Universidad Nacional Autónoma de México

Programa de Maestría y Doctorado en Ciencias Químicas

Método deterministico de optimización global de funciones de energía potencial mediante análisis de intervalos

T E S I S

Para optar por el grado de

Maestro en Ciencias

Presenta

Químico Bryan Alexander Corzo Parra

Asesor

Dr. David Philip Sanders

Departamento de Física

Facultad de Ciencias

Ciudad de México, Septiembre 2020

UNAM
POSGRADO
Ciencias Químicas

Universidad Nacional Autónoma de México

Programa de Maestría y Doctorado en Ciencias Químicas

Método deterministico de optimización global de funciones de energía potencial mediante análisis de intervalos

T E S I S

Para optar por el grado de

Maestro en Ciencias

Presenta

Químico Bryan Alexander Corzo Parra

Posgrado
Ciencias
Químicas

Ciudad de México, Septiembre 2020

Agradecimientos

Estoy profundamente agradecido con la Universidad Nacional Autónoma de México al brindarme la oportunidad de poder crecer como persona y profesional, por el gran y excelente cuerpo académico con el que cuenta, que con sus esfuerzos ofrecen las herramientas para construir un futuro y con todas aquellas personas que hasta ahora me han apoyado a terminar este trabajo y así culminar un paso más en mi camino profesional.

Agradezco en gran medida al Dr. David P. Sanders no sólo por la dirección de este proyecto, sino por su comprensión, apoyo, paciencia y ayuda que me ha brindado durante todo este tiempo; demostrándome que aparte de ser un gran científico es un increíble ser humano.

De igual manera agradezco a mis sinodales, El Dr. Ilyan Kaplan, el Dr. José Barquera, el Dr. Jorge Garza, el Dr. Igancio Garzón y el Dr. Jorge Martín del Campo, por sus valiosos comentarios sobre este trabajo. Asimismo, agradezco al Posgrado de Ciencias Químicas, en particular a su secretaria técnica Josefina Tenopala, quienes siempre estuvieron pendientes ante cualquier problema o inquietud que presenté.

Agradezco a los pocos, pero grandes amigos que conocí en este proceso, a mis amigos mexicanos Ramses Gonzales y Ricardo Vite, con los que he aprendido a valorar la riqueza individual de las personas. A mis amigos colombianos Cristhian Paredes, Daniel Moncada y Nicolas Preciado quienes siempre me han apoyado desde la distancia y han estado para mí en los peores momentos de este proceso y con quienes he podido intercambiar horas y horas de experiencias, conocimientos y demás cosas que han hecho parte de lo que ahora soy.

Agradezco a mis tíos, primos y mi nueva familia mexicana, quienes siempre han estado al pendiente de este proceso y quienes me han hecho llegar sus mejores deseos.

Agradezco al soporte del CONACYT por la Beca Nacional otorgada a través del CVU N. 917885, con la que me fue posible realizar este proyecto.

Agradezco finalmente y de manera enfática a mi padre Manuel Corzo, mi madre Nancy Parra y mi hermana Diana Corzo, quienes siempre han estado apoyando mis sueños, quienes durante todo este proceso estuvieron a mi lado animándome a seguir y quienes han sido mi mayor estímulo para continuar. Agradezco de igual manera a la persona que en los últimos años se ha vuelto en la persona más importante de mi vida, a Karla Maldonado, que gracias a su apoyo incondicional me permitió levantarme y seguir adelante, en la que me he podido apoyar en los momentos de mayor debilidad y con la que he caminado y crecido como persona.

Introducción

Uno de los principales problemas de los cálculos numéricos basados en aritmética de punto flotante son los errores debido al redondeo. La acumulación de este tipo de errores puede generar resultados irrelevantes, los cuales resultan más que desastrosos, especialmente para sistemas críticos. En este punto, los métodos basados en el análisis de intervalos son la única medida existente para acotar de manera rigurosa, los pasos intermedios de un cálculo numérico.

La forma en que el análisis de intervalos calcula con conjuntos revela una atractiva vía para resolver problemas de optimización global. En general, los métodos de optimización global basados en intervalos utilizan un método de branch and bound (B&B), el cual divide de manera recursiva el espacio de búsqueda, y descarta aquellas regiones del espacio que no pueden contener la solución óptima mediante argumentos de refutación, por ejemplo si el límite inferior del rango de la función objetivo en un subconjunto es mayor que el límite superior del mínimo actual, se asegura numéricamente que el subconjunto no puede contener una solución óptima, garantizando de manera rigurosa la obtención del mínimo global de cualquier función continua.

Estos métodos a su vez pueden ser combinados con operaciones de contracción o refinamiento, con el fin de reducir los dominios de las variables, y/o con métodos de optimización local, con el fin de acelerar el proceso de búsqueda sin perder la rigurosidad del método global, sin embargo, la bisección recursiva en regiones no óptimas sigue siendo a veces inevitable, debido a la complejidad exponencial en el número de variables y los problemas de dependencia que esto conlleva, por lo que aún no se puede esperar resolver sistemas mayores a los que solo contengan unas pocas docenas de variables.

El objetivo de esta tesis es presentar un algoritmo de B&B completo, en el cual se unifiquen las distintas metodologías reportadas en la literatura, siendo aplicados a sistemas de prueba en el ámbito de la química computacional, generando un repositorio abierto [1] en el lenguaje de programación Julia [2], para la optimización global de funciones continuas de manera rigurosa, que a su vez complemente el paquete de IntervalArithmetic.jl [3].

Esta tesis intenta abordar 3 conceptos fundamentales los cuales se dividen en 6 capítulos. La primera parte está dedicada a la mecánica molecular, la cual incluye una breve descripción de los potenciales básicos que forman parte de un campo de fuerza. La segunda parte está dedicada a los métodos de optimización actuales, en la cual se incluye una breve descripción de los métodos más comunes usados en la optimización global de funciones. La tercera parte introduce los conceptos básicos de la aritmética de intervalos, incluyendo los distintos métodos de inclusión y refinamientos, además de presentar algunas aplicaciones en problemas de optimización global; los resultados más importantes son presentados en este punto. Finalmente, las conclusiones del trabajó son presentadas en el capítulo 6, resumiendo los resultados obtenidos y presentando algunas perspectivas para futuros trabajos.

ÍNDICE GENERAL

ÍNDICE DE FIGURAS

ÍNDICE DE TABLAS

CAPÍTULO 1

MECÁNICA MOLECULAR

En este capítulo se introduce el concepto de mecánica molecular (MM), la cual se basa en una visión de las moléculas como bolas unidas por resortes. La energía potencial de una molécula se puede escribir como la suma de términos que implican estiramientos de enlaces, flexión de ángulo, ángulos diedros e interacciones no enlazantes. Estos términos forman un campo de fuerza y calcular las constantes en el campo de fuerza constituye la parametrización del campo.

1.1. Introducción

Para poder comprender en detalle la estructura, la función y los mecanismos de reacción de sistemas biológicos y químicos es necesario emplear métodos teóricos de modelado molecular que complementen los resultados obtenidos mediante técnicas experimentales. A pesar de que en la actualidad los métodos instrumentales permiten estudiar las propiedades físicas y estructurales de las moléculas, no son capaces de resolver problemas más complejos como el movimiento de una proteína o su unión a un ligando.

En estos casos es necesario utilizar el **modelado molecular**, cuyo principal objetivo consiste en describir las interacciones moleculares en base a las leyes generales de la química y la física. El uso de estos métodos está estrechamente ligado al uso de computadoras, y aunque a priori se podría llegar a modelar con exactitud cualquier sistema biológico con el suficiente poder de cálculo y un alto nivel de teoría, esto no es siempre posible, debido a limitaciones tanto de tiempo como de recursos computacionales. Por este motivo, es necesario introducir aproximaciones para mantener la viabilidad del experimento aplicando distintos niveles de teoría en función al sistema que va a ser analizado.

De manera general, la descripción más rigurosa de cualquier sistema viene definida por la **mecánica cuántica** (*Quantum Mechanics*, QM), en la que se considera, de manera explícita, los electrones en sus cálculos, y se tienen en cuenta las propiedades que dependen

de la distribución electrónica y la reactividad química, como la formación y ruptura de enlaces. No obstante, su aplicabilidad está sujeta a sistemas con un número restringido de átomos debido a la falta de recursos computacionales. Por tal motivo, el uso de la **mecánica molecular o clásica** (*Molecular Mechanics*, MM), la cual ignora los movimientos electrónicos y calcula la energía de una molécula o conjunto de estas a través de la posición de los núcleos atómicos, genera un nivel teoría aplicable para estudiar macromoléculas como el DNA o las proteínas. Métodos híbridos resultan de la combinación de QM/MM donde una pequeña parte del sistema se estudia mediante QM, mientras el resto se considera mediante MM [4].

1.2. Perspectiva

Uno de los principales problemas a resolver en la química computacional, es la determinación de la estructura molecular o geometría de una molécula, la cual se encuentra estrechamente relacionada con las propiedades que está exhibe, por ejemplo la estructura nativa de las proteínas y por ende la naturaleza de su plegamiento, está estructuralmente relacionada con el mínimo global de su superficie de energía potencial, por lo que el problema consiste en encontrar el mínimo global de la función de energía potencial o superficie de energía potencial (*potential energy surface*, PES) [4, 5].

En general el problema resulta simple de plantear; no obstante, el éxito de los métodos computacionales para resolver este tipo de problema depende principalmente de dos factores; el primero es la selección adecuada de una función de energía potencial para predecir los estados basales[1] del sistema, y segundo los algoritmos de minimización disponibles que se puedan utilizar para encontrar de manera eficaz el mínimo global de dicha función.

De manera general, la información de cualquier sistema se describe como la solución de la ecuación de Schrödinger [2](1.1)

$$\hat{\mathrm{H}}\ \Psi(\mathrm{r},\mathrm{R})\ =\ \mathrm{E}\ \Psi(\mathrm{r},\mathrm{R}), \tag{1.1}$$

donde $\Psi(\mathrm{r},\mathrm{R})$ es una función que depende tanto de las coordenadas electrónicas r, como nucleares R. El hamiltoniano $\hat{\mathrm{H}}$, tiene en cuenta tanto las interacciones electrónicas como

[1] Estado fundamental o de menor energía en el que un átomo o molécula se puede encontrar sin absorber ni emitir energía.

[2] La solución es la función de onda Ψ, la cual contiene toda la información acerca del sistema. El hamiltoniano, $\hat{\mathrm{H}}$, es un operador en el espacio de funciones y los valores propios de $\hat{\mathrm{H}}$ corresponden a los niveles de energía del sistema.

nucleares:

$$\hat{H} \;=\; T_N + T_e + V_{ee} + V_{NN} + V_{Ne}. \tag{1.2}$$

Donde T_N corresponde a la energía cinética de los núcleos, T_e corresponde a la energía cinética de los electrones, V_{ee} son las interacciones electrón-electrón, V_{NN} corresponden a las interacciones núcleo-núcleo y V_{Ne} serían las interacciones núcleo-electrón.

No obstante, esta descripción resulta irresoluble para sistemas compuestos de más de dos cuerpos [6]. Una aproximación que se puede considerar en el sistema, es desacoplar los movimientos nucleares y electrónicos de la ecuación 1.1 y estudiarlos de manera independiente; a esto se le conoce como *la aproximación de Born–Oppenheimer*, BO [7].

La aproximación de BO reconoce la gran diferencia entre la masa de los electrones y las masas de los núcleos atómicos, y en consecuencia las escalas de tiempo de sus movimientos. Dada la misma cantidad de energía cinética, los núcleos se mueven más lentamente que los electrones, generando que el núcleo experimente a los electrones como si estos fueran una nube de carga, mientras que los electrones sienten a los núcleos como si estos estuvieran estáticos [7]. Por lo anterior, la función de onda $\Psi(r; R)$, se puede expresar como la combinación lineal de una función electrónica $\Phi(r; R)$, que depende de las coordenadas de los electrones a una posición fija de los núcleos, y una función nuclear $\chi(R)$, que depende únicamente de las coordenadas de los núcleos (1.3).

De esta manera se puede expresar la ecuación de Schrödinger electrónica (1.4) y nuclear (1.5) como:

$$\Psi(r, R) \;=\; \Phi(r; R) \cdot \chi(R), \tag{1.3}$$

$$\hat{H}_{elec}\Phi(r; R) \;=\; E_{elec}\Phi(r; R), \tag{1.4}$$

$$(T_N + U(R))\chi(R) \;=\; E\chi(R), \tag{1.5}$$

donde $\hat{H}_{elec}$ corresponde al hamiltoniano electrónico[3], la parte electrónica (1.4) describe el movimiento de los electrones cuando los núcleos están fijos y la parte nuclear (1.5)

[3] $\hat{H}_{elec} = T_e + V_{ee} + V_{Ne}$

tiene presente el potencial efectivo (U(R)) al que están sujetos los núcleos. Este potencial efectivo en la base de la QM es conocido como la superficie de energía potencial o función de energía potencial de los núcleos y se define como:

$$U(R) \;=\; E_{elec} + V_{NN}. \tag{1.6}$$

En el caso de la mecánica molecular, solo se trabaja la parte nuclear y se basa en el modelo matemático de una molécula clásica como una colección de masas (esferas) correspondiente a los átomos, unidos mediante resortes que representan los enlaces (figura 1.1); el principio detrás de la MM es el expresar la energía de una molécula en función de su resistencia de estiramiento de enlaces, flexión de enlaces y apiñamiento de los átomos y usar la ecuación resultante para calcular las longitudes de enlace, ángulos de enlace y ángulos diedros correspondientes a los diversos mínimos de la superficie de energía potencial.

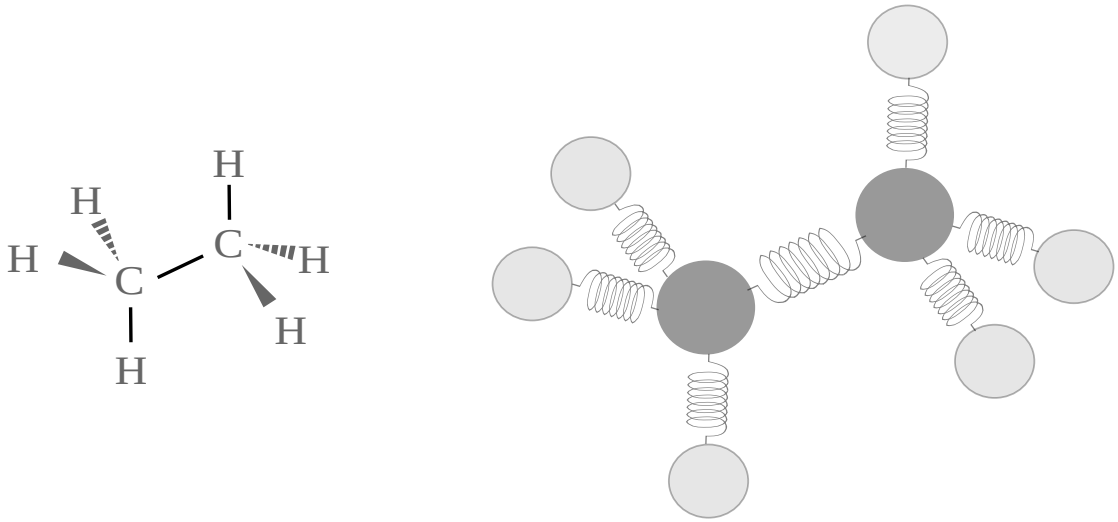

Figura 1.1: Representación del modelo utilizado en la mecánica molecular para el etano, donde los átomos son considerados esferas y los enlaces resortes

Las funciones de energía descritas bajo este razonamiento se conocen como **campos de fuerza** (*Force Fields*), en los que resulta importante conocer los mínimos y los puntos de silla, los cuales se consideran como barreras de mínima energía en los caminos que conectan los distintos mínimos y que corresponden a los estados de transición. Es importante aclarar que el método no hace referencia a los electrones y, por lo tanto, no puede arrojar luz sobre las propiedades electrónicas como la distribución de carga o el comportamiento nucleofílico y electrofílico de la molécula; asimismo, la MM utiliza de manera implícita la aproximación de BO, si los núcleos experimentan lo equivalente a una fuerza de atracción estática, ya sea de electrones o resorte, la molécula tendrá una geometría distinta [4, 5, 8].

Históricamente, la mecánica molecular comenzó como un intento de obtener información cuantitativa sobre las reacciones químicas en el momento donde la aplicabilidad de la mecánica cuántica en algo mucho más grande que el átomo de hidrógeno parecía remota. Los principios de las MM fueron formulados en 1946 por Westheimer y Meyer [9] y Hill [10]. En 1947 Westheimer publicó cálculos detallados en los que utilizó MM para estimar la energía de activación para la racemización de bifenilos. Los principales contribuyentes al desarrollo de la MM han sido Schleyer [11] y Allinger [12], donde esté ultimo ha desarrollado las series MM de programas[4] comenzando con MM1 y continuando con MM2, MM3 y MM4.

1.3. Campos de fuerza

Los primeros campos de fuerzas para simulaciones biomoleculares fueron desarrollados por primera vez en los años 70 [5, 8], y desde entonces un gran número de campos de fuerzas empíricos para MM han sido desarrollados para la simulación de proteínas, ácidos nucleicos, lípidos, entre otras moléculas biológicas. Algunos de los programas más destacados para hacer este tipo de simulaciones son **AMBER** [13], **CHARMM** [14], **GROMOS** [15] y **TINKER**[16]. En general, el campo de fuerza para una molécula puede ser escrito como

$$\mathrm{E} = \sum \mathrm{E}_{\text{enlace}} + \sum \mathrm{E}_{\text{ángulo}} + \sum \mathrm{E}_{\text{torsión}} + \sum \mathrm{E}_{\text{noenlace}}, \tag{1.7}$$

donde los primeros 3 términos son los términos enlazantes que incluyen las contribuciones debidas a los enlaces covalentes, ángulos de valencia y ángulos diedros (ángulos de torsión propios e impropios) respectivamente, mientras los términos no enlazantes se definen por un término de atracción–repulsión de tipo, principalmente, Lennard–Jones para las fuerzas de Van der Waals y un término Coulómbico para las interacciones electrostáticas [4, 5, 8, 17], entre otros; ver figura 1.2.

1.3.1. Términos enlazantes

- **Término de enlace**

  El término de enlace (*bond stretching*), se encarga de mantener las longitudes de enlace cercanas a los valores de equilibrio medidos experimentalmente. En términos

[4]Las series MM se utilizan principalmente para el modelado de geometrías y energías de moléculas biológicas.

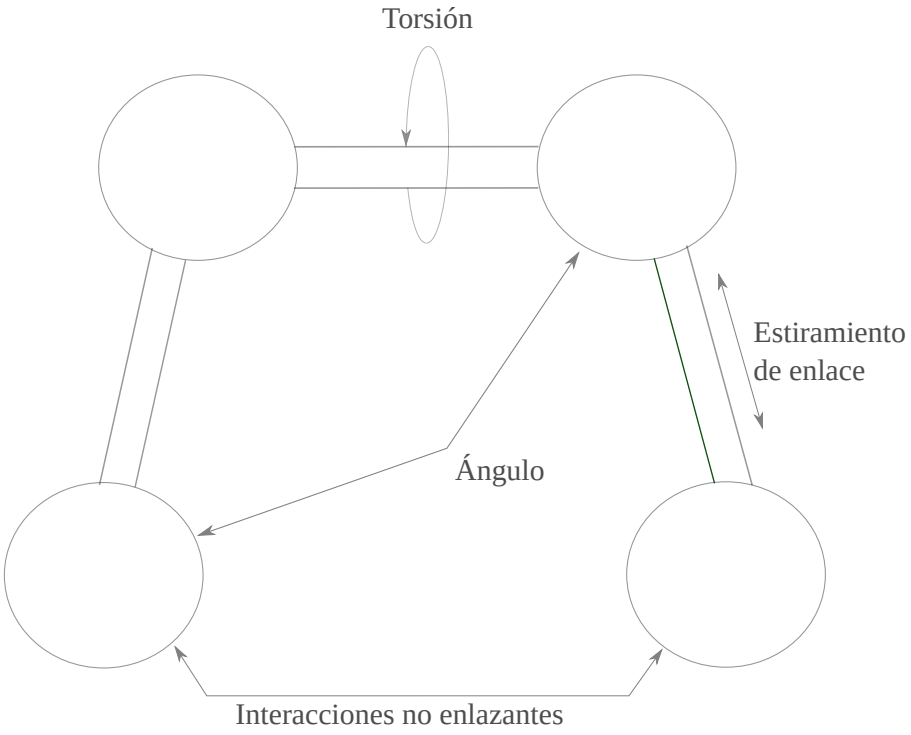

Figura 1.2: Esquema de interacciones enlazantes y no enlazantes tenidas en cuenta en el desarrollo de un campo de fuerzas

generales el potencial para un enlace se puede definir como un potencial de Morse (1.8), no obstante, en los campos de fuerza se suele emplear expresiones más simples como la ley de Hooke (1.9), ya que rara vez los enlaces se desvían significativamente de sus valores de equilibrio.

$$\mathrm{E_{enlace}} = \mathrm{D_e}\ (1 - \mathrm{e}^{-\mathrm{a(r-r_{eq})}})^2, \tag{1.8}$$

$$\mathrm{E_{enlace}} = \mathrm{K_{enlace}}(\mathrm{l} - \mathrm{l_{eq}})^2, \tag{1.9}$$

el termino enlace, corresponde a la constante de proporcionalidad, que en realidad es la mitad de la constante de fuerza del resorte o enlace. l_{eq} corresponde a la longitud de equilibrio o natural del enlace y l es la longitud de estiramiento. La figura 1.3 muestra la diferencia del comportamiento entre los potenciales de Morse y Hooke para un enlace.

