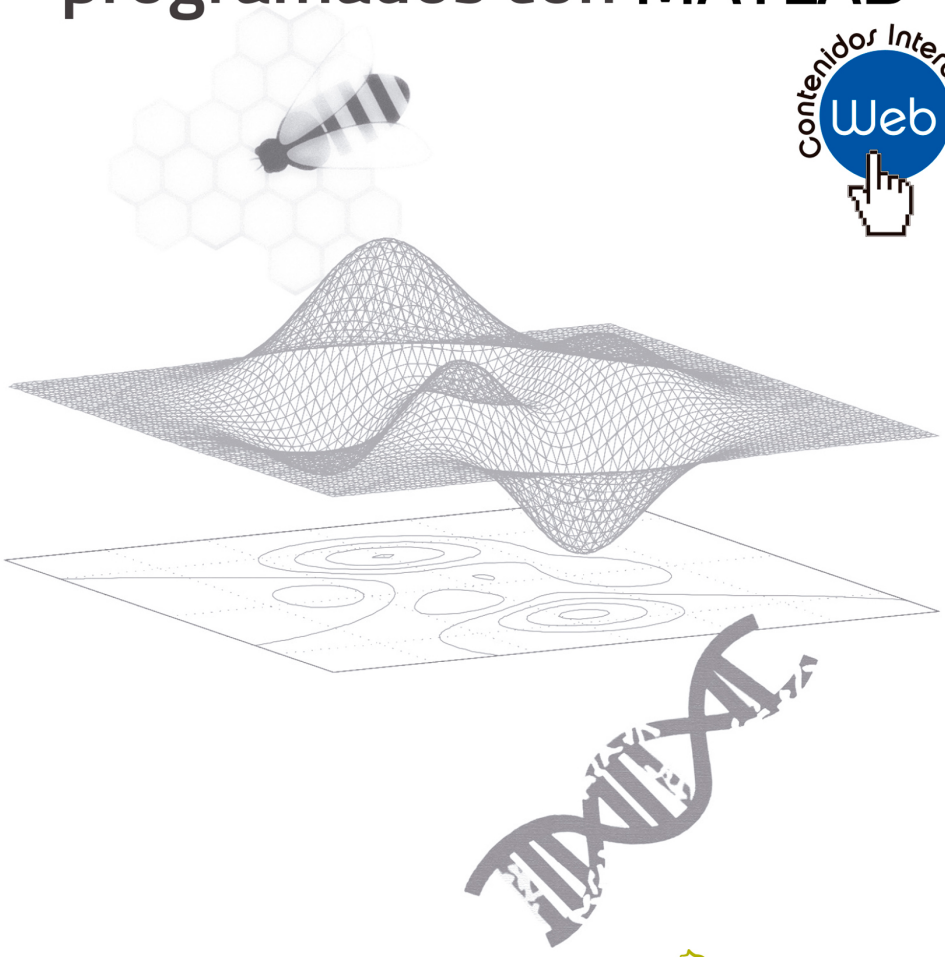
Erik V. Cuevas Jiménez • José V. Osuna Enciso
Diego A. Oliva Navarro • Margarita A. Díaz Cortés

OPTIMIZACIÓN

Algoritmos programados con MATLAB



Contenidos Interactivos Web

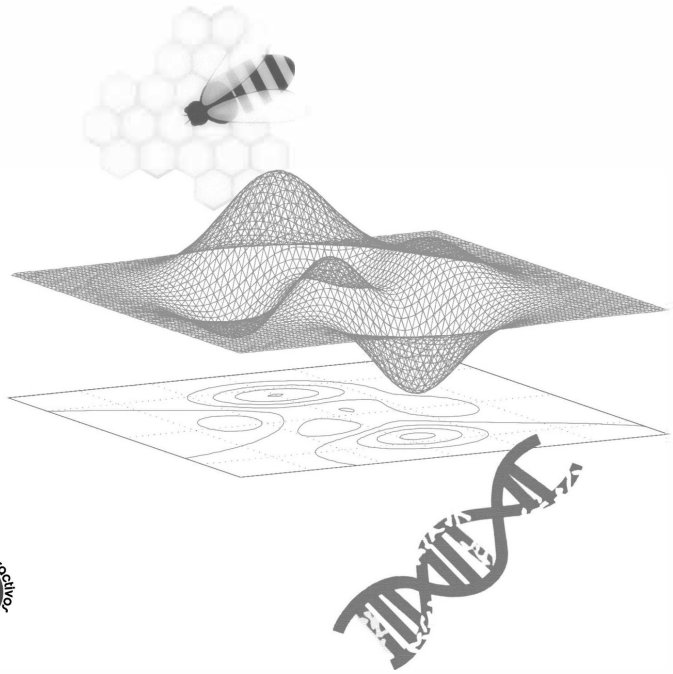
Alfaomega

Marcombo

Erik V. Cuevas Jiménez • José V. Osuna Enciso
Diego A. Oliva Navarro • Margarita A. Díaz Cortés

