

Algoritmo para encontrar rutas óptimas para vehículos eléctricos

Tomas Atehortua Ceferino Universidad Eafit Colombia tatehortuc@eafit.edu.co	Mauricio Toro Universidad Eafit Colombia mtorobe@eafit.edu.co
--	--

NOTA DEL DOCENTE: Para ampliar información sobre los requerimientos aquí descritos, consulten la “Guía para la realización del Proyecto Final de Estructura de Datos 2” que se entrega. Al final: 1. Borrar este texto escrito en rojo, 2. Adecuar los espacios de los textos, 3. Cambiar el color de los textos a negro. Consideren además que:

Textos en negro = Es todo lo que deben hacer en la entrega 1

Textos en Verde = Es todo lo que deben hacer en la Entrega 2

Textos en violeta = Es todo lo que deben hacer en la entrega 3

RESUMEN

El objetivo de este proyecto es diseñar un algoritmo que encuentre rutas óptimas a los vehículos eléctricos, en este caso específico para hacer más eficiente la forma de distribuir mercancías. Esto con el fin de solucionar el problema que se presenta en las baterías de los coches por tener un rango de durabilidad limitado y un tiempo de carga muy largo. ¿Cuál es la solución?, ¿cuáles los resultados? y, ¿Cuáles las conclusiones? Utilizar máximo 200 palabras.

Palabras clave

Optimización - Grafos - Vehículos eléctricos - Rutas - Algoritmo

Palabras clave de la clasificación de la ACM

Theory of computation → Design and analysis of algorithms → Graph algorithms analysis → Shortest paths

Theory of computation → Design and analysis → Approximation algorithms analysis → Routing and network design problems.

Applied computing → Operations research → Transportation.

1. INTRODUCCIÓN

En el siguiente informe se pretende solucionar la problemática que tienen las empresas a la hora de entregar productos con camiones de carga eléctricos, para que puedan entregarlos de la manera más eficiente, teniendo en cuenta las horas de trabajo, número de autos y la vida útil de los Batería de coche. Resolver este problema permitiría mejorar las entregas y asegurar que las empresas utilizarán coches eléctricos, incluso con la necesidad de cargarlos, permitiendo

una reducción de las emisiones de carbono que liberan estos camiones de carga.

2. PROBLEMA

El problema radica en la baja eficiencia que tienen las baterías de los autos eléctricos para distribuir la mercadería, pues demoran mucho en hacer los recorridos debido a que la batería se descarga constantemente. Esto tiene un alto impacto en la sociedad porque si no se soluciona, los transportistas se verían obligados a utilizar otros vehículos más dañinos para el planeta, por eso para evitar que eso suceda es necesario solucionar este problema.

3. TRABAJOS RELACIONADOS

3.1 The Traveler Salesman Problem

Propuesta por William Hamilton y Thomas Kirkman, consiste en buscar el mejor camino que se pueda realizar en un conjunto de n nodos, pasando solo una vez por cada nodo hasta volver al primero. A lo largo de los años se han propuesto varias soluciones a este problema de optimización. Uno de ellos es el algoritmo de Christofides, que consiste en elegir y reemplazar aristas para mantener menos distancia. Es un algoritmo heurístico ya que su funcionamiento busca una ruta aproximada con los pesos que se plantean en los datos [1].

3.2 Vehicle Routing Problem (VRP)

El VRP es la distribución del Problema del vendedor viajero a diferentes vehículos. "Decide qué vehículo maneja qué solicitud en qué secuencia para que todas las rutas del vehículo puedan ejecutarse de manera factible" [2]. En este caso, una solución viable es el algoritmo de ahorro de Clark y Wright, que consiste en combinar dos rutas posibles, tomando los ahorros que se producen de cada una. Así, de forma aproximada, es posible atribuir el mejor recorrido para cada vehículo.

