diseño de los experimentos computacionales y el análisis de los resultados experimentales.

2. REFORMULACIÓN DE QPB. EL MÉTODO DE RAMIFICACIÓN Y ACOTACIÓN DE BURER-CHEN

En la literatura consultada se recogen distintas reformulaciones del problema cuadrático con restricciones de caja. En muchas de ellas se sacrifica la estructura convexa del conjunto factible original con el fin de linealizar la función objetivo. Este es el caso de la formulación equivalente (3-4) propuesta por Burer [3]. Esta reformulación desempeñan un papel determinante en la construcción del algoritmo que aquí se propone.

$$[QPB - BurChen] \quad \min \left\{ \frac{1}{2} \begin{pmatrix} 0 & \tilde{q}^T \\ \tilde{q} & \tilde{Q} \end{pmatrix} \bullet \begin{pmatrix} 1 & \tilde{x}^T \\ \tilde{x} & \tilde{X} \end{pmatrix} \mid (\tilde{x}, \tilde{X}) \in \tilde{B} \right\} \quad (3)$$

$$\tilde{B} = \left\{ (\tilde{x}, \tilde{X}) \in \mathbb{R}^{\tilde{n}} \mid \begin{array}{l} \tilde{A}\tilde{x} = \tilde{b}, \quad \text{diag}(\tilde{A}\tilde{X}\tilde{A}^T) = \tilde{b} \circ \tilde{b}, \\ \tilde{X}_{\tilde{E}} = 0, \\ \begin{pmatrix} 1 & \tilde{x}^T \\ \tilde{x} & \tilde{X} \end{pmatrix} \in \mathcal{C} \end{array} \right\} \quad (4)$$

Aquí las variables (\tilde{x}, \tilde{X}) se definen como: $\tilde{x} = (x \ z \ y \ s)^T \in \mathbb{R}^{4n}$, y $\tilde{X} = \tilde{x}\tilde{x}^T$, donde $z, y \in \mathbb{R}^n$ son los multiplicadores de Lagrange correspondientes a las restricciones ($x \geq 0$) y ($x \leq e$) respectivamente, y $s \in \mathbb{R}^n$ las variables de holgura correspondientes a ($x \leq e$) en (1-2).

Además $\tilde{Q} = \begin{pmatrix} Q & 0 \\ 0 & 0 \end{pmatrix} \in S^{4n \times 4n}$, $\tilde{q} = \begin{pmatrix} q \\ 0 \end{pmatrix} \in \mathbb{R}^{4n}$, $\tilde{b} = \begin{pmatrix} -q \\ e \end{pmatrix} \in \mathbb{R}^{2n}$, $\tilde{A} = \begin{pmatrix} Q & -I & I & 0 \\ I & 0 & 0 & I \end{pmatrix} \in \mathbb{R}^{2n \times 4n}$,

$\tilde{E} = \{(i, n+i) : i = 1, \dots, 3n\}$, y $\tilde{n} = 4n + 2n \times (4n + 1)$.

La multiplicación $\tilde{b} \circ \tilde{b}$ se refiere al producto de Hadamard del vector \tilde{b} por sí mismo, y el producto interno de

matrices se calcula según $\tilde{Q} \bullet \tilde{X} = \text{Trace}(\tilde{Q}^T \tilde{X}) = \sum_i \sum_j \tilde{Q}_{ij} \tilde{X}_{ij}$.

Finalmente, el cono \mathcal{C} de matrices completamente positivas se define como

$$\mathcal{C} = \{M \in S^n \mid \exists N \in \mathbb{R}^{n \times k} \text{ con } k > 0 \text{ y } N_{ij} \geq 0 \ (\forall i, j) \text{ tal que } M = NN^T\} \quad (5)$$

Otra formulación equivalente a [QPB], ampliamente referenciada, es la propuesta por Burer y Vandembussche [2].

$$[QP B - BurVan] \quad \min \left\{ \frac{1}{2} Q \bullet X + q^T x \mid (x, y, z, X) \in \bar{B} \right\} \quad (6)$$

$$\bar{B} = \left\{ (x, y, z, X) \in \mathbb{R}^{n \times (n+3)} \mid \begin{array}{l} 0 \leq x \leq e, \quad X = xx^T, \\ Qx + q + y - z = 0, \quad y, z \geq 0, \\ \frac{1}{2} Q \bullet X + q^T x = \frac{1}{2} (q^T x - e^T y) \\ y \circ (e - x) = 0, \quad z \circ x = 0 \end{array} \right\} \quad (7)$$

La reformulación de [QP B] como un problema de optimización sobre conos de matrices facilita la obtención de problemas relajados convexos, sencillos de resolver, que pueden ser utilizados luego en un esquema de ramificación y acotación, como sucede con los algoritmos $B\&B$ de Burer-Chen y de Burer-Vandembussche.

