|  |
| --- |
| Universidad de Granada |
| Problema del Viajante de Comercio |
| Algorítmica |
| **Aarón Bueno Rodríguez**  **Bryan Moreno Picamán**  **Miguel Ángel Rodríguez Serrano** |
|  |
|  |

|  |
| --- |
| 2º Grado Ingeniería Informática |

El problema del viajante de comercio (TSP, por Traveling Salesman Problem) se define como sigue: dado un conjunto de ciudades y una matriz con las distancias entre todas ellas, un viajante debe recorrer todas las ciudades exactamente una vez, regresando al punto de partida, de forma tal que la distancia recorrida sea mínima.

Mas formalmente, dado un grafo G, conexo y ponderado, se trata de hallar el ciclo hamiltoniano de mínimo peso de ese grafo.

Una solución para TSP es una permutación del conjunto de ciudades que indica el orden en que se deben recorrer. Para el cálculo de la longitud del ciclo no debemos olvidar sumar la distancia que existe entre la última ciudad y la primera (hay que cerrar el ciclo).

Por su interés teórico y práctico, existe una variedad muy amplia de algoritmos para abordar la solución del TSP y sus variantes (siendo un problema NP-Completo, el diseño y aplicación de algoritmos exactos para su resolución no es factible en problemas de cierto tamaño). Nos centraremos en una serie de algoritmos aproximados de tipo greedy y evaluaremos su rendimiento en un conjunto de instancias del TSP. Para el diseño de estos algoritmos, utilizaremos dos enfoques diferentes: a) estrategias basadas en alguna noción de cercanía, y b) estrategias

de inserción.

En el primer caso emplearemos la heurística del vecino más cercano, cuyo funcionamiento es extremadamente simple: dada una ciudad inicial v0, se agrega como ciudad siguiente aquella vi (no incluida en el circuito) que se encuentre más cercana a v0. El procedimiento se repite hasta que todas las ciudades se hayan visitado.

En las estrategias de inserción, la idea es comenzar con un recorrido parcial, que incluya algunas de las ciudades, y luego extender este recorrido insertando las ciudades restantes mediante algún criterio de tipo greedy. Para poder implementar este tipo de estrategia, deben definirse tres elementos:

1. Cómo se construye el recorrido parcial inicial.

2. Cuál es el nodo siguiente a insertar en el recorrido parcial.

3. Dónde se inserta el nodo seleccionado.

El recorrido inicial se puede construir a partir de las tres ciudades que formen un triángulo lo más grande posible: por ejemplo, eligiendo la ciudad que está más a Este, la que está más al Oeste, y la que está más al norte.

Cuando se haya seleccionado una ciudad, ésta se ubicará en el punto del circuito que provoque el menor incremento de su longitud total. Es decir, hemos que comprobar, para cada posible posición, la longitud del circuito resultante y quedarnos con la mejor alternativa. Por último, para decidir cuál es la ciudad que añadiremos a nuestro circuito, podemos aplicar el siguiente criterio, denominado inserción más económica: de entre todas las ciudades no visitadas, elegimos aquella que provoque el menor incremento en la longitud total del circuito.

En otras palabras, cada ciudad debemos insertarla en cada una de las soluciones posibles y quedarnos con la ciudad (y posición) que nos permita obtener un circuito de menor longitud.

Seleccionaremos aquella ciudad que nos proporcione el mínimo de los mínimos calculados para cada una de las ciudades.

|  |
| --- |
| Elementos de Greedy |

Greedy proporciona varios elementos que facilitan la implementación y su correcto uso. Para este problema, del cual tenemos tres estrategias diferentes, obtenemos todos los elementos.

A continuación pasamos a describirlos:

* **Conjunto de Candidatos**: Representa al conjunto de posibles decisiones que se pueden tomar en cada momento
* **Conjunto de Seleccionados:** Representa al conjunto de decisiones tomadas hasta este momento
* **Función Solución:** Determina si se ha alcanzado una solución (no necesariamente óptima)
* **Función de Factibilidad:** Determina si es posible completar el conjunto de candidatos seleccionados para alcanzar una solución al problema (no necesariamente óptima)
* **Función Selección:** Determina el candidato más prometedor del conjunto a seleccionar.
* **Función Objetivo:** Da el valor de la solución alcanzada

|  |
| --- |
| Vecino Más Cercano |

Tomamos como referencia una ciudad, y de manera recursiva con respecto a ella seleccionamos aquella que se encuentre más cerca hasta haber seleccionado todas las ciudades.

