

TÍTULO (DESCRIPCIÓN CORTA DEL PROYECTO. ENTRE 8 Y 12 PALABRAS)

Laura Alzate Madrid
Universidad Eafit
Colombia
dalzatec1@eafit.edu.co

David Alzate Cardona
Universidad Eafit
Colombia
lalzatem@eafit.edu.co

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

RESUMEN

Medellín es una ciudad afectada por el alto tráfico, por ello la importancia del estudio sobre la reducción del tráfico. En este documento se abarcará el problema de optimización de los algoritmos para el uso del carro compartido o también conocido como carpooling. Todo esto con el fin de mejorar a la vez el estilo de vida en Medellín y el medio ambiente.

Palabras clave

Listas, backtracking, grafos tráfico, carpooling, algoritmo, costo mínimo.

1. INTRODUCCIÓN

Uno de los grandes problemas actuales de las ciudades es el alto tráfico, y Medellín es una ciudad gravemente afectada en diversos aspectos tales como la disminución de la calidad de vida y la contaminación.

En el siguiente documento se abarcará este problema y además una solución que en un futuro se espera que tenga un efecto muy positivo. En pocas palabras, se mostrara la estrategia del uso del carro compartido, también conocido como “carpooling”, con el fin de reducir considerablemente el tráfico.

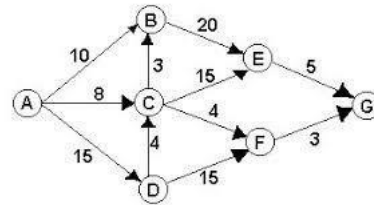
2. PROBLEMA

Se busca una manera efectiva de reducir el tráfico por medio de compartir los vehículos de personas que vayan a un mismo sitio, esto con el fin de reducir la cantidad de CO2 que se produce y la cantidad de vehículos que transitan al tiempo en las ciudades.

3. TRABAJOS RELACIONADOS

3.1 Gestión de envío

Empresas como servientrega o rappi diariamente se enfrentan al problema de recoger y entregar mercancía a diferentes zonas o localidades y estos lo deben de hacer en el menor tiempo posible, además de esto también cuentan con una carga limitada lo que hace que sea un problema más complejo.



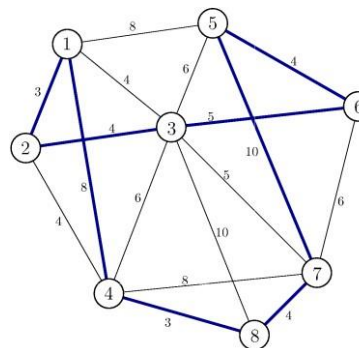
Grafica de un grafo dirigido para llegar de un punto a otro.

3.2 Problema del viajante (Travelling Salesman Problem)

(TSP))

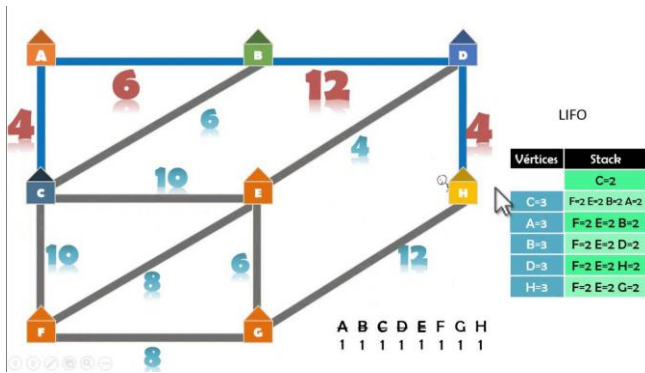
En este problema se intenta encontrar todos los recorridos posibles para visitar varias ciudades, sin repetirlas y volviendo a la ciudad origen, teniendo esto, lo siguiente es buscar la ruta más corta.

Una formulación equivalente en términos de Teoría de grafos es: dado una grafo ponderado completo (donde los vértices representan las ciudades, las aristas representan los caminos y los pesos son el costo o las distancias de estos caminos), encontrar un ciclo de Hamilton con menor peso.



3.3 Algoritmo del vecino más próximo

El algoritmo del vecino más próximo fue, en las ciencias de la computación, uno de los primeros algoritmos utilizados para determinar una solución para el problema del viajante. Este método genera rápidamente un camino corto, pero generalmente no el ideal.



