Pontificia Universidad Católica Madre y Maestra
Facultad de Ciencias de la Ingeniería

Tarea #2:

Problema del Viajante

Presentado Por:

Angel González (2013-0157)

Materia:

Diseno y Analisis de Algoritmos

Profesor:

Juan Núñez.

Marco Teorico

El Problema del Agente Viajero (TSP por sus siglas en inglés) o problema del viajante, responde a la siguiente pregunta: Dada una lista de ciudades y las distancias entre cada par de ellas, ¿cuál es la ruta más corta posible que visita cada ciudad exactamente una vez y regresa a la ciudad origen? Este es un problema NP-duro dentro en la optimización combinatoria, muy importante en la investigación de operaciones y en la ciencia de la computación.

El problema fue formulado por primera vez en 1930 y es uno de los problemas de optimización más estudiados. Es usado como prueba para muchos métodos de optimización. Aunque el problema es computacionalmente complejo, una gran cantidad de heurísticas y métodos exactos son conocidos, de manera que, algunas instancias desde cien hasta miles de ciudades pueden ser resueltas.

El TSP tiene diversas aplicaciones aún en su formulación más simple, tales como: la planificación, la logística y en la fabricación de microchips. Un poco modificado, aparece como: un sub-problema en muchas áreas, como en la secuencia de ADN. En esta aplicación, el concepto de "ciudad" representa, por ejemplo: clientes, puntos de soldadura o fragmentos de ADN y el concepto de "distancia" representa el tiempo de viaje o costo, o una medida de similitud entre los fragmentos de ADN. En muchas aplicaciones, restricciones adicionales como el límite de recurso o las ventanas de tiempo hacen el problema considerablemente difícil. El TSP es un caso especial de los Problemas del Comprador Viajante (travelling purchaser problem).

En la teoría de la complejidad computacional, la versión de decisión del TSP (donde, dado un largo "L", la tarea es decidir cuál grafo tiene un camino menor que L) pertenece a la clase de los problemas NP-completos. Por tanto, es probable que en el caso peor el tiempo de ejecución para cualquier algoritmo que resuelva el TSP aumente de forma exponencial con respecto al número de ciudades.

Codigo

```
import time

def distance(p1, p2):
    return ((p1[0] - p2[0])**2 + (p1[1] - p2[1])**2) ** 0.5

def total_distance(points):

    return sum([distance(point, points[index + 1]) for index, point in
enumerate(points[:-1])])

def optimized_travelling_salesman(points, start=None):

    if start is None:
        start = points[0]
    to_visit = points
    path = [start]
    to_visit.remove(start)
    while to_visit:
        nearest = min(to_visit, key=lambda x: distance(path[-1], x))
        path.append(nearest)
        to_visit.remove(nearest)
    return path

def main():
    f =  open("data.txt", 'r')
    lista =[[float(line.split()[0]), float(line.split()[1])] for line in f]


    answer = total_distance(optimized_travelling_salesman(lista))
    print("The solution is " +  str(int(answer)))

if __name__ == "__main__":
    start_time = time.time()
    main()
    print("--- %s seconds ---" % (time.time() - start_time))
```

Corrida

```
Angels-Mac-mini:t2 angelgonzalez$ python VendedorAmbulante.py
The solution is 7890
--- 0.0131568908691 seconds ---
```