

Proyecto del curso Datos y Algoritmos 2

Semestre 2017-2

Rutas ideales para entregas a domicilio en la ciudad de Medellín

Mateo Marulanda Cifuentes
Universidad de Eafit
Departamento de Ingeniería
Correo: mmaruland@eafit.edu.co

Yashúa Alexander Narvaéz Pulgarín
Universidad de Eafit
Departamento de Ingeniería
Correo: ynarvaezp@eafit.edu.co

Mauricio Toro Bermúdez
Universidad de Eafit
Departamento de Ingeniería
Correo: mtorobe@eafit.edu.co

Palabras Clave

Algoritmo de búsqueda, Teoría de Grafos, Ciencias de la computación, Algoritmos. Shortest path.

ACM palabras clave

Theory of computation, Graph algorithms analysis, Graph algorithms analysis, Approximation algorithms analysis, Shortest paths, Graph drawings

Introducción

Desde la creación de la teoría de grafos por Leonhard Euler en el siglo XVIII, se les han dado diferentes aplicaciones a la lo largo de la historia, entre ellas la búsqueda de rutas más cortas entre diferentes nodos conexos. A continuación se presentará una solución para enrutamiento óptimo dadas coordenadas geográficas, con el fin de solucionar la necesidad de varios restaurantes que prestan servicio de entrega a domicilio.

Problema

En la ciudad de Medellín existen gran cantidad de restaurantes que realizan el servicio de entregas a domicilio. Muchos de ellos manejan la política de: si la comida no llega a su hogar en un tiempo prudente, el servicio prestado será totalmente gratis. Por eso se ven obligados a realizar rutas más óptimas, dados las zonas de la ciudad. El siguiente generador de rutas planea resolver este problema dando las rutas más cortas, dados los diferentes puntos en la ciudad.

Problemas Similares

[1]-Problema de los puentes de Königsberg
El problema original, y por el cual se creó la teoría de grafos. Dado el mapa de Königsberg con el río Pregel dividiendo el plano en cuatro regiones distintas, que están unidas a través de los siete puentes, ¿es posible dar un paseo comenzando desde cualquiera de estas regiones, pasando por todos los puentes, recorriendo

sólo una vez cada uno, y regresando al mismo punto de partida? La respuesta es negativa, es decir, no existe una ruta con estas características. El problema puede resolverse aplicando un método de fuerza bruta, lo que implica probar todos los posibles recorridos existentes. Sin embargo, Euler en 1736 en su publicación «Solutio problematis ad geometriam situs pertinentis» demuestra una solución generalizada del problema, que puede aplicarse a cualquier territorio en que ciertos accesos estén restringidos a ciertas conexiones, tales como los puentes de Königsberg.

[2]-Solución del ruteo de vehículos dependientes del tiempo utilizando un algoritmo genético modificado

El ruteo de vehículos corresponde a un tipo de problemas combinatoriales que en la medida en que se pretenda obtener la o las rutas óptimas que permitan atender completamente un conjunto de clientes determinado, el problema se vuelve muy complicado computacionalmente, de forma que para un número de clientes mayor a 20 con una sola ruta se tiene $20!/2$ combinaciones posibles como se indica en [1], es decir 12×10^{17} combinaciones, siendo esto casi imposible de resolver computacionalmente con los métodos exactos existentes, los cuales resuelven el problema por medio del modelado matemático, es decir buscando cada una de las combinaciones posibles y teniendo en cuenta cada una de las restricciones del problema. El algoritmo se prueba con diferentes instancias del problema de CVRP conocidas como resultados de la ejecución del algoritmo genético para instancias publicadas en la literatura especializada, para luego implementar dicho algoritmo en el caso particular de la distribución de productos lácteos de la empresa COLÁCTEOS en la ciudad de San Juan de Pasto, Nariño. Para lo cual se realiza una georeferenciación de los clientes de la ciudad a los que distribuye la empresa y se obtiene la matriz de costos dada por las distancias entre puntos o clientes, adicionalmente se considera el horario en que se realiza el recorrido de los vehículos y la capacidad máxima que estos tienen junto con la demanda de las tiendas en la ciudad. Con base en la matriz de costos se genera el mejoramiento

de las rutas haciendo uso del algoritmo genético modificado.

