Facultad de Informática

Programación evolutiva

Curso 2024/2025

Práctica 2: Optimización de la Ruta de un Robot de Limpieza

Entrega: 21 de marzo

Parte A : 7 puntos

Diseñar e implementar un algoritmo evolutivo en Java que optimice la ruta de un robot de limpieza. El objetivo es encontrar el recorrido de menor distancia (mínimo coste) que permita al robot visitar/limpiar las 20 habitaciones de una mansión y regresar a la base, evitando los obstáculos distribuidos en un mapa representado por una cuadrícula.

Descripción del Problema

Se considera una casa modelada como una cuadrícula de 15×15 celdas. En dicha cuadrícula se ubica:

- **Base:** Punto de partida y retorno del robot (posición fija en el centro: (7,7)).
- **20 habitaciones:** Cada habitación tiene asignado un identificador (1 a 20), un nombre y coordenadas fijas dentro de la cuadrícula. La representación de la solución será una permutación de los identificadores de las habitaciones, que indica el orden en que el robot las visita.

Obstáculos: Algunas celdas del mapa contienen obstáculos que el robot no puede atravesar. Estos obstáculos se representarán mediante un carácter especial, por ejemplo ■

El robot debe planificar un recorrido que comience en la base, visite todas las habitaciones (en algún orden) y finalmente regrese a la base. La calidad de una solución (fitness) se medirá en función de la distancia total recorrida, considerando el camino óptimo (calculado con un algoritmo de búsqueda, por ejemplo, A*) entre cada par de puntos consecutivos en la ruta. Si en algún segmento no es posible encontrar un camino (por obstáculos, por ejemplo), se aplicará una penalización elevada para desalentar dicha solución.

Requisitos de la Práctica

1. Representación de la Solución

- **Cromosoma:** Cada individuo (solución) será una permutación de números del 1 al 20, donde cada número representa una habitación.
- **Punto de Partida y Regreso:** La base se fija en la posición (7,7) y no se incluye en la permutación; el recorrido se construye iniciando y finalizando en la base.

2. Función de Fitness básica (Fitness 1)

La función de fitness debe evaluar una solución según:

- **Distancia Total Recorrida:** Suma de las distancias desde la base a la primera habitación, entre habitaciones consecutivas y desde la última habitación de regreso a la base.
- **Penalizaciones:** Si por alguna razón en algún segmento no se encuentra un camino (por la presencia de obstáculos), se suma un valor de penalización muy alto para esa solución. Si se usa A* para evaluar esto nunca ocurrirá. Las penalizaciones se podrán probar en la parte B de la práctica.

3. Operadores Genéticos

- **Selección:** Ruleta, Torneos, Estocástico, Restos, Truncamiento y **Ranking**
- **Cruces:** Emparejamiento parcial (**PMX**), Dos Cruces por Orden (**OX y OXPP**), Ciclos (**CX**), Codificación Ordinal (**CO**), Recombinación de rutas (**ERX**) **y un cruce de invención propia.**
- **Mutación:** Inserción, Intercambio, Inversión, Heurística y una mutación de **invención propia**.

4. Interfaz y Visualización mínima

Debes utilizar y adaptar los parámetros del algoritmo y el tipo de gráficas de evolución que utilizaste en la Práctica 1.

Se puede incluir en la práctica cualquiera de las mejoras vistas en clase o en los apuntes.

El programa deberá mostrar:

- **El Mapa Inicial:** Imprimir en consola o en la interfaz una representación de la casa (por ejemplo, una cuadrícula de 15×15 con la base, las habitaciones (números) y los obstáculos (representados con un carácter visible, por ejemplo ■).
- **La Evolución del Fitness:** Dibujar la evolución del mejor fitness de cada generación, del fitness absoluto y del fitness medio a lo largo de las generaciones.
- **El Recorrido Óptimo:** Visualizar el camino del robot sobre el mapa, marcando las celdas del recorrido (por ejemplo, con el carácter * en las celdas vacías).

5. Consideraciones Adicionales

Coordenadas de las Habitaciones: Las 20 habitaciones se colocan en las siguientes posiciones fijas dentro del mapa:

- **Base:** Ubicación: (7,7)
- **Habitaciones:**

ID	Nombre	Símbolo	Coord (fila, columna)
1	Sala	1	(2,2)
2	Cocina	2	(2,12)
3	Dormitorio	3	(12,2)
4	Baño	4	(12,12)
5	Comedor	5	(2,7)
6	Oficina	6	(7,2)
7	Estudio	7	(7,12)

