

OPTIMIZACIÓN DE RUTAS PARA VEHÍCULOS ELÉCTRICOS

Isabela Muriel Roldán
Universidad Eafit
Colombia
imurielr@eafit.edu.co

Mateo Flórez Restrepo
Universidad Eafit
Colombia
mflorezr@eafit.edu.co

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

RESUMEN

El objetivo de este informe es encontrar la manera de encontrar la ruta óptima en un mapa, de manera que se le pueda decir a un camión que necesite llegar a varios clientes el camino más corto y por donde gastará menos energía. Es importante solucionar este problema debido a que en el futuro la mayoría de los vehículos serán eléctricos y esto ayudará a decirle al usuario cual vía es la más apropiada para utilizar, y no solo para vehículos eléctricos, debido a que esto también puede favorecer a los demás vehículos.

Existen varios problemas relacionados a lo que se trata solucionar, entre ellos el problema de enrutamiento de vehículos o el del agente viajero, estos también serán explicados en este informe.

Finalmente se solucionó el problema utilizando un algoritmo que utiliza los principios de “dividir y conquistar” combinado con el vecino más cercano. Esto nos permitió volver el problema más pequeño utilizando varios camiones, cada uno con una ruta diferente hasta lograr que todos los clientes hayan sido visitados.

Palabras clave

Algoritmo, grafo, VRP, ruta óptima, energía, carros eléctricos, enrutamiento

Palabras de la clasificación de la ACM

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

1. INTRODUCCIÓN

El gran avance que ha tenido la tecnología en los últimos años ha generado el deseo de crear objetos inteligentes que le ayuden a las personas en su día a día, esto incluye a los automóviles, que además de querer hacerlos inteligentes, se desea reducir el gasto de combustible para evitar el nivel de contaminación que este genera. Debido a este deseo de cuidar el medio ambiente se han creado carros eléctricos, que utilizan energía como medio de combustible, el problema es que estos no pueden recorrer grandes distancias debido a que se pueden quedar sin energía y no encontrar un lugar donde se pueda recargar.

El objetivo de este proyecto es crear un algoritmo que le ayude a los camiones a encontrar la ruta óptima para llegar a una serie de clientes ubicados en un mapa.

2. PROBLEMA

Sabemos que actualmente la tecnología está tratando todo por ser amigable con el medio ambiente y disminuir todo aquello que perjudique a la sociedad y su entorno. Los vehículos eléctricos han sido un gran ejemplo de esto, una propuesta para reducir la dependencia del petróleo y los gases de efecto invernadero. Sin embargo, el uso de estos vehículos tiene límites en cuanto al rango de conducción y tarda mucho la carga de su batería. El problema en si es crear un algoritmo de recorrido en dos dimensiones que encuentre las rutas óptimas para que un conjunto de vehículos eléctricos visite cierta cantidad de clientes considerando un mínimo gasto neto de tiempo y energía.

3. TRABAJOS RELACIONADOS

3.1 El problema de enrutamiento de vehículos (VRP)

Es un problema de optimización combinatoria y de programación de entero qué pregunta "¿Cuál es el conjunto óptimo de rutas para una flota de vehículos que debe satisfacer las demandas de un conjunto dado de clientes?" Las implementaciones que más se usan para resolver este problema se basan en las heurísticas debido a que para grandes instancias del problema, que como sucede en ejemplos reales, producen buenos resultados. Existen muchas variaciones de este problema según las necesidades de optimización y requeridas para cada situación (Recorrida y entrega, ventajas de tiempo, capacidad, viajes múltiples, vehículo abierto).

Posibles soluciones:

Heurísticas: Las heurísticas son algoritmos con los que se encuentran soluciones aproximadas a las soluciones óptimas, en un tiempo computacional razonable (Olivera, 2004). Una de las heurísticas eficientes para la solución al problema es: La heurística del vecino más cercano.

La Heurística del Vecino más Cercano: Este algoritmo construye las rutas secuencialmente, seleccionando de manera iterativa los puntos con menor distancia para insertarlos a la ruta (Contreras P. & Díaz D., 2010)

3.2 Problema del agente viajero (TSP)

Este problema no es tan diferente al problema anterior, es un problema de optimización combinatoria que consiste en encontrar un recorrido completo que conecte todos los vértices pasando solo una vez por ellos y luego volver al punto de partida considerando el recorrido o distancia más corto en el mínimo de tiempo (Muy parecido a nuestro proyecto).

3.3 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 mas corta entre varias posibles soluciones. Lo que hace este algoritmo es contruir las posibles soluciones e ir escogiendo cual es la más óptima, para de esta manera, actualizar el peso o la distancia que hay entre cada nodo.

