



FACULTADE DE MATEMÁTICAS

Traballo Fin de Grao

# Técnicas de reducción de dominio en optimización polinómica

Pablo Rodríguez Fernández

2022/2023

UNIVERSIDADE DE SANTIAGO DE COMPOSTELA



GRAO DE MATEMÁTICAS

Traballo Fin de Grao

# Técnicas de reducción de dominio en optimización polinómica

Pablo Rodríguez Fernández

Julio, 2023

UNIVERSIDADE DE SANTIAGO DE COMPOSTELA



# Trabajo propuesto

<b>Área de Conocimiento: Estadística e Investigación Operativa</b>
<b>Título: Técnicas de reducción de dominio en optimización polinómica</b>
<b>Breve descripción del contenido</b>
<p>A la hora de buscar óptimos globales en problemas de optimización matemática no convexos, la mayoría de los algoritmos suelen basarse en técnicas de ramificación y acotación. De manera directa o indirecta, estas técnicas buscan subdividir la región factible del problema original en subregiones, de tal manera que al final se pueda explorar toda la región factible del problema original mediante la resolución de subproblemas más sencillos.</p> <p>El rendimiento de estas técnicas depende fuertemente del tamaño del dominio original y, por tanto, es habitual complementarlas con técnicas de reducción de dominio. En este trabajo se revisarán las principales técnicas de reducción de dominio existentes y se estudiará su particularización a problemas de optimización polinómica.</p>
<b>Recomendaciones</b>
No hay ninguna recomendación en particular.



# Índice

<b>Resumen</b>	<b>VII</b>
<b>Introducción</b>	<b>IX</b>
<b>1. Conceptos previos</b>	<b>1</b>
1.1. Conceptos básicos de optimización . . . . .	1
1.2. Análisis Convexo . . . . .	5
1.3. Ramificación y acotación . . . . .	7
<b>2. Técnicas de reducción de dominio</b>	<b>11</b>
2.1. Marco teórico . . . . .	12
2.2. Desigualdades de reducción de dominio . . . . .	17
2.2.1. Otras técnicas de reducción de dominio . . . . .	20
2.3. El algoritmo de ramificación y reducción . . . . .	21
2.3.1. Convergencia de ramificación y reducción . . . . .	26
<b>3. Optimización polinómica</b>	<b>29</b>
3.1. Ramificación y reducción en optimización polinómica . . . . .	30
3.1.1. Polinomios univariables . . . . .	30
3.1.2. Problemas lineales multiplicativos . . . . .	34
3.2. El algoritmo RLT . . . . .	36

<b>4. Resultados experimentales</b>	<b>41</b>
4.1. RAPOSa . . . . .	41
4.2. Estudio computacional . . . . .	43
<b>Bibliografía</b>	<b>47</b>

## **Resumen**

El objetivo de este trabajo es estudiar diversas técnicas de reducción de dominio empleadas en la optimización global y particularizarlas a problemas de optimización polinómica. Junto dichas técnicas se presentarán diversos algoritmos que permiten emplearlas para mejorar el rendimiento en la resolución de estos problemas, incluyendo un estudio computacional para comprobar su eficacia en problemas reales.

## **Abstract**

The aim of this work is to study several domain reduction techniques used in global optimisation and to particularise them to polynomial optimisation problems. Together with these techniques, several algorithms will be presented that allow them to be used to improve the performance in the resolution of these problems, including a computational study to check the effectiveness of these techniques in real problems.



# Introducción

Al intentar resolver un problema de optimización, una de las mayores dificultades es que el conjunto en el que se debe buscar la solución óptima suele ser muy grande. Muchos algoritmos, incluido el conocido algoritmo de *Ramificación y Acotación*, buscan dividir este espacio en regiones más pequeñas e ir descartando aquellas en las que se sabe, por características del problema o de las otras regiones presentes, que no puede encontrarse dicha solución óptima.

Sin embargo, en ocasiones estas regiones siguen teniendo un tamaño demasiado grande, provocando que la resolución de un problema requiera mucho tiempo llegando incluso a ser inviable su resolución por no disponer del tiempo suficiente. En estos casos existen diversos métodos que buscan reducir el tamaño de estas regiones, disminuyendo el espacio en el cual se busca la solución.

En este trabajo se presentarán diversas técnicas que permiten reducir el espacio de búsqueda de solución óptima en problemas de optimización, a fin de facilitar su resolución o acelerarla. Junto con ellas se explicarán varios algoritmos que permiten aprovechar estas mejoras al máximo, bien sea tomándolas como núcleo central de su funcionamiento o añadiéndolas como un paso adicional para mejorar el rendimiento de otros procedimientos que ya se contemplan.

Si bien la mayoría de estas técnicas pueden aplicarse en un conjunto muy amplio de problemas, en esta memoria se particularizarán para los de optimización polinómica, que son aquellos definidos mediante polinomios. Esto puede parecer una simplificación muy grande, pero existen una gran cantidad de situaciones en el mundo real que pueden ser modelizadas mediante el uso de este tipo de problemas: desde el diseño de microchips hasta la elección de rutas de transporte, pasando por el control de plantas químicas o el análisis de mercados. Esto hace que la optimización polinómica sea un campo de investigación muy activo en la actualidad, pues sus aplicaciones en ámbitos tan diversos provocan un gran interés en mejorar los algoritmos utilizados para su resolución.



# Capítulo 1

## Conceptos previos

En este primer capítulo se presentarán los principales conceptos de optimización matemática necesarios para entender el resto del trabajo. También se tratarán algunos aspectos de análisis convexo, ya que las funciones convexas juegan un papel muy importante en muchos métodos de resolución de problemas de optimización, con lo cual será necesario tener unas nociones básicas. Tanto esta parte como la anterior se basan en [González Díaz \(2022b\)](#) y en [González Díaz \(2022c\)](#), así como en conocimientos básicos generales adquiridos a lo largo del grado.

Finalmente se presentará el algoritmo de Ramificación y Acotación, que es uno de los más importantes para la resolución de problemas de optimización, ya que muchos otros algoritmos lo toman como base. Se introducirán junto con él una serie de conceptos y resultados que se utilizarán de manera recurrente a lo largo del trabajo, como el de ajuste de cotas o el de consistencia. Esta parte se basa principalmente en [Horst and Tuy \(1993\)](#) y, en menor medida, en [Ryoo and Sahinidis \(1996\)](#).

### 1.1. Conceptos básicos de optimización

**Definición 1.1.** Un *problema de optimización* viene dado por un par  $(F, c)$ , donde  $F$  es el dominio de puntos factibles, *dominio factible* o *conjunto factible*, y  $c$  es la función de coste. El problema consiste en encontrar un punto factible  $x \in F$  tal que,

$$\forall y \in F, \quad c(x) \leq c(y).$$

Cualquier punto  $x$  que verifique la condición anterior se denomina *óptimo global* del problema. Si un punto  $x'$  cumple la definición anterior pero solamente en  $D \subsetneq F$ , se dirá que  $x'$  es un *óptimo local* del problema en conjunto  $D$ . El proceso de encontrar un óptimo se conoce como *resolver* el problema, y dependiendo del contexto bastará con un óptimo local o será necesario uno global.

La dificultad de resolver un problema de optimización dependerá de cómo sea su función de coste y su dominio factible. Puede suceder que un problema, a pesar de tener una función de coste muy sencilla, sea muy complejo debido a como es su dominio factible. O puede darse el caso contrario, donde se tenga un dominio muy sencillo pero una función de coste con muchos cambios, provocando que identificar si un óptimo es local o global sea muy complejo.

**Ejemplo 1.2.** Supóngase la siguiente función de coste:  $c(x) = \cos(x) + \ln(x)$ . La gráfica asociada a esta función puede verse en la Figura 1.1. En esta gráfica también se representan dos dominios factibles  $P1$  (en violeta) y  $P2$  (en azul). Si se intenta encontrar el óptimo dentro de la región  $P1$ , claramente será el punto  $C$ . Sin embargo, si se amplía esta región hasta  $P2$ , el óptimo será  $A$ . Y si aún se va más allá y se toma como dominio factible todo  $\mathbb{R}^2$ , se puede demostrar que el óptimo tenderá a  $-\infty$  (ni siquiera el punto  $B$ , aunque tome un valor más pequeño que  $A$  y  $C$ ).

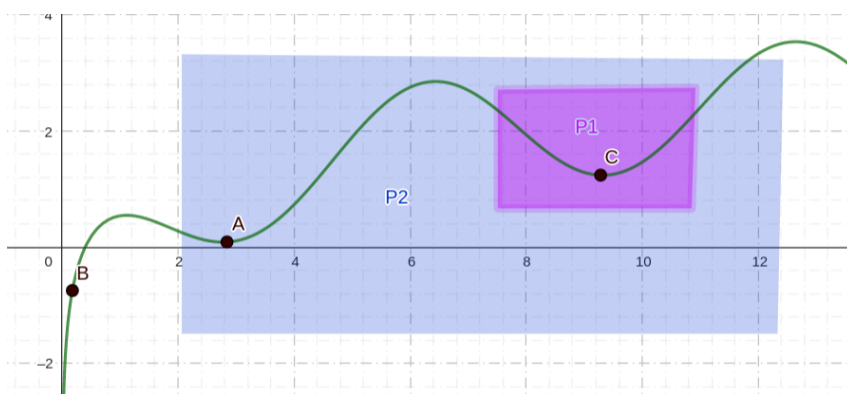


Figura 1.1: Función de coste  $c(x)$  y dominios factibles.

A continuación, se presenta la formulación general de un problema de optimización matemática, que se utilizará recurrentemente a lo largo de todo el trabajo. Como dicha formulación emplea vectores, para facilitar la lectura se seguirá el convenio de denotar a los vectores  $\mathbf{x}$  de  $\mathbb{R}^n$  en negrita.

**Definición 1.3.** Un problema de optimización matemática consiste en encontrar una solución al problema

$$\begin{aligned} & \text{minimizar}_{\mathbf{x} \in \mathbb{R}^n} && c(\mathbf{x}) \\ & \text{sujeto a} && g_i(\mathbf{x}) \leq 0 \quad i = 1, \dots, m \\ & && h_j(\mathbf{x}) = 0 \quad j = 1, \dots, l, \end{aligned} \tag{1.1}$$

donde  $c$  y todas las  $g_i$  y  $h_j$  son funciones.

**Definición 1.4.** Se denomina *restricción* a cada una de las desigualdades o igualdades asociadas a las funciones  $g_i$  y  $h_j$  que aparecen en un problema de optimización matemática.

**Definición 1.5.** Se denomina *restricción de rango* a una desigualdad que afecta a una variable  $x_j$  y no depende del valor de una función, si no de una constante. Si la desigualdad es de tipo  $x_j \leq cte$  se denota la constante como  $x_j^U$  y se dice que es la *cota superior* de la variable. Si por el contrario es de tipo  $x_j \geq cte$ , se denota por  $x_j^L$  y se dice que es la *cota inferior* de la variable.

**Ejemplo 1.6.** Utilizando la nomenclatura y notación anterior, la desigualdad  $2x_1 + 3x_2 \leq 5$  sería una restricción, con  $g(x_1, x_2) = 2x_1 + 3x_2$ . La desigualdad  $x_1 \geq 54$  sería una restricción de rango, con  $x_1^L = 54$  su cota inferior.

**Definición 1.7.** Se dice que una restricción está *activa* o que se encuentra *saturada* en un óptimo si al evaluar la desigualdad en dicho punto se da la igualdad. Es decir, una restricción activa alcanza el máximo valor permitido en el problema para ella.

**Ejemplo 1.8.** Considérese la primera restricción del ejemplo anterior,  $2x_1 + 3x_2 \leq 5$ . Si esta está incluida en un problema de optimización en el cual el punto  $(x_1, x_2) = (1, 1)$  es un óptimo, la restricción estaría activa en dicho punto, ya que si se evalúa, se tiene que  $2x_1 + 3x_2 = 2 \cdot 1 + 3 \cdot 1 = 5$ .

**Definición 1.9.** El *dominio* o *conjunto factible* asociado a un problema de optimización matemática viene dado por aquellos puntos que verifican las restricciones asociadas a las funciones  $g_i$  y  $h_j$ .

*Observación 1.10.* Maximizar una función  $c(\mathbf{x})$  es equivalente a minimizar la función  $-c(\mathbf{x})$ . Por convenio, se trabajará solamente con problemas de minimización.

*Observación 1.11.* Si bien la formulación presentada en la definición 1.3 no incluye explícitamente restricciones del tipo  $g_i(\mathbf{x}) \geq 0$ , estas pueden ser reformuladas para adaptarse a este esquema. Basta con reescribirlas como  $f_i(\mathbf{x}) \leq 0$ , con  $f_i(\mathbf{x}) = -g_i(\mathbf{x})$ .

También es posible escribir el problema solamente empleando restricciones  $g_i(\mathbf{x}) \leq 0$ , sustituyendo la restricción  $h_j(\mathbf{x}) = 0$  por las restricciones  $h_j(\mathbf{x}) \leq 0$  y  $-h_j(\mathbf{x}) \leq 0$ . Esta representación será muy útil más adelante.

**Ejemplo 1.12.** Considérese el siguiente problema de optimización:

$$\begin{aligned} \text{minimizar} \quad & 5x_1 + 3x_2^2 - 5x_1x_2 \\ \text{sujeto a} \quad & x_1 + x_2 \geq 3 \\ & x_1 - x_2 \leq 0 \\ & x_1 + x_2^2 = 8. \end{aligned} \tag{1.2}$$

Para expresarlo como el la Definición 1.3, basta con restar los lados derechos en cada restricción e invertir la desigualdad en la primera restricción:

$$\begin{aligned}
&\text{minimizar} && 5x_1 + 3x_2^2 - 5x_1x_2 \\
&\text{sujeto a} && -x_1 - x_2 + 3 \leq 0 \\
&&& x_1 - x_2 \leq 0 \\
&&& x_1 + x_2^2 - 8 = 0.
\end{aligned} \tag{1.3}$$

También pueden expresarse todas las restricciones como restricciones de tipo  $\leq$  si se desglosa la última restricción de igualdad como se explica en la observación anterior:

$$\begin{aligned}
&\text{minimizar} && 5x_1 + 3x_2^2 - 5x_1x_2 \\
&\text{sujeto a} && -x_1 - x_2 + 3 \leq 0 \\
&&& x_1 - x_2 \leq 0 \\
&&& x_1 + x_2^2 - 8 \leq 0 \\
&&& -x_1 - x_2^2 + 8 \leq 0.
\end{aligned} \tag{1.4}$$

Para un problema de optimización matemática es posible definir un *problema dual* asociado. Existen diferentes tipos de duales en función de como sea el problema de partida, estudiándose en el grado el dual de los problemas lineales en la asignatura *Programación Lineal y Entera*. En este caso, al tratar con problemas más amplios que no tienen por qué ser lineales, será interesante presentar brevemente el denominado *dual lagrangiano*, pues se trabajará con uno de sus elementos durante el Capítulo 2. Se recomienda consultar [González Díaz \(2022c\)](#) para una descripción más detallada de esta construcción y sus propiedades.

Considérese el siguiente problema de optimización, ligera variación de 1.1 y que se tomará como problema *primal*:

$$\begin{aligned}
&\text{minimizar}_{\mathbf{x} \in \mathbb{R}^n} && c(\mathbf{x}) \\
&\text{sujeto a} && g_i(\mathbf{x}) \leq 0 \quad i = 1, \dots, m \\
&&& h_j(\mathbf{x}) = 0 \quad j = 1, \dots, l. \\
&&& \mathbf{x} \in S,
\end{aligned} \tag{1.5}$$

donde el conjunto  $S$  viene dado por restricciones "fáciles", como restricciones de rango o lineales. Partiendo de este problema es posible definir su *función lagrangiana* asociada:

$$L(\mathbf{x}, \mathbf{u}, \mathbf{v}) = c(\mathbf{x}) + \sum_{i=1}^m u_i g_i(\mathbf{x}) + \sum_{j=1}^l v_j h_j(\mathbf{x}). \tag{1.6}$$

Las variables  $\mathbf{u}$  y  $\mathbf{v}$  se llaman *variables duales* o, realizando un pequeño abuso de notación, *multiplicadores de Lagrange* (pues este concepto es ligeramente diferente pero para este trabajo se pueden considerar análogos). A partir de esta función es posible plantear el *dual lagrangiano* del problema 1.1:

$$\begin{aligned}
&\text{maximizar}_{\mathbf{u}, \mathbf{v}} && \inf_{\mathbf{x} \in S} L(\mathbf{x}, \mathbf{u}, \mathbf{v}) \\
&\text{sujeto a} && \mathbf{u} \geq 0.
\end{aligned} \tag{1.7}$$

La resolución del problema dual permite extraer muchas propiedades del primal, siendo la base de construcción de una gran cantidad de técnicas de resolución. En particular, los multiplicadores de Lagrange guardan una gran relación con la solución óptima del problema primal, lo cual será aprovechado en el Capítulo 2 para construir diversos métodos que aprovechen esta información para reducir el tamaño del conjunto factible del problema de partida.

