

**Josue Santana Robledo Corona 325073061**

**Maestría en Ciencias de la Robótica e Inteligencia Artificial**

**Algoritmos Bio-Inspirados**

**Centro Universitario de Ciencias Exactas e Ingeniería.**

**Mtro. Carlos Alberto López Franco**

## 1. Introducción

El algoritmo Firefly es un método de optimización bio-inspirado en el comportamiento de las luciérnagas, utiliza la atracción de las luciérnagas entre ellas y la aleatoriedad, es un algoritmo naturalmente para maximización, ya que la mejor luciérnaga es la que tenga mayor brillo, pero se puede adaptar para problemas de minimización haciendo una conversión a la función, haciéndola negativa funciona como debe.

## 2. Descripción del código

### Variables y parámetros iniciales

El algoritmo recibe los siguientes parámetros:

funcion\_objetivo: función a optimizar.

tamaño\_poblacion: número de luciérnagas.

dimension: número de dimensiones del espacio.

limites: tupla con límite inferior y superior.

iteraciones: número máximo de iteraciones.

alpha: coeficiente de movimiento aleatorio.

beta: coeficiente de atracción base.

gamma: coeficiente de absorción de luz.

minimizar: determina si se minimiza o maximiza.

## 3. Algoritmo Principal

El algoritmo completo sigue estos pasos:

    Inicializar las luciérnagas aleatoriamente.

    Evaluar su brillo (fitness).

    Para cada iteración:

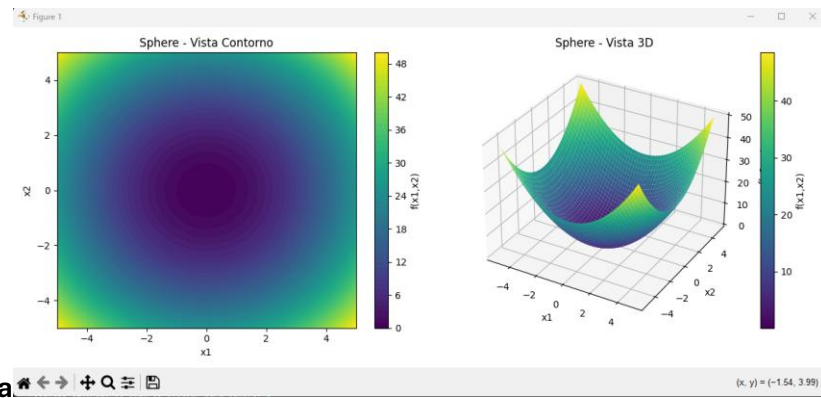
        Para cada pareja de luciérnagas (i, j):

- Si la luciérnaga j es más brillante que i → mover i hacia j.
- Actualizar atracción  $\beta$  dependiendo de la distancia.
- Agregar movimiento aleatorio.
- Recalcular fitness.

    Guardar el mejor valor encontrado.

## 4. Resultados Obtenidos

## Ejemplos en funciones de dos dimensiones



### 1. Esfera

Imagen 1. Grafica de Esfera

#### Minimización

Iteración 38/40 — Mejor valor: 0.000015

Iteración 39/40 — Mejor valor: 0.000012

Iteración 40/40 — Mejor valor: 0.000012

Mejor solución: [0.00267487 0.00226772]

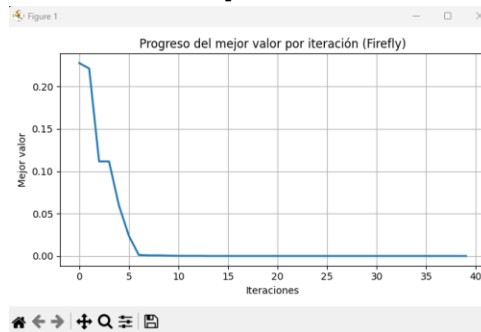


Imagen 2. Descendente Esfera

#### Maximización

Iteración 38/40 — Mejor valor: 50.000000

Iteración 39/40 — Mejor valor: 50.000000

Iteración 40/40 — Mejor valor: 50.000000

Mejor solución: [-5. 5.]

