



ABSTAYLOR: límite superior con regiones internas en problemas de optimización global continua no lineal

Víctor Reyes<sup>1</sup> · Ignacio Araya<sup>1</sup>

Recibido: 15 de enero de 2019 / Aceptado: 14 de enero de 2020 / Publicado en línea: 25 de enero de 2020 © Springer Science+Business Media, LLC, parte de Springer Nature 2020

## Resumen

En este artículo proponemos AbsTaylor, un método simple y rápido para extraer politopos internos, es decir, regiones convexas completamente factibles en las que todos los puntos satisfacen las restricciones. El método realiza una linealización interna de las restricciones no lineales utilizando una forma de Taylor. A diferencia de una propuesta anterior, el punto de expansión de la forma de Taylor no se limita a los límites de los dominios, produciendo así, en general, una aproximación más estrecha. Para probar el enfoque, se introdujo AbsTaylor como un método de límite superior en un optimizador de rama y límite global de última generación. Además, implementamos un método de búsqueda local que extrae politopos internos factibles para encontrar iterativamente mejores soluciones dentro de ellos. En los casos estudiados, el nuevo método encuentra en promedio cuatro veces más regiones internas y mejora significativamente el rendimiento del optimizador.

**Palabras clave** Límite superior · Linealización basada en Taylor · Optimización global continua no lineal

## 1. Introducción

Este artículo trata sobre los problemas de optimización global continua no lineal (NCOP), que se definen de la siguiente manera:

$$\min f(x) \text{ st } g(x) \leq 0, x \in X \quad (1)$$

siendo  $x \in \mathbb{R}^n$  el conjunto de variables que varían en el cuadro 1.  $f: \mathbb{R}^n \rightarrow \mathbb{R}$  es la función objetivo de valor real y  $g: \mathbb{R}^n \rightarrow \mathbb{R}^m$  es un conjunto de restricciones de desigualdad.  $f$  y  $g$  pueden ser funciones no lineales (convexas o no convexas). Una solución factible es un vector  $x$  que satisface todas las restricciones del problema. Una solución óptima  $x^*$  es una solución factible tal que

<sup>1</sup> Un intervalo  $x_i = [x_i^l, x_i^u]$  define el conjunto de reales  $x_i$ , tal que  $x_i^l \leq x_i \leq x_i^u$ .  
 de intervalos  $x_1 \times \dots \times x_i \times \dots \times x_n$ .

— Una caja  $x$  es un producto cartesiano

B Víctor Reyes  
 vareyesrodriguez@lirmm.fr

Ignacio Araya  
 ignacio.araya@pucv.cl

<sup>1</sup> Pontificia Universidad Católica de Valparaíso, Valparaíso, Chile

$f(x) \leq f(x)$  para cualquier solución factible  $x$ . R. En este trabajo nos centramos en solucionadores de optimización global (u optimizadores globales) capaces de encontrar sólo una solución óptima, es decir, una solución factible  $x$  tal que  $f(x) \leq f(x) + \epsilon$ .

Los optimizadores globales generalmente siguen una estrategia de rama y límite para resolver NCOP (por ejemplo, IbexOpt [1], Baron [2], Antigone [3], LINDO [4], Couenne [5], Icos [6], GlobSol [7]). Comenzando con un nodo raíz que considera los dominios variables iniciales, los optimizadores construyen un árbol de búsqueda alternando pasos de filtrado, límite superior y bisección hasta que se encuentra una solución óptima (es decir, garantizan la optimización). El filtrado consiste en eliminar de los dominios los valores que no son consistentes con las restricciones (o con algunas condiciones de optimización). La bisección consiste en expandir el nodo actual dividiendo el dominio de una variable y generando dos hijos. El límite superior consiste en encontrar una solución factible en el cuadro actual que mejore el costo de la mejor solución encontrada hasta el momento. Los optimizadores globales, en general, difieren principalmente en los procedimientos utilizados para el filtrado, la bisección, el límite superior y la selección del siguiente nodo a procesar. En particular, los optimizadores basados en intervalos (por ejemplo, IbexOpt, GlobSol e Icos) son capaces de resolver NCOP de forma rigurosa, es decir, tienen en cuenta los errores de redondeo que implican las operaciones de punto flotante en sus implementaciones. Los optimizadores basados en intervalos se distinguen de otros optimizadores en que (1) tienen en cuenta las incertidumbres en los valores de los parámetros y los errores de cálculo sobre los números de punto flotante y (2) son capaces de tratar el espacio de búsqueda completo, ofreciendo así pruebas de inviabilidad y/o certificación de soluciones. Debido a estas características, estos métodos se pueden utilizar en varias áreas de investigación, incluida la robótica, la localización, la calificación de modelos y el diseño de sistemas de control [8]. De los optimizadores basados en intervalos existentes, IbexOpt parece ser el más robusto [9].

Limitar superiormente la función objetivo es una tarea crucial de los optimizadores globales en general. Encontrar una solución factible xub mejorando la mejor solución encontrada hasta el momento, ayuda al optimizador a reducir el espacio de búsqueda mediante la restricción auxiliar:  $f(x) < f(x_{ub}) - \epsilon$  [8]. Los métodos de búsqueda local rápida se utilizan generalmente para encontrar soluciones factibles en cada cuadro del árbol de búsqueda (por ejemplo, seleccionando el punto medio del cuadro [10], aplicando el método de Newton [6,11] o extrayendo regiones internas convexas [9]), además de algunas pruebas de viabilidad (por ejemplo, la evaluación natural del punto candidato [9,10] o la prueba de Borsuk [6]). Como el límite superior se realiza en todos los nodos del árbol de búsqueda, es posible que los métodos a veces no encuentren soluciones factibles sin comprometer la convergencia global del optimizador.