8	Lavandería	8	(12,7)
9	Terraza	9	(0,7)
10	Garaje	A	(7,0)
11	Jardín	B	(14,7)
12	Sótano	C	(7,14)
13	Balcón	D	(0,0)
14	Despacho	E	(0,14)
15	Vestíbulo	F	(14,0)
16	Cuarto de lavado	G	(14,14)
17	Sala de estar	H	(4,4)
18	Sala de juegos	I	(4,12)
19	Sala de TV	J	(10,4)
20	Sala de reuniones	K	(10,12)

Ubicación de los Obstáculos: Se colocan obstáculos en las siguientes posiciones (carácter ■):

- **Pared horizontal:** Fila 5, columnas 5 a 9
  (Celdas: (5,5), (5,6), (5,7), (5,8), (5,9))
- **Pared vertical:** Columna 10, filas 8 a 12
  (Celdas: (8,10), (9,10), (10,10), (11,10), (12,10))
- **Obstáculos individuales:** (10,3) (11,4)
- **Obstáculo adicional – pared vertical:** Columna 6, filas 10 a 13
  (Celdas: (10,6), (11,6), (12,6), (13,6))
- **Obstáculo adicional – pared horizontal:** Fila 8, columnas 1 a 4
  (Celdas: (8,1), (8,2), (8,3), (8,4))
- **Obstáculos en esquina superior derecha:** (0,13) (1,13)
- **Obstáculo adicional – pared horizontal:** Fila 3, columnas 8 a 11
  (Celdas: (3,8), (3,9), (3,10), (3,11))

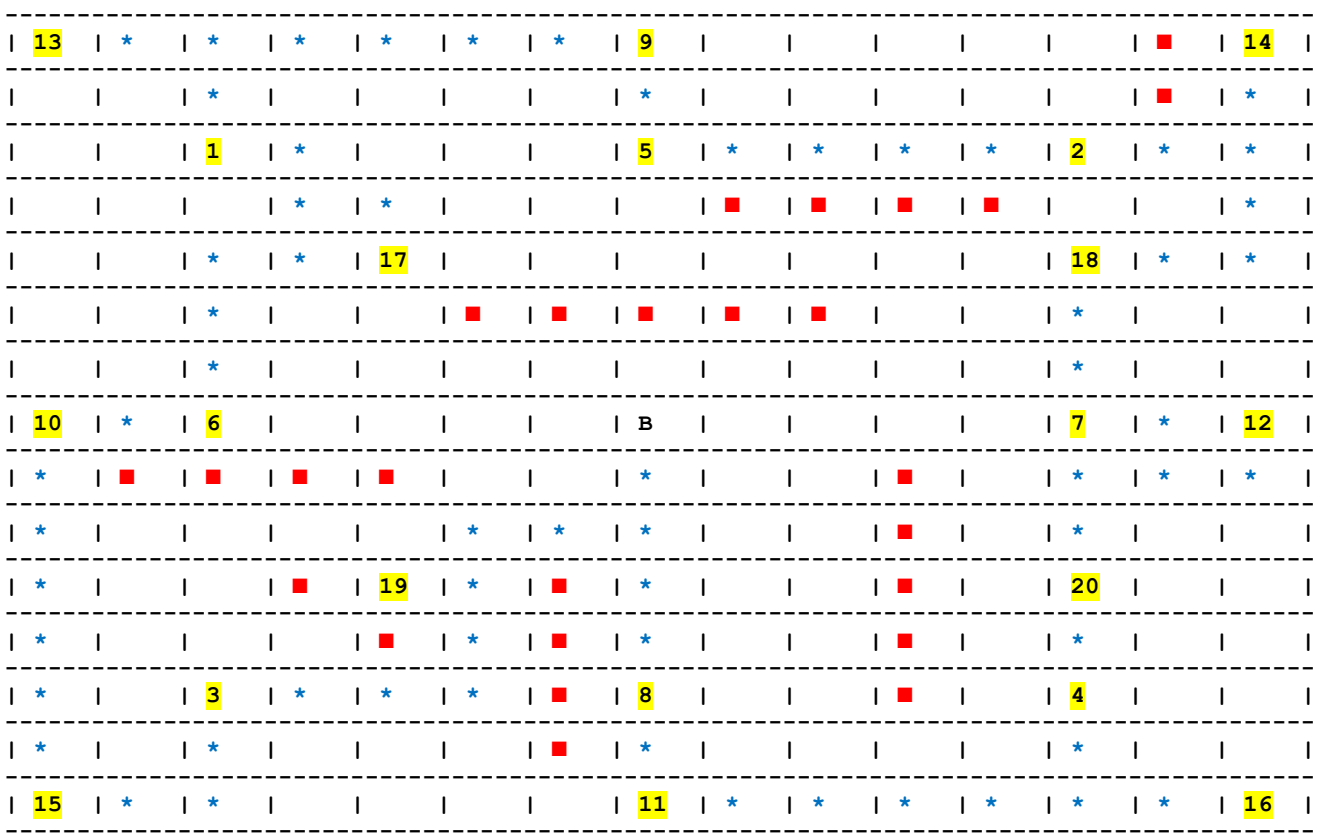
Ejemplo de salida

A continuación, se muestra un ejemplo básico de representación de la solución:

