Universidad de Guadalajara

Ingeniería en computación

Nombre de los alumnos:

Bautista Rico Emmanuel Leonardo

González Medina María Emma Valeria

González Zaragoza Emmanuel Mateo

Zarate Ortíz Jesús Enrique

Clave: IL355

Materia: Análisis de algoritmos

Sección: D01

Profesor: Jorge Ernesto López Arce Delgado

Reporte - Proyecto final

Fecha de entrega: 27 de noviembre de 2023

Problema: Optimización de Rutas para un Servicio de Entrega Local

Planteamiento del problema

La compañía de servicios de entrega local cuenta con dos vehículos de carga y una lista de puntos de entrega, es decir, ubicaciones de clientes que deben visitarse.

El objetivo es encontrar las rutas más cortas para que los vehículos entreguen paquetes a todos los clientes y regresen al almacén central minimizando la distancia total recorrida.

Para la resolución del problema se estableció para cada camión una capacidad específica, el primer camión puede transportar 15 pedidos y el segundo camión puede transportar 10 pedidos.

El almacén ha sido establecido en las coordenadas (20.654986912292184, -103.32528602468746), es decir, CUCEI.

Roles

Bautista Rico Emmanuel Leonardo – Tester

González Medina María Emma Valeria – Project manager

González Zaragoza Emmanuel Mateo – Backend

Zarate Ortíz Jesús Enrique – Frontend

Algoritmo implementado

Implementamos el algoritmo de Clark y Wright, el cual es un método utilizado para resolver problemas de enrutamiento de vehículos, también conocido como el Problema de Ruteo de Vehículos (Vehicle Routing Problem, VRP).

Fue propuesto por los investigadores Stuart Clark y Wright en 1964, se trata de un algoritmo heurístico para encontrar soluciones aproximadas al Problema del Viajante de Comercio (TSP, por sus siglas en inglés). Este algoritmo se centra en optimizar las rutas de un conjunto de vehículos para satisfacer la demanda de un conjunto de clientes, minimizando la distancia total recorrida o algún otro criterio de optimización.

Se basa en la idea de fusionar rutas que conectan ciudades vecinas en una única ruta, siempre que mejore la solución actual. Esta técnica, conocida como el "algoritmo de ahorro de Clarke-Wright", ha demostrado ser efectiva en situaciones prácticas y es ampliamente utilizada en la optimización de rutas y la logística.

Objetivos

- Minimizar la distancia total recorrida: El objetivo principal es reducir la distancia total que los vehículos de entrega deben recorrer para satisfacer las demandas de los clientes. El algoritmo Clarke y Wright está diseñado para encontrar soluciones eficientes en términos de distancia, contribuyendo así a la eficiencia operativa y la reducción de costos.

- Optimizar las rutas considerando la capacidad de los vehículos: Tener en cuenta las capacidades de los vehículos, asegurando que se respeten los límites de carga durante las entregas, en este caso 15 y 10 pedidos respectivamente para cada camión.

- Mejorar la utilización de los vehículos: Optimizar la asignación de clientes a vehículos, permitiendo una mejor utilización de la capacidad de carga de cada vehículo. Esto contribuye a reducir el número de vehículos necesarios para satisfacer todas las demandas, lo que a su vez puede generar ahorros adicionales. En este caso solo contamos con dos vehículos, pero esperamos que el algoritmo se implemente en un futuro con más vehículos.

- Generar soluciones eficientes en términos de tiempo de ejecución: Obtener soluciones de manera rápida y eficiente, permitiendo la implementación práctica en entornos de tiempo real.

- Facilitar la implementación práctica y la aceptación del usuario: Al utilizar el algoritmo de Clarke y Wright, se busca una solución que no solo sea óptima en términos de distancia y capacidad, sino también práctica y fácil de implementar. La aceptación del usuario y la viabilidad operativa son fundamentales para el éxito de cualquier solución de optimización de rutas.

Tareas realizadas

Primera semana

Durante la primera semana se realizaron las siguientes actividades por etapas, en el backend se definió el algoritmo a implementar el cual fue, Clarke y Wright. Así mismo, se realizó una investigación sobre el uso de Git para importar archivos de GitHub desde Python y se hizo el diagrama de flujo.

Con respecto al frontend, se hizo el diagrama de flujo, se realizó una investigación sobre las posibles librerías con las que se podía desarrollar el front, se seleccionó la librería para realizar la interfaz gráfica de usuario (GUI), que es Tkinter; y se realizaron los bocetos de la GUI.

El tester realizó el diagrama de flujo y definió las restricciones del programa.

Además, el administrador de proyecto se encargó de la creación del repositorio en GitHub, de definir los objetivos y entregables. También, se creó el tablero de Trello, se dividió en semanas y cada una de las actividades se le asignaron a la persona correspondiente en un tiempo y formato establecido de entrega.

A continuación, se asignaron los canales de comunicación, los cuales fueron WhatsApp y Meet para las videollamadas. Después, se creó el diagrama de flujo de todo el programa.

- **Segunda semana**

En la segunda semana el integrante del equipo encargado del backend empezó con la codificación del programa en Python, se utilizó la versión 3.12.0. Se creó la matriz de adyacencia, las listas enlazadas; así como, la implementación del algoritmo Clarke y Wright.