3.3 Electric Vehicle Routing Problem (E-VRP)

“A medida que las corporaciones se vuelven más conscientes del medio ambiente y los costos de externalidad asociados, los vehículos comerciales eléctricos están ganando terreno en las empresas que entregan productos” [3]. Bajo esta premisa, se creó una variante del problema de enrutamiento de vehículos, el E-VRP, que se enfoca en encontrar una estrategia de enrutamiento óptima con un costo mínimo de tiempo de viaje. costo de energía y vehículos eléctricos despachados. Hacer un enfoque heurístico como un

algoritmo inspirado en una búsqueda aleatoria o una solución de programación dinámica como Held Karp (con limitaciones de recursos) podría ser la mejor manera de abordar el EVRP.

3.4 Problema de las colonias de hormigas

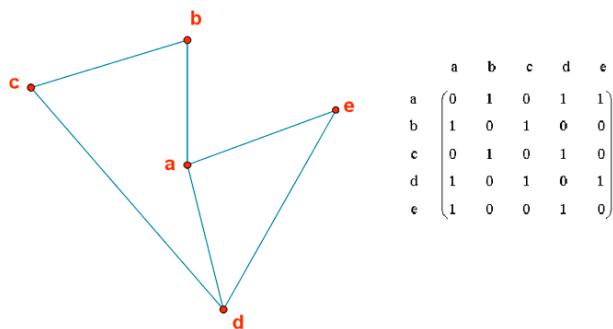
El algoritmo basado en las colonias de hormigas es utilizado para resolver varios problemas que requieren elegir la solución más corta entre varias posibles soluciones. [4] Lo que hace este algoritmo es construir las posibles soluciones e ir escogiendo cual es la óptima, para de esta manera, actualizar el peso o la distancia que hay entre cada nodo.

4. PRIMERA SOLUCIÓN DISEÑADA

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

4.1 Estructura de datos

Diseñen la estructura de datos para resolver el problema y grafiquenla. No usar gráficas extraídas de internet



Gráfica 1: Grafo representado como una matriz de adyacencia

4.2 Operaciones de la estructura de datos

Buscar vértices:

	0	1	2	3
0	-1	2	3	-1
1	6	-1	3	-1
2	-1	3	-1	5
3	4	6	2	-1

Gráfica 2: Dados dos nodos, retornar el vértice (peso), que hay entre los dos nodos, En este caso los que están de color amarillo corresponden al peso que hay para ir de un nodo al otro.

Buscar los vecinos:

	0	1	2	3
0	-1	2	3	-1
1	6	-1	3	-1
2	-1	3	-1	5
3	4	6	2	-1

Gráfica 3: Desde un nodo inicial encontrar todos los nodos a los que se pueden llegar, en este caso para el nodo 0 los vecinos son el 1 y el 2.

Buscar el vecino más cercano:

	0	1	2	3
0	-1	2	3	-1
1	6	-1	3	-1
2	-1	3	-1	5
3	4	6	2	-1

Gráfica 4: De todos los nodos a los que se puede llegar, retornar el que tiene el menor peso (el que está más cerca), en este caso para el nodo 0 el vecino más cercano es el 1.

4.3 Criterios de diseño de la estructura de datos

Elegí representar el grafo con una matriz de adyacencia debido a que su complejidad en el acceso a los datos es de $O(1)$. Con esta complejidad el algoritmo se hace mucho más eficiente a la hora de realizar las operaciones que requieren acceder a los pesos de las aristas de nuestro grafo. Además, teniendo en cuenta que los datasets van desde 320 hasta 360 nodos, la cantidad de memoria que se utiliza en esta matriz no es exorbitante comparada con la eficiencia en tiempo que ganamos.