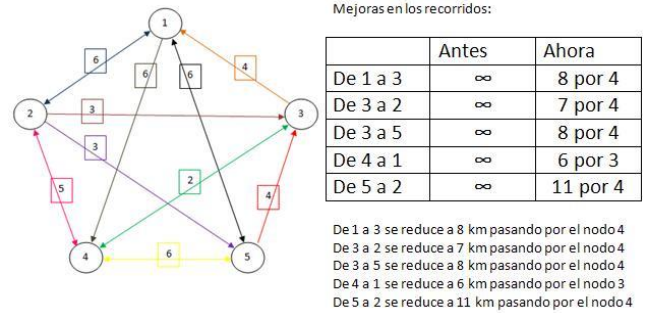
3.4 Algoritmo de búsqueda A*

El algoritmo A* es un algoritmo de búsqueda que puede ser empleado para el cálculo de caminos mínimos en una red. Se va a tratar de un algoritmo heurístico, ya que una de sus principales características es que hará uso de una función de evaluación heurística, mediante la cual etiquetará los diferentes nodos de la red y que servirá para determinar la probabilidad de dichos nodos de pertenecer al camino óptimo.

7	6	5	6	7	8	9	10	11		19	20	21	22
6	5	4	5	6	7	8	9	10		18	19	20	21
5	4	3	4	5	6	7	8	9		17	18	19	20
4	3	2	3	4	5	6	7	8		16	17	18	19
3	2	1	2	3	4	5	6	7		15	16	17	18
2	1	0	1	2	3	4	5	6		14	15	16	17
3	2	1	2	3	4	5	6	7		13	14	15	16
4	3	2	3	4	5	6	7	8		12	13	14	15
5	4	3	4	5	6	7	8	9	10	11	12	13	14
6	5	4	5	6	7	8	9	10	11	12	13	14	15

3.5 El algoritmo de Floyd

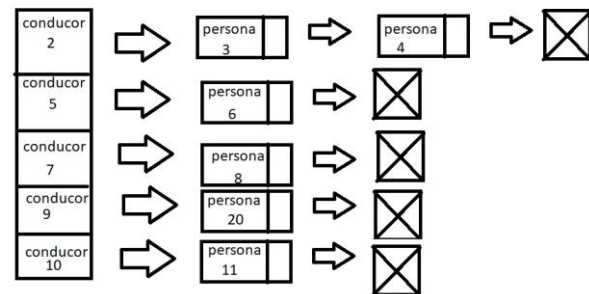
El algoritmo de Floyd es muy similar, pero trabaja con grafos ponderados. Es decir, el valor de la “flecha” que representamos en la matriz puede ser cualquier entero o infinito. Infinito marca que no existe unión entre los nodos. Esta vez, el resultado será una matriz donde estarán representadas las distancias mínimas entre nodos, seleccionando los caminos más convenientes según su ponderación (“peso”). Por ejemplo, si de “A” a “B” hay 36 (km), pero de “A” a “C” hay 2(km) y de “C” a “B” hay 10 (km), el algoritmo nos devolverá finalmente que de “A” a “B” hay 12 (km).



4. TITULO DE UNA SOLUCIÓN DISEÑADA

Las estructuras de datos que se decidió fue LinkedList, y Backtracking. Dichas listas almacenan toda la información sobre las permutaciones posibles en nuestro algoritmo. También se tiene un método que me retorna el camino más corto de un nodo a otro, evitando que se pase del tope de 5 personas (las que caben en el carro), y tratando de no sobrepasar por mucho, el tiempo que cada persona necesitaba para ir a trabajar solo en su carro. Esto lo hace leyendo un archivo y a la vez guarda la solución o soluciones es otro aparte.

4.1 Estructura de datos



Grafica 1: Lista dentro de una lista encadenada de personas (conductores). Cada conductor contiene su número y a quienes va a llevar en su auto.

4.2 Operaciones de la estructura de datos

4.3 Criterios de diseño de la estructura de datos

Decidimos usar estas estructuras de datos, ya que nos pareció que podíamos hacer un buen uso óptimo de las listas (LinkedList) y Backtracking lo usamos porque en un principio es fácil de implementar pero no muy rápido, sin embargo se adapta muy bien a la estructura del problema.

4.4 Análisis de Complejidad

Método	Complejidad
Insertar lista	O(1)

4.5 Algoritmo

4.6 Calculo de la complejidad del algoritmo

Sub problema	Complejidad
Leer archivo	O(n*n)
Escribir archivo	O(n)
Asignar un vehículo	O(n)
Camino más corto	O(n*n)
Complejidad Total	O(n*n)

REFERENCIAS

2. Fischer, G. and Nakakoji, K. Amplifying designers' creativity with domainoriented design environments. in Dartnall, T. ed. Artificial Intelligence and Creativity: An Interdisciplinary Approach, Kluwer Academic Publishers, Dordrecht, 1994, 343-364.

Miquel Àngel Estrada, El algoritmo que abre camino a los mensajeros

<https://www.agenciasinc.es/Noticias/Elalgoritmo-que-abre-camino-a-los-mensajeros>

Algoritmo de búsqueda A*
[https://es.wikipedia.org/wiki/Algoritmo_de_b%C3%BAsq%u%e9da_A*](https://es.wikipedia.org/wiki/Algoritmo_de_b%C3%BAsq%u%e9da_A%2A)