## 1.2. Análisis Convexo

El análisis convexo es una rama que estudia los conjuntos y las funciones convexas. Los objetos con esta propiedad son muy interesantes desde el punto de vista de la optimización, pues gozan de determinadas propiedades que facilitan la resolución de problemas. A continuación, se presentarán los conceptos básicos de convexidad que se utilizarán en este trabajo, que si bien no son muchos, tendrán una gran importancia.

**Definición 1.13.** Dados dos puntos  $\mathbf{x}$  e  $\mathbf{y}$  en  $\mathbb{R}^n$  y dado  $\lambda \in [0, 1]$ , se dice que el punto  $\lambda\mathbf{x} + (1 - \lambda)\mathbf{y}$  es una *combinación convexa* de  $\mathbf{x}$  e  $\mathbf{y}$ . Si cualquier combinación convexa de puntos de un conjunto  $S \subset \mathbb{R}^n$  pertenece también a  $S$ , se dirá que dicho conjunto es un *conjunto convexo*.

**Definición 1.14.** Una función  $f$  es *lineal* si, para cualquier par de puntos  $\mathbf{x}$  e  $\mathbf{y}$  en  $\mathbb{R}^n$  y para cualquier par de escalares  $a, b \in \mathbb{R}$ ,  $f(a\mathbf{x} + b\mathbf{y}) = af(\mathbf{x}) + bf(\mathbf{y})$ .

**Definición 1.15.** Una función  $f$  es *convexa* si, para cualquier par de puntos  $\mathbf{x}$  e  $\mathbf{y}$  en  $\mathbb{R}^n$  y para cualquier  $\lambda \in [0, 1]$ ,  $f(\lambda\mathbf{x} + (1 - \lambda)\mathbf{y}) \leq \lambda f(\mathbf{x}) + (1 - \lambda)f(\mathbf{y})$ .

**Definición 1.16.** Un función  $f$  se dice *cóncava* si  $-f$  es una función convexa.

Partiendo de las definiciones anteriores, es posible definir un nuevo tipo de problema de optimización, el cual es de gran importancia para la resolución de problemas de mayor complejidad.

**Definición 1.17.** Un *problema de optimización matemática convexa* es un problema de optimización matemática en el que las funciones  $c$  y  $g_i$  son convexas, y las funciones  $h_j$  son lineales. Es decir, tenemos un problema donde la función objetivo y todas las restricciones de desigualdad vienen definidas por funciones convexas, y las restricciones de igualdad, por funciones lineales.

**Ejemplo 1.18.** El problema de optimización del Ejemplo 1.12 no es un problema de optimización convexa, pues la función  $h(x_1, x_2) = x_1 + x_2^2 - 8$  no es una función lineal.

**Ejemplo 1.19.** Sea el siguiente problema de optimización:

$$\begin{aligned}
 &\text{minimizar} && 5x_1 + 3x_2 - 5x_3 \\
 &\text{sujeto a} && -x_1 - x_2 - 3 \leq 0 \\
 &&& x_1 - x_2 \leq 0 \\
 &&& x_1 + x_2 + 24x_3 - 8 = 0.
 \end{aligned} \tag{1.8}$$

Este problema sí es un problema convexo, pues su función objetivo y las funciones asociadas a las restricciones de desigualdad son convexas; y la asociada a la restricción de igualdad, lineal.

La razón de que los problemas convexas sean ampliamente estudiados es que poseen propiedades que facilitan su resolución respecto otro tipo de problemas. Una de ellas, la cual se demostrará a continuación, es que todo óptimo local es también un óptimo global. Muchas técnicas de resolución de problemas de optimización emplean este hecho para aproximar un problema por una serie de problemas convexas (los cuales suelen tener una resolución más sencilla) para ir acotando el valor óptimo global a partir de los óptimos locales de estos problemas convexas.

Para demostrar dicha propiedad, es necesario probar antes un lema que describe cómo es el conjunto factible de un problema convexo.

**Lema 1.20.** *El conjunto factible asociado a un problema de optimización convexa es convexo.*

*Demostración.* Sea  $F$  el conjunto factible asociado a un problema de optimización convexa, y sean  $\mathbf{x}, \mathbf{y} \in F$  y  $\lambda \in [0, 1]$ . Se tiene que probar que  $\mathbf{z} = \lambda\mathbf{x} + (1 - \lambda)\mathbf{y} \in F$ . Dada una función  $h_j$ , asociada a una restricción de igualdad, se tiene que

$$h_j(\mathbf{z}) = h_j(\lambda\mathbf{x} + (1 - \lambda)\mathbf{y}) = \lambda h_j(\mathbf{x}) + (1 - \lambda)h_j(\mathbf{y}) = 0,$$

siendo la primera igualdad consecuencia de que  $h_j$  es lineal, y la segunda de que  $\mathbf{x}, \mathbf{y} \in F$  y, por tanto, cumplen la restricción  $h_j$ . Por tanto,  $\mathbf{z}$  satisface las restricciones dadas por las funciones  $h_j$ .

Dada una función  $g_j$ , asociada a una restricción de desigualdad, se tiene que

$$g_i(\mathbf{z}) = g_i(\lambda\mathbf{x} + (1 - \lambda)\mathbf{y}) \leq \lambda g_i(\mathbf{x}) + (1 - \lambda)g_i(\mathbf{y}) \leq 0,$$

siendo la primera desigualdad consecuencia de la convexidad de  $g_i$ , la segunda de que, como en el caso anterior,  $\mathbf{x}, \mathbf{y} \in F$ . Por tanto,  $\mathbf{z}$  también satisface las restricciones dadas por las funciones  $g_i$ . Luego,  $\mathbf{z} \in F$ , con lo que  $F$  es un conjunto convexo.  $\square$

**Teorema 1.21.** *Dado un problema de optimización convexa, todo óptimo local es un óptimo global.*

*Demostración.* Sea  $\mathbf{x}$  un óptimo local de un problema convexo que no es un óptimo global. Entonces, por la definición de óptimo global, existe otro punto factible  $\mathbf{y}$  tal que  $c(\mathbf{y}) < c(\mathbf{x})$ . El lema anterior asegura que el conjunto factible es convexo, y por tanto, para todo  $\lambda \in (0, 1)$ , se tiene que  $\lambda\mathbf{x} + (1 - \lambda)\mathbf{y} \in F$ . Por otro lado, la convexidad de la función  $c$  provoca que para todo  $\lambda \in [0, 1]$ ,

$$c(\lambda\mathbf{x} + (1 - \lambda)\mathbf{y}) \leq \lambda c(\mathbf{x}) + (1 - \lambda)c(\mathbf{y}),$$

y además, que si  $\lambda \in [0, 1)$ ,

$$\lambda c(\mathbf{x}) + (1 - \lambda)c(\mathbf{y}) < \lambda c(\mathbf{x}) + (1 - \lambda)c(\mathbf{x}) = c(\mathbf{x}).$$

Por tanto, cuanto más cerca esté  $\lambda$  de 1, más cerca estará  $\lambda\mathbf{x} + (1 - \lambda)\mathbf{y}$  de  $\mathbf{x}$ . De esto se deduce que es posible construir una sucesión de puntos en el conjunto factible  $F$  que converja a  $\mathbf{x}$  y cuyos elementos tengan un coste menor de  $c(\mathbf{x})$ , lo que contradice que  $\mathbf{x}$  sea un mínimo local.  $\square$

### 1.3. Ramificación y acotación

La técnica de ramificación y acotación (RyA) es una de las más utilizadas en la búsqueda de óptimos globales en problemas de optimización. Esta técnica busca dividir el conjunto factible sucesivamente en regiones o conjuntos más pequeños (ramificación), y obtener cotas inferiores ( $L$ , de *lower bound*) y superiores ( $U$ , de *upper bound*) de la función objetivo en cada uno de ellos para conseguir acotar la solución del problema original (acotación). Repitiendo este proceso (iterándolo) una gran cantidad de veces se puede llegar a que la cota inferior y la cota superior coincidan (o sean diferentes solamente un  $\delta > 0$  prefijado), obteniendo en dicho caso un óptimo.

Una de las principales ventajas de esta técnica es que permite eliminar regiones enteras del espacio de búsqueda durante su ejecución (podar). Así, para una región  $M$  en la iteración  $k$  del algoritmo, si su cota inferior  $L_M$  es superior a la cota superior global del problema  $U$ , se podrá eliminar dicha región, ya que esto significa que ninguna solución que se pudiese encontrar en la región mejoraría el peor valor que se tiene actualmente para la solución global. De esta manera se reduce el dominio de búsqueda de la solución, acelerando el proceso.

En el Algoritmo 1 se muestra el pseudocódigo de una versión de RyA, ya que existen diferentes implementaciones posibles de este algoritmo. En este trabajo se eligió la de [Horst and Tuy \(1993\)](#), pudiendo encontrar información más detallada en dicha referencia.

Considérese a partir de ahora un algoritmo basado en la técnica de RyA, siguiendo la versión que se acaba de presentar.

**Definición 1.22.** Un algoritmo de RyA se dice *finito* si  $L^{(k)} = U^{(k)}$  para algún  $k < \infty$ .

Si un algoritmo de RyA es finito, al terminar el algoritmo se obtendrá una solución global óptima. En caso de no serlo, se deberá fijar algún límite a partir del cual se acepte una solución aproximada por válida, o se decida que no es posible encontrar una solución en un tiempo razonable.

**Definición 1.23.** Un algoritmo de RyA se dice *convergente* si  $\lim_{k \rightarrow \infty} |U^{(k)} - L^{(k)}| = 0$ .

---

**Algoritmo 1** Ramificación y acotación (iteración  $k$ -ésima)
 

---

**Paso 1: Ramificación**

Se divide la región  $M$  en un número finito de subregiones  $M_i$ , con  $i = 1, \dots, s$

**Paso 2: Acotación**

Para cada subregión  $M_i$ , se determina su cota inferior  $L_i$  y superior  $U_i$ , las cuales deben cumplir que  $\forall x \in M_i, L_i \leq f(x) \leq U_i$

**Paso 3: Eliminación de subregiones**

Si la cota inferior de una subregión  $M_i$  es superior a la cota superior global, es decir, si  $L_i \geq U^{(k-1)}$ , se puede eliminar dicha subregión  $M_i$

**Paso 4: Actualización de la cota global**

Se actualizan las cotas globales del problema a partir de las cotas de las subregiones exploradas en la iteración actual:

$$L^{(k)} = \min\{L_i : i = 1, \dots, s\}; \quad U^{(k)} = \min\{U_i : i = 1, \dots, s\}$$

**Paso 5: Criterio de parada y selección de subregión**

**if**  $L^{(k)} = U^{(k)}$  **then**

Se ha encontrado una solución óptima. **FIN**

**else**

Se selecciona una nueva subregión como región principal

Se cambia de iteración:  $k \leftarrow k + 1$

Se vuelve al Paso 1.

**end if**

---

La condición anterior depende del comportamiento en el límite de la diferencia  $U^{(k)} - L^{(k)}$ , donde los valores de  $U^{(k)}$  y  $L^{(k)}$  dependen de la iteración  $k$  y de las regiones que se hayan explorado hasta el momento. Por esa razón, para estudiar propiedades relacionadas con la convergencia de un esquema de RyA, es natural considerar secuencias decrecientes anidadas dentro de las regiones. Así, a partir de ahora se dirá que  $M_{i_q}$  es un *refinamiento* de una región  $M_i$  si  $M_{i_q} \subseteq M_i$ .

La convergencia de un esquema de RyA depende principalmente de tres operaciones que se realizan en él:

- **Ramificación:** cómo se particiona una región. Por ejemplo, podría dividirse simplemente en 4 subregiones del mismo tamaño, pero por lo general existen heurísticas enfocadas a realizar el particionamiento óptimo para cada caso. Se corresponde con el **Paso 1** del Algoritmo 1.

- **Acotación:** cómo eliminar regiones del problema o reducir su tamaño. Para ello, el primer paso es calcular las cotas superiores e inferiores de una región. Tras esto, se pueden imponer nuevas restricciones basadas en estos valores que ayuden a reducir el tamaño de las regiones a explorar. Se corresponde con los **Pasos 2 y 3** del Algoritmo 1.
- **Selección:** qué subregión se elige como nueva región para ramificar. Lo recomendable es elegir aquella que parezca más prometedora, para lo cual pueden resolverse problemas relajados sobre ellas o extraer propiedades intrínsecas del problema. Se corresponde con el **Paso 5** del Algoritmo 1.

La operación de ramificación tiene una gran influencia en la velocidad de convergencia del algoritmo, pero no puede provocar por sí misma que converja o no, dependiendo esto de las otras dos operaciones. Por esa razón, será necesario imponer ciertas condiciones sobre las operaciones para asegurar que el algoritmo converge a una solución óptima.

**Definición 1.24** (Definición IV.4 de [Horst and Tuy \(1993\)](#)). Una operación de acotación se dice *consistente* si en cada paso es capaz de encontrar un refinamiento de alguna subregión no descartada del dominio factible, y cualquier sucesión decreciente de subregiones refinadas  $\{M_{i_q}\}_{q \in \mathbb{N}}$  de una región  $M$  satisface que sus cotas superior e inferior  $U_{i_q}$  y  $L_{i_q}$  cumplen que  $\lim_{q \rightarrow \infty} |U_{i_q} - L_{i_q}| = 0$ .

De esta manera, si una operación es consistente, en cualquier sucesión de refinamientos de una región que converja a un cierto conjunto límite, la cota inferior de este conjunto límite convergerá a la cota superior de la región inicial del refinamiento.

Ahora bien, que las cotas sobre un refinamiento converjan no asegura que las cotas globales del problema lleguen a converger. Para garantizar la convergencia del esquema de RyA, se deben imponer condiciones también sobre la operación de selección.

**Definición 1.25** (Definición IV.6 de [Horst and Tuy \(1993\)](#)). Una operación de selección se dice que *ajusta cotas* si siempre que se seleccione un región esta incluye al menos un elemento que alcance la cota inferior actual.

Una vez definidas las operaciones anteriores, se puede dar un resultado que asegura la convergencia de RyA.

**Teorema 1.26** (Teorema IV.3 de [Horst and Tuy \(1993\)](#)). *En un algoritmo infinito de RyA, supóngase que la operación de acotación es consistente y la operación de selección ajusta cotas. Entonces, el algoritmo es convergente.*

*Demostración.* Sea  $(P)$  un problema de optimización con una región factible inicial  $M_0$  al cual se le quiere aplicar el algoritmo de RyA. Sea  $M_i$  una subregión que mejore la cota inferior

en la iteración  $k$ -ésima del algoritmo, es decir, que su cota inferior  $L_i$  cumpla que  $L^{(k)} = L_i$ . Siempre es posible encontrar una subregión con esta propiedad ya que la operación de selección, por hipótesis, ajusta cotas. Como esta elección se realiza en una iteración  $k$  finita, al ser el algoritmo infinito, se puede encontrar un número infinito de subregiones  $\{M_l\}_{l \in \mathbb{N}}$  cumpliendo esta propiedad.

Considérese a partir de ahora la familia formada por estas subregiones,  $\mathcal{M} = \{M_l\}_{l \in \mathbb{N}}$ , cumpliendo que para cada  $l$ , existe una iteración  $j$  donde  $L^{(j)} = L_l$ . Por el razonamiento anterior, existirá una subregión  $M_{k_0}$  dentro de la región inicial  $M_0$  con infinitas subregiones dentro de  $\mathcal{M}$ . A su vez, dentro de  $M_{k_0}$ , existirá una subregión  $M_{k_1}$  con infinitas subregiones dentro de  $\mathcal{M}$ . Continuando con este proceso se puede observar que existe un refinamiento de subregiones  $\{M_{k_q}\}_{q \in \mathbb{N}}$  donde cada elemento  $M_{k_q}$  tiene infinitas subregiones en  $\mathcal{M}$ .

Sea  $\tilde{M}_{k_q}$  una subregión de  $M_{k_q}$  que está en  $\mathcal{M}$ , y sea  $\tilde{L}$  la cota inferior asociada a esta subregión. Ésta cumple que, para algún  $j$  suficientemente grande,  $L^{(j)} = \tilde{L}$ . Además, por ser una subregión, se cumple que  $\tilde{M}_{k_q} \subseteq M_{k_q}$ , luego  $\tilde{L} \geq L_{k_q}$ . Por tanto, se llega a la cadena de desigualdades

$$L_{k_q} \leq L^{(j)} \leq U^{(j)} \leq U_{k_q}.$$

Finalmente, se tiene que  $\lim_{q \rightarrow \infty} |U_{k_q} - L_{k_q}| = 0$  por ser la operación de acotación consistente por hipótesis, lo cual aplicado a la cadena de desigualdades anteriores y teniendo en cuenta que para un  $q$  suficientemente grande,  $q > j$ , provoca que  $\lim_{q \rightarrow \infty} |U^{(q)} - L^{(q)}| = 0$ , siendo entonces el algoritmo convergente.  $\square$

## Capítulo 2

# Técnicas de reducción de dominio

Uno de los mayores problemas a la hora de resolver un problema de optimización es el tamaño del conjunto factible. Muchas técnicas de resolución para problemas no lineales implican la evaluación de una gran cantidad de puntos, número que aumenta rápidamente a medida que la región factible se hace más grande. Esto provocó que, desde los inicios de esta disciplina, se intentase encontrar maneras de reducir el tamaño de este conjunto, buscando descartar subconjuntos en los cuales no podría encontrarse una solución mejor que la actual.