OPTIMIZACIÓN

Algoritmos programados con MATLAB



Contenidos Interactivos Web

Alfaomega

Marcombo

Optimización. Algoritmos programados con MATLAB
Erik V. Cuevas Jiménez – José V. Osuna Enciso – Diego A. Oliva Navarro – Margarita A. Díaz Cortés

Primera edición: Alfaomega Grupo Editor, México, junio 2016
ISBN: 978-607-622-689-6

Primera edición: MARCOMBO, S.A. 2017

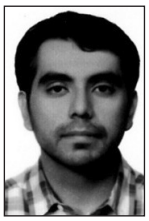
ISBN: 978-84-267-2411-3

D.L.: B-25224-2016

Impreso en Ulzama Digital SL
Printed in Spain

Acerca de los autores

Erik Valdemar Cuevas Jiménez. Es Ingeniero en Comunicaciones y Electrónica por parte de la Universidad de Guadalajara, Maestro en Electrónica Industrial por el ITESO y Doctor en Inteligencia Artificial por la Universidad Libre de Berlín (FU-Berlín) en Alemania. Desde 2007 es Profesor Investigador en el Departamento de Electrónica en el Centro Universitario de Ciencias Exactas e Ingenierías (CUCEI) en Guadalajara, México. Es autor de varios libros y artículos de revistas especializadas. Desde 2012 es miembro del comité editorial de la revista "Mathematical Problems in Enginerring". Sus líneas de investigación incluyen el cómputo evolutivo y sus aplicaciones al procesamiento de imagen.



Diego Alberto Oliva Navarro. En el 2007 obtuvo el grado de Ingeniero en Electrónica y Computación por el Centro de Enseñanza Técnica Industrial (CETI) en Guadalajara, México. En el año 2010 se graduó como Maestro en Ciencias en Ingeniería Electrónica y Computación en el Centro Universitario de Ciencias Exactas e Ingenierías (CUCEI) de la Universidad de Guadalajara (UDG), en México. En el año 2015 obtuvo el grado de Doctor en Ingeniería Informática por la Universidad Complutense de Madrid (UCM) en España. Desde el 2008 ha centrado sus investigaciones en el desarrollo, implementación y diseño de algoritmos metaheurísticos. Ha publicado diversos trabajos en revistas internacionales, en temas referentes a la optimización y sus implementaciones. Desde el 2015 es miembro de la Academia Mexicana de Computación. Actualmente se desempeña como profesor e investigador en el departamento de ciencias computacionales del Tecnológico de Monterrey, campus Guadalajara. Sus principales temas de estudio son: inteligencia artificial, algoritmos de optimización metaheurísticos, optimización multiobjetivo, estimación de parámetros en ingeniería, procesamiento de imágenes y señales digitales.

Margarita Arimatea Díaz Cortés. Recibió el grado en Ingeniera Biomédica en el 2012 en la Universidad de Guadalajara y el grado de Maestra en Ciencias de la Ingeniería Electrónica y Computación en 2014 por la misma institución. Actualmente estudia el Doctorado en Ciencias Computacionales en la Universidad Libre de Berlín. Sus intereses de investigación incluyen el desarrollo de aplicaciones de la inteligencia artificial en el área médica, técnicas de optimización y el desarrollo de nuevos algoritmos evolutivos.

José Valentín Osuna Enciso. Estudió Ingeniería en Electrónica con especialidad en Sistemas Digitales en el Instituto Tecnológico del Mar en Mazatlán, Sinaloa, en 2000, la Maestría en Ciencias en Ingeniería Electrónica y Computación en el Centro Universitario de Ciencias Exactas e Ingenierías de la Universidad de Guadalajara en 2010, y el Doctorado en Ciencias de la Computación por el Centro de Investigación en Computación del Instituto Politécnico Nacional en 2013. Actualmente se desempeña en el Centro Universitario de Tonalá como Profesor Investigador Asociado C, y pertenece al Sistema Nacional de Investigadores, nivel 1. Sus líneas de investigación incluyen algoritmos evolutivos, optimización, y visión por computadora.

Mensaje del editor

Una de las convicciones fundamentales de Alfaomega es que los conocimientos son esenciales en el desempeño profesional, ya que sin ellos es imposible adquirir las habilidades para competir laboralmente. El avance de la ciencia y de la técnica hace necesario actualizar continuamente esos conocimientos, y de acuerdo con esto Alfaomega publica obras actualizadas, con alto rigor científico y técnico, y escritas por los especialistas del área respectiva más destacados.

Consciente del alto nivel competitivo que debe de adquirir el estudiante durante su formación profesional, Alfaomega aporta un fondo editorial que se destaca por sus lineamientos pedagógicos que coadyuvan a desarrollar las competencias requeridas en cada profesión específica.