- **Término de flexión del ángulo**

El término de flexión del ángulo (*angle bending*), correspondiente al ángulo formado

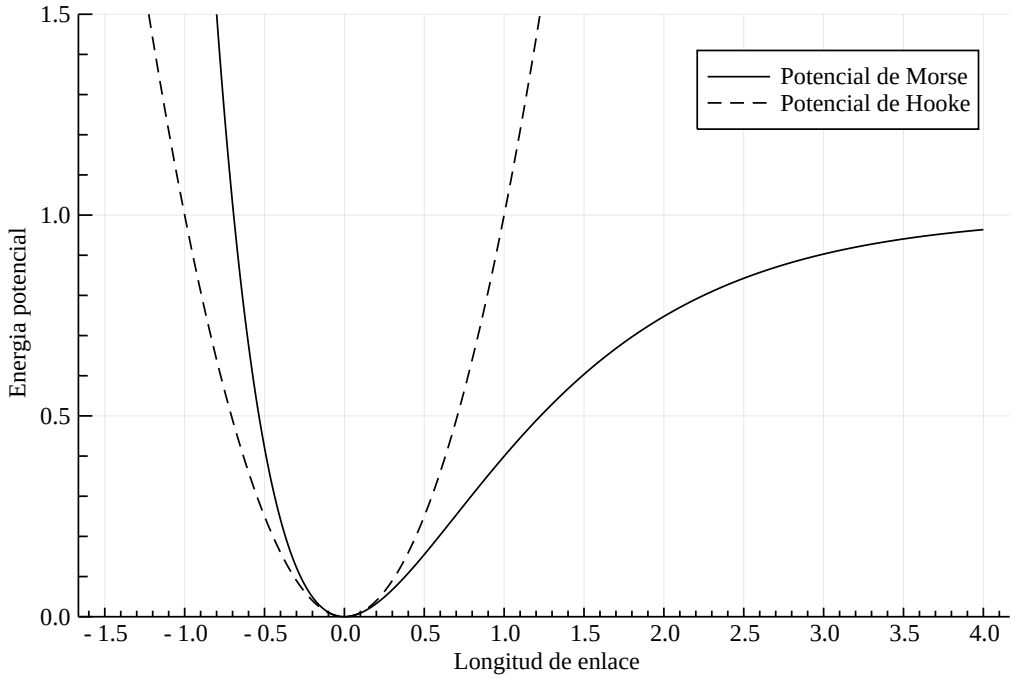

Figura 1.3: Potencial de Morse (línea continua) y potencial de Hooke (línea discontinua). Un cambio en la longitud de enlace o en el ángulo genera un cambio en la energía de la molécula. La energía para estos términos se puede representar de manera aproximada a una función cuadrática.

entre 3 átomos (A–B–C), se puede representar mediante un potencial armónico de Hooke, ecuación (1.10)

$$\mathrm{E_{ángulo}} = \mathrm{K_{flexión}}(\theta - \theta_{\mathrm{eq}})^2, \tag{1.10}$$

donde $\mathrm{K_{flexión}}$ es la constante de proporcionalidad, equivalente a $\frac{1}{2}$ de la fuerza de flexión del ángulo constante, θ_{eq}, corresponde al ángulo de equilibrio o natural y θ es el tamaño del ángulo distorsionado. Al ser de la misma forma que el potencial de la ecuación (1.9), su comportamiento es el mismo al presentado en la figura 1.3.

- **Término de torsión**

El término de torsión (*torsional term*), correspondiente al ángulo diedro conformado en una serie de cuatro átomos A–B–C–D, describe la variación de energía asociada a la rotación del enlace B–C (Figura 1.4). Este potencial se caracteriza por una periodicidad en el ángulo ϕ, si el enlace rota 360°, la energía deberá volver a su valor.

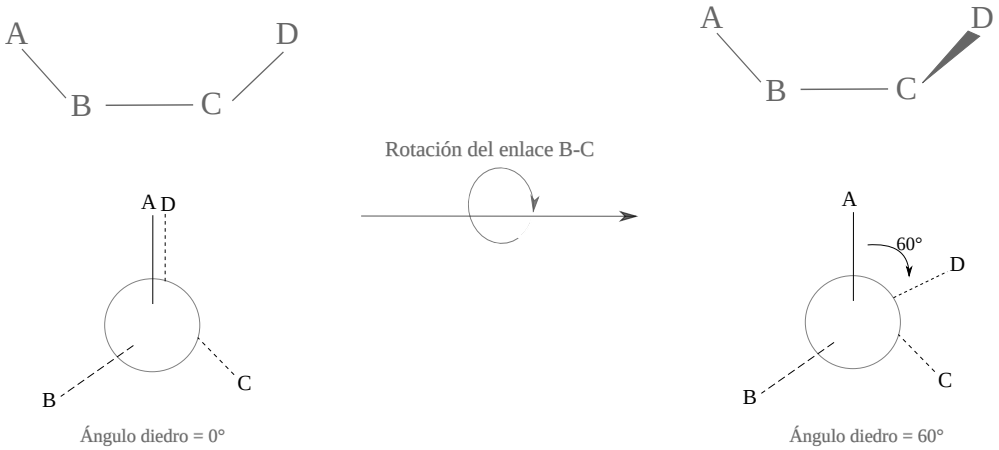

Figura 1.4: Cambio conformacional generado por la rotación del ángulo diedro. Cambia la geometría y la energía del sistema

Al repetirse la geometría cada 360°, la energía varía con el ángulo diedro en un patrón seno o coseno, y su perfil de energía se puede expresar como una serie de Fourier de la forma

$$\mathrm{E_{torsión}} = \frac{1}{2}\sum_{\mathrm{r=1}}^{\mathrm{N}} \mathrm{V_r}[1 + \cos(\mathrm{r}\phi - \gamma)], \tag{1.11}$$

donde la constante $\mathrm{V_r}$ determina la altura de la barrera de torsión alrededor del enlace B–C, N describe la multiplicidad, correspondiente al número de mínimos generado en una rotación de 360°[5], ϕ el ángulo de torsión y γ el ángulo de fase, indica en que punto pasa la torsión por el mínimo de energía.

1.3.2. Términos no enlazantes

La interacción entre dos átomos que no se unen es la causa principal del impedimento estérico, el cual desempeña un papel importante en la geometría molecular. Estas interacciones que no dependen de una relación específica de enlace entre átomos incluye los términos de interacciones de van der Waals, interacciones electrostáticas e interacciones debidas a inducción.

- **Interacciones de Van der Waals**

Las interacciones de van der Waals entre dos átomos se originan a partir de un balance entre fuerzas atractivas y repulsivas. Esta energía de interacción varía en

[5]Si N=1, describe una rotación que es periódica cada 360°, el término N=2 es periódico cada 180°, el término N=3 es periódico cada 120°, y así sucesivamente.

función de la distancia entre los átomos y generalmente es descrita por un potencial tipo Lennard–Jones (LJ):

$$E_{vdw} = \sum_{i<j} \left(\frac{A_{ij}}{r_{ij}^{12}} - \frac{B_{ij}}{r_{ij}^{6}} \right), \tag{1.12}$$

donde r_{ij} es la distancia entre dos átomos, A_{ij} y B_{ij} son constantes de van der Waals y la suma se hace entre todos los átomos presentes no enlazados. La expresión envuelve dos términos [18]:

- Una parte atractiva, proporcional a r_{ij}^{6}.
- Una parte repulsiva, el cual rápidamente crece cuando la distancia disminuye.

- **Interacciones electrostáticas**

  La distribución de carga en una molécula se puede representar en función de las cargas puntuales. Estas cargas reproducen las propiedades electrostáticas de la molécula, y en el caso de que las cargas estén centradas en los núcleos, se las denomina cargas atómicas parciales. La interacción electrostática de una molécula se calcula, por tanto, como la suma de las interacciones entre pares de cargas puntuales según la ley de Coulomb

$$E_{elec} = \sum_{i<j} \frac{q_i q_j}{4\pi\epsilon_0 r_{ij}}, \tag{1.13}$$

  donde q_A y q_B son las cargas puntuales de cada átomo, r_{AB} la distancia entre ellos y ϵ_0 la constante dieléctrica del medio que las separa. De este modo, si las cargas de dos átomos son contrarias, estos se atraerán, de lo contrario se repelan.

1.3.3. Parametrización

Como se ha evidenciado anteriormente los campos de fuerza contienen un gran número de parámetros, incluso si estos están diseñados para el modelado de sistemas pequeños. Un factor importante, por ende, es la fase de parametrización, en la que se busca encontrar el modelo capaz de aproximarse lo mejor posible a los valores experimentales.

De manera general, se utiliza un conjunto de moléculas con características similares (e.g. hidrocarburos lineales) denominados conjuntos de entrenamiento, en los que se busca

ajustar los parámetros en función a los datos estructurales, energéticos y electrónicos obtenidos de manera experimental, o asimismo, de los resultados obtenidos de los cálculos de alto nivel ab initio, DFT o combinaciones de ambos. Un ejemplo de esto puede ser revisado en [4, 17].

CAPÍTULO 2

PROBLEMA DEL CLÚSTER DE LENNARD–JONES

En este capítulo se presenta el problema de los clústeres de Lennard–Jones, los cuales corresponden a un problema de tipo NP-hard [19] dentro de la teoría de la complejidad computacional. De igual manera, se resumen algunos de los métodos de optimización no-lineales más comunes utilizado para abordar este tipo de problemas.

2.1. Introducción

Como se vio en el capítulo anterior, el potencial de van der Waals describe las interacciones atractivas y repulsivas entre un par de átomos no enlazados. Estas interacciones generalmente son descritas por un potencial de Lennard–Jones introducido en la ecuación (2.1), el cual juega un rol importante en los modelos energéticos empíricos, incluidos aquellos que modelan las funciones de energía en sistemas proteicos.

Un sistema que contenga más de un átomo y cuyas interacciones son únicamente de tipo van der Waals se le llama clúster de Lennard–Jones, un ejemplo de esto son los clústeres que consisten en átomos de gases nobles como Ne, Ar, Kr, Xe.

De manera general, un potencial de Lennard Jones se expresa como

$$\mathrm{V_{LJ}} \;=\; 4\,\epsilon\left[\left(\frac{\sigma}{\mathrm{r}}\right)^{12} - \left(\frac{\sigma}{\mathrm{r}}\right)^{6}\right], \tag{2.1}$$

donde ϵ es la profundidad del pozo de potencial, σ es la distancia finita en la que el potencial entre partículas es cero y r es la distancia entre partículas. Los valores reducidos de $\epsilon = 1$ y $\sigma = 1$, son convenientes para simulaciones de dinámica molecular, desde un punto numérico, las ventajas de las unidades sin dimensiones incluyen valores informáticos

que están más cerca de la unidad, utilizando ecuaciones simplificadas y siendo capaces de escalar fácilmente los resultados.

De manera general, la interacción entre un par de átomos neutros es una simple función unimodal, donde el mínimo global de energía potencial se encuentra cuando los átomos están a distancia de 1.12 unidades entre sí. Cuando esta distancia tiende a cero, el potencial tiende a infinito y cuando los átomos se alejan, el potencial tiende a cero. Sin embargo, en un sistema complejo, muchos átomos interaccionan y la suma se realiza entre cada par de átomos en el clúster, obteniendo como resultado una superficie de energía potencial compleja [20].

Resulta importante resaltar que en sistemas complejos la sumatoria se realiza en las condiciones de $i < j$ y no $i \neq j$, debido a que la interacción entre cada par de átomos necesita tenerse en cuenta solo una vez. Con lo anterior, se puede decir que el potencial de LJ es parcialmente separable [1].

El que la función sea parcialmente separable implica que, si se mueve un solo átomo en un grupo, la energía potencial puede ser reevaluada a un costo de $\frac{2}{N}$ del costo total de la evaluación, donde N es el número total de átomos del clúster. El problema entonces se resume a, dado un clúster de LJ conformado de N átomos, encontrar la posición relativa de los átomos en un espacio euclidiano tridimensional que represente la menor energía potencial, correspondiente a la geometría de equilibrio.

2.2. Planteamiento

Sea

$$x_i = (x_{1i},\ x_{2i},\ x_{3i}), \tag{2.2}$$

las coordenadas del átomo i en el espacio tridimensional euclidiano. Tendremos para N átomos en el clúster, el conjunto

$$X = (x_1,\ x_2,\ x_3, \ldots, x_N). \tag{2.3}$$

[1] Una función que es la suma de funciones, cada una de las cuales solo involucra un subconjunto disjunto de las variables, se llama **parcialmente separable**.

De esta manera se puede expresar el problema a resolver como

$$\min_{X \in R^{3N}} \quad V_{LJ} \;=\; \sum_{i=1}^{N-1} \sum_{j=i+1}^{N} \left[||x_i - x_j||^{-12} \;-\; ||x_i - x_j||^{-6} \right]. \tag{2.4}$$

Se ha observado que la determinación del mínimo global en los clústeres de Lennard Jones, en la teoría de la complejidad computacional corresponde a un problema **NP-hard** [19], esto es, problemas que se resuelven con algoritmos no deterministas en tiempo polinómico[2].

La principal dificultad en este tipo de problemas es el hecho que la función a minimizar no es convexa y presenta un gran número de mínimos locales. Emplear métodos probabilísticos o estocásticos, basados en métodos heurísticos de búsqueda, son la alternativa actual a la hora de resolver este tipo de funciones[21, 22], no obstante, en la teoría de la complejidad computacional, existe el consenso de que un problema no está *bien resuelto* hasta que se conozca un algoritmo determinista de tiempo polinomial que lo resuelva. Por tanto, nos referiremos a un problema como intratable, si es tan difícil que no existe algoritmo de tiempo polinomial capaz de resolverlo.

En la sección 2.3, ampliaremos sobre algunos de los principales métodos actuales de optimización no-lineal y en los capítulos 3 y 4 introduciremos los métodos de intervalos como una alternativa atractiva para la optimización global de funciones de manera garantizada y rigurosa.

2.3. Métodos de optimización no–lineales

Como se describió en la sección anterior, unos de los principales problemas para encontrar el mínimo global en clústeres de Lennard–Jones es la cantidad de mínimos locales presentes en una superficie no–convexa, siendo catalogado como un problema de tipo NP-hard.

De manera general, los algoritmos de optimización global se pueden clasificar en **algoritmos deterministicos**, en los que una misma entrada produce invariablemente una misma salida, y **algoritmos estocásticos**, en los que se incluyen variables pseudo-aleatorias que evolucionan en función a otra. Asimismo, dentro de estos modelos, se encuentran otras categorías como, los **métodos libres de derivadas o directos**, que se basan únicamente

[2]Un algoritmo muestra una complejidad polinómica si necesita un tiempo $O(n^k)$, donde n muestra la dimensión de entrada y k es una constante independiente de n.

en la función objetivo f, los **métodos de primer orden**, que se basan en la información del gradiente para ayudar a dirigir la búsqueda de un mínimo, y los **métodos de segundo orden**, en los que se usa la segunda derivada o la hessiana de la función para dirigir la búsqueda [21].

Métodos como *Simulated Annealing*, *métodos de deformación de la superficie* y *algoritmos genéticos* [18], son métodos de optimización global basados en la metaheurística, es decir, métodos que a diferencia de la heurística donde se explota información propia del problema para encontrar una solución '*suficientemente buena*' para un problema específico, estás son ideas algorítmicas generales que pueden aplicarse a una amplia gama de problemas. Otros métodos como el *método Nelder-Mead*, *Descenso de gradiente* y el *método de Newton* son algunos de los algoritmos más populares utilizados en optimizaciones no–lineales. Una revisión más profunda de estos y otros métodos puede ser consultada en [21, 22].

2.3.1. Simulated Annealing

Simulated Annealing (*SA*) es un algoritmo de búsqueda que se clasifica dentro de un método estocástico, debido a que se utiliza una variable pseudo-aleatoria, en este caso la temperatura, en un método de búsqueda metaheurístico.

Está inspirado en procesos metalúrgicos, donde los materiales son calentados en una etapa inicial y sus átomos se mueven de una manera más libre, asentándose en mejores posiciones, seguido de un enfriamiento lento para obtener una estructura cristalina ordenada. Así se comienza la exploración de la superficie a una temperatura lo suficientemente alta, donde se espera que el proceso se mueva libremente en el espacio de búsqueda (a una temperatura lo suficientemente alta, se espera que las barreras energéticas que diferencian los distintos mínimos sean superadas y de esta forma lograr una exploración de la superficie más rápida), con la esperanza de que se encuentre una buena región con el mejor mínimo local, seguido de una disminución lenta de la temperatura, reduciendo la estocasticidad y obligando a la búsqueda a converger al mínimo.

En cada iteración, se muestra una transición candidata de $\mathbf{x} \rightarrow \mathbf{x'}$, en un intento de mejorar progresivamente la solución mediante la mejora iterativa de sus partes. La probabilidad de realizar la transición del estado actual al estado candidato, se especifica mediante una función de probabilidad de aceptación, $\mathrm{P(f(x'),\ f(x),\ T)}$, que dependen de los valores de los estados y un parámetro global variable en el tiempo llamado temperatura, T, ecuación (2.5).

$$\mathrm{P(f(x'),\ f(x),\ T)\ =\ } \begin{cases} 1 & \Delta y \leq 0 \\ \mathrm{mín}(e^{-\Delta y/T},\ 1) & \Delta y > 0 \end{cases} \tag{2.5}$$

donde $\Delta \mathrm{y\ =\ f(x')\ -\ f(x)}$. Esta probabilidad de aceptación es conocida como el *criterio de Metrópolis*, y es la que le permite al algoritmo escapar de mínimos locales cuando la temperatura es alta. Finalmente se utiliza una función para reducir lentamente la temperatura a medida que avanza el algoritmo. La temperatura debe descender lentamente para alentar la convergencia; de lo contrario el método puede no cubrir la porción de la superficie donde se encuentre el mínimo global.

Algoritmo 1 Simulated Annealing

```
1: procedure SA(f, x, T, t, k_max)
2:     y ← f(x)
3:     x_best, y_best ← x, y
4:     for k in 1 : k_max do
5:         x_new = x + rand(T)
6:         y_new = f(x_new)
7:         Δy = y_new − y
8:         if Δy ≤ 0 || rand() < exp(−Δy/t(k)) then
9:             x, y = x_new, y_new
10:        end if
11:        if y_new < y_best then
12:            x_best, y_best = x_new, y_new
13:        end if
14:    end for
15:    return x_best
16: end procedure
```

Variaciones en la forma de reducción de la temperatura pueden ser utilizadas, con el fin de obtener mejores resultados o tiempos de ejecución más cortos. Entre los principales se encuentran un enfriamiento de tipo logarítmico ecuación (2.6), donde se garantiza alcanzar asintóticamente el mínimo global bajo ciertas condiciones específicas con el problema de ser lento computacionalmente, un enfriamiento exponencial ecuación (2.7), el cual es el más común y utiliza un factor de descomposición simple, y un enfriamiento rápido ecuación (2.8).

$$\mathrm{t_k\ =\ } \frac{\mathrm{t_1 \ln(2)}}{\mathrm{\ln(k+1)}} \tag{2.6}$$

$$t_{k+1} \;=\; \gamma\; t_k \tag{2.7}$$

$$t_k \;=\; \frac{t_1}{k} \tag{2.8}$$

Con $\gamma \in (0,\ 1)$ y k el número de iteraciones realizadas. Una implementación básica es mostrada en el algoritmo 1.

2.3.2. Método Nelder-Mead

El algoritmo Nelder-Mead, es un método de optimización directo, que usa un simplex [3] para recorrer el espacio en búsqueda de un mínimo.

Este método utiliza una serie de reglas que dictaminan como se actualiza el simplex en función a la evaluación de la función objetivo en sus vértices. Éste puede moverse en el espacio manteniendo su tamaño y reducirse a medida que se acerca a un nivel óptimo o mínimo.

El simplex consiste entonces en el conjunto de puntos

$$X \;=\; [x_1,\ x_2,\ x_3, \ldots, x_n,\ x_{n+1}], \tag{2.9}$$

donde sí se considera a x_h como el vértice con el **valor de función**[4] más alto, x_s el vértice con el segundo valor de función más alto, x_l el vértice con el valor de función más bajo y $\overline{x}$ la media de todos los vértices con excepción del punto más alto (x_h), centroide del simplex, una única iteración evaluará los siguientes cuatro pasos, Figura 2.1:

- **Reflexión**: refleja el punto x_h sobre el centroide:

  $$x_r \;=\; \overline{x} \;+\; \alpha(\overline{x} \;-\; x_h), \tag{2.10}$$

  donde $\alpha \;>\; 0$.

- **Expansión**: cuando el punto reflejado x_r tiene un valor de función menor que todos

[3] Un Simplex es una generalización de un tetraedro en un espacio n-dimensional.

