

Programación evolutiva

Curso 2023/2024

Práctica 3: Programación genética/Gramáticas evolutivas

El problema del cortacésped

En esta práctica vamos a resolver el problema del cortacésped (LawnMower, John Koza, 1994] utilizando **programación genética** (PG). El objetivo es encontrar un programa o expresión para que el cortacésped “pode” todo el césped de una determinada área.

El cortacésped puede ejecutar las siguientes instrucciones:

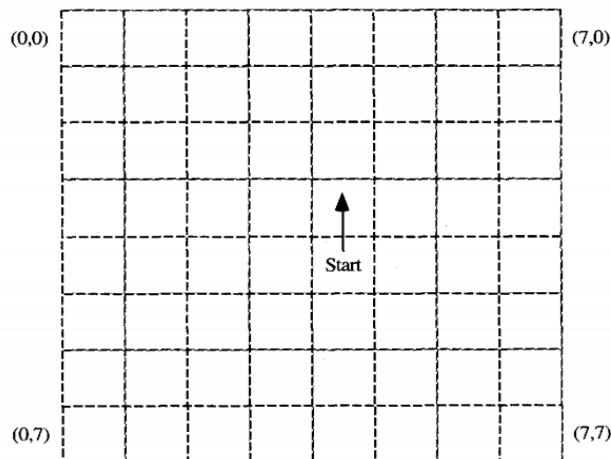
1. Adelante: Avanza un paso y “poda” la casilla a la que llega.
2. Izquierda: Gira 90° a la izquierda.
3. Salta: salta hasta un punto determinado en el césped, según un desplazamiento.

El césped es un área rectangular de $(n \times m)$ casillas, que están "conectadas" en los extremos (forma toroidal). En este ejemplo inicial utilizaremos un área de 8×8 casillas. El cortacésped debe cortar cada casilla de césped.



Cada casilla se identifica como un vector de enteros módulo 8 de la forma (i,j) , donde $0 \leq i,j < 8$.

El cortacésped empieza en la posición $(4,4)$ y mirando al norte. El estado del cortacésped incluye su posición y la dirección actual.



Definición de Funciones y Terminales

Lo primero que debemos hacer en un problema de Programación genética es definir los símbolos de función y terminales. En este ejemplo base usaremos los mismos símbolos que usa Koza en su problema original.

Terminales:

- **IZQUIERDA:** Gire el cortacésped a la izquierda 90°. Devuelve (0,0)
- **AVANZA:** Avanza una casilla en la dirección actual y corta el césped de la celda destino. Devuelve (0,0)
- Contante aleatoria (i,j) , que contiene un vector de dos enteros aleatorios módulo 8. Ejemplo: **(1 , 7)**

Funciones:

- **SUMA (V1 , V2):** Suma vectorial de argumentos módulo 8. Por ejemplo, **(SUMA (1,2) (3,7))** devuelve **(4 , 1)**
- **SALTA (V):** salta a una nueva posición en el césped determinada por el desplazamiento indicado en V. El cortacésped se mueva en la dirección en la que se encuentra actualmente en una cantidad especificada por su argumento y corta la hierba de la casilla destino.

Por ejemplo, si estamos en (1,2) mirando al este, **(SALTA (5 , 3))** hace que la cortadora termine en la ubicación **(6,5)** mirando hacia el este. SALTA actúa como el operador de identidad en su argumento, así, por ejemplo, **(SALTA (5, 3))** devuelve **(5 , 3)**.

- **PROGN (OP1, OP2) :** Ejecuta dos argumentos secuencialmente y devuelve el resultado de la última evaluación.

Ejemplo: **PROGN (SALTA (4,2) (AVANZA))**

Las constantes aleatorias y la función de suma de vectores son necesarias para proporcionar información para la función de salto.

Ejemplo de programa:

```
(SALTA (SUMA (SUMA (SALTA(0,4)) (SALTA(4,2)))  
      PROGN (PROGN (3,2) (AVANZA))  
      (SUMA (AVANZA)(1,4)))))
```

El objetivo final es cortar las 64 casillas mediante la ejecución de un programa o estrategia.