Algunos optimizadores (por ejemplo, Antigone y Baron) utilizan solucionadores locales como CONOPT [12] o IPOpt [13] para el límite superior. Debido a que los solucionadores locales son relativamente costosos en tiempo, generalmente se utilizan sólo al inicio de la búsqueda o, dinámicamente, sólo en algunos nodos del árbol de búsqueda [14] (en el resto de nodos se aplican métodos rápidos). CONOPT implementa el método de gradiente reducido generalizado (GRG) [15], que es una generalización del método de descenso de gradiente para problemas con restricciones no lineales. GRG primero transforma el problema en uno sin restricciones, y luego la dirección de búsqueda se obtiene combinando el gradiente de la función objetivo y un pseudogradient derivado de las restricciones de igualdad. La dirección de búsqueda establece que cualquier restricción activa permanece activa durante algún pequeño paso. IPOpt implementa un método de punto interior dual primario. Los métodos de punto interior [16] son métodos iterativos que aproximan el problema no lineal utilizando la función de barrera lagrangiana y logarítmica para las variables de holgura. Luego, aplican el método de Newton a las condiciones KKT2 de la aproximación para obtener las direcciones de búsqueda del método (las funciones deben ser dos veces diferenciables de forma continua). Finalmente, se utiliza un método de búsqueda de líneas para calcular el tamaño del paso de la iteración. Una decisión importante está relacionada con el procedimiento para elegir el parámetro para la función de barrera [18].

2 Las condiciones de Karush-Kuhn-Tucker (KKT) son condiciones necesarias para que una solución sea óptima [17].

Los métodos de región de confianza para la optimización restringida [19–21] también son algoritmos de búsqueda local que podrían usarse para el límite superior. Estos algoritmos generalmente comienzan desde un punto inicial e intentan obtener un nuevo punto de iteración simplemente buscando en las proximidades del actual. En cada iteración  $k$ , se resuelve una aproximación del problema (1) para una subregión del espacio de búsqueda alrededor de la mejor solución actual  $x_k$  (región de confianza). Si se encuentra una nueva mejor solución  $x_{k+1}$ , es decir,  $f(x_{k+1}) < f(x_k)$ , entonces se actualiza la solución actual. El problema aproximado consiste en minimizar una aproximación de la función objetivo (por ejemplo, una linealización o una aproximación cuadrática) sujeta a una linealización de las restricciones en la región de confianza. Debemos resaltar que este tipo de algoritmos no garantizan la viabilidad de las soluciones encontradas.

En [9] los autores proponen XTaylor, un procedimiento de acotación superior que construye una región convexa interna, es decir, un subconjunto del espacio de búsqueda en el que todas las soluciones son factibles. El procedimiento utiliza una forma de Taylor de primer orden relacionada con una caja  $x$  y genera, para cada desigualdad  $g_j(x) \leq 0$ , una restricción lineal  $h_j(x) \leq 0$ , tal que  $g_j(x) \leq h_j(x)$  para cualquier  $x$  en  $x$ . Si logra construir la región interior, se minimiza una linealización de la función objetivo utilizando un algoritmo simplex y se encuentra una solución factible, que se aproxima a la óptima en el cuadro. El punto de expansión de la linealización está restringido a ser una esquina del cuadro  $x$ , esto significa que incluso conociendo una solución factible  $x$  en el cuadro, no es seguro que el método sea capaz de construir una región interior alrededor de esta solución.

En este trabajo hacemos una doble contribución. Primero, proponemos AbsTaylor, un algoritmo rápido de límite superior que también utiliza linealizaciones basadas en Taylor para construir regiones internas relacionadas con el problema (1). Gracias a la aplicación de una linealización con valores absolutos se supera la dificultad que presenta XTaylor, es decir, puede utilizar cualquier punto dentro de la caja como punto de expansión. A través de experimentos demostramos que, cuando se selecciona el punto medio de la caja como punto de expansión, AbsTaylor encuentra, en promedio, cuatro veces más regiones internas que su predecesor. También mostramos que, cuando IboxOpt utiliza una técnica de límite superior basada en AbsTaylor, reporta una reducción significativa en el tamaño del árbol de búsqueda, especialmente cuando la precisión del optimizador se establece en valores no tan bajos (por ejemplo,  $= 10^{-2}$  o  $= 10^{-4}$ ).

En segundo lugar, proponemos IterativeTaylor, un algoritmo de búsqueda local simple que explora regiones internas aplicando iterativamente AbsTaylor. El procedimiento es similar al seguido por los algoritmos de región de confianza: parte de una solución factible inicial  $x_c$  y construye una región interna alrededor de  $x_c$  usando AbsTaylor. Si se puede encontrar una solución mejor dentro de esta región, entonces esta solución se convierte en el punto de expansión de la siguiente iteración. El proceso se repite hasta que se alcanza un criterio terminal. Integrado en un optimizador de ramas y límites, IterativeTaylor mejora aún más el rendimiento del optimizador. A diferencia de los métodos de límite superior existentes:

- IterativeTaylor es bastante simple: realiza relajaciones del problema de primer orden y utiliza el algoritmo simplex para encontrar soluciones prometedoras dentro de las regiones convexas. No requiere calcular segundas derivadas y no utiliza las condiciones KKT como en el caso de los métodos de punto interior primal-dual.
- IterativeTaylor encuentra una solución factible en cada iteración de la búsqueda porque solo funciona con regiones completamente factibles.<sup>3</sup> Otros métodos, como los algoritmos de región de confianza, el método GRG o los métodos de punto interior, no garantizan que el candidato actual sea una solución factible, pero sólo uno aproximado.

<sup>3</sup> Siempre que se dé una solución inicial factible. De lo contrario, el mismo método puede encontrar una solución inicial factible.

- AbsTaylor es capaz de construir regiones internas alrededor de cualquier solución factible, lo que lo convierte en un componente perfecto para algoritmos de búsqueda local iterativos (como IterativeTaylor). XTaylor, otro enfoque, garantiza el éxito en la búsqueda de regiones interiores sólo si hacemos coincidir una esquina de la caja (el punto de expansión) con una solución factible.