En el primer caso, dado que el problema de programación completamente positiva (3-4) es NP-Hard [11], y conociendo que la condición $\begin{pmatrix} 1 & \tilde{x}^T \\ \tilde{x} & \tilde{X} \end{pmatrix} \in \mathcal{C}$ implica que dicha matriz es doblemente no negativa, se propone reemplazar \mathcal{C} por el cono \mathcal{D} de matrices doblemente no negativas, definido como

$$\mathcal{D} = \{M \in S^n \mid M \succeq 0 \text{ y } M_{ij} \geq 0 \ (\forall i, j)\} \quad (8)$$

para obtener el siguiente problema relajado.

$$[RDNN] \quad \min \left\{ \frac{1}{2} \begin{pmatrix} 0 & \tilde{q}^T \\ \tilde{q} & \tilde{Q} \end{pmatrix} \bullet \begin{pmatrix} 1 & \tilde{x}^T \\ \tilde{x} & \tilde{X} \end{pmatrix} \mid (\tilde{x}, \tilde{X}) \in \tilde{B}_R \right\} \quad (9)$$

$$\tilde{B}_R = \left\{ (\tilde{x}, \tilde{X}) \in \mathbb{R}^{4n+2n \times (4n+1)} \mid \begin{array}{l} \tilde{A}\tilde{x} = \tilde{b}, \quad \text{diag}(\tilde{A}\tilde{X}\tilde{A}^T) = \tilde{b} \circ \tilde{b}, \\ \begin{pmatrix} 1 & \tilde{x}^T \\ \tilde{x} & \tilde{X} \end{pmatrix} \in \mathcal{D} \end{array} \right\} \quad (10)$$

Como [RDNN] tiene solución en tiempo polinomial, en [6] se propone emplear esta formulación para el cálculo de las cotas inferiores dentro del esquema $B\&B$ que se describe a continuación.

1. Solución del problema relajado [RDNN] para obtener las cotas inferiores (f_{LB}) de $f_{minglob}$ en cada subproblema.
2. Evaluación de $f(x)$ en un mínimo local de [QP B] para el cálculo de cotas superiores (f_{UB}), y eliminación de aquellos subproblemas con $f_{LB} \geq f_{UB}^*$ (donde $f_{UB}^* = \min\{f_{UB}\}$).
3. Selección del subproblema con menor cota inferior calculada como el nodo a resolver en la próxima iteración (*Criterio MCI*).
4. Estrategia de ramificación:
 - (a) Selección del índice (j) para el cual la solución (\tilde{x}, \tilde{X}) de (9-10) realiza la mayor violación de las condiciones KKT de complementariedad $\tilde{X}_{\tilde{E}} = 0$ (relajadas en 9-10).
Esto es: $j = \text{argmax}_{(i)} \{\max\{x_i * z_i; \quad z_i * y_i; \quad y_i * s_i\}\}$ (*Criterio SKKT*), y
 - (b) Construcción de dos nuevos subproblemas, forzando en cada nueva subregión factible el cumplimiento de las condiciones KKT para las variables con índice j (seleccionado en (a)), y conservando en ambos la misma función objetivo y el resto de las restricciones (*Regla PKKT*).

Utilizando este mismo esquema, en [2] se emplea una relajación semidefinida del problema (6-7) para el cálculo de las cotas inferiores (ver [10] para más detalles).

Tomando en cuenta que los resultados experimentales publicados en [6] mejoran notablemente los alcanzados en [2], en este trabajo se propone modificar el algoritmo híbrido presentado en [10] empleando aquí el método $B\&B$ de Burer-Chen en lugar del método $B\&B$ de Burer-Vandembussche utilizado en la versión original.

3. DESCRIPCIÓN DEL ALGORITMO HÍBRIDO

De forma análoga a [10], la nueva versión del algoritmo híbrido que se propone en este trabajo consiste en combinar un método de corte y continuación con un método de ramificación y acotación. El primero de ellos (*C&C*) consiste de dos pasos fundamentales:

Paso-1: Partiendo de una solución factible $x^0 \in B$ calcular un mínimo local x^* de [QPB].

Paso-2: Determinar una nueva solución factible $\bar{x} \in B$ tal que $f(\bar{x}) \leq f(x^*) - \varepsilon$, (con $\varepsilon > 0$ suficientemente pequeño), y regresar al paso-1 tomando $x^0 = \bar{x}$.

El segundo paso de este esquema introduce un corte cuadrático no convexo en la caja B que deja fuera del nuevo conjunto factible a todas las soluciones $x \in B$ con valores $f(x) > f(x^*) - \varepsilon$. En [9] se propone resolver este nuevo problema empleando un método de inmersión, cuya realización está estrechamente vinculada a la solución del siguiente problema de optimización paramétrica

$$[P_R(t)] : \min \left\{ F(x, w, t) = \frac{1}{2}[x - x_0]^T D[x - x_0] + d^T x + \frac{1}{2}[w - w_0]^2 + cw \right\} \quad (11)$$

Sujeto a :

$$G_j(x, w, t) = -x_j \leq 0, \quad j = 1, \dots, n$$

$$G_{n+j}(x, w, t) = x_j - 1 \leq 0, \quad j = 1, \dots, n$$

$$G_{2n+1}(x, w, t) = t(\frac{1}{2}x^T Qx + q^T x - q_0) + (1 - t)(w - w_0) \leq 0$$

$$G_{2n+2}(x, w, t) = -w \leq 0$$

– donde $D \in S^n$, $d \in \mathbb{R}^n$, $c \in \mathbb{R}$, $q_0 = f(x^*) - \varepsilon$, $x_0 \in B$, $w_0 \in \mathbb{R}_+$ y $0 \leq t \leq 1$.