De acuerdo con esta misión, con el fin de facilitar la comprensión y apropiación del contenido de esta obra, cada capítulo inicia con el planteamiento de los objetivos del mismo y con una introducción en la que se plantean los antecedentes y una descripción de la estructura lógica de los temas expuestos, asimismo a lo largo de la exposición se presentan ejemplos desarrollados con todo detalle y cada capítulo concluye con un resumen y una serie de ejercicios propuestos.

Además de la estructura pedagógica con que están diseñados nuestros libros, Alfaomega hace uso de los medios impresos tradicionales en combinación con las Tecnologías de la Información y las Comunicaciones (TIC) para facilitar el aprendizaje. Correspondiente a este concepto de edición, todas nuestras obras tienen su complemento en una página Web en donde el alumno y el profesor encontrarán lecturas complementarias así como programas desarrollados en relación con temas específicos de la obra.

Los libros de Alfaomega están diseñados para ser utilizados en los procesos de enseñanza aprendizaje, y pueden ser usados como textos en diversos cursos o como apoyo para reforzar el desarrollo profesional, de esta forma Alfaomega espera contribuir a la formación y al desarrollo de profesionales exitosos para beneficio de la sociedad, y espera ser su compañera profesional en este viaje de por vida por el mundo del conocimiento.

Contenido

Capítulo 8

Capítulo 9

Prólogo

Las aplicaciones de la optimización son incontables. Cada proceso es factible ser optimizado. No existe compañía alguna que no involucre dentro de sus actividades la solución de problemas de optimización. En términos generales muchas aplicaciones interesantes en las ciencias y la industria pueden ser formuladas como procesos de optimización. Optimización se presenta en la minimización del tiempo invertido para la ejecución de una tarea, el costo de un producto y el riesgo en una inversión o en la maximización de las ganancias, la calidad de un producto y la eficiencia de un dispositivo.

La gran mayoría de los problemas de optimización con implicaciones prácticas en ciencias, ingeniería, economía y negocios son muy complejos y difíciles de resolver. Tales problemas no pueden ser resueltos de una manera exacta usando métodos de optimización clásicos. Bajo estas circunstancias los métodos de cómputo evolutivo se han consolidado como una solución alternativa.

Los algoritmos de cómputo evolutivo son considerados herramientas genéricas de optimización que pueden resolver problemas muy complejos caracterizados por contar un espacio de búsqueda demasiado grande. Los métodos de cómputo evolutivo reducen el tamaño efectivo del espacio de búsqueda mediante el uso de estrategias efectivas de búsqueda. En general, dichos métodos permiten resolver los problemas de una manera más rápida y robusta. En comparación a otros algoritmos heurísticos, las técnicas de cómputo evolutivo son más simples de diseñar e implementar.

Los métodos evolutivos representan un área de la optimización en las ciencias de la computación y las matemáticas aplicadas. Los últimos 10 años han sido testigos del desarrollo de numerosos enfoques evolutivos que permiten la intersección de diferentes disciplinas que incluyen la inteligencia artificial, la biología, estudios sociales y las matemáticas. La mayoría de los métodos de cómputo evolutivo usan como inspiración fenómenos biológicos o sociales, los cuales en cierto nivel de abstracción pueden ser considerados como modelos de optimización.

Recientemente, los algoritmos de cómputo evolutivo se han popularizado tanto en la academia como en la industria. Un indicador de esta situación es el gran número de revistas, sesiones y conferencias especializadas en esta área. En la práctica, los métodos evolutivos han despertado gran interés, ya que han demostrado ser herramientas eficientes para la solución de un amplio rango de problemas en dominios tales como la logística, bio-informatica, diseño estructural, data mining, finanzas, etc.

El principal objetivo de este libro es brindar una visión unificada de los métodos de cómputo evolutivo, de tal forma que se presentan los principios de diseño fundamentales así como los operadores de los enfoques evolutivos que son considerados esenciales. En su tratamiento no sólo los aspectos de diseño han sido considerados, sino también su implementación usando la popular plataforma de programación MatLAB. La idea de esta combinación es motivar al lector a que con el conocimiento adquirido de cada método, reutilice el código existente para ser configurado a las necesidades particulares de los problemas que él tuviera necesidad de resolver. Todo el código en MatLAB mostrado en cada uno de los capítulos además de material adicional, puede ser descargado de la página de la editorial.

El libro provee los conceptos necesarios que habilitan al lector a implementar y modificar los métodos de cómputo evolutivo ya conocidos, para obtener los desempeños proyectados para las necesidades especificas de cada problema. Para ello, el libro contiene numerosos ejemplos de problemas y soluciones que demuestran la potencia de estos métodos de optimización.