El cortacésped termina su trabajo cuando ha ejecutado un total de 100 giros a la izquierda o un total de 100 operaciones que causan movimiento (es decir, AVANZA o SALTA).

Procedimiento de resolución:

- Se genera una población inicial de programas aleatorios (árboles) usando el conjunto de funciones y terminales posibles. Estos árboles deben ser sintácticamente correctos. Limitamos la profundidad /mínima 2 máxima 5). Métodos Creciente, Completa y Ramped and half.
- Fitness: El fitness será es la cantidad de hierba (de 0 a 64) cortada dentro de este período de tiempo permitido.
- Operador de Cruce: Intercambiar dos subárboles (elegidos aleatoriamente) entre los dos árboles padres.
- Operadores de Mutación: al menos 3 de los vistos en el material de la asignatura.
- Algún método de controlar el bloating para evitar programas muy largos
- Se mostrarán las gráficas de evolución y la expresión final obtenida, junto con su fitness.
- Al final de la ejecución se mostrará el recorrido realizado por el cortacésped.

Opciones para mejorar nota:

- Escalar el problema a terrenos más grandes: 8x10 y 8x12, etc.
- Añadir obstáculos
- Optimizar las soluciones añadiendo nuevos terminales o funciones, por ejemplo:
 - **IF-DIRTY (A,B)** es un operador condicional de 2 argumentos que evalúa A si la casilla delante de cortacésped está sin podar; Si ha sido podada, evalúa B
 - **(repeat A,B)** , que ejecuta B A mod 8 veces.
- Otras ideas que mejoren la solución....

PARTE OPCIONAL:

- Resolver el mismo problema utilizando **gramáticas evolutivas**
- Se utilizarán los operadores tradicionales sobre cromosomas de enteros.
- Se podrá seleccionar el número de wraps
- Se ha de definir una gramática adaptada al problema

Ejemplo de posibles gramáticas de partida:

```
<e> ::= <op> | <x>
<op> ::= suma(<e>,<e>) | salta(<e>) | progn(<e>,<e>)
<x> ::= izquierda() | avanza() | cte

<start> ::= <op>
<op> ::= (progn2 <op> <op>) | (suma <op> <op>) | (salta <op>)
<op> ::= (avanza) | (izquierda) | cte
```

Normas de Entrega

- **Plazo de entrega: 26 de abril a las 12:00.** Se debe entregar mediante la tarea de entrega del Campus Virtual un archivo comprimido con el código java de la aplicación (**proyecto en Eclipse o NetBeans**) que incluya una breve memoria que contenga el estudio de las gráficas y los resultados obtenidos con cada función. Aquí se valorarán las conclusiones y observaciones que se consideren interesantes respecto al resultado obtenido.
- No olvidéis nombrar correctamente el proyecto e incluir en el código todas las librerías necesarias. El archivo comprimido y el nombre del proyecto Eclipse tienen que ser **GXXP3**, donde XX es el número de grupo.
- Debes incluir una memoria similar a las entregas anteriores e incluir al final de la memoria una breve descripción del **reparto de tareas** para reflejar lo que ha hecho cada miembro del grupo.
- La evaluación de la práctica se complementará con un test de evaluación en laboratorio el día viernes 26 de abril entre las 14:00 y las 16:00 h.

Importante:

- Si se implementa la parte opcional se considerará parte de la actividad adicional de la asignatura (la actividad adicional de la asignatura es hasta el 25%).
- Si no se implementa la parte opcional la calificación de las prácticas supondrá un 75% de la nota total de la asignatura y para obtener el 25% de nota por actividad adicional será necesario hacer un trabajo-memoria extra sobre algún tema de los propuestos en el Tema 8 de la asignatura o sobre cualquier algún artículo científico relacionado con la materia de la asignatura.

Ejemplo ejecución:

(SALTA (SUMA (SUMA (SALTA(0,4)) (SALTA(4,2)))
 PROGN (PROGN (3,2) (AVANZA))
 (SUMA (AVANZA)(1,4))))

