The Traveling Salesman Problem

Análisis del Algoritmo

Universidad de Sonora

Ing. en Sistemas de Información

Estructura de Datos

Contreras Hernández Carlos

Lunes, 13 de mayo de 2013

El problema del viajante o *the traveling salesman problem* (TSP), consiste en encontrar la ruta que lleva a cabo un vendedor, que comenzando por un origen, visita un determinado y preestablecido conjunto de ciudades y vuelve a ese origen, de manera que la distancia recorrida total es mínima y cada ciudad sólo se visita una vez.

El TSP es uno de los más típicos y mejor estudiados problemas de la optimización combinatoria, pero a pesar de ello aún no se conocen algoritmos que lo resuelvan de forma exacta. Los algoritmos utilizados para su resolución sólo ofrecen aproximaciones, y la solución óptima sólo es alcanzable para casos concretos del problema.

Dentro de esta optimización combinatoria de la cual el TSP es objeto de estudio, se le cataloga como un problema NP-completo, lo que significa que el esfuerzo computacional que se debe llevar a cabo para encontrar una solución óptima crece de forma exponencial con la entrada del problema, que en el caso concreto de TSP sería el número de nodos o vértices de la red.

Algunos métodos para solucionar este problema incluye:

* Soluciones exactas
  + Método por fuerza bruta

Cuando uno piensa en resolver el TSP, el primer método que se viene al cabeza es el método por fuerza bruta. El método por fuerza bruta es simplemente generar todos los recorridos posibles y calcular las distancias. El recorrido con menor distancia será por lo tanto el camino más corto.

El número de nodos de la red va a ser fundamental en determinar la complejidad del problema. Cuanto mayor sea el número de nodos, mayor va a ser el número de rutas posibles, y por lo tanto mayor será el esfuerzo requerido para calcular todas ellas. Así, el número de rutas posibles entre N nodos va a ser igual a N!, lo que hace que la resolución del TSP mediante la obtención de todas las rutas posibles y comparación entre ellas sea poco factible incluso para un número de nodos no elevado. Sólo con tener una red simple de 7 nodos, ya sería necesario calcular más de 5000 combinaciones, (7!=5040), y elevando simplemente hasta 10 el número de vértices de la red, las posibles rutas se disparan hasta más de tres millones (10!=3.628.800).

* + Método Branch and Bound

La estrategia Branch and Bound divide el problema para ser resuelto en un número determinado de sub-problemas. El método Branch and Bound es un sistema para resolver una secuencia de sub problemas en donde cada uno de estos puede tener múltiple soluciones y donde la solución escogida por un sub problema puede afectar la posible solución

La resolución del TSP en redes relativamente simples mediante el método de obtención y comparación de todas las rutas es absolutamente inabordable mediante los medios computacionales disponibles actualmente, lo que hace que sea necesario utilizar otras vías que aunque no obtengan la solución óptima, ofrecen una respuesta aproximada lo suficientemente optimizada. Mediante estas aproximaciones se han conseguido resolver TSP con varios miles de nodos, e incluso más, y estas soluciones no son ni un 1% peores que la óptima.

* Soluciones por aproximación
  + Vecino más cercano

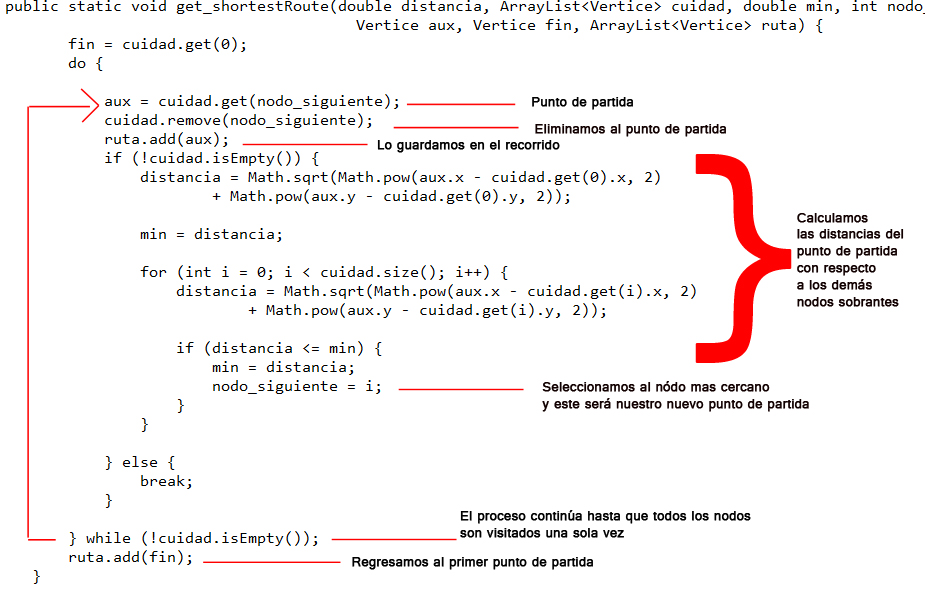
Este es probablemente la manera más simple y directa para resolver el TSP, la clave de este algoritmo es siempre visitar a la cuidad más cercana y regresar al punto de partida una vez que ya se hayan recorrido todas las ciudades.

