

Algoritmo para disminuir el tráfico urbano asignando rutas de vehículos compartido a los empleados de una empresa

Juan Camilo Guerrero Alarcón
Universidad Eafit
Colombia
jcguerrera@eafit.edu.co

Juan José Escudero Valencia
Universidad Eafit
Colombia
jjescuderv@eafit.edu.co

Santiago Pulgarín Vásquez
Universidad Eafit
Colombia
spulgarinv@eafit.edu.co

Mauricio Toro Bermúdez
Universidad Eafit
Colombia
mtorobe@eafit.edu.co

RESUMEN

El problema se basa en la disminución de uso de vehículos particulares en la ciudad de Medellín. Actualmente la ciudad está enfrentando un gran problema ambiental en cuanto a la calidad del aire, y las entidades de movilidad han implementado técnicas como el pico y placa ambiental, la cual contribuye con disminuir la emisión de gases, pero no da una solución permanente. Para encontrar posibles soluciones, analizamos problemas que son parecidos al nuestro en varios aspectos, teniendo como interés la búsqueda de caminos más cortos pasando por varios vértices con un destino común.

Palabras clave

Diseño - Algoritmo – Camino más Corto – Optimización

Palabras clave de la clasificación de la ACM
Theory of computation → Design and analysis of algorithms
→ Graph algorithms analysis → Shortest paths

1. INTRODUCCIÓN

El aumento de tráfico en la ciudad de Medellín se ha convertido en un problema serio que deriva en más de estos, uno de ellos es la polución del aire, mayor tiempo en las transiciones diarias, se reduce la productividad, etc. Una de las posibles soluciones es que los dueños de los vehículos compartan con las demás personas dichos vehículos, con esto no se tendrá vehículo por persona, esto claro se puede aplicar siempre y cuando las personas se dirijan a una misma institución o sitio de empleo, en estos casos sería muy eficiente, reduciendo también el tiempo de espera en un parqueadero.

2. PROBLEMA

El problema radica en la disminución de tráfico en la ciudad, una de las estrategias es que cada uno de los dueños de vehículos de determinada empresa recoja a otras personas que trabajan en esta misma, el conductor puede llevar un máximo de 4 pasajeros y todos se dirigen a al mismo lugar de trabajo. El problema radica en que el tiempo en el que se demora recogiendo a los demás ocupantes no puede aumentar más de lo debido.

3. TRABAJOS RELACIONADOS

3.1 El problema del viajero

Los orígenes del problema del vendedor ambulante no están claros. El problema del vendedor ambulante fue formulado matemáticamente en la década de 1800 por el matemático irlandés WR Hamilton y por el matemático británico Thomas Kirkman. El juego Icosian de Hamilton era un rompecabezas recreativo basado en encontrar un ciclo hamiltoniano. El problema del vendedor viajero (TSP) hace la siguiente pregunta: "Dada una lista de ciudades y las distancias entre cada par de ciudades, ¿cuál es la ruta más corta posible que visita cada ciudad y regresa a la ciudad de origen?". El algoritmo define que el uso de fuerza bruta, aunque esto no denota un rendimiento óptimo, este problema actualmente es de los más estudiados en cuestión de optimización.

3.2 El problema del camino más corto sobre la red del Metrobús y del metro de la Ciudad de México

Los orígenes del problema se dan cuando en la fase de diseño para construir la red de Metrobús y metro en ciudad de México, surge la necesidad de encontrar el camino más corto entre la estación "Cuatro Caminos" y "Tláhuac", sobre la red de metro y Metrobús. Y en donde el diseñador se pregunta ¿Cuál es la ruta de metro y Metrobús más corta posible de recorrer el grafo y volver al punto de origen? Todo esto relacionado con la necesidad de optimizar el tiempo en este tipo de problemas.

3.3 El problema de rutas de vehículos (mensajería)

Este es uno de los problemas más estudiados en investigación operativa. En donde se plantea la necesidad de diseñar el conjunto óptimo de rutas para una flota de vehículos que deben servir a un determinado número de clientes. En donde aparece una empresa repartidora de mensajería que desea encontrar la ruta optima de reparto o recogida desde uno o varios centros de acopio a un número determinado de ciudades o clientes de tal manera que el coste total originado de ese reparto o recogida sea mínimo.

