Inteligencia Artificial
Práctica 4: Recocido simulado

Benjamin Torres Saavedra

19 de febrero de 2019

1. Objetivo

Que el alumno se familiarice con una estrategia de mejoramiento iterativa para la resolución de problemas conocida como recocido simulado.

2. Introducción

2.1. Recocido simulado

Este algoritmo puede ser visto como una mejora al algoritmo de ascenso de colinas, el cual mejora de manera iterativa una representación de la solución a un problema hasta que se encuentre el óptimo o se estanque en un máximo local. Recocido simulado integra una selección de soluciones estocastica, es decir, para elegir la siguiente solucion no siempre se escoge la mejor, con lo cual se le da la libertad de explorar zonas distintas del espacio de soluciones, en las cuales el ascenso de colinas puede detenerse fácilmente.

El pseudocódigo de este algoritmo es descrito en el libro Artificial Intelligence: A Modern Approach en la sección de algoritmos de mejoramiento iterativo y lo podemos ver en la figura 2.1:

```
function SIMULATED-ANNEALING(problem, schedule) returns a solution state
  inputs: problem, a problem
          schedule, a mapping from time to "temperature"
  static: current, a node
          next, a node
          T, a "temperature" controlling the probability of downward steps

  current ← MAKE-NODE(INITIAL-STATE[problem])
  for t ← 1 to ∞ do
      T ← schedule[t]
      if T=0 then return current
      next ← a randomly selected successor of current
      ΔE ← VALUE[next] - VALUE[current]
      if ΔE > 0 then current ← next
      else current ← next only with probability e^(ΔE/T)
```

Figura 1: pseudocódigo del recocido simulado

2.2. Problema del agente viajero (TSP)

Este problema consiste en encontrar, dada una lista de ciudades una ruta que pueda seguir un agente para recorrer todas las ciudades y volver a aquella en la cual inicio el viaje. Para que este recorrido valga la pena debe ser lo mas económico posible, o en su defecto recorrer la menor distancia que permita visitar todas las ciudades.

Este problema es ampliamente conocido por ser de la clase NP-Completo, es decir, que no se puede resolver en un tiempo polinomial por algoritmos deterministas, afortunadamente para esta práctica no usaremos un algoritmo de tal tipo y podremos aproximarnos a una solución en un tiempo razonable

3. Desarrollo e implementación

Se les proporcionará código fuente en Java que deberán completar con la finalidad de resolver el problema del agente viajero.

El código fuente se encuentra en los archivos Solucion.java, que se encargará de representar a una ruta del agente viajero que pase por todas las ciudades y el archivo RecocidoSimulado.java, que implementará el algoritmo anteriormente descrito.

Los métodos a implementar dentro de una clase hija a Solución son:

- public Solucion()

  Este método debera inicializar un representacion a las solucion de un problema, en nuestro caso una ruta del problema del agente viajero.

- public Solucion siguienteSolucion()

  Genera una nueva solución perturbando de manera aleatoria al objeto que lo llama.

- public float evaluar()

  Asigna un valor a la solución actual.

Para la clase RecocidoSimulado:

- public RecocidoSimulado()

  Inicializa los parametros del algoritmo, puedes modificar la firma de este método.

- public float nuevaTemperatura(float temperatura,float decaimiento)

  Calcula la nueva temperatura, se espera que a lo largo de las iteraciones este valor decrezca.

- public Solucion seleccionarSiguienteSolucion(Solucion s) Dada una solución, este método debe modificar al objeto pasado como parametro para obtener una solución nueva y elegir con cierta probabilidad esta nueva solución si es que está es mejor.

- public Solucion ejecutar()

  Ejecuta el algoritmo con los parametros inicializados, puedes modificar la firma si lo consideras necesario.

Finalmente, en la clase Main únicamente se crea un objeto RecocidoSimulado que ejecute por algún número de iteraciones el algoritmo.

En la pagina `http://www.math.uwaterloo.ca/tsp/world/countries.html#DJ` se pueden encontrar una serie de paises con ciudades y sus correspondientes coordenadas, si asumes una conectividad total de las ciudades y usas la distancia euclideana como métrica, tu tarea será tratar de aproximarte lo mejor posible a la longitud del camino óptimo de la ciudad seleccionada, puedes incorporar la información a tu programa como te sea más conveniente.

4. Punto extra

Realiza las modificaciones que consideres pertinentes para poder utilizar una estrategia similar al recocido simulado para minimizar la siguiente función:

$f(x,y) = -20e^{[-0,2\sqrt{0,5(x^2+y^2)}]} - e^{[0,5(cos2\pi x+cos2\pi y)]} + e + 20$

para $-5 \leq x, y \leq 5$

5. Requisitos y resultados

El código debe ser implementado de manera eficiente y estar documentado para esclarecer su funcionamiento. Ademas, debe encontrar una solución válida al problema del agente viajero.