



INSTITUTO POLITÉCNICO NACIONAL
ESCUELA SUPERIOR DE CÓMPUTO
INTELIGENCIA ARTIFICIAL



PRÁCTICA 6
BÚSQUEDA INFORMADA P.III

NOMBRE DEL ALUMNO: GARCÍA QUIROZ GUSTAVO IVAN

NOMBRE DEL PROFESOR: GARCÍA FLORIANO ANDRES

GRUPO: 6CV3

FECHA DE ENTREGA DEL REPORTE: 17/10/2024

Índice

Marco teórico.....	1
Función de Himmelblau	1
Simulated Annealing	1
Introducción.....	2
Material y equipo.	3
Hardware	3
Software.....	3
Desarrollo de la práctica.....	4
Actividades desarrolladas	4
Función de Himmelblau	4
Implementación del algoritmo de recocido simulado	4
Ejecución y ajuste de parámetros.....	5
Visualización y análisis de resultados.....	6
Resultados obtenidos.....	8
Conclusiones.....	10
Referencias.	11
Código.....	11

Marco teórico

Función de Himmelblau

La función de Himmelblau es una función multimodal, definida sobre \mathbb{R}^2 y usada para comprobar el rendimiento de los algoritmos de optimización.

La función se define de la siguiente manera:

$$f(x, y) = (x^2 + y - 11)^2 + (x + y^2 - 7)^2.$$

Figura 1 Función de Himmelblau.

Tiene un máximo local en $(-1, 1)$ y cuatro mínimos locales idénticos (también son mínimos globales):

Figura 2 Mínimos globales.

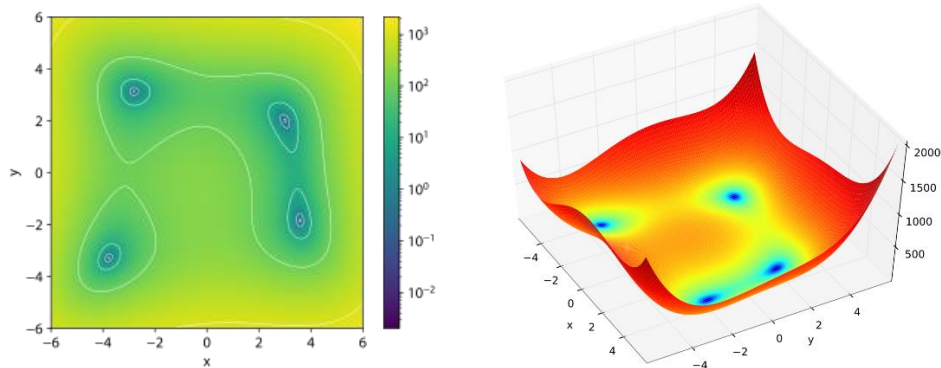


Figura 3 Gráfica de la función de Himmelblau.

Simulated Annealing

Introducción

En la actualidad, la búsqueda de soluciones óptimas en problemas de optimización es un campo de gran relevancia dentro de la inteligencia artificial, las matemáticas aplicadas y otras áreas relacionadas con la ciencia de los datos y la computación. En este contexto, los algoritmos de búsqueda informada juegan un papel fundamental para abordar problemas que involucran la minimización o maximización de funciones complejas. A través de estos algoritmos, se busca encontrar soluciones más eficientes y exactas en espacios de búsqueda de gran tamaño o con la presencia de múltiples mínimos locales, los cuales complican la búsqueda de la solución óptima global.

Una de las técnicas destacadas dentro de la búsqueda informada es el recocido simulado (Simulated Annealing). Este algoritmo, inspirado en los procesos físicos de recocido de metales, ha sido ampliamente utilizado para resolver problemas de optimización global debido a su capacidad de evitar mínimos locales y explorar de manera efectiva el espacio de búsqueda. A diferencia de otros métodos de optimización que tienden a quedarse atrapados en soluciones subóptimas, el recocido simulado introduce una estrategia probabilística que permite aceptar, bajo ciertas condiciones, soluciones con peor desempeño que la actual, con el fin de escapar de mínimos locales y así, eventualmente, converger hacia una solución globalmente óptima.

En esta práctica, hemos empleado el algoritmo de recocido simulado para resolver el problema de minimización de la función de Himmelblau. Esta función es una conocida referencia en la optimización, caracterizada por múltiples mínimos locales, lo que la convierte en un desafío ideal para evaluar la eficacia de los algoritmos de optimización global. La función de Himmelblau se expresa como $f(x, y) = 20 + 23.5x + 18.5y - 2.6x^2 - 2.3y^2 + 1.8xy$ y tiene varios puntos en los que el valor de la función se minimiza. El objetivo principal de esta práctica es aplicar el recocido simulado para identificar los puntos que minimizan dicha función en el intervalo definido de x para ambos parámetros.

