



Métodos computacionales para optimización local y global sin derivadas

por

Johanna Analiz Frau

Presentado ante la Facultad de Matemática, Astronomía, Física y Computación como parte de los requerimientos para la obtención del grado de Doctora en Matemática de la

Universidad Nacional de Córdoba

Julio 2021

Director: Dr. Elvio Angel Pilotta

Tribunal Especial

Titulares

Dra. Ana Gabriela Martínez (Departamento de Matemática, UFPR - Brasil)

Dr. Andrés Alberto Barrea (FAMAFA)

Dr. Juan Pablo Agnelli (FAMAFA)

Suplentes

Dr. Guillermo Alfredo Durán (Departamento de Matemática, UBA)

Dra. Noemí Patricia Kisbye (FAMAFA)

Dra. Ana Georgina Flesia (FAMAFA)



Métodos computacionales para optimización local y global sin derivadas por Johanna Analiz Frau se distribuye bajo una Licencia Creative Commons Atribución-NoComercial-CompartirIgual 4.0 Internacional.

Agradecimientos

A la UNC y UNSE por brindarme educación pública, gratuita y de calidad.

A CONICET por el apoyo económico para realizar mi doctorado.

Al CIEM por brindarme un lugar físico para desarrollar mi trabajo. Especialmente gracias a Nancy y Claudia por ponerle tanta buena onda a los trámites administrativos y burocráticos.

A docentes, no docentes y personal en general de FAMAF con quienes tuve la posibilidad de interactuar y donde encontré mucha calidez humana.

A Damián, Adrián y Germán, la comisión asesora de mi doctorado, por el acompañamiento estos años.

A Gabriela, Andrés y Juampi, tribunal de tesis, por la lectura tan detallada y las observaciones y comentarios que ayudaron a enriquecer la misma.

A todo el grupo de Análisis Numérico.

A Elvio, director de tesis, por la calidez humana, paciencia y aliento en los momentos de crisis.

A mis compañeros de la oficina 229/230 (que además me dieron el título de administradora).

A los amigos y amigas que me dió Córdoba: Lu, Eli, Ceci, Noe, Sonia, Iván, Augusto, Edward, Javier, Gaby, Mari, Euge, Meli, Carde, Vale ...

A Laura Alonso y Mili Teruel por la generosidad que tanto las caracteriza y por abrirme las puertas al mundo nuevo que hoy me apasiona. Agradezco mucho haberme cruzado con mujeres tan increíbles como ustedes!

A la Pi-people por el acompañamiento, crecimiento y principalmente por haber confiado en mi formación.

A mis compañeros de AML por el aguante en esta última etapa.

A Gisel por el soporte emocional, psicológico y por haberme ayudado a (re)construirme desde que trabajamos juntas.

A Mili Langhi. Sos muy especial en mi vida, gracias FAMAF y numérico por haber cruzado nuestras vidas.

A mis amigos y amigas de Santiago (y de la vida) que siempre estuvieron y están ahí: Pame, Lourdes, Mauri, Cristian, Andre, Flor, Anabel ...

A mis padres, Emma y Sergio, que apoyaron y acompañaron siempre mi educación y me dieron el todo el soporte emocional y económico necesario para iniciar mi carrera profesional allá por 2009.

A toda mi familia, por estar, acompañar y, principalmente, entender mis ausencias.

A mi abuela Chela, que decía que no iba a estar para verme recibida de Licenciada. Llegamos mucho más allá mi Chela querida!

A Luis, por tu infinita paciencia, por creer en mi cuándo ni siquiera yo era capaz de hacerlo. Gracias por estar y acompañar(nos).



Resumen

En la actualidad existen muchas aplicaciones de problemas de optimización de la vida real donde la evaluación de la función objetivo en diversos puntos es muy costosa o compleja computacionalmente, o bien la información disponible proviene de datos experimentales y por lo tanto no existe una expresión explícita de tal función. Dado que este tipo de problemas no puede ser resuelto mediante métodos clásicos de optimización, es decir aquellos que involucran el cálculo de derivadas, hace algunas décadas comenzó a desarrollarse un área dentro de la optimización conocida como “Optimización sin derivadas”. Básicamente, estos nuevos modelos y algoritmos, principalmente propuestos para optimización local, con y sin restricciones, están basados en técnicas relacionadas con exploración en entornos y movimientos hacia nuevos puntos que muestren alguna mejora en la evaluación de la función. En esta tesis doctoral presentamos y desarrollamos dos nuevos modelos de optimización sin derivadas.

En primer lugar, basándonos en los trabajos de Torczon et al. ([25, 31, 32]) y el trabajo de Diniz-Ehrhardt et al. [11], proponemos un nuevo método para *optimización local* con restricciones de cotas en las variables el cual utiliza una técnica de búsqueda lineal no monótona en la elección de los nuevos iterados. El nuevo método, cuyo algoritmo es denominado **nmps**, pertenece a la familia de los métodos de búsqueda de patrones y se demuestran resultados de convergencia global de primer orden. Esta convergencia se basa fuertemente en la medida de estacionariedad $\chi(x)$ propuesta por Coon et al. en [8]. Asimismo, esta nueva propuesta algorítmica fue acompañada por un extenso estudio numérico en el cual se utilizaron diferentes estrategias de búsqueda lineal no monótonas. Es importante mencionar que las mismas fueron inicialmente formuladas para resolver problemas de minimización sin restricciones y ecuaciones no lineales, con y sin derivadas, difiriendo así del problema de minimización con restricciones de cajas que resuelve el algoritmo **nmps**.

La segunda propuesta de esta tesis se encuentra enmarcada dentro de los métodos Lipschitzianos para *optimización global* propuestos inicialmente por Jones et al. en [24]. Por un lado, tomando como punto de partida el trabajo de Liuzzi et al. [35] y el trabajo de Paulavičius et al. [43], incorporamos el algoritmo **nmps** (optimización local) dentro del algoritmo BIRECT (optimización global) en la resolución de un problema de optimización con restricciones de cajas o cotas en las variables. Esta combinación de estrategias tuvo por objetivo determinar, mediante un estudio numérico, si la integración de un algoritmo de búsqueda local, del tipo búsqueda de patrones, dentro de un algoritmo de búsqueda global mejora la obtención del punto óptimo al explorar entornos de los iterados. Por otro lado, como última contribución, proponemos e implementamos un nuevo método para problemas de optimización global con restricciones generales que combina la estrategia de BIRECT [43] junto con las ideas propuestas por Jones [23] al adaptar DIRECT al problema con restricciones generales. El nuevo algoritmo hereda las propiedades de convergencia de los algoritmos de la familia DIRECT, las cuales se basan fuertemente en argumentos de densidad. De esta manera, el nuevo algoritmo *BIRECT con restricciones generales* utiliza la estrategia de bisección del espacio, propuesta por Paulavicius et al. [43], incorporando las restricciones generales de un modo similar a lo presentado por Jones en [23]. Debido a la com-

plejidad algorítmica de este problema de optimización global con restricciones y sin derivadas, esta parte del trabajo se caracteriza por un importante desarrollo computacional acompañado de un análisis detallado de la performance del nuevo algoritmo.

Abstract

Nowadays, there are many real life optimization applications where obtaining the value of the objective function in certain points is very expensive or computationally complex. In other cases, available information comes from experimental data and, therefore, there is not an explicit mathematical expression of the function. Since these kinds of problems can not be solved via classical optimization methods, i.e. those involving derivative calculations, a few decades ago a new field inside optimization began to develop. This new area is known as “Derivative-free optimization”. These new models and algorithms, mainly proposed for local optimization, for both constrained and unconstrained problems, are based on techniques related to neighborhoods exploration and movement towards new points that show some improvement in the functional value. In this doctoral thesis, we present and develop two new Derivative-free optimization models.

First, based on work by Torczon et al. ([25, 31, 32]) and Diniz-Ehrhardt et al. [11], we propose a new method for bound constrained *local optimization* that uses a non-monotone line-search technique for choosing a new iterate. The new method, whose algorithm is named **nmps**, belongs to the family of pattern search methods and global first order convergence results are proven. The convergence is strongly based on $\chi(x)$ measure, proposed by Coon et al. in [8]. Furthermore, this new algorithmic proposal was accompanied by an extensive numerical study in which different non-monotone line-search strategies were implemented. It is worth mentioning these strategies were originally proposed for solving unconstrained minimization problems and nonlinear equations, both involving derivatives and Derivative-free calculations. Besides, these methods differ from the bound constrained minimization problem that **nmps** algorithm solves.

During the second part of this thesis, we address lipschitzian methods for *global optimization*, originally proposed by Jones et al. in [24]. On one hand, based on works by Liuzzi et al. [35] and Paulavičius et al. [43], we incorporate the **nmps** algorithm (local optimization) inside the BIRECT algorithm (global optimization) to solve a bound constrained optimization problem. The main goal of this task was to determine if the integration of a local search algorithm (of the pattern search type) into a global search algorithm would improve the optimal point found. On the other hand, as a last contribution, we propose and implement a new method for solving global optimization problems with general constraints, which combine BIRECT’s strategy [43] along with the ideas proposed by Jones in [23], by adapting DIRECT to general constraints. The new algorithm inherits convergence properties from the DIRECT family of algorithms, which are strongly based on density arguments. In this way, the new algorithm *BIRECT with general constraints* uses the bisection strategy proposed by Paulavicius et al. [43] and incorporates general constraints in a similar way as Jones in [23]. Due to the algorithmic complexity of this Derivative-free global constrained optimization problem, this part is characterized by a major computational development along with a detailed analysis of the performance of the new algorithm.



Índice general

Índice general	X
Índice de figuras	XI
Índice de tablas	XIII
Índice de Algoritmos	XV
Notación	XVII
1. Fundamentos de optimización	1
1.1. Optimización sin restricciones	1
1.1.1. Algoritmos para optimización irrestricta: Estrategias de búsqueda lineal	2
1.2. Optimización con restricciones	4
1.2.1. Definiciones y resultados importantes	4
1.3. Optimización global	7
2. Algoritmos de optimización local sin derivadas para problemas con restricciones de cotas en las variables	9
2.1. Introducción	9
2.2. Preliminares: Métodos de búsqueda directa para optimización local	10
2.2.1. El algoritmo compass search: Caso irrestricto	12
2.2.2. El algoritmo compass search: Caso con restricciones de cajas	14
2.3. Nuevo algoritmo para minimización en cajas sin derivadas con búsqueda lineal no monótona.	16
2.3.1. Presentación	16
2.3.2. Algoritmo <code>nmps</code> : Resultados teóricos y análisis de convergencia	18
2.3.3. Experimentos numéricos	23
2.3.3.1. Comparación de performance profiles entre <code>nmps</code> y <code>patternsearch</code>	26
2.3.3.2. Comparación de performance profiles usando otras estrategias de búsqueda lineal	28
2.3.3.3. Experimentos numéricos adicionales	33
3. Algoritmos de optimización global con restricciones	37
3.1. Preliminares: Algoritmos de optimización global lipschitzianos	37
3.1.1. Algoritmo DIRECT irrestricto	37
3.1.1.1. DIRECT en una dimensión	37
3.1.1.2. DIRECT en dimensiones mayores que 1	41
3.1.2. Algoritmo BIRECT irrestricto	43

3.1.2.1.	Las bases de BIRECT	44
3.1.2.2.	Procedimiento	45
3.1.2.3.	Algoritmo y convergencia	46
3.1.3.	Algoritmo DIRECT con restricciones generales.	48
3.1.3.1.	La función auxiliar	48
3.1.3.2.	Procedimiento	49
3.1.3.3.	Los coeficientes c_j	50
3.1.3.4.	Estrategia de desempate en las divisiones	50
3.1.3.5.	La distancia d	51
3.1.3.6.	Algoritmo y convergencia	51
3.1.4.	Algoritmo DIRECT con búsqueda local	53
3.1.4.1.	Algoritmo DIRMIN	54
3.2.	Algoritmo BIRECT irrestricto con búsqueda local del tipo búsqueda de patrones	55
3.2.1.	Introducción	55
3.2.2.	Detalles de implementación de BIRECT con búsqueda de patrones	55
3.2.3.	Análisis y experimentación numérica	56
3.2.4.	Conclusiones y observaciones adicionales	62
3.3.	Algoritmo BIRECT con restricciones generales	63
3.3.1.	Presentación	63
3.3.2.	Los nuevos coeficientes s_j	64
3.3.3.	Selección de hiperrectángulos potencialmente óptimos	66
3.3.3.1.	La cápsula convexa inferior	67
3.3.3.2.	El proceso de selección paso a paso	69
3.3.4.	Convergencia y algoritmo	73
3.3.5.	Resultados numéricos	78
3.3.5.1.	El efecto de la utilización de diferentes tolerancias	80
3.3.5.2.	¿Cuánto impacta la dimensión del problema en la obtención de la solución?	83
3.3.5.3.	Cápsulas convexas y el movimiento hacia la factibilidad	86
4.	Conclusiones	91
A.	Problemas de optimización local con restricciones de cajas	93
B.	Problemas optimización global con restricciones generales	109
C.	Problemas optimización global con restricciones de cotas en las variables	117
	Bibliografía	130

Índice de figuras

1.1. Interpretación gráfica de $\chi(x)$ (Kolda, Lewis y Torczon [25])	6
2.1. Diferentes métodos de búsqueda directa	12
2.2. Posibles ubicaciones de $K(x, \varepsilon)$ y $K^\circ(x, \varepsilon)$ (Kolda, Lewis y Torczon [25])	15
2.3. Comparación de performance profiles de <code>nmps</code> y <code>patternsearch</code>	27
2.4. Comparación de performance profiles de las cinco estrategias consideradas.	32
3.1. Rectas obtenidas al reemplazar l y u en la desigualdad (3.1)	38
3.2. Visualización del procedimiento iterativo del algoritmo de Shubert	39
3.3. Interpretación gráfica de la selección de intervalos potencialmente óptimos	41
3.4. Visualización de los puntos muestreados por BIRECT en 2 dimensiones	45
3.5. Visualización de la función h_r	49
3.6. Visualización de h_r para cada hiperrectángulo de una iteración	49
3.7. Distribución de cantidad de evaluaciones funcionales.	60
3.8. Máximo porcentaje (%) de mejora obtenido en evaluaciones funcionales (EF) con la incorporación de la búsqueda local.	62
3.9. Cantidad de combinaciones de parámetros donde la búsqueda local redujo el número evaluaciones funcionales.	63
3.10. Diferencias en la generación de los nuevos puntos entre DIRECT y BIRECT	65
3.11. Selección de hiperrectángulos potencialmente óptimos - BIRECT	68
3.12. Caso ejemplo: Cuando hiperrectángulo aparece más de una vez en la cápsula convexa	72
3.13. Tiempo en función de las cápsulas convexas calculadas - $\text{tol}_{\text{máx}} = 10^{-4}$	88



Índice de tablas

2.1. Descripción de problemas de la colección Hock–Schittkowski.	23
2.2. Resultados de las comparaciones numéricas	29
2.3. Comparación numérica entre D_{\oplus} y \overline{D}_2 para $n = 2$	34
2.4. Comparación numérica entre D_{\oplus} y \overline{D}_3 para $n = 3$	34
3.1. Descripción de problemas test - BIRECT con búsqueda local.	57
3.2. Esquema de combinaciones experimentales - BIRECT con búsqueda local	58
3.3. Resultados numéricos - BIRECT irrestricto con búsqueda local	61
3.4. Descripción de problemas test - BIRECT con restricciones generales.	79
3.5. Resultados numéricos - BIRECT con restricciones generales	82
3.6. Resultados numéricos - BIRECT con restricciones generales - Dimensiones 2, 3, 4 y 5	84
3.7. Resultados numéricos - BIRECT con restricciones generales - Dimensiones 6, 7 y 8	85
3.8. Resultados cápsula convexa - $\text{tol}_{\text{máx}} = 10^{-4}$	87
3.9. Valor final de los coeficientes c	89



Índice de algoritmos

2.1. Algoritmo compass search	13
2.2. Algoritmo nmps	18
3.1. Procedimiento de división de hiperrectángulos - Algoritmo DIRECT multivariado	42
3.2. Algoritmo DIRECT multivariado	43
3.3. Algoritmo BIRECT irrestricto	47
3.4. DIRECT con restricciones generales	52
3.5. BIRECT con restricciones generales	74
3.6. Identificación de hiperrectángulos potencialmente óptimos	75
3.7. Cápsula convexa	76
3.8. Bisección de hiperrectángulos	77
3.9. Chequeo de factibilidad	78
3.10. Actualización de s	78



Notación

Vectores

x_i Componente i del vector x

Conjuntos

$\{x^k\}$ Sucesión de números reales

\mathcal{P} Conjunto de hiperrectángulos potencialmente óptimos

$|P|$ Cardinalidad del conjunto P

Procesos Iterativos

δ_m Tamaño del hiperrectángulo indexado con m

$\delta_{\mathbb{I}_k}$ Conjunto de tamaños de los hiperrectángulos asociados a la iteración k

$(\delta_{\mathbb{I}_k})_j$ Componente j del conjunto $\delta_{\mathbb{I}_k}$

\mathbb{I}_k Conjunto de índices asociados a \mathcal{P}_k en la iteración k

\mathbb{P}_k Conjunto de índices de hiperrectángulos potencialmente óptimos de la iteración k

\mathcal{I}_k Conjunto de hiperrectángulos de la iteración k

\mathcal{P}_k Conjunto de hiperrectángulos potencialmente óptimos de la iteración k

x^k Vector $x \in \mathbb{R}^n$ de la iteración k

x_j^k Componente j del vector x en la iteración k

Criterios de parada algorítmicos

Δ_{tol} Tolerancia máxima permitida en la longitud de paso de los algoritmos de búsqueda de patrones

$FE_{\text{máx}}$ Cantidad máxima de evaluaciones funcionales permitidas

$K_{\text{máx}}$ Cantidad máxima de iteraciones permitidas

Otros Símbolos

f^* Valor óptimo de un problema de optimización

L Constante de Lipschitz

x^* Punto óptimo de un problema de optimización



Capítulo 1

Fundamentos de optimización

1.1. Optimización sin restricciones

Dentro del campo de la optimización, los problemas más simples de abordar consisten en minimizar (o maximizar) una función de variables reales sin ningún tipo de restricciones sobre las mismas. Estos problemas son generalmente denominados como *problemas de optimización sin restricciones o irrestrictos* y su formulación matemática es

$$\underset{x \in \mathbb{R}^n}{\text{minimizar}} \quad f(x) \tag{1.1}$$

con $f : \mathbb{R}^n \rightarrow \mathbb{R}$ una función no lineal. Lo interesante de los métodos numéricos desarrollados para resolver el problema (1.1) es que muchos de ellos son utilizados como punto de partida para generar otros modelos numéricos que resuelven problemas más complejos (por ejemplo, incluyendo diversas restricciones). A continuación se presentan dos definiciones claves dentro del campo de la optimización irrestricta.

Definición 1.1.1. *El punto x^* es un **minimizador global** de (1.1) si $f(x^*) \leq f(x)$ para todo $x \in \mathbb{R}^n$.*

Definición 1.1.2. *El punto x^* es un **minimizador local** de (1.1) si existe $\delta > 0$ tal que $f(x^*) \leq f(x)$ para todo $x \in \mathbb{R}^n$ con $\|x - x^*\| < \delta$.*

La mayoría de los métodos computacionales y algoritmos de optimización están formulados para obtener minimizadores locales. Una de las razones por la cual esto ocurre es que determinar minimizadores globales es un problema mucho más complejo y, por lo tanto, existen pocos algoritmos determinísticos para encontrarlos. El siguiente teorema establece una condición necesaria para minimizadores locales de funciones suficientemente suaves.

Teorema 1.1.1 (Condición necesaria de optimalidad de primer orden). *Si x^* es un minimizador local del problema (1.1) y f es continuamente diferenciable en un entorno abierto de x^* entonces $\nabla f(x^*) = 0$.*

Demostración. Ver demostración en [41]. □

Dado que todo punto estacionario x es un punto que cumple $\nabla f(x) = 0$, por el teorema anterior se tiene que cualquier minimizador local de (1.1) es un *punto estacionario*. Además, como mencionan Kolda et al. en [25], se dice que un método de optimización tiene **convergencia de primer orden** cuando uno (o algunos o todos) de los puntos límites obtenidos a partir de las iteraciones son puntos estacionarios de la función f .

En esta tesis se omitirá enunciar las condiciones necesarias y suficientes de optimalidad de segundo orden para el problema irrestricto dado que no serán utilizadas en el desarrollo de los siguientes capítulos.

1.1.1. Algoritmos para optimización irrestricta: Estrategias de búsqueda lineal

A lo largo del tiempo, un gran número de algoritmos (iterativos) fueron propuestos para determinar minimizadores locales que resuelvan el problema (1.1). Por la naturaleza de estos problemas cualquier algoritmo para resolverlos necesita ser provisto de un punto inicial, denotado usualmente por x^0 con el cual comenzar las iteraciones. Así, se genera una sucesión $\{x^k\}_{k=0}^{\infty}$ de puntos en los que se espera que el valor funcional de f tienda al valor de un mínimo local. A partir de una iteración general donde x^k es la aproximación actual al minimizador de f , la elección del próximo punto x^{k+1} suele estar basada fundamentalmente en dos tipos de estrategias de globalización¹: *métodos búsqueda de lineal y métodos de región de confianza*. En esta tesis utilizaremos el primer enfoque para trabajar en los nuevos resultados.

La estrategia de búsqueda lineal involucra tres elementos:

- Un iterado actual x^k .
- Una dirección de descenso p^k .
- Una longitud de paso α^k .

Definición 1.1.3. *Un vector $d \in \mathbb{R}^n$ es una **dirección de descenso** de una función $f : \mathbb{R}^n \rightarrow \mathbb{R}$ en el punto x si y sólo si*

$$-\nabla f(x)^T d > 0.$$

De esta manera, combinando la definición anterior con la siguiente igualdad obtenida a partir del Teorema de Taylor

$$f(x + \alpha d) = f(x) + \alpha \nabla f(x)^T d + o(\alpha),$$

se tiene que, para α suficientemente pequeño, el vector d es una dirección de descenso si y sólo si el punto $x + \alpha d$ reduce el valor funcional de f en x .

Así, la idea detrás de un algoritmo de búsqueda lineal es elegir una dirección de descenso p^k a partir del iterado actual x^k y moverse una distancia α^k en esa dirección con el objetivo de alcanzar un nuevo punto, usualmente denotado por x^{k+1} . En general, este nuevo punto se caracteriza por tener un valor funcional menor. Matemáticamente hablando la nueva estimación a la solución es definida por

$$x^{k+1} = x^k + \alpha^k p^k$$

donde el escalar α^k es elegido tal que

$$f(x^{k+1}) < f(x^k). \tag{1.2}$$

En pocas palabras, si p^k es una dirección de descenso, se tiene que la desigualdad $f(x^k + \alpha^k p^k) < f(x^k)$ se cumple al menos para valores pequeños y positivos de α . Por tal motivo es posible asumir que $\alpha > 0$ y, el problema de encontrar una nueva mejor aproximación al mínimo de f se reduce a

$$\underset{\alpha > 0}{\text{minimizar}} \quad f(x^k + \alpha p^k), \tag{1.3}$$

siempre que p^k sea una dirección de descenso en cada iteración k .

¹Por *estrategia de globalización* entendemos el procedimiento llevado a cabo para generar una nueva aproximación a la solución de manera de garantizar *convergencia global*, es decir, convergencia a un punto estacionario (también denominado convergencia de primer orden) partiendo de un punto inicial arbitrario.

Por un lado, si bien a simple vista la resolución del problema unidimensional (1.3) resulta atractiva, es bastante costosa computacionalmente y usualmente injustificada. En lugar de esto, se pueden generar aproximaciones, en una cantidad finita, de valores de α hasta lograr un acercamiento aceptable al mínimo de (1.3). A partir de este nuevo punto se define un nuevo problema, similar a (1.3), y se repite el proceso anterior. De esta manera, se genera un proceso iterativo que termina minimizando el valor de f . En el mejor de los casos, en el límite de las iteraciones, la convergencia al valor óptimo f^* de f es alcanzada.

Por otro lado, para que el proceso iterativo mencionado en el párrafo anterior sea eficiente, es decir, la función sea minimizada en cada paso, resulta importante asegurar que cada iteración tenga una dirección de descenso asociada. En el campo de la optimización basada en derivadas, la dirección de máximo descenso o la dirección de Newton son algunas de las alternativas existentes como direcciones de descenso. En el caso de los métodos sin derivadas, como se profundizará en el Capítulo 2, el requisito es que el conjunto de direcciones elegidas a lo largo de las iteraciones sea un conjunto generador positivo de \mathbb{R}^n .

La convergencia del procedimiento descripto, incluso reforzando los dos puntos anteriores, no resulta tan directa, pues existen casos particulares donde la misma es imposible de alcanzar (para mayor detalle ver [18], [25] y [41]). Sin embargo, agregando hipótesis adicionales sobre la dirección de descenso p^k y la longitud de paso α^k es posible establecer el siguiente resultado de convergencia global de primer orden para los métodos de búsqueda lineal.

Teorema 1.1.2 (Convergencia global de métodos de búsqueda lineal.). *Sea $f : \mathbb{R}^n \rightarrow \mathbb{R}$ una función continuamente diferenciable en \mathbb{R}^n y acotada inferiormente. Sea x^0 un punto inicial dado y definamos la sucesión $\{x^k\}$ como $x^{k+1} = x^k + \alpha^k p^k$, donde p^k son direcciones de búsqueda y $\alpha^k \geq 0$ longitudes de paso. Asumimos además que*

(i) *El conjunto $S = \{x : f(x) \leq f(x^0)\}$ es acotado.*

(ii) *∇f es una función Lipschitz continua para todo x , es decir,*

$$\|\nabla f(x) - \nabla f(y)\| \leq L\|x - y\|,$$

para alguna constante $0 < L < \infty$.

(iii) *Las direcciones p^k satisfacen la condición de descenso suficiente*

$$-\frac{(p^k)^T \nabla f(x^k)}{\|p^k\| \|\nabla f(x^k)\|} \geq \varepsilon > 0.$$

(iv) *Las direcciones son gradiente relacionadas:*

$$\|p^k\| \geq m \|\nabla f(x^k)\| \text{ para todo } k \text{ (con } m > 0)$$

y acotadas en norma, es decir existe $M > 0$ tal que

$$\|p^k\| \leq M \text{ para todo } k.$$

(v) *La longitud de paso α^k es elegida como el primer elemento de la sucesión $\left\{1, \frac{1}{2}, \frac{1}{4}, \dots\right\}$ que satisface la condición de descenso suficiente*

$$f(x^k + \alpha^k p^k) \leq f(x^k) + \mu \alpha^k (p^k)^T \nabla f(x^k) \text{ donde } 0 < \mu < 1.$$

Entonces

$$\lim_{k \rightarrow \infty} \|\nabla f(x^k)\| = 0.$$

Demostración. Ver demostración en [18]. □

De esta manera, el teorema nos asegura que cualquier punto límite de la sucesión $\{x^k\}$ es un punto estacionario.

1.2. Optimización con restricciones

Consideremos ahora el problema general de minimizar una función de valores reales con restricciones en las variables, es decir

$$\begin{aligned} &\underset{x \in \mathbb{R}^n}{\text{minimizar}} && f(x) \\ &\text{sujeto a} && h(x) = 0 \\ &&& c(x) \leq 0, \end{aligned} \tag{1.4}$$

donde $f : \mathbb{R}^n \rightarrow \mathbb{R}$, $h : \mathbb{R}^n \rightarrow \mathbb{R}^m$ y $c : \mathbb{R}^n \rightarrow \mathbb{R}^p$ son funciones continuamente diferenciables. Problemas del tipo (1.4) son denominados **problemas con restricciones generales**. El conjunto de puntos que satisface las restricciones en (1.4) se denomina **conjunto factible** y se denota como

$$\Omega = \{x : h(x) = 0 \text{ y } c(x) \leq 0\}, \tag{1.5}$$

además cuando $x \in \Omega$ se dice que x es un **punto factible**. Notemos que en presencia de un problema de optimización sin restricciones como (1.1) se tiene que $\Omega = \mathbb{R}^n$. Martínez y Santos en [37] presentan de manera clara y resumida cómo, de acuerdo a las características del conjunto Ω , podemos clasificar los problemas de optimización en diferentes tipos. En particular a lo largo de esta tesis nos limitaremos a dos tipos de problemas:

1. Problemas con restricciones de cajas o cotas en las variables:

$$\Omega = \{x \in \mathbb{R}^n : l \leq x \leq u\}.$$

2. Problemas con restricciones generales (no lineales):

$$\Omega = \{x \in \mathbb{R}^n : h(x) = 0 \text{ con } h : \mathbb{R}^n \rightarrow \mathbb{R}^m \text{ y } c(x) \leq 0 \text{ con } c : \mathbb{R}^n \rightarrow \mathbb{R}^p\}.$$

1.2.1. Definiciones y resultados importantes

De forma análoga a la Sección 1.1, se tiene la siguiente definición:

Definición 1.2.1. *El punto x^* es un **minimizador local** de (1.4) si existe $\delta > 0$ tal que $f(x^*) \leq f(x)$ para todo $x \in \Omega$ con $\|x - x^*\| < \delta$.*

Además, para probar resultados de convergencia a un mínimo local del problema (1.4), se requieren adicionalmente las siguientes definiciones.

Definición 1.2.2. *Una restricción i se dice **activa** en un punto factible x si $c_i(x) = 0$.*

Definición 1.2.3. *Dado un punto x , el vector v es un **vector tangente** a Ω en x existe una sucesión de puntos factibles $\{x^k\}$ que converge a x y una sucesión de escalares α^k con $\alpha^k \rightarrow 0$ tal que*

$$\lim_{k \rightarrow \infty} \frac{x^k - x}{\alpha^k} = v.$$

Definición 1.2.4. Dado un punto x , el conjunto de todos los vectores tangentes a Ω en x se denomina **cono tangente** y se denota por $T_\Omega(x)$.

Definición 1.2.5. Un punto $x \in \Omega$ es un **punto regular** si el conjunto de los gradientes de las restricciones activas es linealmente independiente.

A continuación se presenta uno de los resultados más importantes de la teoría de optimización con restricciones que establece **condiciones necesarias de optimalidad de primer orden**. Las mismas son generalmente conocidas como condiciones KKT.

Teorema 1.2.1 (Condiciones KKT). Sea x^* un minimizador local regular del problema (1.4) y sea $\mathcal{I} = \{i : c_i(x^*) = 0 \text{ con } i = 1, 2, \dots, p\}$. Si el conjunto $\{\nabla h_1(x^*), \dots, \nabla h_m(x^*)\} \cup \{\nabla c_i(x^*), i \in \mathcal{I}\}$ es linealmente independiente entonces existen únicos $\lambda_i \in \mathbb{R}$ con $i = 1, 2, \dots, m$ y $\mu_i \geq 0$ para todo $i \in \mathcal{I}$ tal que

$$\nabla f(x^*) + \sum_{i=1}^m \lambda_i \nabla h_i(x^*) + \sum_{i \in \mathcal{I}} \mu_i \nabla c_i(x^*) = 0$$

Demostración. Ver demostración en [41]. □

De esta manera, como mencionan Martínez y Santos en [37], si x es un punto regular y minimizador local del problema (1.4), definiendo $\mu_i = 0$ cuando $i \notin \mathcal{I}$, las condiciones KKT se pueden reescribir de la siguiente manera:

$$\nabla f(x) + \sum_{i=1}^m \lambda_i \nabla h_i(x) + \sum_{i=1}^p \mu_i \nabla c_i(x) = 0, \quad (1.6)$$

$$h(x) = 0, \quad (1.7)$$

$$\mu_i c_i(x) = 0, \text{ para todo } i = 1, 2, \dots, p, \quad (1.8)$$

$$\mu_i \geq 0, \text{ para todo } i = 1, 2, \dots, p, \quad (1.9)$$

$$c_i(x) \leq 0 \text{ para todo } i = 1, 2, \dots, p. \quad (1.10)$$

De esta manera, las $n + m + p$ ecuaciones (1.6)–(1.8) forman un sistema de ecuaciones no lineales con incógnitas $x \in \mathbb{R}^n$, $\lambda \in \mathbb{R}^m$ y $\mu \in \mathbb{R}^p$. Las soluciones de este sistema que satisfacen (1.9) y (1.10) son precisamente los puntos estacionarios del problema (1.4).

Si bien el Teorema 1.2.1 es muy útil, requiere un conocimiento explícito del conjunto factible ya que se basa fuertemente en la descripción algebraica de las restricciones del problema (1.4). Frente a este “inconveniente” existe un resultado alternativo que depende solamente de la geometría de la región factible Ω y que, por lo tanto, no precisa conocer o calcular las derivadas de las restricciones. Para presentar esta condición “geométrica” de optimalidad de primer orden es preciso definir primero el cono normal a un conjunto.

Definición 1.2.6. El **cono normal** $N_\Omega(x)$ al conjunto Ω en el punto $x \in \Omega$ se define como

$$N_\Omega(x) = \{v : v^T w \leq 0 \text{ para todo } w \in T_\Omega(x)\},$$

donde $T_\Omega(x)$ es el cono tangente de la Definición 1.2.4.

Los vectores $v \in N_\Omega(x)$ se denominan de esta manera vectores normales. Es fácil notar que cada vector normal v es al menos perpendicular a cada vector tangente. A continuación se presenta una condición necesaria de optimalidad de primer orden del problema (1.4) que utiliza el cono normal $N_\Omega(x)$.

Teorema 1.2.2. Si x^* es un minimizador local de f en Ω entonces

$$-\nabla f(x^*) \in N_{\Omega}(x^*).$$

Demostración. Ver demostración en [41]. □

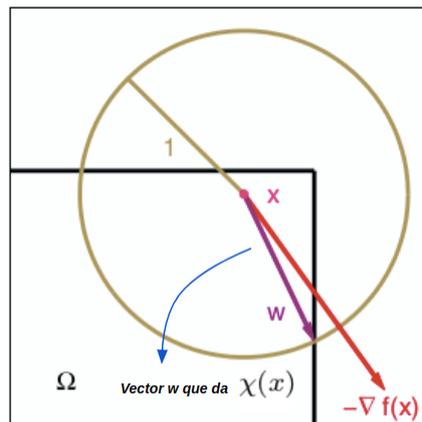
Kolda, et al. en [25] observan que si bien el resultado del Teorema 1.2.2 resulta muy simple y atractivo, su gran inconveniente es que no es útil como una medida continua de estacionariedad para problemas con restricciones puesto que la definición de cono normal no depende de la continuidad de f en la frontera Ω . Para sortear esta dificultad existe otra medida de estacionariedad muy usada dentro de la programación no lineal. La misma fue definida por Conn, Gould y Toint [8] y está dada por

$$\chi(x) = \max_{\substack{x+\omega \in \Omega, \\ \|\omega\| \leq 1}} -\nabla f(x)^T \omega. \quad (1.11)$$

La medida (1.11) permite obtener el grado en el cual la dirección de máximo descenso se encuentra apuntando hacia afuera con respecto a la fracción de la región factible cerca de x . Esta descripción conceptual fue ilustrada en [25] a través de la imagen que se muestra en la Figura 1.1. Por otra parte, en [8] los autores probaron que si Ω es convexo, la función χ satisface las siguientes propiedades:

1. χ es una función continua.
2. $\chi \geq 0$.
3. $\chi(x) = 0$ si y sólo si x es un punto KKT del problema (1.4).

Figura 1.1: Interpretación gráfica de $\chi(x)$
(Kolda, Lewis y Torczon [25])



Lo valioso de las propiedades anteriores es que, para la sucesión de iterados x^k del problema (1.4), mostrando $\chi(x^k) \rightarrow 0$ cuando $k \rightarrow \infty$, se establece un resultado de convergencia global de primer orden el cual resulta equivalente al resultado que involucra ∇f . Esta medida y sus propiedades derivadas serán clave en el desarrollo del próximo capítulo.

1.3. Optimización global

Como se mencionó en las secciones anteriores, encontrar el minimizador global de un problema de optimización es una tarea compleja. La mayoría de los resultados de optimalidad suponen que la función es derivable e incluso, dentro de la optimización basada en derivadas, obtener los gradientes y hessianas es un paso necesario. Estos valores permiten “predecir” el comportamiento de la función en un entorno local. Por lo tanto, a medida que nos alejamos de la ubicación actual, esta “predicción” se vuelve menos precisa. Esta es una de las principales razones por las cuales la información disponible nos permite encontrar solo minimizadores locales. En consecuencia no existen, hasta el momento, resultados similares a los teoremas mencionados anteriormente para minimizadores globales. Es oportuno recordar en este punto, dado que es un resultado ampliamente conocido, que cuando el problema es convexo todo mínimo local es global.

Encontrar el mínimo global de una función requiere, generalmente, un conocimiento profundo de la función en cada punto de la región factible. Sin embargo, en problemas de la vida real es muy poco probable disponer de este nivel de información, principalmente teniendo en cuenta que la mayoría de los datos son experimentales o discretos. Esta es otra de las razones por las cuales los métodos que encuentran minimizadores locales suelen ser los más requeridos en la práctica.

Para reducir la complejidad de un problema de optimización global, un enfoque tradicional es dividir el problema en una sucesión de diferentes subproblemas de optimización local. No obstante, en los últimos años, dentro de la comunidad de optimización, ha resurgido un interés por desarrollar nuevas e innovadoras técnicas que permitan resolver estas clases de problemas. Algunos ejemplos de estas nuevas líneas de investigación se encuentran en las publicaciones de Horst, Panos y Van Thoai [22] del 2000, Hansen y Walster [19] de 2003, Liberti y Maculan [34] de 2006 y Jones [23] de 2001. Particularmente, en esta tesis retomaremos los trabajos de Jones, Perttunen, y Stuckman [24] y Jones [23] (este último publicado en la recopilación realizada por Floudas y Pardalos en [13]) para presentar una nueva propuesta dentro de los métodos de optimización global. Este tema se retomará en mayor profundidad en el Capítulo 3.

Capítulo 2

Algoritmos de optimización local sin derivadas para problemas con restricciones de cotas en las variables

2.1. Introducción

En la actualidad existen muchos problemas de optimización de la vida real donde la función objetivo contiene ruido, las evaluaciones funcionales requieren simulaciones complejas o bien directamente no existe una formulación explícita de la función a optimizar (los comúnmente conocidos como problemas de *caja negra*). Así, por ejemplo, los métodos Quasi-Newton o de diferencias finitas no pueden ser aplicados en estos casos. Esta clase de situaciones pueden originarse en aplicaciones de geometría molecular ([1], [38]), registración de imágenes médicas [42], radiobiología [44] y problemas de optimización de forma y diseño [29].

Los métodos sin derivadas (en inglés conocidos como *derivative-free methods*) han recibido considerable atención por parte de la comunidad de optimización durante las últimas décadas, incluyendo el establecimiento de sólidos fundamentos matemáticos para muchas de las estrategias que se utilizan muy a menudo hoy en la práctica. En particular, los métodos de búsqueda de patrones han tenido éxito donde enfoques más elaborados reportaron fallas. Los mismos pertenecen a la familia de los métodos de búsqueda directa, caracterizados por implementaciones no muy sofisticadas, la ausencia de la construcción de un modelo de la función objetivo y el uso de un conjunto de direcciones de búsqueda para explorar el espacio alrededor del iterado actual. El desarrollo y estudio de los mismos puede encontrarse por ejemplo en [4], [5], [9], [10], [16] y [47].

Los métodos de búsqueda directa fueron inicialmente introducidos por Hooke y Jeeves [21] para problemas de optimización sin restricciones, y más tarde analizados y formalmente presentados por Torczon [47] y Kolda et al. [25]. Recientemente, se incorporaron herramientas usualmente presentes en los métodos que involucran derivadas a los métodos de búsqueda de patrones. En este sentido, al igual que las técnicas de búsqueda lineal basadas en derivadas, los métodos de búsqueda directa han sido dotados con un sólido fundamento matemático. Por ejemplo, Diniz-Ehrhardt, Martínez y Raydan [11] introdujeron una estrategia global basada en las ideas mencionadas en [17], [28], [33] y [36]. En este trabajo, los autores usaron un esquema de búsqueda lineal no monótona dentro de un algoritmo de búsqueda de patrones para el caso de optimización sin restricciones. Lewis y Torczon en [31], por su parte, extendieron el método de búsqueda de patrones para el caso con restricciones de cotas en las variables desde un punto de vista mayormente teórico ya que no reportaron estudios numéricos en ese trabajo. Por otro lado, este problema también fue abordado por Arouxét, Echebest y Pilotta en [3] usando interpolación

polinomial y estrategias de región de confianza, un enfoque bastante diferente a los métodos de búsqueda de patrones.

En esta tesis, y en particular en este capítulo, proponemos y presentamos un nuevo método de búsqueda de patrones que incluye una búsqueda lineal no monótona como estrategia de globalización para el caso de optimización sin derivadas y con restricciones de cotas en las variables. La propuesta mencionada forma parte de los resultados presentados por Frau y Pilotta en [14]. Desde un punto de vista matemático el problema a resolver consiste en

$$\underset{x \in \Omega}{\text{minimizar}} \quad f(x), \tag{2.1}$$

donde $f : \mathbb{R}^n \rightarrow \mathbb{R}$ y $\Omega = \{x \in \mathbb{R}^n : l \leq x \leq u\}$ con $-\infty \leq l \leq u \leq \infty$.

Es importante remarcar que si bien en estos casos, y específicamente con fines teóricos como veremos más adelante, se asume que la función objetivo es continuamente diferenciable en Ω , en la práctica se supone que la información de sus derivadas no existe o no es confiable y por lo tanto no es una herramienta que pueda ser utilizada.

El método propuesto es un caso particular de los métodos de búsqueda de patrones donde el conjunto de direcciones de búsqueda son las direcciones coordenadas. El nuevo algoritmo está basado en las ideas introducidas en [11] por Diniz-Ehrhardt et al. Sin embargo, mientras que el algoritmo de Diniz-Ehrhardt et al. fue desarrollado para el caso irrestricto, el nuevo algoritmo es capaz de manejar problemas con restricciones de cajas. Por otra parte, las demostraciones del principal resultado de convergencia del nuevo método difieren bastante del enfoque utilizado en [11]. Para probar convergencia global, se utilizó la medida de estacionariedad χ definida por Coon et al. en [8] y presentada en el Capítulo 1 de esta tesis.

Desde el punto de vista numérico, el segundo aporte de este capítulo consiste en un extenso estudio numérico de diferentes estrategias de búsqueda lineal no monótona. El mismo se encuentra dividido en dos secciones diferenciadas. En primer lugar, con el objetivo de validar el nuevo algoritmo propuesto, se realizaron experimentos numéricos comparando el mismo con otro algoritmo de búsqueda de patrones bastante utilizado en el campo de la optimización sin derivadas. En segundo lugar, y en base a los resultados obtenidos del análisis anterior, se extendió el estudio numérico incorporando otras estrategias de búsqueda lineal ([7], [40], [48], [51] y [52]). Lo interesante e innovador de esta incorporación es el hecho de que las mismas fueron estrategias inicialmente propuestas para resolver problemas diferentes a (2.1). Entre ellos se encuentran problemas de minimización sin restricciones y sistemas de ecuaciones no lineales abordados tanto con enfoques con derivadas como sin ellas.

2.2. Preliminares: Métodos de búsqueda directa para optimización local

Los métodos de búsqueda directa surgieron, hasta donde se conoce, aproximadamente en los años 1950. Sin embargo, en parte dado que inicialmente estaban basados en heurísticas, y por lo tanto carecían de pruebas de convergencia, y en parte porque se los consideraba lentos a nivel de implementación, fueron menospreciados por la comunidad matemática. A pesar de esto, otras comunidades científicas, por ejemplo relacionadas con el ámbito ingenieril, los siguieron considerando en sus trabajos debido a su facilidad de implementación pero principalmente porque se evitaban calcular derivadas complejas. En esos momentos, el software disponible no era tan preciso como actualmente estamos acostumbrados y los errores en el cálculo de las mismas podían llegar a ser extremadamente grandes.

En el trabajo de Hooke y Jeeves [21] en 1961 se encuentra lo más cercano a una definición de lo que implica un método de búsqueda directa. Sin embargo, recién en 1991 gracias a la publicación

de Virginia Torczon [46] y todas las contribuciones que ella y su equipo comenzaron a realizar (veáse por ejemplo [25], [31] y [47]), estos métodos recuperaron el interés principalmente de los matemáticos de la época.

La definición que dan Hooke y Jeeves en [21] se encuentra lejos de lo que conocemos por definición desde el punto de vista formal matemático. Sin embargo, los autores fueron capaces de poner en palabras simples el proceso algorítmico que define un método de búsqueda directa. En forma resumida, un *método de búsqueda directa* se desarrolla en base a comparaciones funcionales, en pos de llevar a cabo un proceso de optimización, sobre diferentes puntos de prueba pertenecientes a la región factible. A su vez, estos métodos se caracterizan por permitir libertad en la elección de la mejor aproximación y de los puntos de prueba además de evitar, en lo posible, utilizar estrategias basadas en derivadas desde el punto de vista clásico.

Un aspecto notable y diferenciable de estos métodos fue remarcado por Wright [50] en 1996 y posteriormente recuperado por Kolda et al. en 2003 [25] y es la observación de que el objetivo de este tipo de estrategias está lejos de querer aproximar gradientes. De esta manera la metodología de trabajo se aleja bastante de los métodos Quasi-Newton o diferencias finitas.

Los métodos de búsqueda de patrones constituyen una subclase de los métodos de búsqueda directa y en consecuencia son métodos que no requieren, y tampoco buscan, calcular o aproximar derivadas. Específicamente, se distinguen por usar un patrón de puntos para realizar la búsqueda, el cual es independiente de la función objetivo. Es de ahí de donde proviene su nombre.

En 1997, Virginia Torczon [47] presentó la primera generalización formal de estos métodos junto con una robusta teoría de convergencia global para el caso de optimización irrestricta. Este desarrollo luego fue extendido a la resolución de problemas de optimización con restricciones de cotas y con restricciones lineales en [31] y [26, 32] respectivamente. Estos trabajos otorgaron la solidez matemática que tanto había sido criticada y fueron tan importantes que además de ser el puntapié de una larga lista de publicaciones posteriores, aún hoy, más de 20 años más tarde, continúan siendo citados.

Asimismo, la recopilación que hicieron Kolda, Lewis y Torczon en [25] acerca de los métodos de búsqueda directa permitió esclarecer algunos puntos en relación a los métodos de búsqueda de patrones y a los métodos de búsqueda directa en general. Con respecto a esto, los autores también definieron un nuevo conjunto de métodos: los *métodos GSS* (por su derivación del inglés, *generating set search*). Los mismos representan un intermedio entre los métodos de búsqueda directa y los métodos de búsqueda de patrones. En particular, engloban aquel subconjunto de métodos de búsqueda directa con resultados de convergencia tan fuertes como los correspondientes a los métodos de búsqueda lineal basados en derivadas (ver Figura 2.1). De acuerdo a los autores la introducción de este concepto fue necesaria debido a que no todos los métodos de búsqueda directa son capaces de encontrar soluciones confiables. Por ejemplo el algoritmo simplex de Nelder-Mead tiene la capacidad de encontrar soluciones de manera muy eficiente pero al mismo tiempo puede fallar en situaciones no esperadas. Inversamente, el algoritmo *compass search* o la búsqueda de patrones de Hooke y Jeeves demostraron ser bastante confiables aunque no tan rápidos en la práctica.

La lista de los métodos GSS es bastante larga cubriendo diferentes formas de abordar los métodos de búsqueda de patrones, entre ellas encontramos la propuesta de Torczon [47], el algoritmo de Hooke y Jeeves [21], la búsqueda multidireccional también presentada por Torczon [46], entre otros. A su vez, algunos métodos sin derivadas y basados en búsquedas lineales también son categorizados dentro de los métodos GSS. Entre ellos se pueden mencionar por ejemplo los trabajos de Yu [49], Lucidi y Sciandrone [36], García-Palomares y Rodríguez [15]. Como veremos en breve, la categorización de los métodos GSS no solo contribuyó a formalizar una variedad de resultados que venían siendo estudiados, casi aisladamente, sino que también abrió un abanico de propiedades algorítmicas y matemáticas que enriqueció todo el campo de

la optimización sin derivadas.

Figura 2.1: *Diferentes métodos de búsqueda directa*



2.2.1. El algoritmo compass search: Caso irrestricto

El ejemplo más simple de un método de búsqueda de patrones para resolver el problema

$$\underset{x \in \mathbb{R}^n}{\text{minimizar}} \quad f(x) \tag{2.2}$$

con $f : \mathbb{R}^n \rightarrow \mathbb{R}$, es el algoritmo conocido como **compass search**. En este algoritmo, el patrón de direcciones de búsqueda está dado por las direcciones coordenadas. Es decir, dado un punto inicial $x^0 \in \mathbb{R}^n$, las posibles direcciones a partir de las cuales se generan las nuevas iteraciones $x^k \in \mathbb{R}^n$ está dado por el siguiente conjunto

$$\mathcal{D}_{\oplus} = \{e_1, e_2, \dots, e_n, -e_1, -e_2, \dots, -e_n\}.$$

donde e_i denota el i -ésimo vector canónico en \mathbb{R}^n .

En el algoritmo compass search cada nuevo iterado x^{k+1} se obtiene través de la expresión $x^k + \Delta^k d$ donde $d \in \mathcal{D}_{\oplus}$ y Δ^k representa el parámetro de longitud de paso. El paso a paso del algoritmo para el caso irrestricto (2.2) se encuentra descrito en el Algoritmo 2.1.

Algunas observaciones importantes:

1. Los puntos $x^k + \Delta^k d^k$ son usualmente denominados *puntos de prueba*.
2. Un resultado muy importante dentro de los métodos GSS para optimización irrestricta es que si el conjunto de direcciones de búsqueda es un generador positivo¹ de \mathbb{R}^n entonces al menos una de esas direcciones es de descenso. De esta manera, dado que el conjunto \mathcal{D}_{\oplus} genera positivamente a todo \mathbb{R}^n , queda asegurado que en cada iteración del algoritmo compass search existirá al menos una dirección de descenso. La justificación matemática en detalle de este resultado puede encontrarse en [25].
3. Existen varias alternativas válidas para investigar si existen una dirección d que satisfaga la condición de descenso simple

$$f(x^k + \Delta^k d^k) < f(x^k) \tag{2.3}$$

del Algoritmo 2.1, por ejemplo evaluar la función en todas las direcciones disponibles (paralela o secuencialmente) y elegir la dirección con la mayor minimización (en caso que exista) o bien realizar la evaluación de forma secuencial y parar el proceso en la primera dirección de descenso encontrada serían algunas opciones.

¹Un conjunto \mathcal{G} genera positivamente \mathbb{R}^n si y sólo si todo vector de \mathbb{R}^n puede escribirse como una combinación lineal positiva de los vectores de \mathcal{G} .

4. Cuando existe una dirección $d \in \mathcal{D}_\oplus$ tal que la desigualdad (2.3) se cumple, se dice que la iteración k fue *exitosa*. La sucesión de iteraciones exitosas es denotada por \mathcal{S} . Inversamente, cuando la desigualdad no se cumple en ninguna dirección, es decir cuando no se logró minimizar el valor actual de f , se habla de iteraciones *no exitosas* y la sucesión de iteraciones correspondientes se denota por \mathcal{U} .
5. Como se observará en breve en el resultado de convergencia del algoritmo, las iteraciones no exitosas juegan un rol fundamental en la prueba del mismo. La razón es bastante simple: estas iteraciones se encuentran representando el *peor escenario* posible, es decir, dónde no fue posible obtener un descenso en el valor funcional. Dado que la convergencia en el límite de las iteraciones exitosas es bastante intuitiva, lo más importante termina siendo garantizar convergencia en el límite de aquellas iteraciones no exitosas.

Algoritmo 2.1: Algoritmo compass search

Entrada: $f : \mathbb{R}^n \rightarrow \mathbb{R}$ Función a minimizar. $x^0 \in \mathbb{R}^n$: Aproximación inicial a la solución. $\Delta_{\text{tol}} > 0$: Tolerancia usada para el test de convergencia. Δ^0 : Valor inicial del parámetro longitud de paso. $K_{\text{máx}}$: Cantidad máxima de iteraciones del algoritmo.**Salida:** $f_{\text{mín}}$: Valor funcional mínimo encontrado. $x_{\text{mín}}$: Punto donde se da el valor funcional mínimo.