Debido a su contenido y estructura el libro es adecuado para estudiantes y profesionales en el área de ciencias de la computación, inteligencia artificial, investigación de operaciones, matemáticas aplicadas, control y algunas otras disciplinas. De igual manera, muchos ingenieros que laboran en la industria pueden encontrar interesante el contenido de esta obra. En este caso, las sencillas exposiciones y el código provisto pueden servir de ayuda para la rápida solución a problemas optimización que normalmente surgen en varios nichos y proyectos industriales.

Escribir un libro de métodos de cómputo evolutivo parece intrascendente. Existen ya varios libros de texto y de consulta dirigidos a lectores con diferentes niveles de conocimiento en la materia. Sin embargo, de las decenas de libros que existen, estos se reducen a prácticamente ninguno si se restringen las opciones a los escritos en español. Bajo estas circunstancias, cuando se concibió el proyecto de escritura, se decidió que era necesario tener algo que decir sobre el orden, la profundidad y la manera de exponer los conceptos de cómputo evolutivo en nuestro idioma.

Nuestra premisa original fue que los métodos de cómputo evolutivo pueden exponerse de manera comprensible para lectores con poco entrenamiento matemático. En consecuencia, intentamos escribir un libro cuyo contenido fuese no sólo aplicable, sino entendible por cualquier estudiante de licenciatura. Aunque algunos conceptos pueden ser complejos para su comprensión cabal, tratamos de exponerlos con claridad y sin pretender disimular su dificultad implícita. Sin sacrificar las necesidades inmediatas del lector, subrayamos la importancia de que se comprendan los métodos descritos, bajo el supuesto de que a veces es preferible no usar estos métodos de optimización que usarlos mal.

En el largo proceso de escribir este libro, nuestra perspectiva ha variado. Planeado para un curso introductorio, el material que presentamos puede cubrirse en un semestre. El libro está constituido de 11 capítulos, los detalles en el tratamiento de cada uno de ellos se describen a continuación:

En el **capítulo 1** se introduce el concepto de optimización. Una vez formulado de manera genérica el problema de optimización, se clasifican los métodos utilizados para su solución. El capítulo expone las principales características de los algoritmos evolutivos además de introducir al dilema de la exploración y explotación. De manera importante, se analiza la aceptación y selección probabilística, dos de las principales operaciones usadas por la mayoría de los métodos evolutivos. El capítulo termina con tres de los primeros métodos evolutivos que han servido de base para

la creación de nuevos algoritmos, la idea con esta exposición es introducir a los conceptos de los métodos evolutivos, mediante técnicas relativamente fáciles de implementar.

En el **capítulo 2** se analizan los algoritmos genéticos. En él se mencionan algunas de las diferentes variantes que existen hasta el momento, así como algunos puntos importantes a considerar al implementarse como: a) los distintos tipos de cruza, b) se explica el algoritmo de selección de ruleta, y c) se establecen sus principales operadores; se termina con un pequeño código de ejemplo que resuelve una función con cierto grado de complejidad en dos dimensiones.

En el **capítulo 3** se presentan las estrategias evolutivas (EE) las cuales pertenecen al primer bloque de algoritmos inspirados en la evolución. En él se explican de manera detallada cada uno de los operadores, poniendo especial énfasis en la mutación. La parte de las explicaciones se acompaña con segmentos de código en MatLAB, y al final del capítulo se detalla un programa completo de EE en una de sus versiones más utilizadas.

En el **capítulo 4** se describe el algoritmo de optimización por enjambre de partículas (PSO, por sus siglas en inglés, Particle Swarm Optimization), uno de los algoritmos de cómputo evolutivo más usados, debido a la sencillez de sus operadores. En el tratamiento de este capítulo se describen las bases del PSO además de analizar la teoría, se describe un ejemplo práctico y su codificación.

En el **capítulo 5** se realiza un tratamiento del Algoritmo evolución diferencial (DE, por sus siglas en inglés, Differential Evolution). En este capítulo, se discuten los parámetros del algoritmo evolución diferencial, así como los pasos necesarios para implementarlo.

En el **capítulo 6** se presenta el algoritmo de optimización por búsqueda armónica (Harmony Search Algorithm, HSA*)*. Dicho método es relativamente nuevo y cada vez es más usado en un gran número de aplicaciones. En ese capítulo se estudian las bases del algoritmo HSA, se describe su codificación y se verifica su desempeño con un ejemplo práctico.

En el **capítulo 7** se analizan los sistemas inmunes artificiales (SIA). En esta clase de algoritmos confluyen ideas extraídas de algunas teorías de la inmunología natural –tales como la teoría de selección clonal, la teoría de redes, o la teoría de peligro–, con el propósito de desarrollar sistemas capaces de realizar distintas tareas en campos como la investigación o la ingeniería. En el capítulo se dan los detalles en cuanto a los operadores utilizados en uno de los algoritmos más utilizados –algoritmo de selección clonal, ASC–, así como los pormenores referentes a su implementación en código.