Una primera aproximación es la que utiliza el esquema de ramificación y acotación explicado en el capítulo anterior. En éste, una subregión se elimina si su cota inferior es superior a la cota superior del problema. Es decir, si se cumple esta condición, se reduce el dominio sobre el que se buscará el óptimo. Otras técnicas se basan en estudiar las relaciones que se dan entre las restricciones y las cotas de las variables, ya que es muy habitual que una restricción provoque que una variable no pueda llegar a su cota máxima o viceversa, pudiendo actualizar esta y reduciendo el dominio.

En este capítulo se explorará una técnica complementaria a las anteriores que permite reducir el dominio de un problema empleando de las restricciones activas de un subproblema concreto del árbol de ramificación. Además, se explicará un algoritmo basado en ramificación y acotación que implementa esta reducción de manera cíclica para aprovecharla lo máximo posible. Este enfoque fue propuesto en [Ryoo and Sahinidis \(1996\)](#) y es utilizado por el optimizador *BARON*, uno de los estándares para la resolución de problemas de optimización no lineal a nivel industrial.

## 2.1. Marco teórico

En este capítulo se trabajará con la siguiente expresión de un problema de optimización, que es equivalente a la Definición 1.3:

$$\begin{aligned} & \text{minimizar } f(\mathbf{x}) \\ & \text{sujeto a } g(\mathbf{x}) \leq 0 \\ & \mathbf{x} \in \mathbf{X} \subset \mathbb{R}^n. \end{aligned} \tag{P}$$

A partir de ahora y mientras no se diga lo contrario, se considerará que el conjunto  $\mathbf{X}$  al cual debe pertenecer el óptimo es *acotado* y que  $f$  es una función *continua*. Esto es necesario para asegurar determinadas propiedades de los métodos que serán propuestos más adelante, pero no supone una mayor limitación ya que en la práctica se suelen utilizar siempre conjuntos y funciones que cumplan estas propiedades.

Además, será de gran utilidad definir el problema  $(\mathbf{R})$ :

$$\begin{aligned} & \text{minimizar } \bar{f}(\mathbf{x}) \\ & \text{sujeto a } \bar{g}(\mathbf{x}) \leq 0 \\ & \mathbf{x} \in \bar{\mathbf{X}}, \end{aligned} \tag{R}$$

donde

- $(\mathbf{R})$  es un problema de optimización convexa.
- $\bar{f} : \bar{\mathbf{X}} \rightarrow \mathbb{R}^n, \bar{g} : \bar{\mathbf{X}} \rightarrow \mathbb{R}^m, \mathbf{X} \subseteq \bar{\mathbf{X}} \subset \mathbb{R}^n, \bar{\mathbf{X}}$  acotado y  $\bar{f}$  continua.
- Para cualquier punto  $\mathbf{x}$  factible para el problema  $(\mathbf{P})$ , se cumple que  $\bar{f}(\mathbf{x}) \leq f(\mathbf{x})$ . De esta manera, el valor de  $\bar{f}(\mathbf{x})$  proporciona una cota inferior de  $f(\mathbf{x})$ .
- $\{\mathbf{x} : g(\mathbf{x}) \leq 0, \mathbf{x} \in \mathbf{X}\} \subseteq \{\mathbf{x} : \bar{g}(\mathbf{x}) \leq 0, \mathbf{x} \in \bar{\mathbf{X}}\}$ . Es decir, el dominio de  $(\mathbf{R})$  es de mayor tamaño que el dominio de  $(\mathbf{P})$ , lo que asegura que cualquier solución de  $(\mathbf{P})$  será también solución de  $(\mathbf{R})$ .

El problema anterior es lo que se denomina una *relajación* de  $(\mathbf{P})$ , en el sentido de que cualquier solución factible de  $(\mathbf{P})$  será una solución factible de  $(\mathbf{R})$  (pero no al revés), y además  $(\mathbf{R})$  es más sencillo de resolver que el problema original.

Si bien este problema relajado debe cumplir todas las propiedades anteriores, las cuales no son triviales, existen diferentes métodos que permiten obtener estas relajaciones en función de como sea el problema de partida  $(\mathbf{P})$ . Pueden ser métodos *ad hoc* para problemas que cumplan una serie de características muy concretas, o métodos más generales como la técnica *RLT* presentada en [Sherali and Tuncbilek \(1992\)](#) para problemas de optimización polinómica genéricos, que será tratada con más de detalle en el Capítulo 3.

Finalmente, se trabajará también con una versión perturbada del problema  $(\mathbf{R})$ , en el cual las restricciones se ven acotadas por un vector  $\mathbf{y}$  en lugar de por el vector nulo  $\mathbf{0}$ . De esta manera, el valor  $y_i$  acota superiormente a la restricción  $i$ -ésima  $g_i(\mathbf{x})$ :

$$\begin{aligned} \phi(\mathbf{y}) = \text{minimizar} \quad & \bar{f}(\mathbf{x}) \\ \text{sujeto a} \quad & \bar{g}(\mathbf{x}) \leq \mathbf{y} \\ & \mathbf{x} \in \bar{\mathbf{X}}. \end{aligned} \tag{\mathbf{R}_y}$$

Este último problema también es convexo, independientemente del vector  $\mathbf{y}$  considerado, por serlo el problema sin perturbar  $(\mathbf{R})$  (sección 5.6 de [Boyd and Vandenberghe \(2011\)](#)). Respecto a la interpretación de este problema, cuando el valor  $y_i$  asociado a la restricción  $\bar{g}_i(\mathbf{x})$  es positivo, se relaja dicha restricción, ampliando el dominio factible del problema. Por el contrario, si es negativo, el dominio se reduce.

Una vez definidos los problemas con los que se trabajará a lo largo del capítulo, se presenta otro concepto que se utilizará más adelante: el de *desigualdad válida*.

**Definición 2.1.** Sea  $U$  una cota superior del problema  $(\mathbf{P})$ . Se dice que una desigualdad es *válida* para  $(\mathbf{P})$  si ésta no excluye ninguna solución de  $(\mathbf{P})$  cuyo valor de la función objetivo sea mejor que  $U$ .

La definición anterior se puede entender como que una desigualdad es válida si cumple que, al añadirse a un problema, permite reducir el tamaño de su conjunto factible sin eliminar soluciones mejores a la actual.

**Ejemplo 2.2.** Para visualizar mejor el concepto de desigualdad válida, considérese el siguiente problema de optimización lineal:

$$\begin{aligned} \text{minimizar} \quad & 6.5x + 7y \\ \text{sujeto a} \quad & 2x + 3y - 600 \leq 0 \\ & x + y - 500 \leq 0 \\ & 2x + y - 400 \leq 0 \\ & x, y \geq 0. \end{aligned} \tag{2.1}$$

Si se representan las restricciones asociadas al problema anterior, se obtiene la región factible mostrada en la Figura 2.1. En ella se puede ver como la restricción  $x + y - 500 \leq 0$  no aporta ninguna información, pues la región factible queda perfectamente delimitada por las otras dos restricciones y las restricciones de rango  $x \geq 0$  y  $y \geq 0$ . Si se calcula el valor de la función objetivo  $f(x, y) = 6.5x + 7y$  en cada punto extremo del poliedro que forma la región factible (los cuales son los únicos candidatos a óptimos en un problema de optimización lineal como se explica en [González Díaz \(2022a\)](#)), se obtiene lo siguiente:

$$f(A) = 1400, \quad f(B) = 1675, \quad f(C) = 1300, \quad f(D) = 0.$$

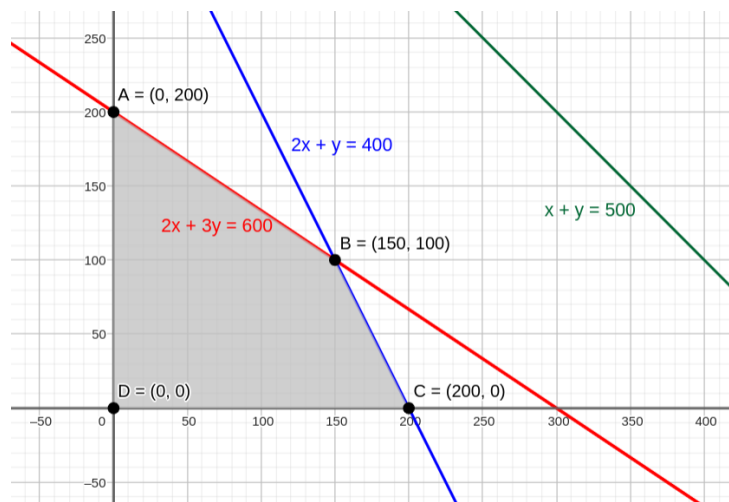


Figura 2.1: Conjunto factible del problema 2.1

En este caso es fácil ver que el óptimo se alcanza en el punto  $D = (0, 0)$ . Sin embargo, supóngase que se tiene un algoritmo un tanto particular que en cada paso analiza un punto siguiendo el orden alfabético de su nombre :  $A - B - C - D$ . El valor del punto  $A$  era 1400, con lo cual al analizar  $B$  no se mejorará la cota superior. Por tanto, si se introduce después de analizar  $A$  una nueva restricción al problema que elimine el punto  $B$  pero no el punto  $C$  (cuyo valor sí mejoraría la cota superior del problema), se podría considerar la desigualdad que define esta restricción como una *desigualdad válida*.

La Figura 2.2 muestra la región factible tras añadir la desigualdad  $2x + 2y \leq 450$ , destacando el área que queda eliminada (en rosa). Esta región es válida y elimina del conjunto factible el punto  $B$ , el cual se sabe que no mejora la cota superior. De esta manera el algoritmo alfabético no comprobaría el valor de  $B$  y saltaría directamente a  $C$ .

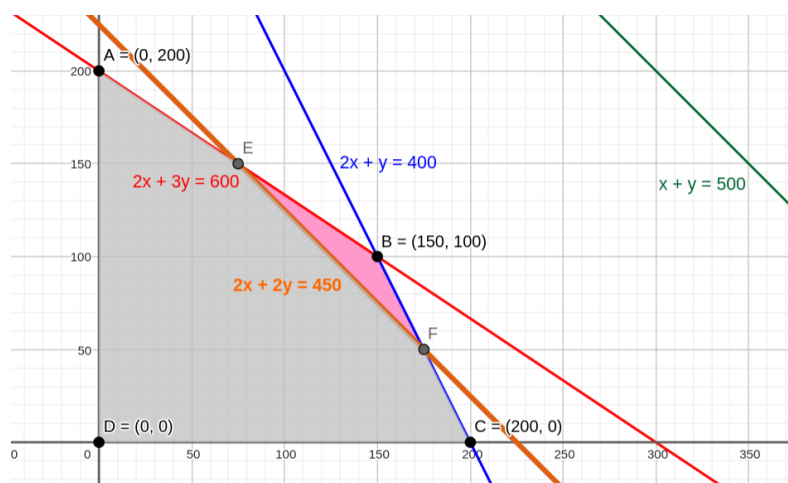


Figura 2.2: Conjunto factible del problema 2.1 con la nueva restricción

Si bien en el ejemplo anterior puede parecer que estas desigualdades no aportan ninguna mejoría, en problemas reales con cientos de variables o desigualdades o en problemas de optimización no lineal, reducir el dominio factible puede suponer la diferencia entre poder resolver un problema en una cantidad de tiempo asumible o no.

En el siguiente apartado se presentarán una serie de desigualdades válidas que permiten reducir el dominio factible, lo cual puede ser aprovechado por diferentes algoritmos de optimización para reducir el coste computacional o espacial de la resolución del problema. Sin embargo, antes de tratar con ellas es necesario presentar una serie de resultados que aportan información adicional sobre cómo se comportan las restricciones en determinados supuestos.

**Lema 2.3.** *Sea una restricción  $\bar{g}_i(\mathbf{x}) \leq 0$  activa en una solución óptima del problema  $(\mathbf{R})$ . Entonces la restricción  $\bar{g}_i(\mathbf{x}) \leq y_i$  también está activa en una solución óptima del problema  $(\mathbf{R}_y)$  para un  $y_i \leq 0$ .*

*Demostración.* Sea  $\bar{\mathbf{x}}$  una solución óptima del problema  $(\mathbf{R})$  en la que la restricción  $\bar{g}_i(\mathbf{x}) \leq 0$  esté activa y supóngase que no hay ningún óptimo del problema perturbado  $(\mathbf{R}_y)$  en el que la restricción  $\bar{g}_i(\mathbf{x}) \leq 0$  esté activa. Sea  $\mathbf{x}^y$  un óptimo del problema perturbado y considérese que la restricción  $\bar{g}_i(\mathbf{x}) \leq y_i$  no se encuentra activa en dicho óptimo. Como el problema perturbado es convexo, se tiene por el Teorema 1.21 que todo óptimo local es un óptimo global. Por tanto, si se elimina el problema perturbado la restricción anterior,  $\mathbf{x}^y$  seguirá siendo un óptimo.

En consecuencia,  $\bar{f}(\mathbf{x}^y) \leq \bar{f}(\bar{\mathbf{x}})$ , y como  $\bar{\mathbf{x}}$  era óptimo de  $(\mathbf{R})$  se tiene también que  $\bar{f}(\bar{\mathbf{x}}) \leq \bar{f}(\mathbf{x}^y)$ , con lo que  $\bar{f}(\bar{\mathbf{x}}) = \bar{f}(\mathbf{x}^y)$ . Por otro lado, por la convexidad de  $(\mathbf{R}_y)$ , cualquier combinación convexa de  $\mathbf{x}^y$  y  $\bar{\mathbf{x}}$  cumple todas las restricciones de  $(\mathbf{R})$  salvo la  $i$ -ésima, que podría o no cumplirla. Además, por convexidad de la función objetivo  $\bar{f}$ , todas las combinaciones convexas tienen también la misma función objetivo.

Por todo lo anterior, como  $g_i(\mathbf{x}^y) < y < 0$  y  $g_i(\bar{\mathbf{x}}) = 0$ , existirá una combinación convexa  $\mathbf{x}^*$  de  $\mathbf{x}^y$  y  $\bar{\mathbf{x}}$  en la que  $g_i(\mathbf{x}^*) = y$ . Luego  $\mathbf{x}^*$  será un óptimo de  $(\mathbf{R}_y)$  en el que la restricción  $i$ -ésima esté activa, contradiciendo el supuesto de partida.  $\square$

Para el siguiente lema es necesario introducir la definición de lo que se conoce como *hiperplano de soporte*.

**Definición 2.4.** Sea  $S$  un conjunto. Se denomina *hiperplano de soporte respecto a  $S$*  a un hiperplano que cumple las siguientes propiedades:

- $S$  está contenido completamente en uno de los subespacios que define el hiperplano.
- El hiperplano contiene, al menos, un punto de la frontera de  $S$ .

**Ejemplo 2.5.** Considérese la región factible del Ejemplo 2.1. En la Figura 2.3 se muestra como la recta  $2x + y - 400 = 0$ , la cual está asociada a una de las desigualdades que limitan la región factible del problema, define un hiperplano de soporte en  $\mathbb{R}^2$  respecto al conjunto factible de dicho ejemplo. Esto es debido a que una parte de la recta está contenida en la frontera del conjunto, y el resto de puntos de éste se encuentran en el mismo subespacio,  $2x + y - 400 < 0$ .