/* Inicialización

*/

1 $\mathcal{D}_\oplus \leftarrow \{\pm e_i : i = 1, 2, \dots, n\}$.2 $f_{\text{mín}} \leftarrow f(x^0)$ 3 $x_{\text{mín}} \leftarrow x^0$ 4 $k \leftarrow 1$

/* Proceso iterativo

*/

5 **mientras** $k < K_{\text{máx}}$ **hacer**6 **si** $f(x^k + \Delta^k d^k) < f(x^k)$ **para alguna dirección** $d \in \mathcal{D}_\oplus$ **entonces**7 $x^{k+1} \leftarrow x^k + \Delta^k d^k$ 8 $\Delta^{k+1} \leftarrow \Delta^k$ 9 **en otro caso**10 $x^{k+1} \leftarrow x^k$ 11 $\Delta^{k+1} \leftarrow \frac{\Delta^k}{2}$ 12 **si** $\Delta^{k+1} < \Delta_{\text{tol}}$ **entonces**13 $f_{\text{mín}} \leftarrow f(x^{k+1})$ 14 $x_{\text{mín}} \leftarrow x^{k+1}$

15 Terminar la ejecución del algoritmo.

16 $k \leftarrow k + 1$ 17 $f_{\text{mín}} \leftarrow f(x^{k+1})$ 18 $x_{\text{mín}} \leftarrow x^{k+1}$ 19 **devolver** $f_{\text{mín}}, x_{\text{mín}}$

 A continuación los resultados de convergencia del algoritmo presentados en [25].

Teorema 2.2.1. Si $f : \mathbb{R}^n \rightarrow \mathbb{R}$ es continuamente diferenciable y ∇f es Lipschitz continua con constante L entonces

$$\|\nabla f(x^k)\| \leq \sqrt{n}L\Delta^k$$

para todo $k \in \mathcal{U}$.

Demostración. Ver demostración en [25]. □

Dado que es posible demostrar que

$$\lim_{\substack{k \rightarrow \infty, \\ k \in \mathcal{U}}} \Delta^k = 0,$$

por más detalles véase [25], se deduce inmediatamente que

$$\lim_{\substack{k \rightarrow \infty, \\ k \in \mathcal{U}}} \|\nabla f(x^k)\| = 0,$$

estableciendo un resultado análogo al Teorema 1.1.2 para los métodos de búsqueda lineal basados en derivadas.

2.2.2. El algoritmo compass search: Caso con restricciones de cajas

Consideremos ahora el problema de optimización con restricciones en las variables dado por

$$\underset{x \in \Omega}{\text{minimizar}} \quad f(x), \tag{2.4}$$

donde $f : \mathbb{R}^n \rightarrow \mathbb{R}$ y $\Omega = \{x \in \mathbb{R}^n : l \leq x \leq u\}$ con $-\infty \leq l \leq u \leq \infty$.

Haciendo una analogía con el apartado anterior, se observa que la diferencia sustancial entre el problema (2.2) y (2.4) es que en el segundo cada uno de los iterados x^k debe ser factible, es decir $x^k \in \Omega$ para todo k (incluyendo al punto inicial x^0). Esto es lo que usualmente se denomina en la bibliografía de optimización como *métodos de puntos factibles*.

El gran desafío que se plantea frente a esta situación es que dos condiciones se cumplan simultáneamente:

1. Que las direcciones elegidas para realizar los movimientos hacia los nuevos iterados sean direcciones de descenso.
2. Que los nuevos iterados sean puntos factibles.

Por un lado, algo bueno es que al estar minimizando en cajas, la noción geométrica de la región factible Ω es bastante intuitiva: en \mathbb{R}^2 sería (aproximadamente) un rectángulo, en \mathbb{R}^3 (aproximadamente) un paralelepípedo, y así sucesivamente. Por otro lado, una observación importante es que las situaciones *críticas o peligrosas* se darán cuando los nuevos puntos estén lo suficientemente cerca de la frontera de Ω , es decir $x^{k+1} \notin \text{int}(\Omega)$, puesto que cualquier punto en el interior de Ω siempre será factible. De esta manera, uno de los focos principales para analizar la convergencia del algoritmo *compass search* para problemas del tipo (2.4) consiste en estudiar lo que sucede en la frontera de Ω . En la sección 8.1.2 de [25] los autores realizan el siguiente comentario:

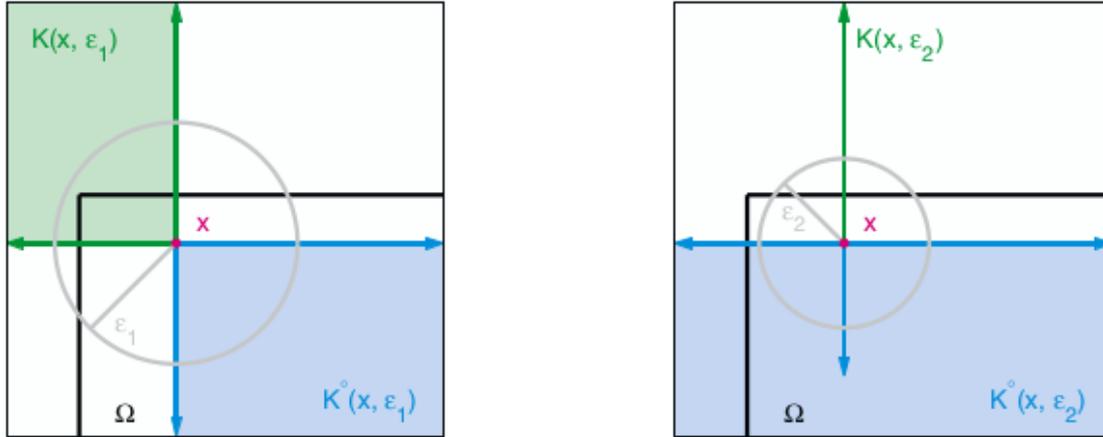
“Un primer paso para mostrar que compass search produce al menos una sucesión de iterados que converge a un punto KKT de (2.4) es mostrar que si el iterado x^k no es un punto KKT, entonces al menos una de las direcciones de búsqueda es una dirección de descenso a lo largo de la cual la búsqueda puede progresar una distancia razonable y mantenerse factible”

En base a lo mencionado hasta el momento, dado $x \in \Omega$, se define $K(x, \varepsilon)$ como el cono generado por 0 y los vectores normales apuntando hacia afuera de las restricciones a una distancia ε de x , es decir

$$\{e_i : u_i - x_i \leq \varepsilon\} \cup \{-e_i : x_i - l_i \leq \varepsilon\}.$$

Como mencionan Kolda, Lewis y Torczon en [25], eligiendo una distancia ε apropiada, el cono polar $K^\circ(x, \varepsilon)$ asociado al cono $K(x, \varepsilon)$ aproxima la región factible cerca de x . Para ilustrar este hecho, en [25] los autores presentan la siguiente figura

Figura 2.2: Posibles ubicaciones de $K(x, \varepsilon)$ y $K^\circ(x, \varepsilon)$
(Kolda, Lewis y Torczon [25])



Los resultados que acompañan la convergencia del algoritmo compass search para el problema (2.4) se listan a continuación.

Proposición 2.2.2. Sea $\mathcal{G} = \mathcal{D}_\oplus$. Sean además $x \in \Omega$, $\varepsilon \geq 0$, $K^\circ = K^\circ(x, \varepsilon)$ y $K = K(x, \varepsilon)$ definidos para el problema (2.4). Si $\mathcal{G}_{K^\circ} \subseteq \mathcal{G}$ es el conjunto de generadores de K° y $[-\nabla f(x)]_{K^\circ} \neq 0$ entonces existe $d \in \mathcal{G}_{K^\circ}$ tal que

$$\frac{1}{\sqrt{n}} \| [-\nabla f(x)]_{K^\circ} \| \leq -\nabla f(x)^T d.$$

Demostración. Ver demostración en [25]. □

Coloquialmente, la proposición anterior plantea que si la proyección de $-\nabla f$ sobre el cono K° es diferente de 0, entonces al menos uno de los generadores de $K^\circ(x, \varepsilon)$ es una dirección de descenso (recordemos que en el caso de compass search, los generadores coinciden con las direcciones coordenadas).

Proposición 2.2.3. Sea $\mathcal{G} = \mathcal{D}_\oplus$. Si $x \in \Omega$, $\varepsilon \geq 0$, $K^\circ = K^\circ(x, \varepsilon)$ y $K = K(x, \varepsilon)$ se definen para el problema (2.4) entonces

$$\chi(x) \leq \| [-\nabla f(x)]_{K^\circ} \| + \sqrt{n} \| [-\nabla f(x)]_K \| \varepsilon.$$

Demostración. Ver demostración en [25]. □

Teorema 2.2.4. Sea $f : \mathbb{R}^n \rightarrow \mathbb{R}$ una función continuamente diferenciable. Si además ∇f es una función Lipschitz continua con constante L y $\| \nabla f(x) \| \leq \gamma$ para todo $x \in \Omega$ entonces el algoritmo compass para el caso con restricciones de cajas (2.4) produce una sucesión de iterados tal que para todo $k \in \mathcal{U}$

$$\chi(x^k) \leq \sqrt{n}(L + \gamma)\Delta^k.$$

Demostración. Ver demostración en [25]. □

De esta manera, en forma análoga al Teorema 2.2.1, Kolda, Lewis y Torczon lograron establecer una cota para $\chi(x^k)$ en términos de la longitud de paso Δ^k para el conjunto de las iteraciones no exitosas. En un desarrollo extenso de la teoría de compass search para el caso con restricciones de cajas, los autores de [25], lograron probar que

$$\liminf_{\substack{k \rightarrow \infty, \\ k \in \mathcal{U}}} \Delta^k = 0,$$

estableciendo así el resultado de convergencia deseado.

2.3. Nuevo algoritmo para minimización en cajas sin derivadas con búsqueda lineal no monótona.

2.3.1. Presentación

Como se adelantó en la introducción de este capítulo, Sección 2.1, uno de los primeros aportes de esta tesis doctoral consiste en una nueva propuesta algorítmica de optimización local para problemas sin derivadas. Particularmente, el problema a resolver consiste en

$$\underset{x \in \Omega}{\text{minimizar}} \quad f(x), \tag{2.5}$$

donde $f : \mathbb{R}^n \rightarrow \mathbb{R}$ es continuamente diferenciable (aunque sus derivadas no están disponibles o no son confiables) y $\Omega = \{x \in \mathbb{R}^n : l \leq x \leq u\}$ con $-\infty \leq l \leq u \leq \infty$.

El nuevo algoritmo se encuentra categorizado dentro de los métodos de búsqueda de patrones, en particular es un algoritmo del tipo *compass search* o de búsqueda coordenadas. Sin embargo, a diferencia de la propuesta de Kolda, Lewis y Torczon introducida en la Sección 2.2, en lugar de exigir un descenso simple en cada iteración similar a (2.3), incorpora una estrategia de descenso no monótona en la aceptación de las nuevas aproximaciones. Así, el nuevo algoritmo, desde este momento denominado **nmps** (por sus siglas en inglés *nonmonotone pattern search*), consiste en un nuevo método de búsqueda lineal no monótona sin derivadas.

Como se mencionó con anterioridad el algoritmo **nmps** tiene sus bases en la propuesta de [11], artículo que a su vez se encuentra basado en las ideas presentadas en [17], [33] y [36]. Sin embargo, el nuevo método posee ciertas características que lo diferecian del trabajo Diniz-Ehrhardt, Martínez y Raydán [11]:

1. En primer lugar, mientras que en [11] los autores se enfocan sólo en el caso irrestricto, el algoritmo **nmps** está destinado a resolver problemas de optimización con cotas en las variables.
2. En segundo lugar, mientras la teoría de convergencia desarrollada por Diniz-Ehrhardt et al. se basa fuertemente en cuestiones probabilísticas, la demostración de la convergencia del nuevo algoritmo sigue la línea desarrollada por Kolda, Lewis y Torczon en [25] enfocándose principalmente en el análisis de las propiedades geométricas de la región factible.
3. En tercer lugar, la búsqueda lineal no monótona que utiliza el algoritmo **nmps** difiere ligeramente del algoritmo modelo introducido en [11], donde la diferencia más notable es que algoritmo **nmps** no utiliza la sucesión acotada $\{\beta^k\}$ definida por los autores en [11].

La búsqueda lineal no monótona en nmps

Teniendo en cuenta las siguientes componentes:

- $M \in \mathbb{Z}_+$: Entero positivo que indica cuántas evaluaciones funcionales previas serán tenidas en cuenta en la búsqueda lineal,
- $\{\eta^k\}$: Sucesión sumable de términos positivos, es decir $\eta^k > 0$ para todo $k = 1, 2, \dots$ y
$$\sum_{k=0}^{\infty} \eta^k = \eta < \infty,$$
- $\Delta^k > 0$: Parámetro positivo que indica la longitud de paso desde x^k asociada a la iteración k ,
- $m(k) = \min\{k, M - 1\}$: número entero tal que la resta $k - m(k)$ indica la cantidad de evaluaciones funcionales consideradas hacia atrás partiendo de $f(x^k)$,

la estrategia de búsqueda lineal no monótona del algoritmo **nmps** se define como

$$f(x^k + \Delta^k d^k) < f_{\max}(x^k) + \eta^k - (\Delta^k)^2 \quad (2.6)$$

con

$$f_{\max}(x^k) = \max\{f(x^k), f(x^{k-1}), \dots, f(x^{k-\min\{k, M-1\}})\} = \max_{0 \leq j \leq m(k)} \{f(x^{k-j})\}, \quad (2.7)$$

y d^k perteneciente al conjunto de direcciones de búsqueda D_{\oplus} .

De esta manera, la desigualdad (2.6) dice que un nuevo punto $x^{k+1} = x^k + \Delta^k d^k$ será aceptado si su valor funcional es al menor que el máximo valor funcional de los últimos M términos más un término positivo (η^k) menos el paso actual (Δ^k) elevado al cuadrado. Esto significa que potenciales incrementos con respecto a la iteración anterior son aceptados. Sin embargo, como se observará en los resultados de convergencia, la no monotonía de la sucesión $\{f(x^k)\}$, la sumabilidad de $\{\eta^k\}$ y el hecho que las longitudes de paso convergan a 0 aseguran que en el límite de las iteraciones del algoritmo **nmps**, la aproximación al mínimo de la función es un punto KKT.

Observación 2.3.1. *La desigualdad (2.6) puede parecer a simple vista una idea contraintuitiva, al menos en la presencia de métodos sin derivadas. Sin embargo, la comparación de $f(x^{k+1})$ no sólo con $f(x^k)$ sino con una cierta cantidad de términos evaluados anteriormente podría brindarle mayor contexto local o conocimiento del comportamiento de la función al algoritmo. En consecuencia, el mismo podría moverse en direcciones adecuadas, acelerando así su convergencia y no gastando recursos en porciones de la región factible lejos del mínimo.*

En esta situación existe “trade-off” bastante evidente. Por un lado, comparar el iterado actual con una gran cantidad de puntos anteriores haría desaparecer la noción de contexto local mencionado anteriormente. Por otro lado, sin embargo, realizar la comparación considerando solo unos pocos puntos anteriores no brindaría mucha información adicional, además de que se asemejaría bastante a una búsqueda clásica del tipo Armijo [2]. Este “trade-off” tiene implicaciones mayormente numéricas ya que desde el punto de vista teórico la convergencia se garantiza para cualquier número M entero positivo.

Ejemplo 2.3.1. *Sea $M = 3$ y supóngase que el algoritmo **nmps** se encuentra en la quinta iteración, es decir $k = 5$. En esta situación se tendrá que $m(k) = m(5) = \min\{5, 2\} = 2$ y por (2.7),*

$$f_{\max}(x^5) = \max\{f(x^5), f(x^4), f(x^3)\},$$

lo cual verifica que el algoritmo operará siempre mirando el punto previo inmediato más dos anteriores para el caso de ejemplo.

2.3.2. Algoritmo nmps: Resultados teóricos y análisis de convergencia

A continuación, en base a lo expuesto en el apartado anterior, introducimos el algoritmo propuesto.

Algoritmo 2.2: Algoritmo nmps

Entrada:

- $f : \mathbb{R}^n \rightarrow \mathbb{R}$ Función a minimizar.
- $x^0 \in \mathbb{R}^n$: Aproximación inicial a la solución.
- $\Delta_{\text{tol}} > 0$: Tolerancia usada para el test de convergencia.
- $\Delta^0 > \Delta_{\text{tol}}$: Valor inicial del parámetro longitud de paso.
- $k < K_{\text{máx}}$: Cantidad máxima de iteraciones del algoritmo.

Salida:

- $f_{\text{mín}}$: Valor funcional mínimo encontrado.
- $x_{\text{mín}}$: Punto donde se da el valor funcional mínimo.

/* Inicialización

***/**

- 1 $\mathcal{D}_{\oplus} \leftarrow \{\pm e_i : i = 1, 2, \dots, n\}$.
 - 2 $f_{\text{mín}} \leftarrow f(x^0)$
 - 3 $x_{\text{mín}} \leftarrow x^0$
 - 4 $k \leftarrow 1$
 - 5 **mientras** $k < K_{\text{máx}}$ **hacer**
 - 6 $m(k) \leftarrow \min\{k, M - 1\}$
 - 7 $f_{\text{máx}}(x^k) \leftarrow \max\{f(x^k), \dots, f(x^{k-\min\{k, M-1\}})\} = \max_{0 \leq j \leq m(k)} \{f(x^{k-j})\}$
 - 8 **si** $f(x^k + \Delta^k d^k) < f_{\text{máx}}(x^k) + \eta^k - (\Delta^k)^2$ *para alguna dirección* $d \in \mathcal{D}_{\oplus}$ **entonces**
 - 9 $\Delta^{k+1} \leftarrow \min\{1, 2\Delta^k\}$
 - 10 $x^{k+1} \leftarrow x^k + \Delta^k d$
 - 11 **en otro caso**
 - 12 $x^{k+1} \leftarrow x^k$
 - 13 $\Delta^k \leftarrow \frac{\Delta^k}{2}$
 - 14 **si** $\Delta^{k+1} < \Delta_{\text{tol}}$ **entonces**
 - 15 $f_{\text{mín}} \leftarrow f(x^{k+1})$
 - 16 $x_{\text{mín}} \leftarrow x^{k+1}$
 - 17 Terminar la ejecución del algoritmo.
 - 18 $f_{\text{mín}} \leftarrow f(x^{k+1})$
 - 19 $x_{\text{mín}} \leftarrow x^{k+1}$
 - 20 **devolver** $f_{\text{mín}}, x_{\text{mín}}$
-

Observación 2.3.2. *Es importante notar que las iteraciones del algoritmo nmps están bien definidas ya que el hecho de que $\eta^k > 0$ para todo k asegura que la condición (2.6) se satisface cuando Δ^k es suficientemente pequeño. Además, si Δ^k es pequeño, es trivial que $(\Delta^k)^2$ es más pequeño aún.*

Así, el Algoritmo 2.2 posee dos principales diferencias con respecto a su antecesor *compass search*. La primera, y la más importante, se encuentra presente en la búsqueda lineal del nuevo iterado, donde el descenso simple (2.3) es sustituido por el descenso no monótono (2.6). En cuanto a la segunda diferencia, se puede observar que el algoritmo nmps define

$$\Delta^{k+1} = \min\{1, 2\Delta^k\} \tag{2.8}$$

en el caso de iteraciones exitosas. El punto de vista clásico nos dice que si la búsqueda logró una mejora (es decir, fue exitosa), la longitud de paso Δ^k debería, al menos, mantenerse en la próxima iteración. Este aspecto fue abordado en varios trabajos de Kolda y su equipo (por ejemplo en [25], [26], [47]) donde los autores investigaron diferentes estrategias fuera de la convención inicial.

La idea fundamental que atraviesa este análisis es que si con una longitud de paso, determinada por Δ^k , se logró minimizar la función, es probable que explorando una región levemente más grande esta mejora sea más significativa aún. Es por esta razón que los autores introducen un factor de aumento, usualmente igual a 2, cuando una iteración exitosa fue obtenida.

A nivel del nuevo método (algoritmo **nmpps**), definir Δ^{k+1} de acuerdo a (2.8) permite explorar una región mayor en la próxima iteración pero sin perder de vista el foco en la búsqueda local. El límite exploratorio está dado por el factor 1 y funciona principalmente como una herramienta que busca acelerar la convergencia del algoritmo.

Con el objetivo de probar el principal resultado de convergencia del algoritmo **nmpps**, algunos resultados auxiliares son requeridos. La siguiente proposición trata acerca de la no monotonía de la sucesión $\{f(x^k)\}$. Este resultado es análogo al Lema 2.3 presentado en [6].

Proposición 2.3.1. *Si $l(k)$ es un entero tal que*

$$kM - m(kM) \leq l(k) \leq kM,$$

con $m(kM) = \min\{kM, M - 1\}$, y

$$\begin{aligned} f(x^{l(k)}) &= \max_{0 \leq j \leq m(kM)} \{f(x^{kM-j})\} \\ &= \max\{f(x^{kM}), f(x^{kM-1}), \dots, f(x^{kM-\min\{kM, M-1\}})\} \end{aligned} \quad (2.9)$$

entonces

$$f(x^{l(k+1)}) \leq f(x^{l(k)}) + \eta_{kM} + \dots + \eta_{kM-M+1} - (\Delta^{l(k+1)-1})^2. \quad (2.10)$$

Demostración.

Supongamos inicialmente que la iteración k es tal que $k \geq M$, es decir

$$m(kM) = \min\{kM, M - 1\} = M - 1. \quad (2.11)$$

Utilizando un argumento inductivo sobre $t = 1, 2, \dots, M$ probaremos a continuación que la desigualdad

$$f(x^{kM+t}) \leq f(x^{l(k)}) + \eta_{kM} + \dots + \eta_{kM-M+1} - (\Delta^{kM+t-1})^2 \quad (2.12)$$

se mantiene para toda iteración $k = 1, 2, \dots$

En efecto, en base a (2.6), (2.7) y (2.11) y la hipótesis (2.9), se tiene que

$$\begin{aligned} f(x^{kM+1}) &\leq f_{\max}(x^{kM}) + \eta_{kM} - (\Delta^{kM})^2 \\ &= \max_{0 \leq j \leq M-1} \{f(x^{kM-j})\} + \eta_{kM} - (\Delta^{kM})^2 \\ &= f(x^{l(k)}) + \eta_{kM} - (\Delta^{kM})^2 \\ &< f(x^{l(k)}) + \eta_{kM}, \end{aligned}$$

para todo $k = 1, 2, \dots$ Es decir, la desigualdad (2.12) se mantiene para $t = 1$.

Utilizando la hipótesis inductiva supongamos ahora que

$$\begin{aligned} f(x^{kM+t'}) &\leq f(x^{l(k)}) + \eta_{kM} + \dots + \eta_{kM+t'-1} - (\Delta^{kM+t'-1})^2 \\ &< f(x^{l(k)}) + \eta_{kM} + \dots + \eta_{kM+t'-1} \end{aligned}$$

para todo $t' = 1, 2, \dots, t$.

A continuación probaremos que la desigualdad (2.12) se cumple para $t + 1$. En efecto,

$$\begin{aligned}
f(x^{kM+t+1}) &\leq f_{\text{máx}}(x^{kM+t}) + \eta_{kM+t} - (\Delta^{kM+t})^2 \\
&= \max_{0 \leq j \leq M-1} \{f(x^{kM+t-j})\} + \eta_{kM+t} - (\Delta^{kM+t})^2 \\
&= \max \left\{ \underbrace{f(x^{kM+t-M+1}), \dots, f(x^{kM+t-1}), f(x^{kM+t})}_{M \text{ elementos}} \right\} + \eta_{kM+t} - (\Delta^{kM+t})^2 \\
&= \max \left\{ \underbrace{\max \{f(x^{kM+t-M+1}), \dots, f(x^{kM-1}), f(x^{kM})\}}_{\leq f(x^{l(k)})}, \underbrace{f(x^{kM+1}), \dots, f(x^{kM+t})}_{\text{últimos } t \text{ elementos}} \right\} \\
&\quad + \eta_{kM+t} - (\Delta^{kM+t})^2 \\
&\leq \max \{f(x^{l(k)}), f(x^{kM+1}), \dots, f(x^{kM+t})\} + \eta_{kM+t} - (\Delta^{kM+t})^2.
\end{aligned}$$

Luego, por paso de inducción se tiene que

$$\max \{f(x^{kM+1}), \dots, f(x^{kM+t})\} < f(x^{l(k)}) + \eta_{kM} + \dots + \eta_{kM+t-1}.$$

De esta manera,

$$f(x^{kM+t+1}) \leq f(x^{l(k)}) + \eta_{kM} + \dots + \eta_{kM+t-1} + \eta_{kM+t} - (\Delta^{kM+t})^2,$$

así la desigualdad (2.12) se mantiene para $t + 1$.

Finalmente, dado que $(k + 1)M - M + 1 \leq l(k + 1) \leq kM + M$, tenemos

$$l(k + 1) = kM + t \text{ para algún } t \in \{1, 2, \dots, M\},$$

completando de esta manera la demostración. \square

La siguiente proposición es una herramienta importante para probar convergencia global del algoritmo `nmps` donde la desigualdad de la Proposición 2.3.1 será aplicada de forma iterativa. Esta idea también fue introducida por Birgin et al. en [6] y adaptada al caso de estudio.

Proposición 2.3.2. *Si $\{f(x^k)\}_{k \in \mathbb{N}}$ está acotada inferiormente entonces $\lim_{k \rightarrow \infty} (\Delta^{l(k)-1})^2 = 0$*

Demostración. Aplicando la desigualdad (2.10) de forma iterativa se tiene que

$$f(x^{l(k+1)}) \leq f(x^0) + \sum_{k=0}^{\infty} \eta_k - \sum_{k=1}^{\infty} (\Delta^{l(k)-1})^2,$$

o bien en su forma equivalente

$$\sum_{k=1}^{\infty} (\Delta^{l(k)-1})^2 \leq f(x^0) - f(x^{l(k+1)}) + \sum_{k=0}^{\infty} \eta_k.$$

Por otro lado, dado que f está acotada inferiormente se tiene $-f(x^k) \leq -C$ para todo k . Además, debido a la sumabilidad de la sucesión $\{\eta^k\}$, se deduce que

$$\sum_{k=1}^{\infty} (\Delta^{l(k)-1})^2 < +\infty,$$

con lo cual $\lim_{k \rightarrow \infty} (\Delta^{l(k)-1})^2 = 0$, probando de esta manera el resultado deseado. \square

Como consecuencia de la proposición anterior, se observa que

$$\lim_{k \rightarrow \infty} \Delta^{l(k)-1} = 0,$$

dado que los pasos Δ^k , por definición, son suficientemente pequeños y positivos.

A continuación se define el conjunto de índices

$$U = \{l(1) - 1, l(2) - 1, l(3) - 1, \dots\} \quad (2.13)$$

donde $\{l(k)\}$ es la sucesión de índices presentada en la Proposición 2.3.1.

A continuación, en base al desarrollo anterior, presentaremos el principal resultado de convergencia del algoritmo **nmps**. El mismo está basado en los resultados de Kolda et al. en [25] utilizando la búsqueda lineal (2.6).

Teorema 2.3.3. *Sea $f : \mathbb{R}^n \rightarrow \mathbb{R}$ continuamente diferenciable, y supongamos que ∇f es Lipschitz continua, con constante L , $\|\nabla f(x)\| \leq \gamma$, para todo $x \in \Omega$ y $\{f(x^k)\}_{k \in \mathbb{N}}$ está acotada inferiormente. Si $\{x^k\}_{k \in U}$ es la sucesión generada por el algoritmo **nmps** entonces*

$$\chi(x^k) \leq \sqrt{n}(L + \gamma)\Delta^k \text{ para todo } k \in U.$$

Demostración. Consideremos dos casos.

Caso 1.

Si $x^k + \Delta^k d \notin \text{int}(\Omega)$ para todo $d \in D_{\oplus}$, entonces $x^k + \Delta^k d$ se encuentra o bien en la frontera o bien afuera de la región factible Ω para todas las direcciones $d \in D_{\oplus}$.

En otras palabras, si $l_i \leq x_i^k \leq u_i$ entonces

$$x_i^k - \Delta^k \leq l_i \text{ y } x_i^k + \Delta^k \geq u_i$$

para todo $i = 1, 2, \dots, n$.

Las últimas desigualdades implican que si $x^k + \omega \in \Omega$, el vector ω no puede tener sus componentes más grandes que Δ^k , esto es, $\omega_i \leq \Delta^k$ para todo i . Por lo tanto, $\|\omega\| \leq \sqrt{n}\Delta^k$.

De esta manera,

$$\begin{aligned} \chi(x^k) &= \max_{\substack{x^k + \omega \in \Omega, \\ \|\omega\| \leq 1}} -\nabla f(x^k)^T \omega \\ &\leq \max_{\substack{x^k + \omega \in \Omega, \\ \|\omega\| \leq 1}} \|\nabla f(x^k)\| \|\omega\| \\ &\leq \|\nabla f(x^k)\| \sqrt{n}\Delta^k \\ &\leq \sqrt{n}\gamma\Delta^k, \end{aligned}$$

lo cual completa la demostración para el Caso 1.

Caso 2.

Supóngase ahora que existe al menos una dirección $d \in D_{\oplus}$ tal que $x^k + \Delta^k d \in \text{int}(\Omega)$. De esta manera, el cono $K^{\circ}(x^k, \Delta^k)$ es generado por todas las direcciones $d \in D_{\oplus}$ tal que $x^k + \Delta^k d \in \text{int}(\Omega)$.

Por el Teorema del valor medio, tenemos que

$$f(x^k + \Delta^k d^k) - f(x^k) = \Delta^k \nabla f(x^k + \lambda^k \Delta^k d^k)^T d^k, \quad (2.14)$$

para algún $\lambda^k \in [0, 1]$. Dado $k \in U$, esto implica

$$0 \leq f(x^k + \Delta^k d^k) - f_{\max}(x^k) - \eta^k + (\Delta^k)^2.$$

Tomando en cuenta que $-f_{\max}(x^k) \leq -f(x^k)$ y $\eta^k > 0$ para todo k , obtenemos

$$0 \leq f(x^k + \Delta^k d^k) - f(x^k) + (\Delta^k)^2. \quad (2.15)$$

Luego, reemplazando (2.14) en (2.15) se tiene que

$$0 \leq \Delta^k \nabla f(x^k + \lambda^k \Delta^k d^k)^T d^k + (\Delta^k)^2.$$

Ahora, dividiendo la última desigualdad por Δ^k y sumando $-\nabla f(x^k)^T d^k$ en ambos miembros, obtenemos la siguiente expresión

$$-\nabla f(x^k)^T d^k \leq (\nabla f(x^k + \lambda^k \Delta^k d^k) - \nabla f(x^k))^T d^k + \Delta^k.$$

Luego, usando la Proposición 2.2.2 tenemos

$$\frac{1}{\sqrt{n}} \|[-\nabla f(x)]_{K^\circ}\| \leq (\nabla f(x^k + \lambda^k \Delta^k d^k) - \nabla f(x^k))^T d^k + \Delta^k.$$

Utilizando la Desigualdad de Cauchy-Schwarz, el hecho de que $\|d^k\| = 1$ para todo k y dado que ∇f una función Lipschitz continua, obtenemos que

$$\frac{1}{\sqrt{n}} \|[-\nabla f(x)]_{K^\circ}\| \leq \|(\nabla f(x^k + \lambda^k \Delta^k d^k) - \nabla f(x^k))\| + \Delta^k \leq L\Delta^k + \Delta^k.$$

En consecuencia,

$$\|[-\nabla f(x)]_{K^\circ}\| \leq \sqrt{n}L\Delta^k + \sqrt{n}\Delta^k \leq \sqrt{n}L\Delta^k.$$

Finalmente, combinando la Proposición 2.2.3 con el resultado anterior

$$\begin{aligned} \chi(x) &\leq \|[-\nabla f(x)]_{K^\circ}\| + \sqrt{n} \|[-\nabla f(x)]_K\| \Delta^k \\ &\leq \|[-\nabla f(x)]_{K^\circ}\| + \sqrt{n}\gamma\Delta^k \leq \sqrt{n}L\Delta^k + \sqrt{n}\gamma\Delta^k \end{aligned}$$

en consecuencia

$$\chi(x) \leq \sqrt{n}(L + \gamma)\Delta^k,$$

y la demostración está completa. \square

Vale la pena mencionar que el algoritmo `nmfs` generará una sucesión de iterados $\{x^k\}_{k \in \mathbb{Z}}$ con $\Delta^k \rightarrow 0$, o bien terminará en un número finito de pasos. En la primera situación, tendremos que $\chi(x) \rightarrow 0$ por el Teorema 2.3.3. El segundo caso, podría pasar cuando el máximo número de iteraciones o el máximo número de evaluaciones funcionales establecidos es alcanzado.

2.3.3. Experimentos numéricos

En esta sección mostramos y analizamos los resultados numéricos obtenidos usando el nuevo algoritmo de búsqueda de patrones, `nmps`, para optimización con restricciones de cotas en las variables. Todos los experimentos numéricos de esta sección fueron llevados a cabo en una computadora con un procesador Intel Core i5-6200u de 2.3 GHz de 8 GB de RAM. Además, el algoritmo `nmps` fue implementado en `matlab R2016b 64-bit`.

El estudio numérico mencionado se encuentra organizado en dos partes con el objetivo de realizar un análisis cuidadoso y detallado de la performance del nuevo algoritmo. En primer lugar, se comparó el algoritmo `nmps` con la rutina `patternsearch` del `toolbox` de optimización de `matlab`, dado que ambas rutinas están basadas en un método de búsqueda de patrones. En segundo lugar, el estudio numérico fue extendido midiendo la performance del algoritmo `nmps` usando diferentes estrategias de búsqueda lineal ([7, 27, 40, 52]) y una adaptación de la clásica regla de Armijo [2] considerando $M = 1$ en el algoritmo `nmps`.

Para llevar a cabo el análisis se seleccionó un conjunto de 63 problemas con restricciones de la colección Hock-Schittkowski [20]. Dado que esta colección contiene solo 9 problemas con restricciones de cotas en las variables exclusivamente, se han modificado otros 54 problemas con restricciones generales, extrayendo las restricciones lineales y no lineales de cada uno de ellos. La lista de estos problemas y sus características es proporcionada en la Tabla 2.1 y en el Apéndice A respectivamente.

Tabla 2.1: Descripción de problemas de la colección Hock-Schittkowski.

Prob.	Nº. HS	Dimensión	nºrestricciones de cajas	Función objetivo
1	1	2	1	Polinomial general
2	2	2	1	Polinomial general
3	3	2	1	Polinomial general
4	4	2	2	Polinomial general
5	5	2	4	General
6	25	3	6	Suma de cuadrados
7	38	4	8	Polinomial general
8	45	5	10	Constante
9	110	10	20	General
10	13	2	2	Cuadrática
11	15	2	1	Polinomial general
12	16	2	3	Polinomial general
13	17	2	3	Polinomial general
14	18	2	4	Cuadrática
15	19	2	4	Polinomial general
16	20	2	2	Polinomial general
17	21	2	4	Cuadrática
18	23	2	4	Cuadrática
19	24	2	2	Polinomial general
20	30	3	6	Cuadrática
21	31	3	6	Cuadrática
22	32	3	3	Cuadrática
23	33	3	4	Polinomial general
24	34	3	6	Lineal
25	35	3	3	Cuadrática

Continua en la página siguiente

Tabla 2.1 – Continuación

Prob.	Nº. HS	Dimensión	nº restricciones de cajas	Función objetivo
26	36	3	6	Polinomial general
27	37	3	6	Polinomial general
28	41	4	8	Polinomial general
29	42	4	2	Cuadrática
30	44	4	4	Cuadrática
31	53	5	10	Cuadrática
32	54	6	12	General
33	55	6	8	General
34	57	2	2	Suma de cuadrados
35	59	2	4	General
36	60	3	6	Polinomial general
37	62	3	6	General
38	63	3	3	Cuadrática
39	64	3	3	Polinomial general
40	65	3	6	Cuadrática
41	66	3	6	Lineal
42	68	4	8	General
43	69	4	8	General
44	71	4	8	Polinomial general
45	72	4	8	Lineal
46	73	4	4	Lineal
47	74	4	8	Polinomial general
48	75	4	8	Polinomial general
49	76	4	4	Cuadrática
50	80	5	10	General
51	81	5	10	General
52	83	5	10	Cuadrática
53	84	5	10	Cuadrática
54	86	5	5	Polinomial general
55	93	6	6	Polinomial general
56	101	7	14	Polinomial general
57	102	7	14	Polinomial general
58	103	7	14	Polinomial general
59	104	8	16	Polinomial general
60	106	8	16	Lineal
61	108	9	2	Cuadrática
62	114	10	20	Cuadrática
63	119	16	32	Polinomial general

Como es usual en la implementación de los métodos de optimización sin derivadas, el mayor interés se encuentra en el número de evaluaciones funcionales necesarias para satisfacer algún criterio de parada. Los mismos pueden ser: obtener un tamaño de paso suficientemente pequeño ($\Delta^k < \Delta_{\text{tol}}$), alcanzar un máximo número de evaluaciones $FE_{\text{máx}}$ o un máximo número de iteraciones $K_{\text{máx}}$ estipulados. En esta tesis, particularmente dentro del estudio numérico de los algoritmos de optimización local, se adoptó el test de convergencia propuesto en [39]. Este test mide la habilidad de un algoritmo de mejorar una aproximación inicial y declarar que un

problema ha sido resuelto si la siguiente condición se cumple:

$$f(x^0) - f(x) \geq (1 - \tau)(f(x^0) - f_L), \quad (2.16)$$

donde x^0 es la aproximación inicial factible, $\tau > 0$ es el nivel de precisión y f_L es el valor funcional más pequeño obtenido entre todos los algoritmos comparados.

Moré y Wild observan en [39] que este test de convergencia es muy útil en un contexto de optimización sin derivadas y cuando se requiere elegir un algoritmo que proporcione una reducción funcional requerida dentro de un límite permitido de evaluaciones funcionales. Coloquialmente, la expresión (2.16), mide la reducción en el valor de la función $f(x^0) - f(x)$ alcanzada por un punto x con respecto a la mejor reducción posible $f(x^0) - f_L$. Para realizar el análisis mencionado, se utilizaron además los gráficos de *performance profile* [12, 39] de forma de ilustrar los resultados obtenidos a partir de (2.16).

Sea P el conjunto de problemas, $|P|$ su correspondiente cardinalidad y S el conjunto de *solvers* considerados. El perfil de rendimiento (por su traducción al español) de un solver $s \in S$ está definido como la fracción de problemas donde el coeficiente de rendimiento $r_{p,s}$ (conocido como *performance ratio* en inglés) es a lo sumo α , esto es,

$$\rho_s(\alpha) = \frac{1}{|P|} \text{size} \{p \in P : r_{p,s} \leq \alpha\},$$

con

$$r_{p,s} = \frac{t_{p,s}}{\min\{t_{p,s} : s \in S\}}$$

y $t_{p,s}$ el número de evaluaciones funcionales requeridas para satisfacer el test de convergencia (2.16). De esta manera, $\rho_s(\alpha)$ es la probabilidad para un *solver* $s \in S$ de que el coeficiente de rendimiento $r_{p,s}$ esté dentro de un factor $\alpha \in \mathbb{R}$ del mejor coeficiente posible.

Como exponen Dolan et al. en [12], la frase *performance profile* es utilizada para hacer referencia a la función de distribución de una métrica de rendimiento. Ejemplos de métricas de rendimiento podrían ser el tiempo de ejecución, el número de iteraciones o evaluaciones funcionales, entre otros. Esta forma de analizar los resultados numéricos son una alternativa eficiente, frente a las tradicionales tablas, cuando se quiere comparar el desempeño de muchos algoritmos en un gran conjunto de datos. Claramente, en estos casos, los resultados tabulares son muy difíciles de analizar y pueden estar sujetos a la subjetividad del programador o cualquier persona que analice los datos.

Una observación importante acerca de la función ρ_s , con $\rho_s : \mathbb{R} \rightarrow [0, 1]$, es que la misma es no decreciente, constante a trozos y continua a la derecha desde cada punto de corte. De acuerdo con el objetivo del análisis, la interpretación de los gráficos de *performance profiles* se realizan de forma diferente. Por ejemplo si el interés es obtener el número de *ganadores*, se deben comparar los valores $\rho_s(1)$ para cada algoritmo estudiado. En cambio, si se busca detectar aquellos *solvers* con una alta probabilidad de éxito, es necesario comparar todos los valores de ρ_s^* , con

$$\rho_s^* \equiv \lim_{\alpha \leftarrow r_M} \rho_s(\alpha) \text{ donde } r_{ps} \in [1, r_M],$$

para cada algoritmo y elegir aquel con el valor más grande. Para mayor detalle de las propiedades teóricas de la función ρ y estudios de casos, véase [12] y [39].

Desde el punto de vista paramétrico, se utilizó la misma aproximación inicial x^0 que la propuesta en [20], proyectando sobre las restricciones de cajas si dicha aproximación no era factible. Además, luego de algunas pruebas numéricas preliminares se adoptó $\eta^k = 1.1^{-k}$ para todo k . Finalmente, se establecieron los siguientes valores:

- $\Delta^0 = 1$,
- $M = 15$,
- $FE_{\text{máx}} = 2500$,
- $K_{\text{máx}} = 5000$,
- $\Delta_{\text{tol}} = \text{TOL} = 10^{-6}$.

Consideramos importante mencionar tres detalles fundamentales de implementación del algoritmo `nmps`. En primer lugar, la función f es evaluada en todas las posibles direcciones coordenadas y la aceptación de una nueva aproximación es tal que produce el mínimo valor funcional entre todos los $f(x^k + \Delta^k d)$. En segundo lugar, la longitud de paso es actualizada usando el siguiente esquema $\Delta^{k+1} = \min\{1, 2\Delta^k\}$, tal cual se observa en el Algoritmo 2.2. En tercer lugar, los puntos aceptados son almacenados en memoria con el propósito de evitar reevaluar puntos viejos sin ralentizar la implementación. Esta última idea fue desarrollada con más profundidad en [30].

2.3.3.1. Comparación de performance profiles entre `nmps` y `patternsearch`

El algoritmo `nmps` fue testeado usando el conjunto de problemas test antes mencionado y comparado con la rutina `patternsearch` de `matlab`. Dado que ambos códigos están basados en un esquema de búsqueda de patrones, se establecieron los mismos parámetros algorítmicos en ambas implementaciones. Los resultados numéricos se muestran en la Tabla 2.2.

En la Figura 2.3 se muestran los gráficos de performance profile usando la condición (2.16) con tres niveles de precisión: $\tau = 10^{-1}$, $\tau = 10^{-3}$ y $\tau = 10^{-5}$. Es importante notar que un pequeño valor de τ significa que la satisfacción de la condición (2.16) es más estricta. En un gráfico de performance profile, la curva superior representa el método más eficiente dentro de un factor τ de la mejor medida. Cuando ambos métodos coinciden con el mejor resultado, ambos son contados como exitosos. En este caso, la suma de los porcentajes exitosos puede exceder 100%.

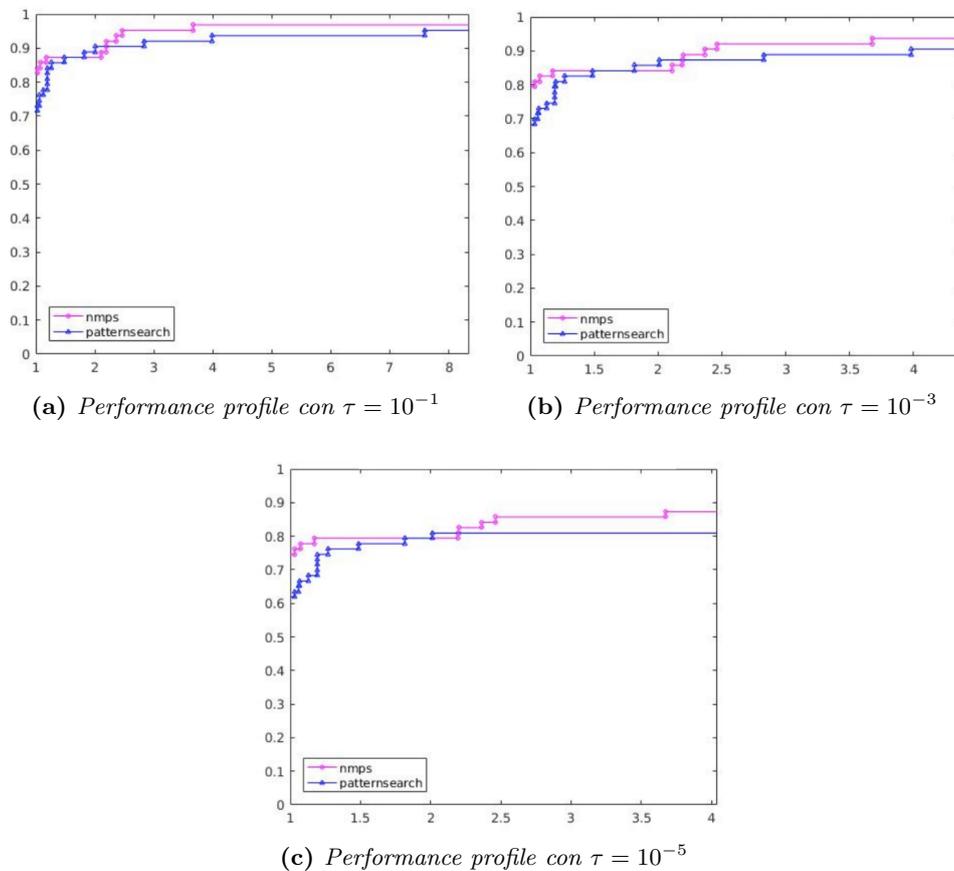


Figura 2.3: Comparación de performance profiles de *nmps* y *patternsearch*.

En la Figura 2.3a, con $\tau = 10^{-1}$, se observa que el algoritmo *nmps* es el mejor solver en el 82% del conjunto de problemas mientras que *patternsearch* lo es en el 71%. Además, con un factor de 1.7 del mejor solver, ambos algoritmos tienen un comportamiento similar y el gráfico de *performance profile* muestra que estos algoritmos pueden resolver un problema con una probabilidad de 0.87 con respecto al mejor solver. Finalmente, si el objetivo es resolver eficientemente el 95% de los problemas, el algoritmo *nmps* logra este porcentaje utilizando 2.4 veces el mínimo número de evaluaciones funcionales mientras que *patternsearch* necesita un factor de 7.5.

En la Figura 2.3b, incrementando el nivel de precisión a $\tau = 10^{-3}$, notamos que *nmps* gana en el 79% de los problemas en comparación con el 68% de *patternsearch*. Además, en caso de requerir que la solución esté en un factor de 1.7, ambos solvers tienen un desempeño equivalente con una probabilidad de 0.84. Aunque los dos algoritmos alcanzan la solución en a lo sumo el 95% de los problemas, *nmps* estuvo más cerca que *patternsearch* resolviendo el 94% de ellos y usando 3.7 veces el mínimo número de evaluaciones funcionales.

Finalmente, se observa en la Figura 2.3c que el desempeño de ambos *solvers* empeora usando $\tau = 10^{-5}$ como nivel de precisión. En cualquier caso, el algoritmo *nmps* gana en el 74% de los problemas mientras que *patternsearch* gana en el 62%. Similar al párrafo anterior, el algoritmo *nmps* se desempeña mejor si se considera un solver que encuentra la solución usando 1.7 veces el mínimo número de evaluaciones funcionales y lo hace con una probabilidad de 0.8. En cualquier situación, podríamos esperar a lo sumo que el algoritmo *nmps* resuelva el 87% de los problemas

mientras que `patternsearch` el 81 %.

Como conclusión de esta sección podemos decir que, independientemente del nivel de precisión, el algoritmo `nmjs` es competitivo con respecto a `patternsearch`, e incluso en algunas circunstancias llega a superarlo (siempre en el contexto del conjunto de problemas test considerado). En efecto, el algoritmo propuesto alcanza hasta 10% más de probabilidad en una gran cantidad de casos, con respecto a `patternsearch`, al momento de obtener la solución.

En la siguiente sección, se continuará analizando el desempeño de la nueva propuesta algorítmica utilizando otras estrategias de búsqueda lineal. El objetivo final es estudiar y entender las ventajas y desventajas de emplear diferentes esquemas de búsqueda lineal no monótonas.

2.3.3.2. Comparación de performance profiles usando otras estrategias de búsqueda lineal

Recientemente, algunos autores propusieron diferentes estrategias de búsqueda lineal no monótona para resolver problemas de minimización sin restricciones y sistemas de ecuaciones no lineales, ambos métodos basados tanto en derivadas como sin derivadas [7, 27, 40, 52]. En esta tesis, adaptamos estas estrategias a la resolución del problema de optimización sin derivadas (2.5).