En el **capítulo 8** se analiza el algoritmo de optimización inspirado en principios del electromagnetismo (Electromagnetism-Like Optimization, EMO). El cual emplea una población de partículas cargadas para encontrar el valor óptimo global en un espacio de búsqueda acotado. La carga de los elementos de la población emula al valor que cada uno ellos tienen en la función objetivo. En este capítulo se describe a detalle la analogía del algoritmo EMO, se analiza cada uno de sus operadores, se presenta un ejemplo práctico y su codificación en MatLAB.

En el **capítulo 9** se presenta el algoritmo colonia artificial de abejas (ABC, por sus siglas en inglés, Artificial Bee Colony). En él se discuten los parámetros del algoritmo, así como la codificación necesaria para implementarlo. De manera adicional, se presenta una reseña acerca del algoritmo ABC aplicado al área de procesamiento de imágenes.

En el **capítulo 10** se analizan las técnicas más comunes, usadas por los métodos de cómputo evolutivo para la optimización de problemas de optimización, planteados como multimodales. Debido a que el algoritmo de función compartida es el más popular, dicho procedimiento es tratado con mayor profundidad en el capítulo. Adicionalmente, en la parte final, se discute el algoritmo de las luciérnagas, el cual es un método inspirado en el comportamiento de atracción de estos insectos, en este se incorporan operadores especiales que le permiten explotar interesantes capacidades multimodales.

El libro está estructurado de tal manera que el lector puede claramente identificar desde el inicio los objetivos de cada capítulo y al final reforzar los conocimientos adquiridos, mediante la conducción de una serie ejercicios propuestos. Los ejercicios están divididos en dos clases de actividades, (A) aquellas en donde se analiza un resultado o se desarrolla manualmente un cálculo y (B) aquellas en las que necesariamente se involucra programación. Con el objeto de identificar claramente cada ejercicio, se utiliza el icono  para referirse a las actividades de programación.

Por más de cinco años hemos ensayado múltiples maneras de exponer este material a auditorios disímiles. En el camino se ha contado con la invaluable tolerancia de nuestros alumnos, principalmente de la Universidad de Guadalajara en México así como del Tec de Monterrey Campus Guadalajara. De primordial importancia ha sido el apoyo de los profesores Raúl Rojas, Gonzalo Pajares y Humberto Sossa. Se agradece además de manera especial a nuestros compañeros profesores del Centro Universitario de Ciencias Exactas e Ingenierías Daniel Zaldívar y Marco Pérez. Tantas colaboraciones, ayudas y discusiones con colegas, ameritarían un capítulo adicional. A todos, los mencionados y en especial a los omitidos, nuestro testimonio de gratitud.

Plataforma de contenidos interactivos

Para tener acceso al material de la plataforma de contenidos interactivos de este libro, siga los siguientes pasos:

1. Abrir la página: http://libroweb.alfaomega.com.mx
2. Ir a la sección *Catálogo* y escribir en el buscador el título de esta obra, al dar doble clic sobre la imagen de la portada, tendrá acceso al material descargable.

NOTA: Se recomienda respaldar los archivos descargados de la página web en un soporte físico.

Erik V. Cuevas
José V. Osuna
Diego A. Oliva
Margarita A. Díaz

Noviembre, 2015
Guadalajara, Jalisco, México

Capítulo 1

Optimización y métodos de cómputo evolutivo

Objetivos

El objetivo de este capítulo es introducir los principales conceptos que envuelve el proceso de optimización. De esta manera, una vez formulado de manera genérica el problema de optimización, se clasificarán los métodos utilizados para su solución. Considerando que el libro se focaliza en el estudio de las técnicas de cómputo evolutivo, los algoritmos tradicionales basados en gradiente serán sólo marginalmente tratados. Otro objetivo importante en este capítulo es exponer las principales características de los algoritmos evolutivos además de introducir al dilema exploración y explotación. De manera importante, se analizará la aceptación y selección probabilística, dos de las principales operaciones usadas por la mayoría de los métodos evolutivos. Finalmente, se expondrá tres de los primeros métodos evolutivos, que han servido de base para la creación de nuevos algoritmos. La idea con este tratamiento es introducir a los conceptos de los métodos evolutivos mediante la implementación de técnicas relativamente fáciles.

1.1 Introducción

Optimización se ha convertido en una parte importante en todas las disciplinas. Una razón de esta inclusión es la motivación de producir productos o servicios de calidad a precios competitivos. En general, optimización representa el proceso de encontrar la "mejor solución" a un problema entre un conjunto muy grande de soluciones posibles [1].