El desarrollo de esta práctica también nos permite profundizar en el entendimiento del comportamiento del algoritmo de recocido simulado. Este tipo de algoritmo utiliza un parámetro denominado “temperatura” que se reduce gradualmente a medida que avanza el proceso de optimización. En las primeras etapas, la temperatura es alta, lo que permite una mayor libertad para aceptar soluciones de peor calidad, favoreciendo la exploración del espacio de búsqueda. Conforme la temperatura disminuye, el algoritmo se enfoca más en la explotación, es decir, en la mejora de soluciones locales cercanas, lo que incrementa la probabilidad de converger a un mínimo global.

En nuestro experimento, hemos utilizado una implementación de este algoritmo en Python, donde los resultados se visualizan a través de gráficos que muestran tanto la función de Himmelblau como el punto mínimo encontrado. Específicamente, se ha utilizado una representación gráfica mediante contornos, lo que facilita la interpretación de los resultados obtenidos y permite identificar visualmente los mínimos de la función. Adicionalmente, se evaluaron diferentes configuraciones del algoritmo, como la temperatura inicial, el factor de enfriamiento y el número de iteraciones, con el fin de analizar su impacto en el rendimiento y en la calidad de las soluciones encontradas.

Material y equipo.

Para esta práctica se usaron las siguientes herramientas de software y hardware necesarias para realizar la práctica.

Hardware

- Computadora.

Software

- Visual Studio Code.
- Python 3.

Desarrollo de la práctica.

La práctica 6 se centra en la aplicación del algoritmo de recocido simulado (Simulated Annealing) para minimizar la función de Himmelblau, una función matemática conocida por sus múltiples mínimos locales. El objetivo de esta práctica es demostrar cómo el algoritmo de recocido simulado es capaz de encontrar soluciones cercanas al mínimo global, evitando quedarse atrapado en mínimos locales que a menudo presentan dificultades para métodos de optimización convencionales. En este desarrollo, se realizó la implementación del algoritmo utilizando Python, y se ejecutó el código en el entorno de Google Colaboratory para obtener los resultados de la optimización.

Actividades desarrolladas

Durante el desarrollo de la práctica, se llevaron a cabo las siguientes actividades:

Función de Himmelblau

El primer paso consistió en definir la función de Himmelblau en el lenguaje Python. Esta función se expresa matemáticamente como:

$$F(x,y) = (x^2 + y - 11)^2 + (x + y^2 - 7)^2$$

Esta función posee cuatro mínimos locales ubicados en , , , y . Estos puntos mínimos tienen un valor de la función igual a 0. El desafío de la práctica es utilizar el recocido simulado para encontrar uno de estos puntos a partir de una posición inicial aleatoria dentro del rango .

```
# Definir la función de Himmelblau
def himmelblau(x, y):
    return (x**2 + y - 11)**2 + (x + y**2 - 7)**2
```

Figura 4 función de Himmelblau.

Implementación del algoritmo de recocido simulado

A continuación, se implementó el algoritmo de recocido simulado. Este algoritmo imita el proceso de recocido físico, en el cual un material se enfría lentamente desde una temperatura alta, permitiendo que los átomos se organicen en un estado de mínima energía. En el contexto de la optimización, el algoritmo simula este proceso al permitir, en las primeras etapas, la aceptación de soluciones de peor calidad (más “energía”) para fomentar la exploración global del espacio de búsqueda. Conforme avanza el algoritmo, la temperatura disminuye y se reduce la probabilidad de aceptar soluciones de peor calidad, lo que permite la convergencia hacia una solución óptima.

El algoritmo de recocido simulado fue implementado mediante la siguiente función principal:

```

# Función para el recocido simulado
def recocido_simulado(func, x0, y0, temp_inicial, temp_final, alpha):
    x, y = x0, y0
    temp = temp_inicial
    while temp > temp_final:
        # Generar un vecino
        x_new, y_new = vecino(x, y)

        # Calcular la diferencia de energía (cambio
# Calcular la diferencia de energía (cambio en la función objetivo)
        energia_actual = func(x, y)
        energia_nueva = func(x_new, y_new)
        delta_energia = energia_nueva - energia_actual

        # Si la nueva energía es menor, aceptar el nuevo estado
        if delta_energia < 0:
            x, y = x_new, y_new
        else:
            # Si la nueva energía es mayor, aceptarla con una probabilidad que depende de la temperatura
            probabilidad = math.exp(-delta_energia / temp)
            if random.uniform(0, 1) < probabilidad:
                x, y = x_new, y_new

        # Enfriar el sistema (disminuir la temperatura)
        temp *= alpha

    return x, y

```

Figura 5 Función recocido simulado.