Tabla 2.2: Resultados de las comparaciones numéricas

Prob.	nmps			patternsearch			λ patternsearch			Cpatternsearch			nmps-M1		
	It.	FE.	stop	It.	FE.	stop	It.	FE.	stop	It.	FE.	stop	It.	FE.	stop
1	341	352	TOL	625	2500	$FE_{m\acute{a}x}$	339	352	TOL	530	352	TOL	1003	2500	$FE_{m\acute{a}x}$
2	402	323	TOL	58	207	TOL	5000	6	$K_{m\acute{a}x}$	628	281	TOL	277	155	TOL
3	277	83	TOL	39	119	TOL	1660	2500	$FE_{m\acute{a}x}$	307	83	TOL	1659	2500	$FE_{m\acute{a}x}$
4	277	46	TOL	34	84	TOL	5000	14	$K_{m\acute{a}x}$	307	46	TOL	277	48	TOL
5	293	305	TOL	36	145	TOL	269	337	TOL	305	314	TOL	269	336	TOL
6	700	2500	$FE_{m\acute{a}x}$	513	2500	$FE_{m\acute{a}x}$	5000	1330	$K_{m\acute{a}x}$	670	2500	$FE_{m\acute{a}x}$	552	2500	$FE_{m\acute{a}x}$
7	1124	2500	$FE_{m\acute{a}x}$	176	1409	TOL	5000	177	$K_{m\acute{a}x}$	974	2500	$FE_{m\acute{a}x}$	455	2500	$FE_{m\acute{a}x}$
8	278	219	TOL	50	290	TOL	5000	134	$K_{m\acute{a}x}$	307	132	TOL	278	248	TOL
9	262	2500	$FE_{m\acute{a}x}$	129	2500	TOL	141	2500	$FE_{m\acute{a}x}$	211	2500	$FE_{m\acute{a}x}$	141	2500	$FE_{m\acute{a}x}$
10	278	64	TOL	26	77	TOL	278	64	TOL	307	64	TOL	278	64	TOL
11	277	158	TOL	625	2500	$FE_{m\acute{a}x}$	5000	106	$K_{m\acute{a}x}$	306	156	TOL	277	539	TOL
12	278	150	TOL	93	343	TOL	5000	98	$K_{m\acute{a}x}$	306	139	TOL	278	143	TOL
13	278	150	TOL	93	343	TOL	5000	98	$K_{m\acute{a}x}$	306	139	TOL	278	143	TOL
14	278	45	TOL	37	92	TOL	5000	17	$K_{m\acute{a}x}$	307	45	TOL	278	45	TOL
15	278	77	TOL	48	133	TOL	278	77	TOL	5000	77	$K_{m\acute{a}x}$	278	77	TOL
16	278	151	TOL	93	345	TOL	5000	99	$K_{m\acute{a}x}$	306	140	TOL	278	144	TOL
17	277	63	TOL	21	64	TOL	277	63	TOL	307	63	TOL	277	63	TOL
18	288	260	TOL	34	137	TOL	277	289	TOL	305	314	TOL	270	369	TOL
19	834	2500	$FE_{m\acute{a}x}$	627	2500	$FE_{m\acute{a}x}$	834	2500	$FE_{m\acute{a}x}$	834	2500	$FE_{m\acute{a}x}$	834	2500	$FE_{m\acute{a}x}$
20	278	108	TOL	26	131	TOL	5000	23	$K_{m\acute{a}x}$	307	108	TOL	278	108	TOL
21	278	106	TOL	26	128	TOL	5000	21	$K_{m\acute{a}x}$	307	106	TOL	278	106	TOL
22	277	76	TOL	47	183	TOL	277	76	TOL	306	76	TOL	277	82	TOL
23	277	712	TOL	37	129	TOL	277	708	TOL	307	791	TOL	277	704	TOL
24	278	480	TOL	128	747	TOL	278	480	TOL	307	480	TOL	278	480	TOL
25	289	480	TOL	130	774	TOL	278	508	TOL	305	522	TOL	273	625	TOL
26	281	192	TOL	95	402	TOL	281	192	TOL	5000	192	$K_{m\acute{a}x}$	281	192	TOL
27	5000	443	$K_{m\acute{a}x}$	34	86	TOL	5000	443	TOL	5000	437	$K_{m\acute{a}x}$	5000	443	$K_{m\acute{a}x}$
28	278	81	TOL	20	81	TOL	5000	13	$K_{m\acute{a}x}$	307	81	TOL	278	81	TOL
29	278	146	TOL	43	259	TOL	5000	44	$K_{m\acute{a}x}$	307	146	TOL	278	146	TOL
30	625	2500	$FE_{m\acute{a}x}$	418	2500	$FE_{m\acute{a}x}$	625	2500	$FE_{m\acute{a}x}$	625	2500	$FE_{m\acute{a}x}$	625	2500	$FE_{m\acute{a}x}$
31	288	800	TOL	39	391	TOL	277	939	TOL	306	965	TOL	270	1130	TOL
32	278	13	TOL	247	2500	$FE_{m\acute{a}x}$	278	13	TOL	307	287	TOL	278	13	TOL
33	277	1818	TOL	51	398	TOL	384	2500	$FE_{m\acute{a}x}$	306	2011	TOL	277	1810	TOL

Continúa en la página siguiente

Tabla 2.2 – Continuación

Prob.	nmps			patternsearch			λ patternsearch			Cpatternsearch			nmps-M1		
	<i>It.</i>	<i>FE.</i>	stop	<i>It.</i>	<i>FE.</i>	stop	<i>It.</i>	<i>FE.</i>	stop	<i>It.</i>	<i>FE.</i>	stop	<i>It.</i>	<i>FE.</i>	stop
34	280	140	TOL	61	191	TOL	5000	77	$K_{\text{máx}}$	307	120	TOL	278	134	TOL
35	278	326	TOL	119	410	TOL	277	320	TOL	307	335	TOL	277	509	TOL
36	403	737	TOL	125	751	TOL	294	746	TOL	322	755	TOL	278	744	TOL
37	285	287	TOL	48	210	TOL	284	122	TOL	5000	275	TOL	284	122	TOL
38	500	2500	$FE_{\text{máx}}$	417	2500	$FE_{\text{máx}}$	500	2500	$FE_{\text{máx}}$	500	2500	$FE_{\text{máx}}$	500	2500	$FE_{\text{máx}}$
39	392	1501	TOL	325	1945	TOL	5000	1135	$K_{\text{máx}}$	5000	1447	$K_{\text{máx}}$	634	2500	TOL
40	278	106	TOL	70	297	TOL	5000	61	$K_{\text{máx}}$	306	106	TOL	278	106	TOL
41	278	592	TOL	140	782	TOL	278	750	TOL	307	592	TOL	420	1842	TOL
42	277	1241	TOL	46	288	TOL	277	1234	TOL	307	1377	TOL	278	1238	TOL
43	282	1373	TOL	48	299	TOL	278	1239	TOL	445	2069	TOL	278	1226	TOL
44	278	110	TOL	53	252	TOL	5000	38	$K_{\text{máx}}$	307	110	TOL	278	110	TOL
45	278	107	TOL	76	431	TOL	5000	39	$K_{\text{máx}}$	306	107	TOL	278	107	TOL
46	278	100	TOL	66	352	TOL	278	100	TOL	306	100	TOL	278	100	TOL
47	278	121	TOL	20	117	TOL	278	121	TOL	307	107	TOL	278	121	TOL
48	278	106	TOL	20	113	TOL	278	106	TOL	307	104	TOL	278	106	TOL
49	278	195	TOL	60	426	TOL	278	195	TOL	306	195	TOL	277	177	TOL
50	278	457	TOL	74	484	TOL	278	457	TOL	307	181	TOL	278	457	TOL
51	287	538	TOL	46	396	TOL	364	2500	$FE_{\text{máx}}$	318	663	TOL	361	2500	$FE_{\text{máx}}$
52	284	1289	TOL	20	101	TOL	284	1289	TOL	5000	155	TOL	284	1289	TOL
53	405	2500	$FE_{\text{máx}}$	625	2500	$FE_{\text{máx}}$	405	2500	$FE_{\text{máx}}$	405	2500	$FE_{\text{máx}}$	405	2500	$FE_{\text{máx}}$
54	602	1443	TOL	85	826	TOL	249	1016	TOL	450	1275	TOL	252	1045	TOL
55	277	1856	TOL	106	824	TOL	278	1861	TOL	307	2055	TOL	277	1849	TOL
56	357	2500	$FE_{\text{máx}}$	233	2500	$FE_{\text{máx}}$	5000	547	$K_{\text{máx}}$	361	2500	$FE_{\text{máx}}$	247	2500	$FE_{\text{máx}}$
57	376	2500	$FE_{\text{máx}}$	233	2500	$FE_{\text{máx}}$	5000	547	$K_{\text{máx}}$	347	2500	$FE_{\text{máx}}$	255	2500	$FE_{\text{máx}}$
58	376	2500	$FE_{\text{máx}}$	232	2500	$FE_{\text{máx}}$	5000	964	$K_{\text{máx}}$	345	2500	$FE_{\text{máx}}$	257	2500	$FE_{\text{máx}}$
59	278	621	TOL	100	1294	TOL	213	2500	$FE_{\text{máx}}$	306	621	TOL	215	2500	$FE_{\text{máx}}$
60	167	2500	$FE_{\text{máx}}$	157	2500	$FE_{\text{máx}}$	167	2500	$FE_{\text{máx}}$	167	2500	$FE_{\text{máx}}$	167	2500	$FE_{\text{máx}}$
61	167	2500	$FE_{\text{máx}}$	157	2500	$FE_{\text{máx}}$	167	2500	$FE_{\text{máx}}$	167	2500	$FE_{\text{máx}}$	167	2500	$FE_{\text{máx}}$
62	149	2500	$FE_{\text{máx}}$	150	2500	$FE_{\text{máx}}$	149	2500	$FE_{\text{máx}}$	149	2500	$FE_{\text{máx}}$	149	2500	$FE_{\text{máx}}$
63	278	2446	TOL	104	2500	$FE_{\text{máx}}$	5000	2142	$K_{\text{máx}}$	307	2446	TOL	278	2446	TOL

Originalmente, Zhang y Hager en [52] propusieron una estrategia no monótona para resolver problemas de optimización sin restricciones y con derivadas. Los autores garantizaron que el promedio de los sucesivos valores funcionales es decreciente. Más tarde, Cheng y Li en [7] combinaron este enfoque con las ideas de Li y Fukushima [33] para resolver sistemas de ecuaciones no lineales sin derivadas. Finalmente, Nikolovski y Stojkoska [40] y Krejić et al. [27] consideraron esta última estrategia para resolver problemas de optimización irrestrictos sin derivadas. Este es el primer enfoque utilizado para las comparaciones numéricas. El mismo fue denominado en esta tesis como *C-line search* e implementado en el algoritmo `Cpatternsearch`. Este esquema es similar a la estrategia de búsqueda lineal (2.6), donde $f_{\max}(x^k)$ es reemplazado por la sucesión $\{C_k\}$ dada por

$$Q_{k+1} = r_k Q_k + 1, \quad C_{k+1} = \frac{r_k Q_k (C_k + \eta^k) + f_{k+1}}{Q_{k+1}} \quad (2.17)$$

con $Q_0 = 1$, $C_0 = f(x^0)$, $r_k \in [0, 1]$ and $\{\eta^k\}$ satisfaciendo $\sum_{k=0}^{\infty} \eta^k = \eta < \infty$ para todo $k = 0, 1, 2, \dots$

La idea detrás del segundo enfoque considerado surgió en un artículo publicado por Ulbrich [48], quien propuso un método de región de confianza para sistemas de ecuaciones no lineales semisuaves con cotas en las variables. Ulbrich reformuló el mismo en un problema de minimización diferenciable con restricciones. Este esquema motivó luego a Yu y Pu [51], quienes propusieron una técnica no monótona para minimización irrestricta con derivadas. En 2013, Nikolovski y Stojkoska [40] adaptaron esta regla para resolver problemas de optimización irrestrictos sin derivadas. Esta fue elegida como segunda estrategia en este estudio numérico. La misma fue denominada λ -*line search* e implementada en el algoritmo `λpatternsearch` y es también análoga (2.6) pero en este caso $f_{\max}(x^k)$ está definida por

$$f_{\max}(x^k) = \max\{f(x^k), \sum_{r=0}^{m(k)-1} \lambda_{k_r} f(x^{k-r})\} \quad (2.18)$$

con $M \in \mathbb{N}$, $m(k) = \min\{k, M - 1\}$, $\lambda_{k_r} \geq \lambda$ y $\sum_{r=0}^{m(k)-1} \lambda_{k_r} = 1$ para todo $k = 0, 1, 2, \dots$

Finalmente, se consideró un caso especial del Algoritmo `nmps` con $M = 1$, es decir, $f_{\max}(x^k)$ fue reemplazado por $f(x^k)$ en (2.6), obteniendo de esta manera

$$f(x^k + \Delta^k d) \leq f(x^k) + \eta^k - (\Delta^k)^2,$$

donde la sucesión $\{\eta^k\}$ y la longitud de paso Δ^k fueron definidos como en el algoritmo `nmps`. La implementación de esta estrategia se realizó en el algoritmo denominado `nmps-M1`.

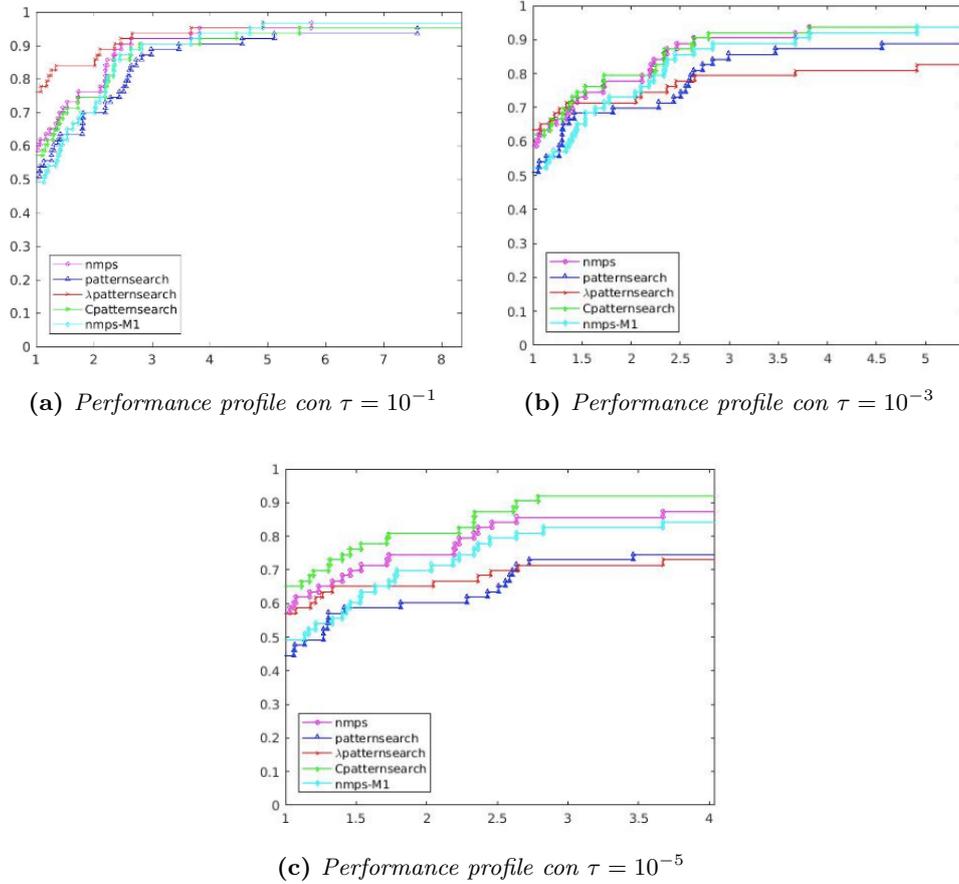


Figura 2.4: Comparación de performance profiles de las cinco estrategias consideradas.

Nuevamente se consideró $\eta^k = 1.1^{-k}$ para todo k , para cada uno de los tres nuevos esquemas. Además, se adoptó $M = 15$ y $\lambda^{k_r} = 1/m(k)$ para todo r en λ patternsearch y $r_k = 0.85$ para todo k en Cpatternsearch.

En la Figura 2.4, se muestran los gráficos de *performance profile* obtenidos con el test de convergencia (2.16) en la resolución de los problemas test utilizando los algoritmos nmmps, patternsearch, Cpatternsearch, lambda patternsearch y nmmps-M1. Los resultados obtenidos se encuentran en la Tabla 2.2.

En la Figura 2.4a, para $\tau = 10^{-1}$, se observa que λ patternsearch alcanza la mejor performance en el 76 % de los problemas, seguido por nmmps con el 59 % y Cpatternsearch con el 57 % y nmmps-M1 con el 56 % y 51 % respectivamente. Además se advierte que, dentro de un factor de 2.5 del mejor solver, λ patternsearch alcanza la probabilidad más grande de resolver un problema (alrededor de 0.92) y seguido por nmmps con 0.9. Además, en este caso patternsearch es el solver con la performance más baja obteniendo una probabilidad de 0.76. Con este nivel de precisión, notamos que nmmps-M1 resuelve a lo sumo el 97 % de los problemas usando un factor de 4.9 veces el mínimo número de evaluaciones funcionales, mientras que nmmps tiene un comportamiento similar requiriendo un factor de 5.75.

La Figura 2.4b muestra el gráfico de *performance profile* para $\tau = 10^{-3}$, incrementando de esta manera el nivel de precisión. Una vez más, λ patternsearch es el *solver* más exitoso con una probabilidad de 0.63 seguido por Cpatternsearch y nmmps con una probabilidad de 0.62 y 0.6, respectivamente. Al último se encuentran nmmps-M1 y patternsearch con 0.52 y 0.51 respectivamente. Dentro de un factor de 2.5 del mejor solver, nmmps exhibe el mejor desempeño

resolviendo 89% de los problemas. Luego, `Cpatternsearch` y `nmps-M1` obtienen 87% y 85% respectivamente. Con este nivel de precisión, se puede esperar a lo sumo resolver el 94% de los problemas dentro de un factor de 3.8 del mejor *solver*. En este sentido, tanto `Cpatternsearch` como `nmps` presentan un comportamiento similar.

Finalmente, la Figura 2.4c presenta el gráfico de *performance profile* para $\tau = 10^{-5}$, exigiendo de esta manera un mayor descenso en el valor de la función objetivo. En este caso se observa que `Cpatternsearch` gana en el 65% de los problemas, seguido por `nmps` y `λpatternsearch` con el 57%, `nmps-M1` con el 49% y `patternsearch` con 44%. Además vemos que, dentro de un factor de 2.5 del mejor *solver*, `Cpatternsearch` obtiene la probabilidad más alta para resolver un problema (0.87) seguido por `nmps` (0.84). En esta Figura, a diferencia de lo que sucedía en las Figuras 2.4a y 2.4b, podemos observar cómo el desempeño de los diferentes métodos se separan significativamente unos de otros. Vale la pena notar en relación a este punto que el algoritmo `Cpatternsearch` se encuentra siempre por encima de los demás *solvers*. En este caso, `Cpatternsearch` alcanza la solución en el 92% de los problemas dentro de un factor de 2.79, seguido por `nmps` que resuelve el 87% de los problemas usando 3.6 veces el mínimo número de evaluaciones funcionales. Los demás algoritmos pueden solo resolver a lo sumo el 80% de los problemas empleando factores mucho más grandes.

A partir del análisis de las figuras incluidas en la Figura 2.4 es posible obtener las siguientes conclusiones. En primer lugar, en la mayoría de los casos el uso de una búsqueda lineal no monótona del tipo (2.6), (2.17) o (2.18) resulta en una ventaja en el desempeño de los algoritmos en comparación con `nmps-M1`. En otras palabras, un mayor esfuerzo dedicado a obtener $f_{\max}(x^k)$ o C_k resulta en un descenso en el número de evaluaciones funcionales llevadas a cabo por el algoritmo, el cual es uno de los objetivos principales de los algoritmos sin derivadas. En segundo lugar, aunque las estrategias (2.6) y (2.18) tienen definiciones similares, el análisis anterior muestra que mientras `λpatternsearch` disminuye su *performance* a medida que el nivel de precisión aumenta, `nmps` se mantiene bastante consistente para todos los valores de τ . De esta manera, si uno quisiera elegir entre estas dos estrategias, `nmps` sería más adecuado. En tercer lugar, el algoritmo `Cpatternsearch` puede ser considerado como el *solver* con la mejor *performance* dado que siempre se encuentra por encima de los demás, incluso a medida que el nivel de precisión aumenta. El algoritmo propuesto en este capítulo, `nmps`, por su parte, resultó ser un buen competidor de `Cpatternsearch`, logrando una *performance* similar con respecto a este último en varios casos. Además, si se tiene en cuenta la probabilidad de resolver un problema, `nmps` siempre obtiene el segundo lugar. Finalmente, para nuestra sorpresa, se observa que la *performance* del algoritmo `patternsearch` está en muchos casos por debajo de los restantes métodos.

2.3.3.3. Experimentos numéricos adicionales

En esta sección, se muestran algunos experimentos numéricos adicionales donde el conjunto de direcciones coordenadas D_{\oplus} es extendido añadiendo un subconjunto de direcciones aleatorias. Esta idea se origina como una alternativa algorítmica propuesta por Kolda et al. [25] donde el conjunto de direcciones está dado por el conjunto D_{\oplus} más un conjunto adicional de direcciones. Los autores afirman que tales direcciones no son relevantes en el análisis de convergencia pero que pueden contribuir a acelerar la convergencia del algoritmo.

Tabla 2.3: Comparación numérica entre D_{\oplus} y \bar{D}_2 para $n = 2$.

N°	Prob.	D_{\oplus}					\bar{D}_2				
		<i>It.</i>	<i>FE.</i>	stop	$\frac{\ x_* - x_{\min}\ }{\ x_*\ }$	$\ f_* - f_{\min}\ $	<i>It.</i>	<i>FE.</i>	stop	$\frac{\ x_* - x_{\min}\ }{\ x_*\ }$	$\ f_* - f_{\min}\ $
1	1	341	352	TOL	0	0	1903	2500	FE_{\max}	1.0222	0.28196
2	2	402	323	TOL	1.2630	4.8908	2014	2500	MaxFE	1.1780	0.79013
3	3	277	83	TOL	0	0	277	145	TOL	0	0
4	4	277	46	TOL	0	0	277	90	TOL	0	0
5	5	293	305	TOL	1.8034e-7	7.5939e-14	293	549	TOL	1.8034e-07	7.5939e-14
6	10	278	64	TOL	0	0	278	108	TOL	0	0
7	11	277	158	TOL	1.7889e-8	0	285	214	TOL	5.4942e-07	8.8290e-12
8	12	278	150	TOL	0	0	325	324	TOL	1.4137e-07	6.2456e-13
9	13	278	150	TOL	0	0	278	271	TOL	4.5334e-07	6.4225e-12
10	14	278	45	TOL	0	0	278	89	TOL	0	0
11	15	278	77	TOL	0	0	278	145	TOL	0	0
12	16	278	151	TOL	1.7889e-11	0	282	244	TOL	7.2292e-08	1.6342e-13
13	17	277	63	TOL	0	0	277	105	TOL	0	0
14	18	288	260	TOL	0	0	288	440	TOL	0	0
15	19	834	2500	MaxFE	1	1.0039e44	498	2500	MaxFE	1	1.0039e+44
16	34	280	140	TOL	1.5996e-6	3.4591e-4	280	264	TOL	1.5996e-06	3.4591e-04
17	35	278	326	TOL	3.6650e-6	2.4636e-4	278	551	TOL	3.6650e-06	2.4636e-04

Tabla 2.4: Comparación numérica entre D_{\oplus} y \bar{D}_3 para $n = 3$.

N°	Prob.	D_{\oplus}					\bar{D}_2				
		<i>It.</i>	<i>FE.</i>	stop	$\frac{\ x_* - x_{\min}\ }{\ x_*\ }$	$\ f_* - f_{\min}\ $	<i>It.</i>	<i>FE.</i>	stop	$\frac{\ x_* - x_{\min}\ }{\ x_*\ }$	$\ f_* - f_{\min}\ $
1	6	700	2500	MaxFE	0.60590	0.20012	441	2500	MaxFE	0.89292	0.0057732
2	20	278	108	TOL	0	0	278	174	TOL	0	0
3	21	278	106	TOL	0	0	278	169	TOL	0	0
4	22	277	76	TOL	0	0	277	145	TOL	0	0
5	23	277	712	TOL	0	0	277	1543	TOL	0	0
6	24	278	480	TOL	0	0	278	321	TOL	0.51509	0
7	25	289	480	TOL	0	0	289	783	TOL	0	0
8	26	281	192	TOL	0	0	280	122	TOL	0	0
9	27	5000	443	MaxIt	0	0	5000	257	TOL	0	0
10	36	403	737	TOL	1.5573e-06	3.6380e-12	403	1256	TOL	1.5573e-06	3.6380e-12
11	37	285	287	TOL	7.1668e-08	8.0806e-07	285	505	TOL	7.1668e-08	8.0806e-07
12	38	500	2500	MaxFE	0	0	278	2500	MaxFE	3.8832	9.0447e+06
13	39	392	1501	TOL	4.0736e-06	2.5720e-08	278	1280	TOL	4.0665e-06	2.5720e-08
14	40	278	106	TOL	0	1.1111e-06	278	80	TOL	0	1.1111e-06
15	41	278	592	TOL	0	0	277	391	TOL	0.61276	0

Estos experimentos numéricos se realizaron usando el conjunto de problemas test con dimensión 2 y 3 de la Tabla 2.1. Los resultados son presentados en la Tabla 2.3 y la Tabla 2.4 respectivamente. En ambos casos las direcciones adicionales fueron generadas aleatoriamente y el conjunto de las mismas fue denotado con \bar{D}_2 y \bar{D}_3 . Es importante mencionar que se mantuvo el conjunto D_{\oplus} por dos razones. En primer lugar, para garantizar tener un conjunto generador positivo de direcciones. En segundo lugar, para asegurar que las búsquedas de los nuevos puntos sea capaz de cubrir la región factible.

Se compararon las soluciones obtenidas por el algoritmo `nmjs` con las soluciones conocidas para estos problemas, utilizando el error relativo definido por $\frac{\|x_* - x_{\min}\|}{\|x_*\|}$. En los casos de los problemas cuya solución es el vector nulo, se optó por calcular el error relativo usando la fórmula $\frac{\|x_* - x_{\min}\|}{1 + \|x_*\|}$.

En la Tabla 2.3, se observa que este cambio en el conjunto de direcciones búsqueda no afecta la precisión de la solución en comparación con las obtenidas previamente. Sin embargo, notamos que el número de evaluaciones funcionales incrementa en casi todos los casos. Por otro lado, en la Tabla 2.4 se observa que, mientras el número de evaluaciones funcionales es comparable en ambas implementaciones, la precisión de la solución muestra algunas diferencias.

Finalmente, para $n = 2$, los resultados numéricos muestran que el conjunto de direcciones coordenadas continua siendo la elección más apropiada. En dimensión 3, aunque los resultados numéricos pueden ser comparables, tener más direcciones podría incrementar el tiempo de CPU.

Capítulo 3

Algoritmos de optimización global con restricciones

Como se introdujo en el Capítulo 1, resolver un problema de optimización global constituye una tarea compleja dentro de la teoría general de optimización. Si bien los métodos de optimización local que abordamos hasta el momento son estrategias bastantes usadas en la práctica, por sus propiedades teóricas y algorítmicas, existen otras situaciones donde disponer de algoritmos globales es necesario. Muchos de estos problemas surgen de aplicaciones de la vida real asociadas a las finanzas, problemas de asignación y ubicación, investigación operativa, estadística, problemas de redes y transporte, diseño molecular y nuclear, entre otras. Para estas situaciones actualmente existen algoritmos determinísticos y heurísticos que encuentran minimizadores globales. En esta tesis, trabajaremos con los primeros, también conocidos como algoritmos lipschitzianos. Para presentar las nuevas contribuciones, necesitamos primero presentar algunos desarrollos preliminares que son base de nuestra investigación.

3.1. Preliminares: Algoritmos de optimización global lipschitzianos

3.1.1. Algoritmo DIRECT irrestricto

En 1993, D. Jones, C. Perttunen y B. Stuckman [24] presentaron un enfoque innovador para resolver problemas de optimización global de funciones multivariadas sujetas a restricciones de cotas. En este trabajo, el objetivo principal de los autores fue proponer la resolución de este tipo de problemas enmarcados dentro del enfoque lipschitziano pero eliminando la necesidad de obtener el valor exacto de la constante de Lipschitz. La idea de fondo consistió en mejorar los tres principales inconvenientes que presentan los métodos lipschitzianos en la práctica: la especificación de la constante de Lipschitz, la velocidad de convergencia y la complejidad computacional en altas dimensiones.

El algoritmo presentado por Jones et al. [24] fue denominado DIRECT por dos razones: la primera porque captura el procedimiento de búsqueda directa implementado y la segunda porque cada paso del algoritmo implica una división de rectángulos (deducido de su acrónimo en inglés Dividing RECTangles).

3.1.1.1. DIRECT en una dimensión

Los autores comenzaron considerando una versión simplificada del problema que consiste en encontrar un mínimo global de una función f sobre un intervalo cerrado $[l, u]$. Para este caso,

los algoritmos bajo la filosofía lipschitziana asumen que existe una cota para la tasa de cambio de la función sobre ese intervalo, es decir existe una constante positiva L tal que

$$|f(x) - f(y)| \leq L |x - y|, \forall x, y \in [l, u]. \quad (3.1)$$

En 1972, Shubert [45] trabajó con este problema proponiendo un método para resolverlo. Dada una función f definida sobre el intervalo $[l, u]$, si el punto y en (3.1) es reemplazado por los extremos l y u respectivamente, y con algunos cálculos sencillos adicionales, se obtienen las siguientes ecuaciones:

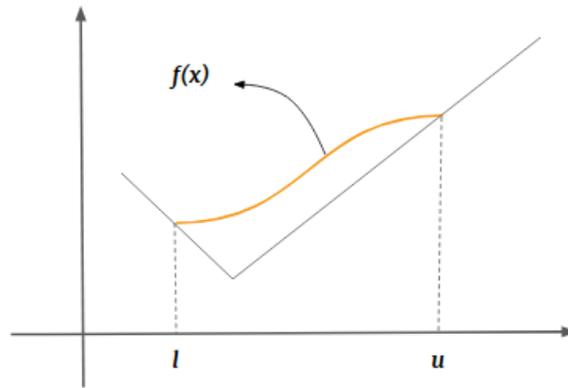
$$\begin{aligned} f(x) &\geq f(l) - L(x - l) \\ f(x) &\geq f(u) + L(x - u), \end{aligned}$$

las cuales definen las dos rectas que se encuentran por debajo de la función f en la Figura 3.1. Observando la Figura 3.1 se deduce casi directamente que el valor mínimo que puede tomar la función en el intervalo $[l, u]$ a lo sumo se alcanzará en la intersección de ambas rectas (vértice). Dado que la información de las rectas se supone conocida, la coordenada de este punto y el valor de la función f en él pueden ser obtenidos con las siguientes fórmulas:

$$x_{\text{vert}} = \frac{l + u}{2} + \frac{f(l) - f(u)}{2L} \quad (3.2)$$

$$f_{\text{vert}} = \frac{f(l) + f(u)}{2} - L(u - l) \quad (3.3)$$

Figura 3.1: Rectas obtenidas al reemplazar l y u en la desigualdad (3.1)



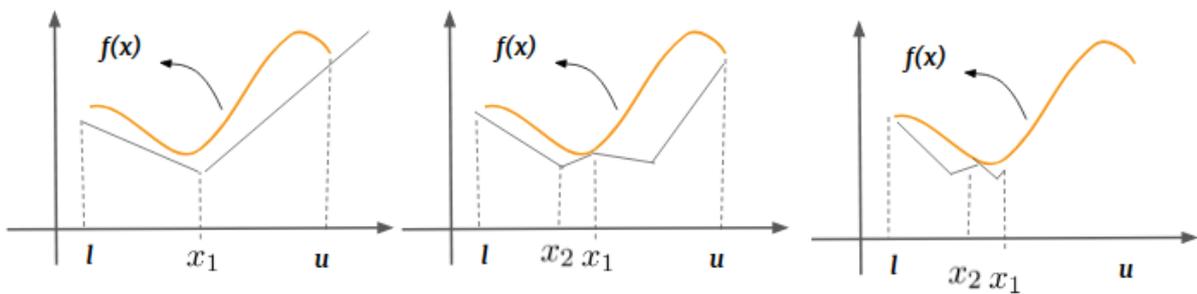
A partir de esta idea, el método propuesto por Shubert consiste en dividir el espacio (en este caso, el intervalo $[l, u]$) utilizando como punto de corte la definición de x_{vert} dada por (3.2). Esto significa que en la primera iteración el intervalo original $[l, u]$ es subdividido en dos: $[l, x_1]$ y $[x_1, u]$ donde x_1 tiene la forma de (3.2). A continuación, cada uno de estos subintervalos es subdividido a su vez definiendo un nuevo punto de corte x_{vert} en base a los extremos, l y x_1 o x_1 y u según corresponda. El subintervalo cuyo punto de corte posea un valor funcional más chico será el elegido para continuar las iteraciones (mientras que el otro será descartado). Así se definen sucesivamente las iteraciones del algoritmo de Shubert hasta que el valor funcional del punto de corte esté lo suficientemente cerca (de acuerdo a una tolerancia predefinida) de la solución del problema de minimización.

El algoritmo de Shubert tiene dos aspectos sobresalientes. El primero relacionado con las iteraciones: la función f termina siendo aproximada por debajo por una función lineal continua

a trozos (Figura 3.2). El segundo punto, y el cual sienta las bases para los siguientes métodos que desarrollaremos en esta tesis, es que la expresión f_{vert} en la ecuación (3.3) implica tanto una **búsqueda local** como una **búsqueda global** sobre el espacio de minimización.

La *búsqueda local* se encuentra presente en el primer término de la expresión (3.3): mientras más bajos los valores funcionales de los extremos (en este caso l y u), más bajo será el valor de $\frac{f(l)+f(u)}{2}$ y esto implica un movimiento hacia un intervalo o conjunto de puntos cercanos con valores funcionales menores (es decir, minimización combinada con *búsqueda local*). Por otra parte, mientras mayor es la amplitud del intervalo, mayor será $u - l$ en el segundo término de (3.3). Por lo tanto, la exploración de puntos se extenderá a una región más amplia (*búsqueda global*). La constante de Lipschitz funciona, en este caso, como un peso relativo entre estos dos tipos de búsqueda. Por ejemplo, mientras mayor es su valor, más peso tendrá la búsqueda global.

Figura 3.2: Visualización del procedimiento iterativo del algoritmo de Shubert



A pesar de que la propuesta de Shubert parecería ser bastante intuitiva y simple de implementar, la misma posee dos grandes desventajas. La primera de ellas se relaciona con la constante de Lipschitz. Por definición, esta constante es una cota superior de la tasa de cambio de la función y por lo tanto, en general, su valor es alto. Es decir, la búsqueda local mencionada en el párrafo anterior sería casi inexistente en la práctica. En segundo lugar, la extensión del algoritmo de Shubert a problemas n -dimensionales se vuelve computacionalmente complejo dado que requiere evaluar los puntos iniciales y los puntos de cortes en dimensiones cada vez más grandes. Esto se traduce a evaluar la función en 2^n puntos sobre un espacio n -dimensional.

Estas desventajas presentes en el algoritmo de Shubert fueron el puntapié inicial del algoritmo DIRECT propuesto por Jones et al. en [24]. Los autores propusieron una versión modificada del algoritmo de Shubert donde los intervalos son evaluados solo en sus centros, en lugar de los extremos l y u . A su vez, estos intervalos son seleccionados en base a cumplir la definición de *intervalo potencialmente óptimo* la cual, en forma simplificada, implica un barrido sobre diferentes valores de la constante de Lipschitz. Desglosando la idea detrás de la propuesta de [24], reemplazando y por c (el centro de intervalo $[l, u]$) en (3.1) y realizando algunos cálculos simples, se deduce que la función f debe satisfacer las siguientes ecuaciones

$$\begin{aligned} f(x) &\geq f(c) - L(x - c), & x &\leq c \\ f(x) &\geq f(c) + L(x - c), & x &\geq c. \end{aligned}$$

Haciendo el análogo con la Figura 3.1, las ecuaciones anteriores implican que el gráfico de f debe estar siempre por encima de dos rectas con pendientes L y $-L$ respectivamente, cuya intersección tiene la forma de una V invertida. Teniendo en cuenta esto, el valor mínimo que

podría llegar a tomar f ocurriría sobre alguno de los extremos l y u y estaría dado por

$$f(c) - L \left(\frac{u-l}{2} \right).$$

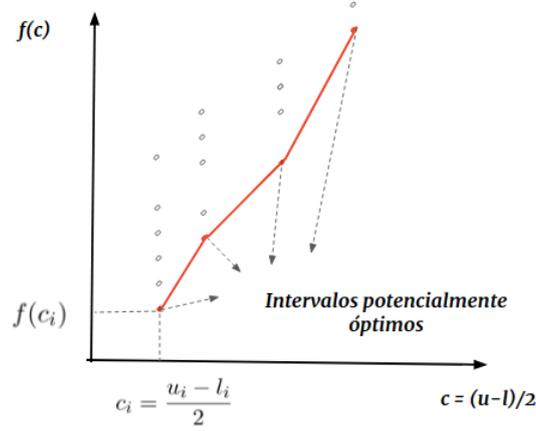
Teniendo identificado el punto central del intervalo $[l, u]$, los autores proponen dividir el intervalo en tercios de manera de asegurar que cada uno de estos tercios sean evaluados en sus centros respectivos en las posteriores iteraciones. En otras palabras, el centro del intervalo $[l, u]$ se convierte en el centro de un intervalo más pequeño, el segundo tercio de $[l, u]$. Luego, a partir de estos centros y en vistas a moverse hacia regiones con valores funcionales menores, la propuesta es elegir aquellos subintervalos que sean potencialmente óptimos y trabajar sobre este subconjunto. Ésta es precisamente una de las claves del algoritmo DIRECT y de todos los algoritmos que se derivaron posteriormente de él.

La idea de Jones et al. en [24] para seleccionar este conjunto de subintervalos se basa en una premisa ya mencionada: la selección debe ser tal que la búsqueda local y la búsqueda global sean lo más ponderadas posible. Partiendo de este objetivo, los autores proponen que la mejor manera de conseguirlo es seleccionando los puntos $(c_i, f(c_i))$ con $c_i = \frac{u_i - l_i}{2}$ pertenecientes a la *cápsula convexa inferior* de la nube total de puntos dada por

$$\left(\frac{u-l}{2}, f \left(\frac{u-l}{2} \right) \right) = (c, f(c)), \quad (3.4)$$

ubicados en el plano coordenado xy , con $x = c$ e $y = f(c)$. De acuerdo a los autores esta selección podría realizarse de forma simple utilizando un algoritmo pertinente como el algoritmo de escaneo de Graham. Incluso sostienen que realizando un reacomodamiento adecuado de los pares de puntos, el orden de las operaciones puede reducirse al número de longitudes de intervalo distintas.

Ante este escenario surge un interrogante importante y es precisamente entender cuál es la relación entre la búsqueda global-local y la cápsula convexa inferior en la nube de puntos. En primer lugar, el plano coordenado representa tanto la búsqueda global como la búsqueda local. La primera coordenada, c , se asocia a la cantidad de territorio sin explorar (recordemos que estamos mirando la distancia del centro del intervalo a cada uno de los extremos). La segunda coordenada por su parte, $f(c)$, captura el comportamiento de los valores funcionales en un entorno local. Partiendo de esta idea, Jones et al. observan que si se trazan rectas con diferentes pendientes positivas y se mueve cada una de ellas hacia arriba hasta intersectar el primer punto de la nube de puntos, se termina obteniendo en cada caso un punto óptimo. La unión de todos ellos forman precisamente la cápsula convexa inferior como muestra la Figura 3.3. Lo interesante de este procedimiento es que estas pendientes no son arbitrarias sino que intentan representar todos los posibles valores que podría tomar la constante de Lipschitz, donde valores más grandes harán mayor énfasis en la búsqueda global y viceversa. Así, utilizando esta técnica se lograría el balance deseado.

Figura 3.3: Interpretación gráfica de la selección de intervalos potencialmente óptimos

A continuación se presenta la definición formal de intervalos potencialmente óptimos para el problema unidimensional.

Definición 3.1.1. *Supóngase que el intervalo $[l, u]$ se ha particionado en m subintervalos $[l_i, u_i]$ con $i = 1, 2, \dots, m$. Sea c_i el punto central del i -ésimo subintervalo, $\varepsilon > 0$ una constante positiva y f_{\min} el mejor valor funcional obtenido hasta el momento. El **intervalo j se dice potencialmente óptimo** si existe alguna tasa de cambio $\tilde{L} > 0$ tal que*

$$f(c_j) - \tilde{L} \left(\frac{u_i - l_i}{2} \right) \leq f(c_i) - \tilde{L} \left(\frac{u_i - l_i}{2} \right), \quad \forall i = 1, \dots, m \quad (3.5)$$

$$f(c_j) - \tilde{L} \left(\frac{u_i - l_i}{2} \right) \leq f_{\min} - \varepsilon |f_{\min}| \quad (3.6)$$

Coloquialmente, la Definición 3.1.1 implica que el intervalo debe encontrarse dentro de la cápsula convexa inferior (desigualdad (3.5)) y que la cota inferior para el intervalo, basada en la tasa de cambio \tilde{L} , excede a la mejor solución por una cantidad prefijada (desigualdad (3.6)). La Figura 3.3 proporciona una forma visualizar de interpretar la Definición 3.1.1.

3.1.1.2. DIRECT en dimensiones mayores que 1

Haciendo la extrapolación del algoritmo DIRECT a n dimensiones, con $n > 1$, el intervalo inicial $[l, u]$ se transforma en un hiperrectángulo de dimensión n . Dado que todo hiperrectángulo de este tipo puede llevarse a un cubo unitario en n dimensiones, mediante una transformación lineal adecuada, trabajar con uno o el otro resulta indistinto. Sin embargo, el trabajo con cubos permite capturar ciertas nociones geométricas (por ejemplo en cuanto a longitudes) desde una perspectiva bastante natural y simple. Por estas razones, y sin pérdida de generalidad, el análisis de DIRECT considera como región de partida el cubo unitario, donde cada variable se encuentra acotada inferiormente por 0 y superiormente por 1.

Habiendo dicho esto y teniendo en cuenta el procedimiento unidimensional descrito previamente, la partición del cubo unitario daría lugar a nuevos hiperrectángulos que serían muestreados a su vez sobre sus respectivos centros. Precisamente en este punto se encuentra el principal “inconveniente” de pensar el algoritmo DIRECT en más dimensiones dado que no resulta trivial deducir la manera más óptima de particionar un hiperrectángulo bajo las ideas desarrolladas hasta el momento.

Para analizar este punto, los autores partieron del caso 2-dimensional y plantearon siguiente interrogante: “¿Cuáles son las opciones posibles de particionar un cuadrado de tal manera que

los rectángulos resultantes contengan un punto muestreado en su centro?." Para pensar esta respuesta, debemos recurrir a la intuición y a la geometría: un cuadrado se puede partir en tercios a lo largo de la coordenada x y luego a lo de la coordenada y , o viceversa. Esto significa que inicialmente se consideran puntos de la forma $c \pm \delta e_i$, con δ representando un tercio del lado del cuadrado y e_i los vectores canónicos (es decir, 4 puntos en total para el caso 2-dimensional).

Este análisis en dos dimensiones genera una nueva pregunta acerca de cuál de los dos caminos posibles debería considerarse. Lo más adecuado siempre es comenzar por la dirección con el valor funcional mínimo, es decir seleccionar la dirección i tal que w sea mínimo, con $w_i = \min\{f(c + \delta e_i), f(c - \delta e_i)\}$. El paso siguiente es subdividir el rectángulo que contiene el centro en tercios a lo largo de la dirección con el menor valor funcional. Un detalle importante a mencionar es que a lo largo de las particiones las regiones rectangulares apreciarán con mayor frecuencia, en estos casos el procedimiento de partición sólo debe considerar las direcciones más largas en longitud. Esta decisión no es arbitraria sino que está relacionada con el hecho de asegurar que la partición se realice en una dirección a la vez y tiene implicación directa en los resultados de convergencia propuestos por los autores.

A continuación y en base al análisis anterior, se presenta la descripción formal de los pasos de la división de hiperrectángulos en dimensiones mayores que 1.

Algoritmo 3.1: Procedimiento de división de hiperrectángulos - Algoritmo DIRECT multivariado

Entrada: Hiperrectángulo a ser particionado.

Salida: Hiperrectángulos resultantes de la partición.

- 1 Identificar el conjunto I de dimensiones con la máxima longitud de lados.
- 2 Definir δ igual a un tercio de esa máxima longitud anterior.
- 3 Obtener el valor de la función en los puntos $c \pm \delta e_i$, $\forall i \in I$, donde c es el centro del rectángulo y e_i es el i -ésimo vector canónico.
- 4 Dividir el hiperrectángulo que contiene c en tercios a lo largo de las dimensiones en I , comenzando con la dimensión con el valor más pequeño w_i , es decir

$$w_i = \min\{f(c + \delta e_i), f(c - \delta e_i)\}$$

y continuando con la dimensión con el valor más alto w_i .

Dado que el valor funcional en cada uno de los centros de los nuevos hiperrectángulos es conocido así como la distancia de estos centros a los vértices, el procedimiento de selección de hiperrectángulos potencialmente óptimos es completamente análogo al caso unidimensional presentado previamente (pudiéndose usar nuevamente el algoritmo de Graham). Desde un punto de vista más formal, en [24] los autores dan la siguiente definición.

Definición 3.1.2. *Supóngase que el hipercubo unitario se ha particionado en m hiperrectángulos. Sea c_i el punto central del i -ésimo hiperrectángulo y sea d_i la distancia desde el punto central a los vértices. Sean además $\varepsilon > 0$ una constante positiva y f_{\min} el mejor valor funcional obtenido hasta el momento. El hiperrectángulo j se dice potencialmente óptimo si existe algún $\tilde{L} > 0$ tal que*

$$\begin{aligned} f(c_j) - \tilde{L}d_j &\leq f(c_i) - \tilde{L}d_i, \text{ para todo } i = 1, \dots, m \\ f(c_j) - \tilde{L}d_j &\leq f_{\min} - \varepsilon|f_{\min}| \end{aligned}$$

Finalmente se presenta a continuación el algoritmo DIRECT multivariado junto con su resultado de convergencia.

Algoritmo 3.2: Algoritmo DIRECT multivariado**Entrada:** $f : \mathbb{R}^n \rightarrow \mathbb{R}$ Función a minimizar. $l, u \in \mathbb{R}^n$: Restricciones sobre las variables del problema de minimización. $FE_{\text{máx}}$: Máximo número de evaluaciones funcionales permitido. $K_{\text{máx}}$: Máximo número de iteraciones permitido.**Salida:** $f_{\text{mín}}$: Valor funcional mínimo encontrado. $x_{\text{mín}}$: Punto donde se da el valor funcional mínimo.

```

1  $D \leftarrow [l, u] = \{x \in \mathbb{R}^n : l_i \leq x_i \leq u_i, i = 1, 2, \dots, n\}$ 
2  $\bar{D} = [0, 1]^n \leftarrow D$  // Normalizar el espacio de búsqueda.
3 Definir  $c_1$  el punto central del hipercubo  $\bar{D}$  y  $f(c_1)$  su valor funcional.
4  $f_{\text{mín}} \leftarrow f(c_1)$ 
5  $FE \leftarrow 1$ 
6  $k \leftarrow 0$ 
7 mientras  $k \leq K_{\text{máx}}$  y  $FE \leq fe_{\text{máx}}$  hacer
8   Identificar el conjunto  $\mathcal{P}$  de hiperrectángulos potencialmente óptimos.
   /* Selección de hiperrectángulos */
9   para  $j \in \mathcal{P}$  hacer
10    Aplicar el Algoritmo 3.1 para obtener la partición del hiperrectángulo  $j$ 
11    Actualizar  $f_{\text{mín}}$  y  $x_{\text{mín}}$ 
12     $FE \leftarrow FE + \Delta FE$  //  $\Delta FE$  es el número de nuevos puntos sampleados.
13     $\mathcal{P} \leftarrow \mathcal{P} - \{j\}$ 
14   $k \leftarrow k + 1$ 
15 devolver  $f_{\text{mín}}, x_{\text{mín}}$ 

```

^aEn [24] el único criterio de parada dado por los autores son las iteraciones, no obstante para ser consistentes con lo desarrollado en este trabajo se agregaron también las evaluaciones funcionales como criterio de parada.

Proposición 3.1.1 (Resultado de densidad). *El conjunto de puntos muestreados por el algoritmo DIRECT multivariado forma un subconjunto denso dentro del hipercubo unitario (ó hiperrectángulo inicial), es decir, dado cualesquiera x y $\delta > 0$ en el hipercubo unitario, DIRECT eventualmente muestreará un punto a a una distancia δ de x .*

Demostración. Ver demostración en [24]. □

Teorema 3.1.2 (Resultado de convergencia). *Si f es una función continua o continua en al menos una vecindad de un óptimo global x^* , entonces dado cualquier $\varepsilon > 0$ existe una iteración k_ε y un hiperrectángulo indexado por i^{k_ε} tal que*

$$\|c^{i^{k_\varepsilon}} - x^*\| \leq \varepsilon \quad (3.7)$$

Demostración. Ver demostración en [24]. □

3.1.2. Algoritmo BIRECT irrestricto

En el año 2016, Remigijus Paulavičius, Lakhdar Chiter y Julius Žilinskas [43] propusieron una nueva variante de los algoritmos tipo DIRECT para resolver problemas de optimización global. Los autores partieron del clásico problema de optimización global con cotas en las variables

$$\underset{x \in D}{\text{minimizar}} \quad f(x), \quad (3.8)$$

donde $D = [l, u] = \{x \in \mathbb{R}^n : l_i \leq x_i \leq u_i \text{ con } i = 1, 2, \dots, n\}$ define la región factible del problema y representa un hiperrectángulo n -dimensional. Además, $l, u \in \mathbb{R}^n$ y f es una función Lipschitz continua.

Basándose en la técnica de partición del espacio del algoritmo DIRECT (DIviding RECTangles) descrita en la sección anterior, Paulavičius et al. [43] propusieron realizar una bisección del espacio en lugar de la clásica trisección. Junto con la bisección también introdujeron una técnica de muestreo sobre dos puntos de la diagonal de un hiperrectángulo (en lugar de evaluar sólo el centro) permitiendo reutilizar puntos ya evaluados en futuras bisecciones. Similar a su antecesor, el algoritmo fue denominado BIRECT por su acrónimo en inglés BIsecting RECTangles.

A continuación se presentan algunos resultados e ideas principales presentes en [43].

3.1.2.1. Las bases de BIRECT

Uno de los primeros puntos que los autores remarcan es que el algoritmo BIRECT trabaja siempre sobre una región factible escalada \bar{D} definida por

$$\bar{D} = [0, 1]^n = \{x \in \mathbb{R}^n : 0 \leq x_i \leq 1 \text{ con } i = 1, 2, \dots, n\}$$

y vuelve al espacio original D sólo cuando necesita evaluar la función objetivo f . Para hacer esto último, se vale de la inversa de la transformación lineal

$$T : D \rightarrow \bar{D}, T(x) = A(x - l) \quad (3.9)$$

donde la matriz A se encuentra definida por $A = \text{diag}\left(\frac{1}{u_1 - l_1}, \frac{1}{u_2 - l_2}, \dots, \frac{1}{u_n - l_n}\right)$. Es decir,

$$T^{-1} : \bar{D} \rightarrow D, \quad T^{-1}(x) = A^{-1}x + l$$

con $A^{-1} = \text{diag}(u_1 - l_1, u_2 - l_2, \dots, u_n - l_n)$. Notar que $l \xrightarrow{T} (0, 0, \dots, 0)$ y $u \xrightarrow{T} (1, 1, \dots, 1)$.

Básicamente se podría definir el algoritmo BIRECT a través de sus dos procedimientos principales: *muestreo de la diagonal de los hiperrectángulos y partición del espacio utilizando una técnica de bisección*.

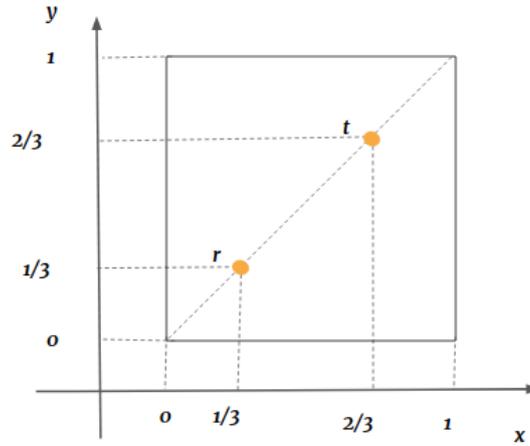
Cada iteración del algoritmo involucra la evaluación de f en dos puntos distintos: “*lower*” o “*inferior*”

$$r = (r_1, r_2, \dots, r_n) = \left(\frac{1}{3}, \frac{1}{3}, \dots, \frac{1}{3}\right)^T$$

y “*upper*” o “*superior*”

$$t = (t_1, t_2, \dots, t_n) = \left(\frac{2}{3}, \frac{2}{3}, \dots, \frac{2}{3}\right)^T$$

que se encuentran sobre la diagonal principal del hipercono \bar{D} y equidistantes entre sí y los vértices de la diagonal como muestra la Figura 3.4. Es importante remarcar que estos nombres son sólo representativos y de ninguna manera se asocian a una posición *fija* dentro de la diagonal del hipercono o hiperrectángulo, la idea central es evaluar dos puntos diferentes sobre la diagonal de un hipercono ubicados a una distancia predeterminada.

Figura 3.4: Visualización de los puntos muestreados por BIRECT en 2 dimensiones

3.1.2.2. Procedimiento

Similar a todo algoritmo de la filosofía DIRECT, BIRECT elige hiperrectángulos “potencialmente óptimos” iteración a iteración con el fin de ir mejorando la aproximación a la solución del problema (3.8). Partiendo de una iteración inicial donde \bar{D} es el único hiperrectángulo potencialmente óptimo, a lo largo de las siguientes iteraciones el algoritmo BIRECT va considerando las diferentes estimaciones de f sobre cada uno de los hiperrectángulos \bar{D}^i involucrados en la actual partición. Así, una partición de la iteración k está definida como

$$\mathcal{P}_k = \{\bar{D}^i : i \in \mathbb{I}_k\}$$

donde

$$\bar{D}^i = [l^i, u^i] = \{x \in \mathbb{R}^n : 0 \leq l_j^i \leq x_j \leq u_j^i \leq 1 \text{ con } j = 1, 2, \dots, n, \text{ para todo } i \in \mathbb{I}_k\},$$

siendo \mathbb{I}_k el conjunto de índices que identifican la actual partición \mathcal{P}_k . La siguiente partición \mathcal{P}_{k+1} es obtenida bisectando el conjunto de hiperrectángulos potencialmente óptimos a partir de la actual partición de acuerdo a la siguiente definición.

Definición 3.1.3. Sean r^i y t^i dos puntos diagonales muestreados y δ_i una medida (distancia, tamaño) del hiperrectángulo \bar{D}^i definida por

$$\delta_i = \frac{2}{3} \|u^i - l^i\|. \quad (3.10)$$

Sean también $\varepsilon > 0$ una constante positiva y f_{\min} la mejor aproximación a la solución conocida del problema (3.8). Se dice que un hiperrectángulo \bar{D}^j , $j \in \mathbb{I}_k$ es **potencialmente óptimo** si existe alguna tasa de cambio (Lipschitz) constante $\tilde{L} > 0$ tal que

$$\min\{f(r^j), f(t^j)\} - \tilde{L}\delta_j \leq \min\{f(r^i), f(t^i)\} - \tilde{L}\delta_i, \text{ para todo } i \in \mathbb{I}_k \quad (3.11)$$

$$\min\{f(r^j), f(t^j)\} - \tilde{L}\delta_j \leq f_{\min} - \varepsilon |f_{\min}|. \quad (3.12)$$

Esta definición establece que el hiperrectángulo \bar{D}^i es potencialmente óptimo si la cota inferior de Lipschitz para la función objetivo calculada en el lado izquierdo de (3.11) es la más pequeña con alguna constante positiva \tilde{L} entre los hiperrectángulos de la actual partición \mathcal{P}_k . De acuerdo a los autores, la desigualdad (3.12) controla un énfasis excesivo en la búsqueda local.

Teniendo presente la definición anterior se describe a continuación el procedimiento de bisección. Partiendo de un hiperrectángulo \bar{D}^i identificado como potencialmente óptimo, $i \in \mathbb{I}_k$, el mismo es bisectado en dos nuevos hiperrectángulos utilizando la coordenada más larga en longitud como punto de referencia (al igual que el algoritmo DIRECT). Esta coordenada recibe el nombre de *variable de ramificación* (branching variable en inglés) y se denota como x_{br} con $br \in \{1, 2, \dots, n\}$. Matemáticamente hablando, br se define como:

$$br = \min \left\{ \arg \max_{1 \leq j \leq n} \{d_j^i = |u_j^i - l_j^i|\} \right\} \quad (3.13)$$

donde d_j^i denota la longitud de la diferencia entre l y u sobre la coordenada j en la iteración i . De acuerdo a este procedimiento cada hiperrectángulo es bisectado solo una vez. En (3.13) podemos ver que en caso de que sea posible elegir más de una variable de ramificación, el criterio es tomar aquella con el índice más bajo.

El siguiente resultado (junto con su demostración) que generaliza el procedimiento de división y muestreo del algoritmo BIRECT es presentado por los autores en [43].