Un problema de optimización puede ser formulado como un proceso donde se desea encontrar el valor óptimo x^* que minimiza o maximiza la función objetivo $f(x)$. Tal que:

$$\begin{array}{ll} \text{Minimizar/ Maximizar} & f(\mathbf{x}),\ \mathbf{x}=(x_1,\ldots,x_d)\in\mathbb{R}^d \\ \text{considerando} & \mathbf{x}\in\mathbf{X} \end{array} \tag{1.1}$$

Donde $\mathbf{x}$ representa el vector de variables de decisión mientras d especifica su número. $\mathbf{X}$ representa el conjunto de soluciones candidato, conocido también como espacio de búsqueda o espacio de soluciones. En muchas ocasiones el espacio de búsqueda se encuentra limitado por los límites inferior (l_i) o superior (u_i) de cada una de las d variables de decisión tal que $\mathbf{X}=\left\{\mathbf{x}\in\mathbb{R}^d \,|\, l_i\le x_i\le u_i, i=1,\ldots,d\right\}$.

Algunas veces es necesario minimizar $f(\mathbf{x})$ otras veces maximizar. Estos dos tipos de problemas son fácilmente convertidos uno a otro mediante la siguiente relación:

$$\max_{x\cdot} f(x) \Leftrightarrow \min_{x\cdot} -1\bullet f(x) \tag{1.2}$$

Con el objetivo de aclarar los conceptos anteriores se presenta como ejemplo el siguiente problema de minimización:

$$\begin{array}{ll} \text{Minimizar} & f(x)=x^4+5x^3+4x^2-4x+1 \\ \text{considerando} & x\in[-4,1] \end{array} \tag{1.3}$$

En esta formulación, se presenta como problema la minimización de una función de una sola variable de decisión ($d = 1$). El espacio de búsqueda $\mathbf{X}$ para este problema está integrado por el intervalo de -4 a 1. Bajo estas circunstancias, la idea es encontrar el valor de x, para el cual $f(x)$ presenta su valor mínimo, considerando como el conjunto de soluciones posibles el intervalo definido $[-4,1]$. El valor de x que resuelve esta formulación es llamado el valor óptimo y es representado como x^*. La figura 1.1 presenta una representación gráfica de la función a minimizar.

De la figura 1.1 pueden distinguirse las soluciones A y B que corresponden a dos diferentes mínimos obtenidos de $f(x)$. Este tipo de funciones son conocidas como multimodales por sus características de contener varios mínimos prominentes. El mínimo representado por el punto A representa la solución óptima x^* del problema, mientras que B es sólo un mínimo local.

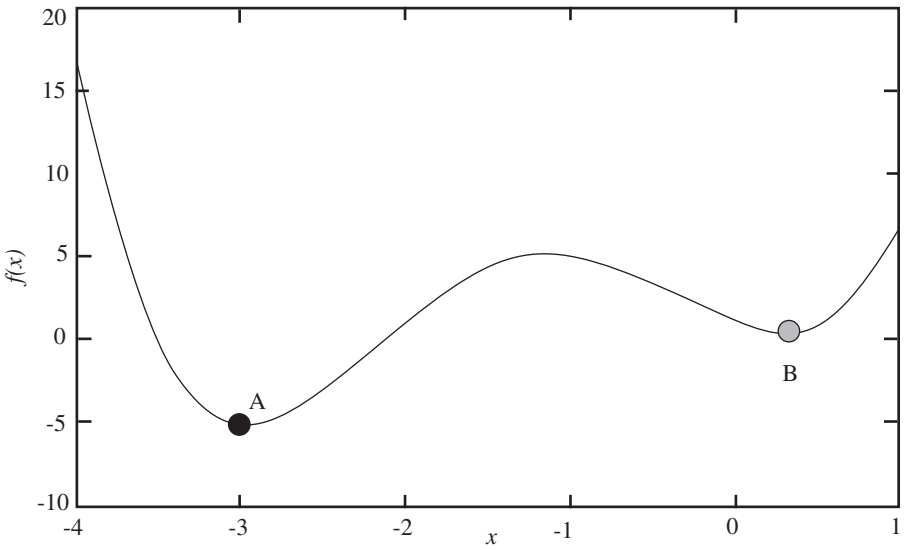


Figura 1.1. Representación gráfica del problema de optimización formulado en la ecuación 1.3.

1.2 Métodos clásicos de optimización

En general, la función $f(\mathbf{x})$ puede tener una forma no lineal con respecto a las variables de decisión **x**. Debido a esta complejidad, los métodos de optimización plantean procesos iterativos para la exploración del eficiente del espacio de búsqueda. Existen 2 tipos de familias de algoritmos usados para resolver esta clase de problemas [2]: las técnicas clásicas y los métodos evolutivos. Los métodos clásicos se basan en el uso del gradiente de la función $f(\mathbf{x})$ para la generación de nuevas soluciones candidatas. En el caso de los métodos evolutivos no necesitan información funcional de la derivada para la búsqueda del vector de decisión que minimiza (o maximiza) una determinada función objetivo. En lugar de ello, en estos métodos se implementa un conjunto de reglas heurísticas que dirigen el proceso de búsqueda. Algunas de estas reglas se basan en la reproducción de fenómenos presentes en la naturaleza o la sociedad.