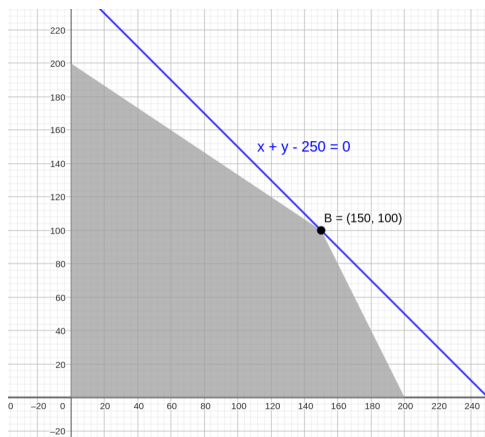


Figura 2.3: Hiperplano de soporte respecto al conjunto factible del problema 2.1

**Lema 2.6** (Teorema 5.4 de [Minoux \(1986\)](#)). Sea  $(\mathbf{R})$  un problema de optimización convexo con un valor óptimo finito. Entonces,  $\mu$  es un multiplicador de punto de silla si y solo si el hiperplano de ecuación  $z = \phi(0) - \mu \cdot \mathbf{y}$  es un hiperplano de soporte en  $\mathbf{y} = \mathbf{0}$  del grafo de la función perturbada  $\phi(\mathbf{y})$ . Es decir,  $\mu$  es un multiplicador de punto de silla si y solo si

$$\forall \mathbf{y} \in \mathbb{R}^n, \phi(\mathbf{y}) \geq \phi(0) - \mu \cdot \mathbf{y}.$$

Se omite la demostración del lema anterior, pues requiere el uso de resultados y conceptos de dualidad en optimización no lineal que van más allá de los objetivos de este trabajo. En este lema se hace empleo de los denominados *multiplicadores de punto de silla*, lo cuales serían el análogo a los multiplicadores de Lagrange en optimización no lineal, si bien en este caso su existencia no está siempre garantizada. Sin embargo, como se explica en [González Díaz \(2022c\)](#), ambos conceptos son equivalentes si se trabaja con problemas de optimización convexa. Como en este trabajo siempre que se necesite trabajar con multiplicadores de punto de silla será en el contexto de optimización convexa, se realizará un abuso de notación y se denominarán directamente *multiplicadores de Lagrange*.

## 2.2. Desigualdades de reducción de dominio

A continuación, se muestran diversos resultados que proporcionan desigualdades que permiten reducir el dominio factible en problemas de optimización matemática. Por esta razón se consideran a estas desigualdades como *técnicas de reducción de dominio*.

**Teorema 2.7** (Teorema 2 de [Ryoo and Sahinidis \(1996\)](#)). *Sea  $(\mathbf{R})$  una relajación convexa de  $(\mathbf{P})$  con un valor óptimo de la función objetivo  $L$ , y considérese una restricción  $\bar{g}_i(\mathbf{x}) \leq 0$  que se encuentra activa en la solución. Sea  $U$  una cota superior conocida del problema  $(\mathbf{P})$ . Si dicha restricción tiene un multiplicador de Lagrange asociado  $\mu_i > 0$ , entonces la siguiente restricción es válida para  $(\mathbf{P})$ :*

$$\bar{g}_i(\mathbf{x}) \geq -\frac{(U - L)}{\mu_i}.$$

*Demostración.* Sea  $(\mathbf{P}_y)$  la siguiente perturbación del problema  $(\mathbf{P})$ :

$$\begin{aligned} \Phi(\mathbf{y}) = \text{mínimo global} \quad & f(\mathbf{x}) \\ \text{sujeto a} \quad & g(\mathbf{x}) \leq \mathbf{y} \\ & \mathbf{x} \in \mathbb{R}^n \end{aligned} \tag{\mathbf{P}_y}$$

Sea  $(\mathbf{R}_y)$  la versión perturbada de  $(\mathbf{R})$ .  $(\mathbf{R}_y)$  es también una relajación convexa de  $(\mathbf{P}_y)$ , con lo cual el valor objetivo del problema  $(\mathbf{R}_y)$  será siempre menor o igual que el del problema  $(\mathbf{P}_y)$ . Por tanto,  $\phi(\mathbf{y}) \leq \Phi(\mathbf{y})$  y cualquier cota inferior válida para  $\phi(\mathbf{y})$  será también una cota inferior válida para  $\Phi(\mathbf{y})$ .

Ahora considérese una perturbación  $(\mathbf{R}_{y_i})$  donde solo está perturbado el lado derecho de la restricción  $\bar{g}_i(\mathbf{x}) \leq y_i$ . Por el Lema 2.6, se tiene que una cota inferior para  $\phi(y_i)$  viene dada por su hiperplano de soporte  $L - \mu_i y_i$ . Así, para cada  $y_i$ , se cumple que  $L - \mu_i y_i \leq \phi(y_i) \leq \Phi(y_i)$ . Ahora bien, el valor de  $\Phi(y_i)$  no debe ser mayor que la cota superior conocida  $U$ , luego se tiene también que  $L - \mu_i y_i \leq U$ . Por hipótesis,  $\bar{g}_i(\mathbf{x}) \leq y_i$  está activa para  $y_i = 0$ .

Para asegurar que la desigualdad encontrada es una desigualdad válida es necesario demostrar también que, para cualquier  $\mathbf{x}$  factible, se cumple que  $L - \mu_i \bar{g}_i(\mathbf{x}) \leq U$ . Sea un  $\mathbf{x}$  factible cualquiera. En particular, en él se cumplirá que  $\bar{g}_i(\mathbf{x}) \leq 0$ . Tomando ahora  $y_i = \bar{g}_i(\mathbf{x})$ , se tiene, por el Lema 2.3, que para el problema  $(\mathbf{R}_{y_i})$  la restricción  $\bar{g}_i(\mathbf{x})$  estará saturada en el óptimo. Por tanto, por el razonamiento anterior, se cumplirá que  $L - \mu_i y_i \leq U$ , y como  $y_i = \bar{g}_i(\mathbf{x})$ , entonces  $L - \mu_i \bar{g}_i(\mathbf{x}) = L - \mu_i y_i \leq U$ .  $\square$

*Observación 2.8.* En el teorema anterior, la razón de denotar al valor de la función objetivo del problema  $(\mathbf{R})$  como  $L$ , mismo símbolo que se utiliza para denotar a una cota inferior, es que este

valor es, en efecto, una cota inferior del problema  $(\mathbf{P})$ . Basta recordar que  $L = \bar{f}(\mathbf{x}) \leq f(\mathbf{x})$  para cualquier  $\mathbf{x}$  factible para  $(\mathbf{P})$ .

El teorema anterior está formulado para una restricción genérica  $\bar{g}_i(\mathbf{x})$ . Los siguientes corolarios muestran su particularización para restricciones definidas por funciones lineales y para restricciones de rango, las cuales son dos de los tipos más comunes de restricciones en la resolución de problemas reales.

**Corolario 2.9.** *Sea  $(\mathbf{R})$  una relajación convexa de  $(\mathbf{P})$  con un valor óptimo de la función objetivo dado por  $L$  y considérese una restricción lineal, la  $i$ -ésima del problema,  $\mathbf{a}_i^t \mathbf{x} - b_i \leq 0$  que está activa en la solución del problema  $(\mathbf{R})$ . Sea  $U$  una cota superior conocida del problema  $(\mathbf{P})$ . Si dicha restricción tiene un multiplicador de Lagrange asociado  $\mu_i > 0$ , entonces la siguiente restricción es válida para  $(\mathbf{P})$ :*

$$\mathbf{a}_i^t \mathbf{x} \geq b_i - \frac{(U - L)}{\mu_i}.$$

*Demostración.* Sea  $\bar{g}_i(\mathbf{x}) := \mathbf{a}_i^t \mathbf{x} - b_i$ . Por el Teorema 2.7,

$$\mathbf{a}_i^t \mathbf{x} - b_i = \bar{g}_i(\mathbf{x}) \geq -\frac{(U - L)}{\mu_i} \iff \mathbf{a}_i^t \mathbf{x} \geq b_i - \frac{(U - L)}{\mu_i}$$

□

Los dos siguientes corolarios, válidos para restricciones de rango, se pueden demostrar de manera análoga por lo que se omite su demostración.

**Corolario 2.10.** *Sea  $(\mathbf{R})$  una relajación convexa de  $(\mathbf{P})$  cumpliendo las condiciones del Problema  $\mathbf{R}$ , con un valor óptimo de la función objetivo dado por  $L$  y considérese una restricción de rango  $\mathbf{x}_j - x_j^U \leq 0$ , la  $j$ -ésima del problema, que está activa en la solución del problema  $(\mathbf{R})$ . Sea  $U$  una cota superior conocida del problema  $(\mathbf{P})$ . Si dicha restricción tiene un multiplicador de Lagrange asociado  $\lambda_j > 0$ , entonces la siguiente restricción es válida para  $(\mathbf{P})$ :*

$$\mathbf{x}_j \geq x_j^U - \frac{(U - L)}{\lambda_j}.$$

**Corolario 2.11.** *Sea  $(\mathbf{R})$  una relajación convexa de  $(\mathbf{P})$  con un valor óptimo de la función objetivo dado por  $L$  y considérese una restricción de rango  $x_j^L - \mathbf{x}_j \leq 0$  que está activa en la solución del problema  $(\mathbf{R})$ . Sea  $U$  una cota superior conocida del problema  $(\mathbf{P})$ . Si dicha restricción tiene un multiplicador de Lagrange asociado  $\lambda_j > 0$ , entonces la siguiente restricción es válida para  $(\mathbf{P})$ :*

$$\mathbf{x}_j \leq x_j^L + \frac{(U - L)}{\lambda_j}.$$

En los resultados anteriores se pide que la restricción sobre la que se quiera aplicar la reducción de cota se encuentre activa. Sin embargo, es posible formular una versión alternativa para una

restricción no activa. Ahora bien, como se verá a continuación, no pedir esta condición implica añadir una nueva desigualdad al problema, a fin de saturar la restricción que se quiere reducir de manera artificial, y resolver el nuevo problema que se forma. Por tanto, se debe valorar seriamente cuánto compensa la posible reducción del dominio, ya que puede ser insuficiente para asumir el nuevo coste que supone esta resolución adicional.

Mencionar que este tipo de técnicas, en las cuales se fija de manera artificial el valor de una restricción o una variable para estudiar como afecta al resto del problema, se denominan *técnicas de probing*.

**Teorema 2.12.** *Sea  $(\mathbf{R})$  una relajación convexa de  $(\mathbf{P})$  y considérese una restricción lineal  $\mathbf{a}_i^t \mathbf{x} - b_i \leq 0$  que no esté activa en una solución de  $(\mathbf{R})$ . Sea  $U$  una cota superior conocida del problema  $(\mathbf{P})$ . Resuélvase  $(\mathbf{R})$  tras fijar  $\mathbf{a}_i^t$  en el valor  $b_i$ , es decir, tras añadir la restricción  $b_i \leq \mathbf{a}_i^t \mathbf{x}$  al problema. Sea  $Z$  un valor óptimo de la función objetivo para este problema relajado. Si la nueva restricción  $b_i \leq \mathbf{a}_i^t \mathbf{x}$  tiene un multiplicador de Lagrange asociado  $\mu_i > 0$ , entonces la siguiente restricción es válida para  $(\mathbf{P})$ :*

$$\mathbf{a}_i^t \mathbf{x} \leq b_i + \frac{(U - Z)}{\mu_i}.$$

*Demostración.* Fijando  $\mathbf{a}_i^t \mathbf{x}$  en  $b_i$ , la restricción  $b_i \leq \mathbf{a}_i^t \mathbf{x}$  pasa a estar activa. La nueva relajación que se forma al añadir esta restricción al problema  $(\mathbf{R})$  sigue siendo convexa, ya que solamente se está añadiendo otra restricción convexa, lo que equivale a intersectar un subconjunto convexo con el dominio factible (que es convexo), siendo la intersección de convexos convexa, lo cual se puede encontrar en [Boyd and Vandenberghe \(2011\)](#). Sea  $g_i(\mathbf{x}) := b_i - \mathbf{a}_i^t \mathbf{x} < 0$ . Como se conoce el óptimo  $Z$  de este nuevo problema, y la restricción tiene un multiplicador de Lagrange  $\mu_i > 0$ , es posible aplicar el teorema 2.7:

$$b_i - \mathbf{a}_i^t \mathbf{x} \geq -\frac{(U - Z)}{\mu_i} \iff -\mathbf{a}_i^t \mathbf{x} \geq -b_i - \frac{(U - Z)}{\mu_i} \iff \mathbf{a}_i^t \mathbf{x} \leq b_i + \frac{(U - Z)}{\mu_i}.$$

□

*Observación 2.13.* El teorema anterior solo proporciona una restricción útil cuando  $Z > U$ :

- Si  $Z = U$ , se añadiría la restricción  $\mathbf{a}_i^t \mathbf{x} \leq b_i$ , equivalente a la inicial  $\mathbf{a}_i^t \mathbf{x} - b_i \leq 0$ .
- Si  $Z < U$ ,  $\mathbf{a}_i^t \mathbf{x} \leq b_i \leq b_i + \frac{(U - Z)}{\mu_i}$ , con lo cual no se mejora la cota dada por la restricción inicial  $\mathbf{a}_i^t \mathbf{x} - b_i \leq 0$ .

De manera análoga al teorema 2.7, se pueden obtener nuevas desigualdades válidas asociadas a restricciones de rango:

**Corolario 2.14.** *Sea  $(\mathbf{R})$  una relajación convexa de  $(\mathbf{P})$  y considérese una restricción de rango  $\mathbf{x}_j - x_j^U \leq 0$  que no esté activa en una solución de  $(\mathbf{R})$ . Sea  $U$  una cota superior conocida del*

problema  $(\mathbf{P})$ . Resuélvase  $(\mathbf{R})$  tras fijar  $\mathbf{x}_j$  en el valor  $x_j^U$ , es decir, tras añadir la restricción  $x_j^U \leq \mathbf{x}_j$  al problema. Sea  $Z$  un valor óptimo de la función objetivo para este problema relajado. Si la nueva restricción  $x_j^U \leq \mathbf{x}_j$  tiene un multiplicador de Lagrange asociado  $\lambda_j > 0$ , entonces la siguiente restricción es válida para  $(\mathbf{P})$ :

$$\mathbf{x}_j \leq x_j^U + \frac{(U - Z)}{\lambda_j}.$$

**Corolario 2.15.** Sea  $(\mathbf{R})$  una relajación convexa de  $(\mathbf{P})$  y considérese una restricción de rango  $\mathbf{x}_j - x_j^L \geq 0$  que no esté activa en una solución de  $(\mathbf{R})$ . Sea  $U$  una cota superior conocida del problema  $(\mathbf{P})$ . Resuélvase  $(\mathbf{R})$  tras fijar  $\mathbf{x}_j$  al valor  $x_j^L$ , es decir, tras añadir la restricción  $\mathbf{x}_j \leq x_j^L$  al problema. Sea  $Z$  un valor óptimo de la función objetivo para este problema relajado. Si la nueva restricción  $\mathbf{x}_j \leq x_j^L$  tiene un multiplicador de Lagrange asociado  $\lambda_j > 0$ , entonces la siguiente restricción es válida para  $(\mathbf{P})$ :

$$\mathbf{x}_j \geq x_j^L - \frac{(U - Z)}{\lambda_j}.$$

### 2.2.1. Otras técnicas de reducción de dominio

Además de las desigualdades que se acaban de presentar, existen técnicas que utilizan otra información disponible en el problema o que es posible obtener a partir de él. Algunas de ellas buscan ajustar las cotas de una variable del problema basándose en las cotas del resto de variables y en las propias restricciones del problema. Este tipo de técnicas se conocen como técnicas de *bound tightening*, pudiendo destacar las siguientes:

- *Feasibility-Based Bound Tightening (FBBT)*: Busca ajustar las cotas de las variables basándose en la factibilidad de estas, es decir, en los valores que pueden tomar cumpliendo las restricciones del problema.
- *Optimality-Based Bound Tightening (OBBT)*: Busca ajustar las cotas de las variables basándose en la optimalidad, resolviendo para cada variable un problema de minimización y uno de maximización a fin de encontrar su valor mínimo y máximo verificando las restricciones del problema.

Si bien ambas técnicas son muy utilizadas en la práctica, el FBBT es el más común, pues requiere un gasto computacional mucho menor al no tener que resolver problemas adicionales como necesita el OBBT, los cuales pueden ser de una complejidad similar al problema original a resolver. Esto provoca que el FBBT sea una técnica muy utilizada como un paso adicional dentro de muchos algoritmos de resolución de problemas de optimización, incluidos algunos que serán tratados en este trabajo. Si bien no se va a entrar en más detalles teóricos o técnicos (para más información consultar [Belotti et al. \(2012\)](#)), se incluye a continuación un ejemplo sencillo que ilustra cómo funciona esta técnica.

