

TRABAJO PRÁCTICO: HEURÍSTICAS Y  
COTAS INFERIORES PARA EL PROBLEMA  
DEL VIAJANTE DE COMERCIO (TSP)

LUCAS SALVATORE

1. Generación de instancias aleatorias y cálculo de valores óptimos, usando *Concorde*:

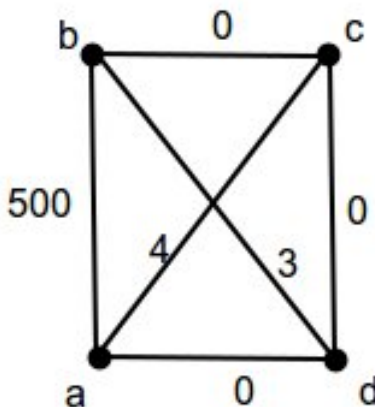
- Se generaron 10 instancias aleatorias con 100 nodos cada uno. Dichas instancias están en el archivo *instanciaN.tsp* donde  $N$  es el número de instancia.
- Los valores obtenidos por Concorde sobre las instancias aleatorias y las instancias del TSPLIB están en la tabla de la página  $x$ .

2. Comportamiento de la heurística *Vecino mas cercano*:

- Se anotó, por cada instancia el valor del tour dado por la heurística del Vecino Más Cercano, a través. Dichos valores están en la tabla de la página  $x$ .

3. Mejorando la heurística *Vecino mas cercano*:

- Se implementó la función `vecino_mas_cercano_mejorado.sci` que toma una instancia TSP y calcula la heurística del Vecino Más Cercano con  $v_1 = v$  para cada nodo  $v$  del grafo, y devuelve el tour de menor valor. La implementación se puede ver en el archivo antes mencionado o en la página  $x_2$ .
- La instancia ad-hoc con la que pruebo tanto la heurística del Vecino Más Cercano Mejorado como las heurísticas siguientes es el grafo  $K_4$  con los siguientes costos:



- Los valores obtenidos por la heurística sobre las instancias aleatorias y las instancias del TSPLIB junto con el comportamiento promedio están en la tabla de la página  $x$ .

4. Comportamiento de las heurísticas de *Insercion*

- Se implementó la función `insercion_mas_cercana.sci` que en base a la implementación de la función `insercion_mas_lejana.sci`, se modificó la iteración en donde se elegía el próximo vértice a insertar en el tour. En la Inserción más Lejana se seleccionaba los vértices con mayor costo. En la Inserción más Cercana se seleccionan los vértices de menor costo. Dicha

implementación se puede ver en el archivo `.sci` o en la página  $x_3$

- Los costos de los tours generados por las dos inserciones junto con el promedio, se puede ver en la página  $x$ .

5. Comportamiento de la cota dada por 1-árbol óptimo:

- La implementación está en el archivo `uno_arbol.sci` o se puede ver en la página  $x_4$ . La idea del algoritmo es encontrar dos aristas incidentes en el vértice inicial tal que la suma de sus costos sea mínima, digamos  $a$ . Luego buscamos un árbol recubridor mínimo del grafo menos el vértice inicial y el costo del árbol lo llamamos  $b$ . La cota inferior del valor óptimo del tour será  $a + b$ .
- Las cotas inferiores dadas por la heurísticas del 1-árbol junto con el comportamiento promedio están en la tabla de la página  $x$ .