Teorema 3.1.3. *Los puntos r^i y t^i del hiperrectángulo $\bar{D}^i = [l_1^i, u_1^i] \times \dots \times [l_n^i, u_n^i]$ se encuentran exactamente sobre las diagonales de los hiperrectángulos \bar{D}^{i+1} y \bar{D}^{i+2} obtenidos cuando \bar{D}^i es bisectado, y su lugar en las diagonales de los dos nuevos hiperrectángulos está exactamente a una distancia de una tercera parte de uno de los vértices de las respectivas diagonales.*

Demostración. Ver demostración en [43]. □

3.1.2.3. Algoritmo y convergencia

Similar a los algoritmos tipo DIRECT, el algoritmo BIRECT tiene una convergencia basada en argumentos de densidad. A continuación se enuncia el resultado de convergencia presentado en [43] seguido por su respectivo algoritmo.

Teorema 3.1.4 (Resultado de convergencia). *Para cada punto de mínimo global $x^* \in D$ de la función f y cualquier $\varepsilon > 0$ existe un número de iteración k_ε y un índice de hiperrectángulo $i^{k_\varepsilon} \in \mathbb{I}_{k_\varepsilon}$ tal que*

$$\max \left\{ \|r^{i^{k_\varepsilon}} - x^*\|, \|t^{i^{k_\varepsilon}} - x^*\| \right\} \leq \varepsilon$$

Demostración. Ver demostración en [43]. □

Algoritmo 3.3: Algoritmo BIRECT irrestricto**Entrada:** $f : \mathbb{R}^n \rightarrow \mathbb{R}$ Función a minimizar. $l, u \in \mathbb{R}^n$: Restricciones sobre las variables del problema de minimización. $FE_{\text{máx}}$: Máximo número de evaluaciones funcionales permitido. $K_{\text{máx}}$: Máximo número de iteraciones permitido.**Salida:** $f_{\text{mín}}$: Valor funcional mínimo encontrado. $x_{\text{mín}}$: Punto donde se da el valor funcional mínimo.

```

1 Normalizar el espacio de búsqueda  $D$  para que sea un hipercubo unitario  $\bar{D}$ 
  /* Inicialización */
2  $k \leftarrow 1$ 
3  $\mathbb{I}_k \leftarrow \{1\}$ 
4  $r^1 \leftarrow (\frac{1}{3}, \frac{1}{3}, \dots, \frac{1}{3})$ 
5  $t^1 \leftarrow (\frac{2}{3}, \frac{2}{3}, \dots, \frac{2}{3})$ 
6  $FE \leftarrow 2$ 
7  $f_{\text{mín}} \leftarrow \min\{f(r^1), f(t^1)\}$ 
8  $x_{\text{mín}} \leftarrow \arg \min_{x \in \{r^1, t^1\}} f(x)$ 
  /* Proceso iterativo */
9 mientras  $FE < FE_{\text{máx}}$  y  $k < K_{\text{máx}}$  hacer
10   Identificar el conjunto de índices  $\mathbb{P}_k \subset \mathbb{I}_k$  de hiperrectángulos potencialmente
      óptimos.
11    $\mathbb{I}_k \leftarrow \mathbb{I}_k \setminus \mathbb{P}_k$ 
12   para  $i \in \mathbb{P}_k$  hacer
13     Seleccionar la variable de ramificación siguiendo la fórmula (3.13).
      /* Copiar los viejos puntos muestrales */
14      $r^{FE+2} \leftarrow r^i, r^{FE+1} \leftarrow r^i$ 
15      $t^{FE+2} \leftarrow t^i, t^{FE+1} \leftarrow t^i$ 
      /* Actualización de los nuevos puntos */
16     si  $r_j^i < t_j^i$  entonces
17        $r_{\text{br}}^{FE+2} \leftarrow r_{\text{br}}^i + \frac{d_{\text{br}}^i}{2}$ 
18        $t_{\text{br}}^{FE+1} \leftarrow t_{\text{br}}^i - \frac{d_{\text{br}}^i}{2}$ 
19       Evaluar  $f$  en los nuevos puntos  $f(r^{FE+2})$  y  $f(t^{FE+1})$ 
20     en otro caso
21        $r_{\text{br}}^{FE+1} \leftarrow r_{\text{br}}^i - \frac{d_{\text{br}}^i}{2}$ 
22        $t_{\text{br}}^{FE+2} \leftarrow t_{\text{br}}^i + \frac{d_{\text{br}}^i}{2}$ 
23       Evaluar  $f$  en los nuevos puntos  $f(r^{FE+1})$  y  $f(t^{FE+2})$ 
24      $f_{\text{mín}}^{FE+1} \leftarrow \min\{f(r^{FE+1}), f(t^{FE+1})\}$ 
25      $f_{\text{mín}}^{FE+2} \leftarrow \min\{f(r^{FE+2}), f(t^{FE+2})\}$ 
26      $f_{\text{mín}} \leftarrow \min\{f_{\text{mín}}, f_{\text{mín}}^{FE+1}, f_{\text{mín}}^{FE+2}\}$ 
27      $x_{\text{mín}} \leftarrow \arg \min_{x \in \{x_{\text{mín}}, x_{\text{mín}}^{FE+1}, x_{\text{mín}}^{FE+2}\}} f(x)$ 
28     Bisectar  $\bar{D}^i$  en dos nuevos hiperrectángulos  $\bar{D}^{FE+1}, \bar{D}^{FE+2}$ 
29     Actualizar  $\delta_{FE+1}, \delta_{FE+2}$ 
30      $k \leftarrow k + 1$ 
31      $\mathbb{I}_k \leftarrow \mathbb{I}_k \cup \{FE + 1, FE + 2\}$ 
32 devolver  $f_{\text{mín}}, x_{\text{mín}}$ 

```

3.1.3. Algoritmo DIRECT con restricciones generales.

En el año 2001, Donald R. Jones [23] propuso una extensión del algoritmo DIRECT [24] publicado en el año 1993, la cual se enfocó en la resolución de problemas de optimización global con restricciones de desigualdad (hasta el momento DIRECT había sido pensando para trabajar sólo con restricciones de cotas). Con la introducción de este cambio, se consideró el problema ¹

$$\begin{aligned}
 & \underset{x \in \mathbb{R}^n}{\text{minimizar}} && f(x_1, x_2, \dots, x_n) \\
 & \text{sujeto a} && g_1(x_1, x_2, \dots, x_n) \leq 0 \\
 & && g_2(x_1, x_2, \dots, x_n) \leq 0 \\
 & && \vdots \\
 & && g_m(x_1, x_2, \dots, x_n) \leq 0 \\
 & && l_i \leq x_i \leq u_i
 \end{aligned} \tag{3.14}$$

con $i = 1, 2, \dots, n$ y g_i , con $i = 1, 2, \dots, m$, restricciones generales de desigualdad.

A continuación presentamos los resultados y las ideas más importantes presentes en [23].

3.1.3.1. La función auxiliar

El primer cambio incorporado por el autor consistió en la introducción de una función auxiliar h_r con la idea de reunir en un sólo lugar información tanto de la función objetivo como de las funciones que forman las restricciones generales de desigualdad. Así, la expresión de h_r resultó en

$$h_r(f^*) = \frac{\text{máx}(f_r - f^*, 0) + \sum_{j=1}^m c_j \text{máx}(g_{rj}, 0)}{d_r}, \tag{3.15}$$

donde f^* denota el valor funcional del minimizador global (si bien el mismo es usualmente desconocido, como se verá en breve, este “inconveniente” será sorteado), f_r es el valor en el punto medio del hiperrectángulo r , c_j con $j = 1, 2, \dots, m$ son coeficientes de pesos positivos para las restricciones de desigualdad, g_{rj} es el valor de la restricción j en el punto medio del hiperrectángulo r y d_r es la distancia desde el centro al vértice del hiperrectángulo r .

Para entender la idea detrás de esta expresión es importante analizarla por partes. Por un lado, $\text{máx}(f_r - f^*, 0)$ penalizará cualquier desviación de f_r por encima del óptimo global f^* , es decir, esta parte se centra en la optimalidad del problema. Por otro lado $\sum_{j=1}^m c_j \text{máx}(g_{rj}, 0)$ consiste en una suma ponderada de las violaciones a las restricciones, haciendo énfasis de esta manera en la factibilidad del problema. Es así como en x^* , el punto óptimo global del problema (3.14), la expresión (3.15) alcanzará su valor mínimo 0 y en cualquier otro punto siempre será positiva ya sea por suboptimalidad o infactibilidad. Por otra parte, para que el minimizador global ocurra en el rectángulo r , la función auxiliar debería irse acercando cada vez más a 0 desde su valor positivo en el centro del mismo. La máxima distancia en la cual esto puede ocurrir es precisamente la distancia del centro al vértice d_r . Esta es la razón por la cual se divide por d_r en (3.15), entendiendo este valor como la mínima tasa de cambio a la cual debería someterse la expresión del numerador.

Finalmente, con respecto a lo mencionado en el primer párrafo, Jones propuso un camino alternativo para trabajar con el valor de f^* usualmente desconocido en este tipo de problemas.

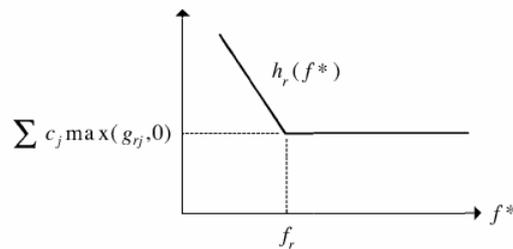
¹La formulación original del problema considera también variables enteras, dado que la temática de este trabajo se enfoca en variables continuas esta faceta se omitirá del análisis.

Teniendo en cuenta que sería interesante seleccionar un conjunto de hiperrectángulos que minimicen h_r , Jones transformó esta idea en la selección de un conjunto de hiperrectángulos que minimizen $h_r(f^*)$ para algún $f^* \leq f_{\min} - \varepsilon$, con f_{\min} conocido.

3.1.3.2. Procedimiento

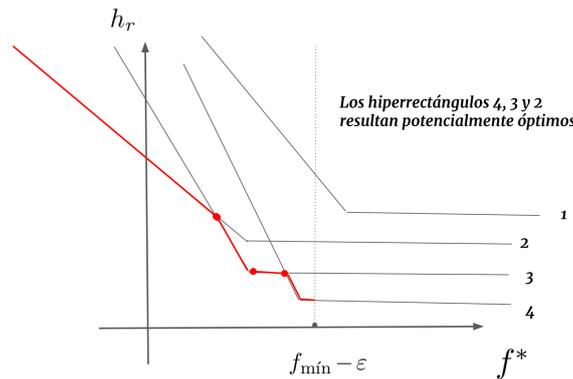
En relación con el punto anterior, y en especial con el último párrafo, el segundo cambio introducido por Jones se relaciona con el procedimiento de selección de hiperrectángulos. En este caso, el diagrama de selección se basa en la relación de h_r como función de f^* (Figura 3.5) para cada uno de los hiperrectángulos considerados en una iteración dada (Figura 3.6).

Figura 3.5: Visualización de la función h_r



Si observamos la Figura 3.5, cuando $f^* \geq f_r$, el término $\max(f_r - f^*, 0)$ de la expresión (3.15) es 0, por lo tanto la función h_r es constante y su valor es precisamente $\frac{\sum_{j=1}^m c_j \max(g_{rj}, 0)}{d_r}$. Contrariamente, cuando $f^* \leq f_r$, el primer término de (3.15) hace que h_r tome valores cada vez mayores. Además, para estos valores de f^* , la pendiente de la recta es precisamente $-\frac{1}{d_r}$.

Figura 3.6: Visualización de h_r para cada hiperrectángulo de una iteración



Retomemos ahora la idea del final de la sección anterior. Dado que se busca encontrar aquellos hiperrectángulos que minimicen h_r para algún $f^* \leq f_{\min} - \varepsilon$, observando la Figura 3.6, estos hiperrectángulos son precisamente aquellos que forman parte de la cápsula convexa inferior de las curvas a la izquierda del punto $f_{\min} - \varepsilon$. En este sentido, el autor señala que el procedimiento para seleccionar cada hiperrectángulo no implica un esfuerzo computacional importante pues es del orden del número total de hiperrectángulos en la partición.

3.1.3.3. Los coeficientes c_j

La idea detrás de la introducción de estos coeficientes resulta bastante interesante. A través de los mismos, Jones incorporó una característica importante al procedimiento descrito hasta el momento: una estimación de las tasas de cambio de la función objetivo y de las restricciones. Siendo x^{mid} el punto medio del rectángulo *padre* y x^{left} y x^{right} los puntos medios de los rectángulos *hijos* a izquierda y a derecha luego del procedimiento de trisección, se definen las siguientes sumas de las tasas de cambio observadas en la función objetivo y en las restricciones como:

$$s_0 = s_0 + \frac{|f(x^{\text{right}}) - f(x^{\text{mid}})|}{\|x^{\text{right}} - x^{\text{mid}}\|} + \frac{|f(x^{\text{left}}) - f(x^{\text{mid}})|}{\|x^{\text{left}} - x^{\text{mid}}\|} \quad (3.16)$$

$$s_j = s_j + \frac{|g_j(x^{\text{right}}) - g_j(x^{\text{mid}})|}{\|x^{\text{right}} - x^{\text{mid}}\|} + \frac{|g_j(x^{\text{left}}) - g_j(x^{\text{mid}})|}{\|x^{\text{left}} - x^{\text{mid}}\|}, j = 1, 2, \dots, m \quad (3.17)$$

De esta manera el promedio de dichas tasas de cambio está dado por $a_0 = \frac{s_0}{N}$ y $a_j = \frac{s_j}{N}$ donde N es el número de tasas de cambio acumuladas en cada una de las sumas. Finalmente, a través del cociente

$$\frac{a_0}{a_j} = \frac{\frac{s_0}{N}}{\frac{s_j}{N}} = \frac{s_0}{s_j}$$

se obtiene una expresión para los coeficientes c_j . Para evitar potenciales divisiones por 0 la expresión se transforma en

$$c_j = \frac{s_0}{\max\{s_j, 10^{-30}\}}. \quad (3.18)$$

Resulta importante recuperar la idea detrás de este cociente, como lo explica muy bien Jones en [23]. Consideremos el caso en el cual el punto medio del hiperrectángulo se encuentra actualmente violando las restricciones, es decir $g_j > 0$, y supongamos conocidas las tasas de cambio de la función objetivo y de la restricción j -ésima, a_0 y a_j respectivamente. Para *deshacernos* de la violación de la restricción, deberíamos movernos una distancia igual a $\frac{g_j}{a_j}$. Si después de realizar esta acción, la función objetivo empeoró en su tasa de cambio promedio, debería empeorar a_0 veces la distancia, es decir, $a_0 \frac{g_j}{a_j} = \frac{a_0}{a_j} g_j$. De esta manera, el cociente $c_j = \frac{a_0}{a_j}$ proporciona una manera de convertir unidades de violación a las restricciones en potenciales incrementos de la función objetivo.

3.1.3.4. Estrategia de desempate en las divisiones

Este punto surge del hecho de que eventualmente un hiperrectángulo podría tener más de un lado *más largo*. Dado que el procedimiento se basa en seleccionar un solo lado a la vez, la estrategia de desempate consiste en llevar un conteo de cuántas veces un hiperrectángulo fue triseccionado a lo largo de la dimensión i . De esta manera, el desempate consiste en favorecer siempre al lado que ha sido dividido menos veces, marcando así un equilibrio en las divisiones sobre diferentes dimensiones. En aquellas situaciones de “doble empate”, es decir lados que fueran divididos igual cantidad de veces, la regla es seleccionar el lado con la dimensión menor (algo similar a lo que ocurre en la variable de ramificación del algoritmo BIRECT).

3.1.3.5. La distancia d

Recordemos que el rango inicial de cada variable es medido utilizando un escalamiento de la región de partida que asigna a esta distancia el valor 1. De esta manera, cuando un hiperrectángulo es subdividido, la longitud de sus lados es reducida por el factor $\frac{1}{3}$. Consideremos un hiperrectángulo que ha sido dividido (trisechado) T veces. Si $j = \text{mód}(T, n)$ entonces $T = kn + j$ con $k = \frac{T-j}{n}$. Luego de las primeras kn trisecciones, todos los n lados habrán sido trisecados k veces y por lo tanto tendrán longitud 3^{-k} . Las restantes j trisecciones harán que j de los lados tengan longitud $3^{-(k+1)}$, dejando así $n - j$ lados con longitud 3^{-k} . Realizando algunas cuentas extra se deduce que la distancia d del centro al vértice está dada por

$$d = \frac{3^{-k}}{2} \sqrt{\frac{j}{9} + n - j}.$$

Para profundizar este tema, en la Sección 5 de [24], los autores explican por primera vez cómo obtener esta distancia.

3.1.3.6. Algoritmo y convergencia

El algoritmo DIRECT con restricciones generales hereda las propiedades de convergencia mostradas por su predecesor (Ver Sección 5 en [24]). A medida que f^* va tomando valores cada vez más negativos y moviéndose hacia $-\infty$, DIRECT seleccionará uno de los hiperrectángulos más grandes. Por esta razón y por el hecho de que siempre subdivide uno de los lados, los hiperrectángulos se volverán cada vez más pequeños y los puntos muestreados serán densos en el espacio. Además dado que se asume que las funciones son continuas en una vecindad del optimizador, esto asegura que estaremos dentro de cualquier tolerancia positiva del optimizador después de un número suficientemente grande de iteraciones.

En el Algoritmo 3.4 se muestra de manera detallada el paso a paso del algoritmo donde Ω denota el conjunto factible del problema (3.14).

Algoritmo 3.4: DIRECT con restricciones generales

Entrada:
 $f : \mathbb{R}^n \rightarrow \mathbb{R}$ Función a minimizar.
 $g_i : \mathbb{R}^n \rightarrow \mathbb{R}$ Restricciones generales del problema de minimización.
 $l, u \in \mathbb{R}^n$: Restricciones sobre las variables del problema de minimización.
 $FE_{\text{máx}}$: Máximo número de evaluaciones funcionales permitido.
 $K_{\text{máx}}$: Máximo número de iteraciones permitido.

Salida:
 $f_{\text{mín}}$: Valor funcional mínimo encontrado
 $x_{\text{mín}}$: Punto factible del problema donde se da el valor funcional mínimo

```

/* Inicialización */
1 Normalizar el espacio de búsqueda para que sea un hipercubo unitario.
2 FE ← 0 // Inicialización de las evaluaciones funcionales
3 IsFactible ← False
4 Sea  $\mathbf{c}$  el punto central del hipercubo y  $f(\mathbf{c})$  su valor funcional.
5 FE ← FE + 1
6  $\mathbb{I}^k \leftarrow \{1\}$ 
7 si  $\mathbf{c} \in \Omega$  entonces
  8    $f_{\text{mín}} \leftarrow f(\mathbf{c})$ 
      $x_{\text{mín}} \leftarrow \mathbf{c}$ 
     IsFactible ← True
9 para  $j \leftarrow 0$  a  $m$  hacer // Inicialización de  $s$ 
10    $s_j \leftarrow 0$ 
11 para  $i \leftarrow 0$  a  $n$  hacer // Inicialización de  $t$ 
12    $t_i \leftarrow 0$ 
13 mientras  $FE < FE_{\text{máx}}$  y  $k < K_{\text{máx}}$  hacer
  /* Selección de hiperrectángulos */
  14 para  $j \leftarrow 1$  a  $m$  hacer // Actualización de  $c$ 
  15    $c_j \leftarrow \frac{s_0}{\max\{s_j, 10^{-30}\}}$ 
  16 si IsFactible = False entonces // Ningún punto factible ha sido encontrado
  17   
$$\mathbb{P}^k \leftarrow \arg \min_{p \in \mathbb{I}^k} \left\{ \frac{\sum_{j=1}^m c_j \max\{g_{\mathbb{I}^p j}, 0\}}{d_{\mathbb{I}^p}} \right\}$$

  18 en otro caso // Existe al menos un punto factible
  19    $\varepsilon \leftarrow \max\{10^{-4}|f_{\text{mín}}|, 10^{-8}\}$ 
  20    $f_{\text{mín}_\varepsilon} \leftarrow f_{\text{mín}} - \varepsilon$ 
  21    $\mathbb{P}^k \leftarrow$  Conjunto de hiperrectángulos que participan en la cápsula convexa inferior de  $h_r$ 
  22    $\mathbb{I}^k \leftarrow \mathbb{I}^k \setminus \{\mathbb{P}^k\}$ 
  23 para  $p \in P_k$  hacer
  24   /* Procedimiento de trisección */
  25   Identificar el conjunto de lados largos del hiperrectángulo  $p$ .
  26   Elegir la dimensión de partición  $i$  usando el lado largo con el valor más pequeño de  $t_i$ .
  27   Trisectar el hiperrectángulo  $p$  a lo largo de la dimensión  $i$ .
  28    $t_i \leftarrow t_i + 1$ 
  29   Samplear el punto medio del hiperrectángulo del tercio izquierdo.
  30   Actualizar las variables  $x_{\text{mín}}, f_{\text{mín}}$  e IsFactible.
  31   FE ← FE + 1.
  32   si  $FE = FE_{\text{máx}}$  entonces
  33     devolver  $f_{\text{mín}}, x_{\text{mín}}$ 
  34   Muestrear el punto medio del hiperrectángulo del tercio derecho.
  35   Repetir los pasos de las líneas 29–32.
  36   Actualizar  $s_0$  y  $s_j$  utilizando las ecuaciones (3.16) y (3.17)
  37   Actualizar  $\mathbb{I}^k$  con los 3 nuevos hiperrectángulos obtenidos de la trisección.
  38    $k \leftarrow k + 1$ 
39 devolver  $f_{\text{mín}}, x_{\text{mín}}$ 

```

Algunas observaciones finales

1. En la primera iteración del algoritmo, todos los coeficientes s_j son iguales 0 (y en consecuencia los c_j también). Es decir, al inicio del proceso estos coeficientes no son particularmente útiles, lo cual tiene sentido dado que el mismo se inicia con un solo hiperrectángulo (todo el espacio de búsqueda). Sin embargo, a medida que las iteraciones progresan, los valores de s_j tienen en cuenta más observaciones, permitiendo así que los valores de c_j sean cada vez más relevantes en el proceso de selección de hiperrectángulos.
2. Para problemas sin restricciones de desigualdad, este algoritmo se reduce a la versión original de DIRECT presentado en la Sección 3.1.1. Así, todo punto evaluado es inmediatamente factible, dejando de lado el concepto de *violación a las restricciones*. La consecuencia directa este punto es que la función auxiliar (3.15) se reduce a $\frac{f_r - f^*}{dr}$, que es precisamente la tasa de cambio de la función cuando se minimiza un problema sin restricciones.
3. Al basarse en un enfoque de partición del espacio, DIRECT requiere que los límites en todas las variables sean razonables. Esto implica que el algoritmo tendrá un desempeño bastante deteriorado si alguna de las variables se mueve, por ejemplo, entre -10^{30} y 10^{30} .
4. La técnica de partición del espacio además es un limitante al momento de trabajar con dimensiones mayores. La cota superior estaría dada, en base a la experiencia práctica, por problemas de hasta dimensión 20.
5. Dado que la formulación del problema no incluye restricciones de igualdad, esta podría considerarse como una desventaja de la propuesta de Jones.
6. Teniendo en cuenta que en general la solución al problema no es conocida, los criterios de parada de algoritmo se reducen mayormente al número de evaluaciones funcionales y el número de iteraciones, siendo estos muchos más débiles que una tolerancia predefinida en el valor funcional.
7. Una gran ventaja de esta propuesta es que DIRECT puede trabajar con diferentes clases de funciones (no suaves, no lineales, multimodales e incluso discontinuas lejos del optimizador global). La presencia del ruido tampoco tendría un gran impacto en la selección de hiperrectángulos.
8. El algoritmo DIRECT es altamente paralelizable lo cual permitiría un mejor aprovechamiento de los recursos computacionales.
9. La manera en la cual DIRECT balancea la búsqueda global con la búsqueda local lo convierte en un algoritmo robusto y fácil de usar.

3.1.4. Algoritmo DIRECT con búsqueda local

En el año 2010, G. Liuzzi, S. Lucidi y V. Piccialli [35] propusieron un nuevo algoritmo basado en el de partición del espacio de DIRECT pero al que le sumaron un procedimiento de minimización local buscando mejorar la performance del algoritmo y la convergencia original dada por DIRECT.

Similar a las propuestas antes mencionadas, los autores partieron del problema de optimización global dado por

$$\underset{x \in D}{\text{minimizar}} \quad f(x) \quad (3.19)$$

con D definido por

$$D = [0, 1]^n = \{x \in \mathbb{R}^n : 0 \leq x_i \leq 1, i = 1, 2, \dots, n\}. \quad (3.20)$$

Como se señaló en las secciones anteriores, cualquier problema de optimización con restricciones de cajas puede ser llevado a una región factible escalada como la presentada en (3.20).

Una de las principales desventajas del algoritmo DIRECT [24] es la necesidad de almacenar en memoria todos los hiperrectángulos generados. A medida que los problemas aumentan en dimensión, esto podría (y en efecto sucede) devenir en un agotamiento de los recursos de memoria disponibles. Entre las consecuencias más *graves* e importantes de este punto es que en la mayoría de los casos el algoritmo no logra explorar una cantidad suficiente de puntos, con lo cual la solución obtenida podría estar bastante lejos de la óptima esperada. En [35], los autores tomaron esta “debilidad” de DIRECT como punto de partida formulando un nuevo algoritmo determinístico “*cuyo principal foco es obtener una buena estimación del óptimo global para problemas a gran escala incluso pagando el precio de incrementar el número de evaluaciones funcionales, tiempo de CPU y explotando, al igual que DIRECT, toda la memoria disponible*” ([35], pág. 354).

El trabajo de Liuzzi et al. [35] contiene dos partes claramente diferenciadas, no obstante siguiendo la línea de esta tesis, solo se introducirá lo relacionado con la primera parte: la incorporación de una búsqueda local dentro de un problema de optimización global a través del Algoritmo DIRMIN el cual se presenta a continuación.

3.1.4.1. Algoritmo DIRMIN

La idea de sumar una búsqueda local dentro de un algoritmo tipo DIRECT, como los vistos hasta ahora en este capítulo, trae consigo una noción bastante intuitiva y coherente: un paso de minimización local ayudaría a explotar regiones suficientemente cerca de los puntos evaluados acelerando así la convergencia hacia el valor funcional óptimo. Además sería una ayuda en aquellos problemas que por cuestiones de dimensionalidad DIRECT no logra una buena performance.

El algoritmo DIRMIN fue inspirado en los enfoques de comienzos múltiples (*multi-start approach* en inglés) para optimización global, donde se realizan múltiples minimizaciones locales comenzando en puntos generados, de acuerdo a algún criterio, sobre la región factible. En este caso, teniendo en cuenta que DIRECT es capaz de localizar una región que contiene el optimizador global bastante rápido (lo más difícil es lograr que determine ese optimizador de manera unívoca), los puntos elegidos para comenzar la minimización local son precisamente los centros, también llamados centroides, de los hiperrectángulos potencialmente óptimos.

El mencionado procedimiento de minimización local es representado por los autores a través de un mapeo $\mathcal{A} : \mathbb{R}^n \rightarrow \mathbb{R}^n$. En vistas de la prueba de convergencia del algoritmo DIRMIN dicho mapeo debe satisfacer la siguiente hipótesis:

Hipótesis 3.1.1. *Para cualquier punto inicial $x \in D$:*

1. $\bar{x} = \mathcal{A}(x)$ es un punto estacionario.
2. Para cada mínimo global x^* y un conjunto abierto \mathcal{L} existe x tal que si $x \in \mathcal{L}$ entonces $x^* = \mathcal{A}(x)$.

Dado que la estructura del algoritmo DIRMIN es bastante similar a los algoritmos presentados hasta el momento, salvo el paso de minimización local, se omitirá presentar la estructura algorítmica del mismo. A continuación se enuncian los resultados teóricos de convergencia del algoritmo DIRMIN.

Proposición 3.1.5. *Si x^* es un minimizador global y \mathcal{L} el correspondiente conjunto introducido en la Hipótesis (3.1.1) entonces existen una iteración k y un índice $h \in \mathbb{I}_k$ tal que $h \in \mathbb{I}_k^*$ y $x_h \in \mathcal{L}$.*

Demostración. Ver demostración en [35]. □

La Hipótesis 3.1.1 en combinación con la Proposición 3.1.5 dan lugar al siguiente resultado de convergencia del algoritmo DIRMIN.

Proposición 3.1.6. *Existe una iteración k y un índice $h \in \mathbb{I}_k^*$ tal que $\mathcal{A}(x_h) = x^*$ donde x^* es un optimizador global.*

Demostración. Ver demostración en [35]. □

3.2. Algoritmo BIRECT irrestricto con búsqueda local del tipo búsqueda de patrones

3.2.1. Introducción

Las ideas de Liuzzi, Lucidi y Piccialli [35] presentadas en la sección anterior motivaron el estudio de una nueva variante algorítmica dentro de los algoritmos del tipo DIRECT, dando lugar así a la segunda contribución de esta tesis doctoral. De esta manera, tomando como base las nociones detrás de DIRMIN, en esta sección indagaremos el impacto en la performance del algoritmo BIRECT [43] cuando el mismo es combinado con un algoritmo de búsqueda local del tipo búsqueda de patrones. El objetivo principal de la incorporación de un algoritmo de este tipo es mejorar la aproximación calculada explorando puntos en una vecindad de la misma. Para realizar este análisis, utilizaremos nuevamente el algoritmo **nm**ps presentado y desarrollado en el Capítulo 2 de esta tesis.

Teniendo en cuenta que ambos algoritmos, BIRECT y **nm**ps, poseen pruebas de convergencia y experimentos numéricos que lo validan, lo esperado es que la combinación de ambos mejore la obtención del optimizador. Existe la posibilidad de que el proceso iterativo llevado a cabo para encontrar dicho punto óptimo incremente significativamente el tiempo de ejecución del algoritmo, principalmente debido a la complejidad computacional adicional de combinar una búsqueda local con una global. Sin embargo, esta faceta se encuentra fuera del objetivo de análisis de esta sección.

Por lo tanto, el propósito de esta sección es determinar ventajas y desventajas de agregar una búsqueda local (del tipo búsqueda de patrones) dentro del procedimiento de minimización global del algoritmo BIRECT para el caso de problemas con restricciones de cotas en las variables.

3.2.2. Detalles de implementación de BIRECT con búsqueda de patrones

La incorporación del algoritmo **nm**ps tiene lugar inmediatamente después de que hiperrectángulo m actual es particionado. Los dos nuevos hiperrectángulos, indexados por $FE + 1$ y $FE + 2$, definen las fronteras (cotas) de dos problemas de optimización local.

Las iteraciones del algoritmo **nm**ps comienzan considerando como punto inicial el asociado al valor funcional mínimo registrado hasta el momento en el hiperrectángulo m . A su vez, la longitud de paso inicial requerida por el procedimiento de búsqueda de patrones es definida como el mínimo entre 1 y las longitudes de los lados del hiperrectángulo, garantizando de esta manera que la búsqueda no se expanda demasiado. Es así como se llevan a cabo los dos procesos de minimización local en los hiperrectángulos $FE + 1$ y $FE + 2$ respectivamente.

En cada caso, las evaluaciones funcionales requeridas son sumadas al conteo total del algoritmo de manera de no exceder el límite establecido. Ante la presencia de un punto con un valor funcional menor al obtenido hasta el momento, este se convierte en la nueva mejor aproximación al óptimo global. Terminadas las iteraciones locales se chequea si el nuevo optimizador satisface alguno de los criterios de convergencia, caso contrario se continua con el procedimiento de partición del espacio y búsqueda global propia del algoritmo BIRECT.

Es importante aclarar que ambos procesos, optimización global y local, son independientes, con lo cual el orden en que se ejecutan, o si se ejecutan ambos simultáneamente, no afecta el resultado final.

3.2.3. Análisis y experimentación numérica

Para llevar a cabo el mencionado análisis se consideró, al igual que en [35] y [43], el conjunto de problemas test con restricciones de cotas de Hedar para optimización global ². Dicho conjunto cuenta con 27 problemas, algunos de los cuales poseen diferentes variantes en su formulación o en dimensión, dando así un total de 54 problemas de prueba. Las características generales (dimensión, tipo de función objetivo, cotas en las variables, número de mínimos locales y valor óptimo global conocido) y el detalle de los mismos se presenta en la Tabla 3.1 y en el Apéndice C respectivamente.

El análisis propuesto se basó en un estudio numérico exhaustivo en el cual se probaron distintas combinaciones de parámetros en la ejecución de ambos algoritmos. Desde el punto de vista de BIRECT se consideraron tres tolerancias diferentes en el criterio de parada relacionado con la aceptación de la aproximación obtenida. Partiendo de la base que todos los problemas de prueba cuentan con una solución óptima global conocida, al igual que en [43], se definió el mencionado criterio en base al error relativo de la aproximación $f(\bar{x})$ respecto a su valor óptimo esperado f^* como

$$\text{tol} = \begin{cases} \frac{f(\bar{x}) - f^*}{|f^*|} \leq \text{tol}_{\text{máx}}, & f^* \neq 0 \\ f(\bar{x}) \leq \text{tol}_{\text{máx}}, & f^* = 0, \end{cases} \quad (3.21)$$

donde se consideró $\text{tol}_{\text{máx}} = 10^{-3}, 10^{-4}$ y 10^{-5} . Es importante notar que la resta del numerador de la primera parte de (3.21) no necesita valor absoluto puesto que se supone que $f(\bar{x})$ siempre será mayor o igual a f^* . En cuanto al algoritmo `nmps` se emplearon 4 cotas diferentes en el criterio de parada relacionado con la longitud de paso, es decir

$$\Delta_k < \Delta_{\text{tol}_{\text{local}}},$$

donde los valores posibles para $\Delta_{\text{tol}_{\text{local}}}$ fueron $10^{-1}, 10^{-2}, 10^{-3}$ y 10^{-4} . Finalmente, en relación al criterio de parada que involucra las evaluaciones funcionales, esto es

$$FE < FE_{\text{máx}_{\text{local}}},$$

se consideraron 4 valores distintos de $FE_{\text{máx}_{\text{local}}}$: 50, 100, 250 y 500.

²<https://www.sfu.ca/~ssurjano/optimization.html>

Prob. N°	Nombre del Problema	Dimensión	Tipo de función objetivo	Cotas de las variables	Número de mínimos locales	Óptimo global f^*
1	Ackley ₂	2	General	$[-15, 35]^2$	multimodal	0
2	Ackley ₅	5	General	$[-15, 35]^5$	multimodal	0
3	Ackley ₁₀	10	General	$[-15, 35]^{10}$	multimodal	0
4	Beale	2	Polinomial general	$[-4.5, 4.5]^2$	multimodal	0
5	Bohachevsky 1	2	General	$[-100, 110]^2$	multimodal	0
6	Bohachevsky 2	2	General	$[-100, 110]^2$	multimodal	0
7	Bohachevsky 3	2	General	$[-100, 110]^2$	multimodal	0
8	Booth	2	Polinomial general	$[-10, 10]^2$	unimodal	0
9	Branin	2	General	$[-5, 10] \times [0, 15]$	3	0.397887
10	Colville	4	Polinomial general	$[-10, 10]^4$	multimodal	0
11	Dixon & Price ₂	2	Polinomial general	$[-10, 10]^2$	unimodal	0
12	Dixon & Price ₅	5	Polinomial general	$[-10, 10]^5$	unimodal	0
13	Dixon & Price ₁₀	10	Polinomial general	$[-10, 10]^{10}$	unimodal	0
14	Easom	2	General	$[-100, 100]^2$	multimodal	-1
15	Goldstein & Price	2	Polinomial general	$[-2, 2]^2$	4	3
16	Griewank	2	General	$[-600, 700]^2$	multimodal	0
17	Hartman ₃	3	General	$[0, 1]^3$	4	-3.86278
18	Hartman ₆	6	General	$[0, 1]^6$	4	-3.32237
19	Hump	2	Polinomial general	$[-3, 3] \times [-2, 3]$	6	-1.0316
20	Levy ₂	2	General	$[-10, 10]^2$	multimodal	0
21	Levy ₅	5	General	$[-10, 10]^5$	multimodal	0
22	Levy ₁₀	10	General	$[-10, 10]^{10}$	multimodal	0
23	Matyas	2	Polinomial general	$[-10, 15]^2$	unimodal	0
24	Michalewics ₂	2	General	$[0, \pi]^2$	2!	-1.80130
25	Michalewics ₅	5	General	$[0, \pi]^5$	5!	-4.68765
26	Michalewics ₁₀	10	General	$[0, \pi]^{10}$	10!	-9.66015
27	Perm	4	General	$[-4, 4]^4$	multimodal	0
28	Powell ₄	4	Polinomial general	$[-4, 5]^4$	multimodal	0
29	Powell ₈	8	Polinomial general	$[-4, 5]^8$	multimodal	0
30	Power Sum	4	Polinomial general	$[0, 4]^4$	multimodal	0
31	Rastrigin ₂	2	General	$[-5.12, 6.12]^2$	multimodal	0
32	Rastrigin ₅	5	General	$[-5.12, 6.12]^5$	multimodal	0
33	Rastrigin ₁₀	10	General	$[-5.12, 6.12]^{10}$	multimodal	0
34	Rosenbrock ₂	2	Polinomial general	$[-5, 10]^2$	unimodal	0
35	Rosenbrock ₅	5	Polinomial general	$[-5, 10]^5$	unimodal	0
36	Rosenbrock ₁₀	10	Polinomial general	$[-5, 10]^{10}$	unimodal	0
37	Schwefel ₂	2	General	$[-500, 500]^2$	unimodal	0
38	Schwefel ₅	5	General	$[-500, 500]^5$	unimodal	0
39	Schwefel ₁₀	10	General	$[-500, 500]^{10}$	unimodal	0
40	Shekel ₅	4	General	$[0, 10]^4$	5	-10.15320
41	Shekel ₇	4	General	$[0, 10]^4$	7	-10.40294
42	Shekel ₁₀	4	General	$[0, 10]^4$	10	-10.53641
43	Shubert ₁₀	2	General	$[-10, 10]^2$	760	-186.73091
44	Sphere ₂	2	Polinomial general	$[-5.12, 6.12]^2$	multimodal	0
45	Sphere ₅	5	Polinomial general	$[-5.12, 6.12]^5$	multimodal	0
46	Sphere ₁₀	10	Polinomial general	$[-5.12, 6.12]^{10}$	multimodal	0
47	Sum squares ₂	2	Polinomial general	$[-10, 15]^2$	unimodal	0
48	Sum squares ₅	5	Polinomial general	$[-10, 15]^5$	unimodal	0
49	Sum squares ₁₀	10	Polinomial general	$[-10, 15]^{10}$	unimodal	0
50	Trid ₆	6	Polinomial general	$[-36, 36]^6$	multimodal	-50
51	Trid ₁₀	10	Polinomial general	$[-100, 100]^{10}$	multimodal	-200
52	Zakharov ₂	2	Polinomial general	$[-5, 11]^2$	unimodal	0
53	Zakharov ₅	5	Polinomial general	$[-5, 11]^5$	unimodal	0
54	Zakharov ₁₀	10	Polinomial general	$[-5, 11]^{10}$	unimodal	0

Tabla 3.1: Descripción de problemas test - BIRECT con búsqueda local.

Un comentario importante a realizar es que la elección de estos valores no fue aleatoria sino que está basada en razones algorítmicas y relacionadas a los métodos de optimización sin derivadas. Por un lado, desde un punto de vista algorítmico, en todo estudio numérico que involucra diferentes parámetros es central encontrar la combinación más óptima que acelera la convergencia del proceso iterativo. Desde el punto de vista de los métodos de optimización sin derivadas, por otro lado, el número de evaluaciones funcionales es el parámetro por excelencia para medir la eficacia de los métodos considerados al encontrar el mínimo. Con respecto a esto es clave tener en cuenta que, al combinar una estrategia de búsqueda local con una global en un solo algoritmo, la cantidad de evaluaciones funcionales finales es la suma resultante de ambos procesos. Asimismo, recordando lo visto en el Capítulo 2 en relación a los métodos de búsqueda de patrones, el tamaño de paso suficientemente pequeño nos asegura estar en una vecindad del óptimo local. Finalmente, cuando la solución de los problemas considerados es conocida, como en el caso del conjunto de problemas test analizados, establecer un umbral o cota superior para el error relativo como en (3.21) constituye un buen criterio para medir la performance del algoritmo.

Por estas razones, ante la presencia de dos filosofías de optimización diferentes (local y global) en un mismo algoritmo, la intuición lleva a pensar que los parámetros involucrados en cada una deberían *ayudarse mutuamente* con el objetivo de alcanzar la convergencia al optimizador lo más rápido posible. Si esto no llegara a ocurrir, es una clara señal de que la combinación no tiene mayores beneficios que la ejecución de cada algoritmo por separado.

Retomando la combinación de parámetros y a modo de cierre de este punto notemos que, por ejemplo, la optimización local podría ser más exigente en cuanto al tamaño del paso o la cantidad de evaluaciones funcionales permitidas (ó ambas) dejando que la optimización global sea más *relajada* en la tolerancia exigida.

En base a los comentarios anteriores, para cada uno de los 54 problemas test se analizaron 48 combinaciones diferentes entre los valores de $\text{tol}_{\text{máx}}$, $\Delta_{\text{tol}_{\text{local}}}$ y $FE_{\text{máx}_{\text{local}}}$, generando un total de 2592 resultados experimentales. El esquema de combinaciones resultante puede observarse en la Tabla 3.2. Para que los resultados sean comparables con los expuestos por Paulavičius et. al. en [43], se siguió el mismo criterio que los autores, estableciendo como tope de evaluaciones funcionales FE el número 500000.

Tabla 3.2: Esquema de combinaciones experimentales - BIRECT con búsqueda local

$\text{tol}_{\text{máx}}$	$\Delta_{\text{tol}_{\text{local}}}$	$FE_{\text{máx}_{\text{local}}}$	$\text{tol}_{\text{máx}}$	$\Delta_{\text{tol}_{\text{local}}}$	$FE_{\text{máx}_{\text{local}}}$	$\text{tol}_{\text{máx}}$	$\Delta_{\text{tol}_{\text{local}}}$	$FE_{\text{máx}_{\text{local}}}$
10^{-3}	10^{-1}	50	10^{-4}	10^{-1}	50	10^{-5}	10^{-1}	50
		100			100			100
		250			250			250
		500			500			500
	10^{-2}	50		10^{-2}	50		10^{-2}	50
		100			100			100
		250			250			250
		500			500			500
	10^{-3}	50		10^{-3}	50		10^{-3}	50
		100			100			100
		250			250			250
		500			500			500
10^{-4}	50	10^{-4}	50	10^{-4}	50			
	100		100		100			
	250		250		250			
	500		500		500			

Los resultados de las combinaciones experimentales anteriores proporcionaron bastante in-

formación relevante. En primer lugar, contrariamente a nuestra intuición, aquellos casos con pocas evaluaciones funcionales locales permitidas (50 y 100) fueron los menos exitosos. Esto significa que para el conjunto de parámetros que involucraban un máximo de 50 o 100 evaluaciones funcionales locales, las aproximaciones al optimizador global y la cantidad de evaluaciones funcionales locales dieron mejor en la versión original de BIRECT (es decir, sin búsqueda local). Una explicación encontrada para este suceso es que ambos valores de FE podrían resultar insuficientes para que la búsqueda local tenga impacto visible dentro de la performance del algoritmo de optimización global. Inversamente, aquellas combinaciones con $FE_{\text{máx}_{\text{local}}}$ igual 250 o 500 mostraron mejoras tanto en el valor funcional óptimo como el número total de evaluaciones funcionales en una buena cantidad de problemas.

Un segundo punto interesante es que los resultados, siempre en comparación a la versión original de BIRECT, mejoran cuando $\text{tol}_{\text{máx}}$ es 10^{-3} o 10^{-4} y $\Delta_{\text{tol}_{\text{local}}}$ es 10^{-1} o 10^{-2} . Es decir, aquellas combinaciones donde ambas tolerancias son altas simultáneamente no serían recomendables. Con esta observación se hace evidente lo mencionado en párrafos anteriores: para que la conjunción de ambas filosofías de optimización funcione, ninguna de ellas debería agotar todos los recursos o ser demasiado exigente de manera de no anular u opacar el trabajo computacional de la otra.

Finalmente, la combinación que mostró resultados más satisfactorios fue la dada por

$$\left\{ \begin{array}{l} \text{tol}_{\text{máx}} = 10^{-3} \\ \Delta_{\text{tol}_{\text{local}}} = 10^{-2} \\ FE_{\text{máx}_{\text{local}}} = 500, \end{array} \right.$$

los mismos se muestran en la Tabla 3.3. En esta tabla observamos que BIRECT combinado con una búsqueda local del tipo búsqueda de patrones mejora la performance de su antecesor en un total de 13 problemas. Estos problemas son de dimensiones bastante variadas lo cual imposibilita encontrar un razón que explique el suceso anterior, al menos desde la perspectiva dimensional. En esta lista encontramos por ejemplo Shubert y Easom de dimensión 2, Powell₄ y Power Sum de dimensión 4, Dixon & Price₅ y Sphere₅ de dimensión 5 y Powell₈ de dimensión 8. Por otro lado, en aproximadamente un 55 % del conjunto de problemas test considerados, el algoritmo BIRECT en su versión original, continua mostrando mejores resultados en relación con la incorporación del algoritmo nmps.

Adicionalmente, en los valores observados en la Tabla 3.3 hay dos detalles notorios que valen la pena señalar. En primer lugar, que en los 4 problemas en los cuales BIRECT *falló* (es decir, alcanzó el tope de 500000 evaluaciones funcionales sin encontrar la solución) también lo hizo el algoritmo BIRECT con búsqueda local. Sin embargo, en estos 4 problemas con *baja performance*, se observa cierta paridad entre los dos métodos comparados. Es así como, mientras BIRECT alcanza un mejor aproximación en Perm y Schwefel₁₀, la aproximación en Michalewics₁₀ y Zahkharov₁₀ es mejor en BIRECT con búsqueda local. Es oportuno notar que salvo Perm, los restantes 3 problemas son de dimensión 10 lo cual podría tener alguna influencia en los resultados.

El segundo detalle a mencionar se encuentra relacionado con aquellos problemas marcados con † en la Tabla 3.3 (7 en total). En estos casos, las comparaciones numéricas entre ambos métodos no arrojaron un resultado determinante e incluso podrían considerarse en cierto punto equiparables. Si bien en estos 7 problemas la aproximación al valor funcional óptimo, tanto en BIRECT como en BIRECT con búsqueda de patrones, alcanzó un orden de precisión aceptable, en algunos casos mientras esta precisión fue mejor en el primer método, el número de evaluaciones funcionales fue menor en el segundo, o viceversa. Algo llamativo e interesante es que, a excepción de Sphere₂, los restantes 6 son todos problemas unimodales. En este tipo de situaciones, muchas veces la decisión del mejor método es dejada a criterio del usuario ya que dependerá de si la

prioridad está puesta en la precisión o en el número de evaluaciones funcionales. Muy a menudo, principalmente cuando hablamos de optimización global, una precisión del orden de 10^{-4} podría considerarse un valor muy exigente.

A modo de tener una visión más global de los resultados y dado que usualmente los valores en forma tabular son difíciles de interpretar, en la Figura 3.7 se muestran los gráficos de distribución de las evaluaciones funcionales en los dos algoritmos estudiados. A nivel general, las distribuciones se observan bastante similares, lo cual es una muy buena señal. Tanto los picos como los valles se ubican aproximadamente en los mismos lugares más allá de algunas diferencias de altura en los valores del eje y , lo cual se explica con el análisis realizado en los párrafos previos.

Figura 3.7: *Distribución de cantidad de evaluaciones funcionales.*

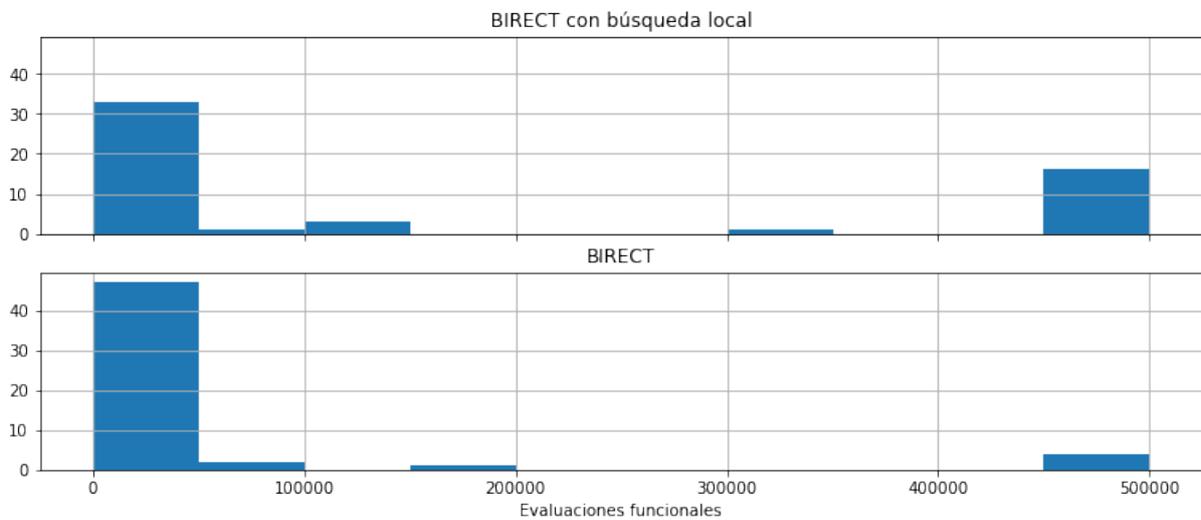


Tabla 3.3: Resultados numéricos - BIRECT irrestricto con búsqueda local

Prob. n°	Problema	BIRECT CON BÚSQUEDA LOCAL		BIRECT	
		$tol_{\text{máx}} = 10^{-3}$ $tol_{\text{local}} = 10^{-2}$ $FE_{\text{máxlocal}} = 500$		$tol_{\text{máx}} = 10^{-4}$	
		$f(\bar{x})$	FE	$f(\bar{x})$	FE
1	Ackley ₂	4.4409e-16	7941	2.54e-5	202
2	Ackley ₅	0.019789	>500000	2.54e-5	1268
3	Ackley ₁₀	0.43613	>500000	2.54e-5	47792
4	Beale	0	349	9.17e-5	436
5	Bohachevsky 1	0	457	4.02e-5	476
6	Bohachevsky 2	0	475	3.35e-5	478
7	Bohachevsky 3	0	475	3.67e-5	480
8	Booth †	2.4414e-04	169	6.10e-5	194
9	Branin	0.39794	210	0.39790	242
10	Colville	8.6248e-04	331453	9.82e-5	794
11	Dixon & Price ₂ †	2.0496e-04	189	4.84e-5	722
12	Dixon & Price ₅	3.3155e-05	1796	7.15e-5	4060
13	Dixon & Price ₁₀ †	5.8332e-04	132569	9.52e-5	164826
14	Easom	-0.99989	11639	-0.99999	16420
15	Goldstein & Price	3	1106	3.00019	274
16	Griewank	1.3570e-05	113127	7.76e-7	5106
17	Hartman ₃	-3.8624	266	-3.86242	352
18	Hartman ₆	-3.3196	8732	-3.32206	764
19	Hump	-1.0316	356	-1.03154	334
20	Levy ₂	3.7154e-05	1306	9.09e-5	152
21	Levy ₅	8.7662e-04	31724	1.83e-5	1024
22	Levy ₁₀	0.029500	>500000	3.55e-5	7904
23	Matyas †	1.0851e-06	176	2.71e-5	94
24	Michalewics ₂	-1.8013	280	-1.80118	126
25	Michalewics ₅	-4.6397	>500000	-4.68736	73866
26	Michalewics ₁₀	-8.4383	>500000	-7.32661	>500000
27	Perm	0.031497	>500000	0.00203	>500000
28	Powell ₄	0	457	4.86e-5	2114
29	Powell ₈	0	1004	9.71e-5	99514
30	Power Sum	0	1378	9e-5	10856
31	Rastrigin ₂	5.8553e-04	1979	4.81e-5	180
32	Rastrigin ₅	6.8886e-04	46380	1.18e-5	1394
33	Rastrigin ₁₀	1.1400	>500000	2.36e-5	40254
34	Rosenbrock ₂	5.7949e-04	6972	9.65e-5	242
35	Rosenbrock ₅	0.0047013	>500000	2.41e-5	1700
36	Rosenbrock ₁₀	4.4238	>500000	5.42e-5	10910
37	Schwefel ₂	1.6575e-05	2845	5.64e-5	236
38	Schwefel ₅	118.44	>500000	6.41e-5	7210
39	Schwefel ₁₀	1214.7	>500000	0.00012	>500000
40	Shekel ₅	-10.148	33165	-10.15307	1200
41	Shekel ₇	-10.395	18822	-10.40269	1180
42	Shekel ₁₀	-10.528	18834	-10.53618	1140
43	Shubert ₁₀	-186.57	172	-186.72441	1780
44	Sphere ₂ †	2.7778e-06	141	1.15e-5	118
45	Sphere ₅	1.1111e-05	687	2.87e-5	712
46	Sphere ₁₀	0.0068802	>500000	5.74e-5	16974
47	Sum squares ₂ †	8.1380e-05	182	7.94e-6	244
48	Sum squares ₅	4.0690e-04	117516	3.97e-5	1034
49	Sum squares ₁₀	0.0068802	>500000	9.11e-6	7688
50	Trid ₆	-50	55064	-49.99512	1506
51	Trid ₁₀	-185.47	>500000	-200.40581	1656
52	Zakharov ₂ †	1.1529e-04	150	2.88e-5	502
53	Zakharov ₅	0.0056695	>500000	6.44e-5	20974
54	Zakharov ₁₀	1.3307	>500000	9.41133	>500000

3.2.4. Conclusiones y observaciones adicionales

Como se puede inferir en base a lo desarrollado hasta el momento en esta sección, si bien la incorporación de un algoritmo de búsqueda de patrones dentro del proceso de optimización global de BIRECT mostró mejoras para la mejor combinación considerada, las mismas no son suficientes para obtener una conclusión firme al respecto.