Obsérvese que la componente (x) de cualquier solución de (11) con $t \geq 1$ ó $w > w_0$, es, a su vez, una solución para el paso-2 del método de corte y continuación. Un procedimiento para obtener una solución con estas características, es el método de continuación [1]. Este algoritmo presupone la aproximación de una curva de puntos críticos generalizados (componente conexa) del problema paramétrico. La existencia de tales curvas está garantizada siempre que la matriz Jacobiana del sistema de puntos críticos correspondiente sea no singular. Las soluciones del sistema que no cumplan esta exigencia se llaman soluciones singulares. Dado que el problema (11) pertenece a la clase \mathcal{F} de Jongen-Jonker-Twilt [9], en él sólo pueden aparecer 4 tipos de singularidades, todas bien caracterizadas en [8]. En [7] se propone el tratamiento de estas singularidades con la aplicación de un método de salto a otra componente conexa de $[P_R(t)]$. Información más detallada sobre la programación paramétrica y el método de corte y continuación aquí empleado puede encontrarse en [7], [9] y [10].

La experiencia práctica ha demostrado que las soluciones obtenidas por *C&C* constituyen excelentes cotas superiores (f_{UB}) para $f_{minglob}$. Utilizando entonces las cotas inferiores (f_{LB}) que se obtienen como parte del método *B&B*, se puede definir el *gab relativo* :

$$gap = \left| \frac{f_{UB}^* - f_{LB}^*}{f_{UB}^*} \right| \quad (12)$$

como instrumento para medir la calidad de las soluciones calculadas en cada iteración.

De esta forma, si el valor $f_{UB}^* = f(x^*)$ del mejor mínimo local x^* encontrado por *C&C* es igual al valor mínimo f_{LB}^* de las cotas inferiores f_{LB} correspondientes a los nodos no eliminados por el método *B&B*, entonces la solución $(x^*, f(x^*))$ es un mínimo global del problema [QPB].

De forma alternativa, se puede plantear un objetivo menos exigente, en que sólo se exija determinar una solución ε -aproximada para $f_{minglob}$. Un criterio de parada apropiado para este objetivo resulta entonces la condición ($gap < \varepsilon$), donde $\varepsilon > 0$ es la tolerancia permitida.

A diferencia del método *B&B* de Burer-Chen, en el algoritmo híbrido que aquí se propone, se asumirá el criterio de selección del índice (j) de las variables a particionar introducido en [10].

[Regla-RSD] (Selección basada en los elementos de la Diagonal de la matriz Q):

Seleccionar entre las variables que violan las condiciones KKT de complementariedad $\tilde{X}_{\bar{E}} = 0$ aquellas cuyo

índice (j) se corresponda con el menor elemento diagonal Q_{jj} de la matriz Q .

A continuación se formula el pseudocódigo del algoritmo híbrido que aquí se propone.

Inicio: $k=1$, $\underline{B} = B_k = B$, fijar $x^{(k)} \in B$ y $\varepsilon > 0$.

Resolver problema relajado $[RDNN]$, y poner f_{LB}^* igual al valor mínimo calculado.

Paso-1: Calcular mínimo local x^* de (1) sobre la caja \underline{B} partiendo de $x^{(k)}$.

Hacer $f_{UB}^{(k)} = f(x^*)$ y $f_{UB}^* = \min\{f_{UB}^{(j)}, j = 1, \dots, k\}$

Si $f_{UB}^* = f_{LB}^*$, FIN del procedimiento. El punto x^* es un mínimo global de [QPB].

(Si $|\frac{f_{UB}^* - f_{LB}^*}{f_{UB}^*}| \leq \varepsilon$, x^* solución ε - aproximada para [QPB])

Paso-2: Determinar una nueva solución factible $\bar{x} \in \underline{B}$ con $f(\bar{x}) \leq f_{UB}^* - \varepsilon$.

Para ello se propone la ejecución de los pasos 2.1 a 2.3

Paso-2.1: Utilizar método de continuación para resolver el problema $[P_R(t)]$ en la caja \underline{B} . Si se alcanza una solución $\bar{x}^{(k)}$ para $[P_R(t)]$ con $t \geq 1$ ó $w \geq w_0$, poner $x^{(k)} = \bar{x}^{(k)}$ y regresar al Paso-1. En caso contrario seguir al paso 2.2.

Paso-2.2: Particionar el conjunto \underline{B} en dos subconjuntos, aplicando la estrategia de ramificación (RSD) para la selección de la variable y la regla ($PKKT$) para su construcción.

Supongamos que \underline{B} corresponde a la subcaja B_m de la partición, ($m \in \{1, \dots, k\}$)

Como resultado del paso de ramificación se obtienen dos nuevas subcajas, indexadas como B_m y B_{k+1} . Continuar al Paso-2.3.

Paso-2.3: Resolver el problema relajado $[RDNN]$ sobre las cajas B_m y B_{k+1} para obtener las cotas inferiores $f_{LB}^{(m)}$ y $f_{LB}^{(k+1)}$. Denotar con \underline{x}^j ($j=m, k+1$) a la componente x de estas soluciones. Eliminar de la búsqueda aquellas subcajas B_j donde $f_{LB}^{(j)} \geq f_{UB}^*$.

Actualizar $f_{LB}^* = \min\{f_{LB}^{(j)}, j \in \{1, \dots, k+1\}\}$.

Determinar índice j^* con $f_{LB}^{(j^*)} = f_{LB}^*$ y seleccionar subcaja $\underline{B} = B_{j^*}$ (*Criterio MCI*).

Hacer $k=k+1$. Poner $x^{(k)} = \underline{x}^{j^*} \in B_{j^*}$, y regresar al Paso-1.