[4] El valor de función se considera como la evaluación del punto en la función objetivo, esto es, $y_i \;=\; f(x_i)$

los puntos en el simplex, esté es alejado aún más:

$$x_e = \overline{x} + \beta(x_r - \overline{x}), \tag{2.11}$$

donde $\beta > \text{máx}(1, \alpha)$

- **Contracción**: el simplex se reduce al alejarse del peor punto:

$$x_c = \overline{x} + \gamma(x_h - \overline{x}), \tag{2.12}$$

donde $\gamma \in (0, 1)$.

- **Encogimiento**, reemplazamos todos los puntos con excepción de x_l:

$$x_i = x_l + \sigma(x_i - x_l), \tag{2.13}$$

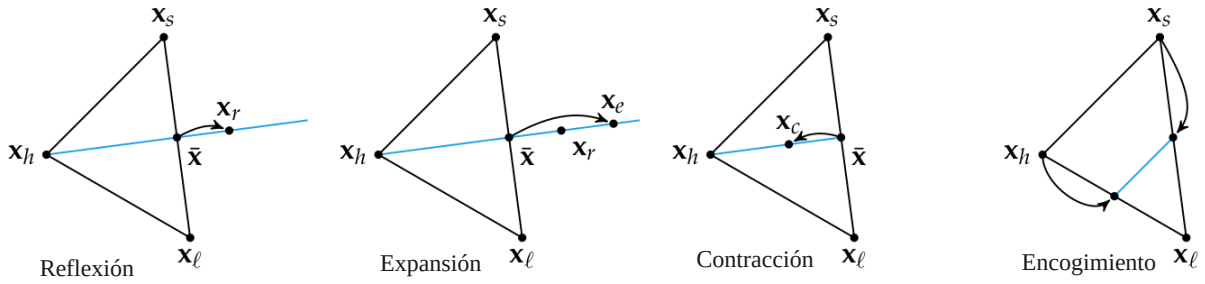

Figura 2.1: Operaciones del simplex de Nelder-Mead en dos dimensiones, tomado de [22]

El criterio de convergencia del método utiliza la desviación estándar de los valores de $y_1, y_2, y_3, \ldots, y_n, y_{n+1}$ a una tolerancia dada ϵ; si estos caen por debajo de la tolerancia, entonces el ciclo se detiene y el punto más bajo en el simplex se devuelve como un óptimo propuesto, siendo esté el principal inconveniente del método, por ejemplo, para una función muy plana en un dominio grande la solución será sensible a la tolerancia. La Figura 2.2 resume el diagrama de flujo del método.

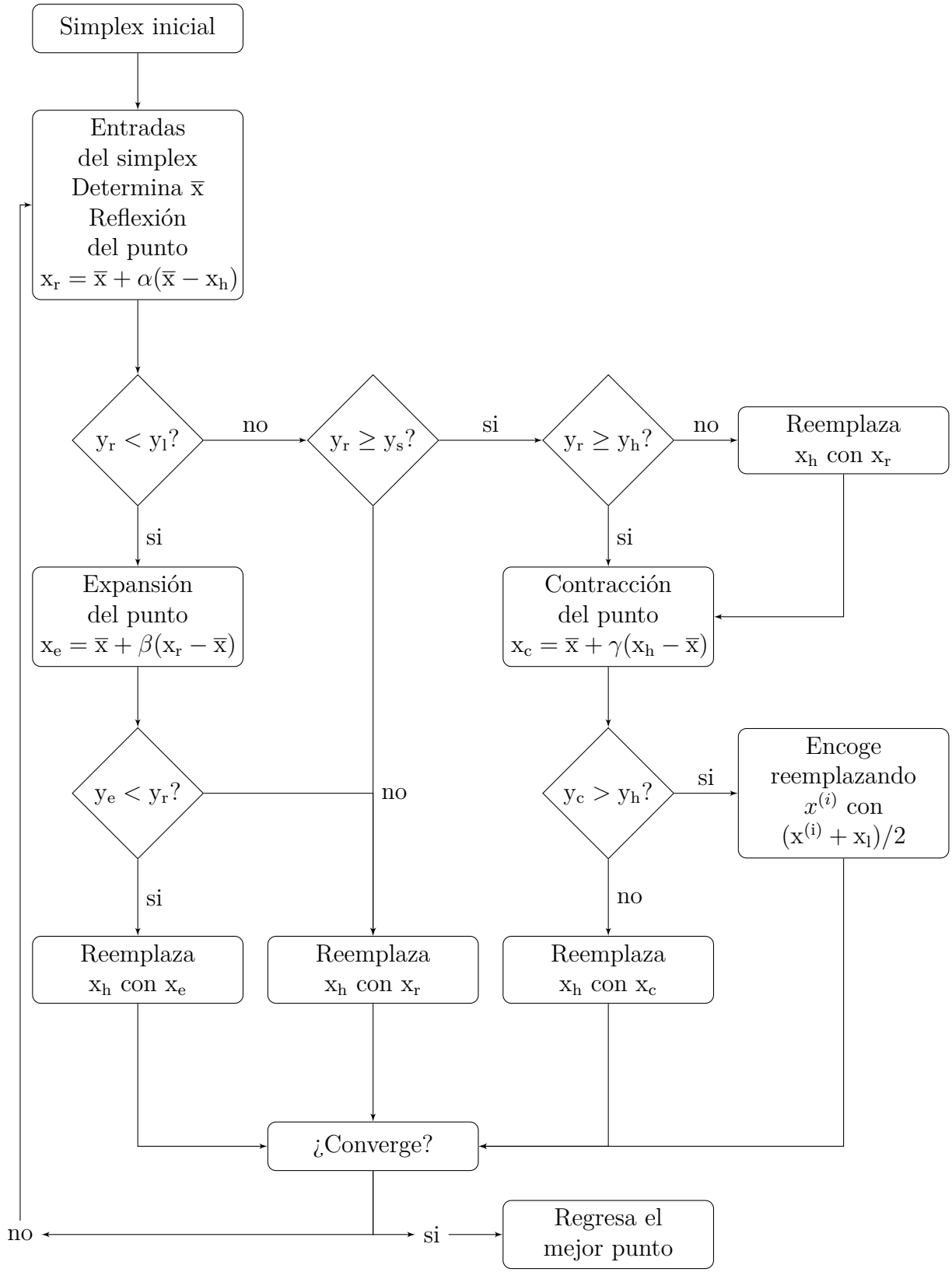

Figura 2.2: Diagrama de los del método de optimización Nelder-Mead [22]

2.3.3. Descenso de Gradiente

Descenso de gradiente es un método de optimización de primer orden, en la que se utiliza el gradiente para elegir la dirección de descenso sobre la superficie, esto es, un descenso más empinado garantiza conducir a una mejora siempre que la función objetivo sea suave, el tamaño del paso sea lo suficientemente pequeño y no estemos en un punto donde el gradiente sea cero.

De manera general se define el gradiente como

$$g_k = \nabla f(x_k), \tag{2.14}$$

donde x_k es nuestro punto asignado en la iteración k. Normalmente la dirección de descenso elegida se encuentra normalizada como

$$d_k = -\frac{g_k}{||g_k||}, \tag{2.15}$$

esto con el fin de asegurar que el algoritmo se mueva en tamaños de paso fijos en cada iteración, y se determina el nuevo punto a evaluar como

$$x_{k+1} = x_k + \alpha_k d_k, \tag{2.16}$$

con $\alpha \in \mathbb{R}^+$. Un problema común, es que este método suele funcionar mal con valles estrechos dentro de la superficie; para este caso métodos como el **gradiente conjugado** suelen funcionar mejor.

Esta última variación tiene en cuenta la información del gradiente y la dirección del descenso del paso anterior. El algoritmo comienza inicialmente con la dirección de descenso más pronunciado, esto es:

$$d_1 = -g_1, \tag{2.17}$$

Se calcula el nuevo punto según la ecuación (2.16). La dirección de descenso del siguiente paso es elegida en base a la información del gradiente del nuevo punto y la contribución de descenso del punto anterior,

$$d_{k+1} = -g_{k+1} + \beta_k d_k, \tag{2.18}$$

donde β es un escalar y valores grandes de este indican que la dirección de descenso anterior contribuye más fuertemente. Determinar los valores de β resulta complicado, por lo que autores como **Fletcher-Reeves**[23] y **Polak-Ribière**[24] proponen métodos alternativos.

2.3.4. Método de Newton

El método de Newton es un método de optimización de segundo orden. Como se vio en el método de descenso con gradiente, el conocer el valor del gradiente en un punto puede ayudar a determinar la dirección del desplazamiento; no obstante, no ayuda a determinar qué tan lejos dar un paso para alcanzar un mínimo local. Sin embargo, un método de segundo orden permite hacer una aproximación cuadrática de la función objetivo y aproximar el tamaño de paso correcto para alcanzar un mínimo local [22].

De manera general, un ajuste cuadrático sobre la función permite obtener analíticamente el punto donde la aproximación cuadrática tiene un gradiente de cero y usar esa ubicación como el punto de partida de la siguiente iteración. En una función univariada la aproximación cuadrática en un punto x_k se obtiene de la expansión de Taylor de segundo orden:

$$f(x) = f(x_k) + (x - x_k)f'(x_k) + \frac{(x - x_k)^2}{2}f''(x_k). \tag{2.19}$$

Derivando con respecto a x e igualando a cero la expresión, se obtiene la ecuación de actualización del punto para el método de Newton

$$\frac{\partial f(x)}{\partial x} = f'(x_k) + (x - x_k)f''(x_k) = 0 \tag{2.20}$$

$$x_{k+1} = x_k - \frac{f'(x_k)}{f''(x_k)}. \tag{2.21}$$

Como se evidencia, la regla de actualización implica dividir por la segunda derivada, por lo que no estará definida si está es cero y presentará inestabilidad cuando esté cercana a cero. Cuando las aproximaciones locales son deficientes, pueden conducir a un bajo ren-

dimiento del método (Figura 2.3).

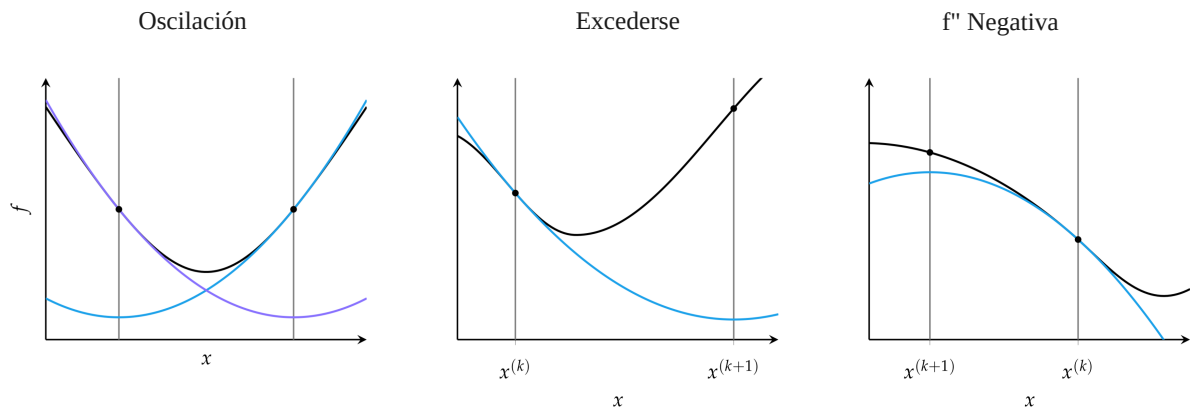

Figura 2.3: Casos en los falla el método de Newton, tomado de [22]

El método de Newton se puede expandir al caso multivariado, teniendo que la expansión de Taylor de segundo orden multivariada es

$$f(x) = f(x_k) + g_k^T(x - x_k) + \frac{1}{2}(x - x_k)^T H_k(x - x_k), \tag{2.22}$$

donde g_k y H_k, son el gradiente y la hessiana en x_k respectivamente. Derivando e igualando a cero, obtenemos

$$\nabla q(x) = g_k + H_k(x - x_k) = 0 \tag{2.23}$$

$$x_{k+1} = x_k - (H_k)^{-1} g_k. \tag{2.24}$$

En general, el método de Newton termina una vez x deja de cambiar en más de una tolerancia dada ϵ. De igual manera, se puede modificar para usar un factor de paso [22], pasos más pequeños pueden aumentar la solidez del método. El algoritmo 2, resume una implementación básica del método.

Algoritmo 2 Método de Newton

```
1: procedure Método de Newton(∇f, H, x, ϵ, k_max)
2:     k, Δ ← 1, fill(Inf, length(x))
3:     while norm(Δ) > ϵ && k ≤ k_max do
4:         Δ = H(x)/∇f(x)
5:         x −= Δ
6:         k += 1
7:     end while
8:     return x
9: end procedure
```

Los métodos Quasi-Newton, como los algoritmos BFGS [23] y L-BFGS [25], se utilizan como alternativas al método de Newton cuando el jacobiano o el hessiano de la función no están disponibles o son demasiado costosos de calcular en cada iteración, estos métodos utilizan sólo información del gradiente, evitando por lo tanto calcular de forma explícita la matriz hessiana [21].

CAPÍTULO 3

ANÁLISIS POR INTERVALOS

En este capítulo se introduce el análisis de intervalos como la rama del análisis numérico dedicada al control de errores de redondeo generados por los métodos convencionales de punto flotante, cuyo principal problema radica en no poder representar de manera exacta todos los números reales en la computadora. Estos métodos permiten obtener cálculos rigurosos; por lo que constituyen un enfoque privilegiado para la optimización global de manera confiable.

3.1. Aritmética en la computadora

Para que una computadora pueda manejar números naturales, enteros, racionales, reales o incluso complejos, es necesario representar estos números en la memoria con un formato bien definido y flexible. De manera general, cualquier máquina discreta, máquina de estados finitos con un formato estático (cantidad fija de memoria), trabaja con una aritmética que utiliza un número finito de dígitos. Un número real tiene, salvo pocas excepciones, un número infinito de dígitos y el representar estos números en computadoras con una capacidad finita de memoria introduce un error debido al redondeo en la representación. Tener dichos errores presentes, genera una propagación del error conforme realizamos operaciones aritméticas y en algunos casos llegando a provocar grandes errores en el resultado final.

3.1.1. Números naturales

Una forma habitual de poder representar los números naturales es utilizar el sistema decimal o de base 10, donde cada número se puede representar por una cadena de dígitos de la forma

$$a_n\ a_{n-1}\ a_{n-2} \cdots a_1\ a_0 \ = \ a_0 + a_1 \cdot 10^1 + \ldots + a_{n-1} \cdot 10^{n-1} + a_n \cdot 10^n, \qquad (3.1)$$

donde $a_i \in \{0,\ 1,\ \ldots,\ 9\}$ y cada dígito se ve afectado por un factor de escala que depende de su posición. Sin embargo, los ordenadores actuales utilizan el sistema binario o de base 2 para la representación de números. Los dígitos de este sistema son 0 y 1, denominados *bits* y se representan físicamente mediante los dos estados de conducción (on) y corte (off) de un transistor funcionando como conmutador. Un ejemplo de esto es

$$(1101)_2 \ = \ 1 \cdot 2^3 + 1 \cdot 2^2 + 0 \cdot 2^1 + 1 \cdot 2^0 \ = \ 8 + 4 + 1 = 13. \tag{3.2}$$

De manera similar el poder obtener la cadena binaria del número entero se realiza dividiendo reiteradamente el número entre dos hasta que el cociente sea la unidad:

123	61	30	15	7	3	1
1	1	0	1	1	1	1

Escribiendo este cociente y todos los restos en orden inverso a como los hemos obtenidos se obtienen los dígitos del número en binario, $123 \ = \ (1111011)_2$.

3.1.2. Números enteros

Para poder representar un número entero, sería necesario dedicar un bit para el signo; sin embargo, entonces el cero tendría dos representaciones equivalentes (+0 y -0). A modo de evitar esta duplicidad, en los ordenadores los números enteros de n dígitos se presentan comúnmente por un complemento a 2.

En el complemento a dos, el bit más significativo representa el signo del número entero. Si es 0, el número es positivo, si es 1 es negativo. El módulo del número negativo se calcula, complementando a uno el número, es decir, cambiando los 0 por 1 y viceversa, y luego sumándole 1.

Consideremos la siguiente cadena de bits 10110101. Considerando lo anteriormente descrito, el primer bit corresponde al signo, siendo este negativo. Con el resto hacemos un complemento a 2 intercambiando los 1 por 0 y viceversa: $0110101 \rightarrow 1001010$, que correspondiente al número 74 que al sumarle 1 quedaría como 75. Así pues, la cadena 10110101 corresponde al número -75.

Resulta evidente que utilizando el complemento a 2, el cero tiene una única representación

(*todos los bits en cero*) y el intervalo de números representables es $[-2^{n-1}, 2^{n-1} - 1]$. La longitud n en bits de un número entero suele ser un múltiplo de 8.

3.1.3. Números de punto flotante

Dado que los números reales pueden tener un número infinito de dígitos, estos se aproximan en la computadora mediante un formato bien definido de punto flotante el cual utiliza solo un número finito de dígitos [26, 27].

Una manera conveniente de poder representar los números reales en la computadora es la siguiente:

$$\mathrm{x} = (-1)^{\sigma} \cdot \mathrm{M} \cdot \beta^{\mathrm{r}}, \tag{3.3}$$

donde σ define el signo, M es la mantisa, β es la base de la representación y r es el exponente entero. La mantisa es un número real y positivo que se representa como

$$\mathrm{M} = \mathrm{b}_0.\mathrm{b}_{-1}\mathrm{b}_{-2}\mathrm{b}_{-3}\ldots\mathrm{b}_{-\mathrm{n}}, \tag{3.4}$$

donde cada dígito $\mathrm{b_k}$ cumple $0 \leq \mathrm{b_k} \leq \beta - 1$. Dicho lo anterior, es fácil notar que bajo esta representación un número en la computadora, la cual utiliza un sistema binario $\beta = 2$, puede tener dos representaciones distintas ($\mathrm{b}_0 = 0,\ 1$), lo que introduce otro tipo de redundancias. Una manera de resolver el problema es normalizar la mantisa, imponiendo que el valor $\mathrm{b}_0 \neq 0$, obteniendo necesariamente $\mathrm{b}_0 = 1$.

Por convención [27] se tiene que, para un sistema de 64 bits estos se encuentran repartidos de la siguiente manera

- 1 bit para σ;
- 11 bits para r;
- 52 bits para M.

Los 11 bits de r se reservan para el exponente y su signo, el cual es un número entero que se representa en exceso a $2^{n-1} - 1$ con n es el número de bits reservados para r. La representación sesgada del exponente es de la forma $\mathrm{r} = \mathrm{r_e} - 1023$, donde $\mathrm{r_e}$ puede tomar los valores en el rango de [0, 2047], no obstante, el exponente se encuentra normalizado,

por lo que en si solo toma los valores entre [1, 2046] y los valores extremos del rango original se reservan para representar valores especiales. El rango del exponente r se puede acotar de una manera $r_{min} \leq r \leq r_{max}$ en el intervalo de [-1022, 1023] y estos constituyen el set de **números normales** [26, 27].

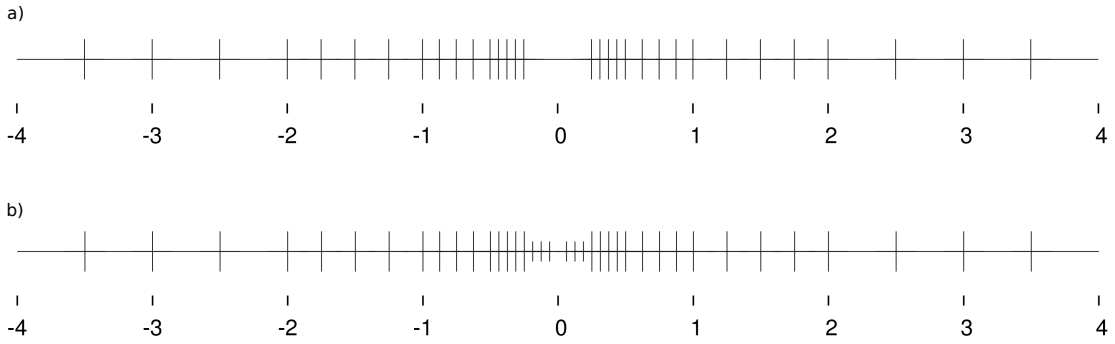

Figura 3.1: Números de punto flotante a) sin incluir los números subnormales y b) incluyendo los números subnormales. Tomada de

La representación de punto flotante permite ampliar el rango representable a costa del aumento del espacio entre los números, *un espacio no uniforme* (Figura 3.1 a), el cual se hace mayor conforme crece el número en magnitud. Esto resulta importante en el rango que divide al real negativo más grande y el real positivo más pequeño. Para contrarrestar este inconveniente es necesario definir otro set de números denominados **números subnormales** [26, 27], los cuales se caracterizan por el uso del bit 0 en vez del 1 en el valor b_o de la mantisa y el valor de 0 en r_e del exponente, que se interpreta como el valor más pequeño permitido. Esto último se hace con el fin de poder representar los números entre el mayor número negativo y el menor número positivo de los números normales, Figura 3.1 b.