- Func: Se refiere a la función objetivo que se desea minimizar, en este caso, la función de Himmelblau.
- X0, y0: Son los valores iniciales de x y y , que se seleccionan aleatoriamente dentro del rango.
- Temp_inicial y temp_final: Representan la temperatura inicial y final del algoritmo, respectivamente. La temperatura inicial es alta para permitir mayor libertad de exploración, y conforme disminuye, el algoritmo se enfoca más en la explotación de soluciones locales.
- Alpha: Es el factor de enfriamiento que controla la tasa de decrecimiento de la temperatura en cada iteración.
- Vecino(x, y): Es una función auxiliar que genera un nuevo punto cercano al actual, simulando el movimiento aleatorio en el espacio de búsqueda.
- Delta_energia: Representa la diferencia entre la energía de la nueva solución propuesta y la solución actual. Si esta diferencia es negativa, el nuevo punto es aceptado; si no lo es, se acepta con cierta probabilidad dependiente de la temperatura.

Ejecución y ajuste de parámetros

El siguiente paso consistió en la ejecución del algoritmo en Google Colaboratory, un entorno basado en la nube que permite ejecutar código Python de manera interactiva. Se establecieron parámetros como la temperatura inicial, el factor de enfriamiento y la temperatura final para optimizar la búsqueda. Los parámetros utilizados fueron los siguientes:

Temperatura inicial: 10,000

Temperatura final: 1×10^{-5}

Factor de enfriamiento (): 0.99

La elección de estos parámetros tuvo como objetivo garantizar que el algoritmo pudiera explorar el espacio de búsqueda de manera eficiente en las primeras etapas, evitando caer en mínimos locales, y posteriormente, explotar los alrededores de una buena solución.

Visualización y análisis de resultados

Tras la ejecución del algoritmo, los resultados obtenidos fueron graficados utilizando la librería matplotlib. Se generó un gráfico de contornos de la función de Himmelblau en el rango , donde se marcaron los puntos mínimos obtenidos por el algoritmo. Esto permitió realizar una verificación visual de que el algoritmo había encontrado un mínimo adecuado.

El algoritmo de recocido simulado fue capaz de encontrar con éxito valores cercanos a los puntos de mínimo de la función de Himmelblau. Dependiendo de los valores iniciales aleatorios y de los parámetros del algoritmo, se obtuvo un mínimo en alguno de los puntos conocidos de la función. Los resultados se presentan gráficamente, mostrando el mínimo encontrado en un gráfico de contornos, lo cual permite visualizar el comportamiento de la función en el espacio de búsqueda y el punto de mínima energía identificado.