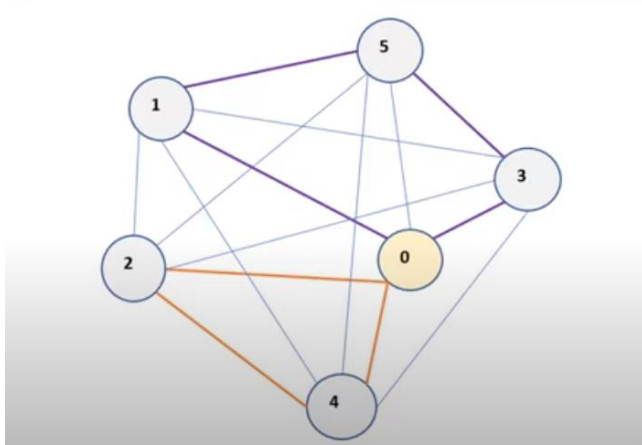
4.4 Análisis de Complejidad

La complejidad de los métodos es la siguiente, en la cual n corresponde al número de nodos

Método	Complejidad
Buscar un vertice	$O(1)$
Agregar un vertice	$O(1)$
Creación de la matriz	$O(n^2)$

Tabla 1: Tabla para reportar la complejidad

4.5 Algoritmo de CI



Gráfica 3: Ejemplo de los caminos (ahorros) que son encontrados por el algoritmo de Clarke-Wright

4.6 Cálculo de la complejidad del algoritmo

La complejidad del algoritmo para el peor de los casos, el mejor de los casos y el caso promedio

Sub problema	Complejidad
Determinar si un nodo se puede visitar	$O(n)$
Calcular el camino para cada vehiculo	$O(n^3)$
Incluir un nodo en un camino	$O(1)$
Encontrar los ahorros	$O(n^2)$
Complejidad Total	$O(n+n^2+n^3 + 1) = O(n^3)$

Tabla 2: complejidad de cada uno de los subproblemas que compoenne el algoritmo. Donde n es la cantidad de nodos del dataset,

4.7 Criterios de diseño del algoritmo

Después de analizar diferentes soluciones al problema, se decidió por implementar una solución basada en el algoritmo de Clarke-Wright. Este algoritmo ayuda bastante para resolver problemas en los que el número de vehículos no es concreto. En esta solución, se procura encontrar una ruta eficiente con limitaciones de tiempo y batería, ya que son vehículos eléctricos. Es bueno mencionar que este algoritmo no siempre encuentra la mejor solución, pero aun así es eficiente en tiempo y da una respuesta suficientemente valida. **4.8 Tiempos de Ejecución**

Calculen, (I) El tiempo de ejecución que consume el programa para varios ejemplos de dataset es el siguiente:

	Conjunto de Datos 1	Conjunto de Datos 2	Conjunto de Datos 3
Mejor caso	4 sg	3 sg	7 sg
Caso promedio	12 sg	10 sg	13 sg
Peor caso	30 sg	19 sg	25 sg

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

4.9 Memoria

La memoria que consume el programa para varios ejemplos de dataset es el siguiente:

	Conjunto de Datos 1	Conjunto de Datos 2	Conjunto de Datos 3
Consumo de memoria	0,74 MB	0,83MB	0,84 MB

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

4.10 Análisis de los resultados

Los resultados obtenidos son los siguientes:

Dataset	Tiempo de ejecución	Numero de clientes	Numero de vehiculos	Total tiempo de la ruta
1	12 s	320	33	2.1500 h
2	11 s	320	27	1,5995h
3	13 s	320	29	1,9882h
4	16 s	320	33	2,6587h

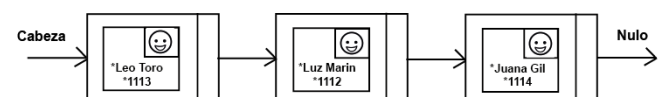
Tabla 5: Análisis de los resultados obtenidos con la implementación del algoritmo

5. SOLUCIÓN FINAL DISEÑADA

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

5.1 Estructura de datos

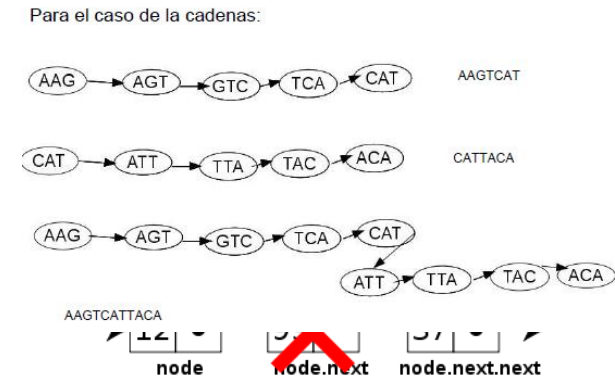
Diseñen la estructura de datos para resolver el problema y gráfiquenla. No usar gráficas extraídas de internet