Para este caso, consideramos los siguientes elementos:

* **Función Selección:** 
  + Elegir del conjunto de candidatos *no seleccionados* la ciudad cuya distancia con respecto a la actual sea menor.
* **Conjunto de Candidatos:**
  + Todas las ciudades
* **Conjuntos de Seleccionados:**
  + Las ciudades que ya han sido elegidas
* **Función de Factibilidad:**
  + *Solo se puede seleccionar si no se ha usado anteriormente*
* **Función Solución:**
  + Cuando se hayan seleccionado todas las ciudades
* **Función Objetivo:**
  + Selección de ciudades ordenadas ascendentemente según su distancia con respecto a la actual

Así pues, si tuviéramos la siguiente situación:

2

**CIUDAD 4**

**CIUDAD 2**

10

11

8

5

10

**CIUDAD 6**

**CIUDAD 5**

**CIUDAD 3**

**CIUDAD 1**

8

Considerando que partimos de la ciudad uno, y que en el gráfico se representa la conectividad entre ciudades con su distancia asociada, visitaríamos inmediatamente después la 6, a continuación la 5, después la 3, seguida por la 4, y finalmente la 2 para regresar a la 1 y cerrar así el ciclo hamiltoniano.

|  |
| --- |
| Inserción más económica |

Aquí lo que se pretende es seleccionar aquellas ciudades que provoquen el menor incremento posible en la longitud total al finalizar el recorrido.

Para este caso, consideramos los siguientes elementos:

* **Función Selección:** 
  + Elegir aquella ciudad cuya selección junto a las siguientes ciudades, provoque un incremento menor en la longitud total al finalizar el recorrido
* **Conjunto de Candidatos:**
  + Todas las ciudades
* **Conjuntos de Seleccionados:**
  + Las ciudades que ya han sido elegidas
* **Función de Factibilidad:** 
  + *Solo se puede seleccionar si no se ha usado anteriormente*
* **Función Solución:**
  + Cuando se hayan seleccionado todas las ciudades
* **Función Objetivo:**
  + Selección de ciudades ordenadas ascendentemente en función del menor incremento que provocan en la longitud final del recorrido

Partimos de 3 ciudades que representan la que está más al norte, más al este y más al oeste. Con respecto cada una de ellas, considerando una ciudad inicial X0, desde la cual, teniendo en cuenta aquellas ciudades con las que tiene conexión X0, se comprobaría cual sería la siguiente que provocaría menor longitud total al finalizar el recorrido hasta pasar por todas las ciudades.

|  |
| --- |
| Heurística Propuesta: Mejor Distancia con Triangulación Matricial |

Tomando como referencia los valores de las distancias entre ciudades, se selecciona el menor de ellos y se almacenan las dos ciudades entre las que existe tal distancia. Este proceso se repite hasta haber seleccionado todas las ciudades.

Usando esto por sí mismo, puede dar lugar a la generación de ciclos entre ciudades, por lo que la selección de las mismas debe tener en cuenta esto también para generar la solución.

Usamos varias cosas:

-Un vector en donde cada posición representa a una ciudad, donde los valores en cada celda pueden variar entre 0 y 2, dependiendo de cuántas veces se haya seleccionado una distancia que involucre a esa ciudad.

-Una matriz binaria con dimensiones nxn, siendo n el nº de ciudades, donde cada fila representa a una ciudad, lo mismo para las columnas, y una celda representa con un 0 si no se ha elegido el camino entre esas dos ciudades, o con un 1 en caso contrario.

Con estas dos representaciones, para elegir una distancia hay que probar lo siguiente:

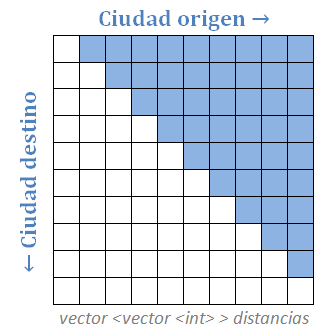
* La distancia a elegir debe ser la menor de las distancia
* Las ciudades que conecta esa distancia no puede haber sido elegida dos veces anteriormente
* No se ha elegido ya esa distancia anteriormente
* El coger esa distancia no crea un circuito antes de terminar el circuito con todas las ciudades

Para este caso, consideramos los siguientes elementos:

* **Función Selección:** 
  + Elegir el par de ciudades cuya distancia entre ellas, comparada con el resto de distancias de las demás ciudades, es menor.
* **Conjunto de Candidatos:**
  + Todas las ciudades
* **Conjuntos de Seleccionados:**
  + Las ciudades que han sido elegidas
* **Función de Factibilidad:** 
  + No se ha seleccionado la ciudad anteriormente más de dos veces, y no se generan ciclos si se inserta en al solución
* **Función Solución:**
  + Cuando se hayan seleccionado todas las ciudades
* **Función Objetivo:**
  + Selección de ciudades ordenadas de manera ascendente según el valor de *distancia*

Al pedirse proponer una heurística propia, se decide implementar una que actúe de la siguiente manera:

Considerando como punto de partida un vector bidimensional que almacena la distancia existente entre todas las ciudades, se concluye que, con el fin de facilitar las operaciones y reducir el tiempo y carga computacional, se tomarán los datos únicamente de la matriz triangular superior de dicho vector.



En ellos quedan reflejados las distancias entre una ciudad de origen y una de destino. De manera recursiva, se analizan todos ellos y se selecciona el menor, guardándose las dos ciudades implicadas, hasta haber realizado esta operación con todas las ciudades.

|  |
| --- |
| Estudio comparativo |

|  |  |  |  |
| --- | --- | --- | --- |
| Algoritmo | Archivo | Ciudades | D |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |