



UNIVERSIDAD DE
GUADALAJARA

Red Universitaria e Institución Benemérita de Jalisco

CUCEI

CENTRO UNIVERSITARIO DE
CIENCIAS EXACTAS E INGENIERÍAS

Inteligencia Artificial

1

Practica 3 - Algoritmos de Optimización 3

Veloz Alcaraz Axel Abraham

Espinoza Sucilla Samuel

Cornejo Chavez Edwin Joel | 220791268 | 27-10-2023

LUIS ANGEL BELTRAN CARRILLO

DIEGO ALBERTO OLIVA NAVARRO



Objetivo:

Aquí se debe hacer la implementación del algoritmo metaheurístico elegido para dar solución a un problema “real”. Estos problemas deben ser elegidos por el equipo algunas sugerencias son:

- Procesamiento de imágenes
- Diseño de celdas solares
- Diseño de filtros digitales
- Planeación de rutas
- Planeación de actividades (scheduling)
- Problemas de diseño mecánico (resortes, tanques de presión, etc)
- Diseño de motores
- Juegos interactivos (tic tac toe, ajedrez, sudoku, etc)

Introducción:

Los algoritmos genéticos son utilizados en una variedad de aplicaciones, desde la optimización de funciones matemáticas hasta la resolución de problemas más complejos en áreas como el diseño de sistemas, la planificación y la optimización en ingeniería, entre otros. La fortaleza de los algoritmos genéticos radica en su capacidad para explorar grandes espacios de búsqueda y encontrar soluciones aceptables en problemas complejos y multidimensionales.

Por ello vamos a optar por hacer una implementación en un problema básico pero que puede ser escalable a situaciones mas complejas, el tema es encontrar la mejor combinación de ingredientes para hacer la receta perfecta de galletas, para esto vamos a definir la aptitud de una receta en función de su sabor y textura.

Antes de comenzar es importante aclarar en que enfoques similares podría ser útil:

- Optimización de Formulaciones de Productos Alimenticios
- En el campo de la gastronomía, chefs y expertos en alimentos podrían utilizar algoritmos genéticos para explorar nuevas combinaciones de ingredientes
- En el ámbito farmacéutico, se podría aplicar un enfoque similar para optimizar las formulaciones de medicamentos, ajustando las cantidades de los ingredientes para mejorar la eficacia y reducir los efectos secundarios.
- en la industria química para diseñar nuevas formulaciones de materiales con propiedades específicas.

Estas son solo algunas ya que este algoritmo es muy versátil aplicable a diversas situaciones

Desarrollo

Primeramente, se definieron los ingredientes disponibles (ingredientes_disponibles) para la receta de galletas, así como sus rangos de cantidad permitidos. Luego de esto Se creó una función de aptitud (aptitud_receta) que evalúa qué tan buena es una receta en función de su sabor y textura, posteriormente Se generó una población inicial de recetas de galletas (poblacion) de manera aleatoria, se utilizó un algoritmo genético para evolucionar la población de recetas a lo largo de varias generaciones. En cada generación, se evaluó la aptitud de cada receta, se seleccionaron las mejores recetas, y se cruzaron y mutaron para producir la siguiente generación, también se seleccionaron los padres para la reproducción utilizando una estrategia de selección de torneo, Se aplicó el operador de cruzamiento para crear nuevos individuos, se aplicó la mutación para introducir pequeñas variaciones en los descendientes, finalmente Se registraron los resultados se ejecutó el algoritmo con los parámetros definidos (tamaño de población, número de generaciones) y se observó la evolución de las recetas a lo largo del tiempo.

