



**INSTITUTO POLITÉCNICO NACIONAL**

**ESCOM**

**INTELIGENCIA ARTIFICIAL**

**LAB 5: BUSQUEDA INFORMADA P2 FUNCION DE  
HIMMELBLAU**

**MARTÍNEZ CHÁVEZ JORGE ALEXIS**

**6CV3**

**10 OCTUBRE 2024**

## Introducción

La función de Himmelblau es una función matemática no convexa, famosa en el campo de la optimización y ampliamente utilizada para probar algoritmos de búsqueda de mínimos globales. Esta función es particularmente interesante debido a sus múltiples mínimos locales, lo que la convierte en un desafío para los métodos de optimización tradicionales, que a menudo pueden quedarse atrapados en estos mínimos locales en lugar de encontrar el óptimo global.

### Definición de la Función de Himmelblau

La función de Himmelblau está definida como:

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

## Código

Primero debemos de definir nuestra función de Himmelblau como una función que va a recibir como parámetros los valores de x, y.

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

Definimos nuestra función principal en donde debemos de pasar como parametros los valores de x, y así como los rangos, que en este caso es de -5 a 5 para ambas variables. Dentro de nuestra función definimos la cola que usaremos en donde iremos guardando los nodos a evaluar, así como los nodos que previamente han sido visitados. Vamos a crear una variable para poder guardar el mejor resultado encontrado y el mínimo valor.

```
def bfs_search_himmelblau(start_x, start_y, x_range=(-5, 5), y_range=(-5, 5), step=0.1):
    visited = set()
    q = queue.Queue()
    q.put((start_x, start_y))
    min_value = float('inf')
    best_coords = (start_x, start_y)
```

Iniciamos a hacer el ciclo, la primera se manda como parametros x=0, y=0, este ciclo se iniciara desde esa posición y se repetira siempre y cuando la cola de nodos frontera no este vacia, despues una vez entrado al ciclo, este nodo lo ingresaremos a la lista de nodos visitados y mandamos a evaluar la función. Si encontramos un valor menor entonces lo reemplazamos como el resultado y guardamos el nodo evaluado.

```
while not q.empty():
    x, y = q.get()

    if (x, y) in visited:
        continue

    visited.add((x, y))

    # Calculamos el valor de la función en el punto actual
    f_value = himmelblau(x, y)

    # Actualizamos el minimo si encontramos un valor menor
    if f_value < min_value:
        min_value = f_value
        best_coords = (x, y)
```

Por ultimo le generamos los nodos hijos, que en este caso, serán nodos vecinos, ya que serán posiciones que estén cercanas en el espacio de búsqueda, estos posibles puntos cercanos pueden ir hacia arriba o abajo; o hacia la derecha e izquierda, para poder hacer los 4 movimientos posibles, debemos de ir sumando o restando el paso,

en este caso el paso es de 0.1.

```
# Generamos los vecinos (nodos) en el espacio de búsqueda
neighbors = [
    (x + step, y),
    (x - step, y),
    (x, y + step),
    (x, y - step)
]

for nx, ny in neighbors:
    # Solo añadimos vecinos que estén dentro del rango permitido
    if x_range[0] <= nx <= x_range[1] and y_range[0] <= ny <= y_range[1]:
        q.put((nx, ny))

return best_coords, min_value
```

Una vez creado los nodos hijos, en caso de que no se encuentren en la lista de visitados, entonces debemos de agregarlos a los nodo frontera, en este caso como estamos haciendo una cola, entonces tendremos esta búsqueda en amplitud. Por ultimo solo imprimimos los resultados, tanto de las coordenadas como el valor minimo encontrado.

```
# Comenzamos la búsqueda desde un punto inicial
start_x = 0
start_y = 0
result = bfs_search_himmelblau(start_x, start_y)

print(f"Las coordenadas (x, y) que minimizan la función son: {result[0]}")
print(f"El valor mínimo de la función es: {result[1]}")
```

Hacemos la ejecución y nuestro resultado sera el siguiente.

```
PS C:\Users\jorge\OneDrive\Documentos\JORGE\ESCOM\9_SEMESTRE> & C:/Users/jorge/AppData/Local/Programs/Python/Python310/python.exe c:/Users/jorge/OneDrive/Documentos/JORGE/ESCOM/9_SEMESTRE/IA/LABORATORIO/5_BUSQUEDA_INFORMADA_2/himmelblau.py
Las coordenadas (x, y) que minimizan la función son: (3.000000000000001, 2.0)
El valor mínimo de la función es: 2.9187853493177437e-29
```

## Enlace

[https://github.com/Jorge300403/IA\\_6CV3\\_MartinezChavez/tree/main/5\\_BUSQUEDA\\_INFORMADA\\_2](https://github.com/Jorge300403/IA_6CV3_MartinezChavez/tree/main/5_BUSQUEDA_INFORMADA_2)