Problema del agente viajero

Nombre: Angel Azahel Ramirez Cabello Matricula: A01383328

Nombre: Luis Angel López Chávez Matricula: A01571000

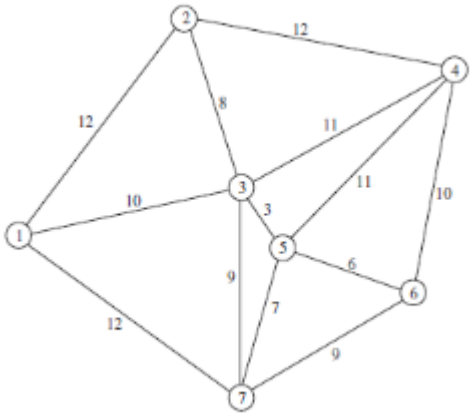
Nombre: Franco Mendoza Muraira Matricula: A01383399

Nombre: Alexis Daniel Leyva Yáñez Matrícula: A01770308

Instrucciones: Resuelve completa y correctamente cada uno de los siguientes puntos.

Se evalúa el procedimiento. Resultado sin procedimiento no tiene puntaje alguno.

En la figura se muestra un pequeño problema del agente viajero con siete ciudades. La ciudad 1 es el lugar de residencia del agente. Por lo tanto, si comienza desde su ciudad, el agente debe elegir una ruta para visitar cada una de las otras ciudades exactamente una vez antes de regresar a su punto de partida. Determina el orden ´optimo en que debe de visitar cada una de las ciudades.



- **Elabora un algoritmo heurístico para obtener una solución factible al problema. Presenta un pseudocódigo o diagrama de flujo.**

Partiendo del algoritmo de inserción más barata se genera la siguiente serie de pasos:

Inicialización usando k vértices haciendo un subtour que empiece y termine en el origen, en este caso se selecciona solamente el primer destino de la lista de siete ciudades para comenzar a generar soluciones:

ruta = [ciudades[0], ciudades[1],ciudades[0]]

Recorrer cada ciudad[i] restante de la lista en orden:

mejor_distancia = infinito

mejor_posicion = 0

Recorrer las posibles posiciones[j] para cada ciudad en la ruta

La posición que reduzca el costo de la ciudad en la ruta a comparación de la ciudad que estaba en esa posición es la que se escoge

d = distancia(ciudades[i], ruta[j]) + distancia(ciudades[i], ruta[j + 1]) - distancia(ruta[j], ruta[j + 1])

if d < mejor_distancia:

mejor_distancia = d

mejor_posicion = j + 1

Insertar la ciudad en la mejor posición encontrada

ruta.insert(mejor_posicion, ciudades[i])

return ruta

- ˆ **Usando el algoritmo propuesto, determine una solución del problema.**

Ruta óptima: [1, 3, 5, 7, 6, 4, 2, 1]

Costo: 63

Time of execution: 0.003991603851318359 seconds

- ˆ **Determine la cantidad de pasos que le toma a su algoritmo resolver el problema.**

Cómo debe escoger la mejor posición para cada ciudad dentro de la ruta, pero la ruta empieza siendo un simple subtour que ya tiene incluido al nodo 2 para la inicialización, entonces en total se tomaron 25 pasos.

- ˆ **Determine la complejidad de su algoritmo.**

Al tener que recorrer cada uno de los **n** nodos y también su posición posible de la ruta para cada uno de ellos, entonces la complejidad es de n^2

- ˆ **Resuelva el problema utilizando GAMS, y compare el resultado obtenido con el algoritmo propuesto**

Modelo matemático: TSP - Problema del viajante

Conjuntos

$icV : 1, 2, \ldots, n$

$i = j$

- n: número de nodos (lugares turísticos)

- $A(i, j)$ icV, jcV, arcos

Parámetros

- T_{ij} Costo en tiempo de ir de i a j (tiempo en minutos)

- r_i Tiempo de recorrido en el lugar i

Variables

- X_{ij} variable binaria para saber si se tomó la ruta de i a j

- X_{ij}, igual a 1 si se asigna $A(i, j)$, igual a 0 en otro caso

Restricciones

- R1:

$$\sum_{j}^{n} X_{ij} = 1, \forall i \neq jcV$$

- R2:

$$\sum_{i}^{n} X_{ij} = 1, \forall j \neq icV$$

- R3:

$$U_i - U_j + nX_{ij} \leq n - 1, \forall i, j, 2 \leq i \neq j \leq n$$

F.O.

$$min \sum_{i}^{n} \sum_{j}^{n} X_{ij} * (T_{ij} + r(i))$$

Ruta óptima: [1,3,5,7,6,4,2,1]
Costo: 63
Time of execution: 0.061 seconds

Se encontró que ambos algoritmos encuentran la solución óptima al problema, pero el heurístico lo logra con un tiempo significativamente más rápido, de igual manera esto puede que se logre porque se trata de solamente siete nodos, lo cual ayuda a que el heurístico sea capaz de resolver a optimalidad.