**Ejemplo 2.16.** Considérese el siguiente problema de optimización, siendo  $f$  un función continua genérica:

$$\begin{aligned} \text{minimizar} \quad & f(x_1, x_2) \\ \text{sujeto a} \quad & x_1 + x_2 \leq 2 \\ & 1 \leq x_1 \leq 5 \\ & -2 \leq x_2 \leq 2. \end{aligned} \tag{2.2}$$

Como  $x_1 \geq 1$ ,  $x_2$  podrá valer como mucho 1 y nunca podrá alcanzar su cota superior. Además, como la cota inferior de  $x_2$  es  $-2$ , para que se verifique esta restricción el valor de  $x_1$  nunca podrá ser superior a 4. Por tanto, es posible modificar las cotas superiores de cada variable utilizando la restricción y las cotas de la otra variable, llegando a que existe un problema con un dominio menor cuya solución óptima es la misma que la del problema de partida:

$$\begin{aligned} \text{minimizar} \quad & f(x_1, x_2) \\ \text{sujeto a} \quad & x_1 + x_2 \leq 2 \\ & 1 \leq x_1 \leq 4 \\ & -2 \leq x_2 \leq 1. \end{aligned} \tag{2.3}$$

### 2.3. El algoritmo de ramificación y reducción

Una manera de aprovechar las desigualdades definidas en el apartado anterior viene presentada en [Ryoo and Sahinidis \(1996\)](#). En dicho artículo se propone el *algoritmo de Ramificación y Reducción* (RyR), el cual es una modificación del esquema de ramificación y acotación explicado en el capítulo anterior. En este algoritmo se introduce un nuevo paso en el cual se calculan de las desigualdades de reducción de dominio demostradas antes, a fin de reducir el dominio factible en cada iteración y acelerar la búsqueda del óptimo global. Además, se plantea realizar esta reducción de manera recurrente en una misma iteración hasta que no sea posible ajustar más el dominio, a fin de aprovechar al máximo las posibles mejoras.

La idea general del algoritmo se muestra en la Figura 2.4. Se pueden ver las etapas esenciales de RyA (selección de subproblema, acotación y ramificación), junto con la nueva etapa de reducción de dominio. Si este se ve reducido bajo ciertas condiciones que se indicarán más adelante, se vuelve a la etapa de acotación para intentar aprovechar los nuevos límites del problema. Junto con esa etapa, se añaden dos pasos complementarios que buscan reducir aún más el dominio del problema (aplicando FBBT) y reducir más rápido la cota superior global.

Cabe mencionar que una de las particularidades de este algoritmo es que trabaja desde el principio con una relajación convexa (**R**) del problema (**P**) a resolver. De esta manera, cuando se ramifica, se selecciona un nuevo subproblema o se reduce su dominio, se está haciendo siempre sobre una relajación o sobre un subproblema derivado de ella, no directamente sobre el problema

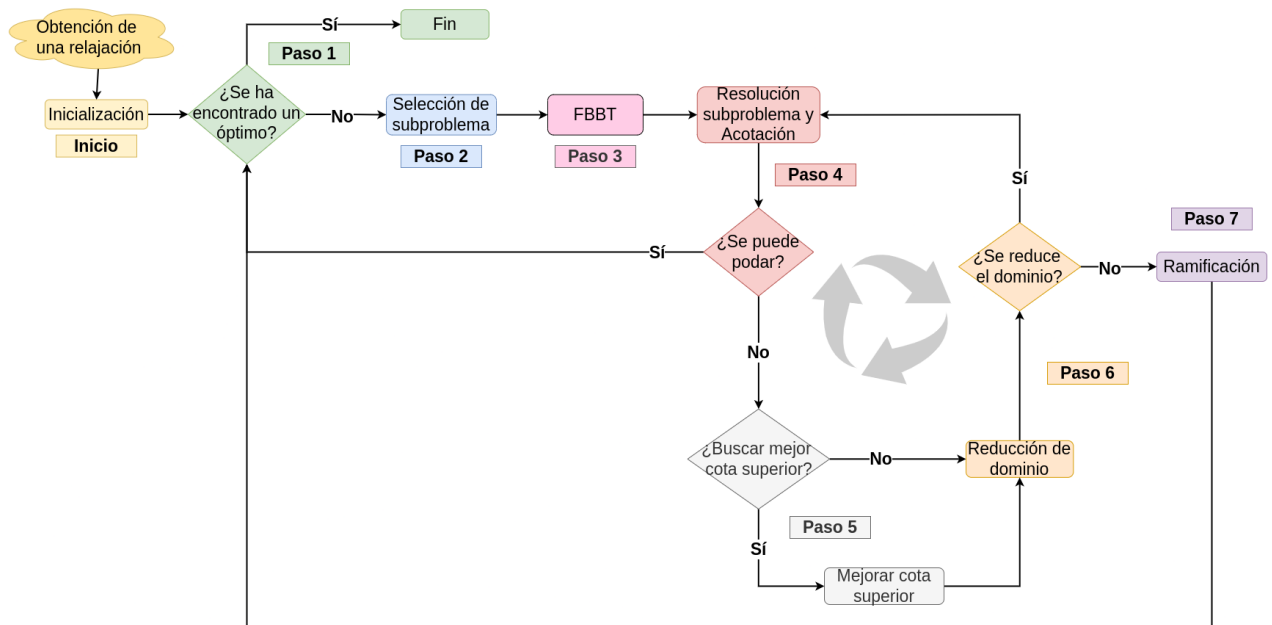


Figura 2.4: Diagrama de flujo del Algoritmo RyR, donde cada color se corresponde con un paso.

que se quiere resolver. Los únicos momentos en los que se utiliza el problema original son para comprobar si se mejora la cota actual (ya que se comprueba si la cota de la relajación actual mejora a la cota del problema original) y para calcular las desigualdades de reducción de dominio. La gran ventaja de trabajar con relajaciones convexas es que los subproblemas que se generan al ramificar también son convexas, con lo cual su resolución (la cual es necesaria para realizar la acotación) se puede hacer de manera sencilla.

En el Algoritmo 2 se encuentra descrito cada paso en detalle. A lo largo de este apartado se discutirán sus características y propiedades más importantes, a fin de poder probar que este algoritmo converge a un óptimo del problema ( $\mathbf{P}$ ).

---

### Algoritmo 2 Ramificación y Reducción

---

#### Iniciación

Sea  $R$  una relajación del problema ( $P$ ) a resolver.

Se fija la iteración inicial  $k = 0$  y la cota superior del problema  $U^{(k)} = +\infty$ .

Se incluye  $R_1 = R$  en la lista *ACTIVOS* de subproblemas activos, fijando su cota inferior  $L_1 = -\infty$

**Bucle principal** (iteración  $k$ -ésima)

#### Paso 1: Finalización o cambio de iteración

Se fija la cota inferior global  $L^{(k)} = \min_{i: R_i \in \text{ACTIVOS}} \{L_i\}$

Se eliminan todos los subproblemas que no podrían mejorar la cota superior global:

$\text{ACTIVOS} \leftarrow \text{ACTIVOS} \setminus \{R_j\}$  para todos los  $R_j$  tales que  $L_j \geq U^{(k)}$

**if**  $\text{ACTIVOS} = \emptyset$  **then**

Se ha encontrado una solución óptima. **FIN**

**else**

Cambio de iteración  $k \leftarrow k + 1$ ,  $U^{(k)} \leftarrow U^{(k-1)}$  y  $L^{(k)} \leftarrow L^{(k-1)}$

Se avanza al Paso 2

**end if**

**Paso 2: Selección de subproblema**

Seleccionar  $R_i$  de *ACTIVOS* utilizando una regla de selección de nodos y actualizar la lista:  
 $ACTIVOS \leftarrow ACTIVOS \setminus \{R_i\}$

Ir al Paso 3

**Paso 3: FBBT**

Reducir el rango de las variables asociadas al subproblema  $R_i$  empleando FBBT.

Ir al Paso 4

**Paso 4: Resolución de subproblema y Acotación**

Resolver o acotar inferiormente el problema  $R_i$ . Sea  $L_i$  esta cota inferior (que será  $\bar{f}(x^i)$  si pudo resolverse). Si el problema no es factible, fijar  $L_i = +\infty$ .

**if**  $x_i$  es factible para el problema ( $P$ ) y  $f(x^i) < U^{(k)}$  **then**

$U^{(k)} \leftarrow f(x^i)$

Tomar  $x^i$  como nueva mejor solución:  $x^* \leftarrow x^i$

**end if**

**if**  $L_i \geq U^{(k)}$  **then**

Ir al Paso 1, ya que no se mejora la cota global actual

**else**

Ir al Paso 5.

**end if**

**Paso 5: Acotación superior (opcional)**

Aplicar una heurística para encontrar una mejor solución factible  $x^h$  para ( $P$ ).

**if**  $f(x^h) < f(x^*)$  **then**

$U^{(k)} \leftarrow f(x^h)$

Tomar  $x^h$  como nueva mejor solución:  $x^* \leftarrow x^h$

**end if**

Ir al Paso 6

**Paso 6: Reducción de dominio**

Calcular las desigualdades de reducción de dominio

**if** Se reduce el dominio de al menos una variable de  $R_i$  en al menos una cantidad prefijada  $\delta > 0$  **then**

Reconstruir el problema  $R_i$  con las nuevas cotas

Ir al Paso 4

**else**

Ir al Paso 7

**end if**

**Paso 7: Ramificación**

Aplicar una regla de ramificación a  $R_i$  para obtener un nuevo conjunto de subproblemas  $R_{i_1}, R_{i_2}, \dots, R_{i_q}$ , y añadirlos todos a *ACTIVOS*

Ir al Paso 1

---

Una de las etapas más complejas en el algoritmo de RyR es el Paso 4, pues implica la resolución del subproblema y su empleo para realizar la acotación, a fin de reducir la cota superior global y, si es posible, dejar de ramificar el subproblema. Para entender correctamente como se realiza este proceso, es necesario realizar algunas aclaraciones respecto a los elementos que aparecen en dicho paso en el Algoritmo 2:

- La razón de emplear como elemento principal  $L_i$  es que, por como se definió la función  $\bar{f}$ , se cumple que en cualquier punto  $x$  factible de  $(\mathbf{P})$ ,  $L_i \leq \bar{f}(x) \leq f(x)$ .
- En caso de que el problema no sea factible, fijar su cota inferior  $L_i$  a  $+\infty$  provoca que se vuelva al Paso 1 y no se llegue a la ramificación (Paso 7), descartando todos los subproblemas que podrían descender de él (podando esa parte del árbol de ramificación), ya que ninguno contiene soluciones factibles de  $R_i$  (y por tanto, ninguna tampoco de  $(\mathbf{P})$ ).
- Respecto a la actualización de la cota superior global, cuando se mira si  $f(x^i)$  es menor que  $U^{(k)}$ , lo que se está haciendo es comprobar si la solución encontrada para el subproblema  $R_i$  mejora a la mejor solución encontrada hasta el momento para el problema  $(\mathbf{P})$ .

Aunque el Paso 4 es el más complejo, también es bueno realizar alguna consideración adicional sobre la aplicación de FBBT (Paso 3) y la acotación superior opcional (Paso 5):

- A pesar de que el FBBT también se podría considerar parte de la reducción del dominio, el algoritmo lo considera un paso aparte pues no lo incluye dentro de los pasos de ciclado (Pasos 4 a 6).
- Para la acotación superior opcional pueden diseñarse heurísticas específicas para un tipo concreto de problemas, en los que se sepa que si sucede una condición, entonces la cota superior debe cumplir cierta propiedad. Otra opción es realizar llamadas a optimizadores no lineales para resolver una versión relajada del problema actual, aunque esto tiene un coste de computación bastante elevado y por eso no debería realizarse en todas las iteraciones (de ahí que este paso sea opcional).

A continuación, se explicarán las propiedades que deben cumplir las operaciones de selección, ramificación y acotación para que el algoritmo de RyR se adapte al esquema de RyA correctamente y se obtenga un algoritmo que converja a una solución del problema  $(\mathbf{P})$ , y no solamente a una solución de la relajación  $(\mathbf{R})$ .

### Operación de selección

En el esquema de RyA, sobre la operación de selección se imponía la condición de que debía *ajustar cotas*, lo cual significa que siempre que se seleccione una nueva subregión, esta debe contener al menos un elemento cuyo valor sea la cota inferior actual del problema.

En el contexto de RyR, una regla de selección que cumple esta condición sería escoger como siguiente subproblema  $R_i$  aquel contenido en la lista *ACTIVOS* que tenga una menor cota inferior  $L_i$ , con  $i \in \operatorname{argmin}_{\{j: R_j \in \text{ACTIVOS}\}} \{L_j\}$ . La condición se cumple debido a que en el Paso 1 de RyR se fija la cota inferior global de la iteración  $k$ -ésima como  $L^{(k)} = \min_{\{j: R_j \in \text{ACTIVOS}\}} \{L_j\}$ . Luego, por definición, esta regla de selección ajusta cotas.

### Operación de ramificación

Si bien en el esquema genérico de RyA no se imponían restricciones sobre la operación de ramificación, en este caso sí es necesario imponer una condición sobre ella:

**Condición 2.17.** *La operación de ramificación debe cumplir que el conjunto de óptimos posibles asociado a un subproblema esté contenido en el conjunto de óptimos posible asociado a su problema padre.*

Expresando lo anterior en los términos de RyR, si  $\bar{X}_i$  es el conjunto de óptimos posibles de un subproblema  $R_i$ , debe cumplirse que  $\bar{X}_{i_q} \subset \bar{X}_i$  para todo  $q$ . De esta manera, lo que se asegura es que al ramificar se está generando una sucesión decreciente de conjuntos, reduciendo siempre el espacio de búsqueda del óptimo.

### Operación de acotación

Finalmente, la operación sobre la que es necesario imponer más condiciones es la de acotación. Esto es debido a que es donde se incorporan al esquema de RyA las técnicas de reducción de dominio explicadas anteriormente, provocando que tenga una mayor complejidad que en el esquema general.

**Condición 2.18.** *Sea  $G_i = \{\mathbf{x} \in \mathbb{R}^n : \bar{g}_l(\mathbf{x}) \leq 0, l = 1, \dots, m\}$  el conjunto de puntos de  $\mathbb{R}^n$  que verifica las restricciones asociadas a un problema  $R_i$ . Para cualquier otro subproblema  $R_{i_q}$  que derive de este se debe cumplir que  $G_{i_q} \subseteq G_i$ .*

Esta primera condición sobre la acotación complementa a la condición 2.17 que se impuso sobre la ramificación. Su propósito es asegurar que el conjunto de puntos que verifiquen las restricciones de un subproblema sea más pequeño que el del problema del que derive.

**Condición 2.19.** *Para un subproblema  $R_i$  se debe cumplir que  $L_i = U_i$  si todos los términos no convexos del problema original ( $P$ ) cumplen que  $x_j^U = x_j^L$ .*

Esta condición dice que cuando en un subproblema todas las variables involucradas en términos no convexos están fijadas (es decir, que sus cotas inferiores y superiores coinciden), entonces

el óptimo de la relajación con esas cotas coincide con el óptimo del problema lineal sujeto a esas mismas cotas. Esto es así porque con dichas cotas no habría ningún término no convexo presente, luego sería como si la relajación y el problema original fueran el mismo (con esas cotas fijadas).

Utilizando las condiciones que se acaban de presentar, es posible demostrar que las subregiones factibles asociadas a un subproblema son un refinamiento de la asociada a su problema padre y que además, su cota inferior es siempre superior. De esta manera se asegura que una subregión nunca va a proporcionar un óptimo peor que el obtenido en su región padre, en caso de tenerlo.

**Proposición 2.20.** *Sean  $M_{i_q} = G_{i_q} \cap \bar{X}_{i_q}$  y  $M_i = G_i \cap \bar{X}_i$ . Entonces,  $M_{i_q} \subset M_i$  y además  $L_{i_q} \geq L_i$  para todo  $q$ .*

*Demostración.* Por la condición 2.17,  $G_{i_q} \subset G_i$ , y por la condición 2.19,  $\bar{X}_{i_q} \subset \bar{X}_i$ . Entonces,  $M_{i_q} \subset M_i$ , y en consecuencia, como  $L_j$  es el mínimo de una región  $M_j$ ,  $L_{i_q} \geq L_i$  para cualquier  $q$ .  $\square$

### 2.3.1. Convergencia de ramificación y reducción

Una vez establecidas todas las propiedades que deben cumplir las operaciones del algoritmo y el problema al que se le quiera aplicar, parece lógico demostrar que este converge a una solución de (P).

Por un lado, es necesario que el esquema de RyA que subyace en RyR sea convergente. Para ello se necesita que la operación de selección ajuste cotas, lo cual se vio antes, y que la operación de acotación sea consistente. Logrado esto, por el Teorema 1.26 se obtiene la convergencia. Se omite la demostración de la consistencia de la acotación, pudiendo encontrarse en [Ryoo and Sahinidis \(1996\)](#).

**Proposición 2.21.** *La operación de acotación en el algoritmo de RyR es consistente.*

Además de la convergencia, es necesario probar que el algoritmo carece de ciclos, es decir, que no existe una secuencia de pasos en la cual puede quedarse indefinidamente, evitando la convergencia del mismo. En concreto, podría existir uno durante el proceso de acotación y reducción de dominio, ya que tras reducir el dominio en el Paso 6 se vuelve al Paso 4, dando la sensación de que podría quedar indefinidamente ahí. Por suerte, el siguiente resultado muestra que no es posible esta situación.