Sin embargo, aunque no fue posible establecer una regla o combinación óptima de parámetros que muestre una mejora sustancial, lo que sí es claro es que la búsqueda local es un complemento positivo y beneficioso en muchos casos. En efecto, considerando la totalidad de los 2592 resultados experimentales, se observa que en el 35 % de los casos (19 de los 54 problemas considerados) la búsqueda local obtuvo una solución óptima global similar a BIRECT (en al menos una combinación) pero con un número evaluaciones funcionales menor. Para observar con más detalle este punto, la Figura 3.8 muestra el porcentaje de la máxima mejora obtenida en el número de evaluaciones funcionales con referencia al valor dado por el algoritmo de Paulavičius et. al. en [43] sobre el total de experimentos. Adicionalmente, la Figura 3.9 expone la cantidad de combinaciones (con respecto a las 48 trabajadas) dónde se produjo este descenso en la variable FE. Ambos gráficos son un claro ejemplo de que esta línea de análisis y experimentación debe seguir siendo investigada en el futuro.

Figura 3.8: *Máximo porcentaje (%) de mejora obtenido en evaluaciones funcionales (EF) con la incorporación de la búsqueda local.*

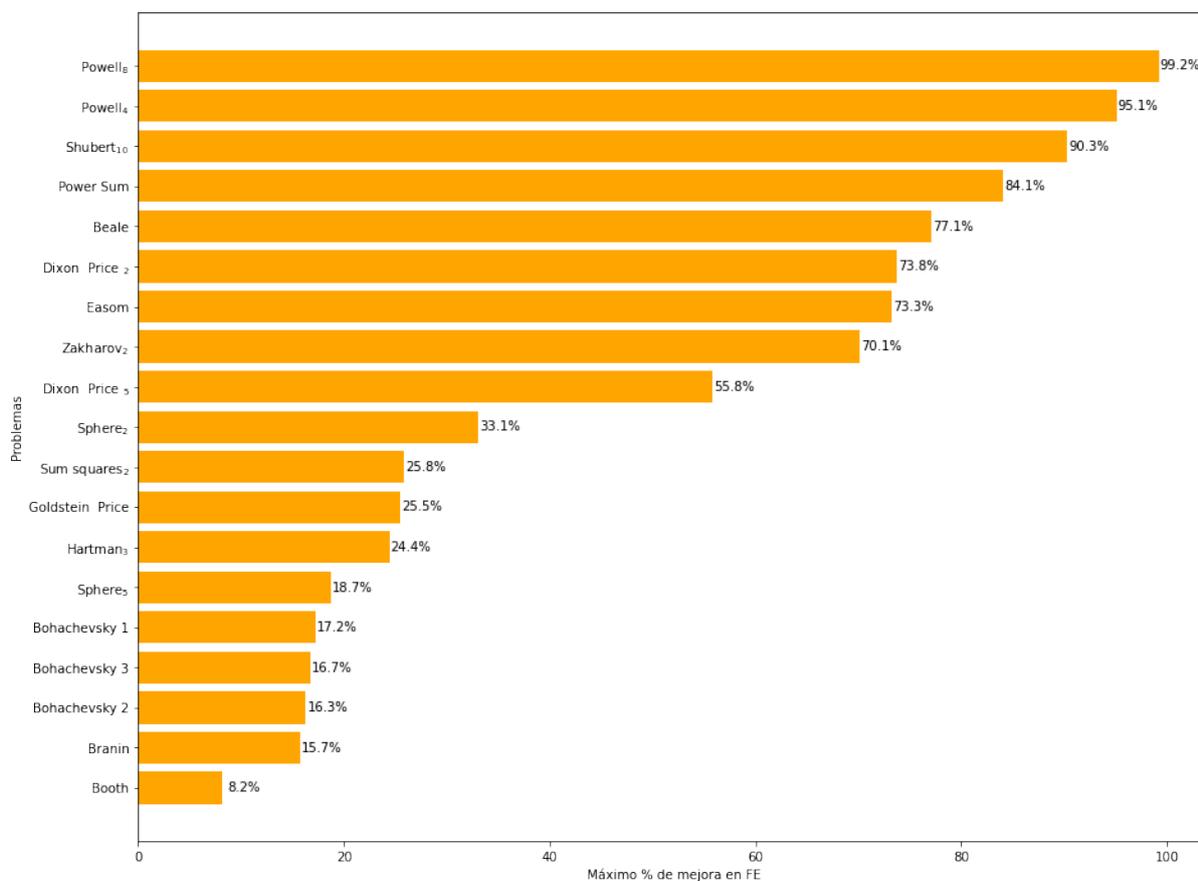
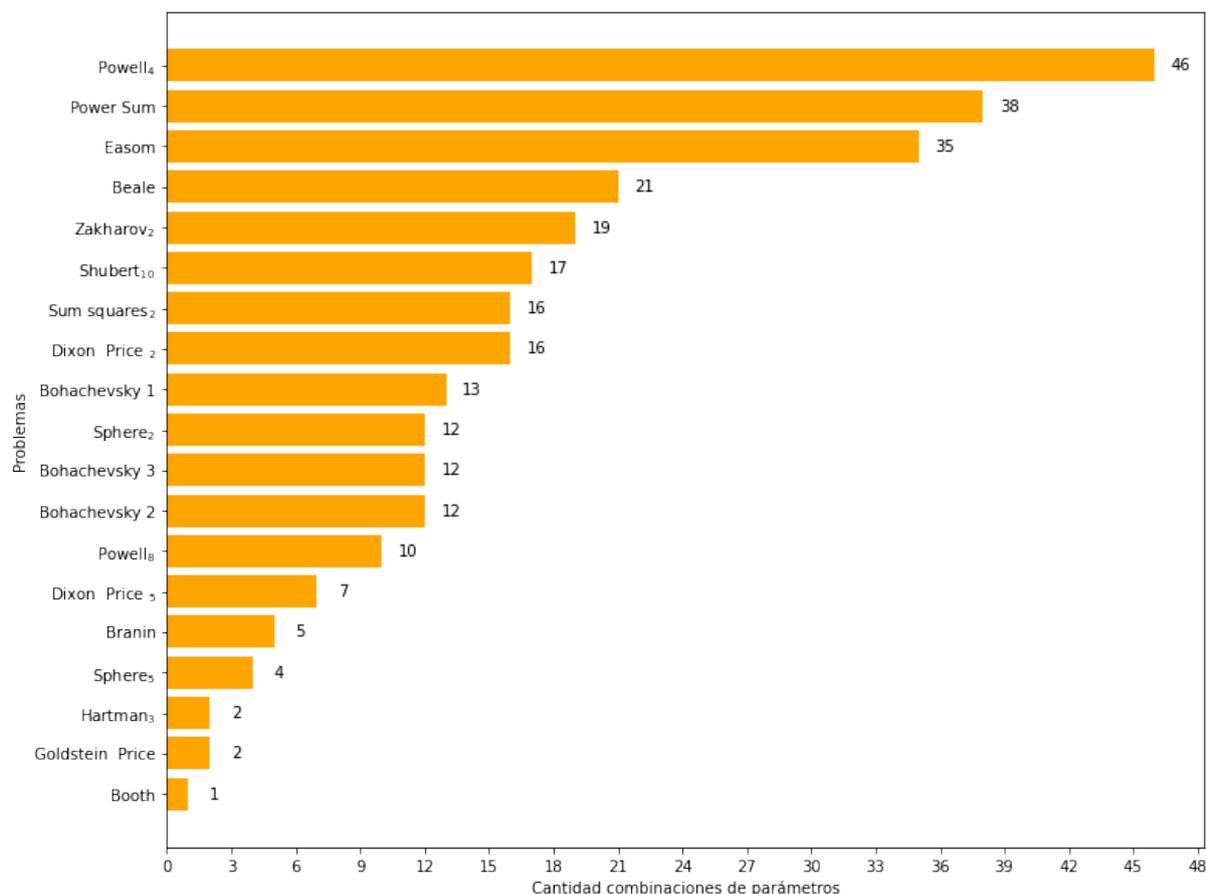


Figura 3.9: Cantidad de combinaciones de parámetros donde la búsqueda local redujo el número evaluaciones funcionales.



3.3. Algoritmo BIRECT con restricciones generales

3.3.1. Presentación

Los trabajos de Jones et al. [24, 23] y de Paulavičius et al. [43] presentados a lo largo de las Secciones 3.1.1 y 3.1.2 fueron el puntapié inicial de la última contribución dada por esta tesis. Por un lado, la propuesta de Jones en [23] para manejar restricciones generales a través del uso de la función auxiliar (3.15) fue bastante atractiva e innovadora, además de que amplió la gama de problemas a los cuales el algoritmo DIRECT puede ser aplicado. Por otro lado, los autores de [43] presentaron una nueva forma de realizar la partición de hiperrectángulos a través del algoritmo BIRECT. Este último con propiedades de convergencia heredadas de DIRECT. Aprovechando las bondades e innovaciones de ambos enfoques, en este trabajo se propone un nuevo algoritmo para resolver problemas de optimización global con restricciones generales utilizando un proceso de bisección como técnica de partición del espacio.

Consideremos entonces el problema de optimización global dado por

$$\begin{aligned}
 & \underset{x \in \mathbb{R}^n}{\text{minimizar}} && f(x_1, x_2, \dots, x_n) \\
 & \text{sujeto a} && g_1(x_1, x_2, \dots, x_n) \leq 0 \\
 & && g_2(x_1, x_2, \dots, x_n) \leq 0 \\
 & && \vdots \\
 & && g_m(x_1, x_2, \dots, x_n) \leq 0 \\
 & && l_i \leq x_i \leq u_i
 \end{aligned} \tag{3.22}$$

con $f : \mathbb{R}^n \rightarrow \mathbb{R}$ la función objetivo, $g : \mathbb{R}^n \rightarrow \mathbb{R}^m$ el conjunto de restricciones generales de desigualdad y $D = [l, u] = \{x \in \mathbb{R}^n : l_i \leq x_i \leq u_i, i = 1, 2, \dots, n\}$ el conjunto definido por las restricciones de cotas. De esta manera, la región factible del problema (3.22) queda determinada por

$$\Omega = \{x \in D : g_j(x) \leq 0, j = 1, 2, \dots, m\}.$$

Siguiendo lo desarrollado hasta el momento en este capítulo, consideremos nuevamente

$$\bar{D} = [0, 1]^n = \{x \in \mathbb{R}^n : 0 \leq x_i \leq 1, i = 1, 2, \dots, n\}$$

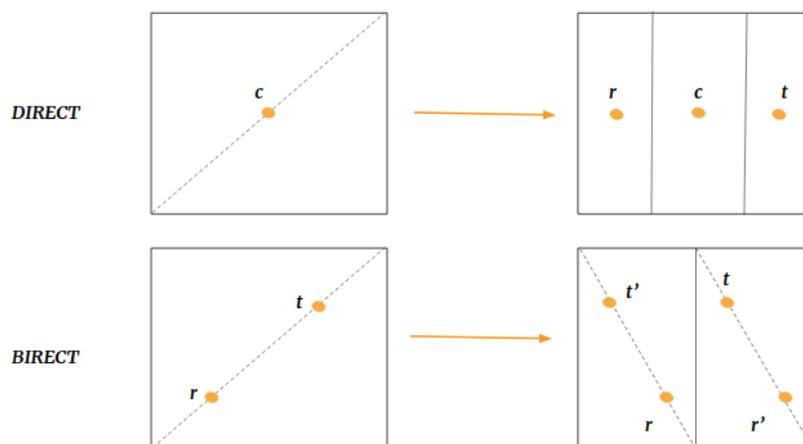
como el hipercono unitario obtenido a partir de la transformación lineal T aplicada al conjunto D , es decir

$$\begin{aligned}
 T : D &\rightarrow \bar{D}, \\
 T(x) &= A(x - l), \text{ con } A = \text{diag} \left(\frac{1}{u_1 - l_1}, \frac{1}{u_2 - l_2}, \dots, \frac{1}{u_n - l_n} \right).
 \end{aligned} \tag{3.23}$$

Obsérvese que $l \xrightarrow{T} (0, 0, \dots, 0)$ y $u \xrightarrow{T} (1, 1, \dots, 1)$.

3.3.2. Los nuevos coeficientes s_j

Recordemos en primer lugar que el principal cambio del algoritmo DIRECT al algoritmo BIRECT (en el caso irrestricto) radica en la técnica de partición de hiperrectángulos. DIRECT por su lado, utilizando una trisección que tiene como referencia el punto central de cada hiperrectángulo, genera dos nuevos puntos a *izquierda* y a *derecha* de este punto central de referencia. Los nuevos puntos y el punto central original, se convierten así en los centros de los 3 nuevos hiperrectángulos. BIRECT, en cambio, mediante una bisección que toma como punto de partida dos puntos equidistantes entre sí ubicados sobre la diagonal del hiperrectángulo también genera dos nuevos puntos. A diferencia de DIRECT, estos dos nuevos puntos pasan a formar parte de las diagonales de los 2 nuevos hiperrectángulos generados, junto con los puntos de partida originales. En la Figura 3.10 se puede apreciar gráficamente esta diferencia en un caso de 2 dimensiones.

Figura 3.10: *Diferencias en la generación de los nuevos puntos entre DIRECT y BIRECT*

Este cambio observado tiene una implicación directa en la forma en la cual se piensan los coeficientes s_0 y s_j , ecuaciones (3.16) y (3.17), cuando un esquema de bisección es aplicado a un problema con restricciones generales como (3.22). Recordemos que la idea detrás de estos coeficientes es obtener una aproximación de las tasas de cambio de la función objetivo (s_0) y de las restricciones (s_j) en los nuevos puntos obtenidos.

Teniendo presente la Figura 3.10, sean r^1 y t^1 los puntos iniciales asociados a la primera iteración y pertenecientes al hiperrectángulo D^1 (la región inicial de partida). Como se explicó en la Sección 3.1.2 el procedimiento de bisección hará que estos puntos pasen a formar parte de la diagonal de dos nuevos hiperrectángulos, D^2 y D^3 respectivamente. Esto explica el sentido de realizar las asignaciones $r^2 \leftarrow r^1$ y $t^3 \leftarrow t^1$.

Observemos que hasta el momento, las diagonales de estos nuevos hiperrectángulos (o rectángulos como en el caso de ejemplo) contienen solo un punto cada uno. El siguiente paso consiste entonces en generar t^2 y r^3 , dentro de las diagonales de D^2 y D^3 respectivamente. Recordando los pasos del Algoritmo 3.3 de la Sección 3.1.2, la bisección es realizada a lo largo de la dirección coordenada que tiene la longitud más larga (la denominada variable de ramificación br definida en la fórmula (3.13)). A continuación, si la condición $r_{\text{bisec}} < t_{\text{bisec}}$ se cumple se realiza la siguiente asignación

$$\begin{aligned} r_{br}^3 &\leftarrow r_{br}^1 + \frac{d_{br}^1}{2} \\ t_{br}^2 &\leftarrow t_{br}^1 - \frac{d_{br}^1}{2}, \end{aligned} \quad (3.24)$$

caso contrario lo anterior es reemplazado por

$$\begin{aligned} r_{br}^2 &\leftarrow r_{br}^1 - \frac{d_{br}^1}{2} \\ t_{br}^3 &\leftarrow t_{br}^1 + \frac{d_{br}^1}{2}. \end{aligned} \quad (3.25)$$

Más adelante, en el detalle del algoritmo se profundizará este punto. Es importante notar que este procedimiento implica que sólo se modifican las coordenadas de r y t en la variable de ramificación br , las demás coordenadas permanecen inmutables.

Llegado a este punto es precisamente donde surge la pregunta de cómo extrapolar las ecuaciones (3.16) y (3.17) al esquema de bisección. Lo importante es no perder de vista que el objetivo

de estos coeficientes es dar una noción o aproximación de las tasas de cambio de las funciones involucradas en el problema (3.22). En esta línea de pensamiento, retomando el caso de ejemplo en dos dimensiones, se podrían definir entonces los coeficientes

$$s_0 = s_0 + \frac{|f(r^3) - f(r^2)|}{\|r^3 - r^2\|} + \frac{|f(t^2) - f(t^3)|}{\|t^2 - t^3\|} \quad (3.26)$$

$$s_j = s_j + \frac{|g_j(r^3) - g_j(r^2)|}{\|r^3 - r^2\|} + \frac{|g_j(t^2) - g_j(t^3)|}{\|t^2 - t^3\|}, j = 1, 2, \dots, m \quad (3.27)$$

Esta forma de obtener una expresión para las tasas de cambio, está basada justamente en las ecuaciones (3.24) y (3.25). Dado que los *nuevos* puntos r se calculan en función de los *viejos* r y los *nuevos* t en función de los *viejos* t , las expresiones (3.26) y (3.27) nos dan una medida del comportamiento de la función objetivo y de las funciones que conforman las restricciones luego de que la bisección es realizada.

Extrapolamos ahora estas ideas al caso general, donde a partir de un hiperrectángulo D^m potencialmente óptimo la bisección generará dos nuevos hiperrectángulos, denotados por D^{m+1} y D^{m+2} . De esta manera, la expresión general de los coeficientes s_0 y s_j queda definida por las siguientes ecuaciones

$$s_0 = s_0 + \frac{|f(r^{m+2}) - f(r^{m+1})|}{\|r^{m+2} - r^{m+1}\|} + \frac{|f(t^{m+1}) - f(t^{m+2})|}{\|r^{m+1} - r^{m+2}\|} \quad (3.28)$$

$$s_j = s_j + \frac{|g_j(r^{m+2}) - g_j(r^{m+1})|}{\|r^{m+2} - r^{m+1}\|} + \frac{|g_j(t^{m+1}) - g_j(t^{m+2})|}{\|t^{m+1} - t^{m+2}\|}, j = 1, 2, \dots, m. \quad (3.29)$$

Los mismos serán calculados en el Algoritmo 3.10.

3.3.3. Selección de hiperrectángulos potencialmente óptimos

Recordemos que la sumatoria $\sum_{j=1}^m c_j \max\{g_{pj}, 0\}$, que forma parte de la función auxiliar en la ecuación (3.15) del algoritmo DIRECT en la Sección 3.1.3.1, representa una suma ponderada de las violaciones a las restricciones en el punto central del hiperrectángulo p . Es decir, g_{pj} es el valor de la restricción j en el centro del hiperrectángulo p . En el caso de DIRECT, esta idea resulta bastante intuitiva dado que cada hiperrectángulo mantiene una *relación 1-1* con su centro, lo cual significa que existe un único punto a través del cual un hiperrectángulo es identificado, y ese punto es su centro. Por otra parte como vimos anteriormente, en el algoritmo BIRECT todo hiperrectángulo es identificado con dos puntos en lugar de uno (denotados en esta tesis con r y t).

Hasta el momento, con la adaptación de los coeficientes s_j presentada en el apartado anterior, el cálculo de los coeficientes c_j es bastante directo y consiste sólo en aplicar la fórmula (3.18). El dilema entonces surge en cómo adaptar la expresión $\max\{g_{pj}, 0\}$, y por lo tanto la función auxiliar, cuando un procedimiento de bisección es aplicado. Resolver este punto es muy relevante dado que tiene implicación directa en la selección de hiperrectángulos potencialmente óptimos como veremos más adelante.

Por un lado, es interesante observar que s_0 y s_j (y en consecuencia c_j), por la forma en la que están contruidos, toman el mismo valor tanto en r como en t . Por otro lado, dado que ambos puntos son siempre distintos, al ser evaluados en las restricciones darán valores distintos, pudiendo ser ambos factibles, ambos infactibles o uno factible y el otro no. Este hecho es importante pues implica que las expresiones

$$\sum_{j=1}^m c_j \max\{g_{rj}, 0\} \text{ y } \sum_{j=1}^m c_j \max\{g_{tj}, 0\}^3$$

serán diferentes a menos que ambos puntos sean factibles (en cuyo caso tomarán el valor 0). Entre los dos casos *extremos*, el menos deseable es precisamente cuando ambos son infactibles pues implica que el algoritmo *está lejos* de la solución óptima. En esta situación, el primer cambio con respecto al algoritmo DIRECT consiste en considerar el punto donde la suma ponderada de la violación de las restricciones sea mínima, es decir

$$\min_{r,t} \left\{ \sum_{j=1}^m c_j \max\{g_{rj}, 0\}, \sum_{j=1}^m c_j \max\{g_{tj}, 0\} \right\}. \quad (3.30)$$

De esta manera, cada hiperrectángulo pasará a ser representado por un único punto: aquel que se encuentre más cerca de la factibilidad. Algo interesante para notar es que mientras para un hiperrectángulo este punto podría ser el asociado a r , para otro el asociado a t . Lo atractivo de esto es que, a diferencia de DIRECT que *se mueve siempre del centro de un hiperrectángulo al centro de otro*, en esta nueva propuesta algorítmica se *prioriza* el punto menos infactible. De esta manera, el movimiento hacia los nuevos iterados puede darse en diferentes regiones de los hiperrectángulos, las cuales dependerán de la ubicación del r o t resultante de (3.30).

Con esto en mente, el paso siguiente del proceso es definir el ***esquema de selección de los hiperrectángulos potencialmente óptimos***. De manera análoga al Algoritmo 3.4 esta selección tendrá dos situaciones posibles que se basan en la presencia o ausencia de puntos factibles. Como se observará en breve en el Algoritmo 3.6, en caso de que en la iteración previa ningún punto factible haya sido encontrado se procederá a seleccionar un único hiperrectángulo como potencialmente óptimo y será aquel que, dentro de todos los hiperrectángulos disponibles, esté más cerca de cumplir con las restricciones, es decir, dónde la violación a las restricciones tome el valor mínimo. Por otro lado en la presencia de factibilidad, la selección se volverá más compleja, pero al mismo tiempo será el caso más deseable, y consistirá el cálculo de una cápsula convexa inferior que se explica a continuación.

3.3.3.1. La cápsula convexa inferior

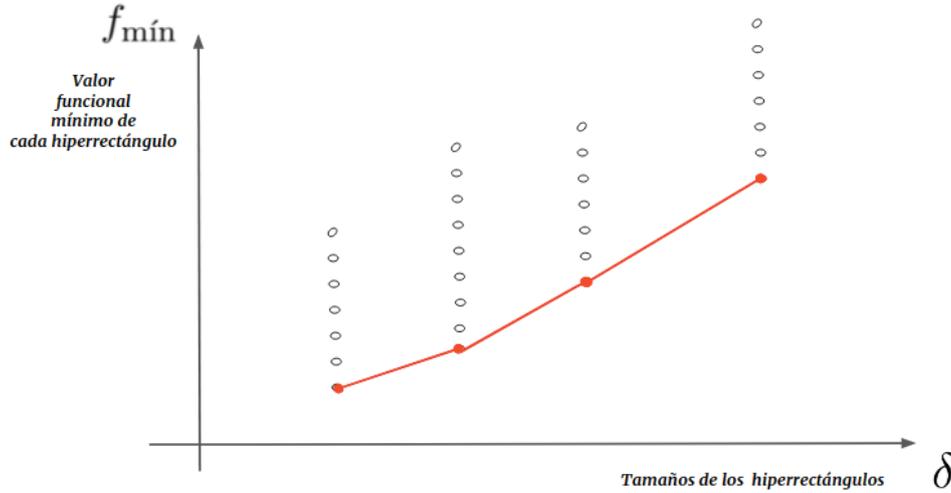
Ante la presencia de restricciones generales, la selección de hiperrectángulos potencialmente óptimos del nuevo algoritmo propuesto se acerca más a la idea de Jones [23] que a la forma presentada por Paulavičius, Chiter y Žilinskas [43], aunque toma un poco de ambos enfoques. En el fondo, ambos desarrollos decantan en la selección del contorno o casco inferior obtenido a partir de una nube de puntos o gráficas de funciones ubicadas en el plano coordenado xy . La principal diferencia se encuentra en lo que representan los ejes x e y en cada caso. Jones, por su parte, considera las tasas de cambio de h_i (eje y) en función de un valor particular de f^* (eje x) como se mostró en la Figura 3.6. Paulavičius et al., en cambio, trabajan con las duplas (δ, f_{\min}) , representando con δ el tamaño del hiperrectángulo y f_{\min} el valor funcional mínimo entre r y t como muestra la Figura 3.11.

Profundizando la última parte del párrafo anterior, podemos observar que existe una correlación entre lo desarrollado por Paulavičius et al. para el algoritmo BIRECT irrestricto y lo explicado hasta el momento sobre el funcionamiento del nuevo algoritmo BIRECT con restricciones generales. Para los autores de [43], la cápsula convexa es determinada teniendo en

³Por simplicidad, transformaremos la expresión $\max\{g_{pj}, 0\}$ a $\max\{g_{rj}, 0\}$, omitiendo el subíndice p y agregando el punto sobre el cual se está evaluando la sumatoria, dado que, por un lado, es claro que siempre se trabaja con un hiperrectángulo a la vez y por otro, al tener dos puntos en cuenta resulta necesario hacer una diferenciación.

cuenta el valor funcional mínimo entre los puntos r y t en cada hiperrectángulo. Es decir, el punto *más óptimo* entre ambos será el representante del hiperrectángulo. Como se detalló en la sección anterior, en el algoritmo BIRECT con restricciones generales cada hiperrectángulo es representado por el punto *más factible* entre sus respectivos r y t . De esta manera, se podría decir que la extensión de BIRECT irrestricto a BIRECT con restricciones sigue una misma línea en la cual de *medir* optimalidad se pasa a *medir* factibilidad. Como se verá a continuación, y retomando lo explicado en la Sección 3.1.3.1, la función auxiliar a partir de la cual se obtiene la cápsula convexa balancea las dos situaciones (factibilidad y optimalidad).

Figura 3.11: Selección de hiperrectángulos potencialmente óptimos - BIRECT



La función auxiliar definida para el Algoritmo BIRECT con restricciones generales es bastante similar a la función auxiliar del Algoritmo DIRECT con restricciones generales (Sección 3.1.3) salvo por dos detalles distintivos. El primero, mencionado en la sección anterior, sólo considerará el punto, entre r y t , en el cual la suma ponderada de las violaciones a las restricciones sobre todo el hiperrectángulo sea mínima, el otro no será tenido en cuenta. El segundo cambio se encuentra relacionado con la definición del tamaño del hiperrectángulo. Dado que la base principal de este nuevo algoritmo es el proceso de bisección en el cual se siempre se encuentran involucrados dos puntos r y t , lo más acertado es considerar como definición del tamaño de los hiperrectángulos la dada en [43], es decir

$$\delta_i = \frac{2}{3} \|t^i - r^i\|. \quad (3.31)$$

En base a estas aclaraciones, dado un hiperrectángulo i construido a partir de los puntos r y t , la función auxiliar asociada a él se define por

$$h_i(f^*) = \frac{\max(f_i - f^*, 0) + \min \left\{ \sum_{j=1}^m c_j \max(g_{rj}, 0), \sum_{j=1}^m c_j \max(g_{tj}, 0), 0 \right\}}{\delta_i} \quad (3.32)$$

Teniendo en cuenta que f^* representa el punto óptimo global del problema y es usualmente desconocido, el mismo criterio utilizado en [23] será considerado en este algoritmo. Es decir, se seleccionarán todos los posibles hiperrectángulos que minimizan h_i para algún $f^* \leq f_{\text{mín}} - \varepsilon$.

Precisamente por esto, esta función es calculada sólo en caso de encontrar factibilidad ya que en esta situación es posible definir f_{\min} y f_i en la expresión (3.32).

De esta manera, el Algoritmo BIRECT con restricciones generales considera la tasa de cambio de h_i como función de f^* para algún $f^* \leq f_{\min} - \varepsilon$ y para cada hiperrectángulo $i \in \mathbb{I}$. *El subconjunto de hiperrectángulos que participan en la cápsula convexa inferior, es decir las curvas formadas a partir de la intersección de estas tasas de cambio, a izquierda de $f_{\min} - \varepsilon$ son precisamente los hiperrectángulos potencialmente óptimos buscados.* Así, la selección de hiperrectángulos es similar a DIRECT con algunos cambios en la función h_i relacionados con el proceso de bisección. En la siguiente sección se profundiza este procedimiento con más detalle.

3.3.3.2. El proceso de selección paso a paso

Como se puede observar en las Figuras 3.6 y 3.11 hay una diferencia sustancial en la determinación de los hiperrectángulos que forman parte de la cápsula convexa inferior entre ambos algoritmos. En el caso de BIRECT irrestricto, Figura 3.11, el tener solo puntos o pares ordenados hace que identificar los puntos que *están por debajo* de la nube sea más simple. Los autores de [43] utilizan para ello el algoritmo de escaneo de Graham modificado. En el caso de la Figura 3.6, como mencionamos previamente, los puntos se originan a partir de las intersecciones de las funciones h_i asociadas a cada hiperrectángulo i , las cuales tienen una forma bastante particular. Dado que la suma ponderada de las violaciones a las restricciones, segundo término de (3.32), no depende de los valores funcionales del hiperrectángulo, representados en el eje x , este valor es siempre constante. Por otro lado, si miramos el primer término de (3.32), se observa que cuando $f_i \leq f^*$ el mismo toma el valor 0. Por lo tanto la función auxiliar es la constante recién mencionada. En el caso inverso, esto es $f_i > f^*$, h_i toma valores cada vez más grandes. En resumen, h_i son rectas constantes a la derecha de f_i y rectas con pendiente negativa igual a $-\frac{1}{\delta_i}$ a la izquierda de f_i .

El procedimiento de selección de hiperrectángulos potencialmente óptimos utilizado por el algoritmo BIRECT con restricciones generales es descrito a través de los Algoritmos 3.6 y 3.7. Como vimos en la la sección 3.3.3, el Algoritmo 3.6 (Identificación de hiperrectángulos potencialmente óptimos) es el encargado de determinar los representantes de cada hiperrectángulo así como la suma ponderada de la violación a las restricciones. En el caso de encontrarse en una situación de infactibilidad total, es decir donde ni r ni t son factibles, con este algoritmo basta para realizar la selección de hiperrectángulos.

En cambio, cuando un punto factible es encontrado, el proceso de selección de hiperrectángulos potencialmente óptimos se realiza utilizando los Algoritmos 3.6 y 3.7 de forma consecutiva. Luego de la determinación de los representantes para cada hiperrectángulo, el siguiente paso consiste en establecer el *valor óptimo global aproximado* a partir del cual se desea minimizar. Moverse levemente por debajo del valor mínimo de referencia es un buen punto de partida. Es por ello que al igual que Jones en [23] adoptamos $f^* = f_{\min_\varepsilon} = f_{\min} - \varepsilon$ con $\varepsilon = \max\{10^{-4}|f_{\min}|, 10^{-8}\}$.

La idea detrás de este punto es muy atractiva pues cualquier valor más pequeño que f^* , digamos \bar{f} , es un candidato a mínimo. Además al trazar una recta vertical a partir del punto $(\bar{f}, 0)$, la primera curva h_i encontrada determina el hiperrectángulo potencialmente óptimo asociado a \bar{f} . Es aquí donde se encuentra la clave del proceso de selección de hiperrectángulos potencialmente óptimos: *para cada valor más chico que f^* encontrar su hiperrectángulo asociado, y en la unión de todos estos hiperrectángulos generados a partir de cada valor posible de f^* , se encontrará el punto que mejore el valor funcional mínimo obtenido hasta el momento.*

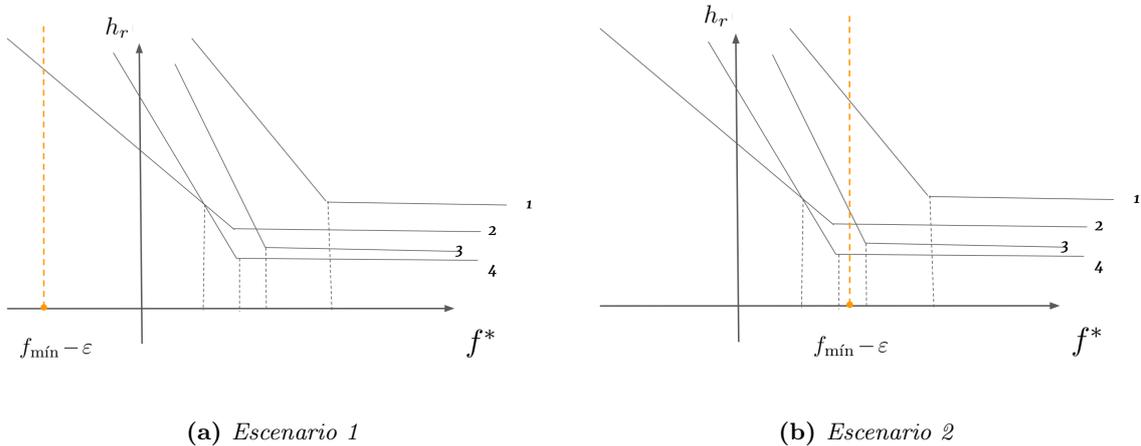
Por ahora el esquema de selección de hiperrectángulos potencialmente óptimos ha sido descrito desde el punto de vista del proceso de optimización. Si bien, ahora sabemos que debemos movernos a izquierda de f^* y *levantar la vista* para determinar cual es el hiperrectángulo más

potencialmente óptimo asociado a este valor, desde el punto de vista algorítmico existen ciertos detalles a tener en cuenta que se describen a continuación.

Habiendo definido $f_{\text{mín}_\varepsilon}$ y dado un conjunto de hiperrectángulos \mathcal{I} a partir del cual se obtiene toda la información necesaria (su punto representativo, su tamaño, el valor funcional y la suma ponderada de la violación a las restricciones en ese punto), dos escenarios diferentes se podrían presentar:

1. Que los valores funcionales de los representantes sean todos mayores que el valor óptimo global aproximado $f_{\text{mín}_\varepsilon}$.
2. Que en al menos uno de los hiperrectángulos, el valor funcional de su representante sea menor que el valor óptimo global aproximado $f_{\text{mín}_\varepsilon}$.

Identificar estas situaciones es importante ya que marcará la forma de trabajar con las rectas y las constantes de las funciones h_i . Ambas situaciones se encuentran ejemplificadas en las Figuras 3.12a y 3.12b.



Escenario 1

En el primer caso, el hecho de que todos los valores funcionales sean más grandes que $f_{\text{mín}_\varepsilon}$ implica que hasta el momento no hay mejora posible sobre f y claramente $f_{\text{mín}_\varepsilon}$ es la mejor opción (las rectas constantes de todos los hiperrectángulos se encuentran a la derecha de $f_{\text{mín}_\varepsilon}$). El siguiente paso, en función de lo desarrollado párrafos anteriores, es identificar cuál es el hiperrectángulo más óptimo asociado a este valor. Para realizar esto, la clave está en volver a la Figura 3.12a y observar que la tarea consiste en identificar, a partir de todas las rectas con pendientes negativas asociadas a cada uno de los hiperrectángulos, la que se encuentra más cerca del eje x . Esto tiene relación directa con lo que menciona Jones en [23] acerca del objetivo de la función auxiliar:

“Para llevar a cabo un proceso de minimización global es necesario que la función objetivo sufra algún tipo de cambio o movimiento, el cual expresamos a través de la función h_r , pero este cambio debe ser al mismo tiempo lo más moderado posible (mínimo)”.

En términos matemáticos, el trabajo consiste en encontrar el valor mínimo entre las rectas dadas por las ecuaciones

$$\overline{\text{mín cons}}_r^i = \frac{1}{\delta_i} (f_{\text{mín}_\varepsilon} - \overline{f}_{\text{mín}}^i), i \in \mathbb{I} \quad (3.33)$$

donde

$$\overline{\text{mín cons}}_{rt}^i = \text{mín} \left\{ \sum_{j=1}^m c_j \text{máx}(g_{rj}, 0), \sum_{j=1}^m c_j \text{máx}(g_{tj}, 0) \right\} \quad (3.34)$$

es el valor mínimo de las sumas ponderadas a las violaciones de las restricciones dada por r y t en el hiperrectángulo i y $\overline{f}_{\text{mín}}^i$ es el valor funcional en el punto originado del cálculo anterior. Además el punto donde la suma ponderada mencionada es mínima es denotado por $\overline{\text{mín}}_{rt}^i$, es decir

$$\overline{\text{mín}}_{rt}^i = \arg \text{mín}_{r,t} \left\{ \text{mín} \left\{ \sum_{j=1}^m c_j \text{máx}(g_{rj}, 0), \sum_{j=1}^m c_j \text{máx}(g_{tj}, 0) \right\} \right\}. \quad (3.35)$$

Si bien esta notación puede resultar algo engorrosa, será necesaria más adelante en la presentación del algoritmo. Notar que el resultado de la expresión (3.34) es un valor ubicado en el eje y . *El hiperrectángulo i donde (3.33) alcanza su valor mínimo determina entonces el primer y único hiperrectángulo potencialmente óptimo en este escenario.*

Escenario 2

El segundo caso es similar al anterior con la consideración de que al trazar la recta vertical desde $(f_{\text{mín}_\varepsilon}, 0)$ es posible encontrar tanto rectas con pendiente negativa como constantes. Esto significa que para aquellos valores funcionales más grandes que $f_{\text{mín}_\varepsilon}$ los hiperrectángulos ya habrán tomado la forma de rectas decrecientes mientras que para los menores todavía serán constantes como muestra la Figura 3.12b. En esta situación es importante observar dos puntos claves. El primero es que para los valores $f_{\text{mín}} \leq f_{\text{mín}_\varepsilon}$, resulta imprescindible realizar un reordenamiento de los $f_{\text{mín}}$ con el objetivo de permitir el movimiento de derecha a izquierda sobre el eje horizontal x de manera tal de ir minimizando el valor de la función f a partir de $f_{\text{mín}_\varepsilon}$. El segundo punto y fuertemente relacionado con el anterior, es que si bien los hiperrectángulos asociados a los valores $f_{\text{mín}} > f_{\text{mín}_\varepsilon}$ podrían llegar a ser potencialmente óptimos, como veremos en breve, sus valores funcionales no son tenidos en cuenta como referencia por ir en contra del proceso de minimización. Por lo tanto en este escenario, si f_r es el valor más próximo a izquierda de $f_{\text{mín}_\varepsilon}$, matemáticamente la tarea consiste en encontrar el valor mínimo entre los máximos de los siguientes valores de cada hiperrectángulo $i \in \mathbb{I}$

$$\text{máx} \left\{ \overline{\text{mín cons}}_{rt}^i, \overline{\text{mín cons}}_{rt}^i - \frac{1}{\delta_i} (f_r - \overline{f}_{\text{mín}}^i) \right\}. \quad (3.36)$$

Observemos que dado que f_r es alguno de los $\overline{f}_{\text{mín}}^i$, existirá al menos un hiperrectángulo en el cual la resta del segundo término será igual a 0. Análogamente al caso anterior, *el índice i resultante de obtener el mínimo de (3.36) de todos los $i \in \mathbb{I}$ proporciona el primer hiperrectángulo potencialmente óptimo.* En este caso, el proceso iterativo se repite cambiando f_r por el valor $f_{\text{mín}}$ más próximo a la izquierda y así hasta haber considerado todos los $f_{\text{mín}} \leq f^*$. *Por lo tanto, este escenario podría dar lugar a más de un hiperrectángulo potencialmente óptimo.*

Habiendo considerado ambos escenarios y detectado el conjunto de hiperrectángulos potencialmente óptimos hasta el momento, la segunda etapa del proceso consiste en continuar con la búsqueda de todos los hiperrectángulos potencialmente óptimos. El punto de partida es $f_{\text{mín}_\varepsilon}$ en el caso del escenario 1 y f_r (el mínimo valor entre todos los $f_{\text{mín}} \leq f_{\text{mín}_\varepsilon}$) en el caso del escenario 2. En esta instancia, todas las funciones auxiliares h_i de cada hiperrectángulo i son ya rectas con pendiente negativa igual a $-\frac{1}{\delta_i}$. Por lo tanto, el trabajo consistirá en obtener la unión de fragmentos de rectas que se encuentran *por debajo de todas las funciones h_i y sus respectivos*

hiperrectángulos asociados, de manera de lograr algo similar a lo visto con DIRECT en la Figura 3.6.

Dado que el interés siempre está puesto en *aquellas rectas (asociadas a cada hiperrectángulo) que se encuentran más cerca del eje x* , el primer paso consiste en identificar y reordenar las mismas de acuerdo a su orden de aparición de abajo hacia arriba. Esto se realiza utilizando los valores calculados en (3.33) y (3.36). Luego, para poder obtener los puntos de corte entre las mismas, usamos la fórmula

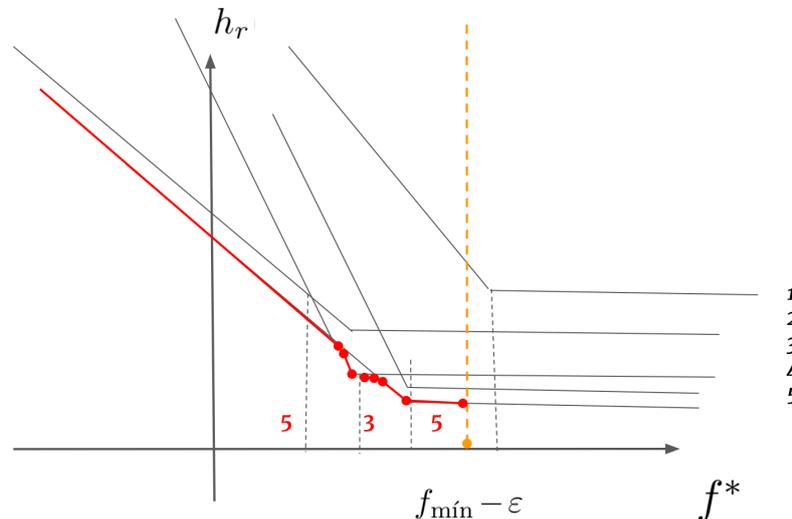
$$b_i = \frac{\text{mín cons}_i}{\delta_i} + \frac{1}{\delta_i} f_{\text{mín}}^i \quad (3.37)$$

para identificar sus ordenadas al origen.

Finalmente, tomando como punto de partida el hiperrectángulo con la componente más pequeña, en base a los resultados de (3.33) y (3.36), se realiza la intersección de este con aquellos hiperrectángulos cuyo tamaño es más grande que el tamaño de este hiperrectángulo de partida, los demás son descartados. La explicación de este punto es que mientras más pequeño es δ_i , más negativo es $-\frac{1}{\delta_i}$ (la pendiente de la recta) y a medida que nos vamos moviendo por la izquierda en el eje de las abscisas estas rectas tienden a hacerse cada vez más verticales. Por lo tanto, la búsqueda siempre se centrará en las rectas “menos verticales” dado que son las candidatas a formar parte de la cápsula convexa inferior de la gráfica. Para identificar cuál es el siguiente hiperrectángulo potencialmente óptimo, basta obtener el máximo valor entre todas las intersecciones calculadas. Si este valor se encuentra a izquierda del valor funcional de partida ($f_{\text{mín}_\varepsilon}$ o f_r de acuerdo al escenario inicial) significa que el hiperrectángulo asociado a él mejora el valor de la función y por lo tanto es el nuevo hiperrectángulo potencialmente óptimo. El procedimiento iterativo continua de esta manera, partiendo del valor de esta intersección (mínimo valor funcional hasta el momento) y repitiendo el proceso descrito en las líneas anteriores hasta encontrar el hiperrectángulo con tamaño máximo. Este hiperrectángulo será *el que quedará por debajo de todos los demás* cuando $f \rightarrow \infty$ y será el último seleccionado. Cada uno de los pasos de este esquema de selección de hiperrectángulos se encuentra formalizado en el Algoritmo 3.7.

Antes de cerrar este apartado es interesante mencionar un punto que también fue expuesto por Jones en [23]. Un hiperrectángulo podría ser potencialmente óptimo más de una vez como se ve en la Figura 3.12, sin embargo en este caso dicho hiperrectángulo sólo debe ser considerado una única vez.

Figura 3.12: Caso ejemplo: Cuando hiperrectángulo aparece más de una vez en la cápsula convexa



3.3.4. Convergencia y algoritmo

Suponiendo que la función objetivo y las funciones que definen las restricciones en el problema (3.22) son continuas en un entorno del optimizador global, el nuevo algoritmo BIRECT con restricciones generales hereda las propiedades de convergencia de los algoritmos BIRECT para restricciones de cotas y DIRECT para restricciones generales, presentados en las Secciones 3.1.2 y 3.1.3 respectivamente.

Por un lado, como se mencionó anteriormente, a medida que el valor de f es cada vez menor, el algoritmo propuesto prioriza en la selección aquellos hiperrectángulos de mayor tamaño. Por otro lado, la variable de ramificación (3.13) asegura que la partición se realice en un solo lado por iteración. De esta manera, siguiendo la característica de los algoritmos de la familia DIRECT, cada hiperrectángulo reducirá su tamaño a través de las sucesivas iteraciones y sus representantes formarán un subconjunto denso. Finalmente, la continuidad en la función objetivo y las restricciones en un entorno del optimizador global, aseguran que, en el límite de las iteraciones, los puntos muestreados se encontrarán lo suficientemente cerca de dicho óptimo.

Antes de presentar la nueva propuesta algorítmica es importante resaltar que cada nuevo iterado obtenido por el algoritmo se encuentra siempre sobre el hipercubo unitario \bar{D} . Esto significa que las restricciones de cotas del problema son satisfechas en todo momento. Por otro lado, el hipercubo D sólo es considerado cada vez que se requiere evaluar la función objetivo, para lo cual se utiliza la transformación lineal inversa de T dada por $T^{-1} : \bar{D} \rightarrow D$ con $T^{-1}(x) = A^{-1}x + l$ y $A^{-1} = \text{diag}(u_1 - l_1, u_2 - l_2, \dots, u_n - l_n)$.

A continuación, en los Algoritmos 3.5 – 3.10 se refleja todo el proceso iterativo realizado por el nuevo Algoritmo BIRECT para problemas con restricciones generales.

Algoritmo 3.5: BIRECT con restricciones generales

Entrada:
 $f : \mathbb{R}^n \rightarrow \mathbb{R}$ Función a minimizar.
 $g : \mathbb{R}^m \rightarrow \mathbb{R}$ Restricciones generales del problema de minimización.
 $l, u \in \mathbb{R}^n$: Restricciones sobre las variables del problema de minimización.
 $FE_{\text{máx}}$: Máximo número de evaluaciones funcionales permitido.
 $K_{\text{máx}}$: Máximo número de iteraciones permitido.
 tol : Tolerancia para el test de convergencia.
 f_{sol} : Solución del problema de optimización (en caso que exista).