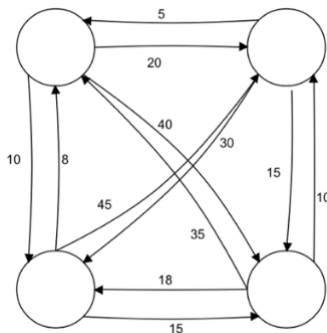
3.4 Algoritmos genéticos

Los algoritmos genéticos pueden ser utilizados para resolver problemas de búsqueda y optimización, estos están basados en el proceso genético de los organismos vivos debido a que lo que el algoritmo hace es crear soluciones e ir evolucionando de manera que las soluciones dadas sean las mejores y más óptimas.

4. 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.

4.1 Estructura de datos



Gráfica 1: Grafo completo para representar los puntos del mapa y las distancias entre ellos

4.2 Criterios de diseño de la estructura de datos

La razón por la que decidimos utilizar el algoritmo del agente viajero y solucionarlo por medio del vecino más cercano es que está es una solución que a pesar de no dar la solución óptima, resuelve el problema con una complejidad mucho menor que otras soluciones como fuerza bruta, que si da la ruta más corta pero el tiempo de ejecución es muy largo.

Lo que hace el algoritmo es recorrer el grafo, siempre buscando el vértice más cercano al actual, y los va marcando a medida que los recorre, esto asegura que no se repetirán. Además, a diferencia de la solución por medio de fuerza bruta o backtracking, el algoritmo no recorre un vértice más de una vez, lo que asegura que este será mucho más eficaz que los otros mencionados anteriormente.

4.3 Tiempos de ejecución

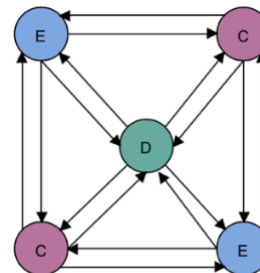
Conjunto de datos	Mejor tiempo	Peor tiempo	Tiempo promedio
tc2c320s24cf0.txt	131 ms	338 ms	177 ms
tc2c320s24cf1.txt	139 ms	365 ms	187 ms
tc2c320s24cf4.txt	137 ms	299 ms	183 ms
tc2c320s24ct0.txt	134 ms	273 ms	175 ms
tc2c320s24ct1.txt	135 ms	356 ms	183 ms
tc2c320s24ct4.txt	140 ms	480 ms	193ms
tc2c320s38cf0.txt	157 ms	368 ms	208 ms
tc2c320s38cf1.txt	149 ms	328 ms	199 ms
tc2c320s38cf4.txt	159 ms	406 ms	210 ms
tc2c320s38ct0.txt	148 ms	505 ms	210 ms
tc2c320s38ct1.txt	153 ms	329 ms	202 ms
tc2c320s38ct4.txt	159 ms	341 ms	209 ms

Tabla 1: Tiempos de ejecución del algoritmo con diferentes conjuntos de datos

5. VECINO MÁS CERCANO

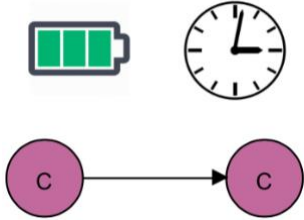
Para la solución final utilizamos el algoritmo del vecino más cercano con algunas modificaciones que nos permitieron dividir el problema con el objetivo de volverlo más eficiente en cuanto al tiempo y poder enviar más de un camión a recorrer el mapa.

5.1 Estructura de datos

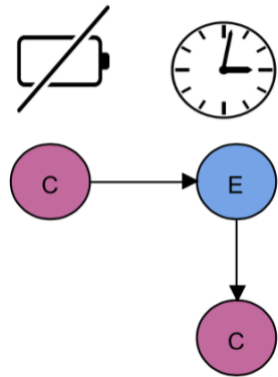


Gráfica 2: Grafo para representar los puntos de un mapa, los cuales pueden ser el deposito(D), clientes(C) o estaciones(E).

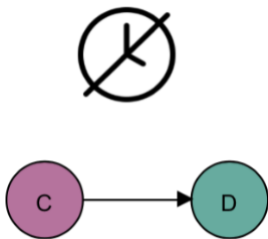
5.2 Operaciones de la estructura de datos



Gráfica 3: Imagen representando movimiento del camión. En caso de tener batería y tiempo, este continua la ruta y se dirige al cliente más cercano.



Gráfica 4: Representación de lo que sucede si se agota la batería y aún queda tiempo de recorrido, en este caso el camión busca la estación más cerna para recargarse y continúa con la ruta.



Gráfica 5: Representación de lo que sucede si se agota el tiempo, ya sea que haya batería o no el camión se devuelve al deposito y sale un nuevo camión para comenzar una nueva ruta.