**Proposición 2.22.** *El algoritmo RyR no cicla entre los Pasos 4 y 6.*

*Demostración.* Por la condición incluida en el Paso 6, solo se repetirá la secuencia de pasos 4-6 si se reduce el rango de alguna variable de  $R_i$  al menos en una cantidad prefijada  $\delta > 0$ . Como la región factible es acotada, estará contenida en un hipercubo suficientemente grande, con lo cual cada variable tendrá un cierto rango de variación finito. Entonces, fijado un valor  $\delta > 0$ , el número de veces que el rango de una variable se podrá reducir en una cantidad delta es también finito. Por tanto, el algoritmo no puede ciclar indefinidamente entre los Pasos 4 y 6 porque siempre que de 6 vuelve a 4 es porque ha reducido el rango de al menos una variable en al menos una cantidad  $\delta$ .  $\square$

**Teorema 2.23.** *El algoritmo de RyR converge a una solución del problema (P).*

*Demostración.* Esta prueba es consecuencia de diversos hechos desarrollados a lo largo de este capítulo:

- Las técnicas de reducción de dominio aplicadas en el Paso 6 de RyR provienen de añadir desigualdades válidas al problema. Luego, aunque se eliminen soluciones factibles de (P), ninguna será mejor que la solución actual.
- El algoritmo carece de ciclos.
- El algoritmo de RyA subyacente es convergente por el Teorema 1.26, ya que la operación de acotación es consistente por la proposición 2.21, y la de selección ajusta cotas.

$\square$



## Capítulo 3

# Optimización polinómica

Dentro del campo de la optimización no lineal, existen un tipo de problemas de especial interés desde el punto de vista industrial. Estos son los *problemas de optimización polinómicos*, en los cuales la función objetivo y restricciones vienen dadas por polinomios. Su importancia radica en que, debido a la versatilidad de los polinomios y a la existencia de una gran cantidad de técnicas para convertir funciones en polinomios (piénsese en las técnicas de interpolación polinómica o en las aproximaciones de Taylor, muy usadas debido a su sencillez), es posible modelizar muchos procesos y situaciones con este tipo de problemas, en ámbitos tan diferentes como la economía, el diseño de plantas químicas, estudio de flujos en redes o el diseño de microchips como se muestra en [Shen et al. \(2020\)](#). Debido a ello, en los últimos años ha crecido el interés en desarrollar métodos que permitan resolverlos de manera más rápida y eficiente, los cuales pueden ser específicos para este tipo de problemas o ser adaptaciones de otros métodos existentes más generales.

En este capítulo se presentarán ejemplos de las dos vertientes. Por un lado, se mostrará cómo es posible adaptar el algoritmo de ramificación y reducción para unos tipos particulares de problemas polinómicos muy comunes en procesos industriales, lo cual permite mejorar su rendimiento. Por otro, se presentará un nuevo algoritmo para la resolución de problemas polinómicos, el cual sigue también esquema de RyA. Además, permite incluir en alguno de sus pasos las desigualdades de reducción de dominio explicadas en el Capítulo 2, mostrando que es posible aprovechar esta técnica más allá del algoritmo de RyR.

Antes de comenzar, es conveniente definir qué se entenderá a partir de ahora como polinomio y como problema de optimización polinómico.

**Definición 3.1.** Dado un vector  $\mathbf{x} \in \mathbb{R}^n$ , se denomina *monomio* a cualquier expresión de la forma  $m(\mathbf{x}) = a \prod_{i=1}^n x_i^{\beta_i}$ , donde  $a \in \mathbb{R}$  y  $\beta_i \in \mathbb{N} \cup \{0\}$ ,  $i = 1, \dots, n$ . Para un monomio  $m(\mathbf{x})$  se define su *grado* como  $\delta = \sum_{i=1}^n \beta_i$ .

**Definición 3.2.** Llamamos *polinomio* a la suma de un número finito de monomios  $m_1(\mathbf{x}), \dots, m_l(\mathbf{x})$

$$p(\mathbf{x}) = \sum_{i=0}^l m_i(\mathbf{x}).$$

El *grado de un polinomio* se corresponde con el mayor de los grados de los monomios que lo conforman. Además, se dirá que una función  $f$  es una *función polinómica* si su expresión viene dada por un polinomio.

Utilizando la definición anterior y siguiendo la notación presentada en la Definición 1.3, se puede introducir lo que se conoce por problema polinómico.

**Definición 3.3.** Un *problema de optimización matemática polinómico* es un problema de optimización matemática en el que las funciones  $c$ ,  $g_i$  y  $h_j$  son polinómicas. Es decir, tenemos un problema donde la función objetivo y todas sus restricciones vienen definidas por polinomios.

**Ejemplo 3.4.** Los problemas definidos en los Ejemplos 1.12 y 1.19 son problemas de optimización polinómica.

### 3.1. Ramificación y reducción en optimización polinómica

El algoritmo de ramificación y reducción (RyR) presentado en el capítulo anterior permitía resolver un problema de minimización genérico cuya función objetivo fuese continua y su solución se buscara en un subconjunto de  $\mathbb{R}^n$  acotado. Como un polinomio es una función continua, en particular puede utilizarse para resolver problemas de optimización polinómica.

En este apartado se presentarán dos tipos de problemas de optimización polinómica concretos para los cuales es posible generar una relajación convexa particular aprovechando las particularidades de los mismos, tal y como se explica en [Ryoo and Sahinidis \(1996\)](#). De esta manera, modificando ligeramente el esquema presentado en el capítulo anterior, se puede llegar a versiones de RyR específicamente pensadas para resolver estos problemas. Se supondrá en ambos casos que se busca una solución en un conjunto  $\mathbf{X}$  acotado.

#### 3.1.1. Polinomios univariados

En primer lugar, se considerará un tipo de problemas cuya función objetivo viene definida por un *polinomio univariable*, que son aquellos formados por la suma de monomios en una única variable; y que las únicas restricciones existentes son de rango sobre dicha variable. Si bien este tipo de problemas pueden resolverse con métodos específicos para optimización sin restricciones, como los estudiados en la asignatura del grado *Métodos Numéricos en Optimización y Ecuaciones*

*Diferenciales*, se pueden plantear también como un problema de optimización como los tratados a lo largo de este trabajo:

$$\begin{aligned} \text{minimizar}_{x \in X \subseteq \mathbb{R}} \quad & f(\mathbf{x}) = \sum_{i=0}^p a_i x^i \\ \text{sujeto a} \quad & x^L \leq x \leq x^U \end{aligned} \quad (3.1)$$

donde  $a_i \in \mathbb{R}$ , para  $i = 1, \dots, p$ .

Para la construcción de una relajación convexa que se ajuste a la presentada en **(R)** y permita aplicar RyR, el primer paso es agrupar los monomios convexos y los cóncavos por separado. Para ello se puede emplear que, si  $0 \notin (x^L, x^U)$ , entonces un monomio es o convexo o cóncavo. Existen diferentes formas para identificar a qué grupo pertenece cada monomio, algunas de las cuales puede consultarse en el Teorema 5 de [Ryoo and Sahinidis \(1996\)](#). A partir de ahora se supondrá que  $0 \notin (x^L, x^U)$ .

Agrupando los términos convexos y los cóncavos, se puede reescribir el problema anterior para reflejar esta diferencia entre monomios:

$$\begin{aligned} \text{minimizar}_{x \in X \subseteq \mathbb{R}} \quad & f(\mathbf{x}) = \sum_{i \in cv} f_i(x) + \sum_{i \in cc} f_i(x) := f_{cv}(x) + f_{cc}(x) \\ \text{sujeto a} \quad & x^L \leq x \leq x^U \end{aligned} \quad (3.2)$$

donde *cv* denota al conjunto de índices de los monomios convexos y *cc* al conjunto de índices de los monomios cóncavos.

La función  $f_{cv}$  es por definición convexa. Para  $f_{cc}$ , cóncava por definición, es posible construir una relajación lineal de tal manera que al sumar esta relajación con  $f_{cv}$  se genere una función  $\bar{f}$  convexa. Esta relajación se puede obtener a partir de la restricción de rango de la variable y del valor que toma la parte cóncava en ellos. Considerando

$$\beta = \frac{f_{cc}(x^U) - f_{cc}(x^L)}{x^U - x^L}, \quad \alpha = f_{cc}(x^L) - \beta x^L, \quad (3.3)$$

entonces  $g(x) = \alpha + \beta x$  es una función lineal que cumple que  $g(x) \leq f_{cc}(x)$  para todo  $x \in [x^L, x^U]$ .

Sumando esta función  $g(x)$  a la parte convexa de  $f$ , se obtiene la expresión de una relajación convexa del problema original:

$$\begin{aligned} \text{minimizar}_{x \in \mathbb{R}} \quad & \bar{f}(x) = f_{cv}(x) + g(x) \\ \text{sujeto a} \quad & x^L \leq x \leq x^U. \end{aligned} \quad (3.4)$$

**Ejemplo 3.5.** Para un caso sencillo es posible calcular  $\bar{f}$  y representarla gráficamente. Considérese el siguiente problema:

$$\begin{aligned} \text{minimizar}_{x \in X \subseteq \mathbb{R}} \quad & f(x) = x^3 + 2x^2 + 1 \\ \text{sujeto a} \quad & -2 \leq x \leq 1. \end{aligned} \quad (3.5)$$

Trivialmente se puede separar el polinomio  $f(x)$  en su parte convexa  $f_{cv}(x) = 2x^2 + 1$  y su parte cóncava  $f_{cc}(x) = x^3$ .

Aplicando la fórmula 3.3, se obtiene una función lineal que relaja la parte convexa:

$$\left. \begin{aligned} \beta &= \frac{f_{cc}(1) - f_{cc}(-2)}{1 - (-2)} = 3 \\ \alpha &= f_{cc}(-1) - \beta(-1) = -2 \end{aligned} \right\} \implies g(x) = \alpha + \beta x = -2 + 3x.$$

Finalmente, sumando la parte convexa, se llega a que la función buscada es

$$\bar{f}(x) = 2x^2 + 3x - 1.$$

En la Figura 3.1 se representa el polinomio  $f(x)$  y el rango de  $x$   $[-2, 1]$  (área sombreada). En la parte inferior se muestra la función  $g(x)$ , la cual es una recta que cumple, para cualquier punto de ese área, que  $g(x) \leq f(x)$ . Finalmente se muestra la relajación  $\bar{f}$ , que toma forma de una parábola cumpliendo que  $\bar{f}(x) \leq f(x)$  en el rango de  $x$ .

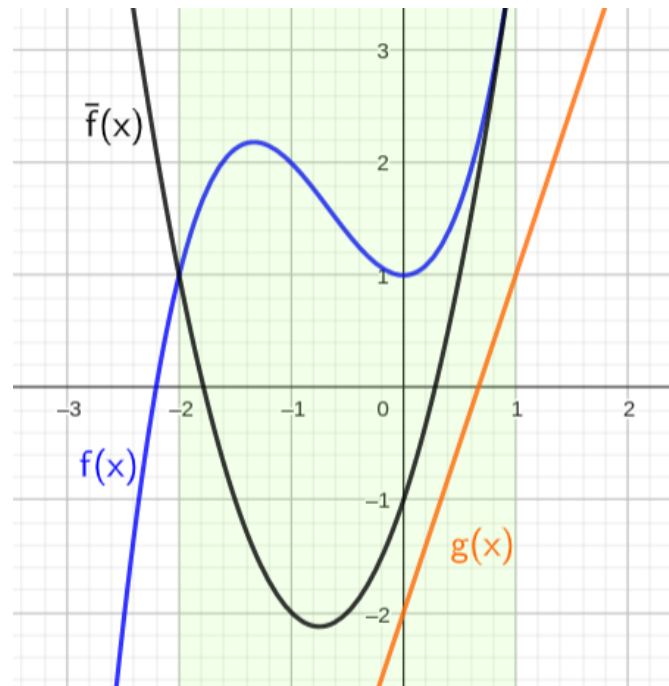


Figura 3.1: Representación gráfica de las funciones involucradas.

La relajación 3.4 puede resolverse utilizando métodos de optimización sin restricciones para funciones convexas, como el *método de Newton-Raphson*, que permite obtener de manera rápida el mínimo de una función. Además, como consecuencia de la convexidad de  $\bar{f}$ , se cumple el siguiente resultado:

**Proposición 3.6.** *Sea  $R_i$  el subproblema que está procesando el algoritmo de RyR en el momento actual, y sean  $x_i^L$  y  $x_i^U$  las cotas inferiores y superiores respectivamente del rango de dicho subproblema. Se tienen las siguientes condiciones:*

- Si  $\bar{f}'(x_i^L) \geq 0$ , entonces  $x_i^L$  es una solución óptima de  $R_i$ .
- Si  $\bar{f}'(x_i^U) \leq 0$ , entonces  $x_i^U$  es una solución óptima de  $R_i$ .

Este resultado permite modificar la fase de acotación del algoritmo, de manera que en algunos casos no sea necesario resolver el subproblema  $R_i$  para obtener su solución. Si bien esto no reduce el dominio de búsqueda de solución, sí permite acelerar la ejecución de RyR, pues el número de subproblemas a resolver a lo largo de la ejecución es menor.

Empleando el resultado anterior y la construcción de la función  $\bar{f}$  explicada en 3.4 se pueden modificar las etapas de inicialización y acotación (Paso 3) de RyR para que el algoritmo aproveche las particularidades de los problemas con polinomios univariados. En el Algoritmo 3 se muestran los pasos que habría que sustituir en el Algoritmo 2 original de RyR.

---

**Algoritmo 3** Inicialización y Acotación de RyR para problemas con polinomios univariados

---

**Inicialización**

Sea  $L^{(0)} = -\infty$  y  $U^{(0)} = +\infty$

Crear la relajación convexa  $R$  del problema original siguiendo 3.4.

**if**  $0 \in (x^L, x^U)$  **then**

Ramificar la relajación  $R$  en dos subproblemas:  $R_1$  definido en  $[x^L, 0]$  y  $R_2$  definido en  $[0, x^U]$ .

Añadir  $R_1$  y  $R_2$  a la lista *ACTIVOS* de problemas activos

**else**

Añadir la relajación  $R$  a la lista *ACTIVOS* de problemas activos

**end if**

**Paso 3: Acotación**

**if**  $\bar{f}'(x_i^L) \geq 0$  **then**

Tomar  $x_i^L$  como solución del subproblema  $R_i$ :  $x^i \leftarrow x_i^L$

**else if**  $\bar{f}'(x_i^U) \leq 0$  **then**

Tomar  $x_i^U$  como solución del subproblema  $R_i$ :  $x^i \leftarrow x_i^U$

**else**

Hallar la solución  $x^i$  de  $R_i$  empleando el método de Newton-Raphson.

**end if**

Fijar  $L_i = \bar{f}(x^i)$

---

### 3.1.2. Problemas lineales multiplicativos

Otro tipo de problemas de optimización polinómica que pueden ser adaptados para el empleo de RyR son los denominados *problemas lineales multiplicativos*. Estos tienen la siguiente forma:

$$\begin{aligned} \text{minimizar}_{\mathbf{x} \in X \subseteq \mathbb{R}^n} \quad & f(\mathbf{x}) = \prod_{i=1}^p f_i(\mathbf{x}) = \prod_{i=1}^p (\mathbf{c}_i^T \mathbf{x} + c_{i0}) \\ \text{sujeto a} \quad & \mathbf{A}\mathbf{x} \leq \mathbf{b} \\ & \mathbf{c}_i^T \mathbf{x} + c_{i0} > 0 \quad i = 1, \dots, p, \end{aligned} \quad (3.6)$$

donde  $\mathbf{A} \in \mathbb{R}^{m \times n}$ ,  $\mathbf{b} \in \mathbb{R}^m$ ,  $\mathbf{c}_i \in \mathbb{R}^n$  y  $c_{i0} \in \mathbb{R}$ , con  $i = 1, \dots, p$ .

Por tanto, se trata de problemas donde las restricciones están definidas por funciones lineales, pero la función objetivo viene dada por el producto de polinomios de primer grado, lo que provoca que realmente sea un polinomio de grado  $p$  (normalmente con  $p > 1$ ) y no pueda considerarse como un problema lineal (y seguramente tampoco convexo), pero sí como uno polinómico.