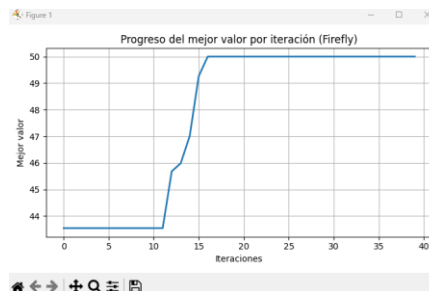
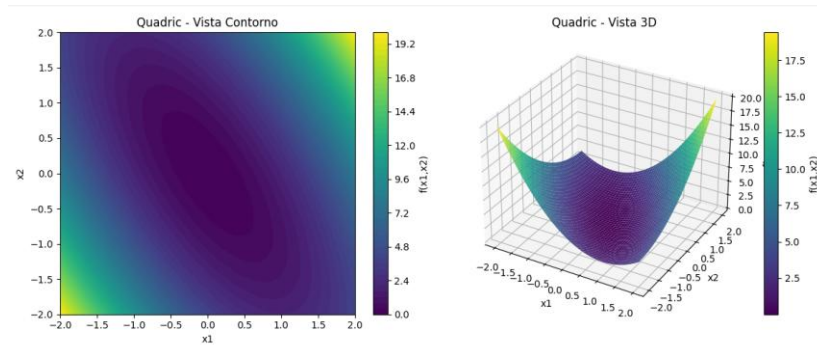


Imagen 3. Ascendente Esfera

## 2. Quadric



**Imagen 4. Grafica de Quadric**

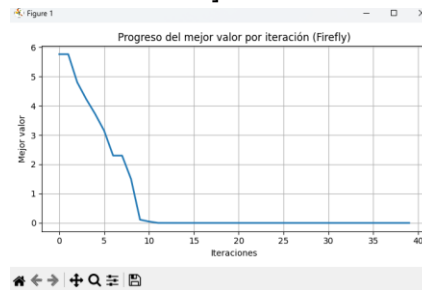
### Minimización

Iteración 38/40 — Mejor valor: 0.000012

Iteración 39/40 — Mejor valor: 0.000012

Iteración 40/40 — Mejor valor: 0.000012

Mejor solución: [-0.00019503 0.00363031]



**Imagen 5. Descendente Quadric**

### Maximización

Iteración 38/40 — Mejor valor: 125.000000

Iteración 39/40 — Mejor valor: 125.000000

Iteración 40/40 — Mejor valor: 125.000000

Mejor solución: [5. 5.]



**Imagen 6. Ascendente Quadric**

## 3. Ackley

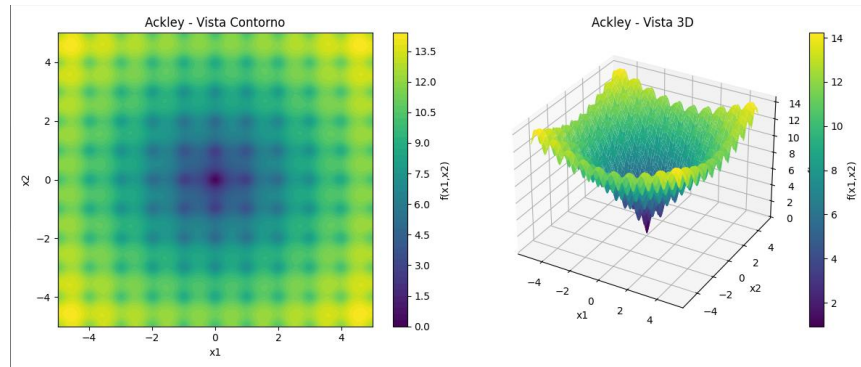


Imagen 7. Grafica de Ackley

### Minimización

Iteración 38/40 — Mejor valor: 3.579836

Iteración 39/40 — Mejor valor: 3.579836

Iteración 40/40 — Mejor valor: 3.579836

Mejor solución: [0.96111937 0.98064127]

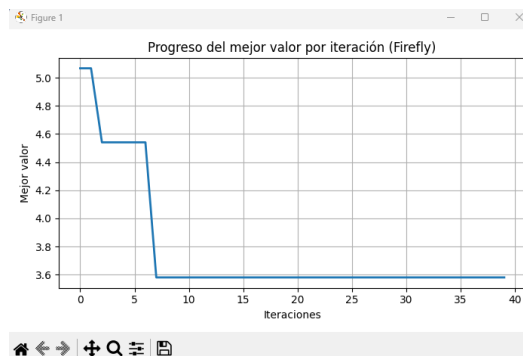


Imagen 8. Descendente Ackley

### Maximización

Iteración 38/40 — Mejor valor: 13.519501

Iteración 39/40 — Mejor valor: 13.519501

Iteración 40/40 — Mejor valor: 13.519501

Mejor solución: [-3.58790552 -4.61737045]