El extremo 2047 del exponente se deja para representar valores especiales ($\pm\infty$ y NaN). Los primeros se representan cuando la mantisa es 0 y se producen cuando una operación aritmética genera un número más grande que el máximo representable, es decir, se produce un desbordamiento por exceso u *overflow*. NaN se genera en operaciones aritméticas de resultado no determinado como $0/0$, ∞/∞, etc. cuya mantisa es diferente de cero. La tabla 3.1 resume los valores que se pueden obtener con la representación de punto flotante.

3.1.3.1. Redondeo

Por su naturaleza, todos los números expresados en un formato de punto flotante se limitan al número de bits de precisión que tenga la máquina, lo que limita el conjunto de números que se puede representar exactamente. Si el número no tiene una representación

r_e en binario	r_e	Valor numérico
$(00000000000)_2$	0	± 0 si, $m_i = 0$ para todo i
$(00000000000)_2$	0	$\pm(0.m_1m_2m_3 \ldots m_52)_2 \times 2^{-1022}$
$(00000000001)_2$	1	$\pm(1.m_1m_2m_3 \ldots m_52)_2 \times 2^{-1022}$
$(00000000010)_2$	2	$\pm(1.m_1m_2m_3 \ldots m_52)_2 \times 2^{-1021}$
$\vdots$	$\vdots$	$\vdots$
$(01111111111)_2$	1023	$\pm(1.m_1m_2m_3 \ldots m_52)_2 \times 2^{0}$
$(10000000000)_2$	1024	$\pm(1.m_1m_2m_3 \ldots m_52)_2 \times 2^{1}$
$\vdots$	$\vdots$	$\vdots$
$(11111111110)_2$	2046	$\pm(1.m_1m_2m_3 \ldots m_52)_2 \times 2^{1023}$
$(11111111111)_2$	2047	$\pm\infty$ si, $m_i = 0$ para todo i
$(11111111111)_2$	2047	NaN si existe algún $m_i \neq 0$

Tabla 3.1: Formato binario de todos los números flotantes de doble precisión

exacta, entonces la conversión requiere una elección de qué número de punto flotante usar para representar el valor original. La representación elegida tendrá un valor diferente del original y a este valor ajustado se le llamará valor redondeado.

Aparte de la precisión de la máquina, la base que se esté usando también determinará cuáles números se pueden representar de manera exacta. En base 2, solo los racionales cuyo denominador sea potencia de 2 como ($\frac{1}{2}$, $\frac{5}{16}$) se pueden representar de manera exacta; cualquier otro racional con un factor primo distinto de 2 tendrá una expansión binaria infinita. Por ejemplo, el número decimal d0.1 no es representable en punto flotante de precisión finita, sino que *la representación binaria de 0.1 tiene una secuencia de "1100" continuando sin fin.*

Con el objetivo de poder representar valores cuya representación binaria es una cadena infinita de bits, se pueden implementar 4 casos de redondeo principales [26, 27]:

- Redondeo hacia arriba (hacia $+\infty$)
- Redondeo hacia abajo (hacia $-\infty$)
- Redondeo hacia el cero (truncado)
- Redondeo al más cercano

Para cualquier tipo de redondeo que se utilice, es necesario el uso de bits adicionales de precisión, **bits de guarda**, denominados guard (G), round (R) y sticky (S). El método que se usa más comúnmente es el redondeo al más cercano, el cual redondea el número al valor representable más cercano, y en caso de empate se elige el valor que haría que el resultado final sea un dígito par.

Además de la incapacidad de no poder representar números como π y 0.1 exactamente, pueden ocurrir otros fenómenos como **cancelación catastrófica**, donde la resta de operando casi iguales puede causar una perdida extrema de precisión, y **pruebas de igualdad problemáticas**, donde dos secuencias computacionales que son matemáticamente iguales pueden producir diferentes valores de punto flotante, un ejemplo de este último seria:

```
In [1]:  a = 1234.567
         b = 1.234567
         c = 3.333333

         (a+b)*c == a*c + b*c
Out[1]:  false
```

donde, al ejecutar las anteriores líneas, la respuesta será que la igualdad es falsa, por lo que no necesariamente la propiedad distributiva se cumple, y de modo similar tampoco la propiedad asociativa. Con lo anterior queda evidenciado que el uso de un formato de punto flotante puede generar pequeños errores que, en un método numérico, en el cual se realizan operaciones de manera sucesiva, pueden acumularse y crecer catastróficamente conduciendo a un resultado final de muy baja exactitud. Modos alternativos como punto flotante de precisión múltiple y la aritmética de intervalos pueden ser utilizados.

3.2. Aritmética de intervalos

Como se vio en la sección anterior la aritmética de punto flotante es un método de representación aproximada de números en máquinas discretas. Un número flotante se representa por un signo, una mantisa y un exponente. Las máquinas discretas utilizan una mantisa

de tamaño finito, lo que conduce a errores de aproximación cuando los números no son representables de manera exacta. Una forma de controlar este tipo de errores fue presentada en la tesis doctoral de Moore [28], en la cual sentó las bases para el cálculo de intervalos. La idea general es representar cada resultado de un cálculo con un intervalo que contenga el resultado exacto. De esta manera, cada número se representa como

$$X \;=\; [a,\; b] = [\underline{X},\; \overline{X}] \qquad\qquad \underline{X} \leq x \leq \overline{X}. \tag{3.5}$$

La aritmética de intervalos extiende la aritmética real a intervalos de manera rigurosa. Las funciones elementales (+, -, ×, /, log, exp, etc.) se implementan redondeando los cálculos hacia afuera, es decir, el límite inferior del intervalo se calcula con un modo de redondeo hacia $-\infty$ y el límite superior se calcula con un modo de redondeo hacia ∞. El valor exacto se garantiza para pertenecer al intervalo obtenido.

3.2.1. Términos y conceptos básicos

Se aclara que nos referiremos como intervalos a los intervalos cerrados, y no a los otros tipos de intervalos que pueden aparecer en matemáticas como lo son los intervalos abiertos o medio abiertos. De igual manera, adoptaremos la convención de escribir un intervalo y sus límites (inferior y superior) en letras mayúsculas, como lo escrito en la ecuación (3.5)

Recordemos que un intervalo cerrado [a, b] representa el conjunto de todos los números reales dados por

$$[a,\; b] \;=\; \{x \;\in \mathbb{R} :\; a \leq x \leq b\} \tag{3.6}$$

Diremos que un intervalo es degenerado cuando $\underline{X} \;=\; \overline{X}$, en cuyo caso solo contendrá un número real x, y por convención se identificará como el número real x que contenga, e.g.

$$2 \;=\; [2,\; 2]. \tag{3.7}$$

Al igual que en la teoría de conjuntos, se define la intersección y unión entre intervalos [29]. La **intersección** entre dos intervalos X y Y es

$$\mathrm{X} \cap \mathrm{Y} \;=\; \begin{cases} \emptyset & \overline{\mathrm{Y}} < \underline{\mathrm{X}} \;\vee\; \overline{\mathrm{X}} < \underline{\mathrm{Y}} \\ [\text{máx}(\underline{\mathrm{X}},\; \underline{\mathrm{Y}}),\; \text{mín}(\overline{\mathrm{X}},\; \overline{\mathrm{Y}})] & \overline{\mathrm{Y}} > \underline{\mathrm{X}} \;\wedge\; \overline{\mathrm{X}} \geq \underline{\mathrm{Y}} \end{cases} \tag{3.8}$$

Sin embargo, la **unión** de dos intervalos, en general, no da como resultado un intervalo, por lo que es necesario redefinir esta operación para intervalos de la siguiente manera

$$\mathrm{X} \cup \mathrm{Y} \;=\; [\text{mín}(\underline{\mathrm{X}},\; \underline{\mathrm{Y}}),\; \text{máx}(\overline{\mathrm{X}},\; \overline{\mathrm{Y}})], \tag{3.9}$$

de esta manera el resultado siempre será un intervalo y siempre se cumple que

$$\mathrm{X} \cup \mathrm{Y} \subseteq \mathrm{X} \cup \mathrm{Y}. \tag{3.10}$$

Otros conceptos importantes por conocer a la hora de trabajar con intervalos son

- **Tamaño de un intervalo**, se define y se denota como

$$\mathrm{w}(\mathrm{X}) \;:=\; \overline{\mathrm{X}} \;-\; \underline{\mathrm{X}}. \tag{3.11}$$

- **Valor absoluto de un intervalo**, se denota como $|X|$ y corresponde al máximo valor absoluto entre sus limites

$$|\mathrm{X}| \;:=\; \text{máx}(|\underline{\mathrm{X}}|,\; |\overline{\mathrm{X}}|). \tag{3.12}$$

- **Punto medio de un intervalo**, dado por

$$\mathrm{m}(\mathrm{X}) \;=\; \frac{1}{2}(\underline{\mathrm{X}} \;+\; \overline{\mathrm{X}}). \tag{3.13}$$

Estos conceptos se pueden extender al caso de vectores y matrices de intervalos [29]. Un vector n-dimensional, ecuación (3.14), se expresa como una n-tupla de intervalos o caja n-dimensional.

$$\mathrm{X} \;=\; (\mathrm{X}_1,\; \mathrm{X}_2,\; \mathrm{X}_3, \ldots,\; \mathrm{X}_\mathrm{n}). \tag{3.14}$$

Con las variaciones adecuadas, ecuaciones (3.15)–(3.17), muchas de las nociones de intervalos ordinarios pueden extenderse a vectores de intervalos.

$$X \cap Y \;=\; \begin{cases} \emptyset & X_i \cap Y_i \;=\; \emptyset \;\; \text{con } i = 1, \ldots, n \\ (X_1 \cap Y_1, \ldots,\; X_n \cap Y_n) & X_i \cap Y_i \;\neq\; \emptyset \;\; \text{con } i = 1, \ldots, n \end{cases} \tag{3.15}$$

$$w(X) \;=\; \max_i w(X_i). \tag{3.16}$$

$$m(X) \;=\; (m(X_1),\; m(X_2), \ldots,\; m(X_n)). \tag{3.17}$$

De esta manera podremos trabajar en un sistema de intervalos con la siguientes propiedades [30]:

- **Correcto**, basado en el teorema fundamental de la aritmética de intervalos, el cual consiste que al evaluar una función usando intervalos, esta produce otro intervalo el cual contiene todos los resultados de evaluaciones puntuales.
- **Total**, ya que se busca que esté definido para todos los argumentos posibles.
- **Cerrado**, ya que es deseable que una operación con intervalos produzca un intervalo.
- **Óptimo**, para que el intervalo calculado no sea más grande de lo necesario.
- **Eficiente**, en donde los tiempos de computación sean razonables.

3.2.2. Operaciones con intervalos

Para poder definir las operaciones básicas para intervalos, resulta necesario tener presente la idea que computar con intervalos es computar con conjuntos. Bajo este razonamiento cualquier operación entre intervalos se puede definir como

$$X \diamond Y \;=\; \{x \diamond y \;:\; x \in X,\; y \in Y\}, \tag{3.18}$$

donde $\diamond$ representa cualquier operación binaria y el resultado de dicha operación es un intervalo que contiene todos los valores de la operación entre cada posible par de números

pertenecientes a los intervalos iniciales [29, 30].

3.2.2.1. Suma de intervalos

Tendiendo dos intervalos X y Y, de manera tal que

$$\underline{X} \leq x \leq \overline{X} \quad \wedge \quad \underline{Y} \leq y \leq \overline{Y},$$

la suma de ambas desigualdades deja en visto que las sumas numéricas $x + y$ se limitan por

$$\underline{X} + \underline{Y} \leq x + y \leq \overline{X} + \overline{Y},$$

por lo tanto, podemos resumir que la suma entre 2 intervalos viene definida por

$$X + Y \;=\; [\underline{X} + \underline{Y},\ \overline{X} + \overline{Y}]. \tag{3.19}$$

3.2.2.2. Resta de intervalos

De manera similar a la suma podemos definir que la resta entre dos intervalos X y Y vales que

$$\underline{X} \leq x \leq \overline{X} \quad \wedge \quad \underline{Y} \leq y \leq \overline{Y},$$

la resta de ambas desigualdades deja en visto que las restas numéricas $x - y$ se limitan por

$$\underline{X} - \underline{Y} \leq x - y \leq \overline{X} - \overline{Y},$$

por lo cual

$$X - Y \;=\; [\underline{X} - \overline{Y},\ \overline{X} - \underline{Y}]. \tag{3.20}$$

3.2.2.3. Multiplicación de intervalos

En términos de los limites superiores e inferiores podemos definir que la multiplicación entre dos intervalos X y Y viene dada por

$$X \cdot Y \;=\; [\text{mín}(S),\ \text{máx}(S)] \quad \text{con} \quad S \;=\; \{\underline{X}\underline{Y},\ \underline{X}\overline{Y},\ \overline{X}\underline{Y},\ \overline{X}\overline{Y}\}. \tag{3.21}$$

Sin embargo, conociendo los signos reales de los límites superiores e inferiores, la fórmula anterior puede ser desglosada en 9 casos específicos (Tabla 3.2), de los cuales en 8 casos solo es necesario computar 2 operaciones. La eficiencia de utilizar la ecuación (3.21) o los casos dados en la tabla 3.2 dependerá en gran medida del lenguaje de programación y el hardware de la host que se vaya a ser utilizado [29].

Caso	$\underline{X \cdot Y}$	$\overline{X \cdot Y}$
$0 \leq \underline{X}$ y $0 \leq \underline{X}$	$\underline{X} \cdot \underline{Y}$	$\overline{X} \cdot \overline{Y}$
$\underline{X} < 0 < \overline{X}$ y $0 \leq \underline{Y}$	$\underline{X} \cdot \overline{Y}$	$\overline{X} \cdot \overline{Y}$
$\overline{X} \leq 0$ y $0 \leq \underline{Y}$	$\underline{X} \cdot \overline{Y}$	$\overline{X} \cdot \underline{Y}$
$0 \leq \underline{X}$ y $\underline{Y} < 0 < \overline{Y}$	$\overline{X} \cdot \underline{Y}$	$\overline{X} \cdot \overline{Y}$
$\overline{X} \leq 0$ y $\underline{Y} < 0 < \overline{Y}$	$\underline{X} \cdot \overline{Y}$	$\underline{X} \cdot \underline{Y}$
$0 \leq \underline{X}$ y $\overline{Y} \leq 0$	$\overline{X} \cdot \underline{Y}$	$\underline{X} \cdot \overline{Y}$
$\underline{X} < 0 < \overline{X}$ y $\overline{Y} \leq 0$	$\overline{X} \cdot \underline{Y}$	$\underline{X} \cdot \underline{Y}$
$\overline{X} \leq 0$ y $\overline{Y} \leq 0$	$\underline{X} \cdot \underline{Y}$	$\overline{X} \cdot \overline{Y}$
$\underline{X} < 0 < \overline{X}$ y $\underline{Y} < 0 < \overline{Y}$	$\text{mín}(\underline{X} \cdot \overline{Y}, \overline{X} \cdot \underline{Y})$	$\text{máx}(\underline{X} \cdot \underline{Y}, \overline{X} \cdot \overline{Y})$

Tabla 3.2: Casos de la multiplicación entre dos intervalos X y Y

3.2.2.4. División de intervalos

Al igual que con los números reales, la división se puede realizar como la multiplicación por el recíproco del segundo operando,

$$X/Y \;=\; X \cdot (1/Y), \tag{3.22}$$

donde

$$1/\mathrm{Y} \;:=\; \{\mathrm{y} \;:\; 1/\mathrm{y} \in \mathrm{Y}\} \;=\; [1/\overline{\mathrm{Y}},\; 1/\underline{\mathrm{Y}}], \tag{3.23}$$

suponiendo que $0 \notin \mathrm{Y}$. No obstante, la anterior definición no resulta suficiente para casos donde $0 \in \mathrm{Y}$ por lo que es necesario extenderla a los demás casos, ecuación (3.24), a esto se le denomina como aritmética de intervalos extendida [31].

$$\frac{1}{[\mathrm{a},\; \mathrm{b}]} = \begin{cases} \emptyset & \text{Si } \mathrm{a} = \mathrm{b} = 0 \\ \left[\frac{1}{\mathrm{b}},\; \frac{1}{\mathrm{a}}\right] & \text{Si } 0 < \mathrm{a} \text{ o } \mathrm{b} < 0 \\ \left[\frac{1}{\mathrm{b}},\; +\infty\right] & \text{Si } \mathrm{a} = 0 \\ \left[-\infty,\; \frac{1}{\mathrm{a}}\right] & \text{Si } \mathrm{b} = 0 \\ \left[-\infty,\; \frac{1}{\mathrm{a}}\right] \cup \left[\frac{1}{\mathrm{b}},\; +\infty\right] & \text{Si } \mathrm{a} < 0 < \mathrm{b} \end{cases} \tag{3.24}$$

3.3. Propiedades algebraicas

Resulta fácil de evidenciar que tanto la suma y la multiplicación de intervalos son conmutativas y asociativas, esto es respectivamente

$$\mathrm{X} + (\mathrm{Y} + \mathrm{Z}) \;=\; (\mathrm{X} + \mathrm{Y}) + \mathrm{Z} \qquad \wedge \qquad \mathrm{X}(\mathrm{YZ}) \;=\; (\mathrm{XY})\mathrm{Z}, \tag{3.25}$$

y se cumple para cualesquiera 3 intervalos X, Y, Z.

Del mismo modo se puede decir que los elementos de identidad para la suma y multiplicación de intervalos son los intervalos degenerados 0, y 1 respectivamente.

$$\begin{aligned} 0 + \mathrm{X} = X + 0 \;&= X, \\ 1 \cdot X = \; X \cdot 1 \;&= X. \end{aligned} \tag{3.26}$$

No obstante, los inversos aditivo y multiplicativo no existen, excepto para los intervalos degenerados, esto es

$$\mathrm{X} + (-\mathrm{X}) \;=\; [\underline{\mathrm{X}} - \overline{\mathrm{X}},\; \overline{\mathrm{X}} - \underline{\mathrm{X}}], \tag{3.27}$$

$$X/X = \begin{cases} [\underline{X}/\overline{X},\ \overline{X}/\underline{X}] & \text{Si } 0 < \underline{X} \\ [\overline{X}/\underline{X},\ \underline{X}/\overline{X}] & \text{Si } \overline{X} < 0 \end{cases} . \tag{3.28}$$

Sin embargo, siempre tendremos la inclusión de que $0 \in X - X$ y $1 \in X/X$. La ley distributiva de la aritmética ordinaria, ecuación (3.29), también falla al referirnos a los intervalos

$$x(y + z) = xy + xz, \tag{3.29}$$

esto se puede evidenciar tomado los intervalos $X = [1,\ 2]$, $Y = [1,\ 1]$ y $Z = -[1,\ 1]$ donde

$$\begin{aligned} X(Y + Z) &= [1,\ 2] \cdot ([1,\ 1] - [1,\ 1]) \\ &= [1,\ 2] \cdot [0,\ 0] \\ &= [0,\ 0], \end{aligned} \tag{3.30}$$

mientras que

$$\begin{aligned} XY + XZ &= [1,\ 2] \cdot [1,\ 1] - [1,\ 2] \cdot [1,\ 1]) \\ &= [1,\ 2] - [1,\ 2] \\ &= [-1,\ 1]. \end{aligned} \tag{3.31}$$

Sin embargo, si existe la *ley subdistributiva*:

$$X(Y + Z) \subseteq XY + XZ. \tag{3.32}$$

Este es el resultado de un fenómeno denominado dependencia de intervalo, el cual será objetivo de discusión más adelante.

3.4. Funciones de inclusión

La principal razón para pasar al reino de los intervalos es que podemos obtener de una manera simple la forma de encerrar el rango de una función real $R(f;\ D) = \{f(x) :\ x \in D\}$. Excepto en los casos más triviales, las matemáticas proporcionan pocas herramientas para describir este conjunto.

Se comenzará extendiendo las funciones reales a funciones de intervalos; esto significa

funciones que toman y dan como resultado intervalos en vez de números reales. En la sección anterior dimos la teoría básica de cómo operar dos intervalos. De esta manera, simplemente sustituyendo todas las ocurrencias de la variable real x por el intervalo X produce una función de intervalo F(X), llamada *extensión de intervalo natural* de f.

Si ninguna singularidad está presente, se cumple que

$$\mathrm{R(f;\ x)\ \subseteq F(X).} \tag{3.33}$$

Funciones multidimensionales, $\mathrm{f} : \mathbb{R}^{\mathrm{n}} \to \mathbb{R}$ también pueden ser extendidas a una función de intervalos de un manera similar. El argumento de la función es entonces un vector de intervalos.

Un ejemplo de esto es la función $\mathrm{f}(\mathrm{x}_1,\ \mathrm{x}_2) = \mathrm{x}_1 \mathrm{e}^{\mathrm{x}_2 - \mathrm{x}_1}$ en el dominio de $\mathrm{x}_1 = [0,\ 1]$ y $\mathrm{x}_2 = [3,\ 4]$. Por la relación (3.2) tendremos

$$\begin{aligned} \mathrm{R\,(f;\ ([0,\ 1],\ [3,\ 4]))} &\subseteq \mathrm{F([0,\ 1],\ [3,\ 4])} \\ &= [0,\ 1]\ \mathrm{e}^{[3,\ 4]-[0,\ 1]} \\ &= [0,\ 1]\ \mathrm{e}^{[2,\ 4]} \\ &= [0,\ 1][\mathrm{e}^2,\ \mathrm{e}^4] = [0,\ \mathrm{e}^4] \end{aligned} \tag{3.34}$$

3.5. Problema de dependencia

Una de las principales razones por las cuales el simple reemplazo de cálculos de punto flotante por intervalos en un algoritmo existente no es probable que conduzca a resultados satisfactorios, es la dificultad que existe en dar aproximaciones óptimas de las imágenes en una función objetivo. Esto se debe a un problema de dependencia que tiene lugar en expresiones con múltiples ocurrencias de variables, esto es, los operadores de intervalos consideran cada ocurrencia de una variable de manera independiente, y de esta manera, se sobreestima el rango de una función de manera dramática.