**Ejemplo 3.7.** El siguiente problema es un ejemplo de problema lineal multiplicativo:

$$\begin{aligned} \text{minimizar}_{\mathbf{x} \in X \subseteq \mathbb{R}^n} \quad & (2x_1 - 3x_2 + 3) \cdot (4x_2 + 25x_3 + 6) \cdot (-8x_3 - 1) \\ \text{sujeto a} \quad & x_1 + 3x_2 \leq 24 \\ & -x_1 + 18x_3 \leq -5 \\ & 2x_1 - 3x_2 + 3 > 0 \\ & 4x_2 + 25x_3 + 6 > 0 \\ & -8x_3 - 1 > 0 \end{aligned} \quad (3.7)$$

donde

$$\mathbf{A} = \begin{bmatrix} 1 & 3 & 0 \\ -1 & 0 & 18 \end{bmatrix}, \quad \mathbf{b} = (24, -5)$$

$$\mathbf{c}_1 = (2, -3, 0), \quad \mathbf{c}_2 = (0, 4, 25), \quad \mathbf{c}_3 = (0, 0, -8)$$

$$c_{10} = 3, \quad c_{20} = 6, \quad c_{30} = -1.$$

Para poder aplicar el algoritmo de RyR, el primer paso es construir una relajación que cumpla las propiedades explicadas en (R). Para facilitar el trabajo, se realizará la siguiente transformación al problema:

$$\begin{aligned} \text{minimizar}_{\mathbf{t} \in \mathbb{R}^p} \quad & \ln(f(\mathbf{t})) = \sum_{i=1}^p \ln(t_i) \\ \text{sujeto a} \quad & \mathbf{A}\mathbf{x} \leq \mathbf{b} \\ & \mathbf{c}_i^T \mathbf{x} + c_{i0} = t_i \quad i = 1, \dots, p, \\ & t_i > 0 \quad i = 1, \dots, p, \end{aligned} \quad (3.8)$$

Por ser el logaritmo una función monótona, el problema anterior es equivalente al problema de partida. Esto quiere decir que, si se obtiene una solución, existe una solución en el problema original con el mismo valor que el obtenido.

Además, por ser el logaritmo una función cóncava, puede obtenerse una relajación lineal de cada sumando de la función objetivo sustituyendo dicha función por la relajación lineal, de una manera similar a como se hizo con el caso de polinomios univariados. Este problema cumple las condiciones para ser una relajación válida sobre la que aplicar RyR:

$$\begin{aligned} \text{minimizar}_{\mathbf{t} \in \mathbb{R}^p} \quad & \bar{f}(\mathbf{t}) = \sum_{i=1}^p \alpha_i + \beta_i t_i \\ \text{sujeto a} \quad & \mathbf{A}\mathbf{x} \leq \mathbf{b} \\ & \mathbf{c}_i^T \mathbf{x} + c_{i0} = t_i \quad i = 1, \dots, p, \\ & t_i > 0 \quad i = 1, \dots, p, \end{aligned} \tag{3.9}$$

donde

$$\beta_i = \frac{\ln(t_i^U) - \ln(t_i^L)}{t_i^U - t_i^L}, \quad \alpha_i = \ln(t_i^L) - \beta_i t_i^L. \tag{3.10}$$

Por como se construye la relajación lineal es necesario obtener, para cada  $i = 1, \dots, p$ , las cotas superiores  $t_i^U$  e inferiores  $t_i^L$  de cada variable  $t_i$ , la cual representa el término multiplicativo  $i$ -ésimo del problema original. Para hallar estos valores se puede añadir un preprocesamiento en la fase de inicialización de RyR:

1. Minimizar individualmente cada  $t_i$  en el problema original, a fin de obtener una cota inferior  $t_i^L$ . Esto se logra resolviendo para cada  $t_i$  el problema lineal

$$\begin{aligned} \text{minimizar}_{\mathbf{x} \in X \subseteq \mathbb{R}^n} \quad & t_i(\mathbf{x}) = c_i^T \mathbf{x} + c_{i0} \\ \text{sujeto a} \quad & \mathbf{A}\mathbf{x} \leq \mathbf{b} \\ & c_i^T \mathbf{x} + c_{i0} > 0 \quad i = 1, \dots, p. \end{aligned} \tag{3.11}$$

Por ser problemas lineales pueden resolverse rápidamente empleando los algoritmos disponibles para este tipo de problemas, como el del Simplex.

2. Una vez resueltos todos los problemas lineales se puede obtener una cota superior al problema de partida. Esta cota proviene de evaluar la función objetivo original  $f$  en las soluciones  $\mathbf{x}^i$  de cada problema, siendo  $U = \min_{i=1, \dots, p} \{f(\mathbf{x}^i)\}$  dicha cota.
3. A partir de  $U$  se puede obtener el corte  $f(\mathbf{x}) = \prod_{i=1}^p t_i \leq U$  del cual, separando el término  $t_i$  del resto del productorio, se llega a la desigualdad

$$t_i \leq \frac{U}{\prod_{j=1, j \neq i}^p t_j} \quad i = 1, \dots, p. \tag{3.12}$$

4. Finalmente, teniendo en cuenta que  $t_j^L \leq t_j \leq t_j^U$  y por tanto  $\frac{1}{t_j^U} \leq \frac{1}{t_j} \leq \frac{1}{t_j^L}$  para todo  $j = 1, \dots, p$ , se llega a la cota superior buscada, dependiente del valor de  $U$  y de las cotas inferiores del resto de términos  $t_i$ :

$$t_i^U \leq \frac{U}{\prod_{j=1, j \neq i}^p t_j^L} \quad i = 1, \dots, p. \quad (3.13)$$

La desigualdad 3.13 puede utilizarse, además de para crear la relajación, para reducir el dominio del subproblema  $R_i$  que se esté procesando en dicho momento, sustituyendo los valores  $t_j^L$  por los que se tengan en dicho momento.

En el Algoritmo 4 se muestra cómo quedaría la fase de inicialización de RyR para este tipo de problemas. En el Paso 4: Reducción de dominio, como se acaba de comentar, se podría añadir a las desigualdades presentadas en el capítulo anterior la desigualdad 3.13 y tratarla como el resto.

---

**Algoritmo 4** Inicialización de RyR para problemas lineales multiplicativos

---

**Inicialización**

Sea  $L^{(0)} = -\infty$

Minimizar individualmente cada  $t_i$  sujeto a las restricciones del problema. Para cada  $i = 1, \dots, p$ , sea  $\mathbf{x}^i$  el vector solución y  $t_i^L$  el valor de dicha solución.

Fijar  $U^0 = \min_{i=1, \dots, p} \{f(\mathbf{x}^i)\}$ .

Obtener los valores de  $t_i^U$  empleando la expresión 3.13 y utilizarlos junto con  $t_i^L$  para calcular la expresión de  $f(\tilde{\mathbf{t}})$ , siguiendo 3.9 y 3.10.

Incluir el problema resultante  $R_1$  en la lista *ACTIVOS* de problemas activos.

---

## 3.2. El algoritmo RLT

Si bien RyR permite resolver problemas de optimización polinómica, existen algoritmos especialmente pensados para este tipo de problemas, siendo la *Reformulation-Linearization Technique* (RLT) de [Sherali and Tuncbilek \(1992\)](#) uno de ellos. Al igual que RyR, se basa en la generación de relajaciones del problema inicial, si bien en este caso dichas relajaciones serán problemas lineales y no simplemente convexos. Empleando un esquema básico de RyA y una regla de ramificación adecuada se puede acotar el óptimo global del problema original a partir de la resolución de las relajaciones, de manera similar a como sucede con RyR.

De ahora en adelante se usará una formulación diferente de un problema de optimización modificando las desigualdades e igualdades presentes. Esta formulación es equivalente a la empleada

hasta el momento, tal y como se vio en la Observación 1.11:

$$\begin{aligned}
& \text{minimizar} && f(\mathbf{x}) \\
& \text{sujeto a} && g_i(\mathbf{x}) \geq \beta_i, \quad i = 1, \dots, R_1 \\
& && h_i(\mathbf{x}) = \beta_i, \quad i = R_1 + 1, \dots, R \\
& && \mathbf{x} \in \Omega \subset \mathbb{R}^n,
\end{aligned} \tag{3.14}$$

donde:

- $g_i(\mathbf{x})$  y  $h_i(\mathbf{x})$  son polinomios de grado  $\delta_i$ .
- $\Omega = \{\mathbf{x} : 0 \leq x_j^L \leq x_j \leq x_j^U < \infty \text{ para todo } j = 1, \dots, n\}$  es un hiperrectángulo que contiene la región factible del problema.

La idea principal tras RLT se encuentra en la sustitución de cada monomio diferente que compone un polinomio por una única variable que lo represente, de tal manera que se eliminen los productos entre variables (o entre una misma variable). Dicha variable, denominada *variable RLT*, será de la forma  $X_{ij}$ , donde los índices  $i$  y  $j$  representan los índices asociados a todas las variables que forman el monomio y pueden aparecer repetidos para representar multiplicidades. El siguiente ejemplo ilustra cómo se construyen dichas variables.

**Ejemplo 3.8.** Considérese el siguiente problema de optimización polinómica:

$$\begin{aligned}
& \text{minimizar} && 5x_1 + 3x_2^2 - 3x_1x_2x_3 + 4x_2x_3^3 - 5x_1x_2 \\
& \text{sujeto a} && -x_1 - x_1x_2 - 3 \leq 0 \\
& && 3x_2^2 - x_2 \leq 0.
\end{aligned} \tag{3.15}$$

Para cada monomio diferente del problema anterior, obviando constantes multiplicativas, existe una variable RLT asociada, las cuales pueden encontrarse en varios polinomios diferentes:

$$X_{22} = x_2^2 \quad X_{123} = x_1x_2x_3 \quad X_{2333} = x_2x_3^3 \quad X_{12} = x_1x_2.$$

Sustituyendo en el problema anterior la variables RLT y teniendo en cuenta que los términos lineales son su propia variable RLT, se llegaría al siguiente problema lineal:

$$\begin{aligned}
& \text{minimizar} && 5x_1 + 3X_{22} - 3X_{123} + 4X_{2333} - 5X_{12} \\
& \text{sujeto a} && -x_1 - X_{12} - 3 \leq 0 \\
& && 3X_{22} - x_2 \leq 0.
\end{aligned} \tag{3.16}$$

Para que el algoritmo funcione mejor y converja antes se añaden una serie de restricciones conocidas como *bounding factors*, las cuales se forman a partir de la combinación de las restricciones de rango del problema original. Para cada variable  $x_j$  del problema, con  $j = 1, \dots, n$ , considérense sus restricciones de rango  $x_j - x_j^L \geq 0$  y  $x_j^U - x_j \geq 0$ . Siendo  $\delta$  el grado del polinomio con grado más alto en el problema, los bound factors se obtienen realizando todas las

combinaciones posibles que se obtienen al multiplicar  $\delta$  restricciones de rango, incluyendo los productos de una restricción consigo misma o con otra restricción multiplicada varias veces. A continuación se incluye un ejemplo para ilustrar su construcción en un caso sencillo.

**Ejemplo 3.9.** Sea un problema de optimización polinómica en las variables  $y$  con  $\delta = 2$  y con las restricciones de rango

$$x_1 - 2 \geq 0, \quad 3 - x_1 \geq 0, \quad x_2 - 21 \geq 0, \quad 43 - x_2 \geq 0.$$

Los bounding factors asociados a este problema serían los siguientes, originados de combinar todas las restricciones anteriores entre sí:

$$\begin{aligned} &\bullet (x_1 - 2)(x_1 - 2) \geq 0 \quad \bullet (x_1 - 2)(3 - x_1) \geq 0 \quad \bullet (x_1 - 2)(x_1 - 21) \geq 0 \quad \bullet (x_1 - 2)(43 - x_2) \geq 0 \\ &\bullet (3 - x_1)(3 - x_1) \geq 0 \quad \bullet (3 - x_1)(x_2 - 21) \geq 0 \quad \bullet (3 - x_1)(43 - x_2) \geq 0 \\ &\bullet (x_2 - 21)(x_2 - 21) \geq 0 \quad \bullet (x_2 - 21)(43 - x_2) \geq 0 \\ &\bullet (43 - x_2)(43 - x_2) \geq 0 \end{aligned}$$

Al conjunto de todos los bounding factors se le denomina *bound factors* y se denotará como  $F_\delta$ . Nótese que todos los puntos del conjunto  $\Omega$  satisfacen los bound factors.

Una vez explicadas las variables RLT y los bound factors, ya es posible obtener la relajación lineal que utiliza RLT:

$$\begin{aligned} &\text{minimizar} && [f(\mathbf{x})]_L \\ &\text{sujeto a} && [g_i(\mathbf{x})]_L \geq \beta_i, \quad i = 1, \dots, R_1 \\ &&& [h_i(\mathbf{x})]_L = \beta_i, \quad i = R_1 + 1, \dots, R \\ &&& [F_\delta]_L \geq 0 \\ &&& \mathbf{x} \in \Omega \subset \mathbb{R}^n, \end{aligned} \tag{LP}$$

donde  $[p(\mathbf{x})]_L$  se corresponde con la linealización del polinomio  $p(\mathbf{x})$  obtenida al sustituir sus monomios por las variables RLT asociadas, y  $[F_\delta]_L$  se corresponde con la linealización de los bound factors.

Para asegurar la convergencia de RLT a un óptimo del problema original en un número finito de iteraciones, cuya demostración puede encontrarse en el Teorema 1 de [Sherali and Tuncbilek \(1992\)](#), además de emplear la relajación (LP) es necesario emplear una determinada regla de ramificación. Cabe mencionar que la convergencia del esquema de RyA subyacente está asegurada de manera similar a como sucedía en RyR, pues la operación de acotación es consistente y la de selección ajusta cotas ya que utiliza la misma regla que RyR, que por definición ajustaba cotas.

La regla de ramificación de RLT se basa en escoger como variable de ramificación aquella que provoque una mayor infactibilidad en el problema original, pudiendo medirse esta infactibilidad

como la diferencia entre una variable RLT,  $X_{ijk}$ , y el producto de la variable original  $x_i$  y la variable RLT sin ella,  $X_{jk}$ . Suponiendo que el vector  $(\bar{\mathbf{X}}, \mathbf{x})$  es solución óptima de la relajación  $(\mathbf{LP})$ , siendo  $(\bar{\mathbf{X}}, \mathbf{x})$  un vector formado por un determinado conjunto  $\bar{\mathbf{X}}$  de variables RLT y por un conjunto  $\mathbf{x}$  de variables originales del problema, la medida de infactibilidad indica que se debe ramificar por la variable  $x_p$ , donde:

$$p \in \operatorname{argmax}_{j \in N} \{\theta_j\}, \text{ siendo}$$

$$\theta_j = \max_{t \in \{1, \dots, \delta-1\}} \max_{J \subset N, |J|=t} \{|\bar{X}_{J \cup j} - \bar{x}_j \bar{X}_J|\}$$

para todo  $j \in N = \{1, \dots, n\}$ .

*Observación 3.10 (Relación entre RLT y Ramificación y Reducción).* La linealización  $(\mathbf{LP})$  puede emplearse también como relajación de inicio en RyR, pues cumple las condiciones impuestas sobre la relajación  $(\mathbf{R})$ , válida para este algoritmo:

- $(\mathbf{LP})$  es un problema de optimización lineal, con lo cual también lo es de optimización convexa.
- $\Omega$  está acotado y  $[f(\mathbf{x})]_L$  es continua por ser lineal.
- Por el Lema 1 de [Sherali and Tuncbilek \(1992\)](#), para cualquier punto  $\mathbf{x}$  factible para el problema no linealizado se cumple que  $[f(\mathbf{x})]_L \leq f(\mathbf{x})$ .
- El dominio de  $(\mathbf{LP})$  es igual o mayor que el dominio del problema original, con lo que cualquier solución del problema original será solución de la relajación.

Al contrario que las relajaciones para RyR mostradas en el apartado anterior, especializadas en un problemas con unas características muy específicas, esta nueva relajación es válida para cualquier problema de optimización polinómica, pudiendo ser utilizada para crear la relajación inicial con la que empezar el proceso de ramificación y reducción. Este hecho también se puede utilizar en la dirección contraria, incluyendo las desigualdades de reducción de dominio empleadas en RyR dentro de RLT, ya que si la relajación RLT cumple las condiciones de  $(\mathbf{R})$  entonces se pueden aplicar todas las desigualdades derivadas del Teorema 2.7. De esta manera se muestra como RyR y RLT son algoritmos complementarios, pudiendo extraer conceptos de un algoritmo para incluirlos en el otro, lo cual se empleará en el siguiente capítulo para estudiar el impacto de las desigualdades de reducción de dominio dentro de una implementación del algoritmo RLT.



## Capítulo 4

# Resultados experimentales

A lo largo de este trabajo se han presentado diferentes algoritmos que buscan resolver problemas de optimización polinómica. Junto con ellos se han estudiado técnicas para reducir el dominio de dichos problemas, lo que permite resolverlos de una manera más sencilla. Si bien esto es interesante desde un punto de vista teórico, su función final es mejorar la resolución computacional de estos, bien sea acelerando el proceso o reduciendo la memoria y la potencia necesaria.

En este capítulo se realizará un pequeño estudio sobre cómo las técnicas de reducción de dominio permiten acelerar la resolución de problemas de optimización polinómica, centrándose en las desigualdades de reducción de dominio y las herramientas de bound tightening (FBBT y OBBT). El estudio se realizará tomando como base una implementación concreta de RLT especializada en la resolución de un subtipo de problemas polinómicos, RAPOSa, creada por miembros del *Centro de Investigación y Tecnología Matemática de Galicia (CITMAGA)*, con los cuales se colaboró en el transcurso de unas prácticas extracurriculares implementando y validando las desigualdades de reducción de dominio.