Como en [10], aquí se mantiene la estrategia de evadir el tratamiento de las singularidades (tipo-3,4 ó 5) que puedan aparecer durante la aplicación del método de continuación (paso 2.1), utilizando para ello una iteración del método $B\&B$ de Burer-Chen.

El algoritmo híbrido descrito anteriormente a sido codificado sobre MatLab para obtener el programa (CC+BB-BurChen.m), compuesto por: el método de corte y continuación ($C\&C$) codificado como (CUTandPAFO.m) y el método $B\&B$ de Burer-Chen (quadprogbb.m) elaborado por Burer y Chen en [6], y disponible en <http://dollar.biz.uiowa.edu/~sburer>, con las modificaciones de acotación y ramificación expuestas en este epígrafe. En la siguiente sección nos referiremos al programa (CC+BB-BurVan.m) para identificar la implementación del algoritmo híbrido que utiliza el método $B\&B$ de Burer-Vandenbussche desarrollado en [10].

4. RESULTADOS EXPERIMENTALES

En esta sección se presentan los resultados de la experimentación computacional realizada para medir el rendimiento de la nueva versión del algoritmo híbrido, tomando en cuenta los siguientes objetivos específicos:

- Comparar el rendimiento de la nueva versión del algoritmo híbrido (CC+BB-BurChen) con la versión anterior (CC+BB-BurVan) y con el método $B\&B$ de Burer-Chen (quadprogbb).
- Analizar la influencia que tienen las características del problema en el rendimiento del algoritmo propuesto.
- Evaluar el comportamiento del nuevo algoritmo en problemas de dimensión superior a las 100 variables.

Los parámetros que se tomarán en cuenta para evaluar el rendimiento de los algoritmos son:

- Calidad de las soluciones alcanzadas.

- Número total de nodos creados (TNC).
- Número total de iteraciones (NTI) (igual al número de nodos explorados).
- Tiempo total Requerido (TTR).
- Iteración en que se determina el mínimo global (ITO).

Todos los experimentos realizados con este fin fueron corridos en una computadora VIT con procesador *core i3* a 2.4 G y 2 GB de memoria RAM.

El epígrafe consta de dos subsecciones. En la primera se describe la base de problemas pruebas utilizada y el diseño de los experimentos computacionales. En la segunda se presentan y analizan los resultados alcanzados.

4.1. BASE DE PROBLEMAS PRUEBAS

Para la ejecución de los experimentos computacionales se ha seleccionado una colección de problemas tipo [QPB] ampliamente referenciada en la última década [2, 5, 4, 6, 13], y públicamente disponible en <http://dollar.biz.uiowa.edu/~sburer>. La misma incluye 54 *problemas básicos* de hasta 60 variables y 36 *problemas extendidos* de entre 70 y 100 variables.

Con el fin de ampliar la experimentación a problemas de mayor dimensión, se generaron 36 *problemas extras* de entre 125 y 200 variables. Para ello se utilizó el programa *genboxqp.m* (disponible en <http://dollar.biz.uiowa.edu/~sburer>), empleado también en la formulación de los 90 problemas (básicos y extendidos) contenidos en la base de problemas pruebas.

Un aspecto no estudiado en la literatura consultada, es el comportamiento del método de ramificación y acotación en relación con las características del problema. Dado que la referida colección de problemas no satisface las condiciones exigidas para tal estudio, se construyeron 81 nuevos problemas (referidos como *problemas diseñados*). La generación de la matriz Q en estos problemas se realiza atendiendo a un diseño de experimentos multifactorial con tres factores: dimensión (N), densidad (D) y por ciento de valores propios negativos (QI). Para cada uno de estos factores se analizan tres niveles, obteniendo con ello ($3^3 = 27$) tipos de problemas. Dando cumplimiento a los principios básicos del diseño de experimento se generan aleatoriamente 3 réplicas para cada combinación de los factores (N, D, QI), dando origen así a los 81 ($= 27 * 3$) problemas diseñados. En la tabla-1 se resumen las características principales de la matriz Q correspondientes a cada una de las clases de problemas utilizados.

Dado que no resulta posible generar matrices de condición controlada, para valores prefijados de QI y D ,

Problemas -	Dimensión (N)	Densidad (D)	Val. Propios Negativos (QI)	Condición $\min(\kappa)$ - $\max(\kappa)$
Básicos	20-60	20%-100%	$\approx 50\%$	27.50 - 1886.71
Extendidos	70-100	25%-75%	$\approx 50\%$	64.09 - 728.49
Extras	125-200	25%-75%	$\approx 50\%$	101.97 - 15164.51
diseñados	20-100	25%-75%	20%-100%	11.74 - 2934.4

Tabla 1: Características de las matrices que conforman la base de problemas pruebas

la influencia del número de condición κ sobre los parámetros (TNC,NTI,TTR,ITO) se analizó mediante una prueba ANOVA simple.

4.2. ANÁLISIS DE LOS RESULTADOS

Para comparar cada par de procedimientos se utilizará una gráfica log-log por cada parámetro (TNC, NTI, ITO, TTR). En cada una de ellas, las coordenadas (x,y) asociadas a una instancia o problema corresponden al valor del parámetro que se mide calculado por cada uno de los procedimientos que se comparan. Una instancia (a,b) ubicada por debajo de la línea diagonal ($y = x$), indica que el valor (b) del parámetro analizado correspondiente

al procedimiento ubicado en la ordenada es inferior al valor (a) obtenido por el procedimiento que se representa en la abscisa.