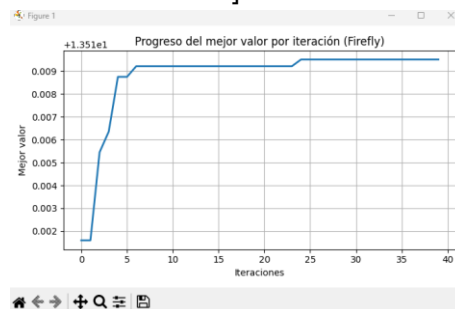


Imagen 9. Ascendente Ackley

## Ejemplos en funciones de n dimensiones (usaremos 4 dimensiones para los ejemplos)

### 1. Schwefel

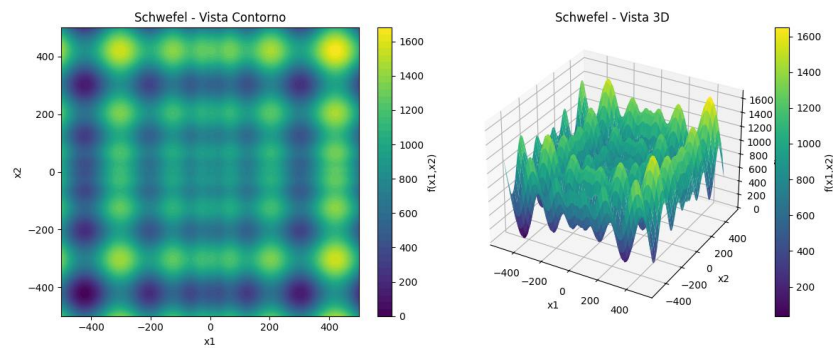


Imagen 10. Grafica de Schwefel

#### Minimización

Iteración 198/200 — Mejor valor: 1481.300178

Iteración 199/200 — Mejor valor: 1481.300178

Iteración 200/200 — Mejor valor: 1481.300178

Mejor solución: [-315.66008134 -213.74819557 -259.20135597 392.17808792 -260.0057636]

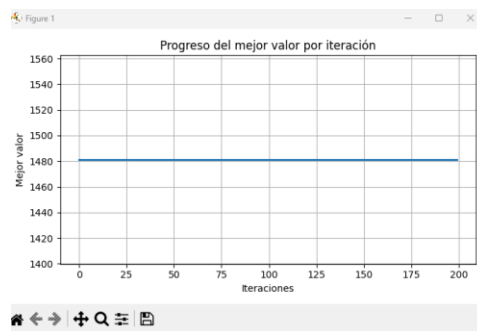


Imagen 11. Descendente de Schwefel

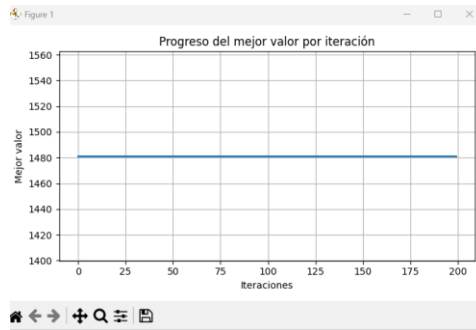
#### Maximización

Iteración 198/200 — Mejor valor: 3212.996167

Iteración 199/200 — Mejor valor: 3212.996167

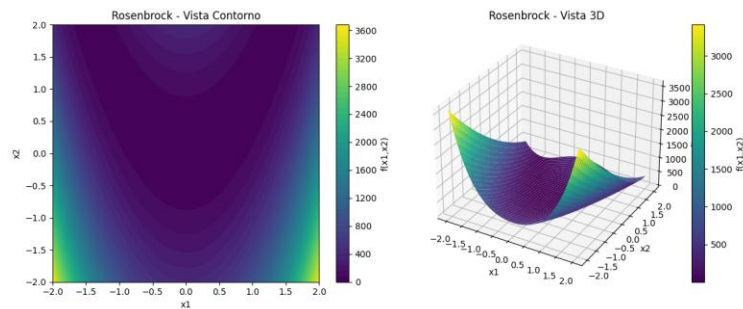
Iteración 200/200 — Mejor valor: 3212.996167

Mejor solución: [-479.71055819 -387.74692669 -401.85923975 40.17530836 -426.95516737]



**Imagen 12. Ascendente de Schwefel**

## 2. Rosenbrock



**Imagen 13. Grafica de Rosenbrock**

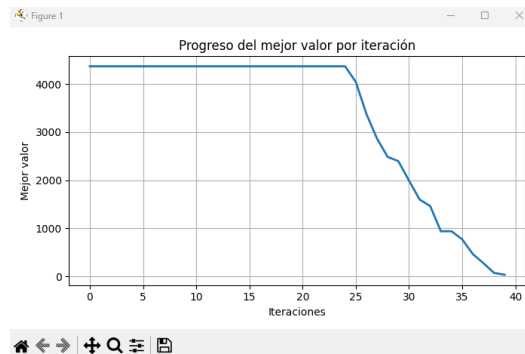
### Minimización

Iteración 38/40 — Mejor valor: 273.019491

Iteración 39/40 — Mejor valor: 70.128739

Iteración 40/40 — Mejor valor: 32.943215

Mejor solución: [1.39271017 1.38131538 1.84672222 3.47186003]



**Imagen 14. Descendente de Rosenbrock**

### Maximización

Iteración 38/40 — Mejor valor: 152854.302938

Iteración 39/40 — Mejor valor: 152854.302938

Iteración 40/40 — Mejor valor: 152854.302938

Mejor solución: [ 5. -5. 3.27488085 -1.76477018]

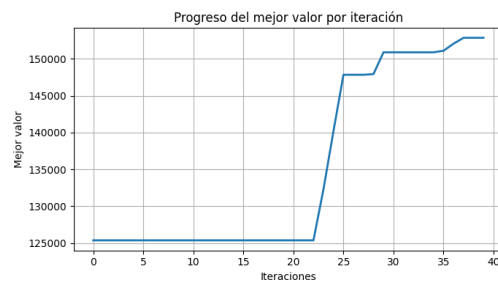


Imagen 15. Ascendente de Rosenbrock

### 3. Rosenbrock

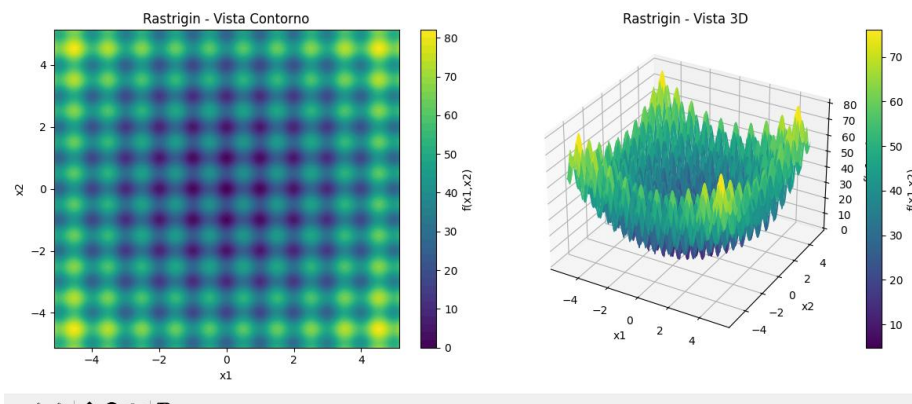


Imagen 13. Grafica de Rastrigin

#### Minimización

Iteración 38/40 — Mejor valor: 1.996673

Iteración 39/40 — Mejor valor: 1.996673

Iteración 40/40 — Mejor valor: 1.996673

Mejor solución: [-1.32365618 -0.19541443 0.08801736 -0.0371961]

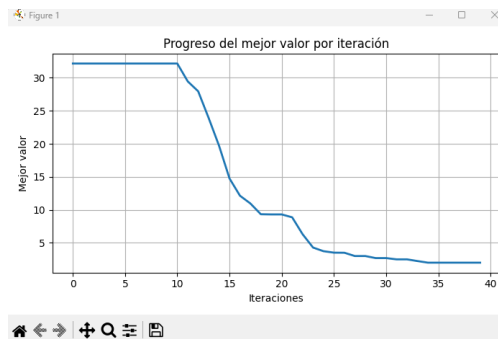


Imagen 14. Descendente de Rastrigin

#### Maximización



Iteración 38/40 — Mejor valor: 645.478386

Iteración 39/40 — Mejor valor: 661.713019

Iteración 40/40 — Mejor valor: 669.834431

Mejor solución: [ 0.6374788 4.19746892 -5. -4.83446656]



**Imagen 15. Ascendente de Rastringin**

## 5. Ventajas del Método

No requiere derivadas.

Combina exploración ( $\alpha$ ) y explotación ( $\beta$ ).

Evita caer fácilmente en mínimos locales gracias al movimiento aleatorio.

## 6. Limitaciones

Puede ser lento en espacios muy grandes

Depende de muchos parámetros

## 7. Conclusiones

El algoritmo Firefly es uno muy interesante, me parece que tiene la fórmula más difícil de recordar de todos los algoritmos, me parece que el algoritmo es flexible y muy potente, pero también creo que requiere un buen manejo de sus parámetros para su mejor funcionamiento. Me recuerda un poco al PSO, en cuestión de que se trata de un enjambre en el que todas comparten información, me imagino que existe una versión similar al PSO LBest en sentido de que se pueden hacer varios enjambres para explorar diferentes puntos, me alegro haber aprendido a usarlo para tenerlo como herramienta pero debe ser el algoritmo Metaheurístico que menos me ha gustado, no porque tenga nada de malo, sino porque siento que otros algoritmos como PSO o DE son más intuitivos en su dinámica y más fáciles de ajustar. También investigando sé que no es el algoritmo que mejor se desempeña en funciones de alta dimensionalidad, y me parece que se nota en el programa, a partir de 5 dimensiones la grafica se quedaba recta, y bajándole la dimensionalidad ya corría como debía.