Salida:
 $f_{\text{mín}}$: Valor funcional mínimo encontrado.
 $x_{\text{mín}}$: Punto factible del problema donde se da el valor funcional mínimo.

```

1  $a, b \leftarrow l, u$  // Aplicando la transformación lineal (3.9)
2  $FE \leftarrow 0$  // Inicialización de las evaluaciones funcionales
3  $IsFactible \leftarrow \text{False}$ 
4 para  $j \leftarrow 0$  a  $m$  hacer // Inicialización de  $s$ 
5    $s_j \leftarrow 0$ 
6  $k \leftarrow 1$  // Inicialización de las iteraciones
7 /* Generación del primer hiperrectángulo */
8  $a^1 \leftarrow a$ 
9  $b^1 \leftarrow b$ 
10  $r^1 \leftarrow \left( \frac{1}{3}, \frac{1}{3}, \dots, \frac{1}{3} \right)$ 
11  $t^1 \leftarrow \left( \frac{2}{3}, \frac{2}{3}, \dots, \frac{2}{3} \right)$ 
12 Evaluar  $f$  en  $r^1$  y  $t^1$ 
13  $FE \leftarrow FE + 2$ 
14  $\mathbb{I}_1 \leftarrow \{1\}$ 
15  $\delta_1 \leftarrow \frac{2}{3} \|b^1 - a^1\|$  // Tamaño del primer hiperrectángulo
16  $\delta_{\mathbb{I}_k} \leftarrow \delta_1$ 
17 /* Chequeo de factibilidad inicial */
18 si  $r^1 \in \Omega$  y  $t^1 \in \Omega$  entonces
19    $f_{\text{mín}} \leftarrow \min\{f(r^1), f(t^1)\}$ 
20    $x_{\text{mín}} \leftarrow \arg \min_{x \in \{r^1, t^1\}} f(x)$ 
21    $IsFactible \leftarrow \text{True}$ 
22 si no, si  $r^1 \in \Omega$  entonces
23    $f_{\text{mín}} \leftarrow f(r^1)$ 
24    $x_{\text{mín}} \leftarrow r^1$ 
25    $IsFactible \leftarrow \text{True}$ 
26 si no, si  $t^1 \in \Omega$  entonces
27    $f_{\text{mín}} \leftarrow f(t^1)$ 
28    $x_{\text{mín}} \leftarrow t^1$ 
29    $IsFactible \leftarrow \text{True}$ 
30 /* Proceso iterativo */
31 mientras  $FE < FE_{\text{máx}}$  y  $k < K_{\text{máx}}$  hacer
32    $\mathbb{P}_k \leftarrow \text{Algoritmo 3.6}$  // Selección de hiperrectángulos potencialmente óptimos
33    $\mathbb{I}_k \leftarrow \mathbb{I}_k \setminus \mathbb{P}_k$ 
34    $\mathbb{I}_k, f_{\text{mín}}, x_{\text{mín}}, FE \leftarrow \text{Algoritmo 3.8}$  // Bisección de hiperrectángulos
35   /* Test de Convergencia */
36   si  $\exists f_{\text{sol}}$  y  $f_{\text{sol}} = 0$  entonces
37      $err \leftarrow f_{\text{mín}}$ 
38   si no, si  $\exists f_{\text{sol}}$  y  $f_{\text{sol}} \neq 0$  entonces
39      $err \leftarrow \left| \frac{f_{\text{mín}} - f_{\text{sol}}}{f_{\text{sol}}} \right|$ 
40   si  $err \leq tol$  entonces
41     devolver  $f_{\text{mín}}, x_{\text{mín}}$ 
42 devolver  $f_{\text{mín}}, x_{\text{mín}}$ 

```

Algoritmo 3.6: Identificación de hiperrectángulos potencialmente óptimos

Entrada: s, k, \mathbb{I}_k, FE
Salida: \mathbb{P}_k

- 1 **para** $j \leftarrow 1$ **a** m **hacer** // Actualización de c
- 2 $c_j \leftarrow \frac{s_0}{\max\{s_j, 10^{-30}\}}$
- 3 **para** $p \in \mathbb{I}_k$ **hacer** // Para cada hiperrectángulo elegir el punto (r ó t) menos infactible
- 4 $\overline{\text{mín cons}_{rt}}^p \leftarrow$ Ecuación (3.34) aplicada al hiperrectángulo p
- 5 $\overline{\text{mín}_{rt}}^p \leftarrow$ Ecuación (3.35) aplicada al hiperrectángulo p
- 6 **si** $IsFactible = False$ **entonces** // Ningún punto factible ha sido encontrado
- 7 $\mathbb{P}_k \leftarrow \arg \min_{p \in \mathbb{I}_k} \left\{ \overline{\text{mín cons}_{rt}}^p \right\}$ // Único hiperrectángulo potencialmente óptimo
- 8 **en otro caso** // Existe al menos un punto factible
- 9 $\varepsilon \leftarrow \max\{10^{-4}|f_{\text{mín}}|, 10^{-8}\}$
- 10 $f_{\text{mín}_\varepsilon} \leftarrow f_{\text{mín}} - \varepsilon$
- 11 $\mathbb{P}_k \leftarrow$ Algoritmo 3.7 // Cápsula convexa
- 12 **devolver** \mathbb{P}_k

Algoritmo 3.7: Cápsula convexa

```

Entrada:  $\mathbb{I}_k, f_{\min_\varepsilon}$ 
Salida:  $\mathbb{P}_k$ 
1 si  $|\mathbb{I}_k| = 1$  entonces
2    $\mathbb{P}_k \leftarrow \mathbb{I}_k$ 
3 en otro caso
4    $\text{mín cons}_{\mathbb{I}_k} \leftarrow \bigcup_{p \in \mathbb{I}_k} \overline{\text{mín cons}_{\text{rt}}^p}$ 
5    $\text{mín}_{\mathbb{I}_k} \leftarrow \bigcup_{p \in \mathbb{I}_k} \overline{\text{mín}_{\text{rt}}^p}$ 
6    $\delta_{\mathbb{I}_k} \leftarrow \bigcup_{p \in \mathbb{I}_k} \delta_p$ 
7    $f_{\text{mín}_{\mathbb{I}_k}} \leftarrow f(\text{mín}_{\text{rt}_{\mathbb{I}_k}})$ 
8   /* Reordenar todos los valores de acuerdo al orden decreciente de  $f_{\text{mín}_{\mathbb{I}_k}}$  */
9    $(f_{\text{mín}_{\mathbb{I}_k}}, \mathbb{I}_k, \text{mín cons}_{\mathbb{I}_k}, \delta_{\mathbb{I}_k}) \leftarrow \text{sort}(f_{\text{mín}_{\mathbb{I}_k}}, \mathbb{I}_k, \text{mín cons}_{\mathbb{I}_k}, \delta_{\mathbb{I}_k})$ 
10  /* Filtrar aquellos hiperrectángulos tales que  $\overline{f_{\text{mín}}^p} \leq f_{\text{mín}_\varepsilon}$  */
11   $f_{\text{reduced}} \leftarrow \{ \overline{f_{\text{mín}}^p} \leq f_{\text{mín}_\varepsilon}, p \in \mathbb{I}_k \}$ 
12  /* PRIMERA ETAPA */
13  si  $f_{\text{reduced}} \neq \emptyset$  entonces // Existen valores funcionales menores que  $f_{\text{mín}_\varepsilon}$ 
14    para  $i \leftarrow 1$  a  $|f_{\text{reduced}}|$  hacer
15      para  $j \leftarrow 1$  a  $|f_{\text{mín}_{\mathbb{I}_k}}|$  hacer // Aplicar fórmula (3.36)
16         $d_j \leftarrow \text{máx} \left\{ (\text{mín cons}_{\mathbb{I}_k})_j, (\text{mín cons}_{\mathbb{I}_k})_j - \frac{1}{(\delta_{\mathbb{I}_k})_j} (f_{\text{reduced}_i} - (f_{\text{mín}_{\mathbb{I}_k}})_j) \right\}$ 
17       $m \leftarrow \arg \text{mín}_{1 \leq j \leq |f_{\text{mín}_{\mathbb{I}_k}}|} \{d_j\}$ 
18       $\mathbb{P}_k \leftarrow \mathbb{I}_m$  // Primer hiperrectángulo potencialmente óptimo de la iteración  $k$ 
19     $f_{\text{last}} \leftarrow f_{\text{reduced}_i}$ 
20  en otro caso // Todos los valores funcionales son mayores que  $f_{\text{mín}_\varepsilon}$ 
21     $f_{\text{last}} \leftarrow f_{\text{mín}_\varepsilon}$ 
22    para  $j \leftarrow 1$  a  $|f_{\text{mín}_{\mathbb{I}_k}}|$  hacer // Aplicar la fórmula (3.33)
23       $d_j \leftarrow (\text{mín cons}_{\mathbb{I}_k})_j - \frac{1}{(\delta_{\mathbb{I}_k})_j} (f_{\text{mín}_\varepsilon} - (f_{\text{mín}_{\mathbb{I}_k}})_j)$ 
24     $m \leftarrow \arg \text{mín}_{1 \leq j \leq |f_{\text{mín}_{\mathbb{I}_k}}|} \{d_j\}$ 
25     $\mathbb{P}_k \leftarrow \mathbb{I}_m$  // Primer hiperrectángulo potencialmente óptimo de la iteración  $k$ 
26  /* SEGUNDA ETAPA */
27  /* Reordenar todos los valores de acuerdo al orden creciente de  $d_j$  */
28   $(d, f_{\text{mín}_{\mathbb{I}_k}}, \mathbb{I}_k, \text{mín cons}_{\mathbb{I}_k}, \delta_{\mathbb{I}_k}) \leftarrow \text{sort}(d, f_{\text{mín}_{\mathbb{I}_k}}, \mathbb{I}_k, \text{mín cons}_{\mathbb{I}_k}, \delta_{\mathbb{I}_k})$ 
29  para  $i \leftarrow 1$  a  $|f_{\text{mín}_{\mathbb{I}_k}}|$  hacer
30     $b_i \leftarrow (\text{mín cons}_{\mathbb{I}_k})_i + \frac{1}{(\delta_{\mathbb{I}_k})_i} (f_{\text{mín}_{\mathbb{I}_k}})_i$  // Aplicar la fórmula (3.37)
31   $b_{\text{start}} \leftarrow b_1$ 
32   $\delta_{\text{start}} \leftarrow \delta_1$ 
33  mientras  $\delta_{\text{start}} \neq \max_{1 \leq j \leq |f_{\text{mín}_{\mathbb{I}_k}}|} \delta_j$  hacer
34     $(\delta_{\mathbb{I}_k}, b, \mathbb{I}_k) \leftarrow \{(\delta_{\mathbb{I}_k}, b, \mathbb{I}_k) : (\delta_{\mathbb{I}_k})_j > \delta_{\text{start}} + 10^{-3}, j = 1, 2, \dots, |f_{\text{mín}_{\mathbb{I}_k}}|\}$ 
35    para  $j \leftarrow 1$  a  $|\mathbb{I}_k|$  hacer
36       $\text{intersec}_j \leftarrow \frac{(b_j - b_{\text{start}})\delta_{\text{start}}\delta_j}{-\delta_j + \delta_{\text{start}}}$ 
37     $z \leftarrow \arg \text{máx}_{1 \leq i \leq |\mathbb{I}_k|} \{\text{intersec}_i : \text{intersec}_i \leq f_{\text{last}}\}$ 
38     $\mathbb{P}_k \leftarrow \mathbb{P}_k \cup \mathbb{I}_z$  // Actualización de hiperrectángulos potencialmente óptimos
39     $f_{\text{last}} \leftarrow \text{intersec}_z$ 
40     $b_{\text{start}} \leftarrow b_z$ 
41     $\delta_{\text{start}} \leftarrow \delta_z$ 
42  devolver  $\mathbb{P}_k$ 

```

Algoritmo 3.8: Bisección de hiperrectángulos

Entrada: $\mathbb{P}_k, f_{\min}, x_{\min}, FE$
Salida: $\mathbb{I}_k, f_{\min}, x_{\min}, FE$

- 1 **para** $i \in \mathbb{P}_k$ **hacer**
- 2 $br \leftarrow$ Ecuación (3.13) aplicada al hiperrectángulo i // Variable de ramificación
- 3 /* Inicialización de a y b en las nuevas particiones */
- 4 /* Las nuevas particiones (hiperrectángulos) son denotadas por $FE+1$ y $FE+2$ */
- 5 $a^{FE+2} = a^{FE+1} \leftarrow a^i$
- 6 $b^{FE+2} = b^{FE+1} \leftarrow b^i$
- 7 /* Inicialización de r y t en las nuevas particiones */
- 8 $r^{FE+2} = r^{FE+1} \leftarrow r^i$
- 9 $t^{FE+2} = t^{FE+1} \leftarrow t^i$
- 10 **si** $r_{br} < t_{br}$ **entonces**
- 11 $r_{br}^{FE+2} \leftarrow r_{br}^i + \frac{d_{br}^i}{2}$
- 12 $t_{br}^{FE+1} \leftarrow t_{br}^i - \frac{d_{br}^i}{2}$
- 13 **en otro caso**
- 14 $r_{br}^{FE+1} \leftarrow r_{br}^i - \frac{d_{br}^i}{2}$
- 15 $t_{br}^{FE+2} \leftarrow t_{br}^i + \frac{d_{br}^i}{2}$
- 16 $x_{\min}^{FE+1}, f_{\min}^{FE+1}, x_{\min}^{FE+2}, f_{\min}^{FE+2}, \text{IsFactible} \leftarrow$ Algoritmo 3.9 // Chequeo de factibilidad
- 17 $FE \leftarrow FE + 2$
- 18 $f_{\min} \leftarrow \min \{ f_{\min}, f_{\min}^{FE+1}, f_{\min}^{FE+2} \}$
- 19 $x_{\min} \leftarrow \arg \min_{x \in \{x_{\min}, x_{\min}^{FE+1}, x_{\min}^{FE+2}\}} f(x)$
- 20 $s \leftarrow$ Algoritmo 3.10 // Actualización de s
- 21 $a_{br}^{FE+2} \leftarrow \frac{b_{br}^i + a_{br}^i}{2}$ // Actualización de la componente br de a y b
- 22 $b_{br}^{FE+1} \leftarrow \frac{b_{br}^i + a_{br}^i}{2}$
- 23 $\delta_{FE+1} \leftarrow \frac{2}{3} \| b^{FE+1} - a^{FE+1} \|$ // Actualización del tamaño de los hiperrectángulos
- 24 $\delta_{FE+2} \leftarrow \frac{2}{3} \| b^{FE+2} - a^{FE+2} \|$
- 25 $\mathbb{I}_k \leftarrow \mathbb{I}_k \cup \{FE+1, FE+2\}$ // Actualización de \mathbb{I}_k
- 26 **devolver** $\mathbb{I}_k, f_{\min}, x_{\min}, FE$

Algoritmo 3.9: Chequeo de factibilidad

Entrada: $r^{\text{FE}+1}, t^{\text{FE}+1}, r^{\text{FE}+2}, t^{\text{FE}+2}$
Salida: $x_{\text{mín}}^{\text{FE}+1}, f_{\text{mín}}^{\text{FE}+1}, x_{\text{mín}}^{\text{FE}+2}, f_{\text{mín}}^{\text{FE}+2}, \text{IsFactible}$
 /* Verificación de factibilidad en los hiperrectángulos FE + 1 y FE + 1 */

- 1 **para** $j = \text{FE} + 1, \text{FE} + 2$ **hacer**
- 2 **si** $r^j \in \Omega$ **y** $t^j \in \Omega$ **entonces**
- 3 $f_{\text{mín}}^j \leftarrow \min\{f(r^j), f(t^j)\}$
- 4 $x_{\text{mín}}^j \leftarrow \arg \min_{x \in \{r^j, t^j\}} f(x)$
- 5 IsFactible \leftarrow True
- 6 **si no, si** $r^j \in \Omega$ **entonces**
- 7 $f_{\text{mín}}^j \leftarrow f(r^j)$
- 8 $x_{\text{mín}}^j \leftarrow r^j$
- 9 IsFactible \leftarrow True
- 10 **si no, si** $t^j \in \Omega$ **entonces**
- 11 $f_{\text{mín}}^j \leftarrow f(t^j)$
- 12 $x_{\text{mín}}^j \leftarrow t^j$
- 13 IsFactible \leftarrow True

13 **devolver** $x_{\text{mín}}^{\text{FE}+1}, f_{\text{mín}}^{\text{FE}+1}, x_{\text{mín}}^{\text{FE}+2}, f_{\text{mín}}^{\text{FE}+2}$ (en caso que existan), IsFactible

Algoritmo 3.10: Actualización de s

Entrada: $s, r^{\text{FE}+1}, r^{\text{FE}+2}, t^{\text{FE}+1}, t^{\text{FE}+2}$
Salida: s

- 1 **para** $j \leftarrow 0$ **a** m **hacer**
- 2 $s_0 \leftarrow s_0 + \frac{|f(r^{\text{FE}+2}) - f(r^{\text{FE}+1})|}{\|r^{\text{FE}+2} - r^{\text{FE}+1}\|} + \frac{|f(t^{\text{FE}+2}) - f(t^{\text{FE}+1})|}{\|t^{\text{FE}+2} - t^{\text{FE}+1}\|}$
- 3 $s_j \leftarrow s_j + \frac{|g_j(r^{\text{FE}+2}) - g_j(r^{\text{FE}+1})|}{\|r^{\text{FE}+2} - r^{\text{FE}+1}\|} + \frac{|g_j(t^{\text{FE}+2}) - g_j(t^{\text{FE}+1})|}{\|t^{\text{FE}+2} - t^{\text{FE}+1}\|}$
- 4 **devolver** s

3.3.5. Resultados numéricos

En esta sección mostramos y analizamos los resultados numéricos obtenidos al aplicar el nuevo algoritmo BIRECT con restricciones generales (Algoritmo 3.5) a diferentes problemas test de optimización global. Dichos experimentos numéricos fueron realizados en una computadora con un procesador Intel Core i5-6200u de 2.3 GHz de 8 GB de RAM y en la computadora Nabucodonosor⁴ perteneciente al Centro de Computación de Alto Desempeño de la Universidad Nacional de Córdoba (CCAD). Asimismo todos los códigos fueron implementados en Octave y los gráficos de esta sección fueron realizados con Python.

Para poder analizar la performance del mencionado algoritmo hemos seleccionado un conjunto de 14 problemas test cuyas funciones objetivo, restricciones generales y solución óptima global conocida se encuentran detalladas en el Apéndice B. La Tabla 3.4 presenta un resumen con la información relevante de los mismos. Es importante observar que en la literatura existen pocos problemas test con restricciones generales en cuales se cuente con una solución global

⁴<https://ccad.unc.edu.ar/equipamiento/computadora-nabucodonosor/>

conocida. Si bien los resultados que se presentan a continuación no pueden ser generalizados, son un puntapié inicial para medir la performance de la nueva propuesta algorítmica.

Prob. N°	Nombre del Problema	Dimensión	N° restricciones generales	Tipo de función objetivo	Óptimo global
1	Epperly 2.1	6	2	Lineal	-17
2	Epperly 2.2	7	3	Lineal	-213
3	Floudas 2.1	5	1	Cuadrática	-17
4	Floudas 3.1	8	6	Lineal	7049.2
5	Floudas 3.2	5	6	Cuadrática	-30665.5387
6	Floudas 3.3	6	6	Cuadrática	-310
7	Floudas 3.4	3	3	Cuadrática	-4
8	Floudas 4.8	2	1	Cuadrática	-16.73889
9	Floudas 4.9	2	2	Lineal	-5.50796
10	Floudas 7.3	8	6	Lineal	709.25
11	Floudas 7.6	3	1	General	-83.254
12	Floudas 7.7	4	2	General	-5.7398
13	Floudas 8.7	5	3	Polinomial General	0.0293
14	G8	2	2	General	-0.095825

Tabla 3.4: Descripción de problemas test - BIRECT con restricciones generales.

En cuanto a la configuración inicial del algoritmo tres criterios de parada fueron establecidos:

1. En relación con la máxima cantidad de evaluaciones funcionales permitida, se configuró el parámetro $FE_{\text{máx}} = 10000 \times n \times m$, siendo n la dimensión del problema y m el número de restricciones. Este criterio no fue aleatorio sino que estuvo pensado en base a la complejidad del proceso algorítmico, en particular al proceso de partición de hiperrectángulos que involucra evaluar dos puntos nuevos en cada paso.
2. En segundo lugar, considerando que todos los problemas test cuentan con su solución óptima global, al igual que en [43] se adoptó el criterio de tolerancia tol en la aceptación de la aproximación obtenida por el algoritmo. El mencionado criterio se encuentra definido por el error relativo de dicha aproximación con respecto a su valor óptimo, es decir

$$tol = \begin{cases} \frac{f(\bar{x}) - f^*}{|f^*|} \leq tol_{\text{máx}}, & f^* \neq 0 \\ f(\bar{x}) \leq tol_{\text{máx}}, & f^* = 0 \end{cases}$$

A nivel experimental, para cada problema se corrieron experimentos con tres valores de $tol_{\text{máx}}$ posibles: 10^{-3} , 10^{-4} y 10^{-5} .

3. Finalmente, relacionado con las iteraciones, se fijó la cantidad máxima $K_{\text{máx}} = 200000$, buscando que este número sea lo suficientemente grande y que sea el criterio alcanzado en caso de que los dos primeros fallen.

Para obtener conclusiones lo más objetivas posible, el estudio numérico mencionado se dividirá en el análisis de diferentes facetas sobre la performance del algoritmo en la resolución de los problemas arriba mencionados. Las mismas pueden enumerarse de acuerdo a las siguientes categorías de análisis:

1. Análisis del impacto de las diferentes tolerancias sobre medidas de performance usualmente utilizadas:
 - Iteraciones.
 - Evaluaciones funcionales.
 - Tolerancia alcanzada.
 - Errores absolutos y relativos de la solución.
 - Tiempo de ejecución.
2. Análisis de las soluciones obtenidas en función de la dimensión de los problemas.
3. Análisis de la obtención de puntos factibles durante el proceso iterativo teniendo en cuenta las cápsulas convexas obtenidas, el tiempo requerido por las mismas y el progreso en el número de evaluaciones funcionales.

Adicionalmente, mostraremos el valor final de los coeficientes c (ecuación (3.18)) sobre las restricciones de cada uno de los problemas test considerados.

3.3.5.1. El efecto de la utilización de diferentes tolerancias

En la Tabla 3.5 se observan los resultados que se obtuvieron aplicando el Algoritmo BIRECT con restricciones generales a los 14 problemas test. La misma se encuentra fraccionada en 3 partes diferenciando así los resultados de las 3 tolerancias máximas consideradas: 10^{-3} , 10^{-4} y 10^{-5} . Asimismo, las subtablas de cada una de estas tolerancias posee 7 columnas representando lo siguiente:

- $It.$: Cantidad de iteraciones alcanzadas.
- $FE.$: Cantidad de evaluaciones funcionales alcanzadas.
- tol : Tolerancia alcanzada o error relativo de la aproximación funcional encontrada con respecto al valor funcional mínimo conocido.
- $\|x^* - x_{\min}\|$: Error absoluto de la aproximación encontrada con respecto al optimizador conocido.
- $\frac{\|x^* - x_{\min}\|}{\|x^*\|}$: Error relativo de la aproximación encontrada con respecto al punto óptimo conocido.
- $\|f^* - f_{\min}\|$: Error absoluto de la aproximación funcional encontrada con respecto al mínimo valor funcional conocido.
- $time_{\text{sec}}$: Tiempo total en segundos requerido para satisfacer alguno de los criterios de convergencia.

Adicionalmente, cada una de las subtablas incluyen al final de cada columna el cálculo del promedio y la mediana de los respectivos valores.

Antes de proceder con el análisis respectivo es importante realizar algunas aclaraciones sobre ciertos símbolos, colores y datos faltantes observados en la Tabla 3.5. El símbolo \dagger que acompaña a algunos problemas en la columna “Nombre del Problema” denota aquellos casos en los que el uso de las tres tolerancias arrojó exactamente los mismos resultados (salvo algunas diferencias en segundos). Por otra parte, teniendo en cuenta que los experimentos pueden terminar de acuerdo

a alguno de los criterios de parada mencionados anteriormente (iteraciones, evaluaciones funcionales o tolerancia), para cada problema y cada tolerancia, se marcó en negrita el criterio de parada con el cual terminó el experimento. El color rojo en la columna $\|f^* - f_{\min}\|$ se encuentra apuntando a la mejor aproximación funcional obtenida entre *todos* los experimentos para cada problema. Finalmente, como se puede observar, los problemas 4 (Floudas 3.1) y 6 (Floudas 3.3) poseen datos faltantes para algunas tolerancias. Esta ausencia de datos se debe a que los mencionados experimentos agotaron los recursos computacionales disponibles sin haber alcanzado una aproximación a la solución (por ejemplo el tope de 2 días establecido en Nabucodonosor). Por esta razón, se decidió no incluirlos en el estudio.

Habiendo realizado las aclaraciones pertinentes, los resultados de la Tabla 3.5 nos muestran aspectos muy interesantes para analizar. Inicialmente, podemos notar que las mejores aproximaciones al valor funcional mínimo se obtienen mayormente cuando $\text{tol}_{\text{máx}} = 10^{-5}$, lo cual es totalmente razonable. Si bien con tolerancias menores en algunos casos se llega al mismo valor, la mediana de la columna $\|f^* - f_{\min}\|$ nos muestra que estas aproximaciones mejoran a medida que somos más exigentes, siendo 0.015641, 0.001300 y 5.775500e-04 para 10^{-3} , 10^{-4} y 10^{-5} respectivamente. En este caso, el promedio de los valores no es representativo dado que se encuentra afectado por valores extremos como sucede con el Problema 10 cuyo error absoluto da 243.91 en esta columna. En contraposición, sin embargo, la mediana del error relativo en los valores de x se mantiene en el orden de 10^{-3} en todas las subtablas de la Tabla 3.5, al igual que su promedio que ronda aproximadamente en 0.10. Una explicación posible a este punto sea quizás la presencia de ciertos mínimos locales que no son globales.

A diferencia de lo que sucede con la columna $\|f^* - f_{\min}\|$, cuando se observa el criterio de parada vemos que en el caso de $\text{tol}_{\text{máx}} = 10^{-3}$ y $\text{tol}_{\text{máx}} = 10^{-4}$ los experimentos en su mayoría terminaron en base al criterio de tolerancia. Para $\text{tol}_{\text{máx}} = 10^{-5}$, en cambio, existe un balance entre haber terminado por tope de evaluaciones funcionales y haber alcanzado la tolerancia. Como era esperable, en ningún caso la finalización de los experimentos fue por número máximo de iteraciones. Adicionalmente, esta faceta de análisis posee otros aspectos interesantes para remarcar. Por un lado, con respecto a la columna “tol” se observa que el promedio de valores es prácticamente el mismo a lo largo de cada tolerancia mientras que la mediana se acerca progresivamente más a 0 a medida que la Tabla 3.5 se recorre de arriba hacia abajo. Los casos con $\text{tol}_{\text{máx}} = 10^{-4}$ y $\text{tol}_{\text{máx}} = 10^{-5}$ comparten una mediana promedio de tolerancia en el orden de 10^{-5} , siendo levemente menor en la última (3.8e-05 en comparación con 7.6e-05). Por otro lado, a nivel iteraciones y evaluaciones funcionales, tanto los promedios como las medianas con $\text{tol}_{\text{máx}} = 10^{-3}$ y $\text{tol}_{\text{máx}} = 10^{-4}$ se encuentran en un rango similar.

En cuanto al tiempo de ejecución, “time_{sec}”, los promedios y medianas comparten nuevamente el mismo orden en 10^{-3} y 10^{-4} sin embargo en 10^{-5} se produce un gran aumento en orden de magnitud. Numéricamente hablando, mientras que la mediana de las dos primeras tolerancias se encuentra alrededor de 5 minutos, en la última se aproxima a las 6.6 horas, es decir se produce un aumento de alrededor 80x en tiempo de ejecución.

Los párrafos anteriores proporcionan una visión inicial de los resultados mostrando las ventajas y desventajas en la elección del valor de la tolerancia máxima. Si bien los resultados en la gran mayoría de los casos son alentadores y se aproximan a lo esperado, no son prueba suficiente para realizar un veredicto acerca de la performance del Algoritmo BIRECT con restricciones generales. Para sumar información a este análisis, a continuación se analizará los resultados en base a la dimensión de los problemas y el proceso de obtención de los puntos factibles durante el proceso iterativo.

Tabla 3.5: Resultados numéricos - BIRECT con restricciones generales

Prob.	Nombre del	tol _{máx} = 10 ⁻³						
		n°	Problema	It.	FE.	tol	$\ x^* - x_{\min}\ $	$\frac{\ x^* - x_{\min}\ }{\ x^*\ }$
1	Epperly 2.1	1816	20330	9.9766e-04	0.016962	9.9263e-04	0.016960	2616
2	Epperly 2.2	841	7434	6.8161e-04	0.14578	6.8138e-04	0.14518	522
3	Floudas 2.1	164	1692	8.4247e-04	1.3495e-04	7.7916e-05	0.014322	16
4	Floudas 3.1	5740	140000	0.11839	1464.9	0.27347	834.59	175586.0
5	Floudas 3.2	316	6830	8.3741e-04	0.35454	0.0033133	25.680	355
6	Floudas 3.3 [†]	3819	120000	0.011780	0.38728	0.029192	3.6518	173062
7	Floudas 3.4	33	582	5.6966e-04	3.3526	1.1023	0.0022786	2
8	Floudas 4.8	40	464	2.7064e-04	0.035891	0.021942	0.0045303	1
9	Floudas 4.9	25	422	8.6384e-04	0.0053452	0.0013564	0.0047581	1
10	Floudas 7.3 [†]	3575	140000	0.034601	197.53	0.036869	243.91	154066
11	Floudas 7.6	47	1238	9.3691e-04	1.0347	0.011673	0.078001	3
12	Floudas 7.7	344	10602	9.5522e-05	0.010934	0.0011017	5.4828e-04	312
13	Floudas 8.7 [†]	10901	80000	0.39989	0.19435	0.055639	0.011713	120337
14	GS	76	990	4.0334e-05	9.5525e-04	2.1615e-04	3.8650e-06	5
	Promedio	1981.214286	37898.857143	0.040771	119.140677	0.109916	79.150721	44777.428571
	Mediana	330	7132	8.53e-04	0.170065	0.007493	0.015641	333.5

Prob.	Nombre del	tol _{máx} = 10 ⁻⁴						
		n°	Problema	It.	FE.	tol	$\ x^* - x_{\min}\ $	$\frac{\ x^* - x_{\min}\ }{\ x^*\ }$
1	Epperly 2.1	4481	68840	3.7362e-05	0.00063518	0.000037171	0.00063515	46483
2	Epperly 2.2	3682	28178	5.5495e-05	0.011849	5.5382e-05	0.011820	9644
3	Floudas 2.1	354	2872	7.6443e-05	1.2967e-05	7.4868e-06	0.0012995	57
4	Floudas 3.1	-	-	-	-	-	-	.
5	Floudas 3.2	418	9910	8.6496e-05	0.040163	3.7533e-04	2.6524	607
6	Floudas 3.3 [†]	3819	120000	0.011780	0.38728	0.029192	3.6518	168861
7	Floudas 3.4	56	1146	7.1208e-05	3.3539	1.1028	2.8483e-04	6
8	Floudas 4.8	99	1782	8.5135e-06	0.0049267	0.0030119	1.4251e-04	8
9	Floudas 4.9	31	618	9.5550e-05	5.9999e-04	1.5225e-04	5.2629e-04	1
10	Floudas 7.3 [†]	3575	140000	0.034601	197.53	0.036869	243.91	159365
11	Floudas 7.6	55	1476	5.2465e-05	0.22626	0.0025525	0.0043679	5
12	Floudas 7.7	344	10602	9.5522e-05	0.010934	0.0011017	5.4828e-04	290
13	Floudas 8.7 [†]	10901	80000	0.39989	0.19435	0.055639	0.011713	127724
14	GS [†]	76	990	4.0334e-05	9.5525e-04	2.1615e-04	3.8650e-06	5
	Promedio	2145.461538	35878	0.034376	15.520144	0.094770	19.249657	39465.846154
	Mediana	354	9910	7.6e-05	0.011849	0.001102	0.001300	290

Prob.	Nombre del	tol _{máx} = 10 ⁻⁵						
		n°	Problema	It.	FE.	tol	$\ x^* - x_{\min}\ $	$\frac{\ x^* - x_{\min}\ }{\ x^*\ }$
1	Epperly 2.1	5101	80000	3.7362e-04	0.00063518	0.000037171	0.00063515	65776.0
2	Epperly 2.2	9953	100000	0.000055495	0.011849	0.000055382	0.011820	173418
3	Floudas 2.1	5473	60000	0.000038222	0.0000064837	0.0000037434	0.00064977	31921.0
4	Floudas 3.1	-	-	-	-	-	-	.
5	Floudas 3.2	3340	110000	0.000040889	0.014153	0.00013227	1.2539	112389
6	Floudas 3.3 [†]	-	-	-	-	-	-	.
7	Floudas 3.4	887	3613	2.5431e-06	3.3541	1.1028	0.000010173	5648
8	Floudas 4.8	99	1782	8.5135e-06	0.0049267	0.0030119	1.4251e-04	8
9	Floudas 4.9	577	40000	0.000013876	0.000075455	0.000019147	0.000076429	4238
10	Floudas 7.3 [†]	3575	140000	0.034601	197.53	0.036869	243.91	156740.0
11	Floudas 7.6	225	17930	6.2454e-06	0.35991	0.0040604	5.1995e-04	382
12	Floudas 7.7	14396	60000	0.000070037	0.0016514	0.00016638	0.00040200	15998
13	Floudas 8.7 [†]	10901	80000	0.39989	0.19435	0.055639	0.011713	123415
14	GS	488	9924	7.4792e-06	3.9196e-04	8.8691e-05	7.1670e-07	338
	Promedio	3501.250000	61314	0.036231	16.789337	0.100240	2.043249e+01	57522.583333
	Mediana	2368	60000	3.8e-05	0.008388	0.000149	5.775500e-04	23959.5

3.3.5.2. ¿Cuánto impacta la dimensión del problema en la obtención de la solución?

En línea con lo mencionado previamente, en esta sección se analizan los resultados de la Tabla 3.5 en función de la dimensión de los problemas. Para ello los resultados de los 14 problemas test han sido distribuidos en dos agrupaciones diferentes: *problemas de dimensión “baja”* y *problemas de dimensión “alta”*. Para cada uno de estos grupos se han calculado nuevamente las medias y medianas de los parámetros analizados. El esquema de distribución de los mismos se muestra a continuación y los resultados se presentan en las Tablas 3.6 y 3.7.

	Dim. 2, 3, 4 y 5	Dim. 6, 7 y 8
1	Floudas 4.8	Epperly 2.1
2	Floudas 4.9	Epperly 2.2
3	G8	Floudas 3.1
4	Floudas 3.4	Floudas 3.3
5	Floudas 7.6	Floudas 7.3
6	Floudas 7.7	
7	Floudas 2.1	
8	Floudas 3.2	
9	Floudas 8.7	

Uno de los primeros puntos que observamos a lo largo de la Tabla 3.6, es que el promedio de los errores absolutos de $\|f^* - f_{\text{mín}}\|$ y de los errores relativos de $\frac{\|x^* - x_{\text{mín}}\|}{\|x^*\|}$ es mínimo cuando $\text{tol}_{\text{máx}} = 10^{-5}$ y $\text{tol}_{\text{máx}} = 10^{-4}$ respectivamente (en este último caso la diferencia con 10^{-5} es mínima). Por otro lado, sin embargo, el promedio de estos errores aumenta en orden de magnitud a la par que lo hace la dimensión, es decir, a menor dimensión más cercanos a 0 son los mismos. Por ejemplo, el valor promedio mínimo en la columna $\|f^* - f_{\text{mín}}\|$ en la Tabla 3.6 es 1.40238e-01 (dimensiones 2, 3, 4 y 5) mientras que en la Tabla 3.7 es 61.893564 (dimensiones 6, 7 y 8).

Para los 9 problemas de la Tabla 3.6, los resultados muestran que un criterio de tolerancia más exigente mejoraría el valor de todos los errores considerados y que, al menos para el conjunto analizado, no existirían diferencias significativas entre $\text{tol}_{\text{máx}} = 10^{-4}$ y $\text{tol}_{\text{máx}} = 10^{-5}$. Asimismo, este esfuerzo del algoritmo por alcanzar una mejor solución se traduce en iteraciones, evaluaciones funcionales y tiempo de ejecución cada vez más mayores. Con respecto a este punto, se observa que el salto (de los valores promedios) de 10^{-4} a 10^{-5} es de, aproximadamente, 4x en la cantidad de evaluaciones funcionales, 3x en las iteraciones y 2x en el tiempo de ejecución. Estos resultados se hacen más notorios cuando se analiza el comportamiento de la mediana de los mismos. Mientras que para 10^{-3} y 10^{-4} los órdenes de magnitud son similares, el pasaje 10^{-4} a 10^{-5} en la mediana implica un aumento de alrededor 9x en las iteraciones, 22x en las evaluaciones funcionales y 700x en el tiempo de ejecución. En este caso, los valores extremos en los resultados de algunos problemas de la Tabla 3.6 se encuentran claramente sesgando el promedio general y es por esta razón por la que se decidió incluir en el estudio el valor de la mediana de los datos. Finalmente, para esta agrupación, desde el ángulo del número de restricciones de cada uno de los problemas, los resultados no muestran ninguna afirmación concluyente.

Tabla 3.6: Resultados numéricos - BIRECT con restricciones generales - Dimensiones 2, 3, 4 y 5

Prob.	Nombre del	$\text{tol}_{\text{máx}} = 10^{-3}$						
n°	Problema	<i>It.</i>	<i>FE.</i>	tol	$\ x^* - x_{\text{mín}}\ $	$\frac{\ x^* - x_{\text{mín}}\ }{\ x^*\ }$	$\ f^* - f_{\text{mín}}\ $	time _{sec}
8	Floudas 4.8	40	464	2.7064e-04	0.035891	0.021942	0.0045303	1
9	Floudas 4.9	25	422	8.6384e-04	0.0053452	0.0013564	0.0047581	1
14	G8	76	990	4.0334e-05	9.5525e-04	2.1615e-04	3.8650e-06	5
7	Floudas 3.4	33	582	5.6966e-04	3.3526	1.1023	0.0022786	2
11	Floudas 7.6	47	1238	9.3691e-04	1.0347	0.011673	0.078001	3
12	Floudas 7.7	344	10602	9.5522e-05	0.010934	0.0011017	5.4828e-04	312
3	Floudas 2.1	164	1692	8.4247e-04	1.3495e-04	7.7916e-05	0.014322	16
5	Floudas 3.2	316	6830	8.3741e-04	0.35454	0.0033133	25.680	355
13	Floudas 8.7	10901	80000	0.39989	0.19435	0.055639	0.011713	120337
	Promedio	1327.333333	11424.4	0.044927	0.554383	0.133069	2.866239	13448
	Mediana	76	1238	0.000837	0.035891	0.003313	0.004758	5

Prob.	Nombre del	$\text{tol}_{\text{máx}} = 10^{-4}$						
n°	Problema	<i>It.</i>	<i>FE.</i>	tol	$\ x^* - x_{\text{mín}}\ $	$\frac{\ x^* - x_{\text{mín}}\ }{\ x^*\ }$	$\ f^* - f_{\text{mín}}\ $	time _{sec}
8	Floudas 4.8	99	1782	8.5135e-06	0.0049267	0.0030119	1.4251e-04	8
9	Floudas 4.9	31	618	9.5550e-05	5.9999e-04	1.5225e-04	5.2629e-04	1
14	G8	76	990	4.0334e-05	9.5525e-04	2.1615e-04	3.8650e-06	5
7	Floudas 3.4	56	1146	7.1208e-05	3.3539	1.1028	2.8483e-04	6
11	Floudas 7.6	55	1476	5.2465e-05	0.22626	0.0025525	0.0043679	5
12	Floudas 7.7	344	10602	9.5522e-05	0.010934	0.0011017	5.4828e-04	290
3	Floudas 2.1	354	2872	7.6443e-05	1.2967e-05	7.4868e-06	0.0012995	57
5	Floudas 3.2	418	9910	8.6496e-05	0.040163	3.7533e-04	2.6524	607
13	Floudas 8.7	10901	80000	0.39989	0.19435	0.055639	0.011713	127724
	Promedio	1370.4	12155.1	0.044491	0.425789	0.129540	0.296810	14300.333333
	Mediana	99	1782	0.000076	0.010934	0.001102	0.000548	8

Prob.	Nombre del	$\text{tol}_{\text{máx}} = 10^{-5}$						
n°	Problema	<i>It.</i>	<i>FE.</i>	tol	$\ x^* - x_{\text{mín}}\ $	$\frac{\ x^* - x_{\text{mín}}\ }{\ x^*\ }$	$\ f^* - f_{\text{mín}}\ $	time _{sec}
8	Floudas 4.8	99	1782	8.5135e-06	0.0049267	0.0030119	1.4251e-04	8
9	Floudas 4.9	577	40000	1.3876e-05	7.5455e-05	1.9147e-05	7.6429e-05	4238
14	G8	488	9924	7.4792e-06	3.9196e-04	8.8691e-05	7.1670e-07	338
7	Floudas 3.4	887	36132	2.5431e-06	3.3541	1.1028	0.000010173	5648
11	Floudas 7.6	225	17930	6.2454e-06	0.35991	0.0040604	5.1995e-04	382
12	Floudas 7.7	14396	60000	7.0037e-05	0.0016514	0.00016638	0.00040200	15998
3	Floudas 2.1	5473	60000	3.8222e-05	6.4837e-06	3.7434e-06	6.4977e-04	31921.0
5	Floudas 3.2	3340	110000	4.0889e-05	0.014153	1.3227e-04	1.2539	112389
13	Floudas 8.7	10901	80000	0.39989	0.19435	0.055639	0.011713	123415
	Promedio	4042.9	46196.4	0.044453	0.436618	0.129547	1.408238e-01	32704.1
	Mediana	887	40000	0.000014	0.004927	0.000166	4.020000e-04	5648

La Tabla 3.7 que abarca las dimensiones 6, 7 y 8 es bastante particular puesto que $\text{tol} = 10^{-3}$

es la única tolerancia con todos los experimentos completos y $\text{tol} = 10^{-5}$ es la que menos datos contiene. Además, en esta parte del análisis estamos en presencia de los problemas con mayor número de restricciones: Floudas 3.1, Floudas 3.3 y Floudas 7.3 poseen 6 restricciones cada uno. El primer punto notable en esta tabla es que la mayoría de los experimentos terminaron por tope de evaluaciones funcionales. Observando los resultados de los mismos, se podría decir que el rendimiento en estas dimensiones no fue lo suficientemente bueno, particularmente en lo que hace referencia a la columna de los errores. Si bien a mayor exigencia, se observa una disminución en estos valores, los mismos continúan siendo grandes en comparación a lo esperado y a lo visto en dimensiones menores, siendo el promedio 0.012321 en la columna $\frac{\|x^* - x_{\min}\|}{\|x^*\|}$ el número más bajo obtenido.

Por otro lado, es muy probable que dada la complejidad de los problemas (dimensión denotada por n más el número de restricciones denotado por m) y del proceso algorítmico propiamente dicho, el máximo en las evaluaciones funcionales $10000 \times n \times m$ resulte algo limitado en estos casos. Observando la columna “tol” es claro que muchos de los valores de esta columna se encuentran cerca de la tolerancia fijada, lo cual hace suponer que no *faltaría mucho esfuerzo* para alcanzar una solución aceptable u óptima. Asimismo, el punto anterior combinado con la disponibilidad de más recursos de hardware también induce a pensar que aquellos experimentos que no fueron posibles de completar puedan ser ejecutados satisfactoriamente con estos cambios. Nótese que los mismos son justamente aquellos con 6 restricciones.

Tabla 3.7: Resultados numéricos - BIRECT con restricciones generales - Dimensiones 6, 7 y 8

Prob.	Nombre del	tol _{máx} = 10 ⁻³						
		n°	Problema	It.	FE.	tol	$\ x^* - x_{\min}\ $	$\frac{\ x^* - x_{\min}\ }{\ x^*\ }$
1	Epperly 2.1	1816	20330	9.9766e-04	0.016962	9.9263e-04	0.016960	2616
6	Floudas 3.3	3819	120000	0.011780	0.38728	0.029192	3.6518	173062
2	Epperly 2.2	841	7434	6.8161e-04	0.14578	6.8138e-04	0.14518	522
4	Floudas 3.1	5740	140000	0.11839	1464.9	0.27347	834.59	175586.0
10	Floudas 7.3	3575	140000	0.034601	197.53	0.036869	243.91	154066
	Promedio	3158.2	85552.8	0.033290	332.596004	0.068241	216.462788	101170.4
	Mediana	3575	120000	0.011780	0.387280	0.029192	3.651800	154066

Prob.	Nombre del	tol _{máx} = 10 ⁻⁴						
		n°	Problema	It.	FE.	tol	$\ x^* - x_{\min}\ $	$\frac{\ x^* - x_{\min}\ }{\ x^*\ }$
1	Epperly 2.1	4481	68840	3.7362e-05	6.3518e-04	3.7171e-05	6.3515e-04	46483
6	Floudas 3.3	3819	120000	0.011780	0.38728	0.029192	3.6518	168861
2	Epperly 2.2	3682	28178	5.5495e-05	0.011849	5.5382e-05	0.011820	9644
4	Floudas 3.1	-	-	-	-	-	-	.
10	Floudas 7.3	3575	140000	0.034601	197.53	0.036869	243.91	159365
	Promedio	3889.25	89254.5	0.011618	49.482441	0.016538	61.893564	96088.25
	Mediana	3750.5	94420	0.005918	0.199565	0.014624	1.831810	102924

Prob.	Nombre del	$\text{tol}_{\text{máx}} = 10^{-5}$						
n°	Problema	<i>It.</i>	<i>FE.</i>	tol	$\ x^* - x_{\text{mín}}\ $	$\frac{\ x^* - x_{\text{mín}}\ }{\ x^*\ }$	$\ f^* - f_{\text{mín}}\ $	time _{sec}
1	Epperly 2.1	5101	80000	3.7362e-04	6.3518e-04	3.7171e-05	6.3515e-04	65776.0
6	Floudas 3.3	-	-	-	-	-	-	-
2	Epperly 2.2	9953	100000	5.5495e-04	0.011849	5.5382e-05	0.011820	173418
4	Floudas 3.1	-	-	-	-	-	-	-
10	Floudas 7.3	3575	140000	0.034601	197.53	0.036869	243.91	156740.0
	Promedio	6209.666667	106666.666667	0.011677	65.847495	0.012321	81.307485	131978
	Mediana	5101	100000	3.74e-04	0.011849	5.5e-05	0.011820	156740

A modo de cierre de esta arista del análisis numérico del algoritmo BIRECT con restricciones generales y respondiendo a la pregunta del título de esta sección se podría decir que la dimensión del problema incide en una buena proporción en la calidad de los resultados obtenidos. Como se notó previamente, debido a la complejidad de todo el proceso, es muy probable que no sea el único factor a considerar. Sin embargo es claro que mayores dimensiones implican un esfuerzo más grande del algoritmo en su ejecución interna y eso tiene impacto en todos los parámetros analizados.

3.3.5.3. Cápsulas convexas y el movimiento hacia la factibilidad

En esta sección se mostrarán algunos aspectos interesantes acerca de las cápsulas convexas recopilados durante la experimentación numérica. Como se desarrolló en secciones anteriores, el cálculo de las mismas es una parte fundamental dentro del algoritmo BIRECT con restricciones generales. Recordemos que el llamado al Algoritmo 3.7 implica que el proceso iterativo encontró al menos un punto factible. Dado que, en base a lo analizado en el apartado anterior, la tolerancia $\text{tol}_{\text{máx}} = 10^{-4}$ fue la que reportó resultados más “estables” en las diferentes líneas de análisis sólo se expondrán valores obtenidos dentro de este contexto.

La Tabla 3.8 muestra un resumen de la información extraída a partir de los resultados. En la misma, siguiendo las agrupaciones por dimensión realizadas en la sección anterior, para cada problema y cada dimensión se definieron las siguientes variables:

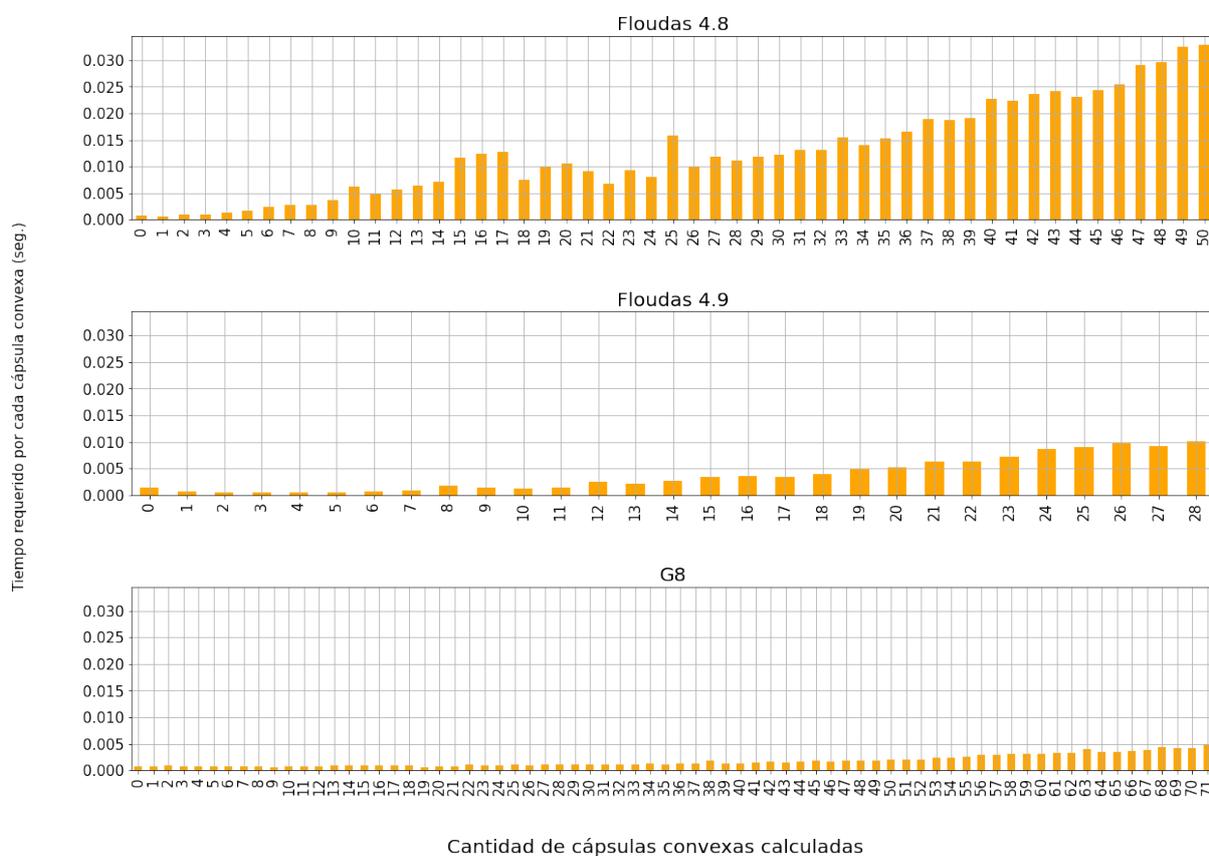
- **# Cápsulas convexas:** Cantidad total de cápsulas convexas calculadas hasta satisfacer alguno de los criterios de convergencia.
- **Tiempo:** Promedio y mediana del tiempo (en segundos) que el algoritmo requirió en total para obtener las cápsulas anteriores.
- **Tasa de crecimiento *FE*:** Medida definida como el promedio de las diferencias en el número de evaluaciones funcionales entre dos cápsulas convexas consecutivas. Es decir, si por ejemplo, se calcularon 4 cápsulas convexas con 12, 20, 28 y 40 evaluaciones funcionales respectivamente, la diferencia entre dos consecutivas estaría dada por el vector con componentes 8, 8 y 12. Luego, su promedio sería 9.3 y este es valor que se coloca en la Tabla 3.8. Si bien la variable definida de esta manera no es una tasa de crecimiento en el sentido matemático estricto, la idea detrás de su definición es que la misma represente si el aumento de evaluaciones funcionales dentro del cálculo de las cápsulas convexas del Algoritmo 3.7 fue significativo o no. Dado que el número de evaluaciones funcionales es el parámetro por excelencia en los métodos de optimización sin derivadas, la tasa de crecimiento *FE* brindaría información acerca de la complejidad computacional del Algoritmo 3.4.

Los resultados de la Tabla 3.8 muestran que para problemas de dimensión “baja” (2, 3, 4 y 5) el tiempo requerido y la *velocidad* de crecimiento en las evaluaciones funcionales se encuentran aproximadamente dentro del mismo orden de magnitud. Si bien para problemas de dimensión 5 el número de cápsulas convexas calculadas aumenta hasta 60 veces con respecto a la dimensión anterior, esto parecería no tener impacto en el tiempo y en la variable *FE*. De esta manera, se podría decir que, para el conjunto de problemas considerados, el esfuerzo computacional para encontrar puntos factibles es relativamente el mismo en dimensión 2, 3, 4 y 5. Como es esperable para dimensiones mayores a estas, el aumento en las variables se observa mayormente en la cantidad de cápsulas y el tiempo de ejecución, mientras que las evaluaciones funcionales crecen a una *velocidad* menor. A simple vista, parecería que la dimensión tiene más impacto que el número de restricciones del problema en la determinación de puntos factibles. Si bien este análisis no es, por ahora, generalizable tiene sentido que encontrar puntos factibles sea más difícil y demande más tiempo a medida que se incorporan más variables.

Tabla 3.8: Resultados cápsula convexa - $tol_{\text{máx}} = 10^{-4}$

		Dimensión 2			Dimensión 6 y 7		
		Floudas 4.8	Floudas 4.9	G8	Epperly 2.1	Floudas 3.3	Epperly 2.2
# Cápsulas convexas		51	29	72	4181	3168	1909
Tiempo	Promedio	0.012809	0.003783	0.001743	3.145382	8.892716	0.153639
	Mediana	0.011860	0.002668	0.001210	2.250600	6.396125	0.120567
Tasa de crecimiento <i>FE</i>		34.56	20.928571	13.774648	16.463158	37.888854	14.757862
		Dimensión 3 y 4			Dimensión 8		
		Floudas 3.4	Floudas 7.6	Floudas 7.7	Floudas 7.3	-	-
# Cápsulas convexas		55	41	261	1692	-	-
Tiempo	Promedio	0.006904	0.010037	0.121922	11.598394	-	-
	Mediana	0.006354	0.006625	0.097810	6.792540	-	-
Tasa de crecimiento <i>FE</i>		20.814815	36.2	40.469231	82.474276	-	-
		Dimensión 5			-		
		Floudas 2.1	Floudas 3.2	Floudas 8.7	-	-	-
# Cápsulas convexas		354	398	2382	-	-	-
Tiempo	Promedio	0.001190	0.054770	0.442642	-	-	-
	Mediana	0.000764	0.051946	0.373547	-	-	-
Tasa de crecimiento <i>FE</i>		8.113314	24.871537	32.864343	-	-	-

La Figura 3.13 ejemplifica lo expuesto hasta el momento en el caso de los problemas de dimensión 2. En la misma se observa claramente que cada nueva cápsula convexa calculada requiere, en la mayoría de las situaciones, un mayor tiempo de ejecución en comparación con la consecutiva anterior. Este comportamiento se repite en las restantes dimensiones y posiblemente esté relacionado con el hecho que mientras más cerca el algoritmo se encuentra de la solución más difícil es determinar con precisión la misma.

Figura 3.13: Tiempo en función de las cápsulas convexas calculadas - $tol_{m\acute{a}x} = 10^{-4}$ 

Algunos datos adicionales

La Tabla 3.9 muestra el último valor obtenido de los coeficientes c , ecuación (3.18), para cada una de las restricciones de los problemas de la Tabla 3.4. Observar estos valores es importante porque uno podría pensar que el denominador de la función auxiliar (3.32) es una función de penalidad. Sin embargo, como remarca Jones en [23] este no es el caso ya que, entre otros puntos, en una función de penalidad estándar es un requisito importante que los coeficientes sean lo suficientemente grandes para evitar minimización en regiones no factibles. Tanto en el caso de Jones como en el caso del nuevo método propuesto, Algoritmo 3.5, la función auxiliar es minimizada en el óptimo global para cualquier combinación de coeficientes c positivos. Los valores de la Tabla 3.9 reafirman este hecho ya que los mismos no toman valores extremadamente grandes. Asimismo, un punto interesante de notar es que el valor del parámetro $tol_{m\acute{a}x}$ no tiene influencia en los resultados finales ya que todo indicaría que llega un momento en el cual estos coeficientes se mantienen alrededor de cierto orden de valores.

Tabla 3.9: Valor final de los coeficientes c .

Prob. N°	Nombre del Problema	tol _{máx}	Coeficientes c					
1	Epperly 2.1	10 ⁻³	0.79744	11.63602	-	-	-	-
		10 ⁻⁴	0.72819	7.07397	-	-	-	-
		10 ⁻⁵	0.70120	6.21253	-	-	-	-
2	Epperly 2.2	10 ⁻³	0.30986	9.02630	2.89740	-	-	-
		10 ⁻⁴	0.27108	12.81182	2.93400	-	-	-
		10 ⁻⁵	0.36001	14.51186	4.01577	-	-	-
3	Floudas 2.1	10 ⁻³	3.6410	-	-	-	-	-
		10 ⁻⁴	3.7762	-	-	-	-	-
		10 ⁻⁵	3.4854	-	-	-	-	-
4	Floudas 3.1	10 ⁻³	3667.5377254	2338.3629793	918.6044904	0.0069529	0.0032750	0.0029326
		10 ⁻⁴	-	-	-	-	-	-
		10 ⁻⁵	-	-	-	-	-	-
5	Floudas 3.2	10 ⁻³	1216.17	1216.17	636.01	636.01	796.53	796.53
		10 ⁻⁴	1195.57	1195.57	616.02	616.02	793.40	793.40
		10 ⁻⁵	1228.89	1228.89	647.68	647.68	804.11	804.11
6	Floudas 3.3	10 ⁻³	77.897	39.779	77.685	132.187	132.187	132.187
		10 ⁻⁴	77.897	39.779	77.685	132.187	132.187	132.187
		10 ⁻⁵	-	-	-	-	-	-
7	Floudas 3.4	10 ⁻³	0.35704	1.21534	0.89177	-	-	-
		10 ⁻⁴	0.36843	1.21678	0.89710	-	-	-
		10 ⁻⁵	0.40675	1.21776	0.88208	-	-	-
8	Floudas 4.8	10 ⁻³	2.7590	-	-	-	-	-
		10 ⁻⁴	3.0128	-	-	-	-	-
		10 ⁻⁵	3.0128	-	-	-	-	-
9	Floudas 4.9	10 ⁻³	0.22641	0.31881	-	-	-	-
		10 ⁻⁴	0.23806	0.35494	-	-	-	-
		10 ⁻⁵	0.25368	0.38924	-	-	-	-
10	Floudas 7.3	10 ⁻³	1013.27	1687.14	3923.80	3291.07	2137.79	899.48
		10 ⁻⁴	1013.27	1687.14	3923.80	3291.07	2137.79	899.48
		10 ⁻⁵	1013.27	1687.14	3923.80	3291.07	2137.79	899.48
11	Floudas 7.6	10 ⁻³	20.182	-	-	-	-	-
		10 ⁻⁴	20.784	-	-	-	-	-
		10 ⁻⁵	29.826	-	-	-	-	-
12	Floudas 7.7	10 ⁻³	4.8582	2.6949	-	-	-	-
		10 ⁻⁴	4.8582	2.6949	-	-	-	-
		10 ⁻⁵	5.3082	3.3042	-	-	-	-
13	Floudas 8.7	10 ⁻³	20.290	52.669	79.094	-	-	-
		10 ⁻⁴	20.290	52.669	79.094	-	-	-
		10 ⁻⁵	20.290	52.669	79.094	-	-	-
14	G8	10 ⁻³	6.1359	10.3313	-	-	-	-
		10 ⁻⁴	6.1359	10.3313	-	-	-	-
		10 ⁻⁵	9.6464	14.1615	-	-	-	-

Capítulo 4

Conclusiones

A lo largo de esta tesis doctoral hemos propuesto y analizado diferentes métodos para resolver problemas de optimización local y global, sin derivadas, con diferentes clases de restricciones.