[3]- **Modelos y algoritmos solución para un problema de control óptimo de semáforos. aplicación a cruces con 2, 3, 4 o 6 fases en la ciudad de a coruña**

Problema de control óptimo que consiste en asignar tiempos a luces de semáforos que regulan cruces de dos calles con los dos sentidos de circulación, y aplicar los resultados a cruces (con congestiones frecuentes) de la ciudad de A Coruña. El criterio de comparación se expresa en una función objetivo a minimizar que puede ser: longitud media de la cola en el carril con mayores colas, longitud de la cola más larga, tiempo medio de espera en el carril con mayor tiempo de espera, una combinación de las anteriores, entre otras posibilidades. Para su solución, se propone un método híbrido que combina una heurística, tipo recocido simulado, para calcular una buena aproximación a la solución, que luego será mejorada con un algoritmo quasi-Newton para problemas de optimización no suaves. Expondremos los resultados numéricos obtenidos a partir de la programación de los algoritmos anteriores en MATLAB

[4] **Implementación de un algoritmo metaheurístico para la solución de un problema de programación de transporte terrestre internacional**

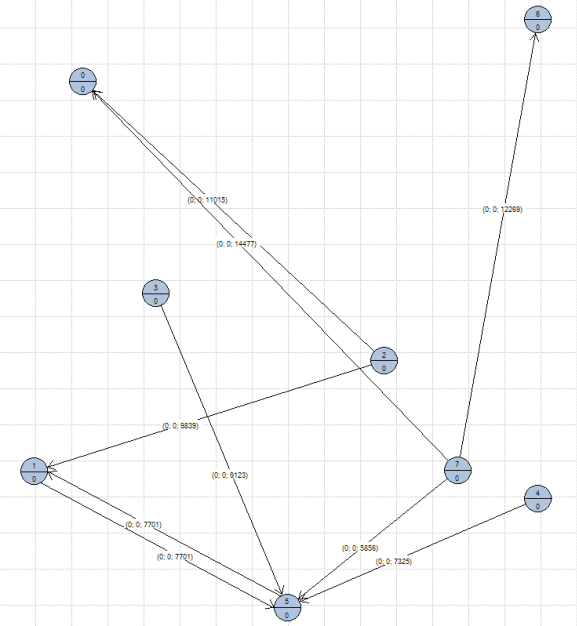
El problema consiste en realizar la programación de operaciones de transporte internacional entre dos países, Colombia y Venezuela, para un horizonte de tiempo con múltiples variantes; entre ellas podemos resaltar: una flota heterogénea de vehículos y tráileres, múltiples depots (clientes), restricciones de ventanas de tiempo en los depots, diversas modalidades de servicio, entre otras. La solución se realizó a través de la implementación de la metaheurística Recocido Simulado (SA, Simulated Annealing), la cual se probó iniciando con soluciones factibles generadas por dos algoritmos heurísticos; el primero se basa en la heurística clásica conocida como el vecino más cercano y el segundo genera soluciones de forma aleatoria. Los resultados del Recocido Simulado usando las soluciones generadas con el primer algoritmo, no mostraron una mejora frente a dicha solución inicial, contrario a lo hallado con el uso de las soluciones del segundo método, donde se alcanzó hasta cerca de un 50% de mejor.

tanto a izquierda como a derecha. Los párrafos deberán ser escritos a simple espacio.

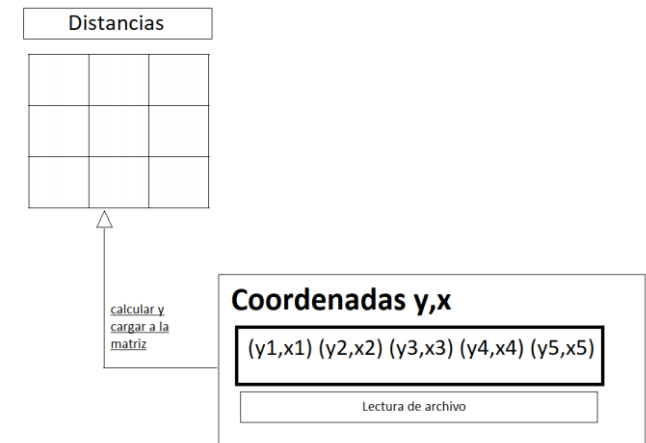
Las diferentes secciones estarán numeradas con números arábigos. En caso que deba separar una sección en subsecciones, adopte el estilo utilizado en estas instrucciones.