Aquí la solución a la problemática mencionada

```
import random
import matplotlib.pyplot as plt

# Definir los ingredientes disponibles y sus rangos de cantidad
ingredientes_disponibles = {
    'Harina': (100, 300),
    'Azúcar': (50, 150),
    'Mantequilla': (50, 200),
    'Huevos': (1, 3),
    'Chocolate': (0, 100),
    'Levadura': (1, 10),
}

# Función de aptitud: evalúa qué tan buena es una receta
def aptitud_receta(receta):
    sabor = receta['Azúcar'] + receta['Chocolate']
    textura = receta['Harina'] + receta['Mantequilla'] +
receta['Levadura']
    return sabor + textura

# Algoritmo genético
def algoritmo_genetico_mejorado(ingredientes_disponibles,
poblacion_tamano, generaciones):
    poblacion = [generar_receta(ingredientes_disponibles) for _ in
range(poblacion_tamano)]
    mejores_apitudes = []

    for generacion in range(generaciones):
        aptitudes = [aptitud_receta(receta) for receta in poblacion]
```

```

    mejores_apitudes.append(max(apitudes))

    mejores_recetas = [poblacion[i] for i in
sorted(range(len(apitudes)), key=lambda k: apitudes[k],
reverse=True)[:5]]

    poblacion = []
    for _ in range(poblacion_tamano):
        padre1, padre2 = random.sample(mejores_recetas, 2)
        hijo = {}
        for ingrediente in ingredientes_disponibles:
            hijo[ingrediente] = random.choice([padre1[ingrediente],
padre2[ingrediente]])
            hijo[ingrediente] += random.uniform(-10, 10)
            hijo[ingrediente] =
max(ingredientes_disponibles[ingrediente][0], min(hijo[ingrediente],
ingredientes_disponibles[ingrediente][1]))
        poblacion.append(hijo)

    # Introduce nuevos individuos aleatorios para mantener la