El documento está organizado de la siguiente manera. La sección 2 proporciona nociones básicas relacionadas con la aritmética de intervalos. Insecto. 3 describimos nuestra linealización basada en Taylor que se compara con la propuesta anterior. La sección 4 describe nuestro algoritmo de búsqueda local e informa los resultados experimentales. La Sección 5 presenta nuestras conclusiones y el trabajo futuro.

## 2. Fondo

En esta sección presentamos algunos conceptos básicos relacionados con la aritmética de intervalos y la extracción de regiones internas.

### 2.1 Intervalos

Un intervalo  $x_i = [x_i, \bar{x}_i]$  define el conjunto de reales  $x_i$  st  $x_i \leq x \leq \bar{x}_i$ , donde  $x_i$  y  $\bar{x}_i$  son números de coma flotante. El tamaño o ancho de  $x_i$  se define como  $\text{wid}(x_i) = \bar{x}_i - x_i$ .  $\text{mid}(x_i)$  denota el punto medio de  $x_i$ . El casco de dos intervalos se define como  $\text{casco}(x_i, x_j) = [\min(x_i, x_j), \max(x_i, x_j)]$ .

Un cuadro  $x = (x_1, \dots, x_n)$  representa el producto cartesiano de intervalos  $x_1 \times \dots \times x_n$ . El tamaño de una caja es  $\text{wid}(x) = \max_i \text{wid}(x_i)$ . Intervalo la aritmética define la extensión de operadores unarios y binarios, por ejemplo:

$$\begin{aligned} x_1 + x_2 &= [x_1 + x_2, x_1 + x_2] \\ &= [x_1 - x_2, x_1 - x_2] \\ &= [\min(x_1, x_2), \max(x_1, x_2)] \end{aligned}$$

Una función  $f: \mathbb{R}^n \rightarrow \mathbb{R}$  es factorizable si se puede calcular en un número finito de pasos simples, utilizando operadores unarios y binarios.  $f: \mathbb{R}^n \rightarrow \mathbb{R}$  se dice que es una extensión de una función real factorizable  $f$  a intervalos si:

$$x \in \mathbb{R}^n, f(x) \in \{f(x), x\}.$$

La imagen óptima  $f$  es el intervalo más nítido que contiene la imagen de  $f(x)$  sobre  $x$ . Hay varios tipos de extensiones, en particular, la extensión natural  $f$  corresponde a mapear una función real  $n$ -dimensional  $f$  a intervalos mediante el uso de aritmética de intervalos.

### 2.2 Extracción de regiones factibles convexas

El método XTaylor [9] extrae regiones internas utilizando una forma de Taylor de intervalo de primer orden del vector de función  $g$ . La forma de intervalo de Taylor garantiza que para todo  $x$  y  $j \in \{1, \dots, m\}$ :

$$g_j(x) - g_j(x_i) \leq \sum_{i=1}^n J_{ji} \cdot (x_i - x_i), \quad (2)$$

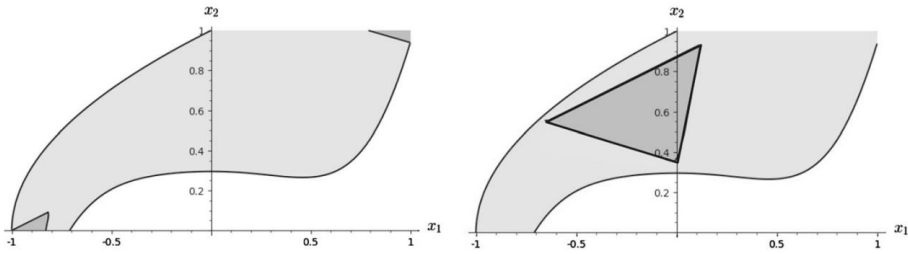


Fig. 1 Linealizaciones internas utilizando la forma de intervalo de Taylor. (izquierda) Polítopos internos generados por XTaylor (gris oscuro) dentro de la región factible (gris claro); (derecha) polítopo interno generado por AbsTaylor usando el punto medio (gris oscuro) dentro de la región factible (gris claro)

donde  $J$  es el intervalo de la matriz jacobiana de  $g$  en el cuadro  $x$ , es decir, cualquier elemento  $J_{ij}$  en la matriz corresponde a una sobreestimación de intervalo de la imagen de la derivada parcial  $(x)$  sobre  $\frac{\partial g_i}{\partial x_j}$ .  $x$  corresponde al punto de expansión.

Debido a que  $J_{ij}$  es un intervalo, la aproximación no es convexa a menos que el punto de expansión es una esquina de la caja  $x$ . El método XTaylor propone utilizar una esquina de la caja  $x_c$  como punto de expansión y la aproximación lineal se convierte en:

$$g_i(x) \leq h_i(x) = g_i(x_c) + \sum_{j=1}^n J_{ij} (x_j - x_{c,j}) \quad (3)$$

donde,  $J_{ij} = \frac{\partial g_i}{\partial x_j} \big|_{x_c} = x_{c,j}$  en caso contrario  $J_{ij} = \overline{J_{ij}}$ .

El conjunto de soluciones que satisfacen  $h(x) \leq 0$  denota una región interior convexa o polítopo interior que es un subconjunto del conjunto de soluciones que satisfacen  $g(x) \leq 0$ .

Ejemplo 1 Consideremos el siguiente sistema de restricciones:

$$\begin{aligned} g_1(x) &= x_1^5 + 0,5 \cos(x_1) + \sin(x_2) - 2x_2 - 0,2 \leq 0 \\ g_2(x) &= -x_1 + x_2^2 - 1 \leq 0 \end{aligned}$$

en el cuadro  $x = [-1, 1] \times [0, 1]$ . Aplicando XTaylor con cada una de las cuatro esquinas de la caja como punto de expansión, solo obtenemos dos regiones interiores no vacías. Usando  $x_c = (x_1, x_2)$  obtenemos:

$$\begin{aligned} h_1(x) &= 5,4207x_1 - x_2 + 4,4909 \leq 0 \\ h_2(x) &= -x_1 + 2x_2 - 1 \leq 0 \end{aligned}$$

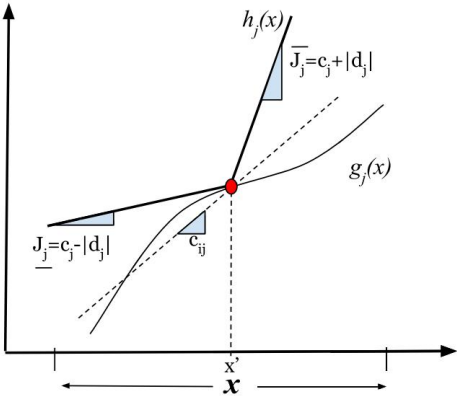
Y usando  $x_c = (x_1, x_2)$  obtenemos:

$$\begin{aligned} h_1(x) &= -0,4207x_1 - 1,4596x_2 + 1,792 \leq 0 \\ h_2(x) &= -x_1 \leq 0 \end{aligned}$$

La Figura 1 a la izquierda muestra en gris claro la región factible relacionada con la restricción original  $g(x) < 0$  y en gris oscuro las regiones interiores convexas generadas por XTaylor.

Para encontrar una solución prometedora dentro del polítopo, los autores proponen utilizar el Algoritmo simplex para minimizar una linealización de la función objetivo, es decir,  $\min f(x) + \sum_{i=1}^n \text{mid}(J_{ij}) \cdot (x_i - x_{c,i})$ , sujeto a la restricción  $h(x) \leq 0$ .

Fig. 2 Ejemplo de linealización basada en Taylor con valores absolutos para una función univariada  $g_j(x)$



Tenga en cuenta que el método XTaylor requiere que el punto de expansión esté en una esquina del cuadro. En este trabajo proponemos AbsTaylor, un nuevo método para extraer regiones internas que puede usar cualquier punto dentro de  $x$  como punto de expansión.

3 ABSTAYLOR: una linealización interna basada en valores absolutos

En esta sección proponemos un método simple para extraer regiones internas que puede usar cualquier punto del cuadro (por ejemplo, el punto medio) como punto de expansión. Para hacer eso, primero descomponemos las derivadas parciales del intervalo en dos valores:  $J_j = c_j + d_j$ , donde el valor real  $c_j$  es el punto medio de la derivada del intervalo y  $d_j = J_j - c_j$  (tenga en cuenta que  $d_j = -d_j$ ).

Luego, considerando la relación (2), proponemos la siguiente linealización interna basada en Taylor con valores absolutos:

$$g_j(x) \leq h_j(x) = g_j(x_i) + \sum_i c_j \cdot (x_i - x_i) + |d_j| \cdot |x_i - x_i|, \tag{4}$$

donde  $x_i$  puede ser cualquier punto dentro del cuadro  $x$ , por ejemplo  $x_i = \text{mid}(x)$ , y  $|d_j| = d_j = -d_j$ . La Figura 2 ilustra un ejemplo de una función univariada. Los triángulos representan la pendiente de los segmentos de línea correspondientes. La función  $h_j(x)$ , que consta de los dos segmentos gruesos situados encima de  $g_j(x)$ , comienza con una pendiente igual a  $J_j$  que se convierte en  $J_j$  después del punto de expansión. La línea de puntos con pendiente igual a  $c_j$  representa la función  $h_j(x)$  sin el término  $|d_j| \cdot |x - x_i|$ .

Ejemplo 2 Consideremos el mismo sistema y caja del Ejemplo 1. Aplicando (4), y usando el punto medio de la caja como punto de expansión, obtenemos la siguiente linealización:

$$g_1 \leq 2,5x_1 + 2,9207|x_1| - 1,2298x_2 + 0,2298|x_2 - 0,5| + 0,4944 \leq 0 \quad g_2 \leq -x_1 + x_2 + |x_2 - 0,5| - 1,25 \leq 0$$

El politopo obtenido mediante este proceso se puede ver en la Fig. 1-derecha. Tenga en cuenta que en este caso el politopo obtenido es mucho más grande que los generados por XTaylor.

De manera similar a lo propuesto en [9], para encontrar una solución factible prometedora en la región interior, proponemos minimizar una linealización de la función objetivo sujeta al sistema de restricciones generado por (4), es decir,

$$\begin{aligned} \min_x f(x) + \sum_i \text{medio}(J_i) \cdot (x_i - x_i) \\ \text{stg}j(x) + \sum_i c_i j \cdot (x_i - x_i) + |d_i j| \cdot |x_i - x_i| \leq 0. \end{aligned} \quad (5)$$

Para trabajar con las funciones de valor absoluto, reemplazamos las expresiones  $|x_i - x_i|$  por variables auxiliares  $u_i$ . Luego, como la derivada parcial de las funciones de restricción es positiva respecto de las variables auxiliares (es decir,  $|d_i j| \geq 0$ ), sumamos las restricciones  $u_i \geq x_i - x_i$  y  $u_i \geq -(x_i - x_i)$  y resolvemos la siguiente programa lineal equivalente utilizando el algoritmo simplex:

$$\begin{aligned} \min_x f(x) + \sum_i \text{medio}(J_i) \cdot (x_i - x_i) \\ \text{stg}j(x) + \sum_i c_i j \cdot (x_i - x_i) + |d_i j| \cdot u_i \leq 0, \quad j = 1 \dots m \\ u_i \geq x_i - x_i, \quad y_0 = 1 \dots n \\ u_i \geq -(x_i - x_i), \quad i = 1 \dots n. \end{aligned} \quad (6)$$

## experimentos

Las propuestas se implementaron como procedimientos de límite superior en el optimizador global IbexOpt [1]. En IbexOpt, L-Smear [22] se utiliza para seleccionar la siguiente variable a dividir. Para el filtrado se utilizan una serie de métodos de última generación: ACID(HC4) [23,24] más un contratista basado en relajación lineal que combina dos métodos de relajación (AF2 [10] y XNewton [25]). De forma predeterminada, IbexOpt utiliza un método de límite superior basado en XTaylor que primero extrae una región interior (usando una esquina aleatoria del cuadro como punto de expansión) y luego, para encontrar soluciones prometedoras dentro de esta región, minimiza una linealización de la función objetivo mediante utilizando el algoritmo Simplex (XTaylor&S). El optimizador también utiliza un método de límite superior adicional basado en la extracción de cuadros factibles [9]. Si los métodos de límite superior anteriores fallan, el optimizador simplemente intenta con el punto medio del cuadro.

Nuestra estrategia AbsTaylor&S consiste en sustituir XTaylor por AbsTaylor en el procedimiento de delimitación superior. Como las regiones internas encontradas por AbsTaylor no son necesariamente mejores, en calidad de solución, que las regiones encontradas por XTaylor, implementamos TwiceTaylor&S, una estrategia que aplica XTaylor&S y luego AbsTaylor&S. Observe que para estos experimentos, AbsTaylor siempre usa el punto medio de la caja como punto de expansión.

Los experimentos se realizaron en una computadora con una CPU Intel Core i7-4700MQ de 2,40 GHz y 8 GB de RAM, con Ubuntu 16.04. El conjunto de instancias se seleccionó de los puntos de referencia de COCONUT para la optimización global.<sup>4</sup> Seleccionamos todas las instancias resueltas por la estrategia predeterminada en un tiempo comprendido entre 2 y 3600 segundos cuando  $\epsilon = 10^{-6}$  (44 instancias). Los detalles de estos casos se muestran en la Tabla 1.

**Definición 1** Ganancia en tiempo relativo Definimos como tiempo relativo  $\text{tr}(a, b, \Pi)$  la relación entre el tiempo medio ( $t$ ) tomado por una estrategia  $a$  y la suma de los tiempos medios tomados por las estrategias  $a$  y  $b$  al resolver una instancia  $\pi$ , es decir,  $\text{tr}(a, b, \Pi) = \frac{t(a, \Pi)}{t(a, \Pi) + t(b, \Pi)}$ . Así, la ganancia

El tiempo de una estrategia  $a$  frente a una estrategia  $b$  está dado  $\frac{\pi}{\Pi} \frac{\text{tr}(b, a, \Pi)}{\text{tr}(a, b, \Pi)}$  donde  $\Pi$  es el conjunto de por los casos considerados.

<sup>4</sup> <http://www.mat.univie.ac.at/~neum/glopt/coconut/Benchmark/Benchmark.html>.

Tabla 1 Detalles del benchmark  
instancias utilizadas en los experimentos

| Punto de referencia nm Convexo |    |     |      | Punto de referencia nm Convexo |    |     |      |
|--------------------------------|----|-----|------|--------------------------------|----|-----|------|
| ex2_1_9                        | 11 | 2   | ?    | ex14_2_3                       | 7  | 10? |      |
| ex5_4_3                        | 17 | 14? |      | ex14_2_7                       | 7  | 10  | ?    |
| ex6_1_1                        | 9  | 7   | ?    | hh justo                       | 30 | 26? |      |
| ex6_1_3                        | 13 | 10? |      | himmel                         | 16 | 19  | 22 ? |
| ex6_2_6                        | 4  | 2   | ?    | hidro                          | 32 | 25  | ?    |
| ex6_2_8                        | 4  | 2   | ?    | immune                         | 22 |     | 8 si |
| ex6_2_9                        | 5  | 3   | ?    | lanzamiento                    | 39 | 29? |      |
| ex6_2_10                       | 7  | 4   | ?    | lineal                         | 25 | 21  | ?    |
| ex6_2_11                       | 4  | 2   | ?    | mediavar                       | 9  |     | 3 si |
| ex6_2_12                       | 5  | 3   | ?    | ramsey                         | 34 | 23? |      |
| ex7_2_3                        | 9  | 7   | ?    | srcpm                          | 40 | 28  | si   |
| ex7_2_4                        | 9  | 5   | ?    | lote                           | 47 | 74  | si   |
| ex7_2_8                        | 8  | 4   | ?    | dipigri                        | 7  |     | 4 ?  |
| ex7_2_9                        | 10 | 7   | ?    | disco2                         | 28 | 23? |      |
| ex7_3_4                        | 13 | 18? |      | dixchlng                       | 10 |     | 5 ?  |
| ex7_3_5                        | 14 | 16? |      | himmelbk                       | 24 | 14  | ?    |
| ex8_4_4                        | 18 | 13  | ?    | hs088                          | 33 | 32  | ?    |
| ex8_4_5                        | 16 | 12  | ?    | hs093                          | 7  |     | 3 ?  |
| ex8_5_1                        | 7  | 5   | ?    | hs113                          | 11 |     | 4 si |
| ex8_5_2                        | 7  | 5   | ?    | hs119                          | 17 |     | 9 ?  |
| ex8_5_6                        | 7  | 5   | ?    | accesorios                     | 10 |     | 6 si |
| ex14_1_7                       | 11 |     | 18 ? | pentágono                      | 6  | 12? |      |

La información de convexidad se extrajo del material complementario de [3] (los casos sin información de convexidad se denotan con (?))

La Tabla 2 presenta una comparación de las diferentes estrategias. Resultados para tres valores diferentes. de son reportados. Cada celda muestra el tiempo de CPU consumido en segundos (arriba) y el número de nodos generados (abajo) por una estrategia en la instancia dada para el valor correspondiente de

Las últimas filas de cada columna corresponden a la ganancia (en tiempo relativo y nodos) de cada estrategia wrt XTaylor&S considerando todo el conjunto de instancias. Debido a algo estocástico mecanismos utilizados por lbexOpt,5 cada estrategia se ejecutó 5 veces en cada instancia y el Los resultados promedio se presentan en la tabla. Sólo los casos con grandes diferencias horarias entre se informan las estrategias (es decir, casos tales que el cociente entre el mayor y el mayor) El tiempo más pequeño para cierta precisión es mayor que 2). Destacamos en negrita el mejor tiempo de CPU y número mínimo de nodos alcanzados para alguna estrategia para cada instancia y precisión. Resultados para pares instancia-precisión donde se indican todas las estrategias que tardaron menos de 0,5 s con un guión (–).

En comparación con XTaylor&S, AbsTaylor&S reporta mejores resultados cuando es grande. Para Por ejemplo, cuando = 10–2, la ganancia en tiempo relativo (resp. en nodos) es 1,30 (resp. 1,48). En varios casos se informan reducciones significativas tanto en el tiempo como en los nodos (por ejemplo, ex8\_4\_4,

5 Por ejemplo, de manera similar a XTaylor, XNewton genera relajaciones lineales de las restricciones eligiendo esquinas de la caja como puntos de expansión.



ramsey, srcpm y hs119). Sin embargo, cuando  $n = 10^6$  el método se comporta de manera similar a XTaylor&S ya que obtenemos casi los mismos resultados en términos de tiempo de CPU con una mejora marginal en el tamaño del árbol de búsqueda.

Por otro lado, a pesar del costo adicional en tiempo de CPU que supone realizar más linealizaciones y ejecuciones simplex, TwiceTaylor&S supera a XTaylor&S y AbsTaylor&S para valores grandes  $n$ . Cuando  $n = 10^6$ , TwiceTaylor&S informa una reducción importante en el número de nodos (ganancia de 1,13 frente a XTaylor&S), sin embargo, esta ganancia se ve mitigada por la costo relativamente caro del procedimiento.

## Eficacia de ABSTAYLOR

Además, evaluamos la efectividad de AbsTaylor y XTaylor contando el Número de veces que cada método contribuye a encontrar nuevos límites superiores dentro de TwiceTaylor. acercarse. Observamos que, en promedio, AbsTaylor contribuye a encontrar nuevos límites superiores. El 63% de las veces y XTaylor sólo el 24%. También contamos el número de veces que cada La estrategia tuvo éxito al encontrar una región factible en un nodo del árbol de búsqueda. abstaylor tuvo éxito en el 20% de los casos, mientras que XTaylor tuvo éxito sólo en el 5% de ellos (nosotros no consideró 17 casos en los que ambas estrategias siempre tuvieron éxito).

## 4 ITERATIVETAYLOR: un método de búsqueda local iterativo basado en información interna regiones

En esta sección proponemos IterativeTaylor, un método iterativo simple para encontrar resultados prometedores. soluciones factibles en una caja  $x$ . A partir de un punto inicial  $x$  (por ejemplo, el punto medio de la caja  $x$ ), IterativeTaylor primero intenta encontrar una solución factible candidata  $xc$  en  $x$  usando AbsTaylor&S con  $xc$  como punto de expansión de la linealización (ver Algoritmo 1).

Si tiene éxito, entonces el algoritmo realiza un bucle mientras se busca la solución candidata. mejorado. En cada iteración, se reduce el tamaño de  $x$  y se busca una nueva solución factible  $xc$  en el nuevo cuadro aplicando AbsTaylor y el método simplex. Tenga en cuenta que lo anterior encontrado La solución  $x$  se utiliza como punto de expansión en la iteración actual. El algoritmo continúa p hasta que el coste de la nueva solución  $xc$  no mejore la anterior más que  $\alpha$ ) mueve la caja  $x$

La función `reducir_box(x, xpag, centrándola en el punto  $x$  ypag) reduce su tamaño: el tamaño de cada intervalo se establece en una fracción del tamaño anterior, es decir,  $wid(x_i) \leftarrow \alpha \cdot wid(x_i)$  para todo  $i = 1 \dots n$ , con  $\alpha \in [0, 1[$  un parámetro definido por el usuario. el proporcional La reducción del tamaño de la caja fuerza la complejidad del peor de los casos en el número de pasos del todo el procedimiento es logarítmico.`

Tenga en cuenta que XTaylor no se puede utilizar para encontrar regiones internas en el bucle while porque  $x_p$  no es una esquina de la caja sino el punto medio.

## experimentos

Como segunda serie de experimentos, implementamos IterativeTaylor como límite superior procedimiento en el optimizador de ramas y enlaces lbexOpt. Agregamos una condición adicional en línea 3 para entrar en el bucle: la primera solución candidata  $xc$  tiene que mejorar la mejor solución encontrada por el optimizador hasta ahora. La configuración de los experimentos es la misma que la de la primera serie de experimentos.

[illegible]

123

```

1 procedimiento IterativeTaylor ( f , g , x , x p , , α); fuera: xc
2   xc ← AbsTaylor&S( f , g , x , x p);
3   mientras xc existe y f (xc) < f (x p) – hacer
4     xneg ← xc;
5     x ← reducir_box(x , x α)pag.;
6     xc ← AbsTaylor&S( f , g , x , x p);
7   volver xc

```

Algoritmo 1: el algoritmo IterativeTaylor para encontrar soluciones prometedoras

## ITERATIVETAYLOR versus TWICETAYLOR&S

La Tabla 3 informa los resultados obtenidos por TwiceTaylor&S e IterativeTaylor utilizando el Los dos mejores valores de  $\alpha$  obtenidos a través de una serie de experimentos no reportados.<sup>6</sup> Las últimas filas informe la ganancia en tiempo relativo y nodos de cada estrategia y precisión con TwiceTaylor&S. Hemos resaltado en negrita los mejores resultados. Resultados donde se gastaron todas las estrategias. menos de 0,5 s se indican con un guión (–).

Primero observe que en ambos casos, es decir, cuando  $\alpha = 0,95$  o  $\alpha = 0,5$ , obtenemos importantes ganancias en tiempo y nodos. Los mejores resultados se obtienen cuando  $\alpha = 0,5$ . Esto probablemente se debe a que el algoritmo 1 converge rápidamente para valores pequeños de  $\alpha$ . Cuando  $\alpha = 10^{-6}$ , IterativeTaylor( $\alpha = 0,5$ ) supera a TwiceTaylor&S con ganancias en tiempo y nodos de 1,33 y 1,10 respectivamente.

## Usando TWICETAYLOR&S para encontrar el candidato inicial

Como última serie de experimentos, reemplazamos la llamada a AbsTaylor&S por una llamada a TwiceTaylor&S en la línea 2 del Algoritmo 1. De esta manera tenemos más posibilidades de encontrar una inicial solución candidata factible xc. La nueva variante (IterativeTaylor2) informa una ganancia en el tiempo relativo con respecto a IterativeTaylor de 1,20, 1,09 y 1,16 para los valores de precisión  $10^{-2}$ ,  $10^{-4}$ . y  $10^{-6}$  respectivamente. En número de cajas, las ganancias fueron de 1,22, 1,11 y 1,08 respectivamente.

La Figura 3 muestra un perfil de desempeño [26] que compara las estrategias con  $\alpha = 10^{-6}$ . Cada curva informa la proporción de instancias resueltas por la estrategia correspondiente en menos que o igual a factor veces el mejor tiempo de CPU informado. Tenga en cuenta que todas las propuestas superan XTaylor&S, la actual estrategia por defecto utilizada por IbexOpt. IterativeTaylor2 informa el mejores resultados en la comparación.

Finalmente, la Fig. 4 muestra una comparación informativa entre IbexOpt, utilizando Iterative Taylor2, y otros optimizadores globales conocidos. Los resultados reportados por otros solucionadores. se obtuvieron del material complementario de [3].<sup>7</sup> Cada curva informa la proporción de instancias resueltas por la estrategia correspondiente en menos de una cierta cantidad de tiempo medido en segundos. La muestra de instancias de benchmarks utilizadas en esta comparación corresponde a todos las instancias NCOP convexas no reportadas de la página web COCONUT (130 instancias).

En la figura podemos ver que IbexOpt, utilizando IterativeTaylor2, es bastante competitivo. con los mejores optimizadores globales (Baron y Antigone) en problemas gastando más de 300 segundos de tiempo de CPU. Tenga en cuenta que, a excepción de IbexOpt, todos los solucionadores comparados son

<sup>6</sup> Los otros valores probados de  $\alpha$  fueron 0,1, 0,2, 0,7, 0,8, 0,9 y 0,99.

<sup>7</sup> Se ejecutaron en una computadora con un procesador Intel Xeon X5650 de 2,67 GHz con 24 GB de RAM en una computadora de 64 bits. Versión Linux.

123

[illegible]

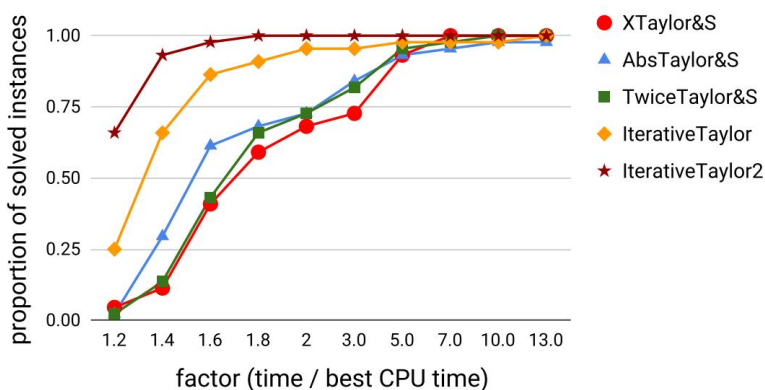


Fig. 3 Perfil de rendimiento. Comparación entre los resultados reportados por XTaylor&S y las estrategias propuestas basadas en AbsTaylor con  $\epsilon = 10^{-6}$ . Para una estrategia determinada, un punto  $(f, p)$  en la curva correspondiente indica que una proporción  $p$  de las instancias se resolvió en menos de  $f$  veces el mejor tiempo de CPU informado.

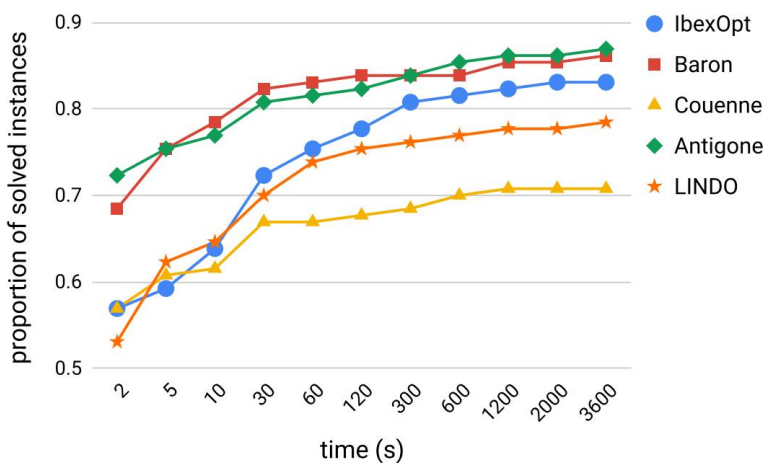


Fig. 4 Perfil de rendimiento. Para una estrategia dada, un punto  $(t, p)$  en la curva correspondiente indica que una proporción  $p$  de los 130 casos se resolvió en menos de  $t$  segundos.

no son rigurosos, es decir, no pueden garantizar sus resultados. Otros solucionadores rigurosos, como GlobSol e Icos, no son nada competitivos, por lo que no se incluyeron en la comparación.

## 5. Conclusiones

En este trabajo presentamos AbsTaylor, una nueva técnica de linealización interna basada en Taylor que puede usarse para encontrar soluciones factibles de NCOP. Además, hemos presentado IterativeTaylor, un algoritmo de búsqueda local iterativo simple que utiliza regiones internas y simplex para encontrar soluciones prometedoras.

Los experimentos muestran resultados prometedores. Por un lado, la combinación de nuestro enfoque AbsTaylor y el último modelo XTaylor parece mejorar significativamente el rendimiento de los optimizadores de intervalos de rama y límite, especialmente cuando se requiere una precisión baja.

para los resultados. Además, IterativeTaylor e IterativeTaylor2 reducen aún más el tiempo de CPU invertido por el optimizador para cualquier precisión requerida.

Como trabajo futuro planeamos mejorar el algoritmo de búsqueda local incorporando un procedimiento de optimización global que busca la solución óptima en el segmento de línea factible que une la solución candidata actual con la siguiente. También planeamos comparar IterativeTaylor con otros enfoques iterativos básicos, como la búsqueda de líneas y los métodos de región de confianza [19,20].

Agradecimientos Este trabajo cuenta con el apoyo del Proyecto Fondecyt 1160224. Víctor Reyes cuenta con el apoyo de la Beca Postgrado PUCV 2018.

## Referencias

1. Trombettoni, G., Araya, I., Neveu, B., Chabert, G.: Regiones internas y linealizaciones de intervalos para la optimización global. En: Conferencia AAAI sobre Inteligencia Artificial, (2011)
2. Sahinidis, NV: Baron: un paquete de software de optimización global de propósito general. J. Globo. Óptimo. 8(2), 201–205 (1996)
3. Misener, R., Floudas, CA: Antigone: algoritmos para la optimización global continua/entera de no lineales ecuaciones. J. Globo. Óptimo. 59, 1–24 (2013)
4. Schrage, L.: Programación lineal, entera y cuadrática con LINDO. La prensa científica, Singapur (1986)
5. Belotti, P.: Couenne: un manual de usuario. Informe técnico, Universidad de Lehigh, Tech. Rep. (2009)
6. Lebbah, Y.: ICOS: un solucionador basado en ramas y límites para una optimización global rigurosa. Óptimo. Métodos Softw. 24(4–5), 709–726 (2009)
7. Kearfott, RB: Problemas continuos de búsqueda global rigurosa. Springer, Berlín (1996)
8. Araya, I., Reyes, V.: Algoritmos de rama y límite de intervalo para optimización y satisfacción de restricciones: una encuesta y perspectivas. J. Globo. Óptimo. 65(4), 837–866 (2016)
9. Araya, I., Trombettoni, G., Neveu, B., Chabert, G.: Límite superior en regiones internas para la optimización global bajo restricciones de desigualdad. J. Globo. Óptimo. 60(2), 145–164 (2014)
10. Ninin, J., Messine, F., Hansen, P.: Un método confiable de relajación afín para la optimización global. 4OR 13(3), 247–277 (2015)
11. Goldsztejn, A., Lebbah, Y., Michel, C., Rueher, M.: Revisando el proceso de límite superior en una rama segura y un algoritmo de límite. En: Beck, JC (ed.) Principios y práctica de la programación con restricciones, págs. 598–602. Springer, Berlín (2008)
12. Drud, AS: CONOPT: un código GRG a gran escala. ORSA J. Computación. 6(2), 207–216 (1994)
13. Wächter, A., Biegler, LT: Sobre la implementación de un algoritmo de búsqueda de líneas de filtro de punto interior para programación no lineal a gran escala. Matemáticas. Programa. 106(1), 25–57 (2006)
14. Khajavirad, A., Sahinidis, NV: Un paradigma híbrido LP/NLP para relajaciones de optimización global. Matemáticas. Programa. Computadora. 10(3), 383–421 (2018)
15. Lasdon, LS, Waren, AD, Jain, A., Ratner, M.: Diseño y prueba de un código de gradiente reducido generalizado para programación no lineal. Laboratorio de optimización de sistemas CA de la Universidad de Stanford, informe técnico (1976)
16. Wright, SJ: Métodos de punto interior dual-primario, vol. 54. SIAM, Filadelfia (1997)
17. Kuhn, H., Tucker, A.: Programación no lineal. En: Segundo Simposio de Berkeley sobre estadística matemática Tics y probabilidad, págs. 481–492 (1951)
18. Nocedal, J., Wächter, A., Waltz, RA: Estrategias de actualización de barreras adaptativas para métodos interiores no lineales. SIAM J. Optim. 19 (4), 1674–1693 (2009)
19. Yuan, Y.-X.: Avances recientes en algoritmos de regiones de confianza. Matemáticas. Programa. 151(1), 249–281 (2015)
20. Coleman, TF, Li, Y.: Un enfoque de región de confianza interior para la minimización no lineal sujeta a límites. SIAM J. Optim. 6(2), 418–445 (1996)
21. Omojokun, EO: Algoritmos de región de confianza para la optimización con igualdad no lineal y desigualdad tensiones. Tesis doctoral, Universidad de Colorado (1989)
22. Araya, I., Neveu, B.: Ismeat: una estrategia de selección de variables para solucionadores de rama de intervalo y límite. J. Globo. Óptimo. 71, 1–18 (2017)
23. Benhamou, F., Goulard, F., Granvilliers, L., Puget, J.-F.: Revisión de la consistencia del casco y la caja. En: Interna Congreso Nacional sobre Programación Lógica. Citador (1999)



24. Trombettoni, G., Chabert, G.: Disyunción de intervalo constructivo. En: Bessiere, C. (ed.) Principios y práctica de la programación con restricciones-CP 2007, págs. 635–650. Springer, Berlín (2007)
25. Araya, I., Trombettoni, G., Neveu, B.: Un contratista basado en el intervalo convexo de Taylor, en: Integración de técnicas de IA y OR en programación de contraintes para problemas de optimización combinatoria, págs. Saltador (2012)
26. Dolan, ED, Moré, JJ: Software de optimización de evaluación comparativa con perfiles de rendimiento. Matemáticas. Programa. 91(2), 201–213 (2002)

Nota del editor Springer Nature se mantiene neutral con respecto a reclamos jurisdiccionales en mapas publicados y afiliaciones institucionales.