Modelación y simulación computacional basada en agentes
Práctica 3: Algoritmos evolutivos

Edgar Quiroz

12 de febrero de 2021

Resumen

Se presenta implementación de modelos de algoritmos genéticos y programación evolutiva para resolver problemas de otpmización de funciones continuas, de combinatoria y de juegos.

1. Algoritmo Genético Simple

Implementar el esquema visto en clase del Algoritmo Genético Simple para maximizar funciones de dos variables considerando los siguientes puntos

- Representación de los individuos: cadenas de bits
- Inicialización de la población: a través de un generador de números pseudo-aleatorios.
- Tamaño de la población: constante
- Condición de término: Dos opciones
  - Número fijo de generaciones
  - Cuando ya no haya mejoría de los individuos
- Mecanismo de selección: Proporicional a su aptitud (ruleta)
- Recombinación: 1-crossover
- Mutación: bitflip, misma probabilidad para todo el genotipo
- Operador de reemplazo: generacional, es decir, se renueva toda la población en la siguiente generación

Para encontrar el **máximo global** de la **función de Schwefel** (ecuación (1)) para $d = 2$ en el intervalo $(-500 \times 500)^2$. Las soluciones deberán tener al menos seis decimalaies de precisión.

$$f(x_1, \ldots, x_d) = \sum_{i=1}^{d} x_i \sin\left(\sqrt{|x_i|}\right) \tag{1}$$

1.1. Modelación

1.2. Simulación

Establezca una población fija de 100 individuos, con tasa de recombinación en $pc = 0{,}5$, tasa de mutación $pm = 0{,}1$ y condición de término al finalizar 500 generaciones.

- ¿Cuál fue el mejor individuo? Reporte la estrcutura del genotipo, fenotipo y fitness. ¿Es el máximo global?
- Grafique la evolución de la mejor aptitud, aptitud promedio y fitness promedio.
- Graique en el plano x, y los individuos de las generaciones 1, 10, 50, 100.

1.3. Exploración

Genere otros experimentos cambiando la tasa de recombinacioń y de mutación. Grafique los puntos anteriores para las diferentes configuraciones. ¿Con qué parámetros se encuentra más rápido el máximo global?

2. Las ocho reinas

El problema de las ocho reinas consiste en colocar ocho reinas en un tablero de ajedrez si que se puedan capturar entre sí en un movimiento. El problema fue propuesto en 1848 por Max Bezzel. Tiene 92 posibles soluciones, contando reflexiones y rotaciones.

Considere una solución con las siguientes características

- Representación: Permutación
- Recombinación: `cut & crossfit` con probabilidad 1
- Mutación: swap con probabilidad 0.8
- Selección de padres: mejores 2 de 5 aleatorios
- Operador de reemplazo: reemplazar a los peores

- Tamaño de la población: 100 individuos
- Inicialización: Aleatoria
- Condición de paro: Encontrar la solución o evaluación de 10,000 fitness.

A partir del planteamiento realizado en clase, resolver este problema a través de algoritmos evolutivos.

- Reporte la solución: genotipo, fenotipo y fitness.
- Grafique la evolución de la mejor aptitud y fitness promedio.
- Con su implementación resuelva el problema para $n = 30$ reinas.

3. Hormiga artificial

En 1991, David Jefferson y Robert Collins desarrollaron un mecanismo para "adaptar" un organismo artificail a un ambiente dado. El objetivo es conducir una hormiga artificial a través de un camino irregular formado por rastros de comida.

La hormiga artificial se desarrolla en una retícula toroide de 32×32 sobre el plano. La hormiga inicia su recorrido en la celda superior izquierda identificada con las coordenadas $(0, 0)$ viendo hacia el este. El sistema de coordenadas es (r, c).

El recorrido que intentará la hormiga se denomina "Santa Fe Trail", que consiste en 89 migajas de comida distribuidas sobre la retícula como se muestra en la figura k. Las celdas en negro con las migajas de comida y las celdas en amarillo son huecos donde se pierde el rastro. Exist huecos únicos, huecos dobles, huecos únicos en las orillas y huecos dobles en las orillas. Este rastros fue diseñado por Christopher Langton. Los números indican el conteo de las 89 migaas de comida.

La hormiga artificial tiene una visión limitada del mundo. En particular la horimga tiene un sensor con el cual solo puede ver lo que hau inmediatamente en la celda adyacente en la dirección que está viendo.

La hormiga puede ejecutar cualquiera de las siguientes acciones

- RIGTH: gira la hormiga 90° a la derecha, mateniendo la posición
- LEFT: gira la hormiga 90° a la izquierda, manteniendo la posición.
- MOVE: mueve la hormiga hacia adelante en la dirección que está "viendo". Cuando se mueve a la ceda sguiente se come la comida, eliminando el rastro de comida.

La hormaiga también cuenta con las siguiente funciones

- IF-FOOD-AHEAD: es un operador condicional de 2 argumentos. Ejecuta una acción dependiendo si hay o no comida.
- PROGN2: operador no condicional. Ejecuta dos instrucciones de forma secuencial.
- PROGN3: idem PROG2, pero con tres instrucciones.

El problema queda determinado por los siguientes puntos

- Objetivo: Encontrar un programa que controle el funcionamiento de la hormiga artificial tal que encuentre las 89 migajas de comida en el rastro.
- Conjunto terminal: LEFT, RIGHT,MOVE
- Conjunto función: IF-FOOD-AHEAD, PROGN2, PROGN3
- Casos de fitness: un solo caso
- Raw fitness: número de piezas de comida levantadas antes de 400 operaciones.
- Parámetros: población de tamaño 500 y 51 generaciones.

Usando programación genétca, encuentra la mejor estrategia para que la hormiga artificial se coma todas lasmigajas en un tiempo dada.

- Encontrar el programa más adecuado para resolver el problema.
- Graficar al evolución del "fitness" del mejor individuo y "fitness" promedio.