6. Prueba de las heurísticas implementadas:

```
A =

      Inf      0.    500.      3.
      0.      Inf      4.      0.
    500.      4.     Inf      0.
      3.      0.      0.     Inf
-->[tour , valor] = vecino_mas_cercano(A,1)
valor =

      500.
tour =

      1.
      2.
      4.
      3.
-->[tour , valor] = vecino_mas_cercano_mejor(A)
valor =

      7.
tour =

      2.
      1.
      4.
      3.
-->[tour , valor] = insercion_mas_cercana(A)
valor =

      7.
tour =

      1.      4.      3.      2.
```

```

-->[tour,valor] = insercion_mas_lejana(A)
valor =

    7.
tour =

    1.    2.    3.    4.
-->uno_arbol(A,1)
ans =

    3.
-->uno_arbol(A,2)
ans =

    3.
-->uno_arbol(A,3)
ans =

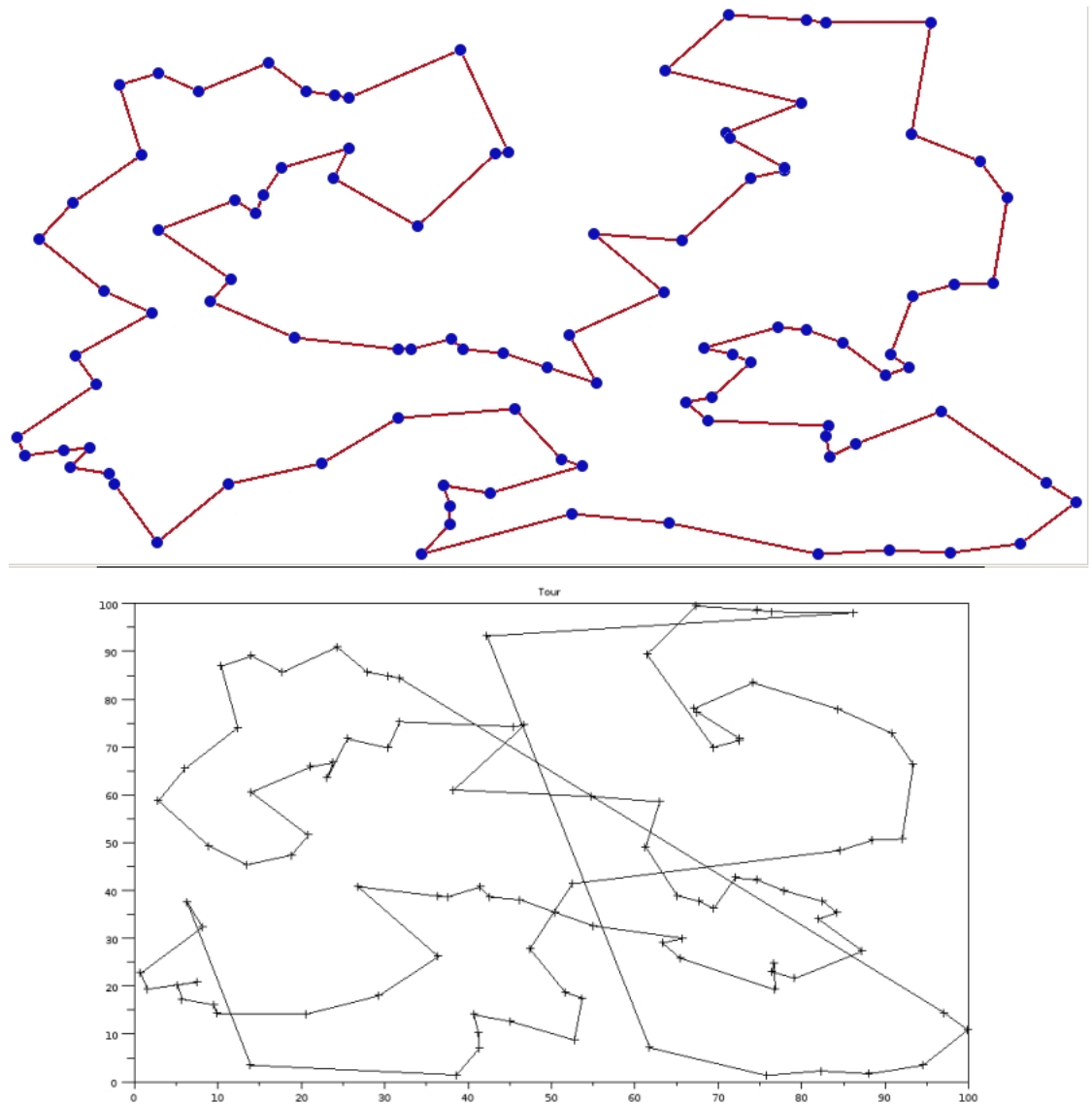
    4.
-->uno_arbol(A,4)
ans =

    4.

```

### Observaciones:

- Se pudo ver que el máximo cociente entre el valor de la heurística del vecino más cercano y el óptimo se da en la instancia 2. El gráfico siguiente muestra primero el tour óptimo calculado por Concorde y luego se muestra el tour calculado por la heurística.



- De acuerdo a los valores promedios provistos por la tabla y los valores promedios brindados por el libro de Cook, se puede ver que hay una escasa diferencia entre ellos:

Heuristica	Tabla	Libro Cook
Vecino Más Cercano	$1.24 \times OPT$	$1.26 \times OPT$
Insercion Más Lejana	$1.09 \times OPT$	$1.16 \times OPT$
Cota 1-árbol	$0.88 \times OPT$	$0.90 \times OPT$

- d