**Problema de vendedor ambulante**

1. Introducción

El problema del vendedor viajero o también conocido como TSP (por siglas en ingles). Consiste en encontrar, en un grupo finito de ciudades y sus respectivos costos de viajes entre ciudad y ciudad, el camino más económico de recorrer todas las ciudades. Dicho recorrido tiene ciertas condiciones, se debe iniciar el recorrido en una ciudad, pasar por todas las ciudades, solamente una vez, y terminar en la ciudad en la que se comenzó. Esto debe ser logrado teniendo en consideración que el resultado debe ser el recorrido que tenga el menor costo.

A lo largo de la historia, se han diversas soluciones a este problema. Por ejemplo:

Karl Menger (*Shortest Hamiltonian Path*, 1930), J.B. Robinson (*Hamiltonian game*, 1949), G. Dantzig, R. Fulkerson, y S. Johnson (*Solution of a large-scale traveling-salesman problem*, 1954), M. Held and R.M. Karp (Introducción de heurísticas basadas en programación dinámica, 1962), entre otros. (Espinoza, 2006)

La principal motivación para el desarrollo del proyecto, consiste en las diferentes aplicaciones del TSP. Algunas de ellas son estudio en la secuencia de genes, diseño de chips, enrutamiento de vehículos, entre otras.

1. Objetivo

El presente trabajo tiene como objetivo principal encontrar una solución parcial al problema de TSP usando una *dataset* del Ministerio de Educación del Perú. La cual contiene la posición geográfica (latitud y longitud) de cada centro poblado del Perú.

1. Marco Teórico

El algoritmo de búsqueda por fuerza bruta, conocido también como búsqueda exhaustiva. Es una técnica de programación, la cual consiste en encontrar todos los candidatos a solución y verificar, uno por uno, si satisfacen al problema.

El *backtracking* es una técnica de programación para hacer una búsqueda a través de todas las alternativas en un espacio de búsqueda. Para esto, los algoritmos que emplean *backtracking* construyen posibles soluciones candidatas. Por ejemplo, dada una solución candidata “s” se verifica si es solución, si no lo es se construyen todas las posibles extensiones de “s” y se invoca recursivamente al algoritmo con todas ellas.

1. Soluciones

Fuerza Bruta(Solucion Ingenua)

1) Considere la ciudad 1 como el punto inicial y final.   
2) Generar todos (n-1)! [Permutaciones](https://www.geeksforgeeks.org/write-a-c-program-to-print-all-permutations-of-a-given-string/) de las ciudades.   
3) Calcule el costo de cada permutación y realice un seguimiento de la permutación del costo mínimo.   
4) Devolver la permutación con costo mínimo.

BackTracking

En este caso, primero creamos todos los ciclos hamiltonianos utilizando conceptos de retroceso profundo. Entonces entre

En estos ciclos hamiltonianos, el problema del vendedor ambulante corresponde a los ciclos de costo mínimo.

Algoritmo:

* Primero, hemos creado la estructura de datos del gráfico utilizando la matriz de adyacencia para almacenar cada vértices.
* Generado el peso aleatorio o la matriz de conexiones.
* Luego extraemos todos los ciclos hamiltonianos usando la función tsp () en nuestroprograma. Esta función funciona como:
* Primero, enviamos el vértice inicial como argumento en función.
* Ahora, el siguiente fragmento de código se procesa solo en vértices no visitados. También tenemos Estructura de datos de ruta mantenida para almacenar la ruta hamiltoniana. Visitas de ruta cada vértice exactamente una vez.
* Compruebe si agregar vértice a la ruta lleva a la solución o no es
* Contribuyendo entonces considérelo en el camino hamiltoniano.
* De lo contrario Retrocede hasta que te recuperes.

4. También hemos implementado el código para obtener una ruta óptima dentro de la propia función.

Verificamos la ruta, si es la ruta, calculamos el costo de los ciclos y

Compáralo con el ciclo global hasta ahora. Y si queda satisfecho entonces almacenamos el camino óptimo en la estructura de datos para imprimirlo después del final del programa.

1. Complejidad

La complejidad del algoritmo de la búsqueda por la fuerza bruta en el tiempo es O (N!). Por lo que nunca debes usar en listas largas.

En el caso del algoritmo heurístico la complejidad es de tiempo O (N ^ 2)

1. Conclusiones

Luego de hacer las pruebas debidas, hemos comprobado que se ha podido implementar una solución para TSP en los nodos de tamaño corto, debido a que en el peor de los casos es posible factorial de n

Referencias

Espinoza, D. (2006). El Problema del Vendedor Viajero (TSP) y Programación Entera (IP) [diapositivas de PowerPoint]. Recuperado de: <http://www.dii.uchile.cl/~daespino/PApers/TSP_and_IP_chile_050820.pdf>

Baier, J. (s.f.). Backtracking [diapositivas de PowerPoint]. Recuperado de: <http://jabaier.sitios.ing.uc.cl/iic2552/backtracking.pdf>