3.4 The School Bus Routing Problem

Muchas escuelas tienen contratos con empresas privadas de transporte para tener cierto número de buses a disposición de sus estudiantes. Estos contratos se basan en costos por número de buses y la distancia que estos recorran. El problema que enfrentan las escuelas es encontrar rutas para

sus buses de tal manera que se utilicen el menor número de estos, se cubran todas las estaciones sin recorrer mucha distancia y que los recorridos no duren más de 45 minutos.

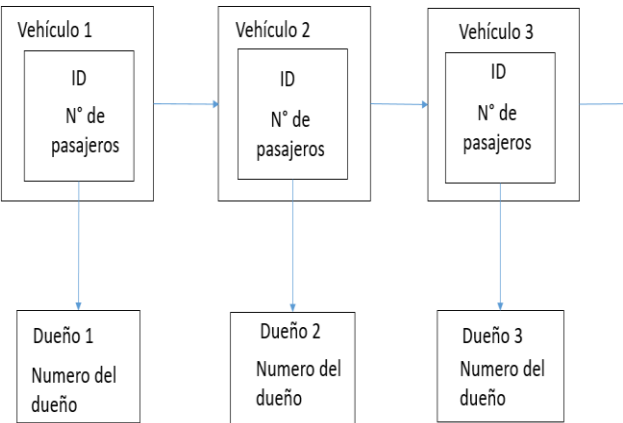
Solución: Primero generamos un grafo que contenga la cantidad mínima de estaciones de buses posible, tal como se ve en la figura 1. Para realizar este análisis, buscamos las estaciones que conecten con la mayor cantidad de casas de los estudiantes, con la restricción de que ningún estudiante se quede por fuera del análisis, y que, de ser necesario, se agregue otra estación para un estudiante.

Luego, se buscan las estaciones que conecten con todos los estudiantes de un subconjunto, y si esa estación no existe, entonces se empieza una búsqueda del camino más corto entre ellas, para luego añadirlas a una ruta final que no exceda más de 45 minutos de recorrido. Creando entonces varias paradas y varias rutas para poder recoger a todos los estudiantes.

4. Doble Lista Enlazada con Permutaciones

A continuación, explicamos la estructura de datos y el algoritmo.

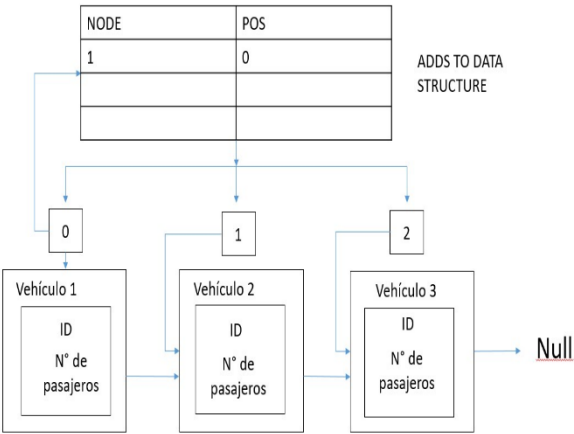
4.1 Estructura de datos



Gráfica 1: LinkedList doble con permutaciones, en las cuales incluyes cada lista un cierto número de pasajeros

4.2 Operaciones de la estructura de datos

Adición de datos a LinkedList.



Gráfica 2: Ilustración de la adición en la estructura de datos escogida previamente

4.3 Criterios de diseño de la estructura de datos

La selección de dicha estructura de datos fue implementada ya que esta nos permitía dar una solución previa a la definitiva, ya que con dichas listas podíamos simular tanto los conjuntos como los subconjuntos para realizar las correspondientes permutaciones y así empezar a comparar todos los casos, eso además de que no proporciona un gasto inmenso o considerable de memoria, aunque cabe aclarar que puede tener cierto margen de error.