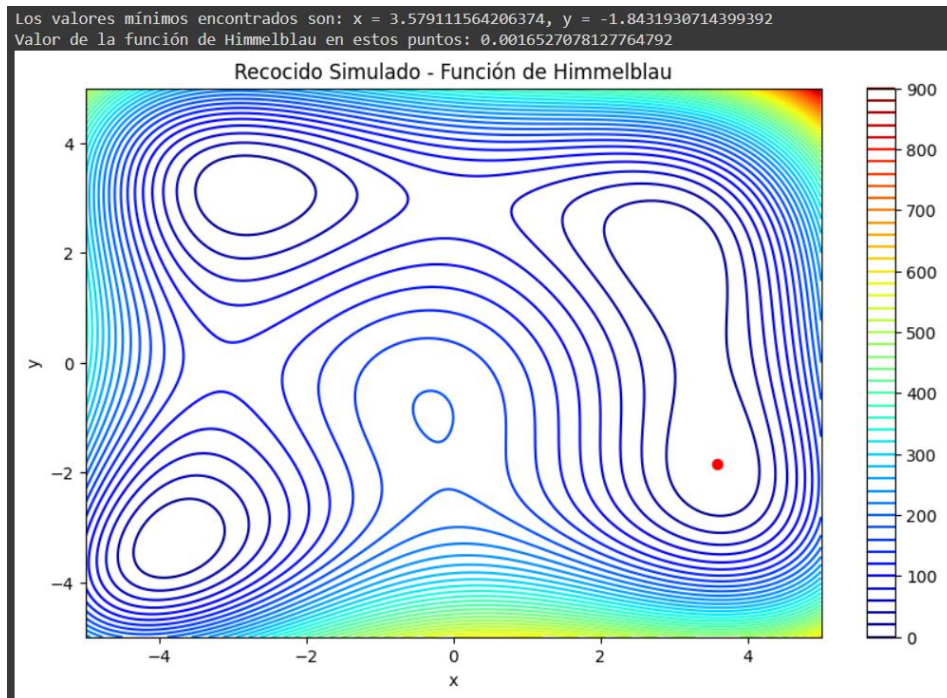
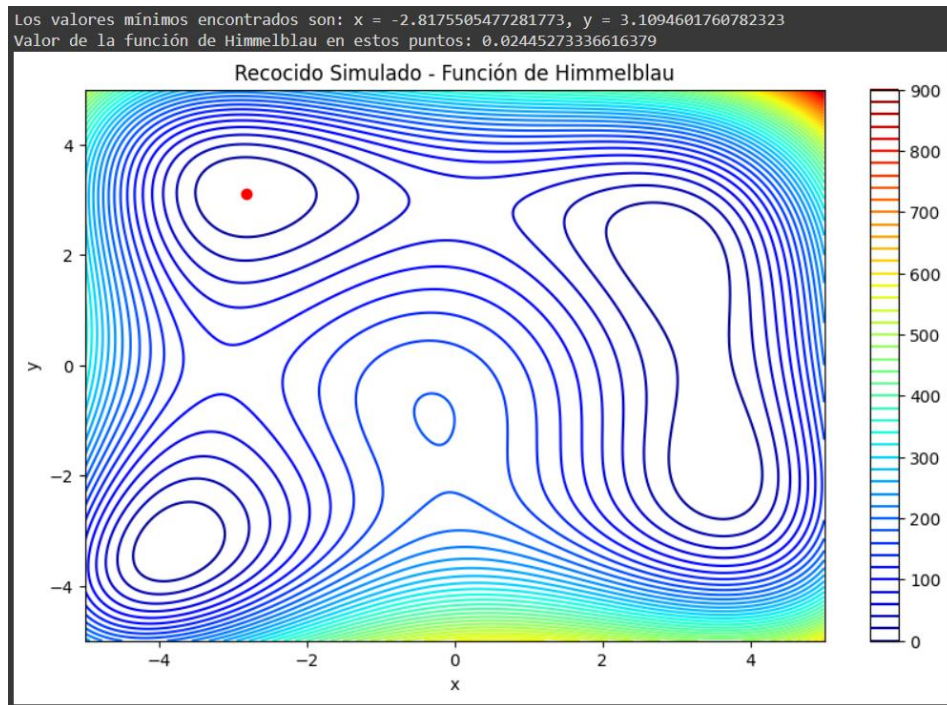
En términos generales, el recocido simulado demostró ser eficaz para abordar la optimización de la función de Himmelblau, encontrando soluciones cercanas al mínimo global a pesar de la presencia de múltiples mínimos locales.

Los valores encontrados son

1. Los valores mínimos encontrados son: $x = -2.8175505477281773$, $y = 3.1094601760782323$.
 - a. Valor de la función de Himmelblau en estos puntos:
0.02445273336616379.
2. Los valores mínimos encontrados son: $x = 3.579111564206374$, $y = -1.8431930714399392$
 - a. Valor de la función de Himmelblau en estos puntos:
0.0016527078127764792
3. Los valores mínimos encontrados son: $x = -3.780453070647157$, $y = -3.270338645296473$
 - a. Valor de la función de Himmelblau en estos puntos:
0.007744292523462435

- b. Los valores mínimos encontrados son: $x = 3.00192501327816$, $y = 2.005590386822577$
- c. Valor de la función de Himmelblau en estos puntos:
0.0008852786680910701