Consideremos la resta de intervalos $\mathrm{X} - \mathrm{X}$, con $\mathrm{X} = [-1,\ 1]$

$$\mathrm{X - X} = [-1,\ 1] - [-1,\ 1] = [-2,\ 2]. \tag{3.35}$$

Aquí aun cuando esperamos que el resultado sea el intervalo degenerado 0, el error de

la sobreestimación es de $2(b-a)$, el cual corresponde a dos veces el ancho del intervalo X. La aritmética de intervalos al ignorar las múltiples ocurrencias de X, construye una extensión natural de la función f(X) con bastantes grados de libertad. No obstante, en el teorema fundamental de la aritmética de intervalos [29], se demostró que, en circunstancias particulares, la aritmética de intervalos no sobreestima el rango de la función.

Lo anterior se cumple si en una función f continua sobre una caja X en el espacio, todas las variables ocurren como máximo una vez en la expresión de la función, de esta manera la extensión del intervalo natural de f proporciona la imagen óptima, esto es:

$$F_{nat}(X) = f(X). \tag{3.36}$$

Es importante recalcar que la sobreestimación también puede ser producto de la discontinuidad de la función en alguna región del espació [32].

3.6. Refinamientos más eficientes

En este punto, resulta claro que resolver cualquier tipo de problema en el que exista múltiples ocurrencias de las variables utilizando métodos de intervalos es, por lo tanto, una tarea ardua. Sin embargo, diversos autores han abordado el problema de la sobreestimación [29, 32, 33, 34] y han propuesto distintos métodos que van desde reescribir la expresión de la función, extensiones de segundo orden, pruebas de monotonicidad y reducir el ancho del intervalo, con el fin de minimizar el problema de dependencia lo máximo posible.

Uno de los métodos más sencillos es reescribiendo la expresión de la función, siempre y cuando sea posible. Con esto se busca minimizar la cantidad de ocurrencias de las variables dentro de la expresión y en el caso más ideal que las variables tengan una ocurrencia única, logrando eliminar el efecto de dependencia.

Por ejemplo consideremos entonces la función $f(x,\ y) = \frac{x-y}{x+y}$ con $x \in X = [6,\ 8]$ y $y \in Y = [2,\ 4]$. El rango exacto de la función en dicha región del espacio está dado por $f(X,\ Y) = [\frac{1}{5},\ \frac{3}{5}]$. La extensión de intervalo natural de f produce el siguiente resultado

$$F_{nat}(X,\ Y) = \frac{X-Y}{X+Y} = \frac{[6,\ 8]-[2,\ 4]}{[6,\ 8]+[2,\ 4]} = \frac{[2,\ 6]}{[8,\ 12]} = \left[\frac{1}{6},\ \frac{3}{4}\right], \tag{3.37}$$

así se tiene que $f(X,\ Y) \subset F_{nat}(X,\ Y)$. Ahora reescribiendo f como

$$f(x,\ y) \ = \ \frac{x-y}{x+y} \ = \ \frac{x+y-2y}{x+y} \ = \ 1-\frac{2y}{x+y} \ = \ 1-\frac{2}{1+\frac{x}{y}}, \tag{3.38}$$

la extensión de intervalo natural de la función reescrita produce el siguiente rango

$$F_{nat2}(X,\ Y) \ = \ 1-\frac{2}{1+\frac{X}{Y}} \ = \ 1-\frac{2}{1+\frac{[6,\ 8]}{[2,\ 4]}} \ = \ \left[\frac{1}{5},\ \frac{3}{5}\right], \tag{3.39}$$

con lo que decimos que $f(X,\ Y) = F_{at2}(X,\ Y)$, esto es, reescribiendo la expresión de tal modo que las variables tengan una única ocurrencia en la expresión nos conduce al rango exacto. No obstante, como se describió anteriormente, la mayoría de los problemas actuales son complejos, por lo que reescribir la expresión de la función, normalmente no es posible.

3.6.1. Extensiones de segundo orden

Consideremos $f : D \subset \mathbb{R} \to \mathbb{R},\ X \in \mathbb{IR}(D)$[1] y $c \in X$ (por ejemplo el punto medio de X). El teorema de Taylor establece que, cuando f es $m-1$ veces diferenciable en c y m veces diferenciable en el intervalo abierto, tenemos para $x \in X$

$$\begin{aligned} f(x) \ &= \ \sum_{k=0}^{m-1} \frac{f^{(k)}(c)}{k!}(x-c)^k + \frac{f^{(m)}(\xi)}{m!}(x-c)^m \\ &\in \ \sum_{k=0}^{m-1} \frac{f^{(k)}(c)}{k!}(x-c)^k + \frac{F^{(m)}(\xi)}{m!}(x-c)^m. \end{aligned} \tag{3.40}$$

donde ξ es un número real entre c y x, y $F^{(m)}$ es la extensión natural de intervalo de $f^{(m)}$. Esta inclusión define la extensión de intervalo de Taylor de orden m. Las formas de Taylor más comunes son la lineal ($m \ = \ 1$) y la cuadrática ($m \ = \ 2$)[32].

La forma lineal de Taylor se conoce comúnmente como el **mean value form** (F_{mv}), ecuación (3.2), donde $c \ = \ m(X)$, y F_{mv} tiene la propiedad de ser de inclusión isotónica[2] si F' también es de inclusión isotónica [33], y tiene una convergencia cuadrática si F' es una función de Lipschitz continua [35]

$$F_{mv}(X,\ c) \ = \ f(c) + F'(X)(X-c). \tag{3.41}$$

[1] $\mathbb{IR}$ representa el conjunto de intervalos con limites reales

[2] Una función $F = F(X_1, \ldots,\ X_n)$ es de inclusión isotónica si para todo $Y_i \subseteq X_i$ con $i = 1, \ldots,\ n$ se cumple que $F(Y_1, \ldots,\ Y_n) \subseteq F(X_1, \ldots,\ X_n)$

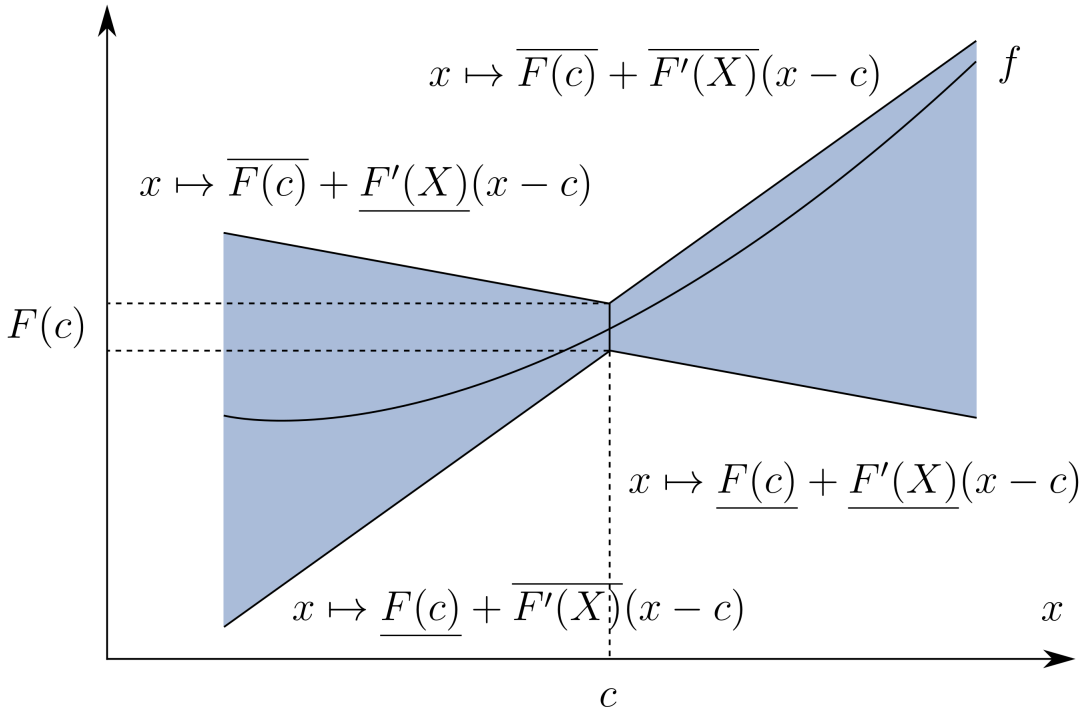

Figura 3.2: Extensión del mean value form tomada de [32]

La figura 3.2 representa la extensión de intervalo utilizando el mean value form sobre una función univariada. Esta extensión de Taylor puede ser fácilmente ampliada a funciones multivariadas $f : D \subset \mathbb{R}^n \to \mathbb{R}$:

$$f(X) \subset F(c) + \sum_{i=1}^{n} \frac{\partial F}{\partial x_i}(X) \cdot (X_i - c_i), \tag{3.42}$$

donde $X = (X_1, X_2, \ldots, X_n) \in \mathbb{IR}(D)$, $c = (c_1, c_2, \ldots, c_n) \in X$ y $\frac{\partial F}{\partial x_i}$ es la extensión de intervalo para la i^{th} derivada parcial de f.

En este punto es importante aclarar que para la mayoría de los problemas, elegir el punto medio del intervalo como c no es la mejor opción, por lo que Baumann [34] dio expresiones analíticas de los centros óptimos a utilizar, c_B^- y c_B^+, con el fin de maximizar el límite inferior y minimizar el límite superior respectivamente. Esta variación se le conoce como **Optimal centered form**.

$$c_B^- = \begin{cases} \underline{X} & \text{Si } \ 0 \leq F'(X) \\ \overline{X} & \text{Si } \ 0 \geq F'(X) \\ \frac{U\underline{X} - L\overline{X}}{U - L} & \text{Cualquier otro caso} \end{cases} \tag{3.43}$$

$$c_B^+ = \begin{cases} \overline{X} & \text{Si } \ 0 \leq F'(X) \\ \underline{X} & \text{Si } \ 0 \geq F'(X) \\ \frac{U\overline{X} - L\underline{X}}{U - L} & \text{Cualquier otro caso} \end{cases} \tag{3.44}$$

Resulta importante recalcar que dado los centros óptimos; al no ser idénticos a los puntos medios del intervalo, la inclusión isotónica generada en el mean value form no está garantizada en el optimal centered form, no obstante, una propiedad ligeramente más débil conocida como isotonicidad unilateral si es válida [34], la cual resulta más que suficiente para una gran gama de aplicaciones.

3.6.2. Extensión basada en la monotonicidad

La existencia de monotonicidad local de una función con respecto a alguna de sus variables elimina el efecto de dependencia relacionado a estas variables y así lograr una mejor aproximación a la imagen exacta, incluso mayor que las extensiones de intervalo con convergencia lineal o cuadrática.

Diremos que para una función continua $f : D \subset \mathbb{R}^n \to \mathbb{R}$, con F la extensión de intervalo de f y $X \in \mathbb{IR}(D)$, presenta monotonicidad local, aumentando o disminuyendo localmente, para una variable x_i en X si, $\frac{\partial F}{\partial x_i}(X)$ es no negativa o no positiva respectivamente.

De esta manera la extensión basada en la monotonicidad, F_M, cumple que

$$f(X) \subset F_M(X) \subset F_N(X), \tag{3.45}$$

siempre y cuando se detecte f localmente monotónico en X con respecto a variables que tienen múltiples ocurrencias en su expresión. De esta manera podemos definir F_M como

$$F_M \ = \ [\underline{F_N(X^-)}, \ \overline{F_N(X^+)}], \tag{3.46}$$

donde $X^- = (X_1^-,\ X_2^-, \ldots, X_n^-)$ y $X^+ = (X_1^+,\ X_2^+, \ldots, X_n^+)$ denotan las cajas definidas por

$$X_i^- = \begin{cases} \underline{X_i} & \text{Si f incrementa con } x_i \\ \overline{X_i} & \text{Si f disminuye con } x_i \\ X_i & \text{Cualquier otro caso} \end{cases} \tag{3.47}$$

$$X_i^+ = \begin{cases} \overline{X_i} & \text{Si f incrementa con } x_i \\ \underline{X_i} & \text{Si f disminuye con } x_i \\ X_i & \text{Cualquier otro caso} \end{cases} \tag{3.48}$$

Como las variables con respecto a las cuales f es monótono se reemplazan por uno de sus límites en X^- y X^+, el problema de dependencia relacionado con estas variables desaparece en F_M.

Como ejemplo consideremos la función $f(X) = -x_1^2 + x_1x_2 + x_2x_3 - 3x_3$ sobre la caja $X = [6,\ 8] \times [2,\ 4] \times [7,\ 15]$. La extensión natural F_N viene dada por

$$F_N = -[6,\ 8]^2 + [6,\ 8][2,\ 4] + [2,\ 4][7,\ 15] - 3[7,\ 15] = [-83,\ 35]. \tag{3.49}$$

Ahora, la extensión natural de las derivadas parciales de f con respecto a cada variable x_i son

$$\frac{\partial F}{\partial x_1}(X) = -2X_1 + X_2 = [-14,\ -8] \leq 0, \tag{3.50}$$

$$\frac{\partial F}{\partial x_2}(X) = X_1 + X_3 = [13,\ 23] \geq 0, \tag{3.51}$$

$$\frac{\partial F}{\partial x_3}(X) = X_2 - 3 = [-1,\ 1], \tag{3.52}$$

de lo anterior podemos decir que f decrece con respecto a x_1, incrementa con respecto a x_2 y no es monotónica con respecto a x_3 en X. De lo anterior podemos definir que F_M es

$$F_M = [\underline{F_N(8,\ 2,\ [7,\ 15])},\ \overline{F_N(6,\ 4,\ [7,\ 15])}] = [-79,\ 27] \subset F_N(X). \tag{3.53}$$

CAPÍTULO 4

OPTIMIZACIÓN GLOBAL CON INTERVALOS

En este capítulo se introducen los métodos de optimización basados en aritmética de intervalos, que proporcionan una garantía matemática y computacional para encontrar los máximos y mínimos globales en funciones univariadas o multivariadas. El dominio original de las variables se divide sucesivamente, y los límites inferior y superior de la función se estiman en cada subregión. Al descartar subregiones donde la solución global no puede existir, siempre se puede encontrar la solución con cotas rigurosas.

4.1. Algoritmo Branch and Bound

Los algoritmos de búsqueda Branch-and-Bound (B&B) [36] son algoritmos de búsqueda global deterministas, en los que se divide repetitivamente el espacio de búsqueda y se descartan las partes que no contienen la solución óptima. Estos métodos van alternando un paso de filtrado y un paso de ramificación para explorar de forma exhaustiva el espacio siguiendo un árbol de búsqueda; en cada paso se divide el intervalo en subintervalos o ramas y se utilizan sus límites inferiores como criterio de descarte [1]. Habitualmente el criterio de detención es el tamaño de los dominios, esto es, cada rama es procesada hasta alcanzar un tamaño mínimo, ϵ-cajas; con esto, se desarrolla un método donde los límites superiores e inferiores convergen de manera garantizada al mínimo global.

El algoritmo prototípico de Branch-and-Bound en intervalos (IBB) es el conocido como el algoritmo de Skelboe–Moore [29]. Una implementación básica de éste se resume en el algoritmo 3. De manera general, el espacio de búsqueda se divide en cajas; las cuales se evalúan mediante extensiones de intervalo de la función objetivo. Si una caja no se puede

[1] Un intervalo cuyo límite inferior sea más grande al menor límite superior encontrado no puede contener un minimizador global para la función f.

descartar, se divide en subcajas las cuales se acomodan en una lista de prioridad[2] y se procesan en una etapa posterior.

Aunque el algoritmo Skelboe–Moore es simple y se recomienda en ciertos contextos, hoy en día es principalmente de importancia histórica en el contexto de la optimización global. Asimismo, los métodos de B&B funcionan bien para problemas de baja dimensión, pero no son prácticos para problemas de alta dimensión, ya que el número de subconjuntos crece de manera exponencial. Este problema está relacionado con el problema de dependencia dado propiamente por la base de la aritmética de intervalos, la cual ya fue introducida en la sección 3.5.

Algoritmo 3 Método Skelboe–Moore

```
1:  procedure SKELBOE–MOORE(f, X_0, ϵ)
2:      L_ub ← F(X_0)
3:      L ← X_0 inicializa la lista L
4:      while length(L) > 0 do
5:          (X_0, v_0) ← (X_i, v_i) donde v_i = min(v_j) para j = 1, 2...n en L
6:          if w(X_0) < ϵ then
7:              C ← X_0 entra X_0 en la lista C
8:          end if
9:          if L_ub < v_0 then
10:             Descarta X_0
11:         end if
12:         X_1, X_2 = bisect(X_0)
13:         L_ub ← min(F(X_1), F(X_2), L_ub)
14:         v1, v2 ← F(X_1), F(X_2)
15:         L ← (X_1, v1), (X_2, v2) ingrese los pares en orden
16:     end while
17: end procedure
```

En la actualidad diversos investigadores de cómputo de intervalos que trabajan en la optimización global han proporcionado muchas mejoras y extensiones sofisticadas al algoritmo Skelboe-Moore; mientras mantienen un rigor matemático completo en los resultados [37, 38, 39].

Cabe resaltar que la heurística juega un papel fundamental en la velocidad de convergencia de los algoritmos de Branch-and-Bound de intervalos y, en general, depende del problema. Esto se puede dividir en dos categorías principales, la heurística de exploración

[2]Una lista de prioridad es un tipo de datos abstracto similar a una cola en la que cada elemento tiene además una prioridad asociada; de esta manera se mantiene el orden entre elementos de manera ya sea ascendente o descendente dependiendo de su prioridad.

y la heurística de partición; es importante aclarar que la heurística de este método difiere a la heurística utilizada en métodos como Simulated Annealing, para el caso particular de Branch-and-Bound de intervalos siempre hay cotas rigurosas y todo el espacio de búsqueda es analizado.

4.1.1. Exploración del espacio de búsqueda

Uno de los principales propósitos en los algoritmos de IBB es el poder explorar de manera eficiente y rápida el espacio de búsqueda. Por este motivo, la estrategia de búsqueda empleada en cualquier tipo de problema deberá encontrar cajas representativas del conjunto de soluciones en las primeras etapas de la búsqueda. Esto dependerá del orden en el que se insertan y extraen las subcajas de la lista de prioridad.

Las estrategias de búsqueda más utilizadas son:

- **Best-first search:**, Esta estrategia promueve la exploración en los subespacios más prometedores, esto es, la subcaja con el $\underline{\mathrm{F}_{\mathrm{nat}}(\mathrm{X})}$ más bajo se extrae de la lista de prioridad.

- **Breadth first search:**, Promueve la exploración sistemática del espacio de búsqueda, examinando todas las posibles subcajas del espacio de manera uniforme, esto es, la subcaja con el mayor tamaño se extrae de la lista de prioridad.

- **Depth-first search:**, Promueve la exploración ordenada pero no uniforme de todo el espacio de búsqueda, explorando todas y cada una de las subcajas que va localizando, de forma recurrente. Esto es, explorar primero los subconjuntos más recientes.