5.3 Criterios de diseño de la estructura de datos

La razón por la que decidimos utilizar esta estructura de datos es que nos permite almacenar cada punto del mapa, ya sea depositito, clientes o estaciones. Además, nos permite hallar rápidamente cual es el sucesor más cercano y así poder aproximarnos a la respuesta óptima.

Escogimos esta solución debido a que con el método utilizado anteriormente podíamos recorrer el mapa únicamente con un camión por lo que no se tenía en cuenta el tiempo máximo que puede tener una ruta ni la batería del camión. En la solución final se considera tanto el tiempo como la batería y es posible utilizar varios camiones para

recorrer el mapa, esto hace que el problema sea cada vez mas pequeño debido a que nunca se visitan clientes que ya hayan sido visitados.

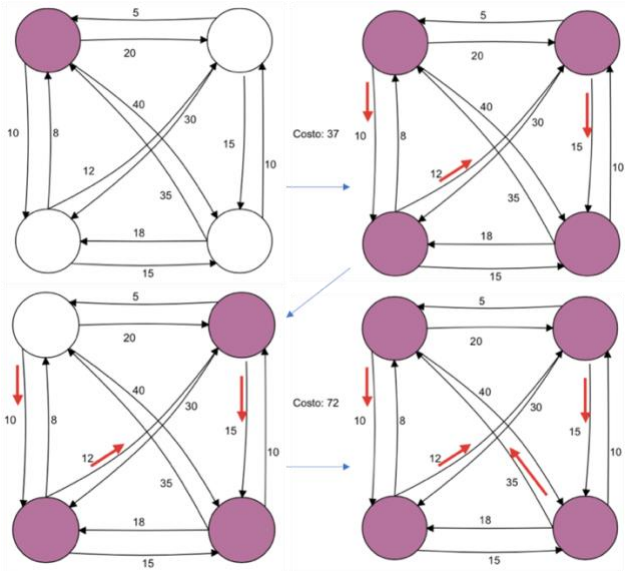
5.4 Análisis de Complejidad

Método	Complejidad
Leer el archivo	$O(1)$
Creación del grafo	$O(n)$
Calcular distancias entre nodos	$O(n^2)$
Verificar el tiempo y la energía	$O(1)$
Encontrarla ruta mas corta	$O(n^2)$
Imprimir ruta mas corta	$O(n^2)$

Tabla 2: Tabla para reportar la complejidad

5.5 Algoritmo

El algoritmo utilizado busca encontrar el vértice más cercano dependiendo del estado en el que se encuentre la batería y el tiempo, como se muestra en la gráfica 6 si se agota la batería y aún queda tiempo el camión se dirige a la estación más cercana. Si por el contrario el camión cuenta con batería y tiempo suficiente, este se dirige al cliente más cercano. En la gráfica 8 se muestra como es la selección de vértice cercano para continuar la ruta.



Grafica 6: Selección del vecino más cercano

5.6 Cálculo de la complejidad del algoritmo

Sub problema	Complejidad
Leer cada archivo separando variables	$O(n)$

Crear grafo de listas enlazadas	$O(V)$
Calcular la distancia entre todos los nodos	$O(V^2)$
Verificar tiempo y verificar la energía	$O(1)$
Encontrar el vecino mas cercano	$O(V^2)$
Encontrar las posibles rutas	$O(R)$
Complejidad total	$O(V+V^2+R+1) = O(V^2)$

Tabla 3: 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 R 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 estaciones.

5.7 Criterios de diseño del algoritmo

Para la creación del algoritmo que da una posible solución al problema planteado tuvimos 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 planteamos para el problema, podríamos decir que lo más óptimo, 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 debimos 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

Conjunto de datos	Mejor tiempo	Peor tiempo	Tiempo promedio
tc2c320s24cf0.txt	21 ms	694 ms	154 ms
tc2c320s24cf1.txt	16 ms	355 ms	123 ms
tc2c320s24cf4.txt	17 ms	785 ms	78 ms
tc2c320s24ct0.txt	17 ms	469 ms	150 ms
tc2c320s24ct1.txt	21 ms	542 ms	135 ms
tc2c320s24ct4.txt	16 ms	550 ms	65ms
tc2c320s38cf0.txt	17 ms	788 ms	77 ms

tc2c320s38cf1.txt	16 ms	499 ms	68 ms
tc2c320s38cf4.txt	17 ms	449 ms	63 ms
tc2c320s38ct0.txt	20 ms	619 ms	92 ms
tc2c320s38ct1.txt	17 ms	745 ms	109 ms
tc2c320s38ct4.txt	18 ms	619 ms	67 ms