Resultados obtenidos



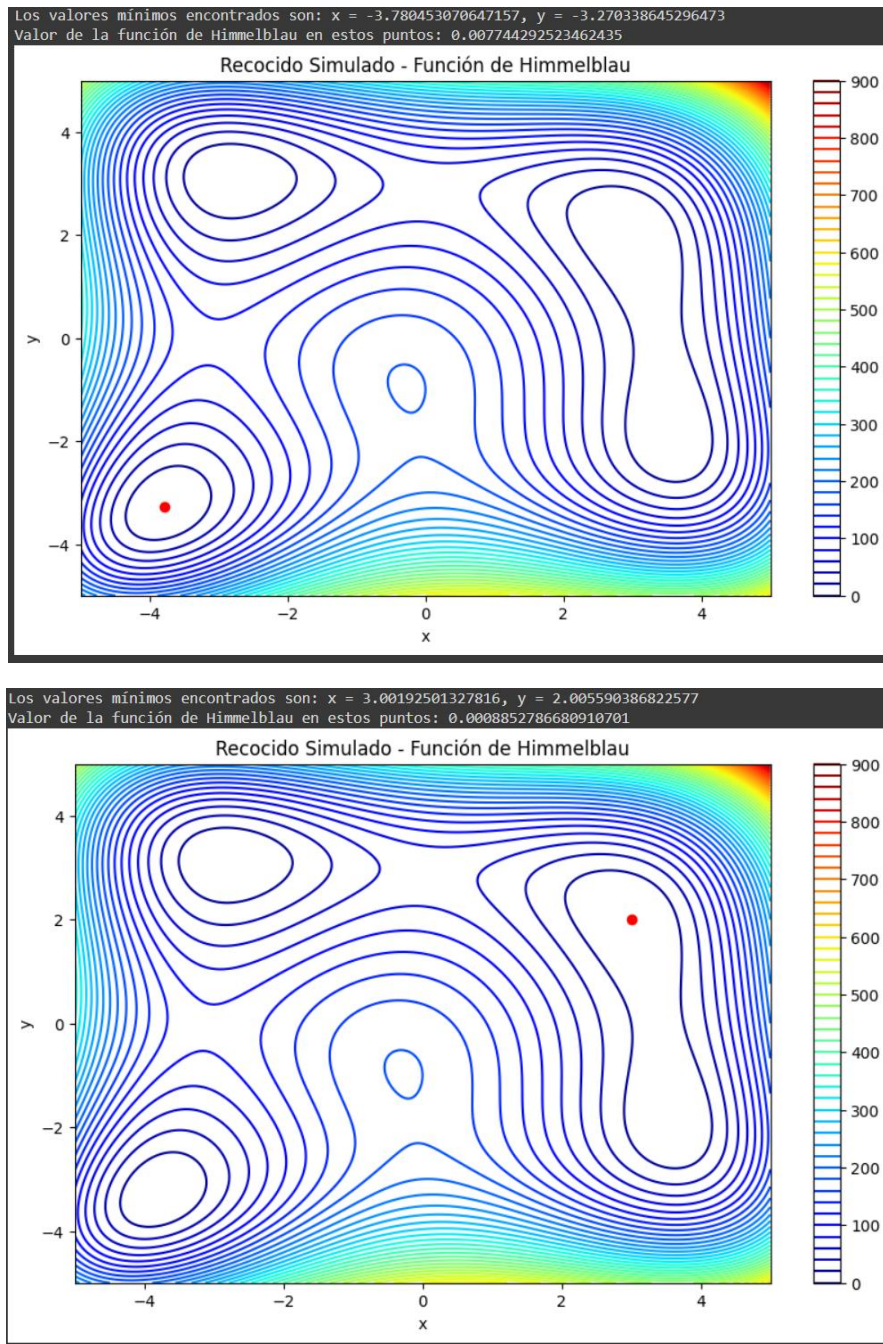


Figura 6 Resultados obtenidos.

Conclusiones

En esta práctica, hemos aprendido la aplicación práctica del algoritmo de recocido simulado en la búsqueda de soluciones óptimas para funciones con múltiples mínimos locales, como la función de Himmelblau. El uso de este algoritmo nos permitió explorar de manera eficiente el espacio de búsqueda, evitando quedar atrapados en soluciones subóptimas. Hemos comprendido cómo los parámetros del algoritmo, como la temperatura inicial y el factor de enfriamiento, influyen en el proceso de optimización y en la calidad de las soluciones obtenidas. Además, la visualización gráfica de los resultados nos ha permitido verificar la precisión del algoritmo, lo que resalta su utilidad en la resolución de problemas complejos de optimización global. En resumen, la práctica nos proporcionó una sólida comprensión de cómo ajustar y aplicar técnicas avanzadas de búsqueda informada para la resolución de problemas reales en entornos computacionales.

Referencias.

[1] 5_Informed_Search at main · GUSTAVOIVANGQ/AI. .

Código

https://github.com/GUSTAVOIVANGQ/AI/tree/main/5_Informed_Search