* + Acercamiento voraz

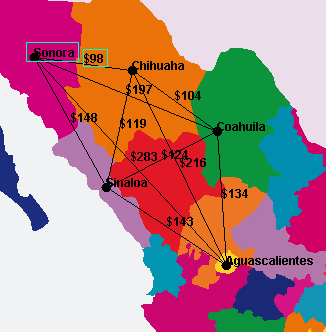
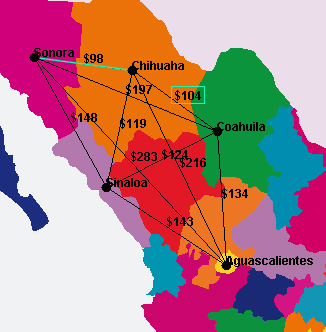
El algoritmo voraz más intuitivo para resolver TSP está basado en el algoritmo del vecino más cercano.

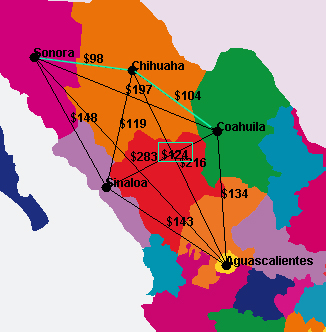
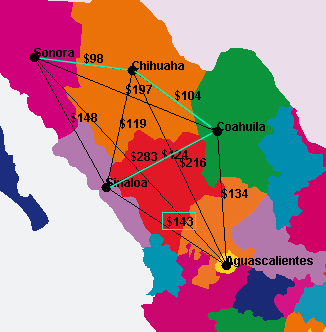
El acercamiento voraz es el algoritmo de mejora más simple. Comienza con el nodo 1 como punto de partida. El algoritmo calcula todas las distancias a los otros nodos n-1. Se va al nodo más cercano. Toma a este nodo como nuevo el punto de partida y selecciona al nodo más cercano de los otros nodos n-2 sobrantes. El proceso continúa hasta que todos los nodos son visitados una sola vez y se regresa al nodo 1. Cuando el algoritmo es terminado la secuencia es devuelta como el mejor recorrido.

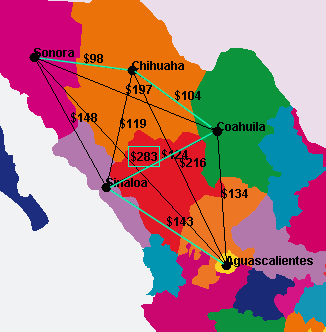
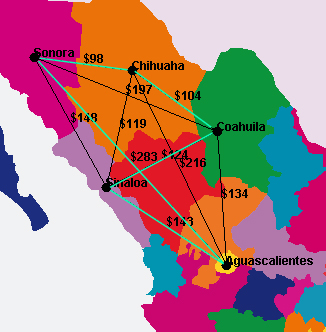
Implementación de este algoritmo en Java:



Implementación gráfica de este algoritmo:

Los algoritmos voraces buscan soluciones simples, fáciles de implementar a problemas complejos, decidiendo cuál es paso siguiente qué proporcionará el beneficio más obvio. Se les llama algoritmos voraces porque mientras la solución óptima para cada instancia del problema más pequeño proporciona una salida inmediata, el algoritmo no considera a todo problema en su totalidad. Una vez que se toma una decisión, nunca es reconsiderada.

La ventaja de usar de un algoritmo voraz consiste en que las soluciones a las instancias más pequeñas del problema pueden ser francas y fáciles para entender. La desventaja es que es completamente posible que las soluciones óptimas a corto plazo pueden conducir al peor resultado a largo plazo.

Como se mencionó anteriormente no existen métodos que resuelvan de manera exacta cualquier TSP, pero sí que existen soluciones exactas para algunos casos concretos. A lo largo de los últimos cincuenta años se han producido continuos avances en la resolución óptima del TSP, lo que ha permitido obtener soluciones óptimas para TSP con cada vez mayor número de nodos, hasta alcanzar una cifra de casi 86.000.

**Referencias**

http://whatis.techtarget.com/definition/greedy-algorithm

http://es.wikipedia.org/wiki/Problema\_del\_viajante

http://cs.indstate.edu/~zeeshan/aman.pdf

http://neofosis.com/2011/08/el-problema-del-viajante-que-es-soluciones-y-aplicaciones-en-nuestra-vida-cotidiana/