Seguidamente se comparan los algoritmos híbridos entre sí, y el híbrido que utiliza la relajación doblemente no negativa con el método *B&B* de Burer y Chen. Con este fin se evalúa el valor de los parámetros (TNC,NTI,ITO,TTR) obtenidos al resolver los 90 problemas (básico y extendidos) de la colección de problemas pruebas referenciada. En cada gráfica se identifica a los problemas básicos con círculos rojos y los extendidos con círculos negros. En todos los experimentos se ha fijado un tiempo límite de cálculo igual a 40000 CPU-seg y una tolerancia $\varepsilon=0.01$.

Comparación entre las dos versiones del algoritmo híbrido.

Según lo establecido anteriormente, los resultados que se muestran en las figuras-1 y -2 permiten inferir que la nueva versión del algoritmo híbrido tiene un desempeño superior al algoritmo propuesto por [10]. Obsérvese que, en la gran mayoría de los problemas, las observaciones correspondientes a los 4 parámetros evaluados se ubican por debajo de la línea diagonal $y=x$.

Como dato adicional, vale destacar que el programa *CC+BB-BurVan* sólo resuelve 77 de los 90 problemas dentro

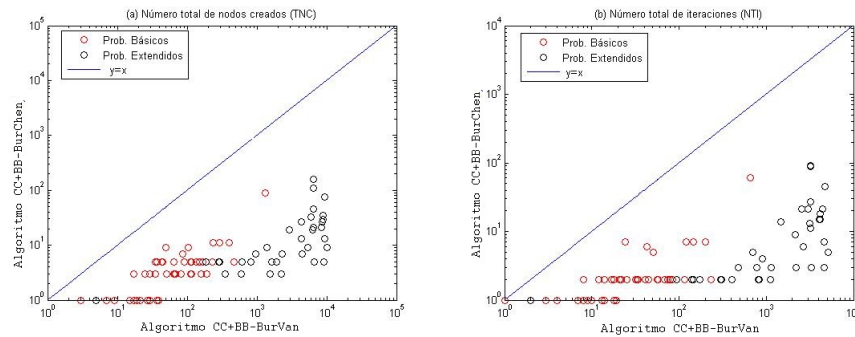


Figura 1: Comparación entre *CC+BB-BurVan.m* y *CC+BB-BurChen.m* respecto a: (a) el número de nodos creados y (b) el número total de iteraciones

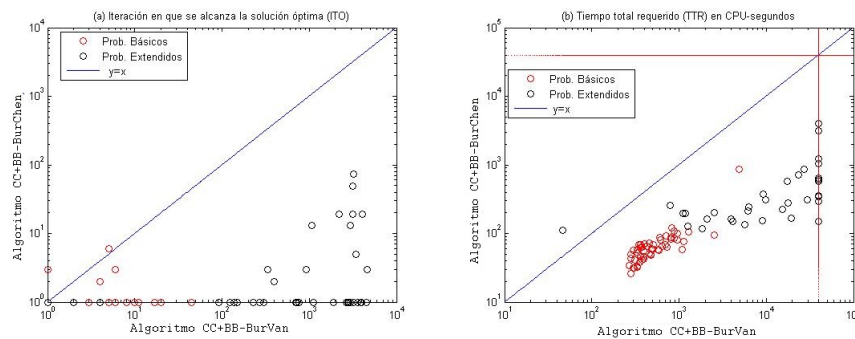


Figura 2: Comparación entre *CC+BB-BurVan.m* y *CC+BB-BurChen.m* respecto a: (a) el número de iteraciones realizadas para encontrar la solución óptima, y (b) el tiempo total requerido por el algoritmo

del límite de tiempo prefijado. En las 13 instancias restantes (ubicadas sobre la línea vertical en la figura-2b) el algoritmo se ve forzado a abortar sin verificar la optimalidad de la mejor solución calculada, pues el valor del *gap relativo* sobrepasa la tolerancia establecida.

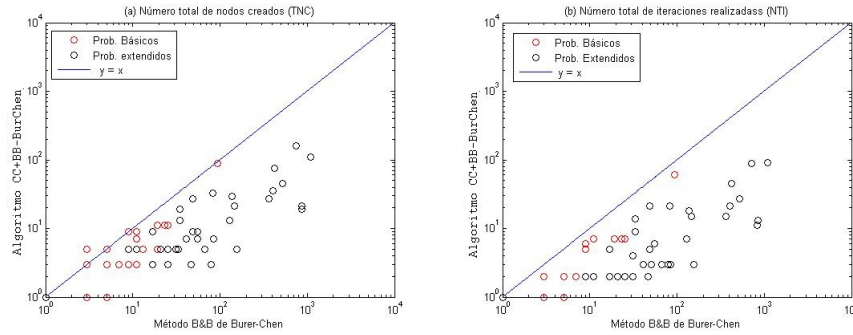


Figura 3: Comparación entre $CC + BB - BurChen.m$ y $quadprogbb.m$ respecto a: (a) el total de nodos creados y (b) el tiempo total requerido

Comparación entre el algoritmo propuesto y el método $B\&B$ de Burer-Chen.

De forma similar al caso anterior, en las figuras-3 y -4 se puede verificar que el algoritmo híbrido propuesto alcanza, en sentido general, un mejor desempeño que el método $B\&B$ de Burer-Chen.

Respecto al tiempo total requerido (figura-4b), sólo un problema no puede ser resuelto por $quadprogbb.m$ dentro

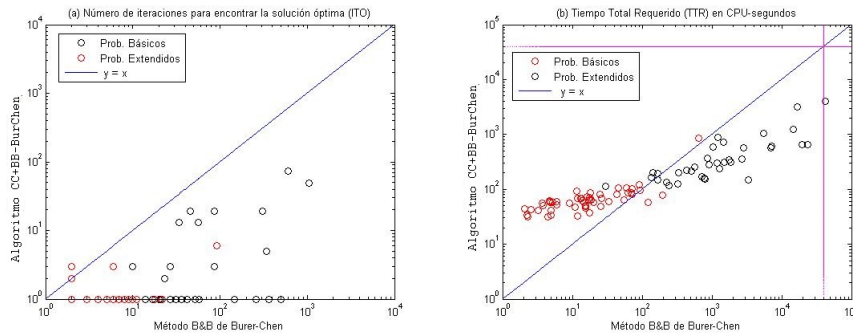


Figura 4: Comparación entre $CC+BB-BurChen.m$ y $quadprogbb.m$ respecto a: (a) el número de iteraciones necesarias para encontrar una solución óptima y (b) el número total de iteraciones requerido para reconocer que la solución calculada es un mínimo global del problema

del límite de tiempo prefijado. En esa misma figura se observa, además, que el método $B\&B$ de Burer-Chen resuelve los problemas de menor dimensión (problemas básicos - en color rojo) en menos tiempo. Sin embargo, la tendencia general indica que el algoritmo híbrido es más rápido cuando los problemas tienen mayor dimensión (problemas extendidos - representados en color negro). Tal comportamiento presupone que el esfuerzo computacional requerido para mejorar la calidad de las cotas superiores (método de corte y continuación) sólo tiene su recompensa cuando la dimensión de los problemas es superior a las 60 variables.

Los resultados computacionales presentados hasta aquí evidencian un comportamiento ventajoso del algoritmo híbrido propuesto sobre el algoritmo presentado en [10] y el método $B\&B$ de Burer-Chen, los cuales constituyen los mejores resultados publicados hasta la fecha al resolver problemas cuadráticos no convexos con restricciones de caja.

Influencia de las características del problema en el desempeño del algoritmo

Para estudiar los efectos que puedan tener los cambios en las características del problema (en lo adelante, *factores*: N,D,QI) en el desempeño del algoritmo, se procede a resolver los 81 problemas pruebas diseñados para este fin. El análisis estadístico de los resultados fue ejecutado con el programa STATGRAPHICS, y las variables de respuesta (parámetros) seleccionadas para medir el rendimiento del algoritmo serán las mismas que se han empleado en toda la sección (TNC,NTI,ITO,TTR).

En las figuras-5 y 6 se muestran las gráficas de Pareto resultantes del análisis de varianza (ANOVA), en las que se representa, en orden decreciente por su influencia, el efecto principal que produce cada factor y sus interacciones, sobre cada uno de los parámetros estudiados. En el centro de cada gráfica se distingue una

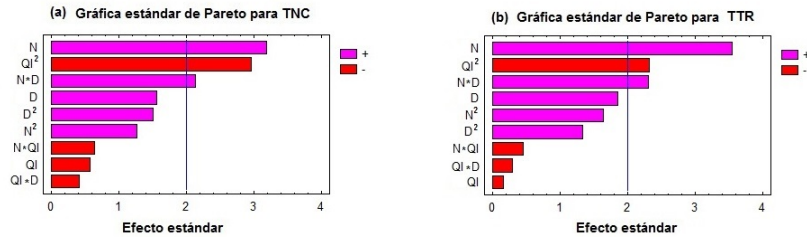


Figura 5: Diagrama de Pareto correspondiente al ANOVA de los parámetros TNC y TTR, versus los factores N,D,QI y sus interacciones

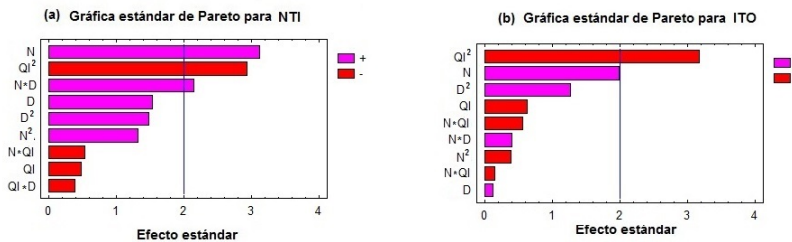


Figura 6: Diagrama de Pareto correspondiente al ANOVA de los parámetros NTI e ITO, versus los factores N,D,QI y sus interacciones

línea vertical, que identifica cuales de los factores estudiados influye efectivamente (con un nivel $\alpha = 0.05$ de significación) sobre las variables de respuesta analizadas. Partiendo de estos resultados, se reconoce (resultado esperado) que la dimensión (N) de los problemas tiene un efecto principal positivo significativo en el comportamiento de los cuatro parámetros.

Así mismo se pudo identificar un efecto significativo negativo del porcentaje de valores propios negativos $(QI)^2$ de la matriz Q sobre los cuatro parámetros analizados, mientras que la interacción de la dimensión (N) y la densidad (D) de la matriz Q tienen una influencia positiva sustancial sobre las variables TNC,TTR y NTI.

Dado que no fue posible generar matrices de condición (κ) controlada, para valores prefijados de QI y D, no fue posible incluir el estudio de este factor dentro del diseño experimental. En ese caso, y tomando como referencia los resultados de los 90 problemas pruebas (*básicos y extendidos*) se realizó una prueba ANOVA simple para este factor respecto a los parámetros (TNC,TTR,NTI,ITO) establecidos. Como resultado de esta se puede inferir que, aparentemente, (κ) no tiene una influencia significativa sobre las variables que determinan el desempeño del algoritmo (p-value=0.1353, 0.0746, 0.1739 y 0.7587 respectivamente, con $\alpha=0.05$).

Tomando en cuenta los factores significativos identificados para cada parámetro, pueden construirse, por re-

gresión, las siguientes ecuaciones de segundo orden:

$$TNC = -8,28704 + 0,202778 * N + 1,525 * QI + 0,623333 * D + 0,0103889 * N * D - 0,0127083 * QI^2 \quad (13)$$

$$TTR = 16,0792 + 42,6929 * N - 7,48549 * QI + 24,0321 * D + 0,400534 * N * D - 0,355774 * QI^2 \quad (14)$$

$$NTI = -3,55556 + 0,103704 * N + 0,733333 * QI + 0,305 * D + 0,00508333 * N * D - 0,00611111 * QI^2 \quad (15)$$

$$ITO = -3,13426 + 0,0222222 * N + 0,184722 * QI - 0,00153935 * QI^2 \quad (16)$$

No obstante el valor del estadígrafo R^2 obtenido en cada uno de los casos (0.32, 0.32,0.32, 0.24 respectivamente) indica que estas ecuaciones sólo explican pobremente la variabilidad de cada uno de los parámetros. En contraste, al estudiar las relaciones de los parámetros entre sí, se evidencia que existe una fuerte relación entre ellos (ver R^2).

$$NTI = 0.4851 * TNC - 0.1259 \quad (R^2 = 0.9994) \quad (17)$$

$$TTR = 73.12 * NTI + 7.335 * ITO + 1.278 \quad (R^2 = 0.9682) \quad (18)$$

La ecuación-(17) explica que el algoritmo sólo resuelve (NTI) aproximadamente la mitad (0.48) de los nodos generados (TNC) (el resto de los nodos resultan eliminados). Este resultado evidencia la fuerza que aporta la determinación de una cota superior f_{UB}^* más fuerte sobre el criterio de eliminación asumido en el algoritmo.

La ecuación-(18), por su parte, explica que cada iteración *B&B* (NTI) consume aproximadamente 73 CPU-segundos, mientras que cada iteración que demora el algoritmo en encontrar la solución óptima (ITO) cuesta aproximadamente 7 CPU-segundos.

Las relaciones anteriores pueden ser utilizadas para estimar aproximadamente el comportamiento del algoritmo al resolver problemas con características conocidas. Así, por ejemplo, usando las ecuación-(14) es posible tener un pronóstico para el tiempo de ejecución del algoritmo al resolver problemas con más de 100 variables. En la tabla-2 se muestran el promedio y el valor máximo estimados, de los tiempos de ejecución (medidos en CPU-segundos) calculados para problemas de hasta 200 variables.

Una evaluación optimista de los resultados anteriores permiten conjeturar que el algoritmo propuesto tiene

Dimensión	125	150	175	200
TTR-Promedio	7.7939e+03	9.3619e+03	1.0930e+04	1.2498e+04
TTR-Máximo	9.6464e+03	1.1465e+04	1.3283e+04	1.5101e+04

Tabla 2: Tabla con valores pronosticados para la variable TTR en problemas de más de 100 variables

capacidad para resolver problemas de hasta 200 variables dentro del límite de tiempo prefijado.

Solución de problemas con más de 100 variables

Para medir las posibilidades del algoritmo propuesto al resolver problemas con más de 100 variables y comparar su rendimiento con el método *B&B* de Burer-Chen se resuelven los 36 *problemas extras* descritos al inicio del capítulo. En la tabla-3 se resumen los resultados alcanzados por los dos algoritmos, agrupando los problemas por su dimensión. Las tripleta (**R-T-M**) que aparece en las dos últimas columnas de la tabla, indican: (**R**) la cantidad de problemas Resueltos por el algoritmo; (**T**) la cantidad de problemas para los que el algoritmo sobrepasa el límite de Tiempo prefijado; y (**M**), la cantidad de problemas en los que el algoritmo se detiene por limitaciones de Memoria.

Estos resultados evidencian un comportamiento superior del algoritmo propuesto respecto al método *B&B* de Burer-Chen. Puede observarse, además que la limitación fundamental de ambos algoritmos para resolver problemas de mayor dimensión no es precisamente el tiempo, sino la memoria disponible en la computadora. Estos resultados se corresponden con los mostrados en las figura-3, en la cual se observa que el nuevo híbrido propuesto genera menos nodos y precisa menos iteraciones que el método *B&B* de Burer-Chen, como consecuencia de lo cual tiene menos exigencias de memoria.

Dimensión	Problemas	CC+BB-BurChen R-T-M	B&B de Burer-Chen R-T-M
125	9	7-0-2	4-1-4
150	9	7-1-1	0-0-9
175	9	6-0-3	0-0-9
200	9	2-0-7	0-0-9

Tabla 3: Resumen de los resultados para los problemas extras

Adicionalmente, en la figura-7(b) se puede constatar que los tiempos reportados por el algoritmo híbrido, están ligeramente por encima de los tiempos pronosticados en la tabla-2, cuando se utiliza la ecuación (14). Para finalizar, en la figura-7(a) se representa el gap relativo correspondiente a las soluciones calculadas por ambos algoritmos. En ella se observa que el algoritmo CC+BB-BurChen alcanza aproximaciones de mejor calidad que el método B&B-BurChen. Estos resultados confirman el buen comportamiento del algoritmo híbrido propuesto.