Gráfica 4: Lista simplemente encadenada de personas. Una persona es una clase que contiene nombre, cédula y foto

5.2 Operaciones de la estructura de datos

Diseñen las operaciones de la estructura de datos para solucionar el problema eficientemente. Incluyan una imagen



explicando cada operación

Gráfica 5: Imagen de una operación de borrado de una lista encadenada

5.3 Criterios de diseño de la estructura de datos

Expliquen con criterios objetivos, por qué diseñaron así la estructura de datos. Criterios objetivos son, por ejemplo, la eficiencia en tiempo y memoria. Criterios no objetivos y que rebajan la nota son: “me enfermé”, “fue la primera que encontré”, “la hice el último día”, etc. Recuerden: este es el numeral que más vale en la evaluación con 40%

5.4 Análisis de Complejidad

Calculen la complejidad de las operaciones de la estructura de datos para el peor de los casos. Vean un ejemplo para reportarla:

Método	Complejidad
Búsqueda Fonética	$O(1)$
Imprimir búsqueda fonética	$O(m)$
Insertar palabra búsqueda fonética	$O(1)$
Búsqueda autocompletado	$O(s + t)$
Insertar palabra en TrieHash	$O(s)$
Añadir búsqueda	$O(s)$

Tabla 6: Tabla para reportar la complejidad

5.5 AGENTE VIAJERO, VECINO MÁS CERCANO

El algoritmo diseñado consiste en buscar el vértice más cercano al que se está ubicado en el momento, esto con el fin de encontrar una ruta optima.

Gráfica 6: Grafica donde se explique el algoritmo

5.6 Cálculo de la complejidad del algoritmo

Calculen la complejidad del algoritmo para el peor de los casos, el mejor de los casos y el caso promedio

Sub problema	Complejidad
Calcular la distancia entre todos los nodos	$O(V^2)$
Verificar tiempo y verificar la energia	$O(1)$
Encontrar el vecino mas cercano	$O(V^2)$
Encontrar las posibles rutas	$O(M)$
Crear grafo con listas de adyacencia	$O(V)$
Leer cada archivo y separar las variables	$O(N)$

Complejidad Total $O(V+V^2+M+1+V^2)$
 $= O(V^2)$

Tabla 7: complejidad de cada uno de los sub-problemas que componen el algoritmo. Sea V el tamaño o número de nodos que hay en el grafo. Y M el conjunto de clientes que han sido ya visitados por cada ruta, donde equivale a recorrer el tamaño del nodo V menos la cantidad de estacione, a su vez N indica la cantidad de archivos que se lee.

5.7 Criterios de diseño del algoritmo

Para la creación del algoritmo que da una posible solución al problema planteado se tuvo en cuenta la eficiencia tanto del tiempo como de la memoria empleada por este, y como la razón entre estos varía dependiendo de los algoritmos que usamos. Además de encontrar una solución óptima y que se adapte a los requisitos funcionales y no funcionales dados en el problema. Después de analizar las diferentes aplicaciones de las soluciones que se planteó para el problema, podría decir que lo más optimo, teniendo en cuenta también la razón entre el tiempo y la memoria fue usar un algoritmo voraz, específicamente la heurística del vecino más cercano; sin embargo, para adaptarlo al problema tuve que modificar el algoritmo, creando nuevas técnicas que optimizaran aún más el funcionamiento del este. Concluyendo así, que el vecino más cercano es una óptima solución ya que puede encontrar rutas optimas en donde se pueda emplear menos tiempo y energía y visitar varios clientes, utilizando más de un camión. Una solución simple, eficiente y eficaz.