Tabla 4: Tiempos de ejecución del algoritmo con diferentes conjuntos de datos

5.9 Memoria

Conjunto de datos	Memoria
tc2c320s24cf0.txt	8.5 MB
tc2c320s24cf1.txt	8.4 MB
tc2c320s24cf4.txt	8.5 MB
tc2c320s24ct0.txt	8.8 MB
tc2c320s24ct1.txt	8.4 MB
tc2c320s24ct4.txt	8.7 MB
tc2c320s38cf0.txt	8.7 MB
tc2c320s38cf1.txt	8.9 MB
tc2c320s38cf4.txt	7.5 MB
tc2c320s38ct0.txt	6.9 MB
tc2c320s38ct1.txt	7.9 MB
tc2c320s38ct4.txt	7.4 MB

Tabla 5: Consumo de memoria del algoritmo con diferentes conjuntos de datos

5.10 Análisis de los resultados

Como bien sabemos el algoritmo implementado, basado en la heurística del vecino más cercano, tiene teóricamente una complejidad en el peor de los casos de $O(n^2)$, según nuestros resultados en el cálculo de la complejidad (5.6). En el numeral 5.8 se realizó el cálculo experimental del tiempo de ejecución del algoritmo diseñado. De lo que concluimos que, aunque la complejidad no sea la más optima, los resultados siguen siendo óptimos para la solución al problema planteado relativo a otros algoritmos, 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.



Grafica 7: Análisis de los resultados obtenidos con la implementación del algoritmo

6. CONCLUSIONES

Un algoritmo debe ser eficiente en todos los criterios de su diseño, si quiere dar una solución óptima a un problema que se aplica a la vida real, como lo es el problema planteado en este reporte. Se deben tener en cuenta muchas razones y requisitos para llegar una solución final, que sea de calidad para dar complacencia a lo que se esta buscando. Teniendo en cuenta lo anterior, diseñamos un algoritmo que cumple con los requisitos de optimización no solo en tiempo y memoria, como lo comprueban los resultados finales que obtuvimos, si no también que nos llevó a la solución final que puede ser óptima para el planteamiento inicial.

En comparación al primer algoritmo diseñado, la solución final a la que llegamos fue principalmente un avance; una mejora de esta fase inicial, ya que se notó que el algoritmo inicial comparado con otros algoritmos o trabajos que investigamos tenía un buen potencial para llegar ser una buena opción para la solución total óptima. El problema principal fue escoger un algoritmo que tuviera una buena razón entre la complejidad, el tiempo y la memoria y que a su vez fuera optimo con el resultado esperado. Lo mejor que pudimos hacer fue dividir el problema en etapas, para crear soluciones paso a paso, que nos llevaran a la final.

Aunque llegamos a buenos resultados para esta etapa del problema, la solución aún se puede mejorar más, y tener un índice de error menor al actual, aun se puede tener un mejor manejo de la eficiencia tanto de la complejidad como de la memoria que se utiliza y llegar a resultados aún mejores. En un futuro con más investigaciones, análisis y procedimientos podremos llegar al propósito buscado.

6.1 Trabajos futuros

En un futuro se podría mejorar la eficiencia de la complejidad del algoritmo, buscar la más eficiente en cuanto a tiempo y memoria. También se podría proyectar la solución del algoritmo de manera gráfica, donde se muestre lo que hace el algoritmo paso a paso para llegar a

la solución. Se podrían reutilizar rutas, para acortar el trayecto de las rutas más largas, se utilizarían más objetos que influya en el problema. También se podría aplicar a otros problemas relacionados, o en otros mapas más grandes de otras ciudades del mundo, llevarlo un poco más a la vida real, como es el programa de Google maps, entre otros.

AGRADECIMIENTOS

Esta investigación fue parcialmente soportada por el icetex.

REFERENCIAS

1. Ramírez, L.E, una solución al problema de ruteo de vehículos abierto (OVRP), implementando la heurística del vecino más cercano. Recuperado Junio 20, 2016, de la facultad de ingeniería de la Universidad Distrital Francisco José de Caldas: <http://repository.udistrital.edu.co/bitstream/11349/2985/1/RamirezRodriguezLuisErnesto.pdf>
2. 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>
3. Wikipedia. Problema de enrutamiento de vehículos. Recuperado Diciembre 12, 2017. https://es.wikipedia.org/wiki/Problema_de_enrutamiento_de_veh%C3%ADcul
4. Caparrini, F. and Work, W. 2016. Algoritmos de hormigas y el problema del viajante. *Cs.us.es*. <http://www.cs.us.es/~fsancho/?e=71>.
5. Robles Algarín, C. 2010. *Optimización por colonia de hormigas: aplicaciones y tendencias*.