Otras estrategias han sido empleadas con el fin de extenderse en todo el espacio búsqueda en las primeras etapas:

- **Most distant-first search** [40], para distribuir ϵ-cajas en el espacio de búsqueda, la búsqueda de una nueva ϵ-caja debe apuntar a maximizar la distancia con respecto a las ϵ-cajas encontradas hasta el momento. De esta manera buscamos que la siguiente caja a explorar sea la que maximice la función:

$$\mathrm{MinDis(X)} \;=\; \mathrm{mín}\{\mathrm{d(X,\ S_1)},\ \mathrm{d(X,\ S_2)}, \ldots, \mathrm{d(X,\ S_m)}\}, \tag{4.1}$$

donde $(S_1,\ S_2, \ldots,\ S_m)$ son las ϵ-cajas encontradas hasta el momento. La distancia d es una distancia arbitraria entre elementos del espacio de búsqueda dada por

$$d(X,\ Y)\ =\ máx\{d(x,\ y)\ con\ x \in X,\ y \in Y\}. \tag{4.2}$$

De esta manera cada elemento de en la lista de prioridad se organiza de manera decreciente de acuerdo con el MinDis, el cual se actualiza cada vez que una nueva ϵ-caja se encuentra. Un ejemplo de esto es considerar dos ϵ-cajas, S_1 y S_2, y dos cajas A y B faltantes por explorar, según se muestra en la figura 4.1. Evaluando el MinDis para cada una de las cajas faltantes tendremos

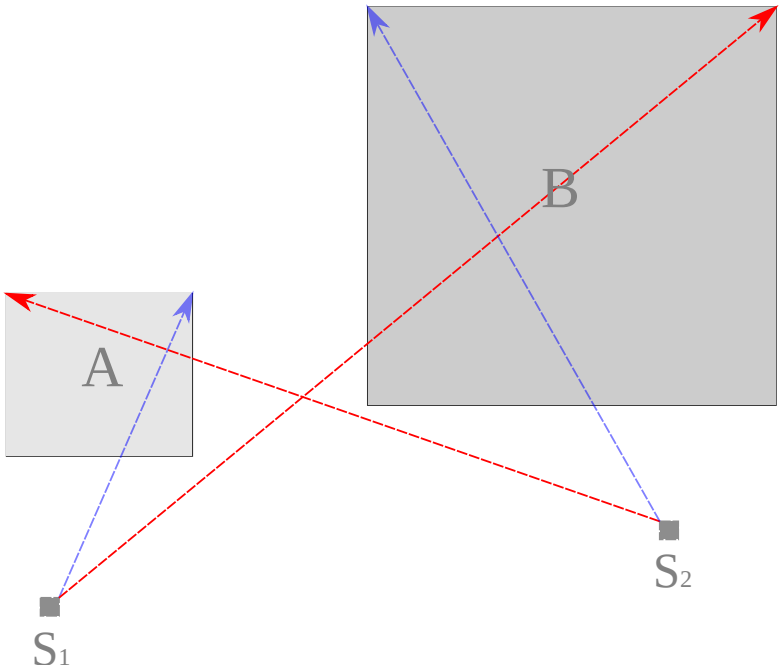

Figura 4.1: Ejemplo de evaluación de la función MinDis

$$MinDis(A)\ =\ mín\{d(A,\ S_1),\ d(A,\ S_2)\}\ =\ d(A,\ S_1), \tag{4.3}$$

$$MinDis(B)\ =\ mín\{d(B,\ S_1),\ d(B,\ S_2)\}\ =\ d(B,\ S_2), \tag{4.4}$$

donde $d(B,\ S_2) > d(A,\ S_1)$. De esta manera tendremos que la caja que maximiza la distancia respecto a las ϵ-cajas es la caja B; la cual sería la siguiente en explorar.

- **Depth and most distant-first search**, es un método híbrido entre Most distant-first search (MDFS) y Depth-first search (DFS) utilizado con el fin de mantener las ventajas de ambos enfoques. Al igual que el MDFS la lista de prioridad se organiza de acuerdo con la distancia a las ϵ-cajas encontradas hasta el momento y se actualiza cada vez que se encuentra una nueva; en cada paso las cajas de mayor profundidad son insertadas al inicio de la lista y se utiliza una heurística adicional para elegir entre cajas con la misma profundidad, aquella caja que maximice el MinDis se explora primero [40].

Como ejemplo consideremos el caso evidenciado en la Figura 4.2a, en el cual se ha encontrado una ϵ-caja, caja negra, y el problema tiene 3 soluciones restantes (x_1, x_2 y x_3). El cuadro de la derecha maximiza la función MinDis, línea punteada, por lo que se procesa este cuadro primero, bisecándolo y filtrándolo, obteniendo finalmente la Figura 4.2b.

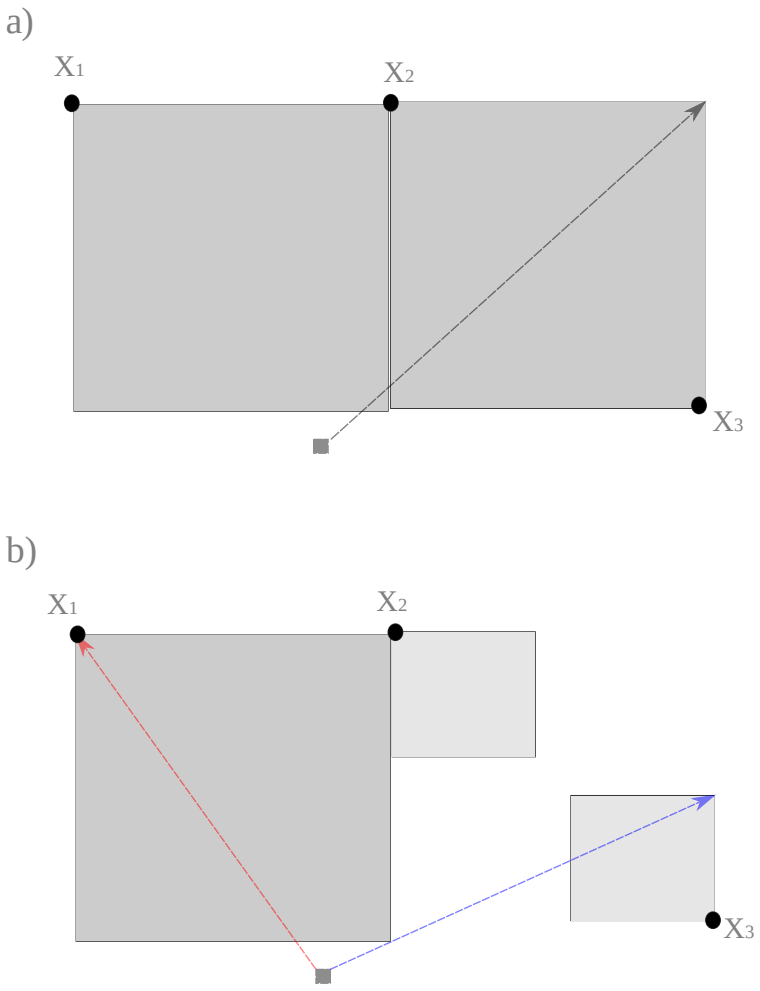

Figura 4.2: Diferencia entre la convergencia de MDFS y DMDFS

Hasta este punto tanto Most distant-first search como Depth and most distant-first search (DMDFS) actúan de la misma manera, sin embargo, en el siguiente paso de la búsqueda el MDFS procesa la caja que maximiza la distancia a la ϵ-caja entre las tres cajas existentes, es decir, la caja de la izquierda (línea roja) y eventualmente convergerá a x_1; mientras que el DMDFS procesa la caja que maximiza la distancia a la ϵ-caja entre las cajas de mayor profundidad (línea azul), y eventualmente convergerá a x_3.

4.1.2. Métodos de bisección

Históricamente se han propuesto varias estrategias generales de bisección donde la mayoría usa información local para seleccionar la siguiente variable o dirección a bisecar [41]; entre éstas se destacan dos heurísticas:

- La variable con el dominio más grande es particionada.
- Las variables se dividen una tras otra en una planificación Round-robin [42].

El objetivo principal es producir un fuerte impacto en el sistema y maximizar la posibilidad de que los nodos secundarios puedan ser descartados prontamente. En los últimos años los métodos basados en Smear [43, 44, 45] han mostrado ser competitivos con respecto a las 2 estrategias ya mencionadas. Estos métodos estiman el impacto de una variable en el sistema utilizando el límite superior para las derivadas parciales de la función y los anchos del dominio, ecuación (4.5).

$$\mathrm{Smear}(x_i, f) = \overline{|[a_i]| \cdot w(x_i)}, \tag{4.5}$$

donde $[a_i]$ es la extensión natural de intervalo de la derivada parcial de f con respecto a la variable x_i, $\frac{\partial f}{\partial x_i}$ y $w(x_i)$ es el ancho del intervalo x_i; la variable que maximiza este valor es seleccionada.

Como ejemplo consideremos la función, $f(x_1,\ x_2) = 10x_1^2 + 3x_2^2 - 5$ en la caja $[0,\ 10] \times [0,\ 10]$. Los valores de Smear para cada una de las variables serian

$$\mathrm{Smear}(x_1, f) = \overline{|20x_1| \cdot w(x_1)} = 2000, \tag{4.6}$$

$$\mathrm{Smear}(x_2, f) = \overline{|6x_2| \cdot w(x_2)} = 600, \tag{4.7}$$

con lo anterior se concluye que la variable a bisecar es x_1.

4.2. Técnicas de aceleración

Uno de los principales problemas en los algoritmos de IBB es la manera poco eficiente de descartar subcajas no óptimas en primeras etapas de la búsqueda o incluso ϵ-cajas al final del proceso, aun cuando éstas son muy pequeñas; esto debido a la imposibilidad de determinar con precisión dónde están los optimizadores globales.

Lo anterior se debe al hecho de que la función de inclusión utilizada no es lo suficientemente eficiente para producir rangos cercanos al rango real de la función, generando a su vez un aumento en el tiempo de ejecución del algoritmo. Varias metodologías como extensiones de segundo orden (Sección 3.6.1), pruebas de eliminación y métodos de contracción han sido consideras con el fin de limitar estos efectos negativos. Alternativas más actuales como affine arithmetic, la cual permite un cálculo más estrecho del rango de la función F(X) puede ser revisado en [46, 47, 48]

4.2.1. Pruebas de eliminación

4.2.1.1. Test del punto medio

El límite superior $\overline{F(X)}$ de una caja factible X garantiza ser un límite superior del mínimo global $f(x^*)$; una mejor descripción de éste límite permite una prueba de eliminación más eficiente[3][48, 49]. La evaluación de cualquier caja X para este fin puede ser mejorada si se evalúa en cualquier punto $c \in X$, ya que $f(x^*) \leq \overline{F(c)} \leq \overline{F(X)}$. **Ichida y Fujii** [37] propusieron la evaluación sistemática en el punto medio de la caja F(m(X)); en cada paso del algoritmo la función F se evalúa en el centro de una subcaja X considerada y se compara con la mejor solución actual, esto es

$$\overline{f} \leftarrow \text{mín}(\overline{f},\ \overline{F(m(X))}) \tag{4.8}$$

Para ejemplificar lo anterior, consideremos la función $f(x) = x^2\cos(x) + x$ sobre el intervalo $X = [-5,\ 3]$.

[3]Sea $\overline{f}$ el mejor límite superior encontrado hasta el momento. Si para un caja X se tiene que $\overline{f} < \underline{F(X)}$ la caja es descartada ya que no puede contener el minimizador global.

La caja X se divide generando las subcajas $X_1 = [-5, -1]$ y $X_2 = [-1, 3]$. Procesando la caja X_1 tendremos que

$$F(X_1) = [-5, -1]^2 cos([-5, -1]) + [-5, -1] = [-5, -1] = [-30, 12.5076], \quad (4.9)$$

obteniendo el límite superior para el mínimo global de la función, encontrado por el momento, correspondiente a 12.5076. Ahora si consideramos evaluar la caja X_1 en el punto medio tendremos que:

$$F(m(X_1)) = [-3, -3]^2 cos([-3, -3]) + [-3, -3] = [-11.91, -11.9099] \quad (4.10)$$

logrando así una mejor descripción de la cota superior del mínimo global, ahora con un valor de -11.9099.

4.2.2. Métodos de contracción

Actualmente, los métodos IBB utilizan procedimientos de contracción, con el fin de poder reducir los dominios de las cajas en el espacio de búsqueda, eliminando aquellos valores que no satisfacen una o más restricciones (por ejemplo, valores mayores al menor límite superior encontrado). Estos procesos de contracción no pierden soluciones, es decir, se implementan de manera rigurosa y no implican heurísticas que puedan ignorar las soluciones.

4.2.2.1. Interval Newton

Uno de los algoritmos de filtrado más efectivos utilizados en el análisis de intervalos es la extensión del método de Newton a intervalos (Interval Newton) [50]. El objetivo del método clásico de Newton es encontrar sucesivamente mejores aproximaciones a los ceros de una función real. Consideremos una función univariada continua y diferenciable f(x) y su derivada f′(x). Si x_l es una aproximación de una solución de la ecuación f(x) = 0, entonces se obtiene una mejor aproximación mediante

$$x_{l+1} = x_l - \frac{f(x_l)}{f'(x_l)}. \quad (4.11)$$

Ahora consideremos la generalización del método clásico a intervalos. Para esto tomemos una función univariada f(x) y un intervalo inicial X. El método aplica sucesivamente el paso de contracción

$$X_{l+1} \leftarrow X_l \cap \left(c - \frac{f(c)}{F'(X_l)} \right), \tag{4.12}$$

donde c corresponde al punto medio del intervalo X_l, $c = m(X_l)$. Los factores dentro del paréntesis corresponden al operador de newton dado por

$$N_f(X,\ c) \ = \ c - \frac{f(c)}{F'(X)}. \tag{4.13}$$

Si el procedimiento converge a un punto fijo, $X_{l+1} \subset X_l$, devolverá un intervalo con una tolerancia ϵ que contiene una única solución de la ecuación $f(x) = 0$ en X. Si una iteración del paso de contracción devuelve un intervalo vacío, $X_{l+1} = \emptyset$, entonces no hay solución en X. Cuando las iteraciones no convergen a un punto fijo, no podemos predecir la presencia o ausencia de soluciones en X y la caja por en ende sera dividida [29, 32].

Por ejemplo, consideremos la función $f(x) = x^2 - 2$ y su derivada $f'(x) = 2x$. Se debe encontrar los ceros de la función en el intervalo inicial de $X = [-3,\ 2]$ y escogiendo $c = m(X) = -0.5$[4]. La primera iteración de Newton seria

$$N_f(X,\ c) \ = \ c - \frac{F(c)}{F'(X)} \ = \ -0.5 - \frac{(-0.5^2) - 2}{2[-3,\ 2]} \ = \ \left[-\infty,\ -\frac{19}{24}\right] \cup \left[-\frac{1}{16},\ \infty\right]. \tag{4.14}$$

Ahora la intersección con el intervalo inicial X sera

$$X \cap N_f(X,\ c) \ = \ \left[-3,\ -\frac{19}{24}\right] \cup \left[-\frac{1}{16},\ 2\right]. \tag{4.15}$$

Es de notar que la intersección se compone de 2 subintervalos, $X_1 = \left[-3,\ -\frac{19}{24}\right]$ y $X_4 = \left[-\frac{1}{16},\ 2\right]$, a los cuales se les tiene que aplicar el método de newton recursivamente. Los resultados de las siguientes dos iteraciones para cada intervalo se resumen en la Tabla 4.1.

[4] -0.5 corresponde al intervalo degenerado [-0.5, 0.5]

k	X_k	$F'(X_k)$	$N_f(X_k,\ c_k)$	$X_k \cup N_f(X_k,\ c_k)$
1	$[-3,\ -0.792]$	$[-6,\ -1.583]$	$[-1.630,\ -0.889]$	$X_2 = [-1.630,\ -0.889]$
2	$[-1.630,\ -0.889]$	$[-3.260,\ -1.778]$	$[-1.492,\ -1.386]$	$X_3 = [-1.492,\ -1.386]$
4	$[0.063,\ 2]$	$[-0.125,\ 4]$	$[-\infty,\ -7.523] \cup [1.234,\ \infty]$	$X_5 = [1.234,\ 2]$
5	$[1.234,\ 2]$	$[3.468,\ 4]$	$[1.368,\ 1.463]$	$X_6 = [1.368,\ 1.463]$

Tabla 4.1: Interval Newton para la función $f(x) = x^2 - 2$ en el intervalo inicial $X = [-3,\ 2]$

Los dos ceros posibles de f en X están delimitados por los intervalos $[-1.492,\ -1.386]$ y $[1.368,\ 1.463]$ respectivamente. Su existencia es garantizada durante las iteraciones de $k = 1$ y $k = 5$ ya que $X_2 \subset X_1$ y $X_6 \subset X_5$.

Este método puede ser extendido al caso multivariado de n ecuaciones con n incógnitas. Sea entonces $f : \mathbb{R}^n \to \mathbb{R}^n$ una función diferenciable, X una caja en el espacio y $x_k \in X$ un punto. La ecuación de $f(x) = 0$ puede ser linealizada en el punto x_k usando mean value form como

$$f(x_k) + J_f(X)(X - x_k) = 0, \tag{4.16}$$

donde J_f es la extensión de intervalo del Jacobiano de f, matriz de las primeras derivadas parciales de la función, sobre X. Teniendo que $Y_k = X - x_k$, tenemos que

$$Y_k \ = \ -J_f(X)^{-1} f(x_k). \tag{4.17}$$

En la práctica, Y_k se obtiene resolviendo una versión preacondicionada del sistema lineal de intervalos

$$J_f(X)Y_k \ = \ -f(x_k) \tag{4.18}$$

4.2.2.2. Algoritmos de restricción–propagación

Estos algoritmos se centran en la reducción del dominio con respecto a una ecuación y/o restricción y se realizan mediante un procedimiento de revisión. Luego se utiliza un algoritmo de propagación, similar al conocido algoritmo **AC3** [51], para propagar las reducciones a todo el sistema hasta que se alcanza un punto fijo.

Todos los algoritmos de propagación basados en intervalos tienen procedimientos de propagación similares a AC3, que difieren principalmente en el procedimiento de revisión. En general, los procedimientos de revisión intentan imponer consistencias parciales sobre los elementos de los dominios, de tal manera que deberían encontrar el intervalo "mínimo" para una variable xi que contenga todos los valores consistentes relacionados con una restricción C en una caja X. Sin embargo, debido principalmente al problema de dependencia, generalmente no es posible implementar dichos métodos de manera eficiente. Una forma intuitiva de filtrar los dominios de las variables es aislar la variable que queremos filtrar y calcular la imagen de la función generada. El resultado se debe intersecar con el dominio inicial.

Como ejemplo consideremos entonces la ecuación $x_1 + 2x_2 = 10$, con los dominios de $X_1 = [0,\ 5]$ y $X_2 = [1,\ 3]$. Si queremos reducir el dominio de X_1, debemos aislarla la variable de la ecuación de la siguiente manera

$$X_1 \ = \ 10 - 2X_2. \tag{4.19}$$

Utilizando alguna extensión de intervalo (por ejemplo, la extensión de intervalo natural), se puede calcular el rango de $F(X_1)$. El nuevo dominio será la intersección de la proyección generada y el intervalo inicial, esto es

$$X_1 \leftarrow X_1 \cap (10 - 2X_2) \ = \ [0,\ 5] \cap [4,\ 8] \ = \ [4,\ 5]. \tag{4.20}$$

Tengamos en cuenta que la imagen de la función de proyección se calcula utilizando extensiones de intervalo, por lo cual este método también sufre el problema de dependencia. De igual manera, otro problema ocurre cuando una variable aparece varias veces en una restricción, y el aislamiento es generalmente imposible. En este caso, cada ocurrencia de la variable se puede aislar de forma independiente.

Como ejemplo consideremos la ecuación $x^3 + 3x = 14$, en el dominio $X = [0,\ 3]$. Para

reducir el dominio de la variable, podríamos considerar las funciones de proyección de cada ocurrencia e intersecar sus aproximaciones de imagen con el intervalo inicial X;

$$X \leftarrow X \cap (-3X + 14)^{\frac{1}{3}} = [0,\ 3] \cap [1.71,\ 2.41] = [1.71,\ 2.41]. \tag{4.21}$$

$$X \leftarrow X \cap \frac{-X^3 + 14}{3} = [0,\ 3] \cap [-4.33,\ 4.67] = [0,\ 3]. \tag{4.22}$$

De lo cual se obtiene que la ahora $X = [1.71,\ 2.41]$. Después de varias iteraciones, X converge al intervalo $[2,\ 2]$.

Sin embargo, si las funciones de proyección tienen varias ocurrencias de la variable a proyectar (es decir, la variable aparece más de dos veces en la restricción), debido al problema de dependencia, la proyección convergerá a dominios subóptimos; este problema particular se conoce como el problema de aislamiento.

Uno de los algoritmos más utilizados es el algoritmo **HC4-Revise** [52], el cual filtra los dominios de las variables sin generar explícitamente las funciones de proyección. HC4-Revise requiere atravesar la expresión solo dos veces para realizar la proyección sobre cada aparición de una variable. El algoritmo funciona en dos fases, la fase de hacia adelante, *forward phase*, en la que se calcula, utilizando la aritmética de intervalos, la inclusión natural de las subexpresiones representadas por cada una de las operaciones existentes en la función, denominadas nodos, y la fase hacia atrás atraviesa, *backward phase*, aplicando en cada nodo un operador de proyección relacionado. El operador de proyección reduce los intervalos de los nodos, eliminando valores inconsistentes con respecto al operador básico unario o binario correspondiente.