5.8 Tiempos de Ejecución

	<i>Conjunto de Datos 1</i>	<i>Conjunto de Datos 2</i>	<i>...Conjunto de Datos n</i>
<i>Mejor caso</i>	10 sg	20 sg	5 sg
<i>Caso promedio</i>	12 sg	10 sg	35 sg
<i>Peor caso</i>	15 sg	21 sg	35 sg

Tabla 8: Tiempos de ejecución del algoritmo con diferentes conjuntos de datos

5.9 Memoria

Mencionar la memoria que consume el programa para varios ejemplos

<i>Conjunto de Datos 1</i>	<i>Conjunto de Datos 2</i>	<i>...Conjunto de Datos n</i>
----------------------------	----------------------------	-------------------------------

Consumo de memoria 10 MB 20 MB 5 MB

Tabla 9: Consumo de memoria del algoritmo con diferentes conjuntos de datos

5.10 Análisis de los resultados

Como bien sabemos el algoritmo implementado, basado en el algoritmo del vecino más cercano, tiene teóricamente una complejidad en el peor de los casos de $O(n^2)$, De lo que se puede concluir que, aunque la complejidad no sea la más optima, los resultados siguen siendo óptimos para la solución al problema planteado relativo a otros algoritmos gracias a los datasets utulizados, ya que el algoritmo se ejecuta rápidamente, para cada uno de los casos de prueba, manteniendo la eficiencia en cuanto al tiempo y memoria, aun cuando toma valores grandes.

Tabla 10: Análisis de los resultados obtenidos con la implementación del algoritmo

6. CONCLUSIONES

Para escribirlas, procedan de la siguiente forma: 1. En un párrafo escriban un resumen de lo más importante que hablaron en el reporte. 2. En otro expliquen los resultados más importantes, por ejemplo, los que se obtuvieron con la solución final. 3. Luego, comparen la primera solución que hicieron con los trabajos relacionados y la solución final. 4. Por último, expliquen los trabajos futuros para una posible continuación de este Proyecto. Aquí también pueden mencionar los problemas que tuvieron durante el desarrollo del proyecto

6.1 Trabajos futuros

Respondan ¿Qué les gustaría mejorar en el futuro? ¿Qué les gustaría mejor al algoritmo, estructura de datos, implementación?

AGRADECIMIENTOS

Identifiquen el tipo de agradecimiento que van a escribir: para una persona o para una institución. Luego, escríbalo de acuerdo al idioma y tengan en cuenta que: 1. El nombre del docente no va porque él es autor. 2. Tampoco sitios de internet ni autores de artículo leídos con quienes no se han contactado. 3. Los nombres que sí van son quienes ayudaron, compañeros del curso o docentes de otros cursos.

Aquí un ejemplo en inglés: This research was supported/partially supported by [Name of Foundation, Grant maker, Donor].

We thank for assistance with [particular technique, methodology] to [Name Surname, position, institution name] for comments that greatly improved the manuscript.

REFERENCIAS

- [1] Bernal, J., Hontoria, E. and Aleksovski, D., 2015. El problema del viajante de comercio: Búsqueda de soluciones y herramientas asequibles. ASEPUMA, 16(2), pp.117-133.
- [2] Irnich, S., Toth, P., and Vigo, D., 2014. Vehicle Routing Problems, Methods, and Applications. Philadelphia: Society for Industrial and Applied Mathematics.
- [3] Jane Lin, Wei Zhou, Ouri Wolfson, Electric Vehicle Routing Problem, Transportation Research Procedia, Volume 12, 2016, 508-521, <https://doi.org/10.1016/j.trpro.2016.02.007>.
- [4] Sancho, F. Algoritmos de hormigas y el problema del viajante. Recuperado Noviembre 17, 2016, del Dpto. de Ciencias de la Computación e Inteligencia Artificial de la Universidad de Sevilla: <http://www.cs.us.es/~fsancho/?e=71>