

```

import numpy as np

# Parámetros del algoritmo
n_bats = 50
n_iterations = 5000
freq_min = 0
freq_max = 1
alpha = 0.9
gamma = 0.9

# Inicialización
bats = np.random.uniform(0, 100, (n_bats, 2))
velocities = np.zeros((n_bats, 2))
frequencies = np.zeros(n_bats)
loudness = np.ones(n_bats)
pulse_rates = np.ones(n_bats)

# Función objetivo
def objective_function(x):
    return -(x[0] + x[1])


# Restricciones
def constraint1(x):
    return 120*x[0] + 96*x[1] - 4000

def constraint2(x):
    return 300000*x[0] + 1000000*x[1] - 30000000

# Algoritmo de murciélago
for t in range(n_iterations):
    for i in range(n_bats):
        frequencies[i] = freq_min + (freq_max - freq_min) * np.random.uniform(0, 1)
        velocities[i] += (bats[i] - bats[np.argmin([objective_function(bat) for bat in bats])]) * frequencies[i]
        new_bat = bats[i] + velocities[i]
        if np.random.uniform(0, 1) > pulse_rates[i]:
            new_bat = bats[np.argmin([objective_function(bat) for bat in bats])] + np.random.uniform(-1, 1)
        if np.random.uniform(0, 1) < loudness[i] and constraint1(new_bat) <= 0 and constraint2(new_bat) <= 0:
            bats[i] = new_bat
        if objective_function(bats[i]) < objective_function(bats[np.argmin([objective_function(bat) for bat in bats])]) and constraint1(bats[i]) <= 0 and constraint2(bats[i]) <= 0:
            loudness[i] *= alpha
            pulse_rates[i] = 1 - (1 - pulse_rates[i]) * np.exp(-gamma)

# Imprimir la solución
print('Número máximo de equipos de fútbol patrocinados: ', int(bats[np.argmin([objective_function(bat) for bat in bats])][0]))
print('Número máximo de equipos de básquetbol patrocinados: ', int(bats[np.argmin([objective_function(bat) for bat in bats])][1]))

```


 Número máximo de equipos de fútbol patrocinados: 12
 Número máximo de equipos de básquetbol patrocinados: 26



Este es un algoritmo basado en el algoritmo de murcielago para resolver un problema en específico como el siguiente: The Really Big Shoe es un fabricante de calzado deportivo para básquetbol y fútbol. El gerente de marketing, Ed Sullivan, tiene que decidir la mejor forma de gastar los recursos destinados a publicidad. Cada uno de los equipos de fútbol patrocinados requiere 120 pares de zapatos. Cada equipo de básquetbol requiere 32 pares de zapatos. Los entrenadores de fútbol reciben \$300,000 por concepto de patrocinio para calzado, y los entrenadores de básquetbol reciben \$1,000,000. El presupuesto de Sullivan para promociones asciende a \$30,000,000. The Really Big Shoe dispone de una provisión limitada (4 litros, o sea, 4,000 centímetros cúbicos) de flubber, un compuesto raro y costoso que se utiliza en la fabricación del calzado atlético de promoción. Cada par de zapatos para básquetbol requiere 3 cc de flubber y cada par de zapatos de fútbol requiere 1 cc. Sullivan desea patrocinar el mayor número de equipos de básquetbol y fútbol que sus recursos le permitan.

- Formule un conjunto de ecuaciones lineales para describir la función objetivo y las restricciones.
- Utilice el algoritmo evolutivo de su elección.
- ¿Cuál es el número máximo de cada tipo de equipo que The Really Big Shoe podrá patrocinar?

Solución 1:

a) El planteamiento del problema de programación lineal sería:
Variables:

- x = Número de equipos de futbol a patrocinar
- y = Número de equipos de básquetbol a patrocinar

Función Objetivo:

$Z = \text{Maximizar } (x + y)$

Restricciones:

- Presupuesto: $300,000x + 1,000,000y \leq 30,000,000$
- Flubber: $120x + 96y \leq 4000$
- No negatividad: $x, y \geq 0$

b) Puedes ver el paso a paso de la elaboración del gráfico en este enlace.

El área de color azul representa la región factible y la línea de color rojo indica la función objetivo en su punto óptimo.*
*Los mismos colores se utilizarán para todos los problemas.
En el vértice D se tiene los valores máximos:

- $x = 700/57 = 12.28$
- $y = 500/19 = 26.32$
- $Z = 38.60$

c) Dado que el número de equipos no puede ser un valor decimal consideramos los siguientes valores:

Este es un algoritmo basado en el algoritmo de murcielago para resolver un problema en específico como el siguiente: The Really Big Shoe es un fabricante de calzado deportivo para básquetbol y fútbol. El gerente de marketing, Ed Sullivan, tiene que decidir la mejor forma de gastar los recursos destinados a publicidad. Cada uno de los equipos de fútbol patrocinados requiere 120 pares de zapatos. Cada equipo de básquetbol requiere 32 pares de zapatos. Los entrenadores de fútbol reciben 300, 000 *por concepto de patrocinio para calzado, y los entrenador* 1,000,000. El presupuesto de Sullivan para promociones asciende a \$30,000,000. The Really Big Shoe dispone de una provisión limitada (4 litros, o sea, 4,000 centímetros cúbicos) de flubber, un compuesto raro y costoso que se utiliza en la fabricación del calzado atlético de promoción. Cada par de zapatos para básquetbol requiere 3 cc de flubber y cada par de zapatos de fútbol requiere 1 cc. Sullivan desea patrocinar el mayor número de equipos de básquetbol y fútbol que sus recursos le permitan.

- Formule un conjunto de ecuaciones lineales para describir la función objetivo y las restricciones.
- Utilice el algoritmo evolutivo de su elección.
- ¿Cuál es el número máximo de cada tipo de equipo que The Really Big Shoe podrá patrocinar?

Solución 1:

a) El planteamiento del problema de programación lineal sería:
Variables:

- x = Número de equipos de futbol a patrocinar
- y = Número de equipos de básquetbol a patrocinar

Función Objetivo:

$Z = \text{Maximizar } (x + y)$

Restricciones:

- Presupuesto: $300,000x + 1,000,000y \leq 30,000,000$
- Flubber: $120x + 96y \leq 4000$
- No negatividad: $x, y \geq 0$

b) Puedes ver el paso a paso de la elaboración del gráfico en este enlace.

El área de color azul representa la región factible y la línea de color rojo indica la función objetivo en su punto óptimo.*
*Los mismos colores se utilizarán para todos los problemas. En el vértice D se tiene los valores máximos:

- $x = 700/57 = 12.28$
- $y = 500/19 = 26.32$
- $Z = 38.60$

c) Dado que el número de equipos no puede ser un valor decimal