En cuanto al frontend, se desarrolló la interfaz de usuario con la librería Tkinter en base a los bocetos que previamente se desarrollaron.

Por su parte, el tester generó tres archivos tipo JSON con datos de entrada para la posterior prueba del código.

- **Tercera semana**

Durante la tercera semana se continuó con la codificación del algoritmo para afinar detalles como la implementación del Big O, que se utiliza para analizar la complejidad del algoritmo.

Por consiguiente, en cuanto el algoritmo estuvo listo, se hizo la unión del backend con el frontend.

Además, el tester se encargo de hacer las pruebas necesarias para comprobar el correcto funcionamiento del programa y evaluar su rendimiento.

Un punto para destacar en la realización de este proyecto son las constantes reuniones de equipo que se hicieron a lo largo de las tres semanas, esto con el fin de tener sincronía en la realización de cada una de las partes.

Registro de problemas

Los problemas que se presentaron en el backend fueron:

- Conocer las limitantes a implementar en el algoritmo de nuestra solución al VRP.

- Delimitar cuales serían los datos de entrada y su obtención (decidir si iban a ser modificables o no).

- La precisión en las condicionales del algoritmo.

Por su parte, en el frontend se presentaron los siguientes problemas:

- Conexión de front-back, pasar la función "algoritmos_json" al front.

- Implementación del mapa por medio de un archivo html para mostrarlo en el navegador-web.

Por otro lado, durante el testing hubo complicaciones en cuanto:

- Creación de grafos

- Complicaciones en la comprobación de código por las diversas versiones de Python.

En general, los limitantes que como administradora de proyecto pude detectar fueron:

- Desconocimiento del uso de archivos tipo JSON para la generación de datos de entrada.

- Creación de grafos por la complejidad del problema.

- Tiempo debido a que nos encontramos en finales de semestre y tenemos que desarrollar diversas tareas y proyectos.

Continuamente se hizo revisión de recursos, incluyendo el tiempo y miembros del equipo. Esto con el fin de identificar si se estaban utilizando de manera eficiente.

Asimismo, se hizo revisión de calidad con el objetivo de conocer si las tareas realizadas semana con semana nos estaban acercando a los resultados o no. De igual manera, se mantuvo un control sobre los riesgos, es decir, identificar nuevos riesgos potenciales como el tiempo y proporcionar una solución a dichos inconvenientes.

Afortunadamente no tuvimos problema en cuestión de que alguno de los integrantes del equipo no pudiera realizar alguna de las tareas. Sin embrago, como equipo nos apoyamos para dudas técnicas que fueron surgiendo.

Los avances, así como el algoritmo y documentación de cada etapa se encuentran en GitHub.

Conclusiones

En conclusión, el código desarrollado en backend implementa el algoritmo Clarke-Wright para resolver el problema de optimización de rutas para un servicio de entrega local. Este algoritmo es una heurística de ahorro que se utiliza para encontrar rutas eficientes en problemas de distribución.

Por otro lado, en el frontend se utilizó la biblioteca Tkinter para crear una interfaz gráfica simple que permita al usuario ingresar coordenadas y nombres de destinos, agregar puntos de entrega, guardar esos puntos en un archivo JSON, y luego mostrar y visualizar las rutas calculadas en un mapa interactivo.

Además, con la creación de los archivos de datos de prueba se pudo comprobar el correcto funcionamiento del backend y frontend. Se realizaron diferentes pruebas, como la ejecución del código, prueba de la Interfaz Gráfica (GUI), prueba de código con los datos de entrada, prueba del algoritmo, prueba del mapa y el manejo de excepciones.

Como equipo aprendimos mucho, desde el uso de Git hasta la conexión de backend con frontend. Sin duda alguna, este proyecto nos ayudó a darnos cuenta de cómo es la realización de proyectos en la industria, la documentación necesaria y el seguimiento del trabajo en equipo que se debe implementar para obtener el resultado esperado.

Links de interés

- Link a GitHub
  https://github.com/EmmaValeria/ProyectoFinalAnalisis.git

- Link a Trello

  https://trello.com/invite/b/MkLco02V/ATTIeba80f5e4fb561e19e6cd0a0f66e44e3831CE991/proyecto-final-analisis

Bibliografía

Levitin, A. (2012b). Introduction to the design & analysis of algorithms.

Kuri, Á. (2002). Algoritmos genéticos: (ed.). Instituto Politécnico Nacional. https://elibro-net.wdg.biblio.udg.mx:8443/es/lc/udg/titulos/71925

Montiel, O. (2022, 22 febrero). La guía para principiantes de Git y GitHub. freeCodeCamp.org. https://www.freecodecamp.org/espanol/news/guia-para-principiantes-de-git-y-github/

Ismael, E. (s. f.). Article. https://pywombat.com/articles/json-python

Tkinter — Interface de Python para TCL/TK. (s. f.). Python documentation. https://docs.python.org/es/3/library/tkinter.html

Git. (s. f.). https://git-scm.com/

Welcome to Python.org. (2023). Python.org. https://www.python.org/