### 4.1. RAPOSa

RAPOSa (**R**eformulation **A**lgorithm for **P**olynomial **O**ptimization - **S**antiago) es un optimizador global diseñado específicamente para la resolución de problema de optimización polinómica donde las variables estén acotadas. RAPOSa está implementando en *C++* y se basa en el algoritmo RLT explicado en el Capítulo 3, al cual se le añaden una serie de mejoras que se pueden consultar en [González-Rodríguez et al. \(2023\)](#).

La estructura del algoritmo RLT presente en RAPOSa puede consultarse en la Figura 4.1. Si bien a simple vista es muy similar a la estructura de ramificación y reducción mostrada en la Figura 2.4, existen grandes diferencias entre ambos algoritmos, pudiendo destacarse las siguientes:

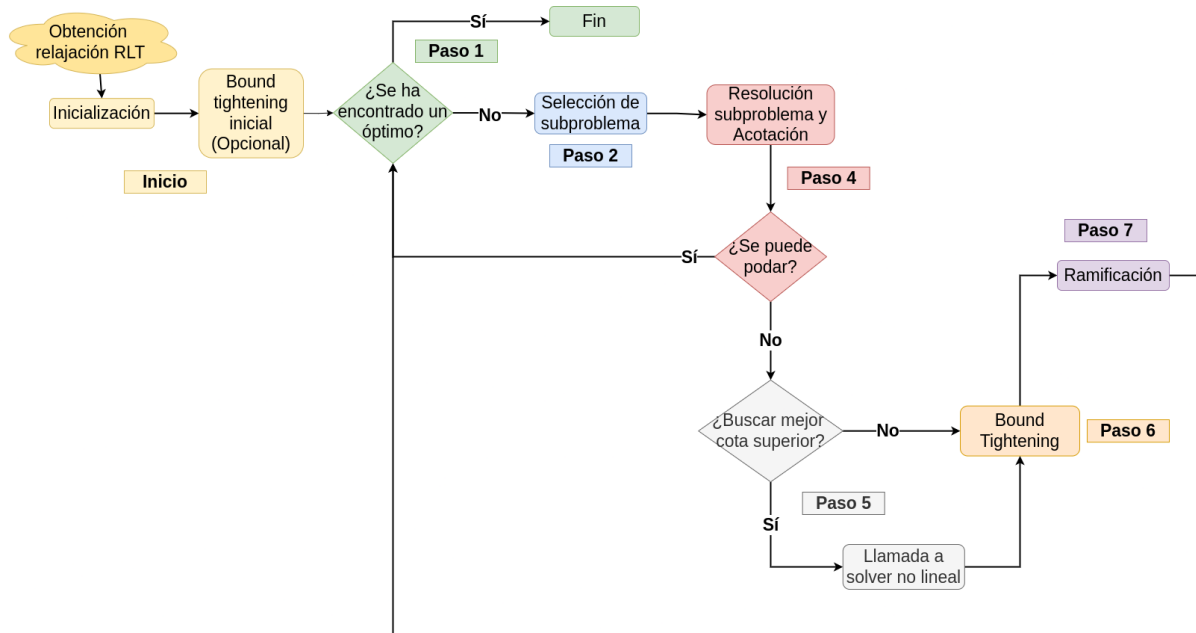


Figura 4.1: Diagrama de flujo de RAPOSa, manteniendo la notación usada en RyR

- **Relajación:** La relajación empleada por RAPOSa es la relajación (LP) propuesta por RLT, si bien esta era una de las relajaciones válidas para iniciar también el algoritmo de RyR.
- **Inicio:** Se añade una etapa adicional opcional de bound tightening en la cual se puede aplicar OBBT y/o FBBT según se considere, buscando restringir desde el inicio las cotas de las variables.
- **Paso 3:** Se elimina la etapa específica de FBBT antes de la resolución de la relajación, pues ya se aplica esta técnica en el nodo padre en alguna de las etapas de bound tightening (en la inicialiación o en el Paso 6).
- **Paso 5:** Para buscar una cota mejor cota superior global, en RAPOSa se puede realizar una llamada a un solver no lineal local, el cual intenta encontrar una solución para el estado actual del problema.
- **Paso 6:** En este paso, renombrado como *Bound Tightening* para diferenciarlo de la reducción de dominio de RyR, se intentan reducir las cotas de las variables y ajustar las cotas de las restricciones. Para ello se utilizará el *FBBT*, el cual se emplea de manera diferente a co-

mo aparece en RyR, ya que obtiene información adicional de la resolución de la relajación, y por eso se incluye después de que se realice esta en el Paso 4.

- **Paso 7:** Para realizar la ramificación se utiliza la regla explicada en el capítulo anterior o alguna otra de las presentes en [González-Rodríguez et al. \(2023\)](#).
- **Eliminación del ciclado:** La técnica RLT no contempla el ciclado de reducción de dominio entre los pasos 4 y 6 característico de RyR, tampoco incluyéndose en RAPOSa.

## 4.2. Estudio computacional

Las técnicas de reducción de dominio estudiadas en este trabajo buscan acelerar el incremento de la cota inferior, lo cual provoca que el *gap* (diferencia entre la cota superior y la cota inferior) se reduzca antes. Esta aceleración puede medirse de manera empírica observando el número de subproblemas generados, los cuales a partir de ahora se denominarán *nodos*. Sin embargo, en ocasiones esta reducción no es beneficiosa, pudiendo suceder que se genere un árbol de ramificación más pequeño pero con nodos de mayor complejidad, provocando que el *gap* disminuya más lentamente que otro con más nodos pero donde las relajaciones a resolver son más sencillas. Además, es necesario tener en cuenta el sobrecoste computacional de estas operaciones de reducción de dominio, pues no siempre es rentable respecto a la mejora que suponen.

En este apartado se estudiará cómo las desigualdades de reducción de dominio ayudan a reducir el número de nodos generados y su relación con el tiempo de ejecución y la capacidad de resolución de los problemas. Para ello, se tomarán varias configuraciones de RAPOSa que emplean diferentes técnicas para mejorar el rendimiento en la resolución de problemas:

- *Sin BT*: Ejecución del flujo de RAPOSa sin realizar ninguna clase de bound tightening.
- *Sólo OBBT*: Ejecución realizando solamente OBBT en el bound tightening inicial y sin realizar la etapa específica bound tightening.
- *Sólo FBBT*: Ejecución realizando FBBT en la etapa de bound tightening general.
- *Baseline BT*: Ejecución completa del flujo de RAPOSa, aplicando OBBT y FBBT en la etapa de inicialización y FBBT en la de bound tightening.
- *RyR*: Ejecución de todas las opciones incluidas en *Baseline BT*, añadiendo dentro de la fase de bound tightening algunas de las desigualdades de reducción de dominio empleadas en el algoritmo de ramificación y reducción (de ahí el nombre de esta configuración). En este caso solo se consideran las desigualdades asociadas a restricciones activas, ya que las que necesitan de *probing* no están implementadas en RAPOSa.

Las configuraciones anteriores se utilizarán para la resolución de dos conjuntos de problemas de optimización polinómica de características diferentes ampliamente usados en la literatura. El primero de ellos es la librería DS-TS [Dalkiran and Sherahli \(2016\)](#), compuesta por 180 problemas

generados de forma aleatoria de grado 2 a grado 7 con un gran número de restricciones. El otro conjunto es la librería MINLPLib [Bussieck et al. \(2003\)](#), la cual incluye 166 problemas surgidos de contextos reales, hecho que la hace más interesante que la librería anterior pues permite obtener resultados para problemas similares a los que se podría enfrentar la herramienta en un supuesto de uso sobre problemas reales.

Para analizar el impacto de cada configuración en la resolución de los problemas se obtendrán una serie de medidas durante este proceso. Cada una de ellas es calculada empleando un número diferente de problemas, eliminando aquellos cuyas características podrían distorsionar los resultados. Las medidas son la siguientes:

- *Gap*: gap medio entre todos los problemas. Por lo general un problema de optimización se da por resuelto cuando el gap es menor que un determinado  $\delta > 0$ , el cual es un valor de gap normalizado de forma que no dependa del rango de los valores de las variables del problema, ya que no es lo mismo tratar con variables que puedan tomar valores en el rango  $[0, 2000]$  que en el rango  $[0, 1]$ .
- *Tiempo*: tiempo medio de resolución de los problemas expresado en segundos.
- *Nodos*: número medio de nodos (subproblemas) generados. En este caso es la fila más importante, pues es la métrica que permite medir la influencia de las desigualdades de reducción de dominio.
- *Resueltos*: Número de problemas resueltos en el tiempo máximo de ejecución asignado para la resolución de cada problema. Si no se ha encontrado una solución óptima en menos de ese tiempo, el problema se marca como no resuelto, aunque dicho óptimo pueda existir.

En la Tabla 4.1 se muestran las medidas anteriores asociadas a la ejecución durante un máximo de una hora de las configuraciones explicadas con los problemas de MINLPLib. Las configuraciones se ordenan de menor a mayor complejidad de izquierda a derecha, siendo la primera la más básica (*Sin BT*) y la última la más compleja (*RyR*, pues incluye elementos del resto). Si se analizan los resultados se puede ver como todas las medidas disminuyen cuando más a la derecha esté la configuración, a excepción del número de problemas resueltos, que aumenta. Esto significa que cada configuración está mejorando a la anterior, disminuyendo el gap, tiempo y número de nodos medios pero resolviendo un número mayor o igual de problemas.

	Sin BT	Sólo OBBT	Sólo FBBT	Baseline BT	RyR
<b>Gap</b>	0.427744	0.404498	0.343428	0.333237	0.332244
<b>Tiempo</b>	1154.754060	1151.226465	938.811309	916.968279	912.228445
<b>Nº Nodos</b>	55909.778761	19084.362832	9462.274336	8232.362832	6696.964602
<b>Resueltos</b>	115.000000	115.000000	126.000000	126.000000	126.000000

Tabla 4.1: Resultados de las diferentes configuraciones en la librería MINLPLib

Si bien ya se pueden extraer conclusiones a partir de los datos anteriores, para facilitar su interpretación se incluye la Tabla 4.2, el cual muestra la diferencia en tanto por cien de cada medida respecto a la configuración base *Sin BT*. Los tonos verdes indican una mejora respecto a la versión básica, estando destacada en un tono más intenso aquella con el mayor porcentaje de mejora. En este caso se puede ver que, para los problemas de MINLPLib, *RyR* es la mejor configuración ya que obtiene la mayor disminución en todas las medidas sin resolver un número de menor de problemas. Además, observando los porcentajes concretos se puede ver como el mayor aumento se produce al añadir FBBT a una opción, pasando por ejemplo la disminución de tiempo de *Sólo OBBT* de  $-0.31\%$  a  $-20.59\%$  si se añaden el empleo de FBBT, como se puede ver en el *Baseline BT*.

	Sin BT	Sólo OBBT	Sólo FBBT	Baseline BT	RyR
<b>Gap</b>	0.0 %	-5.43 %	-19.71 %	-22.09 %	-22.33 %
<b>Tiempo</b>	0.0 %	-0.31 %	-18.7 %	-20.59 %	-21.0 %
<b>Nº Nodos</b>	0.0 %	-65.87 %	-83.08 %	-85.28 %	-88.02 %
<b>Resueltos</b>	0.0 %	0.0 %	9.57 %	9.57 %	9.57 %

Tabla 4.2: Mejoría porcentual respecto a la configuración de referencia en la librería MINLPLib

Sin embargo, si se observan los resultados asociados a la librería DS-TS en la Tabla 4.3, la situación es totalmente distinta. En este caso las medidas no parecen seguir la tendencia de disminución hacia la derecha que se veía en MINLPLib, sin ninguna configuración que concentre los mejores valores. De hecho, son las opciones más a la izquierda las que resuelven un mayor número de problemas (162) frente a los 159 del *Baseline BT* o los 154 de *RyR*.

	Sin BT	Sólo OBBT	Sólo FBBT	Baseline BT	RyR
<b>Gap</b>	0.002612	0.002624	0.002753	0.002766	0.002926
<b>Tiempo</b>	570.656682	569.559264	662.540975	600.037545	627.300479
<b>Nº Nodos</b>	2248.000000	1822.106667	2149.986667	1773.586667	1650.613333
<b>Resueltos</b>	162.000000	162.000000	157.000000	159.000000	154.000000

Tabla 4.3: Resultados de las diferentes configuraciones en la librería DS-TS

Analizando los cambios porcentuales se puede ver como la mayoría de las configuraciones no solo no mejoran a la versión base, si no que la empeoran, lo cual se muestra con tonos rojizos (destacando en este caso la peor opción). La única opción que podría considerarse como alternativa a la base es *Sólo OBBT*, pues resuelve el mismo número de problemas (162) disminuyendo ligeramente el tiempo, aunque aumenta el gap. Es muy llamativo lo que sucede con el número de nodos, pues si bien todas las configuraciones mejoran a la básica, en este caso el FBBT no es el

responsable, sino el OBBT (basta comparar los valores de esta medida en *Sólo OBBT* y *Baseline BT* con los de *Baseline BT*). Las desigualdades de reducción de dominio incluidas en *RyR* son efectivas para bajar el número de nodos (logrando esta opción la mayor disminución), pero esto no se ve reflejado en el resto de medidas, pues es la que obtiene un mayor gap medio y la que menos problemas resuelve. Estos resultados tan diferentes a los de MINLPLib son consecuencia de cómo están contruidos los problemas de DS-TS, cuyas cotas están bastante ajustadas de inicio y por tanto el gasto computacional empleado en reducirlas más no compensa la mejoría.

	Sin BT	Sólo OBBT	Sólo FBBT	Baseline BT	RyR
Gap	0.0 %	0.49 %	5.43 %	5.92 %	12.05 %
Tiempo	0.0 %	-0.19 %	16.1 %	5.15 %	9.93 %
Nº Nodos	0.0 %	-18.95 %	-4.36 %	-21.1 %	-26.57 %
Resueltos	0.0 %	0.0 %	-3.09 %	-1.85 %	-4.94 %

Tabla 4.4: Cambios porcentuales respecto a la configuración de referencia en la librería DS-TS

Este pequeño estudio permite ver como las técnicas de reducción de dominio, bien sean el FBBT, el OBBT, las desigualdades de reducción u otras, pueden ayudar a reducir el tiempo de resolución de los problemas de optimización polinómicos, pero solo en aquellos casos cuya complejidad sea lo suficientemente grande. Es necesario estudiar las características del problema a resolver antes de aplicar estas técnicas y seleccionar cuáles son las más adecuadas, pues puede suceder que utilizarlas solo añada un gasto temporal y computacional innecesario, incluso obteniendo resultados peores a los que se lograrían si se obviase su uso.

# Bibliografía

- Belotti, P., Cafieri, S., Lee, J., and Liberti, L. (2012). On feasibility based bounds tightening. [http://www.optimization-online.org/DB\\_HTML/2012/01/3325.html](http://www.optimization-online.org/DB_HTML/2012/01/3325.html).
- Boyd, S. P. and Vandenberghe, L. (2011). *Convex optimization*. Cambridge Univ. Pr.
- Bussieck, M. R., Drud, A. S., and Meeraus, A. (2003). Minlplib—a collection of test models for mixed-integer nonlinear programming. *INFORMS Journal on Computing*, 15(1):114–119.
- Dalkiran, E. and Sherali, H. (2016). Rlt-pos: Reformulation-linearization technique-based optimization software for solving polynomial programming problems. *Mathematical Programming Computation*, 8.
- González Díaz, J. (2022a). *Apuntes de Modelos y Técnicas de Optimización*. Universidad de Santiago de Compostela, Grado en Ingeniería Informática.
- González Díaz, J. (2022b). *Apuntes de Programación Lineal y Entera*. Universidad de Santiago de Compostela, Grado en Matemáticas.
- González Díaz, J. (2022c). *Apuntes de Programación Matemática*. Universidad de Santiago de Compostela, Máster en Técnicas Estadísticas.
- González-Rodríguez, B., Ossorio-Castillo, J., González-Díaz, J., Ángel M. González-Rueda, Penas, D. R., and Rodríguez-Martínez, D. (2023). Computational advances in polynomial optimization: RAPOSa, a freely available global solver. *Journal of Global Optimization*, 85(3):541–568.
- Horst, R. and Tuy, H. (1993). *Global Optimization: Deterministic Approaches*. Springer-Verlag, 2nd edition.
- Minoux, M. (1986). *Mathematical programming: Theory and applications*. Wiley.
- Ryoo, H. S. and Sahinidis, N. V. (1996). A branch-and-reduce approach to global optimization. *Journal of Global Optimization*, 8(2):107–138.

- Shen, P., Wang, K., and Lu, T. (2020). Global optimization algorithm for solving linear multiplicative programming problems. *Optimization*, 71(6):1421–1441.
- Sherali, H. D. and Tuncbilek, C. H. (1992). A global optimization algorithm for polynomial programming problems using a reformulation-linearization technique. *Journal of Global Optimization*, 2(1):101–112.