En la primera parte, Capítulo 2, abordamos la resolución de un problema de optimización local con restricciones de cajas, basándonos en la filosofía de los métodos de búsqueda de patrones y estudiando su combinación con una técnica de búsqueda lineal no monótona. El desarrollo de la investigación de este punto resultó en la primera contribución de esta tesis doctoral. La misma consistió en un nuevo algoritmo de búsqueda de patrones (o búsqueda coordenadas) que utiliza una búsqueda lineal no monótona como estrategia de globalización, para el caso de optimización sin derivadas y con restricciones de cotas en las variables. Asimismo, se demostró que el nuevo algoritmo, denominado **nmps**, bajo hipótesis usuales en la teoría de optimización, garantiza la convergencia a un punto KKT del problema. Este resultado se encuentra fuertemente basado en la relación entre la longitud del paso y la medida de estacionariedad $\chi(x)$.

Desde el punto de vista numérico, se analizó el desempeño del nuevo método dividiendo el estudio en dos secciones. Por un lado, comenzamos con una comparación con su par **patternsearch**, disponible dentro del *toolbox* de optimización de **matlab**. Dado que ambos algoritmos se encuentran basados en un método de búsqueda de patrones, donde su principal diferencia radica en cómo y cuándo se acepta un nuevo iterado, consideramos que este paso inicial era fundamental para definir el camino de la experimentación numérica. Estos resultados experimentales preliminares (Sección 2.3.3.1) mostraron que el Algoritmo **nmps** logró competir, y en algunos casos superar, a **patternsearch** de **matlab**. En base a lo alentador de estos resultados experimentales, a continuación extendimos la comparación y el análisis numérico, utilizando otros algoritmos definidos para diferentes problemas (por ejemplo para solución de ecuaciones no lineales o para problemas de optimización con derivadas y sin restricciones) pero con la característica en común de utilizar esquemas alternativos de búsqueda lineal no monótona. Como conclusión final de esta parte, logramos validar que el desempeño del Algoritmo **nmps** es bastante estable a lo largo de diferentes escenarios, resultando competitivo frente métodos robustos propuestos en el pasado en el área de optimización.

Como trabajo futuro, quedaría realizar la extensión de esta nueva propuesta a problemas con restricciones lineales. En este caso, la mayor complejidad se encuentra en el manejo de la factibilidad de los iterados que involucra trabajar con los conos definidos en el Capítulo 1. En los últimos años han surgido trabajos que abordan este problema utilizando técnicas de filtro implícito y funciones con ruido. Los mismos podrían ser un disparador para nuevas investigaciones en esta línea.

La segunda parte de esta tesis doctoral, Capítulo 3, se encuentra enmarcada dentro de la teoría de optimización global sin derivadas. Así, dejando de lado el problema de encontrar mínimos locales, nos propusimos investigar diferentes alternativas para resolver problemas de opti-

mización global con restricciones generales. Luego del estudio de diferentes trabajos relacionados con el tema, surgió una posibilidad intermedia al objetivo propuesto. La misma consistió en aprovechar las bondades mostradas, a nivel teórico y numérico, del Algoritmo `nmps` (optimización local) dentro de un algoritmo de la familia DIRECT (optimización global), específicamente el Algoritmo BIRECT propuesto para restricciones de cotas en las variables. Esta idea nos resultó muy atractiva por varias razones. En primer lugar, vimos otra posibilidad de seguir validando el desempeño del algoritmo propuesto en el Capítulo 2. En segundo lugar y principalmente, nos propusimos determinar si la combinación de una búsqueda local (del tipo búsqueda de patrones) dentro de un algoritmo de optimización global, impacta positivamente o mejora la obtención de la solución encontrada. Dado que tanto el algoritmo BIRECT como el algoritmo `nmps` cuentan con resultados de convergencia, a nivel teórico la convergencia a la solución quedaba garantizada. Por esta razón, la investigación estuvo focalizada en un análisis numérico exhaustivo y formó parte de la segunda contribución de este tesis doctoral. Si bien, como se mencionó en las conclusiones de la Sección 3.2, los resultados no son aún generalizables pero sí alentadores. Esto es así puesto que se evidenció, en una buena porción de los casos, mejoras en la obtención de la solución y principalmente reducción en la cantidad de evaluaciones funcionales (parámetro por excelencia en la optimización sin derivadas). Algunas hipótesis que quedan por validar a futuro tienen que ver con: si la ejecución de la búsqueda local debe ser realizarse *todo el tiempo*, cada cierta cantidad de iteraciones (similar a técnicas de recomienzos) o en algún momento específico del proceso iterativo (como puede ser cuando los iterados se encuentren *lo suficientemente cerca de la solución*). De esta manera, consideramos que en esta línea hay mucho por explorar todavía.

Llegando a la parte final de este trabajo y por lo tanto a la última contribución, logramos proponer y validar un nuevo método de optimización global, sin derivadas y con restricciones generales. Así, inspirados en la técnica de partición del espacio del algoritmo BIRECT y en el manejo de las restricciones generales del algoritmo DIRECT, logramos extender el algoritmo BIRECT, inicialmente propuesto para restricciones de cajas, al caso con restricciones generales. Dado que el nuevo método hereda las propiedades de convergencia de sus algoritmos base la mayor complejidad estuvo dada por su implementación numérica debido a todas las variables involucradas y subproblemas relacionados. De esta manera, para poder lograr un desarrollo algorítmico lo más esquemático posible, por un lado se analizaron y presentaron los cambios propuestos para el caso general, particularmente en el cálculo de los coeficientes s_j . Por otro lado, se presentó un desarrollo detallado del proceso de selección de hiperrectángulos potencialmente óptimos para el nuevo problema y su relación con la cápsula convexa inferior. Así, se logró presentar el algoritmo BIRECT con restricciones generales como un conjunto de algoritmos mutuamente relacionados englobando todo el procedimiento.

Considerando la complejidad del problema, se decidió medir el desempeño algorítmico del nuevo método a través de un estudio numérico de las diversas variables involucradas: dimensionalidad, criterios de parada, obtención de puntos factibles y comportamiento de las cápsulas convexas. Si bien el conjunto de problemas test disponibles en la literatura para esta clase de problemas no es lo suficientemente amplio, se mostraron y analizaron algunos resultados numéricos. Además pudimos determinar que nuestra propuesta es una alternativa viable y eventualmente competitiva para resolver problemas de optimización global con restricciones generales. Asimismo, quedó claro que al momento de implementarlo en algún problema real, es importante tener presente todas las variables involucradas y determinar de antemano una escala de prioridad o *trade-off* entre ellas, para poder interpretar los resultados de la manera más apropiada.

A modo de cierre, y pensando en trabajos futuros, consideramos que este nuevo método de optimización global para el caso general tiene mucho aún para investigar, analizar y mejorar.

Apéndice A

Problemas de optimización local con restricciones de cajas

Colección Hock–Schittkowski (con modificaciones).

Problema A.1. *Problema 1 - Hock–Schittkowski*

$$\begin{aligned} \underset{x \in \mathbb{R}^2}{\text{minimizar}} \quad & f(x) = 100(x_2 - x_1^2)^2 + (1 - x_1)^2 \\ \text{sujeto a} \quad & -1.5 \leq x_2 \end{aligned}$$

Punto inicial: $x^0 = (-2 \ 1)$ con $f(x^0) = 909$.

Mínimo local: $f(x^*) = 0$ en $x^* = (1 \ 1)$.

Problema A.2. *Problema 2 - Hock–Schittkowski*

$$\begin{aligned} \underset{x \in \mathbb{R}^2}{\text{minimizar}} \quad & f(x) = 100(x_2 - x_1^2)^2 + (1 - x_1)^2 \\ \text{sujeto a} \quad & 1.5 \leq x_2 \end{aligned}$$

Punto inicial: $x^0 = (-2 \ 1.5)$ con $f(x^0) = 634$.

Mínimo local: $f(x^*) = 0.054261879$ en $x^* = \left(2a \cos\left(\frac{1}{3} \arccos\left(\frac{1}{b}\right)\right) \ 1.5 \right)$, donde $a = \sqrt{\frac{598}{1200}}$ y $b = 400a^3$.

Problema A.3. *Problema 3 - Hock–Schittkowski*

$$\begin{aligned} \underset{x \in \mathbb{R}^2}{\text{minimizar}} \quad & f(x) = x_2 + 10^{-5}(x_2 - x_1)^2 \\ \text{sujeto a} \quad & 0 \leq x_2 \end{aligned}$$

Punto inicial: $x^0 = (10 \ 1)$ con $f(x^0) = 1.00081$.

Mínimo local: $f(x^*) = 0$ en $x^* = (0 \ 0)$.

Problema A.4. *Problema 4 - Hock–Schittkowski*

$$\begin{aligned} \underset{x \in \mathbb{R}^2}{\text{minimizar}} \quad & f(x) = \frac{1}{3}(x_1 + 1)^3 + x_2 \\ \text{sujeto a} \quad & 1 \leq x_1 \\ & 0 \leq x_2 \end{aligned}$$

Punto inicial: $x^0 = (1.125 \ 0.125)$ con $f(x^0) = 3.323568$.

Mínimo local: $f(x^*) = \frac{8}{3}$ en $x^* = (1 \ 0)$.

Problema A.5. *Problema 5 - Hock-Schittkowski*

$$\begin{array}{ll} \underset{x \in \mathbb{R}^2}{\text{minimizar}} & f(x) = \sin(x_1 + x_2) + (x_1 - x_2)^2 - 1.5x_1 + 2.5x_2 + 1 \\ \text{sujeto a} & -1.5 \leq x_1 \leq 4 \\ & -3 \leq x_2 \leq 3 \end{array}$$

Punto inicial: $x^0 = (0 \ 0)$ con $f(x^0) = 1$.

Mínimo local: $f(x^*) = -\frac{1}{2}\sqrt{3} - \frac{\pi}{3}$ en $x^* = \left(-\frac{\pi}{3} + \frac{1}{2} \quad -\frac{\pi}{3} - \frac{1}{2}\right)$.

Problema A.6. *Problema 25 - Hock-Schittkowski*

$$\begin{array}{ll} \underset{x \in \mathbb{R}^3}{\text{minimizar}} & f(x) = \sum_{i=1}^{99} f_i(x)^2 \\ \text{sujeto a} & 0.1 \leq x_1 \leq 100 \\ & 0 \leq x_2 \leq 25.6 \\ & 0 \leq x_3 \leq 5 \end{array}$$

con

$$\begin{aligned} f_i(x) &= -0.01i + \exp\left(-\frac{1}{x_i}(u_i(x) - x_2)^{x_3}\right) \text{ y} \\ u_i(x) &= 25 + \left(-50 \ln(0.01i)\right)^{\frac{2}{3}} \text{ para } i = 1, \dots, 99. \end{aligned}$$

Punto inicial: $x^0 = (100 \ 12.5 \ 3)$ con $f(x^0) = 32.835$.

Mínimo local: $f(x^*) = 0$ en $x^* = (50 \ 25 \ 1.5)$.

Problema A.7. *Problema 38 - Hock-Schittkowski*

$$\begin{array}{ll} \underset{x \in \mathbb{R}^4}{\text{minimizar}} & f(x) = 100(x_2 - x_1^2)^2 + (1 - x_1)^2 + 90(x_4 - x_3^2)^2 + (1 - x_3)^2 \\ & + 10.1\left((x_2 - 1)^2 + (x_4 - 1)^2\right) + 19.8(x_2 - 1)(x_4 - 1) \\ \text{sujeto a} & -10 \leq x_i \leq 10, \ i = 1, 2, 3, 4. \end{array}$$

Punto inicial: $x^0 = -(3 \ 1 \ 3 \ 1)$ con $f(x^0) = 19192$.

Mínimo local: $f(x^*) = 0$ en $x^* = (1 \ 1 \ 1 \ 1)$.

Problema A.8. *Problema 45 - Hock-Schittkowski*

$$\begin{array}{ll} \underset{x \in \mathbb{R}^5}{\text{minimizar}} & f(x) = 2 - \frac{1}{120}x_1x_2x_3x_4x_5 \\ \text{sujeto a} & 0 \leq x_i \leq i, \ i = 1, \dots, 5. \end{array}$$

Punto inicial: $x^0 = (1 \ 2 \ 2 \ 2 \ 2)$ con $f(x^0) = 1.8\widehat{6}$.

Mínimo local: $f(x^*) = 1$ en $x^* = (1 \ 2 \ 3 \ 4 \ 5)$.

Problema A.9. *Problema 110 - Hock-Schittkowski*

$$\begin{aligned} \underset{x \in \mathbb{R}^{10}}{\text{minimizar}} \quad & f(x) = \sum_{i=1}^{10} \left(\left(\ln(x_i - 2) \right)^2 + \left(\ln(10 - x_i) \right)^2 \right) - \left(\prod_{i=1}^{10} x_i \right)^2 \\ \text{sujeeto a} \quad & 2.001 \leq x_i \leq 9.999, \quad i = 1, \dots, 10. \end{aligned}$$

Punto inicial: $x^0 = (9 \ 9 \ 9 \ 9 \ 9 \ 9 \ 9 \ 9 \ 9 \ 9)$ con $f(x^0) = -43.134337$.

Mínimo local: $f(x^*) = -45.77846971$ en

$$x^* = (9.35025655 \ 9.35025655 \ 9.35025655 \ 9.35025655 \ 9.35025655 \\ 9.35025655 \ 9.35025655 \ 9.35025655 \ 9.35025655 \ 9.35025655)$$

Problema A.10. *Problema 13 - Hock-Schittkowski*

$$\begin{aligned} \underset{x \in \mathbb{R}^2}{\text{minimizar}} \quad & f(x) = (x_1 - 2)^2 + x_2^2 \\ \text{sujeeto a} \quad & 0 \leq x_i, \quad i = 1, 2. \end{aligned}$$

Punto inicial: $x^0 = (0 \ 0)$ con $f(x^0) = 4$.

Mínimo local: $f(x^*) = 0$ en $x^* = (2 \ 0)$.

Problema A.11. *Problema 15 - Hock-Schittkowski*

$$\begin{aligned} \underset{x \in \mathbb{R}^2}{\text{minimizar}} \quad & f(x) = 100(x_2 - x_1^2)^2 + (1 - x_1)^2 \\ \text{sujeeto a} \quad & x_1 \leq 0.5 \end{aligned}$$

Punto inicial: $x^0 = (-2 \ 1)$ con $f(x^0) = 909$.

Mínimo local: $f(x^*) = 0.25$ en $x^* = (0.5 \ 0.249)$.

Problema A.12. *Problema 16 - Hock-Schittkowski*

$$\begin{aligned} \underset{x \in \mathbb{R}^2}{\text{minimizar}} \quad & f(x) = 100(x_2 - x_1^2)^2 + (1 - x_1)^2 \\ \text{sujeeto a} \quad & -0.5 \leq x_1 \leq 0.5 \\ & x_2 \leq 1 \end{aligned}$$

Punto inicial: $x^0 = (-0.5 \ 1)$ con $f(x^0) = 58.5$.

Mínimo local: $f(x^*) = 0.25$ en $x^* = (0.5 \ 0.24999999)$.

Problema A.13. *Problema 17 - Hock-Schittkowski*

$$\begin{aligned} \underset{x \in \mathbb{R}^2}{\text{minimizar}} \quad & f(x) = 100(x_2 - x_1^2)^2 + (1 - x_1)^2 \\ \text{sujeeto a} \quad & -0.5 \leq x_1 \leq 0.5 \\ & 0 \leq x_2 \leq 2 \end{aligned}$$

Punto inicial: $x^0 = (-0.5 \ 1)$ con $f(x^0) = 58.5$.

Mínimo local: $f(x^*) = 0.25$ en $x^* = (0.5 \ 0.24999999)$.

Problema A.14. *Problema 18 - Hock-Schittkowski*

$$\begin{aligned} \underset{x \in \mathbb{R}^2}{\text{minimizar}} \quad & f(x) = 0.01x_1^2 + x_2^2 \\ \text{sujeto a} \quad & 2 \leq x_1 \leq 50 \\ & 0 \leq x_2 \leq 50 \end{aligned}$$

Punto inicial: $x^0 = (2 \ 2)$ con $f(x^0) = 4.04$.

Mínimo local: $f(x^*) = 0.04$ en $x^* = (2 \ 0)$.

Problema A.15. *Problema 19 - Hock-Schittkowski*

$$\begin{aligned} \underset{x \in \mathbb{R}^2}{\text{minimizar}} \quad & f(x) = (x_1 - 10)^3 + (x_2 - 20)^3 \\ \text{sujeto a} \quad & 13 \leq x_1 \leq 100 \\ & 0 \leq x_2 \leq 100 \end{aligned}$$

Punto inicial: $x^0 = (20.1 \ 5.84)$ con $f(x^0) = -1808.858296$.

Mínimo local: $f(x^*) = -7973$ en $x^* = (13 \ 0)$.

Problema A.16. *Problema 20 - Hock-Schittkowski*

$$\begin{aligned} \underset{x \in \mathbb{R}^2}{\text{minimizar}} \quad & f(x) = 100(x_2 - x_1^2)^2 + (1 - x_1)^2 \\ \text{sujeto a} \quad & -0.5 \leq x_1 \leq 0.5 \end{aligned}$$

Punto inicial: $x^0 = (-0.5 \ 1)$ con $f(x^0) = 58.5$.

Mínimo local: $f(x^*) = 0.25$ en $x^* = (0.5 \ 0.24999999)$.

Problema A.17. *Problema 21 - Hock-Schittkowski*

$$\begin{aligned} \underset{x \in \mathbb{R}^2}{\text{minimizar}} \quad & f(x) = 0.01x_1^2 + x_2^2 - 100 \\ \text{sujeto a} \quad & 2 \leq x_1 \leq 50 \\ & -50 \leq x_2 \leq 50 \end{aligned}$$

Punto inicial: $x^0 = (2 \ -1)$ con $f(x^0) = -98.96$.

Mínimo local: $f(x^*) = -99.96$ en $x^* = (2 \ 0)$.

Problema A.18. *Problema 23 - Hock-Schittkowski*

$$\begin{aligned} \underset{x \in \mathbb{R}^2}{\text{minimizar}} \quad & f(x) = x_1^2 + x_2^2 \\ \text{sujeto a} \quad & -50 \leq x_i \leq 50, \quad i = 1, 2. \end{aligned}$$

Punto inicial: $x^0 = (3 \ 1)$ con $f(x^0) = 10$.

Mínimo local: $f(x^*) = 0$ en $x^* = (0 \ 0)$.

Problema A.19. *Problema 24 - Hock-Schittkowski*

$$\begin{aligned} \underset{x \in \mathbb{R}^2}{\text{minimizar}} \quad & f(x) = \frac{1}{27\sqrt{3}} \left((x_1 - 3)^2 - 9 \right) x_2^3 \\ \text{sujeto a} \quad & 0 \leq x_i, \quad i = 1, 2. \end{aligned}$$

Punto inicial: $x^0 = (1 \ 0.5)$ con $f(x^0) = -0.01336459$.

Mínimo local: $f(x^*) = -1.00385108 \times 10^{44}$ en $x^* = (5.1817337 \ 1.03452576 \times 10^{15})$.

Problema A.20. *Problema 30 - Hock-Schittkowski*

$$\begin{aligned} \underset{x \in \mathbb{R}^3}{\text{minimizar}} \quad & f(x) = x_1^2 + x_2^2 + x_3^2 \\ \text{sujeto a} \quad & 1 \leq x_1 \leq 10 \\ & -10 \leq x_2 \leq 10, \quad i = 2, 3 \end{aligned}$$

Punto inicial: $x^0 = (1 \ 1 \ 1)$ con $f(x^0) = 3$.

Mínimo local: $f(x^*) = 1$ en $x^* = (1 \ 0 \ 0)$.

Problema A.21. *Problema 31 - Hock-Schittkowski*

$$\begin{aligned} \underset{x \in \mathbb{R}^3}{\text{minimizar}} \quad & f(x) = 9x_1^2 + x_2^2 + 9x_3^2 \\ \text{sujeto a} \quad & -10 \leq x_1 \leq 10 \\ & 1 \leq x_2 \leq 10 \\ & -10 \leq x_3 \leq 1 \end{aligned}$$

Punto inicial: $x^0 = (1 \ 1 \ 1)$ con $f(x^0) = 19$.

Mínimo local: $f(x^*) = 1$ en $x^* = (0 \ 1 \ 0)$.

Problema A.22. *Problema 32 - Hock-Schittkowski*

$$\begin{aligned} \underset{x \in \mathbb{R}^3}{\text{minimizar}} \quad & f(x) = (x_1 + 3x_2 + x_3)^2 + 4(x_1 - x_2)^2 \\ \text{sujeto a} \quad & 0 \leq x_i, \quad i = 1, 2, 3. \end{aligned}$$

Punto inicial: $x^0 = (0.1 \ 0.7 \ 0.2)$ con $f(x^0) = 7.2$.

Mínimo local: $f(x^*) = 0$ en $x^* = (0 \ 0 \ 0)$.

Problema A.23. *Problema 33 - Hock-Schittkowski*

$$\begin{aligned} \underset{x \in \mathbb{R}^3}{\text{minimizar}} \quad & f(x) = (x_1 - 1)(x_1 - 2)(x_1 - 3) + x_3 \\ \text{sujeto a} \quad & 0 \leq x_1 \\ & 0 \leq x_2 \\ & 0 \leq x_3 \leq 5 \end{aligned}$$

Punto inicial: $x^0 = (0 \ 0 \ 3)$ con $f(x^0) = -3$.

Mínimo local: $f(x^*) = -6$ en $x^* = (0 \ 0 \ 0)$ y $x^* = (0 \ 15.68358993 \ 0)$.

Problema A.24. *Problema 34 - Hock-Schittkowski*

$$\begin{aligned} \underset{x \in \mathbb{R}^3}{\text{minimizar}} \quad & f(x) = -x_1 \\ \text{sujeto a} \quad & 0 \leq x_i \leq 100, \quad i = 1, 2. \\ & 0 \leq x_3 \leq 10 \end{aligned}$$

Punto inicial: $x^0 = (0 \ 1.05 \ 2.9)$ con $f(x^0) = 0$.

Mínimo local: $f(x^*) = -100$ en $x^* = (100 \ 1.05 \ 2.9)$.

Problema A.25. *Problema 35 - Hock-Schittkowski*

$$\begin{aligned} \underset{x \in \mathbb{R}^3}{\text{minimizar}} \quad & f(x) = 9 - 8x_1 - 6x_2 - 4x_3 + 2x_1^2 + 2x_2^2 \\ & + x_3^2 + 2x_1x_2 + 2x_1x_3 \\ \text{sujeto a} \quad & 0 \leq x_i, \quad i = 1, 2, 3. \end{aligned}$$

Punto inicial: $x^0 = (0.5 \ 0.5 \ 0.5)$ con $f(x^0) = 2.25$.

Mínimo local: $f(x^*) = 0$ en $x^* = (1 \ 1 \ 1)$.

Problema A.26. *Problema 36 - Hock-Schittkowski*

$$\begin{aligned} \underset{x \in \mathbb{R}^3}{\text{minimizar}} \quad & f(x) = -x_1x_2x_3 \\ \text{sujeto a} \quad & 0 \leq x_1 \leq 20 \\ & 0 \leq x_2 \leq 11 \\ & 0 \leq x_3 \leq 42 \end{aligned}$$

Punto inicial: $x^0 = (10 \ 10 \ 10)$ con $f(x^0) = -1000$.

Mínimo local: $f(x^*) = -9240$ en $x^* = (20 \ 11 \ 42)$.

Problema A.27. *Problema 37 - Hock-Schittkowski*

$$\begin{aligned} \underset{x \in \mathbb{R}^3}{\text{minimizar}} \quad & f(x) = -x_1x_2x_3 \\ \text{sujeto a} \quad & 0 \leq x_i \leq 42, \quad i = 1, 2, 3. \end{aligned}$$

Punto inicial: $x^0 = (10 \ 10 \ 10)$ con $f(x^0) = -1000$.

Mínimo local: $f(x^*) = -74088$ en $x^* = (42 \ 42 \ 42)$.

Problema A.28. *Problema 41 - Hock-Schittkowski*

$$\begin{aligned} \underset{x \in \mathbb{R}^4}{\text{minimizar}} \quad & f(x) = 2 - x_1x_2x_3 \\ \text{sujeto a} \quad & 0 \leq x_i \leq 1, \quad i = 1, 2, 3 \\ & 0 \leq x_4 \leq 2 \end{aligned}$$

Punto inicial: $x^0 = (1 \ 1 \ 1 \ 2)$ con $f(x^0) = 1$.

Mínimo local: $f(x^*) = 1$ en $x^* = (1 \ 1 \ 1 \ 2)$.

Problema A.29. *Problema 42 - Hock-Schittkowski*¹

$$\begin{aligned} \underset{x \in \mathbb{R}^4}{\text{minimizar}} \quad & f(x) = (x_1 - 1)^2 + (x_2 - 2)^2 + (x_3 - 3)^2 + (x_4 - 4)^2 \\ \text{sujeto a} \quad & x_1 - 2 = 0 \end{aligned}$$

Punto inicial: $x^0 = (2 \ 1 \ 1 \ 1)$ con $f(x^0) = 15$.

Mínimo local: $f(x^*) = 1$ en $x^* = (2 \ 1.\widehat{9} \ 2.\widehat{9} \ 2.\widehat{9} \ 3.\widehat{9})$.

¹A nivel implementación, la restricción de igualdad de este problema fue transformada en dos desigualdades equivalentes, $2 \leq x_1 \leq 2$, adaptándose así al tipo de entrada que recibe el algoritmo.

Problema A.30. *Problema 44 HS*

$$\begin{aligned} \underset{x \in \mathbb{R}^4}{\text{minimizar}} \quad & f(x) = x_1 - x_2 - x_3 - x_1x_3 + x_1x_4 + x_2x_3 - x_2x_4 \\ \text{sujeto a} \quad & 0 \leq x_i, \quad i = 1, \dots, 4. \end{aligned}$$

Punto inicial: $x^0 = (0 \ 0 \ 0 \ 0)$ con $f(x^0) = 0$.

Mínimo local: $f(x^*) = -15$ en $x^* = (0 \ 3 \ 0 \ 4)$.

Problema A.31. *Problema 53 - Hock-Schittkowski*

$$\begin{aligned} \underset{x \in \mathbb{R}^5}{\text{minimizar}} \quad & f(x) = (x_1 - x_2)^2 + (x_2 + x_3 - 2)^2 + (x_4 - 1)^2 + (x_5 - 1)^2 \\ \text{sujeto a} \quad & -10 \leq x_i \leq 10, \quad i = 1, \dots, 5. \end{aligned}$$

Punto inicial: $x^0 = (2 \ 2 \ 2 \ 2 \ 2)$ con $f(x^0) = 6$.

Mínimo global: $f(x^*) = 0$ en $x^* = (2 \ 2 \ 0 \ 1 \ 1)$.

Mínimos locales: $f(x^*) = 0.5$ en $x^* = (0.5 \ 0 \ 1.5 \ 1 \ 1)$ y $x^* = (1.5 \ 2 \ 0.5 \ 1 \ 1)$.

Problema A.32. *Problema 54 - Hock-Schittkowski*

$$\underset{x \in \mathbb{R}^6}{\text{minimizar}} \quad f(x) = \exp\left(\frac{-h(x)}{2}\right)$$

$$\begin{aligned} \text{sujeto a} \quad & 0 \leq x_1 \leq 2 \times 10^4 \\ & -10 \leq x_2 \leq 10 \\ & 0 \leq x_3 \leq 10^7 \\ & 0 \leq x_4 \leq 20 \\ & -1 \leq x_5 \leq 1 \\ & 0 \leq x_6 \leq 2 \times 10^8 \end{aligned}$$

donde

$$\begin{aligned} h(x) = & \frac{1}{1 - 0.2^2} \left(\frac{(x_1 - 10^4)^2}{6.4 \times 10^7} + \frac{(x_1 - 10^4)(x_2 - 1)}{2 \times 10^4} + (x_2 - 1)^2 \right) + \frac{(x_3 - 2 \times 10^6)^2}{4.9 \times 10^{13}} \\ & + \frac{(x_4 - 10)^2}{2.5 \times 10^3} + \frac{(x_5 - 10^{-3})^2}{2.5 \times 10^{-3}} + \frac{(x_6 - 10^8)^2}{2.5 \times 10^{17}} \end{aligned}$$

Punto inicial y punto mínimo local:

$x^0 = x^* = (6 \times 10^3 \ 1.5 \ 4 \times 10^6 \ 2 \ 3 \times 10^{-3} \ 5 \times 10^7)$ con $f(x^0) = f(x^*) = -0.7651$.

Problema A.33. *Problema 55 - Hock-Schittkowski*

$$\underset{x \in \mathbb{R}^5}{\text{minimizar}} \quad f(x) = x_1 + 2x_2 + 4x_5 + \exp(x_1x_4)$$

$$\begin{aligned} \text{sujeto a} \quad & 0 \leq x_1 \leq 1 \\ & 0 \leq x_2 \\ & 0 \leq x_3 \\ & 0 \leq x_4 \leq 1 \\ & 0 \leq x_5 \\ & 0 \leq x_6 \end{aligned}$$

Punto inicial: $x^0 = (1 \ 2 \ 0 \ 0 \ 0 \ 2)$ con $f(x^0) = 6$.

Mínimo local: $f(x^*) = 1$ en $x^* = (0 \ 0 \ 0 \ 0 \ 0 \ 2)$ y $x^* = (0 \ 0 \ 15.68358994 \ 0 \ 0 \ 2)$.

Problema A.34. Problema 57 - Hock-Schittkowski

$$\underset{x \in \mathbb{R}^2}{\text{minimizar}} f(x) = \sum_{i=1}^{44} \left(b_i - x_1 - (0.49 - x_1) \exp(-x_2(a_i - 8)) \right)^2$$

$$\text{sujeto a } 0.4 \leq x_1$$

$$-4 \leq x_2$$

con

$$a = \begin{pmatrix} 8 & 10 & 10 & 10 & 10 & 12 & 12 & 12 & 12 & 14 & 14 & 14 & 16 & 16 & 16 & 18 & 18 & 20 \\ 20 & 20 & 22 & 22 & 22 & 24 & 24 & 24 & 26 & 26 & 26 & 28 & 28 & 30 & 30 & 30 & 32 & 32 \\ 34 & 36 & 36 & 38 & 38 & 40 & 42 & 42 \end{pmatrix}$$

y

$$b = \begin{pmatrix} 49 & 0.49 & 0.48 & 0.47 & 0.48 & 0.47 & 0.46 & 0.46 & 0.45 & 0.43 & 0.45 & 0.43 & 0.43 \\ 0.44 & 0.43 & 0.43 & 0.46 & 0.45 & 0.42 & 0.42 & 0.43 & 0.41 & 0.41 & 0.40 & 0.42 & 0.40 \\ 0.40 & 0.41 & 0.40 & 0.41 & 0.41 & 0.40 & 0.40 & 0.40 & 0.38 & 0.41 & 0.40 & 0.40 & 0.41 \\ 0.38 & 0.40 & 0.40 & 0.39 & 0.39 \end{pmatrix}$$

Punto inicial: $x^0 = (0.42 \ 5)$ con $f(x^0) = 0.030798602$.

Mínimo local: $f(x^*) = -45.77846971$ en $x^* = (0.4 \ 0.12934752)$.

Problema A.35. Problema 59 - Hock-Schittkowski

$$\underset{x \in \mathbb{R}^2}{\text{minimizar}} f(x) =$$

$$\begin{aligned} & -75.196 + 3.8112x_1 + 0.0020567x_1^3 + 1.034 \times 10^{-5}x_1^4 + 6.8306x_2 \\ & - 0.030234x_1x_2 + 1.28134 \times 10^{-3}x_2x_2^2 - 0.12694x_1^2 + 2.266 \times 10^{-7}x_1^4x_2 \\ & - 0.25645x_2^2 + 0.0034604x_2^3 - 1.3514 \times 10^{-5}x_2^4 + \frac{28.106}{x_2 + 1} + 5.2375 \times 10^{-6}x_1^2x_2^2 \\ & + 6.3 \times 10^{-8}x_1^3x_2^2 - 7 \times 10^{-10}x_1^3x_2^3 - 3.405 \times 10^{-4}x_1x_2^2 + 1.6638 \times 10^{-6}x_1x_2^3 \\ & + 2.8673 \exp(0.0005x_1x_2) - 3.5256 \times 10^{-5}x_1^3x_2 \end{aligned}$$

$$\text{sujeto a } 0 \leq x_1 \leq 75$$

$$0 \leq x_2 \leq 65$$

Punto inicial: $x^0 = (75 \ 10)$ con $f(x^0) = 64.909$.

Mínimo local: $f(x^*) = -51.738$ en $x^* = (0 \ 1.1713)$.

Problema A.36. Problema 60 - Hock-Schittkowski

$$\underset{x \in \mathbb{R}^3}{\text{minimizar}} (x_1 - 1)^2 + (x_1 - x_2)^2 + (x_2 - x_3)^4$$

$$\text{sujeto a } -10 \leq x_i \leq 10, \ i = 1, 2, 3.$$

Punto inicial: $x^0 = (2 \ 2 \ 2)$ con $f(x^0) = 1$.

Mínimo local: $f(x^*) = 0$ en $x^* = (1 \ 1 \ 1)$.

Problema A.37. *Problema 62 - Hock-Schittkowski*

$$\begin{aligned} \underset{x \in \mathbb{R}^3}{\text{minimizar}} \quad f(x) = & -32.174 \left(255 \ln \left(\frac{x_1 + x_2 + x_3 + 0.03}{0.09x_1 + x_2 + x_3 + 0.03} \right) \right. \\ & \left. + 280 \ln \left(\frac{x_2 + x_3 + 0.03}{0.07x_2 + x_3 + 0.03} \right) + 290 \ln \left(\frac{x_3 + 0.03}{0.13x_3 + 0.03} \right) \right) \\ \text{sujeto a} \quad & 0 \leq x_i \leq 1, i = 1, 2, 3. \end{aligned}$$

Punto inicial: $x^0 = (0.7 \ 0.2 \ 0.1)$ con $f(x^0) = -25698.3$.

Mínimo local: $f(x^*) = -30525.10150981$ en $x^* = (1 \ 1 \ 0.11803961)$.

Problema A.38. *Problema 63 - Hock-Schittkowski*

$$\begin{aligned} \underset{x \in \mathbb{R}^3}{\text{minimizar}} \quad f(x) = & 1000 - x_1^2 - 2x_2^2 - x_3^2 - x_1x_2 - x_1x_3 \\ \text{sujeto a} \quad & 0 \leq x_i, i = 1, 2, 3. \end{aligned}$$

Punto inicial: $x^0 = (2 \ 2 \ 2)$ con $f(x^0) = 976$.

Mínimo local: $f(x^*) = -1.28157512 \times 10^{22}$ en $x^* = (40301977842.4316 \ 59516905687.0246 \ 25831518467.7993)$.

Problema A.39. *Problema 64 - Hock-Schittkowski*

$$\begin{aligned} \underset{x \in \mathbb{R}^3}{\text{minimizar}} \quad f(x) = & 5x_1 + \frac{50000}{x_1} + 20x_2 + \frac{72000}{x_2} + 10x_3 + \frac{144000}{x_3} \\ \text{sujeto a} \quad & 10^{-5} \leq x_i, i = 1, 2, 3. \end{aligned}$$

Punto inicial: $x^0 = (1 \ 1 \ 1)$ con $f(x^0) = 266035$.

Mínimo local: $f(x^*) = 5800.00000003$ en $x^* = (99.99938221 \ 60.00000694 \ 119.99971808)$.

Problema A.40. *Problema 65 - Hock-Schittkowski*

$$\begin{aligned} \underset{x \in \mathbb{R}^3}{\text{minimizar}} \quad f(x) = & (x_1 - x_2)^2 + \frac{(x_1 + x_2 - 10)^2}{9} + (x_3 - 5)^2 \\ \text{sujeto a} \quad & -4.5 \leq x_i \leq 4.5, i = 1, 2, \\ & -5 \leq x_3 \leq 5 \end{aligned}$$

Punto inicial: $x^0 = (-4.5 \ 4.5 \ 0)$ con $f(x_0) = \frac{1054}{9}$.

Mínimo local: $f(x^*) = \frac{1}{9}$ en $x^* = (4.5 \ 4.5 \ 5)$.

Problema A.41. *Problema 66 - Hock-Schittkowski*

$$\begin{aligned} \underset{x \in \mathbb{R}^3}{\text{minimizar}} \quad f(x) = & 0.2x_3 - 0.8x_1 \\ \text{sujeto a} \quad & 0 \leq x_i \leq 100, i = 1, 2, \\ & 0 \leq x_3 \leq 10 \end{aligned}$$

Punto inicial: $x^0 = (0 \ 1.05 \ 2.9)$ con $f(x_0) = 0.58$.

Mínimo local: $f(x^*) = -80$ en $x^* = (100 \ 1.05 \ 0)$.

Problema A.42. *Problema 68 - Hock-Schittkowski*

$$\begin{aligned} \underset{x \in \mathbb{R}^4}{\text{minimizar}} f(x) &= \frac{0.0024 - \left(\frac{\exp(x_1) - 1 - x_3}{\exp(x_1) - 1 + x_4} \right) x_4}{x_1} \\ \text{sujeto a } &0.0001 \leq x_1 \leq 100 \\ &0 \leq x_2 \leq 100 \\ &0 \leq x_3 \leq 2 \\ &0 \leq x_4 \leq 2 \end{aligned}$$

Punto inicial: $x^0 = (1 \ 1 \ 1 \ 1)$ con $f(x^0) = \frac{-2449}{9353}$.

Mínimo local: $f(x^*) = -\frac{467}{474}$ en

$x^* = (0.24425466 \ 1 \ 0 \ 2)$ y $x^* = (0.24425466 \ 12.41983414 \ 0 \ 2)$.

Problema A.43. *Problema 69 - Hock-Schittkowski*

$$\begin{aligned} \underset{x \in \mathbb{R}^4}{\text{minimizar}} f(x) &= \frac{0.4 - \left(\frac{1000(\exp(x_1) - 1) - x_3}{\exp(x_1) - 1 + x_4} \right) x_4}{x_1} \\ \text{sujeto a } &0.0001 \leq x_1 \leq 100 \\ &0 \leq x_2 \leq 100 \\ &0 \leq x_3 \leq 2 \\ &0 \leq x_4 \leq 2 \end{aligned}$$

Punto inicial: $x^0 = (1 \ 1 \ 1 \ 1)$ con $f(x^0) = \frac{-49671}{50}$.

Mínimo local: $f(x^*) = -\frac{93579}{94}$ en

$x^* = (0.134047 \ 1 \ 0 \ 2)$ y $x_* = (0.134047 \ 56.32727432 \ 0 \ 2)$.

Problema A.44. *Problema 71 - Hock-Schittkowski*

$$\begin{aligned} \underset{x \in \mathbb{R}^4}{\text{minimizar}} f(x) &= x_1 x_4 (x_1 + x_2 + x_3) + x_3 \\ \text{sujeto a } &1 \leq x_i \leq 5, \ i = 1, \dots, 4. \end{aligned}$$

Punto inicial: $x^0 = (1 \ 5 \ 5 \ 1)$ con $f(x^0) = 16$.

Mínimo local: $f(x^*) = 4$ en $x^* = (1 \ 1 \ 1 \ 1)$.

Problema A.45. *Problema 72 - Hock-Schittkowski*

$$\begin{aligned} \underset{x \in \mathbb{R}^4}{\text{minimizar}} f(x) &= 1 + x_1 + x_2 + x_3 + x_4 \\ \text{sujeto a } &0.001 \leq x_i \leq (5 - i) \times 10^5 \ i = 1, \dots, 4 \end{aligned}$$

Punto inicial: $x^0 = (1 \ 1 \ 1 \ 1)$ en $f(x_0) = 5$.

Mínimo local: $f(x^*) = \frac{251}{250}$ en $x^* = (9.999 \times 10^{-4} \ 10^{-3} \ 10^{-3} \ 10^{-3})$.

Problema A.46. *Problema 73 - Hock-Schittkowski*

$$\underset{x \in \mathbb{R}^4}{\text{minimizar}} \quad f(x) = 24.55x_1 + 26.75x_2 + 39x_3 + 40.50x_4$$

$$\text{sujeto a} \quad 0 \leq x_i, \quad i = 1, \dots, 4,$$

Punto inicial: $x^0 = (1 \quad 1 \quad 1 \quad 1)$ con $f(x^0) = 130.8$.

Mínimo local: $f(x^*) = 0$ en $x^* = (0 \quad 0 \quad 0 \quad 0)$.

Problema A.47. *Problema 74 - Hock-Schittkowski*

$$\underset{x \in \mathbb{R}^4}{\text{minimizar}} \quad f(x) = 3x_1 + 10^{-6}x_1^3 + 2x_2 + \frac{2}{3}10^{-6}x_2^3$$

$$\text{sujeto a} \quad 0 \leq x_1 \leq 1200$$

$$0 \leq x_2 \leq 1200$$

$$0 \leq x_3 \leq 0.55$$

$$0 \leq x_4 \leq 0.55$$

Punto inicial: $x^0 = (679.9453 \quad 1026.067 \quad 0.1188764 \quad -0.3962336)$ con $f(x^0) = 5126.4981$.

Mínimo local: $f(x^*) = 0$ en $x^* = (0 \quad 0 \quad 0 \quad 0)$.

Problema A.48. *Problema 75 - Hock-Schittkowski*

$$\underset{x \in \mathbb{R}^4}{\text{minimizar}} \quad f(x) = 3x_1 + 10^{-6}x_1^3 + 2x_2 + \frac{2}{3}10^{-6}x_2^3$$

$$\text{sujeto a} \quad 0 \leq x_i \leq 1200, \quad i = 1, 2,$$

$$-0.48 \leq x_i \leq 0.48, \quad i = 3, 4.$$

Punto inicial: $x^0 = (776.1592 \quad 925.1949 \quad 0.05110879 \quad -0.4288911)$ con $f(x^0) = 5174.4129$.

Mínimo local: $f(x^*) = 0$ en $x^* = (0 \quad 0 \quad 0 \quad 0)$.

Problema A.49. *Problema 76 - Hock-Schittkowski*

$$\underset{x \in \mathbb{R}^4}{\text{minimizar}} \quad f(x) = x_1^2 + 0.5x_2^2 + x_3^2 + 0.5x_4^2$$

$$-x_1x_3 + x_3x_4 - x_1 - 3x_2 + x_3 - x_4$$

$$\text{sujeto a} \quad 0 \leq x_i, \quad i = 1, \dots, 4.$$

Punto inicial: $x^0 = (0.5 \quad 0.5 \quad 0.5 \quad 0.5)$ con $f(x^0) = -1.25$.

Mínimo local: $f(x^*) = -5.25$ en $x^* = (0.49 \quad 3 \quad 0 \quad 0.9)$.

Problema A.50. *Problema 80 - Hock-Schittkowski*

$$\underset{x \in \mathbb{R}^5}{\text{minimizar}} \quad f(x) = \exp(x_1x_2x_3x_4x_5)$$

$$\text{sujeto a} \quad -2.3 \leq x_i \leq 2.3, \quad i = 1, 2,$$

$$-3.2 \leq x_i \leq 3.2, \quad i = 3, 4, 5.$$

Punto inicial: $x^0 = (-2 \quad 2 \quad 2 \quad -1 \quad -1)$, con $f(x_0) = 3.3546 \times 10^{-4}$.

Mínimo local: $f(x^*) = 5.48446889 \times 10^{-10}$ en

$x^* = (-2.20591573 \quad 2.20591572 \quad 2.20591572 \quad -1.4094519 \quad -1.40945189)$.

Problema A.51. *Problema 81 - Hock-Schittkowski*

$$\begin{aligned} \underset{x \in \mathbb{R}^5}{\text{minimizar}} \quad & f(x) = \exp(x_1 x_2 x_3 x_4 x_5) - 0.5(x_1^3 + x_2^3 + 1)^2 \\ \text{sujeto a} \quad & -2.3 \leq x_i \leq 2.3, \quad i = 1, 2, \\ & -3.2 \leq x_i \leq 3.2, \quad i = 3, 4, 5. \end{aligned}$$

Punto inicial: $x^0 = (-2 \quad 2 \quad 2 \quad -1 \quad -1)$ con $f(x^0) = -0.49966$.

Mínimo local: $f(x^*) = -86.67591657$ en

$$x^* = (-0.07671159 \quad 2.29999999 \quad 3.2 \quad -3.2 \quad -3.2).$$

Problema A.52. *Problema 83 - Hock-Schittkowski*

$$\begin{aligned} \underset{x \in \mathbb{R}^5}{\text{minimizar}} \quad & f(x) = 5.3578547x_3^2 + 0.8356891x_1x_5 + 37.293239x_1 - 40792.141 \\ \text{sujeto a} \quad & 78 \leq x_1 \leq 102 \\ & 33 \leq x_2 \leq 45 \\ & 27 \leq x_i \leq 45, \quad i = 3, 4, 5. \end{aligned}$$

Punto inicial: $x^0 = (78 \quad 33 \quad 27 \quad 27 \quad 27)$ con $f(x^0) = -3.2217 \times 10^4$.

Mínimo local: $f(x^*) = -3.2217 \times 10^4$ en $x^* = (78 \quad 33 \quad 27 \quad 27 \quad 27)$.

Problema A.53. *Problema 84 - Hock-Schittkowski*

$$\begin{aligned} \underset{x \in \mathbb{R}^5}{\text{minimizar}} \quad & f(x) = 24345 + 8720288.849x_1 - 150512.5253x_1x_2 \\ & + 156.6950325x_1x_3 - 476470.3222x_1x_4 - 729482.8271x_1x_5 \\ \text{sujeto a} \quad & 0 \leq x_1 \leq 1000 \\ & 1.2 \leq x_2 \leq 2.4 \\ & 20 \leq x_3 \leq 60 \\ & 9 \leq x_4 \leq 9.3 \\ & 6.5 \leq x_5 \leq 7 \end{aligned}$$

Punto inicial: $x^0 = (2.52 \quad 2 \quad 37.5 \quad 9.25 \quad 6.8)$ con $f(x^0) = -2.3512 \times 10^6$.

Mínimo local: $f(x^*) = -1175336752.23$ en $x^* = (1000 \quad 2.4 \quad 20 \quad 9.3 \quad 7)$.

Problema A.54. *Problema 86 - Hock-Schittkowski*

$$\begin{aligned} \underset{x \in \mathbb{R}^5}{\text{minimizar}} \quad & f(x) = \sum_{j=1}^5 e_j x_j + \sum_{i=1}^5 \sum_{j=1}^5 c_{ij} x_i x_j + \sum_{j=1}^5 d_j x_j^3 \\ \text{sujeto a} \quad & 0 \leq x_i, \quad i = 1, \dots, 5 \end{aligned}$$

donde

$$e = (-15 \quad -27 \quad -36 \quad -18 \quad -12), \quad c = \begin{pmatrix} 30 & -20 & -10 & 32 & -10 \\ -20 & 39 & -6 & -31 & 32 \\ -10 & -6 & 10 & -6 & -10 \\ 32 & -31 & -6 & 39 & -20 \\ -10 & 32 & -10 & -20 & 30 \end{pmatrix},$$

$$y \quad d = (4 \quad 8 \quad 10 \quad 6 \quad 2).$$

Punto inicial: $x^0 = (0 \ 0 \ 0 \ 0 \ 0)$ con $f(x^0) = 20$.

Mínimo local: $f(x^*) = -61.44833483$ en

$$x^* = (0.52426687 \ 0.88267453 \ 1.25847852 \ 0.74115071 \ 0.33556878).$$

Problema A.55. *Problema 93 - Hock-Schittkowski*

$$\begin{aligned} \underset{x \in \mathbb{R}^6}{\text{minimizar}} \quad & f(x) = 0.0204x_1x_4(x_1 + x_2 + x_3) \\ & + 0.0187x_2x_3(x_1 + 1.57x_2 + x_4) \\ & + 0.0607x_1x_4x_5^2(x_1 + x_2 + x_3) \\ & + 0.0437x_2x_3x_6^2(x_1 + 1.57x_2 + x_4) \\ \text{sujeto a} \quad & 0 \leq x_i, \ i = 1, \dots, 6 \end{aligned}$$

Punto inicial: $x^0 = (5.54 \ 4.4 \ 12.02 \ 11.82 \ 0.702 \ 0.852)$ con $f(x^0) = 137.066$.

Mínimo local: $f(x^*) = 0$ en $x^* = (0 \ 0 \ 3.34132911 \ 3.03225739 \ 0 \ 0)$.

Problema A.56. *Problema 101 - Hock-Schittkowski*

$$\begin{aligned} \underset{x \in \mathbb{R}^7}{\text{minimizar}} \quad & f(x) = 10x_1x_2^{-1}x_4^2x_6^{-3}x_7^{-0.25} \\ & + 15x_1^{-1}x_2^{-2}x_3x_4x_5^{-1}x_7^{-0.5} + \\ & 20x_1^{-2}x_2x_4^{-1}x_5^{-2}x_6 + 25x_1^2x_2^2x_3^{-1}x_5^{0.5}x_6^{-2}x_7 \\ \text{sujeto a} \quad & 0.1 \leq x_i \leq 10, \ i = 1, \dots, 7 \end{aligned}$$

Punto inicial: $x^0 = (6 \ 6 \ 6 \ 6 \ 6 \ 6 \ 6)$ con $f(x^0) = 2205.868$.

Mínimo local: $f(x^*) = 0.69340$ en $x^* = (6.92088783 \ 0.26314591 \ 0.1 \ 0.1 \ 10 \ 10 \ 0.01)$.

Problema A.57. *Problema 102 - Hock-Schittkowski*

$$\begin{aligned} \underset{x \in \mathbb{R}^7}{\text{minimizar}} \quad & f(x) = 10x_1x_2^{-1}x_4^2x_6^{-3}x_7^{0.125} + 15x_1^{-1}x_2^{-2}x_3x_4x_5^{-1}x_7^{-0.5} + \\ & 20x_1^{-2}x_2x_4^{-1}x_5^{-2}x_6 + 25x_1^2x_2^2x_3^{-1}x_5^{0.5}x_6^{-2}x_7 \\ \text{sujeto a} \quad & 0.1 \leq x_i \leq 10, \ i = 1, \dots, 6, \\ & 0.01 \leq x_7 \leq 10, \end{aligned}$$

Punto inicial: $x^0 = (6 \ 6 \ 6 \ 6 \ 6 \ 6 \ 6)$ con $f(x^0) = 2208.886$.

Mínimo local: $f(x_*) = 0.68648863$ en

$$x^* = (7.00440918 \ 0.26050608 \ 0.1 \ 0.1 \ 10 \ 10 \ 0.01).$$

Problema A.58. *Problema 103 - Hock-Schittkowski*

$$\begin{aligned} \underset{x \in \mathbb{R}^7}{\text{minimizar}} \quad & f(x) = 10x_1x_2^{-1}x_4^2x_6^{-3}x_7^{0.5} + 15x_1^{-1}x_2^{-2}x_3x_4x_5^{-1}x_7^{-0.5} + \\ & 20x_1^{-2}x_2x_4^{-1}x_5^{-2}x_6 + 25x_1^2x_2^2x_3^{-1}x_5^{0.5}x_6^{-2}x_7 \\ \text{sujeto a} \quad & 0.1 \leq x_i \leq 10, \ i = 1, \dots, 6, \\ & 0.01 \leq x_7 \leq 10, \end{aligned}$$

Punto inicial: $x^0 = (6 \ 6 \ 6 \ 6 \ 6 \ 6 \ 6)$ con $f(x^0) = 2205.889$.