Como ejemplo, consideremos entonces la ecuación $x^2 - 3x + y = z$, con los dominios iniciales de $X = [4,\ 10]$, $Y = [-80,\ 30]$ y $Z = [0,\ 0]$. Los nodos en la fase hacia adelante se muestran en las ecuaciones (4.23) - (4.26), una representación esquemática de esta fase se muestra en la Figura 4.3.

$$N_1 := X^2 = [4,\ 10]^2 = [16,\ 100]. \tag{4.23}$$

$$N_2 := 3X = 3[4,\ 10] = [12,\ 30]. \tag{4.24}$$

$$N_3 := N_1 - N_2 = [16,\ 100] - [12,\ 30] = [-14,\ 88]. \tag{4.25}$$

$$N_4 := N_3 + Y = [-14,\ 88] + [-80,\ 30] = [-94,\ 118]. \tag{4.26}$$

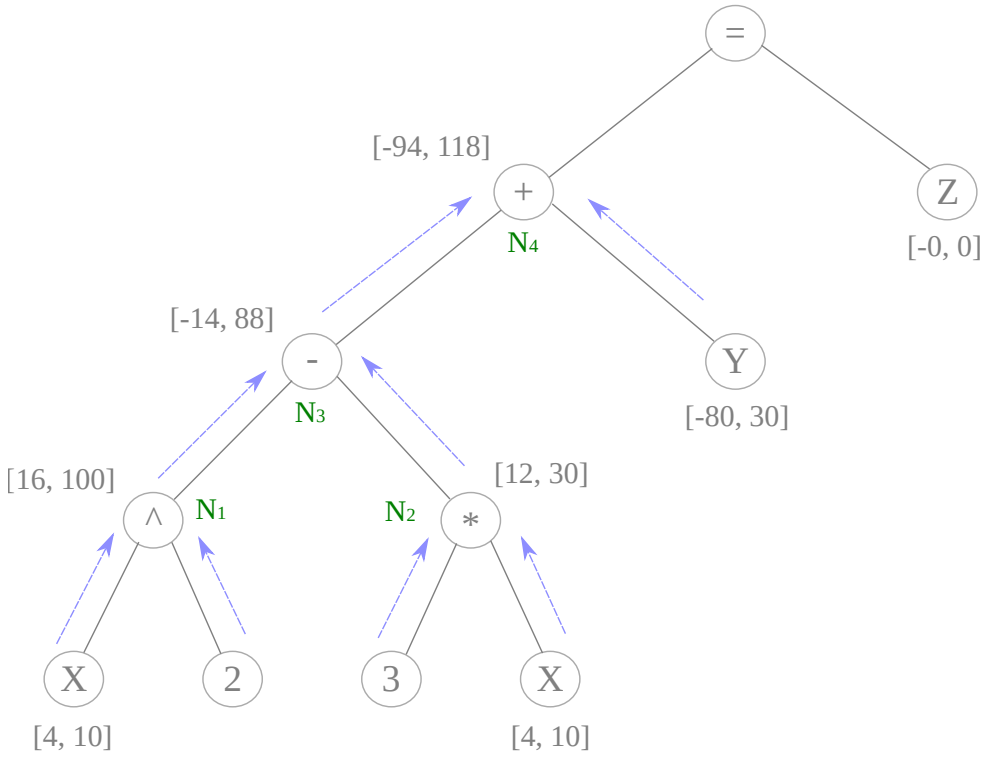

Figura 4.3: HC4-Revise: fase de propagación hacia adelante

La fase de propagación hacia atrás, Figura 4.4, comienza con la intersección de los rangos de los lados izquierdo (N4) y derecho (Z). Luego propaga la restricción hacia abajo, evaluando las funciones de proyección en cada nodo, de la siguiente manera:

$$N_4^* = N_4 \cap Z = [-94,\ 118] \cap [0,\ 0] = [0,\ 0]. \tag{4.27}$$

$$Y^* = Y \cap (N_4^* - N_3) = [-80,\ 30] \cap ([0,\ 0] - [-14,\ 88]) = [-80,\ 14]. \tag{4.28}$$

$$N_3^* = N_3 \cap (N_4^* - Y) = [-14,\ 88] \cap ([0,\ 0] - [-80,\ 30]) = [-14,\ 80]. \tag{4.29}$$

$$N_2^* = N_2 \cap (N_1 - N_3^*) = [12,\ 30] \cap ([16,\ 100] - [-14,\ 80]) = [12,\ 30]. \tag{4.30}$$

$$N_1^* = N_1 \cap (N_3^* + N_2) = [16,\ 100] \cap ([-14,\ 80] + [12,\ 30]) = [16,\ 100], \tag{4.31}$$

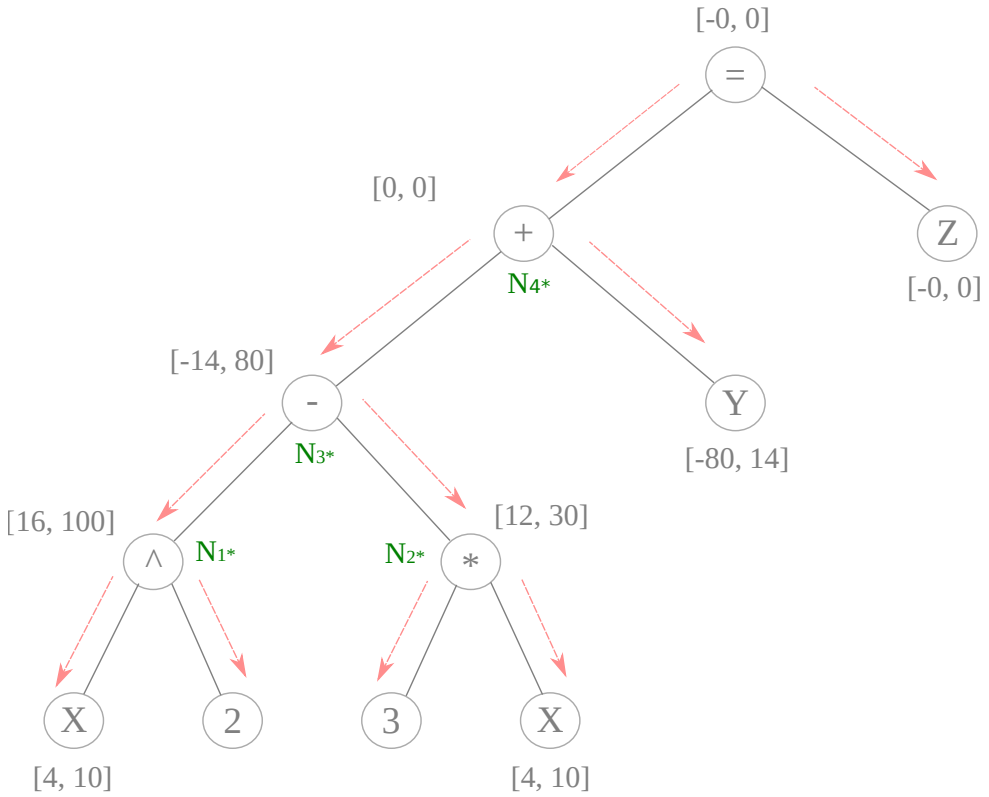

Figura 4.4: HC4-Revise: fase de propagación hacia atrás

con lo anterior, aunque se logra un estrechamiento en la variable $Y = [-80,\ 14]$, pero no permite una reducción en la variable $X = [4,\ 10]$ la cual no se modifica.

Es importante aclarar que HC4-Revise realiza una proyección automática sobre los intervalos, utilizando una inclusión natural. Trabajos recientes describen una variante de HC4-Revise llamada FBBT [53], la cual utiliza una extensión basada en monotonicidad para realizar el filtrado. Otros métodos pueden ser revisados en [32, 52].

CAPÍTULO 5

PROBLEMAS Y RESULTADOS

5.1. Propuesta

Se unificaron las distintas técnicas de búsqueda, eliminación y contracción en un algoritmo de IBB, en un lenguaje de programación dinámico y rápido como lo es Julia [2] y hecho de código abierto. Éste compone un repositorio el cual puede ser revisado en [1].

La eficiencia de este algoritmo en el ámbito de la química computacional, se evaluó conforme a los problemas presentados por Lavor [54], en la optimización de estructuras moleculares y Vanaret [32], en la optimización de clústeres de Lennard–Jones, específicamente en el clúster compuesto de 5 esferas. Una explicación más detallada del código es presentada en el Anexo 1

5.2. Optimización de estructuras moleculares

Se consideró el modelo molecular simplificado estudiado por Lavor [54, 55], con el fin de tener un punto de comparación respecto al rendimiento de nuestra metodología y la presentada por Lavor.

Como lo sugerido por Lavor, se considera una cadena lineal de N esferas, átomos unidos, centrados en $x_1,\ x_2, \ldots,\ x_N$ en un espacio tridimensional. La energía potencial del sistema está dada por

$$V \ = \ V_1 + V_2 + V_3 + V_4, \tag{5.1}$$

donde V_1 corresponde a la energía potencial debida al estiramiento de enlace entre cada par de esferas consecutivas:

$$V_1 = \sum_{(i,j)\in M_1} c_{ij}^{(1)}(r_{ij} - r_{ij}^0)^2, \tag{5.2}$$

con $c_{ij}^{(1)}$ la constante de fuerza de estiramiento de enlace y r_{ij}^0, la longitud de enlace de equilibrio. V_2, es la contribución debido a la flexión del ángulo de enlace cada tres esferas consecutivas,

$$V_2 = \sum_{(i,j)\in M_2} c_{ij}^{(2)}(\theta_{ij} - \theta_{ij}^0)^2, \tag{5.3}$$

con $c_{ij}^{(2)}$ la constante de fuerza de flexión del ángulo y θ_{ij}^0, el ángulo de enlace de equilibrio; V_3 es la contribución debido al ángulo diédrico sobre cada cuatro esferas consecutivas:

$$V_3 = \sum_{(i,j)\in M3} c_{ij}^{(3)}(1 + \cos(3\omega_{ij} - \omega_{ij}^0)), \tag{5.4}$$

con $c_{ij}^{(3)}$ la constante de fuerza de torsión y ω_{ij}^0, el ángulo de equilibrio. V_4 es el potencial que caracteriza las interacciones de dos cuerpos, entre cada par de cuentas separadas por más de tres enlaces covalentes, a lo largo de la cadena. Aquí, solo se consideran las interacciones entre pares separados exactamente por tres enlaces covalentes, y el modelo utilizado es

$$V_4 = \sum_{(i,j)\in M3} \frac{(-1)^i}{r_{ij}}, \tag{5.5}$$

con r_{ij} la distancia que separa las esferas i y j. La notación Mi, con $i = 1, 2, 3$, indica los conjuntos de pares de esferas separadas por enlaces covalentes. En este problema, todas las longitudes de enlace y ángulos de enlace han sido fijados en los valores de equilibrio $r_{ij}^0 = 1.526$Å y $\theta_{ij}^0 = 1.91$, valores característicos de una cadena de hidrocarburos. También, los valores de $c_{ij}^{(3)} = 1$ y $\omega_{ij}^0 = 0$ fueron establecidos.

Por las consideraciones anteriores, la función de energía potencial queda finalmente de la forma

$$V = V3 + V4 = \sum_{(i,j)\in M3} (1 + \cos(3\omega_{ij})) + \frac{(-1)^i}{r_{ij}}, \tag{5.6}$$

donde r_{ij} puede ser expresada en función del ángulo diedro como

$$r_{ij} = \sqrt{0.60099896 - 4.14720682 \cos(\omega_{ij})} \tag{5.7}$$

El problema es entonces encontrar el mínimo global de esta función potencial, que ahora es una función solo de los ángulos diedros ω_{ij}. Los intervalos iniciales para la búsqueda se establecen como $\omega_{ij} = [0, 5]$. Este modelo molecular simplificado es un muy buen problema de prueba gracias al mínimo global conocido [54] y al gran número de mínimos locales que están presentes[1].

La Tabla 5.1, resume los resultados obtenidos con el algoritmo propuesto y la metodología utilizada por Lavor [54] y Lin [55]. En cada caso se realizó una exploración del espacio con el método de búsqueda de Best First Seach, igual que los otros autores. El proceso se llevó en un procesador Intel Core i7-8550U 1.80GHz, a una tolerancia $\epsilon = 10^{-5}$.

N	Exploración	Min. Global	Método			Lin [55]		Lavor [54]
			a	b	c	a*	b*	
5	BFS	-0.082236	0.0012	0.0009	0.0037	0.0009	0.003	-
10	BFS	-0.589388	0.09	0.11	0.05	0.02	0.05	2.47
15	BFS	-0.493418	8.55	9.22	0.18	0.16	0.55	34.56
20	BFS	-1.000569	1013.37	877.51	0.471	1.53	9.89	1167
25	BFS	-0.9045997	>3600	2685	1.27	8.31	168.5	34657
30	BFS	-0.9045997	>3600	>3600	5.79	76.02	>3600	-

Tabla 5.1: Resultados de la optimización de las cadenas de hidrocarburos de 5 a 30 esferas utilizando el método Best-first search (BFS) a una tolerancia $\epsilon = 10^{-5}$. a = Inclusión de intervalo natural + Optimización local. b = Mean Value Form + Optimización local. c = Mean Value Form + Optimización local + Restricción de la forma HC4-revise, a* = Interval Newton + Optimización local + Expansión de Taylor de Primer Orden como función de Inclusión. b* = Optimización local + Expansión de Taylor de Primer Orden.

[1] El número de mínimos locales viene dado por 2^{n-3} con n el número de átomos o esferas en la cadena.

5.3. Clúster de Lennard–Jones

En esta sección se realiza la prueba rigurosa de optimización para los clústeres de Lennard–Jones de 2 a 5 átomos. Hasta la fecha, la única prueba rigurosa fue realizada por Vanaret [32]; demostrando que la solución más conocida, que nunca había sido certificada numéricamente, es realmente la óptima global.

El potencial de Lennard–Jones descrito en la ecuación (2.1), generalmente dada en unidades reducidas $\sigma = 1$ y $\epsilon = 1$, contiene dos ocurrencias de la variable r. Un truco para reducir la dependencia es completar el cuadrado, obteniendo

$$V = \left[4\left(\frac{1}{r^6} - \frac{1}{2}\right)^2 - 1\right], \tag{5.8}$$

de esta forma, la función de inclusión natural del potencial V se vuelve óptima con respecto a las ocurrencias de r. Sin embargo, se debe tener en cuenta que la función objetivo, ecuación (5.8), todavía sufre dependencia, ya que las coordenadas $(x_i,\ y_i,\ z_i)$ de los átomos tienen múltiples ocurrencias en los términos de distancia r_{ij}.

Una reducción de la simetría se puede lograr fijando algunos la posición de algunos átomos y así mismo reduciendo el espacio de búsqueda. Si se considera la primera partícula en el origen de coordenadas, el resto de las partículas se ubicarían con respecto a ésta en

$$\begin{cases} 1 & x_1 = y_1 = z_1 = 0 \\ 2 & x_2 > 0,\ y_2 = z_2 = 0 \\ 3 & x_3 > 0,\ y_3 > 0,\ z_3 = 0 \\ 4 & x_4 > 0,\ y_i > 0,\ z_4 > 0 \\ \geq 5 & x_i \neq 0,\ y_i \neq 0,\ z_i \neq 0 \end{cases} \tag{5.9}$$

Con lo anterior se realizaron los cálculos para los clúster de 2 a 4 átomos inicialmente en el intervalo de [0, 1.2] para cada variable. Los resultados obtenidos con los distintos métodos aplicados se resumen en la Tabla 5.3.

N	Exploración	Método	Mínimo Global	T.F.L.	Iter.	CPU (s)
2	DFS	S	-1.00000	1	29	0.00018
	BrFS	S	-1.00000	1	29	0.00014
	BFS	S	-1.00000	1	29	0.00018
	MDFS	S	-1.00000	1	29	0.00030
	DMDFS	S	-1.00000	1	29	0.00023
3	DFS	S	-3.00000	3	265	0.0038
		MVF	-3.00000	3	265	0.0046
		C	-3.00000	1	25	0.0005
		C + MVF	-3.00000	1	25	0.0005
	BrFS	S	-3.00000	3	263	0.0054
		MVF	-3.00000	3	263	0.0058
		C	-3.00000	1	27	0.0005
		C + MVF	-3.00000	1	27	0.0005
	BFS	S	-3.00000	3	245	0.0042
		MVF	-3.00000	3	245	0.0035
		C	-3.00000	1	27	0.0005
		C + MVF	-3.00000	1	27	0.0010
	MDFS	S	-3.00000	3	245	0.0064
		MVF	-3.00000	3	245	0.0064
		C	-3.00000	1	27	0.0008
		C + MVF	-3.00000	1	27	0.0008

N	Exploración	Método	Mínimo Global	T.F.L.	Iter.	CPU (s)
3	DMDFS	S	-3.00000	3	245	0.004
		MVF	-3.00000	3	245	0.004
		C	-3.00000	1	27	0.0009
		C + MVF	-3.00000	1	27	0.0010
4	DFS	S	-6.00000	30	6185	0.33
		MVF	-6.00000	30	6185	0.33
		C	-6.00000	3	203	0.011
		C + MVF	-6.00000	3	203	0.11
	BrFS	S	-6.00000	30	6737	0.34
		MVF	-6.00000	30	6737	0.35
		C	-6.00000	4	145	0.007
		C + MVF	-6.00000	4	145	0.007
	BFS	S	-6.00000	30	5143	0.27
		MVF	-6.00000	30	5143	0.30
		C	-6.00000	2	125	0.009
		C + MVF	-6.00000	2	125	0.01
	MDFS	S	-6.00000	30	5181	0.22
		MVF	-6.00000	30	5181	0.24
		C	-6.00000	2	125	0.006
		C + MVF	-6.00000	2	125	0.008

N	Exploración	Método	Mínimo Global	T.F.L.	Iter.	CPU (s)
4	DMDFS	S	-6.00000	30	5181	0.22
		MVF	-6.00000	30	5181	0.22
		C	-6.00000	2	125	0.006
		C + MVF	-6.00000	2	125	0.007

Tabla 5.2: Resultados de la optimización de los sistemas de Lennard Jones de 2 a 4 átomos, utilizando los métodos de búsqueda, Depth-first search (DFS), Breadth-first search (BrFS), Best-first search (BFS), Most distant-first search (MDFS) y Depth Most distant first-search (DMDFS). Los intervalos iniciales de búsqueda para todas las variables corresponden a [0, 1.2]. La tolerancia utilizada fue de $\epsilon = 10^{-4}$. S = Solo se utiliza inclusión natural. MVF = Se utiliza el mean value form como inclusión. C = Se utiliza restricciones de la forma HC4-revise. En todos los casos se combina un método de optimización local basado en descenso de gradiente.

El problema de interés es la optimización del clúster de LJ de 5 partículas. Este problema tiene el principal inconveniente que, debido a la cantidad de mínimos presentes en su superficie de energía potencial y la cantidad de ocurrencias de cada variable, impide el rechazo de manera óptima de las cajas.

Se sugirió la generación de puntos aleatorios en la caja para encontrar de una manera rápida un mejor punto óptimo para la evaluación del mean value form y la optimización local. El evaluar el tamaño final de la lista es un indicio de la eficiencia del descarte de cajas que no contienen el minimizador global durante el proceso. La Figura 5.1a, resume el tamaño de la lista final en función a la cantidad de puntos aleatorios generados, la Figura 5.1b, resume el tiempo de ejecución total del algoritmo en función a los puntos aleatorios generados.

La obtención de un buen límite superior para el mínimo global en las primeras etapas de la búsqueda puede indicar una disminución en el tiempo de ejecución del algoritmo; la Figura 5.2, resume la actualización del mínimo encontrado en función al número de iteraciones para cada método de búsqueda.

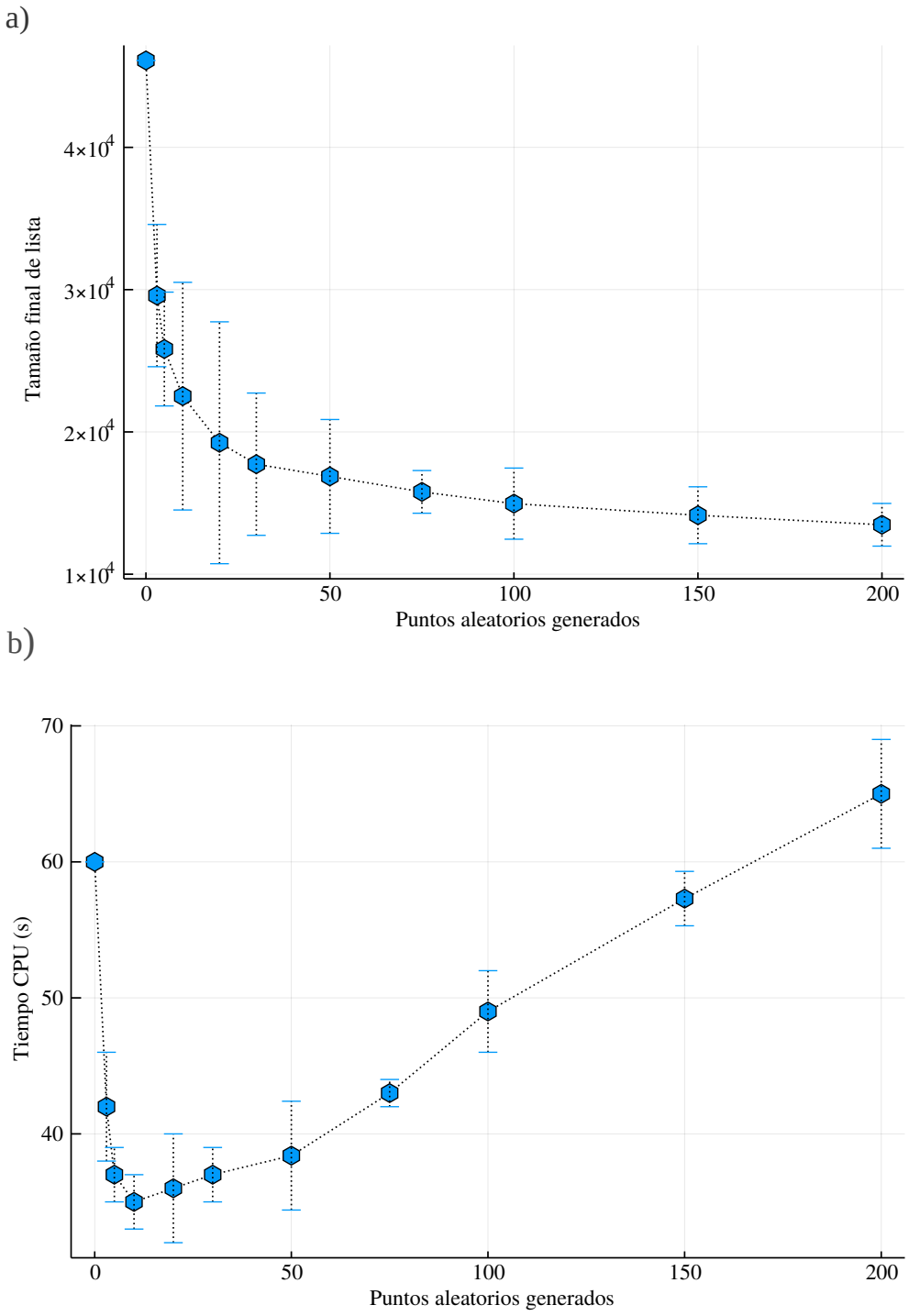

Figura 5.1: Gráfica de a) tamaño de la lista final vs número puntos aleatorios generados y b) Tiempo de ejecución del algoritmo vs número de puntos aleatorios generados; para el clúster de Lennard–Jones de 5 átomos, en los intervalos iniciales de [0, 1.2] para cuatro átomos y [-1.2, 1-2] para el quinto átomo, a una tolerancia de $\epsilon = 10^{-2}$, con un método de búsqueda de Best-first search, utilizando restricciones y método de optimización local.