4.4 Análisis de Complejidad

Método	Complejidad
leerArchivo()	$O(n^2)$
asignarVehiculos()	$O(n)$
guardarArchivo()	$O(n)$

Tabla 1: Tabla para reportar la complejidad

4.5 Algoritmo

Diseñen el algoritmo para resolver el problema y gráfiquenlo. No usen gráficas extraídas de internet

Gráfica 3: Paso a paso cómo se ensamblan fragmentos de ADN utilizando los grafos de Bruijn.

4.6 Cálculo de la complejidad del algoritmo

Sub problema	Complejidad
--------------	-------------

Leer el archivo y guardar los datos en la estructura de datos	$O(n^2)$
Asignar los vehículos compartidos	$O(n)$
Guarda el archivo en un archivo .txt	$O(n)$
Complejidad Total	$O(n^2)$

Tabla 2: complejidad de cada uno de los sub problemas que componen el algoritmo. Sea n el número de dueños de vehículos y la empresa.

Criterios de diseño del algoritmo

El algoritmo se realizó de esta manera ya que nos permite asemejar el problema como un conjunto de datos con el cual podemos realizar las permutaciones para así comparar con los demás vecinos y cada subconjunto de dichos datos, también porque la estructura de los datos no presenta un mayor inconveniente para su implementación y al momento de realizar las restricciones correspondientes se disminuye el margen de error, ya que podemos agrupar estos datos y asignar a el conjunto del que hablamos anteriormente. Otro factor importante fue el uso de memoria, ya que su ejecución no requería cantidades considerables.

4.8 Tiempos de Ejecución

	<i>Conjunto de Datos 1 (205, 1.1)</i>	<i>Conjunto de Datos 2 (205, 1.2)</i>	<i>Conjunto de Datos 3 (205, 1.3)</i>
<i>Mejor caso</i>	0 ms	0 ms	0 ms
<i>Caso promedio</i>	0 ms	4 ms	5 ms
<i>Peor caso</i>	1 ms	16 ms	16 ms

Tabla 3: Tiempos de ejecución del algoritmo con diferentes conjuntos de datos

4.9 Memoria

<i>Conjunto de Datos 1 (205, 1.1)</i>	<i>Conjunto de Datos 2 (205, 1.2)</i>	<i>Conjunto de Datos 3 (205, 1.3)</i>
---------------------------------------	---------------------------------------	---------------------------------------

Consumo de memoria 371 MB 400 MB 437 MB

Tabla 4: Consumo de memoria del algoritmo con diferentes conjuntos de datos

4.10 Análisis de los resultados

Expliquen los resultados obtenidos. Hagan una gráfica con los datos obtenidos, como por ejemplo:

Tabla de valores durante la ejecución			
Estructuras de autocompletado	LinkedList	Arrays	HashMap
Espacio en el Heap	60MB	175MB	384MB
Tiempo creación	1.16 - 1.34 s	0.82 - 1.1 s	2.23 - 2.6 s
Tiempo búsqueda ("a")	0.31 - 0.39 s	0.37 - 0.7 s	0.22 - 0.28 s
Tiempo búsqueda ("zyzzvyas")	0.088 ms	0.038 ms	0.06 ms
Búsqueda ("aerobacteriologically")	0.077 ms	0.041 ms	0.058 ms
Tiempo búsqueda todas las palabras	6.1 - 8.02 s	4.07 - 5.19 s	4.79 - 5.8 s

Tabla 5: Análisis de los resultados obtenidos con la implementación del algoritmo

REFERENCIAS

- [1] D. L.; Bixby, R. M.; Chvátal, V.; Cook, W. J (2007). Travelling_salesman_problem. https://en.wikipedia.org/wiki/Travellingsalesman_problem#History.
- [2] (Hois, s.f.). El problema del camino más corto sobre la red del Metrobús y del metro de la Ciudad de México (Solución por Dijkstra y Simplex Dual) <http://eenube.com/images/pdf/reporte.pdf>
- [3] Calviño, A. (2011). Cooperación en los problemas del viajante (TSP) y de rutas de vehículos (VRP): una panorámica.
- [4] R. Lewis, K. Smith-Miles, K. Phillips. (s.f). The School Bus Routing Problem: An Analysis and Algorithm. <https://core.ac.uk/download/pdf/158355938.pdf>