Mínimo local: $f(x^*) = 0.68524295$ en

$$x^* = (7.02003104 \ 0.26002001 \ 0.1 \ 0.1 \ 10 \ 10 \ 0.01).$$

Problema A.59. *Problema 104 - Hock-Schittkowski*

$$\begin{aligned} & \underset{x \in \mathbb{R}^8}{\text{minimizar}} f(x) = 0.4x_1^{0.67}x_7^{-0.67} + 0.4x_2^{0.67}x_8^{-0.67} + 10 - x_1 - x_2 \\ & \text{sujeto a } 0.1 \leq x_i \leq 10, i = 1, \dots, 6. \\ & \quad 0.01 \leq x_i \leq 10, i = 7, 8. \end{aligned}$$

Punto inicial: $x^0 = (6 \ 3 \ 0.4 \ 0.2 \ 6 \ 6 \ 1 \ 0.5)$ con $f(x^0) = 3.65$.

Mínimo local: $f(x^*) = -9.2$ en $x^* = (10 \ 10 \ 0.4 \ 0.2 \ 6 \ 6 \ 10 \ 10)$.

Problema A.60. *Problema 106 - Hock-Schittkowski*

$$\begin{aligned} & \underset{x \in \mathbb{R}^8}{\text{minimizar}} f(x) = x_1 + x_2 + x_3 \\ & \text{sujeto a } 100 \leq x_1 \leq 10000 \\ & \quad 1000 \leq x_2 \leq 10000 \\ & \quad 1000 \leq x_3 \leq 10000 \\ & \quad 10 \leq x_i \leq 10000, i = 4, \dots, 8. \end{aligned}$$

Punto inicial: $x^0 = (5000 \ 5000 \ 5000 \ 200 \ 350 \ 150 \ 225 \ 425)$ con $f(x^0) = 15000$.

Mínimo local: $f(x^*) = 2100$ en $x^* = (100 \ 1000 \ 1000 \ 200 \ 350 \ 150 \ 225 \ 425)$.

Problema A.61. *Problema 108 - Hock-Schittkowski*

$$\begin{aligned} & \underset{x \in \mathbb{R}^9}{\text{minimizar}} f(x) = -0.5(x_1x_4 - x_2x_3 + x_3x_9 - x_5x_9 + x_5x_8 - x_6x_7) \\ & \text{sujeto a } 0 \leq x_9 \leq 1 \end{aligned}$$

Punto inicial: $x^0 = (1 \ 1 \ 1 \ 1 \ 1 \ 1 \ 1 \ 1 \ 1)$ con $f(x^0) = 0$.

Mínimo local: $f(x^*) = -4.1769 \times 10^{19}$ en

$$x^* = \begin{pmatrix} 8.8097 \times 10^9 & -6.7457 \times 10^9 & 2.0640 \times 10^9 & 8.8097 \times 10^9 & 2.0640 \times 10^9 \\ -4.6817 \times 10^9 & -4.6817 \times 10^9 & 6.7457 \times 10^9 & 1 & 0 \end{pmatrix}.$$

Problema A.62. *Problema 114 - Hock-Schittkowski*

$$\begin{aligned} & \underset{x \in \mathbb{R}^{10}}{\text{minimizar}} f(x) = 5.04x_1 + 0.035x_2 + 10x_3 + 3.36x_5 - 0.063x_4x_7 \\ & \text{sujeto a } 0.00001 \leq x_1 \leq 2000 \\ & \quad 0.00001 \leq x_2 \leq 16000 \\ & \quad 0.00001 \leq x_3 \leq 120 \\ & \quad 0.00001 \leq x_4 \leq 5000 \\ & \quad 0.00001 \leq x_5 \leq 2000 \\ & \quad 85 \leq x_6 \leq 93 \\ & \quad 90 \leq x_7 \leq 95 \\ & \quad 3 \leq x_8 \leq 12 \\ & \quad 1.2 \leq x_9 \leq 4 \\ & \quad 145 \leq x_{10} \leq 162 \end{aligned}$$

Punto inicial: $x^0 = (1745 \ 12000 \ 110 \ 3048 \ 1974 \ 89.2 \ 92.8 \ 8 \ 3.6 \ 145)$
con $f(x^0) = -872.3872$.

Mínimo local: $f(x^*) = -29924.99981565$ en

$$x^* = \begin{pmatrix} 1.0000000003 \times 10^{-5} & 9.99999975 \times 10^{-6} & 1.0000000003 \times 10^{-5} & 5 \times 10^3 \\ 9.99999975 \times 10^{-6} & 8.92 \times 10 & 9.5 \times 10 & 8 \\ 3.6 & 1.45 \times 10^2 & & \end{pmatrix}$$

Problema A.63. *Problema 119 - Hock-Schittkowski*

$$\begin{aligned} \underset{x \in \mathbb{R}^{16}}{\text{minimizar}} \quad & f(x) = \sum_{i=1}^{16} \sum_{j=1}^{16} a_{ij} (x_i^2 + x_i + 1)(x_j^2 + x_j + 1) \\ \text{sujeto a} \quad & 0 \leq x_i \leq 5 \quad i = 1, \dots, 16 \end{aligned}$$

donde

$$A = \begin{pmatrix} 1 & 0 & 0 & 1 & 0 & 0 & 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \\ 0 & 1 & 1 & 0 & 0 & 0 & 1 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 1 & 0 & 1 & 1 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 1 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 1 & 1 & 0 & 0 & 0 & 1 & 0 & 1 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 1 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 1 & 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 1 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \end{pmatrix}$$

Punto inicial: $x^0 = (1 \ 1 \ 1 \ 1 \ 1 \ 1 \ 1 \ 1 \ 1 \ 1 \ 1 \ 1 \ 1 \ 1 \ 1 \ 1 \ 1)$
con $f(x^0) = 566766$.

Mínimo local: $f(x^*) = 46$ en

$$x^* = (0 \ 0 \ 0 \ 0 \ 0 \ 0 \ 0 \ 0 \ 0 \ 0 \ 0 \ 0 \ 0 \ 0 \ 0 \ 0 \ 0).$$

Apéndice B

Problemas optimización global con restricciones generales

Problema B.1. *Problema 2.1 - Epperly*

$$\begin{aligned} & \underset{x \in \mathbb{R}^6}{\text{minimizar}} f(x) = x_1 \\ & \text{sujeto a} \quad g_1(x) \leq 0 \\ & \quad \quad \quad g_2(x) \leq 0 \\ & \quad \quad \quad -250 \leq x_1 \leq 225.5 \\ & \quad \quad \quad 0 \leq x_i \leq 1, i = 2, \dots, 6 \end{aligned}$$

donde

$$\begin{aligned} g_1(x) &= -x_1 + 42x_2 + 44x_3 + 45x_4 + 47x_5 + 47.5x_6 - 50x_2^2 - 50x_3^2 - 50x_4^2 - 50x_5^2 - 50x_6^2 \text{ y} \\ g_2(x) &= 20x_2 + 12x_3 + 11x_4 + 7x_5 + 4x_6 - 40. \end{aligned}$$

Mínimo global conocido: $f(x^) = -17$ en $x^* = (-17 \ 1 \ 1 \ 0 \ 1 \ 0)$.*

Problema B.2. *Problema 2.2 - Epperly*

$$\begin{aligned} & \underset{x \in \mathbb{R}^7}{\text{minimizar}} f(x) = x_1 \\ & \text{sujeto a} \quad g_1(x) \leq 0 \\ & \quad \quad \quad g_2(x) \leq 0 \\ & \quad \quad \quad g_3(x) \leq 0 \\ & \quad \quad \quad -230 \leq x_1 \leq 0 \\ & \quad \quad \quad 0 \leq x_i \leq 1, i = 2, \dots, 6. \\ & \quad \quad \quad 0 \leq x_7 \leq 20 \end{aligned}$$

donde

$$\begin{aligned} g_1(x) &= -x_1 - 10.5x_2 - 7.5x_3 - 3.5x_4 - 2.5x_5 - 1.5x_6 \\ & \quad - 10x_7 - 0.5x_2^2 - 0.5x_3^2 - 0.5x_4^2 - 0.5x_5^2 - 0.5x_6^2 \\ g_2(x) &= 6x_2 + 3x_3 + 3x_4 + 2x_5 + x_6 - 6.5 \text{ y} \\ g_3(x) &= 10x_2 + 10x_4 + x_7 - 20 \end{aligned}$$

Mínimo global conocido: $f(x^*) = -213$
 en $x^* = (-213 \quad 5.884182031 \times 10^{-15} \quad 1 \quad 0 \quad 1 \quad 1 \quad 20)$.

Problema B.3. *Problema 1.2 - Floudas*

$$\begin{aligned} & \underset{x \in \mathbb{R}^5}{\text{minimizar}} && f(x) = c^T x - 0.5x^T Q x \\ & \text{sujeto a} && g_1(x) \leq 0 \\ & && 0 \leq x_i \leq 1, \quad i = 1, \dots, 5. \end{aligned}$$

donde

$$g_1(x) = 20x_1 + 12x_2 + 11x_3 + 7x_4 + 4x_5 - 40, \quad c = (42 \quad 44 \quad 45 \quad 47 \quad 47.5),$$

$$y \quad Q = \begin{pmatrix} 100 & 0 & 0 & 0 & 0 \\ 0 & 100 & 0 & 0 & 0 \\ 0 & 0 & 100 & 0 & 0 \\ 0 & 0 & 0 & 100 & 0 \\ 0 & 0 & 0 & 0 & 100 \end{pmatrix}.$$

Mínimo global conocido: $f(x^*) = -17$ en $x^* = (1 \quad 1 \quad 0 \quad 1 \quad 0)$.

Problema B.4. *Problema 1.3 - Floudas*

$$\begin{aligned} & \underset{x \in \mathbb{R}^8}{\text{minimizar}} && f(x) = x_1 + x_2 + x_3 \\ & \text{sujeto a} && g_1(x) \leq 0 \\ & && g_2(x) \leq 0 \\ & && g_3(x) \leq 0 \\ & && g_4(x) \leq 0 \\ & && g_5(x) \leq 0 \\ & && g_6(x) \leq 0 \\ & && 100 \leq x_1 \leq 10000 \\ & && 1000 \leq x_i \leq 10000, \quad i = 2, 3. \\ & && 10 \leq x_i \leq 1000, \quad i = 4, \dots, 8. \end{aligned}$$

donde

$$\begin{aligned} g_1(x) &= -1 + 0.0025(x_4 + x_6) \\ g_2(x) &= -1 + 0.0025(-x_4 + x_5 + x_7) \\ g_3(x) &= -1 + 0.01(-x_5 + x_8) \\ g_4(x) &= 100x_1 - x_1x_6 + 833.33252x_4 - 83333.333 \\ g_5(x) &= x_2x_4 - x_2x_7 - 1250x_4 + 1250x_5 \quad y \\ g_6(x) &= x_3x_5 - x_3x_8 - 2500x_5 + 1250000 \end{aligned}$$

Mínimo global conocido: $f(x^*) = 7049.2$ en

$$x^* = \begin{pmatrix} 579.2770328 & 1359.958638 & 5110.012347 & 182.0152233 \\ 295.5995061 & 217.9847767 & 268.4157171 & 395.5995061 \end{pmatrix}.$$

Problema B.5. *Problema 2.3 - Floudas*

$$\begin{aligned}
& \underset{x \in \mathbb{R}^5}{\text{minimizar}} && f(x) = 37.293239x_1 + 0.8356891x_1x_5 + 5.3578547x_3^2 - 40792.141 \\
& \text{sujeto a} && g_1(x) \leq 0 \\
& && g_2(x) \leq 0 \\
& && g_3(x) \leq 0 \\
& && g_4(x) \leq 0 \\
& && g_5(x) \leq 0 \\
& && g_6(x) \leq 0 \\
& && 78 \leq x_1 \leq 102 \\
& && 33 \leq x_2 \leq 45 \\
& && 27 \leq x_i \leq 45, \quad i = 3, 4, 5.
\end{aligned}$$

donde

$$\begin{aligned}
g_1(x) &= -0.0022053x_3x_5 + 0.0056858x_2x_5 + 0.0006262x_1x_4 - 6.655593 \\
g_2(x) &= 0.022053x_3x_5 - 0.0056858x_2x_5 - 0.0006262x_1x_4 - 85.334407 \\
g_3(x) &= 0.0071317x_2x_5 + 0.0021813x_3^2 + 0.0029955x_1x_2 - 29.48751 \\
g_4(x) &= -0.0071317x_2x_5 - 0.0021813x_3^2 - 0.0029955x_1x_2 + 9.48751 \\
g_5(x) &= 0.0047026x_3x_5 + 0.0019085x_3x_4 + 0.0012547x_1x_3 - 15.699039 \\
g_6(x) &= -0.0047026x_3x_5 - 0.0019085x_3x_4 - 0.0012547x_1x_3 + 10.699039.
\end{aligned}$$

Mínimo global conocido: $f(x^*) = -30665.5387$ en $x^* = (78 \quad 33 \quad 29.9953 \quad 45 \quad 36.7758)$.

Problema B.6. *Problema 3.3 - Floudas*

$$\begin{aligned}
& \underset{x \in \mathbb{R}^6}{\text{minimizar}} && f(x) = -25(x_1 - 2)^2 - (x_2 - 2)^2 - (x_3 - 1)^2 \\
& && \quad - (x_4 - 4)^2 - (x_5 - 1)^2 - (x_6 - 4)^2 \\
& \text{sujeto a} && g_1(x) \leq 0 \\
& && g_2(x) \leq 0 \\
& && g_3(x) \leq 0 \\
& && g_4(x) \leq 0 \\
& && g_5(x) \leq 0 \\
& && g_6(x) \leq 0 \\
& && 0 \leq x_1 \leq 10 \\
& && 0 \leq x_2 \leq 5 \\
& && 1 \leq x_3 \leq 5 \\
& && 0 \leq x_4 \leq 6 \\
& && 1 \leq x_5 \leq 5 \\
& && 0 \leq x_6 \leq 10
\end{aligned}$$

donde

$$g_1(x) = 4 - (x_3 - 3)^2 - x_4$$

$$g_2(x) = 4 - (x_5 - 3)^2 - x_6$$

$$g_3(x) = x_1 - 3x_2 - 2$$

$$g_4(x) = -x_1 + x_2 - 2$$

$$g_5(x) = x_1 + x_2 - 6 \quad y$$

$$g_6(x) = 2 - x_1 - x_2.$$

Mínimo global conocido: $f(x^*) = -310$ en $x^* = (5 \ 1 \ 5 \ 0 \ 5 \ 10)$.

Problema B.7. Problema 4.3 - Floudas

$$\underset{x \in \mathbb{R}^3}{\text{minimizar}} \quad f(x) = -2x_1 + x_2 - x_3$$

$$\text{sujeto a} \quad g_1(x) \leq 0$$

$$g_2(x) \leq 0$$

$$g_3(x) \leq 0$$

$$0 \leq x_1 \leq 2$$

$$0 \leq x_2 \leq 2$$

$$0 \leq x_3 \leq 3$$

donde

$$g_1(x) = -x^T A^T A x + 2y^T A x - \|y\|^2 + 0.25 \|b - z\|^2$$

$$g_2(x) = x_1 + x_2 + x_3 - 4 \quad y$$

$$g_3(x) = 3x_2 + x_3 - 6$$

con

$$A = \begin{pmatrix} 0 & 0 & 1 \\ 0 & -1 & 0 \\ -2 & 1 & -1 \end{pmatrix}, \quad b = (3 \ 0 \ -4), \quad y = (1.5 \ -0.5 \ -5) \quad y \quad z = (0 \ -1 \ -6).$$

Mínimo global conocido: $f(x^*) = -4$ en $x^* = (0.5 \ 0 \ 3)$.

Problema B.8. Problema 8.4 - Floudas

$$\underset{x \in \mathbb{R}^2}{\text{minimizar}} \quad f(x) = -12x_1 - 7x_2 + x_2^2$$

$$\text{sujeto a} \quad g_1(x) = 0$$

$$0 \leq x_1 \leq 2$$

$$0 \leq x_2 \leq 3$$

donde $g_1(x) = -2x_1^4 + 2 - x_2$.

Mínimo global conocido: $f(x^*) = -16.73889$ en $x^* = (0.7175 \ 1.47)$.

Problema B.9. *Problema 9.4 - Floudas*

$$\begin{aligned} & \underset{x \in \mathbb{R}^2}{\text{minimizar}} && f(x) = -x_1 - x_2 \\ & \text{sujeto a} && g_1(x) \leq 0 \\ & && g_2(x) \leq 0 \\ & && 0 \leq x_1 \leq 3 \\ & && 0 \leq x_2 \leq 4 \end{aligned}$$

donde

$$\begin{aligned} g_1(x) &= x_2 - 2x_1^4 + 8x_1^3 - 8x_1^2 \\ g_2(x) &= x_2 - 4x_1^4 + 32x_1^3 - 88x_1^2 + 96x_1 - 36. \end{aligned}$$

Mínimo global conocido: $f(x^*) = -5.50796$ en $x^* = (2.3295 \quad 3.17846)$.

Problema B.10. *Problema 3.7 - Floudas*

$$\begin{aligned} & \underset{x \in \mathbb{R}^8}{\text{minimizar}} && f(x) = x_1 + x_2 + x_3 \\ & \text{sujeto a} && g_1(x) \leq 0 \\ & && g_2(x) \leq 0 \\ & && g_3(x) \leq 0 \\ & && g_4(x) \leq 0 \\ & && g_5(x) \leq 0 \\ & && g_6(x) \leq 0 \\ & && 100 \leq x_1 \leq 10000 \\ & && 1000 \leq x_i \leq 10000, \quad i = 2, 3. \\ & && 10 \leq x_i \leq 1000, \quad i = 4, \dots, 8. \end{aligned}$$

donde

$$\begin{aligned} g_1(x) &= 833.33252x_1^{-1}x_4x_6^{-1} + 100.0x_6^{-1} - 83333.333x_1^{-1}x_6^{-1} - 1 \\ g_2(x) &= 1250.0x_2^{-1}x_5x_7^{-1} + 1.0x_4x_7^{-1} - 1250.0x_2^{-1}x_4x_7^{-1} - 1 \\ g_3(x) &= 1250000x_3^{-1}x_8^{-1} + 1.0x_5x_8^{-1} - 2500.0x_3^{-1}x_5x_8^{-1} - 1 \\ g_4(x) &= 0.0025x_4 + 0.0025x_6 - 1 \\ g_5(x) &= -0.0025x_4 + 0.0025x_5 + 0.0025x_7 - 1 \\ g_6(x) &= 0.01x_8 - 0.01x_5 - 1. \end{aligned}$$

Mínimo global conocido: $f(x^*) = 7049.25$ en

$$x^* = (579.31 \quad 1359.97 \quad 5109.97 \quad 182.01 \quad 295.6 \quad 217.98 \quad 286.42 \quad 395.60).$$

Problema B.11. *Problema 6.7 - Floudas*

$$\begin{aligned} \underset{x \in \mathbb{R}^3}{\text{minimizar}} \quad & f(x) = 0.5x_1x_2^{-1} - x_1 - 5x_2^{-1} \\ \text{sujeto a} \quad & g_1(x) \leq 0 \\ & 1 \leq x_1 \leq 100 \\ & 1 \leq x_2 \leq 100 \\ & 1 \leq x_3 \leq 100 \end{aligned}$$

donde $g_1(x) = 0.01x_2x_3^{-1} + 0.01x_1 + 0.0005x_1x_3 - 1$.

Mínimo global conocido: $f(x^*) = -83.254$ en $x^* = (88.2890 \quad 7.7737 \quad 1.3120)$.

Problema B.12. *Problema 7.7 - Floudas*

$$\begin{aligned} \underset{x \in \mathbb{R}^4}{\text{minimizar}} \quad & f(x) = s - x_1 + 0.4x_1^{0.67}x_3^{-0.67} \\ \text{sujeto a} \quad & g_1(x) \leq 0 \\ & g_2(x) \leq 0 \\ & 0.1 \leq x_i \leq 10, \quad i = 1, 2, 3, 4. \end{aligned}$$

donde

$$\begin{aligned} g_1(x) &= 0.05882x_3x_4 + 0.1x_1 - 1 \quad y \\ g_2(x) &= 4x_2x_4^{-1} + 2x_2^{-0.71}x_4^{-1} + 0.05882x_2^{-1.3}x_3 - 1. \end{aligned}$$

Mínimo global conocido: $f(x^*) = -5.7398$ en $x^* = (8.1267 \quad 0.6154 \quad 0.5650 \quad 5.6368)$.

Problema B.13. *Problema 7.8 - Floudas*

$$\begin{aligned} \underset{x \in \mathbb{R}^5}{\text{minimizar}} \quad & f(x) = (x_1 - 1)^2 + (x_1 - x_2)^2 + (x_2 - x_3)^3 + (x_3 - x_4)^4 + (x_4 - x_5)^4 \\ \text{sujeto a} \quad & g_1(x) = 0 \\ & g_2(x) = 0 \\ & g_3(x) = 0 \\ & -5 \leq x_i \leq 5, \quad i = 1, \dots, 5. \end{aligned}$$

donde

$$\begin{aligned} g_1(x) &= x_1 + x_2^2 + x_3^3 - 3\sqrt{2} - 2 \\ g_2(x) &= x_2 - x_3^2 + x_4 - 2\sqrt{2} + 2 \quad y \\ g_3(x) &= x_1x_5 - 2. \end{aligned}$$

Mínimo global conocido: $f(x^*) = 0.0293$ en $x^* = (1.1166 \quad 1.2204 \quad 1.5378 \quad 1.9728 \quad 1.7911)$.

Problema B.14. *Problema G8*

$$\begin{aligned} \underset{x \in \mathbb{R}^2}{\text{minimizar}} \quad & f(x) = -\frac{\sin^3(2\pi x_1) \sin(2\pi x_2)}{x_1^3(x_1 + x_2)} \\ \text{sujeto a} \quad & g_1(x) \leq 0 \\ & g_2(x) \leq 0 \\ & 0 \leq x_i \leq 10, \quad i = 1, 2. \end{aligned}$$

donde

$$\begin{aligned} g_1(x) &= x_1^2 - x_2 + 1 \quad y \\ g_2(x) &= 1 - x_1 + (x_2 - 4)^2. \end{aligned}$$

Mínimo global conocido: $f(x^*) = -0.095825$ en $x^* = (1.2279713 \quad 4.2453733)$.

Apéndice C

Problemas optimización global con restricciones de cotas en las variables

Problema C.1. *Ackley₂*

$$\underset{x \in \mathbb{R}^d}{\text{minimizar}} \quad f(x) = -a \exp \left(-b \sqrt{\frac{1}{d} \sum_{i=1}^d x_i^2} \right) - \exp \left(\frac{1}{d} \sum_{i=1}^d \cos(cx_i) \right) + a + \exp(1)$$

$$\text{sujeto a} \quad -15 \leq x_i \leq 35, i = 1, \dots, d$$

con $d = 2$, $a = 20$, $b = 0.2$ y $c = 2\pi$.

Mínimo global: $f(x^*) = 0$ en $x^* = (0 \dots 0)$.

Problema C.2. *Ackley₅*

Idéntico al Problema C.1 con $d=5$.

Problema C.3. *Ackley₁₀*

Idéntico al Problema C.1 con $d=10$.

Problema C.4. *Beale*

$$\underset{x \in \mathbb{R}^2}{\text{minimizar}} \quad f(x) = (1.5 - x_1 + x_1x_2)^2 + (2.25 - x_1 + x_1x_2^2)^2 + (2.625 - x_1 + x_1x_2^3)^2$$

$$\text{sujeto a} \quad -4.5 \leq x_i \leq 4.5, i = 1, 2.$$

Mínimo global: $f(x^*) = 0$ en $x^* = (3 \ 0.5)$.

Problema C.5. *Bohachevsky 1*

$$\underset{x \in \mathbb{R}^2}{\text{minimizar}} \quad f(x) = x_1^2 + 2x_2^2 - 0.3 \cos(3\pi x_1) - 0.4 \cos(4\pi x_2) + 0.7$$

$$\text{sujeto a} \quad -100 \leq x_i \leq 110, i = 1, 2.$$

Mínimo global: $f(x^*) = 0$ en $x^* = (0 \ 0)$.

Problema C.6. *Bohachevsky 2*

$$\underset{x \in \mathbb{R}^2}{\text{minimizar}} \quad f(x) = x_1^2 + 2x_2^2 - 0.3 \cos(3\pi x_1) \cos(4\pi x_2) + 0.3$$

$$\text{sujeto a} \quad -100 \leq x_i \leq 110, i = 1, 2.$$

Mínimo global: $f(x^*) = 0$ en $x^* = (0 \ 0)$.

Problema C.7. *Bohachevsky 3*

$$\underset{x \in \mathbb{R}^2}{\text{minimizar}} \quad f(x) = x_1^2 + 2x_2^2 - 0.3 \cos(3\pi x_1 + 4\pi x_2) + 0.3$$

$$\text{sujeto a} \quad -100 \leq x_i \leq 110, i = 1, 2.$$

Mínimo global: $f(x^*) = 0$ en $x^* = (0 \ 0)$.

Problema C.8. *Booth*

$$\underset{x \in \mathbb{R}^2}{\text{minimizar}} \quad f(x) = (x_1 + 2x_2 - 7)^2 + (2x_1 + x_2 - 5)^2$$

$$\text{sujeto a} \quad -10 \leq x_i \leq 10, i = 1, 2.$$

Mínimo global: $f(x^*) = 0$ en $x^* = (1 \ 3)$.

Problema C.9. *Branin*

$$\underset{x \in \mathbb{R}^2}{\text{minimizar}} \quad f(x) = a(x_2 - bx_1^2 + cx_1 - r)^2 + s(1 - t) \cos(x_1) + s$$

$$\text{sujeto a} \quad \begin{array}{l} -5 \leq x_1 \leq 10 \\ 0 \leq x_2 \leq 15 \end{array}$$

donde $a=1$, $b = \frac{5.1}{4\pi^2}$, $c = \frac{5}{\pi}$, $r=6$, $s=10$ y $t = \frac{1}{8\pi}$.

Mínimo global: $f(x^*) = 0.397887$ en $x^* = (-\pi, 12.275)$ y $x^* = (\pi \ 2.275)$.

Problema C.10. *Colville*

$$\underset{x \in \mathbb{R}^4}{\text{minimizar}} \quad f(x) = 100(x_1^2 - x_2)^2 + (x_1 - 1) + (x_3 - 1)^2 + 90(x_3^2 - x_4)^2$$

$$+ 10.1((x_2 - 1)^2 + (x_4 - 1)) + 19.8(x_2 - 1)(x_4 - 1)$$

$$\text{sujeto a} \quad -10 \leq x_i \leq 10, i = 1, \dots, 4.$$

Mínimo global: $f(x^*) = 0$ en $x^* = (1 \ 1 \ 1 \ 1)$.

Problema C.11. *Dixon & Price₂*

$$\underset{x \in \mathbb{R}^d}{\text{minimizar}} \quad f(x) = (x_1 - 1)^2 + \sum_{i=2}^d i(2x_i^2 - x_{i-1})^2$$

$$\text{sujeto a} \quad -10 \leq x_i \leq 10, i = 1, \dots, d.$$

con $d = 2$.

Mínimo global: $f(x^*) = 0$ en $x_i^* = 2^{-\frac{i-2}{2^i}}$ para $i = 1, \dots, d$.

Problema C.12. *Dixon & Price₅*

Idéntico al Problema C.11 con $d=5$.

Problema C.13. *Dixon & Price₁₀*

Idéntico al Problema C.11 con $d=10$.

Problema C.14. *Easom*

$$\begin{aligned} \underset{x \in \mathbb{R}^2}{\text{minimizar}} \quad & f(x) = -\cos(x_1)\cos(x_2)\exp(-1(x_1 - \pi)^2 - (x_2 - \pi)^2) \\ \text{sujeeto a} \quad & -100 \leq x_i \leq 100, i = 1, 2. \end{aligned}$$

Mínimo global: $f(x^) = -1$ en $x^* = (\pi \ \pi)$.*

Problema C.15. *Goldstein & Price*

$$\begin{aligned} \underset{x \in \mathbb{R}^2}{\text{minimizar}} \quad & f(x) = [1 + (x_1 + x_2 + 1)^2(19 - 14x_1 + 3x_1^2 - 14x_2 + 6x_1x_2 + 3x_2^2)] \\ & \times [30 + (2x_1 - 3x_2)^2(18 - 32x_1 + 12x_1^2 + 48x_2 - 36x_1x_2 + 27x_2^2)] \\ \text{sujeeto a} \quad & -2 \leq x_i \leq 2, i = 1, 2. \end{aligned}$$

Mínimo global: $f(x^) = 3$ en $x^* = (0 \ -1)$.*

Problema C.16. *Griewank*

$$\begin{aligned} \underset{x \in \mathbb{R}^2}{\text{minimizar}} \quad & f(x) = \sum_{i=1}^2 \frac{x_i^2}{4000} - \prod_{i=1}^2 \cos\left(\frac{x_i}{\sqrt{i}}\right) + 1 \\ \text{sujeeto a} \quad & -600 \leq x_i \leq 700, i = 1, 2. \end{aligned}$$

Mínimo global: $f(x^) = 0$ en $x^* = (0 \ 0)$.*

Problema C.17. *Hartman₃*

$$\begin{aligned} \underset{x \in \mathbb{R}^3}{\text{minimizar}} \quad & f(x) = -\sum_{i=1}^4 \alpha_i \exp\left(\sum_{j=1}^3 A_{ij}(x_j - P_{ij})^2\right) \\ \text{sujeeto a} \quad & 0 \leq x_i \leq 1, i = 1, 2, 3. \end{aligned}$$

$$\text{con } \alpha = (1 \ 1.2 \ 3 \ 3.2), A = \begin{pmatrix} 3 & 10 & 30 \\ 0.1 & 10 & 35 \\ 3 & 10 & 30 \\ 0.1 & 10 & 35 \end{pmatrix} \text{ y } P = 10^{-4} \begin{pmatrix} 3689 & 1170 & 2673 \\ 4699 & 4387 & 7470 \\ 1091 & 8732 & 5547 \\ 381 & 5743 & 8828 \end{pmatrix}.$$

Mínimo global: $f(x^) = -3.86278$ en $x^* = (0.114611 \ 0.555649 \ 0.852547)$.*

Problema C.18. *Hartman*₆

$$\begin{aligned} \underset{x \in \mathbb{R}^6}{\text{minimizar}} \quad & f(x) = - \sum_{i=1}^4 \alpha_i \exp \left(\sum_{j=1}^6 A_{ij} (x_j - P_{ij})^2 \right) \\ \text{sujeto a} \quad & 0 \leq x_i \leq 1, i = 1, \dots, 6. \end{aligned}$$

con

$$\alpha = (1 \quad 1.2 \quad 3 \quad 3.2), \quad A = \begin{pmatrix} 10 & 3 & 17 & 3.5 & 1.7 & 8 \\ 0.05 & 10 & 17 & 0.1 & 8 & 14 \\ 3 & 3.5 & 1.7 & 10 & 17 & 8 \\ 17 & 8 & 0.05 & 10 & 0.1 & 14 \end{pmatrix},$$

$$\text{y } P = 10^{-4} \begin{pmatrix} 1312 & 1696 & 5569 & 124 & 8283 & 5886 \\ 2329 & 4135 & 8307 & 3736 & 1004 & 9991 \\ 2348 & 1451 & 3522 & 2883 & 3047 & 6650 \\ 4047 & 8828 & 8732 & 5743 & 1091 & 381 \end{pmatrix}.$$

Mínimo global: $f(x^*) = -3.32237$ en
 $x^* = (0.20169 \quad 0.150011 \quad 0.476874 \quad 0.275332 \quad 0.311652 \quad 0.6573)$.

Problema C.19. *Hump*

$$\begin{aligned} \underset{x \in \mathbb{R}^2}{\text{minimizar}} \quad & f(x) = 2x_1^2 - 1.05x_1^4 + \frac{x_1^6}{6} + x_1x_2 + x_2^2 \\ \text{sujeto a} \quad & -5 \leq x_i \leq 5, i = 1, 2. \end{aligned}$$

Mínimo global: $f(x^*) = 0$ en $x^* = (0 \quad 0)$.

Problema C.20. *Levy*₂

$$\begin{aligned} \underset{x \in \mathbb{R}^d}{\text{minimizar}} \quad & f(x) = \sin^2(\pi\omega_1) \\ & + \sum_{i=1}^{d-1} (\omega_i - 1)^2 [1 + 10 \sin^2(\pi\omega_i + 1)] + (\omega_d - 1)^2 [1 + \sin^2(2\pi\omega_d)] \\ \text{sujeto a} \quad & -10 \leq x_i \leq 10, i = 1, \dots, d. \end{aligned}$$

con $d=2$ y $\omega_i = 1 + \frac{x_i - 1}{4}$ para $i = 1, \dots, d$.

Mínimo global: $f(x^*) = 0$ en $x^* = (1 \quad \dots \quad 1)$.

Problema C.21. *Levy*₅

Idéntico al Problema C.20 con $d=5$.

Problema C.22. *Levy*₁₀

Idéntico al Problema C.20 con $d=10$.

Problema C.23. *Matyas*

$$\begin{aligned} \underset{x \in \mathbb{R}^2}{\text{minimizar}} \quad & f(x) = 0.26(x_1^2 + x_2^2) - 0.48x_1x_2 \\ \text{sujeeto a} \quad & -10 \leq x_i \leq 15, i = 1, 2. \end{aligned}$$

Mínimo global: $f(x^*) = 0$ en $x^* = (0 \ 0)$.

Problema C.24. *Michalewics₂*

$$\begin{aligned} \underset{x \in \mathbb{R}^d}{\text{minimizar}} \quad & f(x) = - \sum_{i=1}^d \sin(x_i) \sin^{2m} \left(\frac{ix_i^2}{\pi} \right) \\ \text{sujeeto a} \quad & 0 \leq x_i \leq \pi, i = 1, \dots, d. \end{aligned}$$

con $d=2$ y $m=10$.

Mínimo global: $f(x^*) = -1.8013$ en $x^* = (2.20 \ 1.57)$.

Problema C.25. *Michalewics₅*

Idéntico al Problema C.24 con $d=5$.

Mínimo global: $f(x^*) = -4.687658$.

Problema C.26. *Michalewics₁₀*

Idéntico al Problema C.24 con $d=10$.

Mínimo global: $f(x^*) = -9.66015$.

Problema C.27. *Perm*

$$\begin{aligned} \underset{x \in \mathbb{R}^4}{\text{minimizar}} \quad & f(x) = \sum_{i=1}^d \left(\sum_{j=1}^d (j^i + \beta) \left(\left(\frac{x_j}{j} \right)^i - 1 \right) \right)^2 \\ \text{sujeeto a} \quad & -4 \leq x_i \leq 4, i = 1, \dots, 4. \end{aligned}$$

con $\beta = 0.5$.

Mínimo global: $f(x^*) = 0$ en $x^* = (1 \ 2 \ 3 \ 4)$.

Problema C.28. *Powell₄*

$$\begin{aligned} \underset{x \in \mathbb{R}^d}{\text{minimizar}} \quad & f(x) = \sum_{i=1}^{\frac{d}{4}} [(x_{4i-3} + 10x_{4i-2})^2 + 5(x_{4i-1} - x_{4i})^2 \\ & + (x_{4i-2} - 2x_{4i-1})^4 + 10(x_{4i-3} - x_{4i})^4] \\ \text{sujeeto a} \quad & -4 \leq x_i \leq 5, i = 1, \dots, d. \end{aligned}$$

con $d=4$.

Mínimo global: $f(x^*) = 0$ en $x^* = (0 \ \dots \ 0)$.

Problema C.29. *Powell₈*

Idéntico al Problema C.28 con $d=8$.

Problema C.30. *Power Sum*

$$\begin{aligned} \underset{x \in \mathbb{R}^4}{\text{minimizar}} \quad & f(x) = \sum_{i=1}^4 \left[\left(\sum_{j=1}^4 x_j^i \right) - b_i \right]^2 \\ \text{sujeto a} \quad & 0 \leq x_i \leq 4, i = 1, \dots, 4. \end{aligned}$$

con $b = (8 \ 18 \ 44 \ 114)$.

Mínimo global: $f(x^*) = 0$.

Problema C.31. *Rastrigin₂*

$$\begin{aligned} \underset{x \in \mathbb{R}^d}{\text{minimizar}} \quad & f(x) = 10d + \sum_{i=1}^d [x_i^2 - 10 \cos(2\pi x_i)] \\ \text{sujeto a} \quad & -5.12 \leq x_i \leq 6.12, i = 1, \dots, d. \end{aligned}$$

con $d=2$.

Mínimo global: $f(x^*) = 0$ en $x^* = (0 \ \dots \ 0)$.

Problema C.32. *Rastrigin₅*

Idéntico al Problema C.31 con $d=5$.

Problema C.33. *Rastrigin₁₀*

Idéntico al Problema C.31 con $d=10$.

Problema C.34. *Rosenbrock₂*

$$\begin{aligned} \underset{x \in \mathbb{R}^d}{\text{minimizar}} \quad & f(x) = \sum_{i=1}^{d-1} [100(x_{i+1} - x_i^2)^2 + (x_i - 1)^2] \\ \text{sujeto a} \quad & -5 \leq x_i \leq 10, i = 1, \dots, d. \end{aligned}$$

con $d=2$.

Mínimo global: $f(x^*) = 0$ en $x^* = (1 \ \dots \ 1)$.

Problema C.35. *Rosenbrock₅*

Idéntico al Problema C.34 con $d=5$.

Problema C.36. *Rosenbrock₁₀*

Idéntico al Problema C.34 con $d=10$.

Problema C.37. *Schwefel₂*

$$\begin{aligned} \underset{x \in \mathbb{R}^d}{\text{minimizar}} \quad & f(x) = 418.9829d - \sum_{i=1}^d x_i \sin(\sqrt{|x_i|}) \\ \text{sujeto a} \quad & -500 \leq x_i \leq 500, i = 1, \dots, d. \end{aligned}$$

con $d=2$.

Mínimo global: $f(x^*) = 0$ en $x^* = (420.9687 \dots 420.9687)$.

Problema C.38. *Schwefel₅*

Idéntico al Problema C.37 con $d=5$.

Problema C.39. *Schwefel₁₀*

Idéntico al Problema C.37 con $d=10$.

Problema C.40. *Shekel₅*

$$\begin{aligned} \underset{x \in \mathbb{R}^4}{\text{minimizar}} \quad & f(x) = - \sum_{i=1}^m \left(\sum_{j=1}^m (x_j - C_{ji})^2 + \beta_i \right)^{-1} \\ \text{sujeto a} \quad & 0 \leq x_i \leq 10, i = 1, \dots, 4. \end{aligned}$$

con

$$m=5, \beta = \frac{1}{10} (1 \ 2 \ 2 \ 4 \ 4 \ 6 \ 3 \ 7 \ 5 \ 5) \quad y \quad A = \begin{pmatrix} 4 & 1 & 8 & 6 & 3 & 2 & 5 & 8 & 6 & 7 \\ 4 & 1 & 8 & 6 & 7 & 9 & 3 & 1 & 2 & 3.6 \\ 4 & 1 & 8 & 6 & 3 & 2 & 5 & 8 & 6 & 7 \\ 4 & 1 & 8 & 6 & 7 & 9 & 3 & 1 & 2 & 3.6 \end{pmatrix}.$$

Mínimo global: $f(x^*) = -10.15320$ en $x^* = (4 \ 4 \ 4 \ 4)$.

Problema C.41. *Shekel₇*

Idéntico al Problema C.40 con $m=7$.

Mínimo global: $f(x^*) = -10.4029$ en $x^* = (4 \ 4 \ 4 \ 4)$.

Problema C.42. *Shekel₁₀*

Idéntico al Problema C.40 con $m=10$.

Mínimo global: $f(x^*) = -10.5364$ en $x^* = (4 \ 4 \ 4 \ 4)$.

Problema C.43. *Shubert*

$$\begin{aligned} \underset{x \in \mathbb{R}^2}{\text{minimizar}} \quad & f(x) = \left(\sum_{i=1}^5 i \cos((i+1)x_1 + i) \right) \left(\sum_{i=1}^5 i \cos((i+1)x_2 + i) \right) \\ \text{sujeto a} \quad & -10 \leq x_i \leq 10, i = 1, 2. \end{aligned}$$

Mínimo global: $f(x^*) = -186.7309$.

Problema C.44. *Sphere₂*

$$\begin{aligned} \underset{x \in \mathbb{R}^d}{\text{minimizar}} \quad & f(x) = \sum_{i=1}^d x_i^2 \\ \text{sujeto a} \quad & -5.12 \leq x_i \leq 6.12, i = 1, \dots, d. \end{aligned}$$

con $d = 2$.

Mínimo global: $f(x^*) = 0$ en $x^* = (0 \ \dots \ 0)$.

Problema C.45. *Sphere₅*

Idéntico al Problema C.44 con $d=5$.

Problema C.46. *Sphere₁₀*

Idéntico al Problema C.44 con $d=10$.

Problema C.47. *Sum squares₂*

$$\begin{aligned} \underset{x \in \mathbb{R}^d}{\text{minimizar}} \quad & f(x) = \sum_{i=1}^d ix_i^2 \\ \text{sujeto a} \quad & -10 \leq x_i \leq 15, i = 1, \dots, d. \end{aligned}$$

con $d = 2$.

Mínimo global: $f(x^*) = 0$ en $x^* = (0 \ \dots \ 0)$.

Problema C.48. *Sum squares₅*

Idéntico al Problema C.47 con $d=5$.

Problema C.49. *Sum squares₁₀*

Idéntico al Problema C.47 con $d=10$.

Problema C.50. *Trid₆*

$$\begin{aligned} \underset{x \in \mathbb{R}^d}{\text{minimizar}} \quad & f(x) = \sum_{i=1}^d (x_i - 1)^2 - \sum_{i=2}^d x_i x_{i-1} \\ \text{sujeto a} \quad & -d^2 \leq x_i \leq d^2, i = 1, \dots, d. \end{aligned}$$

con $d = 2$.

Mínimo global: $f(x^*) = \frac{-d(d+4)(d+1)}{6}$ en $x_i^* = i(d+1-i)$, $i = 1, \dots, d$.

Problema C.51. *Trid*₁₀

Idéntico al Problema C.50 con $d=10$.

Problema C.52. *Zakharov*₂

$$\begin{aligned} \underset{x \in \mathbb{R}^d}{\text{minimizar}} \quad & f(x) = \sum_{i=1}^d x_i^2 + \left(\sum_{i=1}^d 0.5ix_i \right)^2 + \left(\sum_{i=1}^d 0.5ix_i \right)^4 \\ \text{sujeto a} \quad & -5 \leq x_i \leq 11, i = 1, \dots, d. \end{aligned}$$

con $d = 2$.

Mínimo global: $f(x^) = 0$ en $x_i^* = (0 \ \dots \ 0)$.*

Problema C.53. *Zakharov*₅

Idéntico al Problema C.52 con $d=5$.

Problema C.54. *Zakharov*₁₀

Idéntico al Problema C.52 con $d=10$.

Bibliografía

- [1] ALBERTO, P., NOGUEIRA, F., ROCHA, H., AND VICENTE, L. N. Pattern search methods for user-provided points: Application to molecular geometry problems. *SIAM Journal on Optimization* 14, 4 (2004), 1216–1236.
- [2] ARMIJO, L. Minimization of functions having lipschitz continuous first partial derivatives. *Pacific Journal of mathematics* 16, 1 (1966), 1–3.
- [3] AROUXÉT, M., ECHEBEST, N., PILOTTA, E. A., ET AL. Active-set strategy in powell’s method for optimization without derivatives. *Computational & Applied Mathematics* 30, 1 (2011), 171–196.
- [4] AUDET, C., AND DENNIS JR, J. E. Analysis of generalized pattern searches. *SIAM Journal on optimization* 13, 3 (2002), 889–903.
- [5] AUDET, C., AND HARE, W. Derivative-free and blackbox optimization.
- [6] BIRGIN, E. G., MARTÍNEZ, J. M., AND RAYDAN, M. Inexact spectral projected gradient methods on convex sets. *IMA Journal of Numerical Analysis* 23, 4 (2003), 539–559.
- [7] CHENG, W., AND LI, D.-H. A derivative-free nonmonotone line search and its application to the spectral residual method. *IMA Journal of Numerical Analysis* 29, 3 (2009), 814–825.
- [8] CONN, A. R., GOULD, N. I., AND TOINT, P. L. *Trust region methods*. SIAM, 2000.
- [9] CONN, A. R., SCHEINBERG, K., AND VICENTE, L. N. *Introduction to derivative-free optimization*. SIAM, 2009.
- [10] DINIZ-EHRHARDT, M., FERREIRA, D., AND SANTOS, S. A. A pattern search and implicit filtering algorithm for solving linearly constrained minimization problems with noisy objective functions. *Optimization Methods and Software* 34, 4 (2019), 827–852.
- [11] DINIZ-EHRHARDT, M., MARTÍNEZ, J., AND RAYDÁN, M. A derivative-free nonmonotone line-search technique for unconstrained optimization. *Journal of computational and applied mathematics* 219, 2 (2008), 383–397.
- [12] DOLAN, E. D., AND MORÉ, J. J. Benchmarking optimization software with performance profiles. *Mathematical programming* 91, 2 (2002), 201–213.
- [13] FLOUDAS, C. A., AND PARDALOS, P. M. *Encyclopedia of optimization*. Springer Science & Business Media, 2008.
- [14] FRAU, J. A., AND PILOTTA, E. Nonmonotone coordinate search method for bound constrained optimization. *Malaysian journal of mathematical sciences* 14, 3 (2020), 451–471.

-
- [15] GARCÍA-PALOMARES, U. M., AND RODRÍGUEZ, J. F. New sequential and parallel derivative-free algorithms for unconstrained minimization. *SIAM Journal on optimization* 13, 1 (2002), 79–96.
- [16] GARDEUX, V., OMRAN, M. G., CHELOUAH, R., SIARRY, P., AND GLOVER, F. Adaptive pattern search for large-scale optimization. *Applied Intelligence* 47, 2 (2017), 319–330.
- [17] GRIPPO, L., LAMPARIELLO, F., AND LUCIDI, S. A nonmonotone line search technique for newton’s method. *SIAM Journal on Numerical Analysis* 23, 4 (1986), 707–716.
- [18] GRIVA, I., NASH, S. G., AND SOFER, A. *Linear and nonlinear optimization*, vol. 108. Siam, 2009.
- [19] HANSEN, E., AND WALSTER, G. W. *Global optimization using interval analysis: revised and expanded*, vol. 264. CRC Press, 2003.
- [20] HOCK, W., AND SCHITTKOWSKI, K. Test examples for nonlinear programming codes. *Journal of optimization theory and applications* 30, 1 (1980), 127–129.
- [21] HOOKE, R., AND JEEVES, T. A. “direct search” solution of numerical and statistical problems. *Journal of the ACM (JACM)* 8, 2 (1961), 212–229.
- [22] HORST, R., PARDALOS, P. M., AND VAN THOAI, N. *Introduction to global optimization*. Springer Science & Business Media, 2000.
- [23] JONES, D. R. *Direct global optimization algorithm*. Springer US, Boston, MA, 2001, pp. 431–440.
- [24] JONES, D. R., PERTTUNEN, C. D., AND STUCKMAN, B. E. Lipschitzian optimization without the lipschitz constant. *Journal of optimization Theory and Applications* 79, 1 (1993), 157–181.
- [25] KOLDA, T. G., LEWIS, R. M., AND TORCZON, V. Optimization by direct search: New perspectives on some classical and modern methods. *SIAM review* 45, 3 (2003), 385–482.
- [26] KOLDA, T. G., LEWIS, R. M., AND TORCZON, V. Stationarity results for generating set search for linearly constrained optimization. *SIAM Journal on Optimization* 17, 4 (2007), 943–968.
- [27] KREJIĆ, N., LUŽANIN, Z., NIKOLOVSKI, F., AND STOJKOVSKA, I. A nonmonotone line search method for noisy minimization. *Optimization Letters* 9, 7 (2015), 1371–1391.
- [28] LA CRUZ, W., MARTÍNEZ, J., AND RAYDAN, M. Spectral residual method without gradient information for solving large-scale nonlinear systems of equations. *Mathematics of Computation* 75, 255 (2006), 1429–1448.
- [29] LEE, K., KIM, M., HA, M. Y., AND MIN, J. K. Investigation of heat-exchanger-sizing methods using genetic, pattern search, and simulated annealing algorithms and the effect of entropy generation. *Journal of Mechanical Science and Technology* 32, 2 (2018), 915–928.
- [30] LEWIS, R. M., SHEPHERD, A., AND TORCZON, V. Implementing generating set search methods for linearly constrained minimization. *SIAM Journal on Scientific Computing* 29, 6 (2007), 2507–2530.
- [31] LEWIS, R. M., AND TORCZON, V. Pattern search algorithms for bound constrained minimization. *SIAM Journal on optimization* 9, 4 (1999), 1082–1099.

-
- [32] LEWIS, R. M., AND TORCZON, V. Pattern search methods for linearly constrained minimization. *SIAM Journal on Optimization* 10, 3 (2000), 917–941.
- [33] LI, D.-H., AND FUKUSHIMA, M. A derivative-free line search and global convergence of broyden-like method for nonlinear equations. *Optimization Methods and Software* 13, 3 (2000), 181–201.
- [34] LIBERTI, L., AND MACULAN, N. *Global optimization: from theory to implementation*, vol. 84. Springer Science & Business Media, 2006.
- [35] LIUZZI, G., LUCIDI, S., AND PICCIALI, V. A direct-based approach exploiting local minimizations for the solution of large-scale global optimization problems. *Computational Optimization and Applications* 45, 2 (2010), 353–375.
- [36] LUCIDI, S., AND SCIANDRONE, M. On the global convergence of derivative-free methods for unconstrained optimization. *SIAM Journal on Optimization* 13, 1 (2002), 97–116.
- [37] MARTÍNEZ, J. M., AND SANTOS, S. A. Métodos computacionais de otimização. *Colóquio Brasileiro de Matemática, Apostilas* 20 (1995).
- [38] MEZA, J. C., AND MARTINEZ, M. L. Direct search methods for the molecular conformation problem. *Journal of Computational Chemistry* 15, 6 (1994), 627–632.
- [39] MORÉ, J. J., AND WILD, S. M. Benchmarking derivative-free optimization algorithms. *SIAM Journal on Optimization* 20, 1 (2009), 172–191.
- [40] NIKOLOVSKI, F., AND STOJKOVSKA, I. New derivative-free nonmonotone line search methods for unconstrained minimization. In *Proceedings of the Fifth International Scientific Conference-FMNS2013*, vol. 1, pp. 47–53.
- [41] NOCEDAL, J., AND WRIGHT, S. *Numerical optimization*. Springer Science & Business Media, 2006.
- [42] OEUVRAY, R., AND BIERLAIRE, M. A new derivative-free algorithm for the medical image registration problem. *International Journal of Modelling and Simulation* 27, 2 (2007), 115–124.
- [43] PAULAVIČIUS, R., CHITER, L., AND ŽILINSKAS, J. Global optimization based on bisection of rectangles, function values at diagonals, and a set of lipschitz constants. *Journal of Global Optimization* 71, 1 (2018), 5–20.
- [44] RASHID, H., SIAM, F. M., MAAN, N., AND ABD RAHMAN, W. N. W. Parameter estimation for a model of ionizing radiation effects on targeted cells using genetic algorithm and pattern search method. *MATEMATIKA: Malaysian Journal of Industrial and Applied Mathematics* 34, 3 (2018), 1–13.
- [45] SHUBERT, B. O. A sequential method seeking the global maximum of a function. *SIAM Journal on Numerical Analysis* 9, 3 (1972), 379–388.
- [46] TORCZON, V. On the convergence of the multidirectional search algorithm. *SIAM journal on Optimization* 1, 1 (1991), 123–145.
- [47] TORCZON, V. On the convergence of pattern search algorithms. *SIAM Journal on optimization* 7, 1 (1997), 1–25.

-
- [48] ULBRICH, M. Nonmonotone trust-region methods for bound-constrained semismooth equations with applications to nonlinear mixed complementarity problems. *SIAM Journal on Optimization* 11, 4 (2001), 889–917.
- [49] W., Y. Positive basis and a class of direct search techniques. *Scientia Sinica (in Chinese)* 9, S1 (1979), 53–67.
- [50] WRIGHT, M. Direct search methods: Once scorned, now respectable. In *Numerical Analysis: Proceedings of the 1995 Dundee Biennial Conference in Numerical Analysis* (1996), Addison-Wesley, pp. 191–208.
- [51] YU, Z., AND PU, D. A new nonmonotone line search technique for unconstrained optimization. *Journal of Computational and Applied Mathematics* 219, 1 (2008), 134–144.
- [52] ZHANG, H., AND HAGER, W. W. A nonmonotone line search technique and its application to unconstrained optimization. *SIAM journal on Optimization* 14, 4 (2004), 1043–1056.