diversidad
    poblacion.extend([generar_receta(ingredientes_disponibles) for
_ in range(poblacion_tamano // 5)])

    mejor_receta = max(poblacion, key=aptitud_receta)
    print(f"Generación {generacion}: {mejor_receta} (Aptitud:
{aptitud_receta(mejor_receta)})")

    # Graficar resultados
    plt.plot(range(generaciones), mejores_apitudes)
    plt.title('Mejor Aptitud por Generación')
    plt.xlabel('Generación')
    plt.ylabel('Aptitud')
    plt.show()

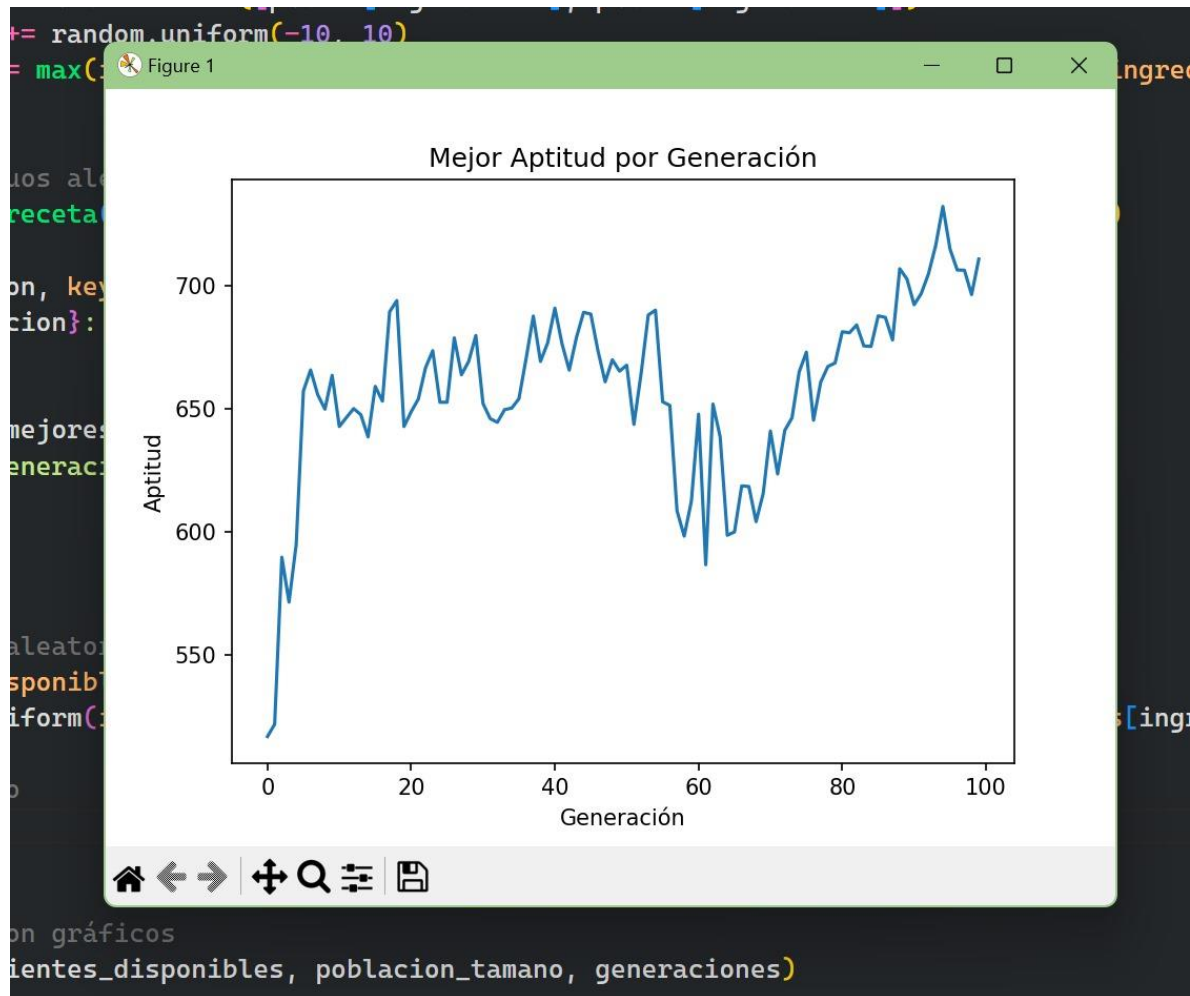
# Función para generar una receta aleatoria
def generar_receta(ingredientes_disponibles):
    return {ingrediente:
random.uniform(ingredientes_disponibles[ingrediente][0],
ingredientes_disponibles[ingrediente][1]) for ingrediente in
ingredientes_disponibles}

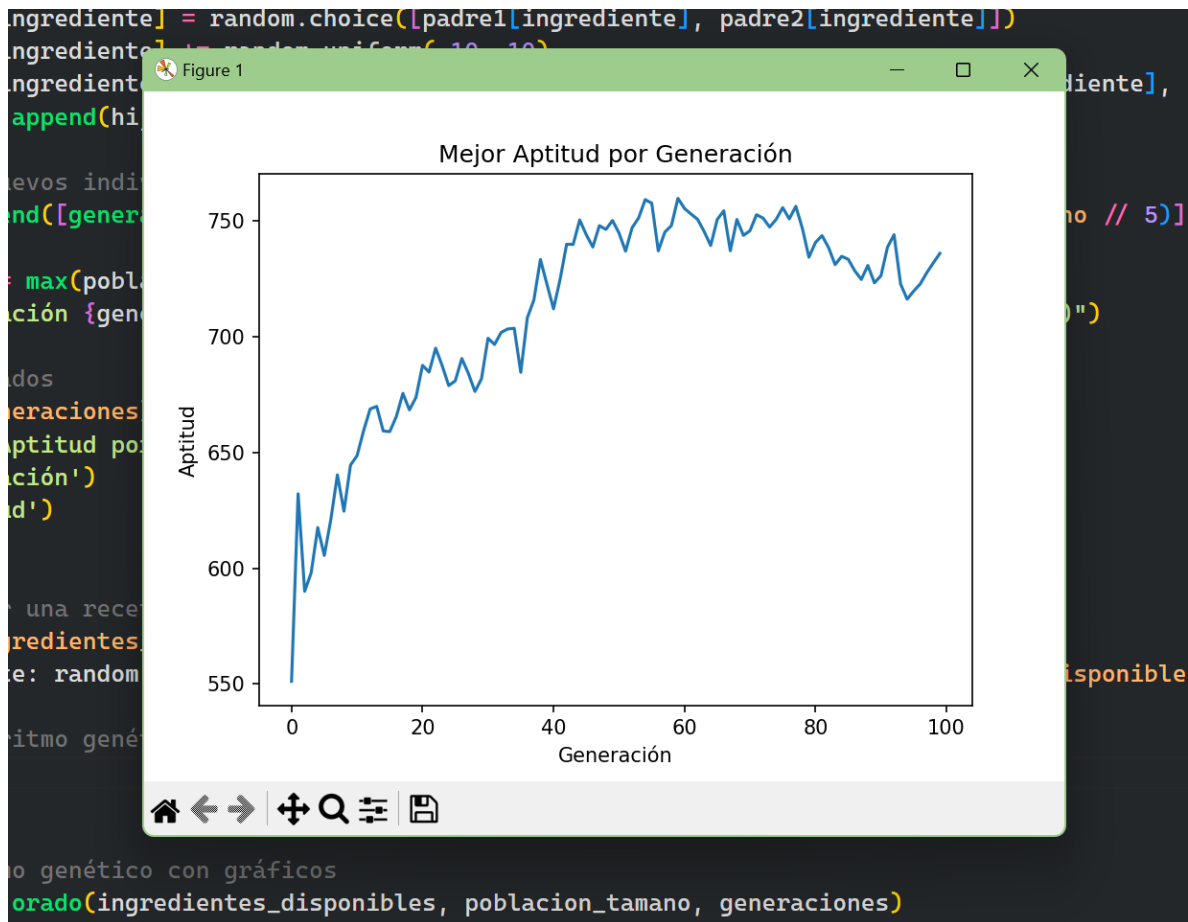
# Parámetros del algoritmo genético
poblacion_tamano = 6
generaciones = 100

```

```
# Ejecutar el algoritmo genético con gráficos  
algoritmo_genetico_mejorado(ingredientes_disponibles, poblacion_tamano,  
generaciones)
```

aquí algunas de las muestra de los resultados





Conclusión

En conclusión, el algoritmo genético para la receta perfecta de galletas es una herramienta eficiente y adaptable que emplea principios evolutivos para optimizar las combinaciones de ingredientes. A través de generaciones sucesivas, el algoritmo busca automáticamente las mejores recetas, permitiendo ajustes flexibles y continuas mejoras. Esta metodología es aplicable en diversas áreas, desde la industria alimentaria hasta la química, y destaca la capacidad de los algoritmos genéticos para abordar problemas de optimización y diseño en contextos del mundo real.