Figuras y Tablas

La estructura que se usa en la solución son los grafos



Operaciones de la estructura



Complejidades de los Métodos

Método	Complejidad
Read	$O(n)$
Read Link	$O(n)$
Rebuild	$O(n^2)$
Distance	$O(1)$
minCost	$O(2^n * n^2)$
generarLink	$O(n)$
cordenatesTour	$O(n)$

Toma de tiempos

Mejor Ms	Peor Ms	Promedio	Nodos	Memoria mejor	Memoria peor	Memoria promedio
59	71	65	9	109341	109341	109341
5523	5850	5686	15	81426	86191	83808
25984	26146	26065	16	64973	66050	65511

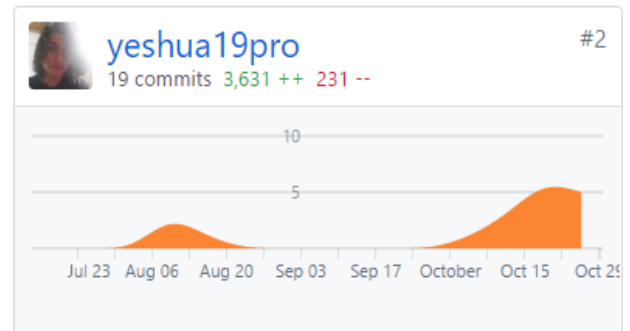
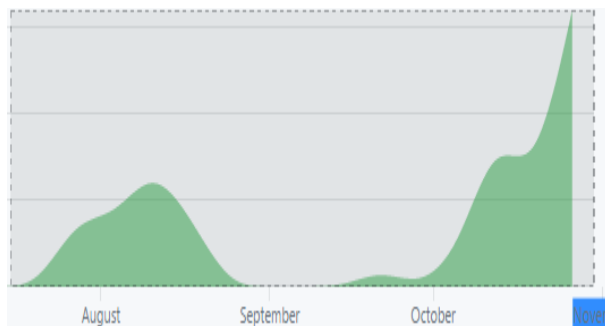
Criterios de la estructura de datos

A la hora de decidir la estructura de datos en la búsqueda del camino más corto entre nodos, decidimos utilizar un Hashmap donde se guardará como clave el identificador del nodo y como valor las coordenadas del nodo. El identificador del nodo se guardará en un ArrayList para luego poder consultar las coordenadas y calcular las distancias cuando sean cargadas a una matriz. Utilizamos 2 matrices ya que las coordenadas de longitud y latitud se manejan con punto decimal y debido a ello tenemos una matriz que almacena valores en punto flotantes y otra matriz en enteros para redondear el valor a metros y poder enviarla como parámetro al método que calcula las distancias entre puntos.

Criterios del algoritmo

Calculamos la distancia entre todos los puntos haciendo uso de una matriz de distancias y luego implementamos el algoritmo de Held Karp para calcular la ruta más corta y así poder tener el orden de los nodos a visitar, como último casteamos la cadena accediendo en el orden específico de los nodos recuperando sus valores utilizando los identificadores guardados en el ArrayList.

Trabajo de equipo en Git



Conclusiones

El algoritmo entre más cantidad de nodos el tiempo que demora en calcular la ruta más óptima crece exponencialmente, con una cantidad de nodos inferior a once (11) el algoritmo es muy eficiente, pero con una mayor carga se relentiza considerablemente.

La primera solución fue la solución definitiva por lo cual los datos que se obtienen son definitivos para el problema planteado.

Para una futura continuación de este proyecto se podría trabajar con rutas marítimas para los barcos o rutas aéreas generando los menores costos. También

Se podrían crear programas de turismo para crear rutas de paso por la ciudad.

Agradecimientos

Damos gracias al docente de la universidad de Eafit, Mauricio Todo Bermúdez del curso de Estructuras de Datos y algoritmos 2 que nos ha acompañado durante un corto año, y nos ha brindado los conocimientos necesarios para realizar varios proyectos en nuestro pregrado.

Referencias

- [1] Problema de los puentes de Königsberg, tomado de: https://es.wikipedia.org/wiki/Problema_de_los_puentes_de_K%C3%B6nigsberg. 1/11/2017
- [2] Fredy Alexander Guasmayan Guasmayan,, Solucion Del Problema De Ruteo De Vehiculos Dependientes Del Tiempo, utilizando un algoritmo genetico modificado. recuperado el 1/11/2107 de <http://repositorio.utp.edu.co/dspace/bitstream/handle/11059/4562/5196G917.pdf;sequence=1>
- [3] Carmen Socorro Lema Fernández. Modelos Y Algoritmos Solución Para Un Problema De Control Óptimo De Semáforos. Aplicación A Cruces Con 2, 3, 4 O 6 Fases En La Ciudad De A Coruña. Recuperado el 1/11/2017 de http://ruc.udc.es/dspace/bitstream/handle/2183/10090/LemaFernandez_CarmenSocorro_TD_2012.pdf?sequence=5