Por las características de funcionamiento de los algoritmos clásicos de optimización, éstos requieren que la función objetivo $f(\mathbf{x})$ presente dos requisitos fundamentales para su uso: que sea dos veces diferenciable y que sea unimodal [3].

En esta sección se hace una revisión básica de las técnicas de optimización basadas en gradiente. Como el libro discute en profundidad los métodos evolutivos, el tema de la optimización clásica es tratado sólo marginalmente, se recomienda al lector referirse a otros libros si desea profundizar en estos temas.

1.2.1 Método del gradiente descendiente ❯

El método del gradiente descendiente, también conocido solamente como el método de gradiente, es una de las primeras técnicas utilizadas para la minimización de funciones objetivo multidimensionales [4]. Este método forma la base en la que se fundamentan varios otros algoritmos de optimización más sofisticados. A pesar de su lenta convergencia, el método de gradiente descendiente es el que más frecuentemente se usa para la optimización de funciones no lineales. Tal hecho se debe a su simplicidad y facilidad de programación.

En este método, partiendo de un punto inicial $\mathbf{x}^0$, el vector de decisión se modifica iterativamente hasta que transcurrido un número *Niter* de iteraciones pueda tentativamente encontrarse la solución óptima $\mathbf{x}^*$. Tal modificación iterativa es determinada por la siguiente expresión [5]:

$$\mathbf{x}^{k+1} = \mathbf{x}^k - \alpha \cdot \mathbf{g}(f(\mathbf{x})), \tag{1.4}$$

donde k representa la iteración actual y α representa el tamaño del paso de la búsqueda.

En 1.4 el término $\mathbf{g}(f(\mathbf{x}))$ representa el gradiente de la función $f(\mathbf{x})$. El gradiente $\mathbf{g}$ de una función $f(\mathbf{x})$ en el punto $\mathbf{x}$ expresa la dirección en la que la función $f(\mathbf{x})$ presenta su máximo crecimiento. De esta manera, en el caso de un problema de minimización, la dirección de descenso puede ser obtenida en sentido contrario a $\mathbf{g}$ (multiplicando por -1). Bajo esta regla, se garantiza que $f(\mathbf{x}^{k+1}) < f(\mathbf{x}^k)$, lo cual implica que la nueva solución generada es mejor que la anterior.

Normalmente, aunque la formulación de un problema de optimización involucra la definición de una función objetivo $f(\mathbf{x})$, esto es sólo con fines didácticos y de demostración. En la práctica no se conoce en forma determinista la definición de $f(\mathbf{x})$. Sus valores son conocidos solamente en los puntos requeridos por el algoritmo de optimización. Bajo estas circunstancias, el gradiente $\mathbf{g}$ es calculado mediante el uso de métodos numéricos.

1.2.2 Cálculo del gradiente ❯

El gradiente de una función multidimensional $f(\mathbf{x})$ $(\mathbf{x} = (x_1,\ldots,x_d) \in \mathbb{R}^d)$ representa la manera en que la función varía con respecto a uno de sus d dimensiones. De esta manera el gradiente g_{x_1} expresa la forma en que varía $f(\mathbf{x})$ con respecto a x_1. Dicho gradiente es definido apropiadamente como:

$$g_{x_1} = \frac{\partial f(\mathbf{x})}{\partial x_1} \tag{1.5}$$

Para calcular numéricamente el gradiente g_{x_i} se sigue el siguiente procedimiento [6].

1. Se genera un nuevo vector de decisión $\tilde{\mathbf{x}}_i$. Dicho vector es igual en todas las variables de decisión que $\mathbf{x}$ excepto en x_i. Este valor será sustituido por $x_i + h$, donde h es un valor muy pequeño. Bajo estas condiciones el nuevo vector es definido como:

$$\tilde{\mathbf{x}}_i = (x_1, x_2, \ldots, x_i + h, \ldots, x_d) \tag{1.6}$$

2. Se calcula el gradiente g_{x_i} mediante el siguiente modelo:

$$g_{x_i} \approx \frac{f(\tilde{\mathbf{x}}_i) - f(\mathbf{x})}{h} \tag{1.7}$$

La figura 1.2 muestra una representación gráfica del proceso de cálculo numérico del gradiente considerando un ejemplo simple, donde $f(\mathbf{x})$ tiene 2 dimensiones (x_1, x_2).

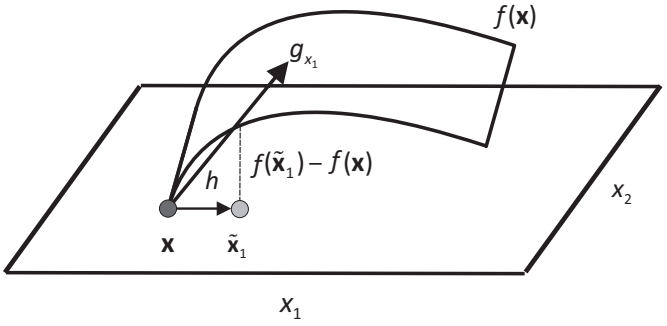


Figura 1.2. Representación gráfica del cálculo numérico del gradiente.