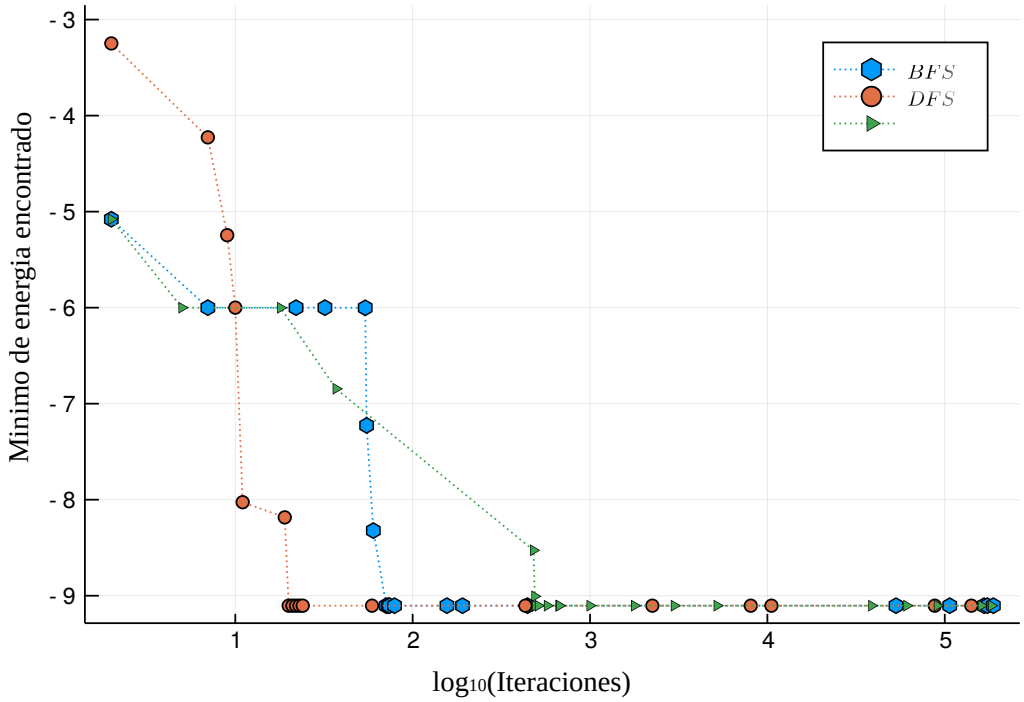

Figura 5.2: Mínimo encontrado en función al logaritmo del número de iteraciones para los métodos de búsqueda de Depth-first search (DFS), Breadth-first search (BrFS) y Best-first search (BFS), para el clúster de Lennard–Jones de 5 átomos en los intervalos iniciales de [0, 1.2] para cuatro átomos y [-1.2, 1-2] para el quinto átomo, a una tolerancia de $\epsilon = 10^{-2}$.

Para los cálculos del clúster de Lennard–Jones de 5 átomos, se consideraron los intervalos iniciales de [0, 1.2] para cada variable de cuatro átomos y un intervalo inicial de [-1.2, 1.2] para las variables del quinto átomo. Los resultados obtenidos se resumen en la Tabla 5.3. Las geometrías finales obtenidas para cada caso se evidencian en la Figura 5.3.

N	Exploración	Método	Mínimo Global	T.F.L.	Iter.	CPU (s)
	DFS	MVF + C	-9.10385	199293	2689433	381
		MVF + C + P.A.	-9.10385	199070	1961361	506
	BrFS	MVF + C	9.10385	210914	2415159	362
		MVF + C + P.A.	-9.10385	177586	2225225	489

N	Exploración	Método	Mínimo Global	T.F.L.	Iter.	CPU (s)
5	BFS	MVF + C	-9.10385	204982	2443277	393
		MVF + C + P.A.	-9.10385	183961	2248949	509
	MDFS	MVF + C	-9.10385	118394	2017263	>3600
	DMDFS	MVF + C	-9.10385	137839	2091328	>3600

Tabla 5.3: Resultados de la optimización del sistema de Lennard–Jones de 5 átomos, utilizando los métodos de búsqueda, Depth-first search (DFS), Breadth-first search (BrFS), Best-first search (BFS), Most distant-first search (MDFS) y Depth Most distant-first search (DMDFS). Los intervalos iniciales de búsqueda para todas las variables de los primeros cuatro átomos corresponden a [0, 1.2], para el quinto átomo las variables están en el intervalo de búsqueda de [-1.2, 1.2]. La tolerancia fue de $\epsilon = 10^{-3}$. MVF = Se utiliza el mean value form como inclusión. C = Se utiliza restricción de la forma HC4-revise, P.A. = generación de punto aleatorios.

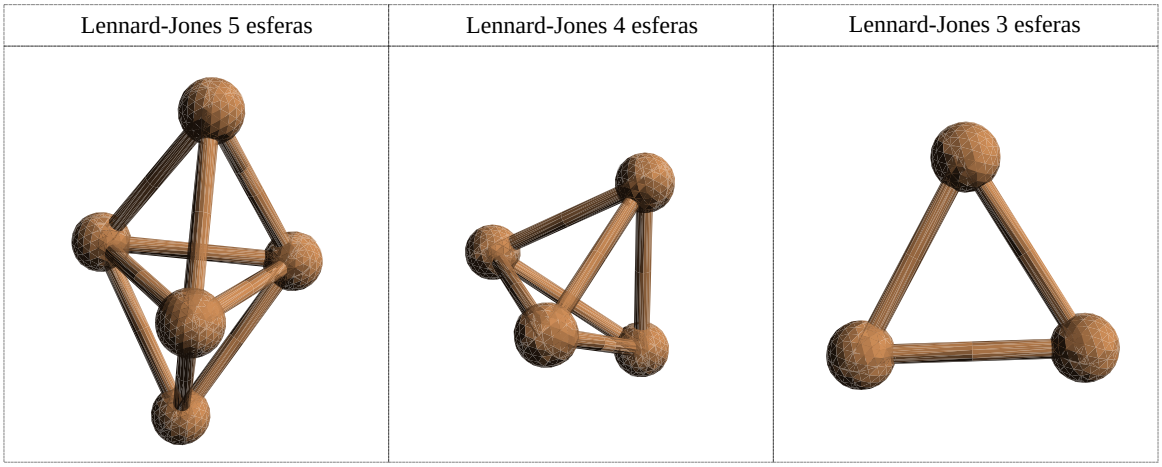

Figura 5.3: Geometrías obtenidas para los clúster de Lennard-Jones de 3 a 5 esferas.

CAPÍTULO 6

CONCLUSIONES Y PERSPECTIVAS

El trabajo unifico distintas metodologías y técnicas de aceleración en la base de un algoritmo prototipo de IBB (algoritmo Skelboe–Moore), enfocado en la optimización global de funciones de manera rigurosa y garantizada. Se evidencio que la principal técnica de aceleración fueron los métodos de contracción, generando una reducción considerable en el tiempo de ejecución y en el tamaño final de lista para cada caso estudiado.

Emplear métodos de optimización local, permitió la obtención de mejores cotas superiores para el mínimo global en las primeras etapas del algoritmo, además de ofrecer una mejora en la eficiencia de descarte de subcajas que no contienen optimizadores globales, generando de igual manera una disminución en el número de iteraciones generadas para cada método de exploración en todos los casos de estudio.

Uno de los principales inconvenientes que se evidencio, es la sobrestimaron del rango de la función, aun cuando se utilizan métodos de inclusión de segundo orden y las cajas a evaluar son pequeñas. EL emplear números aleatorios para la obtención de un mejor centro para la evaluación del mean value form, produce una disminución en el tamaño final de la lista, con el inconveniente de presentar un aumento considerable en el tiempo de ejecución del algoritmo.

Una ventaja de este tipo de algoritmos es la posibilidad de ser paralelizados de manera eficiente, se propone, para futuros trabajos, el paralelizar estos procesos para obtener una mejora en el rendimiento global del algoritmo, al igual que emplear métodos como affine arithmetic para una mejor descripción del rango de la función. Se cree que, combinado ambos métodos descritos anteriormente en el algoritmo propuesto, se puede obtener una metodología que compita en cuestiones de rendimiento, con algoritmos más complejos como el presentado por Vanaret [32].

De igual forma, aunque el enfoque del trabajo fue la optimización de funciones de energía

potencial, la metodología propuesta no se limita solo a este tipo de problemas, y puede extenderse a sistemas más complejos en ámbitos diferentes. Al mismo tiempo, el código se encuentra de manera abierta [1], con el fin de que pueda ser revisado y utilizado para cualquier persona que tenga un interés particular en el tema.

BIBLIOGRAFÍA

[1] Bryan A. Corzo. IntervalOptimization.jl. https://github.com/BryanCorzo/IntervalOptimization.

[2] The Julia Language. https://julialang.org/.

[3] David P. Sanders. IntervalArithmetic.jl. https://github.com/JuliaIntervals/IntervalArithmetic.jl.

[4] Errol G. Lewars. *Computational Chemistry. Introduction to the Theory and Applications of Molecular and Quantum Mechanics*. Springer Science, 2011.

[5] P. Tieleman L. Monticelli. Force fields for classical molecular dynamics. *Methods Mol Biol.*, pages 197–213, 2013.

[6] J. D. Fetter, A. L.; Walecka. *The Quantum Mechanics of Many-Body Systems*. New York: Dover, 2003.

[7] R. Oppenheimer M. Born. On the quantum theory of molecules. *Ann. d. Physik*, 84:457, 1927.

[8] M.A. González. Force fields and molecular dynamics simulations. *Collection SFN*, 12:169 – 200, 2011.

[9] J.E. Mayer F.H. Westheimer. The theory of the racemization of optically active derivatives of diphenyl. *J Chem Phys*, 14:733, 1946.

[10] T.L. Hill. On steric effects. *J Chem Phys*, 14:465, 1946.

[11] P.P. Scheleyer E.M. Engler. Critical evaluation of molecular mechanics. *J Am Chem Soc*, 95:8005, 1973.

[12] N. L. Allinger U. Burkert. *MolecularMechanics*. ACS Monograph 177 American

Chemical Society, Washington, DC, 1982.

[13] R.C. Walker R. Salomon-Ferrer, D.A. Case. An overview of the amber biomolecular simulation package. *WIREs Comput. Mol. Sci.*, 3:198, 2013.

[14] M.C. Nicklaus. Conformational energies calculated by the molecular mechanics program charmm. *J Comp Chem*, 18:1056, 1997.

[15] I. G. Tironi A. E. Mark S. R. Billeter J. Fennen A. E. Torda T. Huber P. Krüger W. F. van Gunsteren W. R. P. Scott, P. H. Hünenberger. The gromos biomolecular simulation program package. *J. Phys. Chem. A*, 103:3596, 1999.

[16] F. Lipparini F. Aviat B. Stamm Z.F Jing M. Harger H. Torabifard A. Cisneros M. Schnieders N. Gresh Y. Maday P. Ren J. Ponder J.P. Piquemal L. Lagardere, L. H. Jolly. Tinker-hp: a massively parallel molecular dynamics package for multiscale simulations of large complex systems with advanced point dipole polarizable force fields. *Chem. Sci.*, 9:956, 2018.

[17] A. R. Leach. *Molecular Modelling: Principles and Applications (2da Edición)*. Prentice Hall, 2001.

[18] Ilya G. Kaplan. *Intermolecular Interactions: Physical Picture, Computational Methods and Model Potentials*. John Wiley Sons Ltd, 2006.

[19] J. van Leeuwen. *Handbook of Theoretical Computer Science. Vol. A*. Amsterdam: Elsevier, 1998.

[20] J. P. K. Doye D. J. Wales. Global optimization by basin-hopping and the lowest energy structures of lennard-jones clusters containing up to 110 atoms. *J. Phys. Chem. A*, 101, 1997.

[21] S. J. Wrigh J. Nocedal. *Numerical Optimization*. Springer Science, 2000.

[22] T. A. Wheeler M. J. Kochenderfer. *Algorithms for Optimization*. MIT Press, 2019.

[23] C. M. Reeves R. Fletcher. Function minimization by conjugate gradients. *Comput. J.*, 7, 1964.

[24] G. Ribière E. Polak. Note sur la convergence de méthodes de directions conjuguées. *Rev. Fr. Inform. Rech. O.*, 3, 1969.

[25] J. Nocedal. Theory of algorithms for unconstrained optimization. *Math. Comput.*, 35, 1980.

[26] G. G. Hough. *Computer. IEEE*, 52:109, 2019.

[27] IEEE Computer Society (2019-07-22). *IEEE Standard for Floating-Point Arithmetic. IEEE STD 754-2019*. IEEE, 2019.

[28] R. E. Moore. Interval analysis. *Prentice Hall*, 1966.

[29] M. J. Cloud R. E. Moore, R. B. Kearfott. *Introduction to IntervalL Analysis*. SIAM, 2009.

[30] M.H. van Emden T. Hickey, Q. Ju. Interval arithmetic: From principles to implementation. *J. ACM*, 48:1038, 2001.

[31] D. Ratz. *On extended interval arithmetic and inclusion isotonicity*. Institut fur Angewandte Mathematik, Universitat Karlsruhe, 1996.

[32] C. Vanaret. *Hybridization of interval methods and evolutionary algorithms for solving difficult optimization problems*. Tesis de Doctorado, Universite de Toulouse, 2015.

[33] K. Madsen O. Caprani. Mean value forms in interval analysis. *Computing*, 25:147, 1980.

[34] E. Baumann. Optimal centered forms. *BIT*, 28:80, 1988.

[35] K. Nickel R. Krawczyk. Centered form in interval arithmetics: quadratic convergence and inclusion isotonicity. *Computing*, 28:117, 1982.

[36] D. E. Wood E. L. Lawler. Branch-and-bound methods: A survey. *Oper. Res.*, 14:699, 1966.

[37] Y. Fujii K. Ichida. An interval arithmetic method for global optimization. *Computing*, 23:85, 1979.

[38] V. Reyes I. Araya. Interval branch-and-bound algorithms for optimization and constraint satisfaction: a survey and prospects. *J. Global Optim*, 65:837, 2016.

[39] T. N. Grapsa D. G. Sotiropoulos. *An Interval Branch-and-Bound Algorithm for Global Optimization using Pruning Steps*. University of Patras, 2001.

[40] T.N. Grapsa R. Chenouard, A. Goldsztejn. *Search Strategies for an Anytime Usage*

of the Branch and Prune Algorithm. IJCAI, 2008.

[41] C. Oreallana I. Araya, V. Reyes. More smear-based variable selection heuristics for ncsps. *IEEE 25th International Conference on Tools with Artificial Intelligence*, 2013.

[42] C. Andrea H. Remzi. *Operating Systems: Three Easy Pieces.* Arpaci-Dusseau Books, 2014.

[43] D. Ratz T. Csendes. Generalized subinterval selection criteria for interval global optimization. *SIAM J. Numer. Anal*, 34:922, 2003.

[44] I. Araya B. Neveu, G.Trombettoni. Node selection strategies in interval branch and bound algorithms. *J. Global Optim*, 64:289, 2016.

[45] B. Neveu I. Araya. lsmear: a variable selection strategy for interval branch and bound solvers. *J. Global Optim*, 71:483, 2018.

[46] M. Kashiwagi S. Rump. Implementation and improvements of affine arithmetic. *NOLTA, IEICE*, 2015.

[47] P. Hansen J. Ninin, F. Messine. A reliable affine relaxation method for global optimization. *4OR-Q. J. Oper. Res.*, 13, 2015.

[48] F. Messine. Extentions of affine arithmetic: Application to unconstrained global optimization. *J. Univers. Comput. Sci.*, 8, 2002.

[49] R.L.Voller H. Ratschek. What can interval analysis do for global optimization? *J. Global Optim.*, 1, 1991.

[50] H. Eldon. Sharpening interval computations. *Reliable Computing*, 12:253, 2006.

[51] A.K. Mackworth. Consistency in networks of relations. *Artif. Intell.*, 8:99, 1977.

[52] G. Trombettoni I. Araya, B. Neveu. An interval constraint propagation algorithm exploiting monotonicity. *International Workshop IntCP, Interval Analysis, Constraint Propagation, Applications, at CP Conference*, 65, 2009.

[53] L. Liberti F. Margot A. Wächter P. Belotti, L. Lee. Branching and bounds tighteningtechniques for non-convex minlp. *Optim Method Softw*, 24, 2009.

[54] C. Lavor. A deterministic approach for global minimization of molecular potential energy functions. *Int. J. Quantum Chem.*, 95, 2003.

[55] M. A. Stadtherr Y. Lin. Deterministic global optimization of molecular structures using interval analysis. *J. Comput. Chem.*, 26, 2005.

[56] JuliaDiff.jl. https://github.com/JuliaDiff/.

[57] David P. Sanders. IntervalConstraintProgramming.jl. https://github.com/JuliaIntervals/IntervalConstraintProgramming.jl.

[58] T. Papamarkou J. Revels, M. Lubin. ForwardDiff. https://github.com/JuliaDiff/ForwardDiff.jl.

APÉNDICE A

ANEXO 1

El enfoque principal del algoritmo propuesto, *optimize*, está en la optimización global de manera rigurosa de funciones continuas y sin restricciones, mediante el análisis de intervalos. El código está escrito completamente en Julia y utiliza el paquete de IntervalArithmetic.jl [3] para la aritmética entre intervalos.

Al ser un código escrito completamente en Julia tiene acceso a las funciones de diferenciación automática a través de los paquetes en JuliaDiff [56]; de igual manera, permite la selección por parte del usuario de varios pasos en el algoritmo, modificando las posibilidades predefinidas elegidas por el desarrollador.

Para mostrar cómo se puede implementar el algoritmo, minimizaremos la función de Rosenbrock, un problema clásico de optimización. Se define la función como

```
function f(X::IntervalBox)
    x_1, x_2 = X
    return (1.0 - x_1)^2 + 100.0 * (x_2 - x^2)^2
end
```

Una vez definida la función simplemente especificamos los intervalos iniciales y llamamos a *optimize* con la función y los intervalos de partida

```
X = IntervalBox(-2..2, 2)
optimize(f, X)
```

Es importante aclarar, que el algoritmo entrega como respuesta, el mínimo, el minimizador y el número de iteraciones; la tolerancia ϵ está dada por defecto como $\epsilon = 10^{-3}$. Los parámetros y tipos de entrada se resumen a continuación

- **Función**, la función de entrada puede ser una función básica de Julia o una función de tipo constraint, para este último caso revisar IntervalConstraintProgramming.jl [57].

- **Intervalos iniciales**, *optimize* recibe como argumento de inicio un IntervalBox que debe contener los intervalos iniciales para cada una de las variables.

- **Tolerancia**, se tiene por defecto un $\epsilon = 10^{-3}$, este valor puede ser modificado a cualquier valor de tipo Float64.

- **Funciones de inclusión**, por defecto utiliza una inclusión natural de la función, no obstante, se puede modificar para que utilice el mean value form, en este caso, el usurario puede proporcionar un gradiente de lo contrario se construirá un gradiente utilizando ForwardDiff.jl [58].

- **Métodos de búsqueda** por defecto se utiliza el método de Best-First Seach (BFS), sin embargo, se puede modificar por Depth-first search (DFS), Breadth-first search (BrFS), Most distant-first search (MDFS) y Depth Most distant-first search (DMDFS).

- **Optimización local** se utiliza un método de descenso de gradiente.

- **Métodos de bisección**, por defecto se realiza la bisección en la variable con el dominio más amplio, sin embargo, se puede modificar esta metodología para utilizar métodos de Smear.

- **Generación de puntos aleatorios** Por defecto se generan 75 puntos aleatorios para la búsqueda; este valor se puede modificar a cualquier valor de tipo Int64.

Volviendo a la función de Rosenbrock, una implementación completa seria

```
optimize(f, X, eps = 0.01, search = MDFS, localOpt = true,bisect = Smear, num_alea =  50)
```