Mejor ruta encontrada:

Fitness = 92.0

Ruta: 19→3→15→10→6→17→1→13→9→5→2→14→18→7→12→20→4→16→11→8



- Las celdas en blanco representan zonas transitables.
- Las letras y números indican la ubicación de la base y las habitaciones.
    - **B** : Base central (inicio/fin)
    - **1-20** : Habitaciones numeradas
- ■ : Obstáculos/paredes
- * : Ruta óptima del robot

Parte B : 3 puntos

6. Aspectos adicionales para obtener máxima calificación

Visualización gráfica:

Se puede mejorar la visualización de la casa y el recorrido en la propia interfaz gráfica, para mostrar la casa y la evolución del recorrido del robot, en lugar de solo imprimir la cuadrícula en consola.

Añadir nuevas funciones de fitness

Manteniendo siempre la función de fitness original (fitness 1), se pueden añadir funciones de fitness nuevas que hagan que el algoritmo no solo minimice la distancia total, sino que también favorezca rutas "más eficientes" o "más suaves", se pueden incluir varios criterios adicionales. De este modo **se podrá seleccionar** desde la interfaz la función o funciones de de fitness utilizadas.

Algunas ideas de nuevas funciones de fitness (que obviamente deben incluir el fitness 1 original):

Penalización por proximidad a obstáculos:
Si el camino pasa muy cerca de un obstáculo, se puede sumar una penalización extra para evitar rutas que, en la práctica, tengan más riesgo. Se podría integrar esta penalización directamente en el costo g de A*.

Penalización por giros o cambios de dirección:
Cada vez que el robot cambia de dirección, se podría sumar una penalización al fitness original. Esto incentivar rutas que tengan trayectos más rectos.

Uso de una heurística más sofisticada:
Por ejemplo, reemplazar la distancia Manhattan por una heurística mejorada, una que sume un pequeño costo por la densidad de obstáculos en la dirección del objetivo.

Costo de velocidad o tiempo:
Además de la distancia, se podría simular un "tiempo de recorrido" en función de la velocidad y los cambios de dirección, penalizando rutas que requieran más tiempo.

Comparación con la distancia euclidiana:
Se puede calcular la distancia en línea recta entre dos puntos (distancia euclidiana) y, si la ruta encontrada es significativamente mayor que esa distancia, agregar una penalización proporcional a la desviación.

Multicriterio (Multiobjetivo):
Integrar varios de los puntos anteriores para obtener una función de fitness que combine la distancia, el número de giros y la cercanía a obstáculos.

Entrega

- **Plazo de entrega: 21 de marzo a las 16:00**. Debes 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. 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 **GXXP2**, donde XX es el número de grupo. Ejemplo nombre del proyecto-archivo: **G01P2** (por ejemplo, para el grupo 01).

- ❑ En el archivo comprimido se incluirá una muy breve memoria con una portada con el nombre de los integrantes del grupo y el número de grupo. La memoria deberá contener:
    - Ejemplo de las 5 mejores ejecuciones indicando el individuo solución, aptitud máxima, mínima, media, total de cruces, total de mutaciones y Gráficas de evolución.
    - Visualización del recorrido del robot de limpieza correspondiente al individuo solución.
    - Conclusiones y descripción de algunos detalles de la implementación y arquitectura de la aplicación.
    - Al final de la memoria una breve descripción del **reparto de tareas** para reflejar lo que ha hecho cada miembro del grupo.

- ❑ La corrección de la práctica tiene dos fases o partes:
    - **Corrección del código y la memoria** verificando que funciona correctamente e incluye todos los apartados pedidos.
    - Realización de un **cuestionario en el Campus Virtual** sobre la propia práctica y sobre cualquier concepto relacionado con la misma. Es importante conocer bien la práctica y los aspectos teóricos en los que se basa. Se realizará presencialmente el **miércoles 26 de marzo en el laboratorio 9.**
    - Para aprobar la práctica será necesario aprobar ambas partes, con nota mayor o igual de 5.

- ❑ Se realizará control de copias de todas las prácticas, comparando las entregas de todos los grupos de PE. Si se detecta algún tipo de copia sin justificar se calificará como suspenso.

Plan de entrega y corrección

- **Viernes 21 de marzo, 16:00**: Entrega del código + memoria mediante tarea del campus virtual
- **Miércoles 26 de marzo, 14:00**: Cuestionario Campus Virtual, Laboratorio 9