1.2.3 Ejemplo computacional en MatLAB ❯

Con el objetivo de ejemplificar el funcionamiento del algoritmo de gradiente descendiente y su implementación práctica en MatLAB, se pretende resolver el siguiente problema de minimización:

Minimizar
$$f(x_1, x_2) = 10 - e^{-\left(x_1^2 + 3x_2^2\right)}$$

considerando
$$-1 \le x_1 \le 1 \qquad (1.8)$$
$$-1 \le x_2 \le 1$$

En esta formulación, se considera una función bidimensional $f(x_1, x_2)$ con un espacio de búsqueda definido en el intervalo $[-1,1]$ para cada una de las variables de decisión x_1 y x_2. La figura 1.3 muestra una representación de la función objetivo $f(x_1, x_2)$.

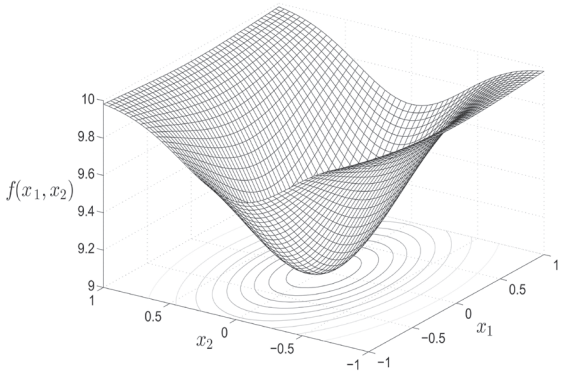


Figura 1.3. Representación gráfica de la función $f(x_1, x_2) = 10 - e^{-\left(x_1^2 + 3x_2^2\right)}$.

Como puede apreciarse, la función $f(x_1, x_2)$ puede ser derivada dos veces y es unimodal (sólo tiene un mínimo). Por tales razones cumple los requisitos para poder ser minimizada bajo el método del gradiente descendiente.

El proceso para la minimización de $f(x_1, x_2)$ mediante el método del gradiente descendiente es presentado en forma de pseudocódigo en el algoritmo 1.1. El procedimiento inicia seleccionando aleatoriamente un vector de decisión $\mathbf{x}^k$ (k = 0), dentro del espacio de búsqueda definido en el intervalo $[-1,1]$ para cada una de las variables x_1 y x_2. Después, se calcula los gradientes g_{x_1} y g_{x_2} en el punto $\mathbf{x}^k$. Con esta información, se obtiene una nueva solución $\mathbf{x}^{k+1}$ como consecuencia de aplicar la ecuación de actualización 1.4. Como una posible solución a $f(x_1, x_2)$ está integrada por dos variables de decisión, la actualización de la solución definida por 1.4 considera a cada variable de forma desacoplada. Dicha actualización se define como:

$$\begin{aligned} x_1^{k+1} &= x_1^k - \alpha \cdot g_{x_1} \\ x_1^{k+1} &= x_2^k - \alpha \cdot g_{x_1} \end{aligned} \tag{1.9}$$

Este proceso es repetido iterativamente hasta que un número máximo de iteraciones *Niter* se hayan alcanzado.

Con el objetivo de brindar al lector una ayuda en la implementación el algoritmo 1.1, el programa 1.1 muestra el código en MatLAB. En el funcionamiento del programa 1.1 primero es graficada la función $f(x_1, x_2)$ con el objetivo de apreciar sus características. Después, de manera iterativa se muestra la trayectoria que siguen en las soluciones candidato desde su valor inicial $\mathbf{x}^0$ hasta encontrar el valor óptimo $\mathbf{x}^*$. La figura 1.4 muestra un ejemplo de esta trayectoria, sobre un contorno de MatLAB, obtenida por el programa 1.1.

Algoritmo 1.1 Método de gradiente descendiente para resolver 1.8	
1.	k $\leftarrow$ 0
2.	$x_1^k \leftarrow$ Aleatorio $[-1,1]$, $x_2^k \leftarrow$ Aleatorio $[-1,1]$
3.	**while** (k $<$ *Niter*) **{**
4.	$g_{x_1} \leftarrow \dfrac{f(x_1^k + h, x_2^k) - f(x_1^k, x_2^k)}{h}$, $g_{x_1} \leftarrow \dfrac{f(x_1^k + h, x_2^k) - f(x_1^k, x_2^k)}{h}$
5.	$x_1^{k+1} \leftarrow x_1^k - \alpha \cdot g_{x_1}$, $x_2^k \leftarrow x_2^k - \alpha \cdot g_{x_2}$
6.	k $\leftarrow$ k+1 **}**