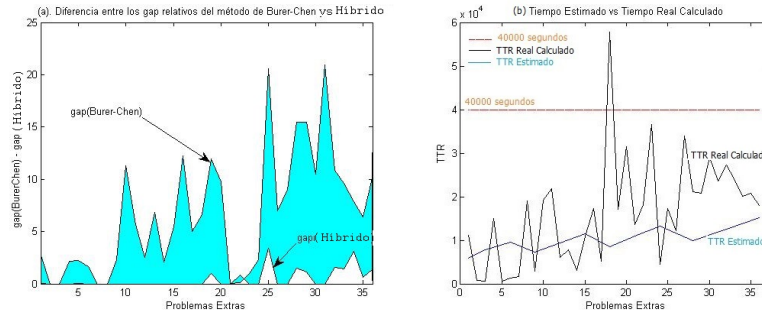


Figura 7: Comparación entre el algoritmo híbrido y el método de Burer-Chen respecto al gap relativo y entre el tiempo estimado y real calculado por el algoritmo híbrido en la solución de los Problemas Extras

5. CONCLUSIONES Y TRABAJOS FUTUROS

En este trabajo se propone un nuevo esquema híbrido consistente en sustituir el método de ramificación y acotación de Burer-Vandenbusche por el de Burer-Chen, dentro del mismo esquema híbrido propuesto por [10]. Se utiliza una base de problemas pruebas más amplia y diversa que la empleada en los trabajos referenciados. La experimentación computacional realizada permitió comprobar un mejor desempeño del nuevo algoritmo en comparación con el método *CC+BB-BurVan* y *B&B* de Burer-Chen, reportados en la literatura con los mejores resultados al resolver problemas de programación cuadrática no convexa con restricciones de caja. En particular se demuestra que la variación propuesta permite duplicar la dimensión de los problemas resueltos con un costo computacional inferior al de los algoritmos referidos.

El diseño experimental realizado permitió estudiar la influencia de las características del problema en el comportamiento del algoritmo, identificando a la dimensión de los problemas y su interacción con la densidad de la matriz Q , así como el porcentaje de valores propios negativos de la misma, como los factores que inciden significativamente en su desempeño.

Como trabajo futuro, resulta interesante investigar el comportamiento del algoritmo propuesto en problemas con 200 variables o más en computadores más potentes y buscar nuevas vías para investigar la incidencia de la condición de la matriz Q en el desempeño del algoritmo.

RECEIVED: MAY, 2017.

REFERENCIAS

- [1] ALLGOWER, E.L. and GEORG, K. (1990): **Numerical Continuation Methods: An Introduction**. Springer Verlag, Berlin.
- [2] BURER, S. and VANDENBUSSCHE, D. (2008): A finite branch-and-bound algorithm for nonconvex quadratic programming via semidefinite relaxations. **Math. Program.**, Serie A, 113, 259-282.
- [3] BURER, S. (2009): On the copositive representation of binary and continuous nonconvex quadratic programs. **Math. Program.**, 120, 479-495.
- [4] BURER, S. and CHEN, J.(2011): Relaxing the optimality conditions of box QP. **Computational Optimization and Applications**, 48, 653-673.
- [5] CHEN, J. (2010): Convex relaxations in nonconvex and applied optimization. University of Iowa. Disponible en <http://ir.uiowa.edu/etd/654>. Consulted 25-10,2014.
- [6] CHEN, J. and BURER,S.(2012): Globally solving nonconvex quadratic programming problems via completely positive programming. **Math. Prog. Comp.**, 4, 33-52.
- [7] GÓMEZ, W. and GUDDAT, J. and JONGEN, H.TH. and RUCKMANN, J. and SOLANO, C.(2000): Curvas críticas y saltos en optimización no lineal. Disponible en <http://www.emis.de/monographs/curvas/index.html>. Consulted 12-9,2014.
- [8] JONGEN, H.TH. and JONKER, P. and TWILT, F.(1986): Critical sets in parametric optimization. **Mathematical Programming**, 34, 333-353.
- [9] MIRANDA, R. (2010): Theoretical approach for the calculation of a feasible point on the special non-convex set described by a polyhedron and a quadratic fuction. **Optimization**, 59, 6, 917-927.
- [10] MIRANDA, R., and ALLENDE, S., and PÉREZ, B. and BOUZA, G.(2015): Algoritmo de corte y continuación con ramificación y acotación para la solución del problema cuadrático con restricciones de caja. **Revista Ciencias Matemáticas**, 29, 1, 19-29.
- [11] MURTY, K.G. and KABADI, S.N.(1987): Some NP-Complete problems in quadratic and nonlinear programming. **Mathematic Programming**, 39, 117-129.
- [12] PARDALOS, P.M. and VAVASIS, S.(1991): Quadratic programming with one negative eigenvalue is NP-Hard. **Journal of Global Optimization**, 1, 15-22.
- [13] VANDENBUSSCHE, D. and NEMHAUSER, G.L.(2005): A branch-and-cut algorithm for nonconvex quadratic programs with box constraints. **Math. Program.**, Serie A, 102, 559-575.