

Planificación Inteligente
Proyecto sobre COP
Prof. Carlos B. Ogando M.

PROYECTO VEHICLE ROUTING PROBLEM

HONESTIDAD ACADÉMICA

Como de costumbre, se aplica el código de honor estándar y la política de probidad académica. Las presentaciones isomórficas a (1) las que existen en cualquier lugar en línea, (2) las enviadas por sus compañeros de clase, o (3) las enviadas por los estudiantes en semestres anteriores serán consideradas plagio.

INSTRUCCIONES

En este proyecto construirán un planificador de rutas de vehículos de distribución de productos para resolver el VRP (Vehicle Routing Problem).

Lea atentamente todas las secciones de las instrucciones.

I. Introducción

II. Métodos de solución

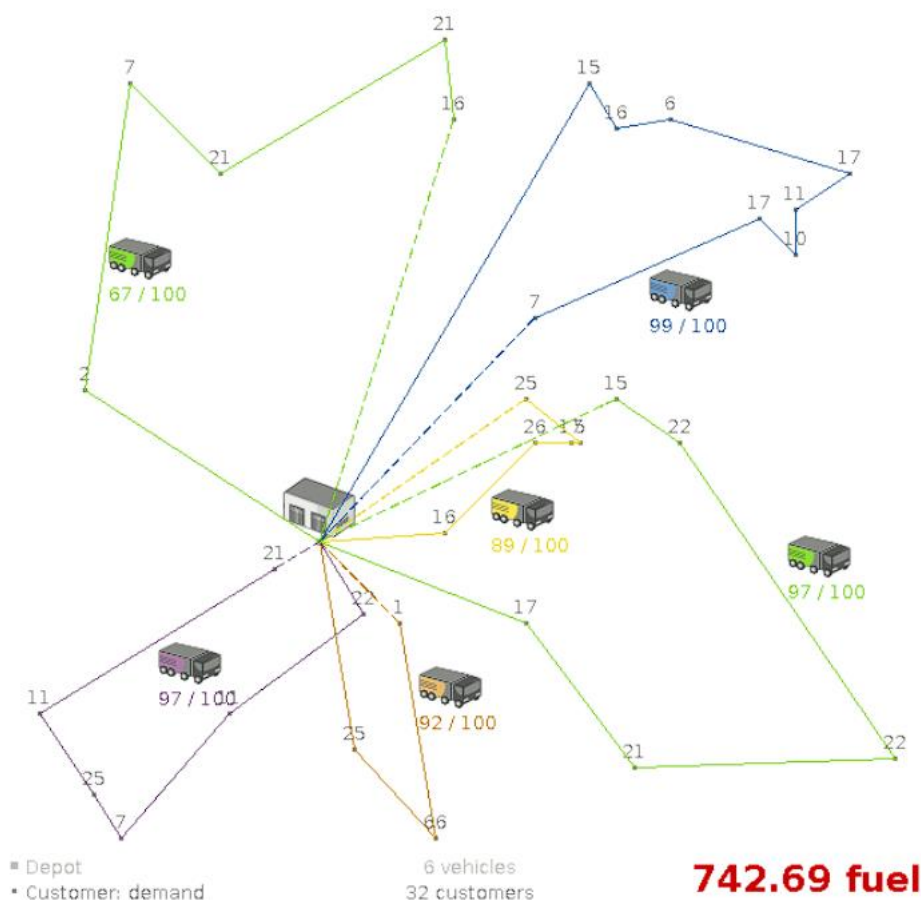
III. Qué necesita enviar

IV. Requerimientos de la entrega

I. Introducción

El problema de enrutamiento de vehículos (VRP, por sus siglas en inglés) es un problema de optimización combinatoria y de programación de entero que pregunta "¿Cuál es el conjunto óptimo de rutas para una flota de vehículos que debe satisfacer las demandas de un conjunto dado de clientes?". Es una generalización del conocido Problema del Viajante (TSP, por sus siglas en inglés). La primera definición aparece en un artículo de George Dantzig y John Ramser en 1959, en donde plantea una aproximación algorítmica y fue aplicado para entregas de gasolina. El problema, requiere la entrega de cierto producto, almacenado en un único local, a los clientes los cuales poseen cierta demanda; el objetivo fundamental es minimizar el coste total de las rutas trazadas. En 1964, Clarke y Wright mejoraron la aproximación de Dantzig y Ramser utilizando una aproximación "greedy" conocido como algoritmo de ahorros.

Utilizando una flota de vehículos, recoge los objetos de cada cliente y llévalos al depósito. Cada vehículo puede atender a múltiples clientes, pero tiene una capacidad limitada.

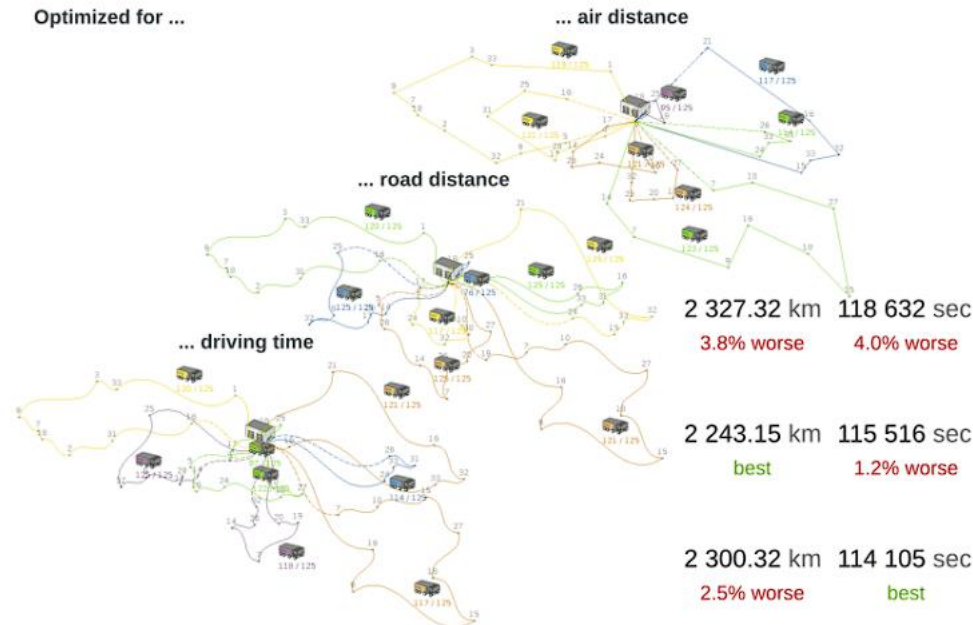


Además del caso básico (CVRP), también existe una variante con ventanas de tiempo (CVRPTW).

En el mundo real, los vehículos no pueden seguir una línea recta de un lugar a otro: tienen que usar carreteras y autopistas. Desde un punto de vista comercial, esto importa mucho.

Vehicle routing distance type

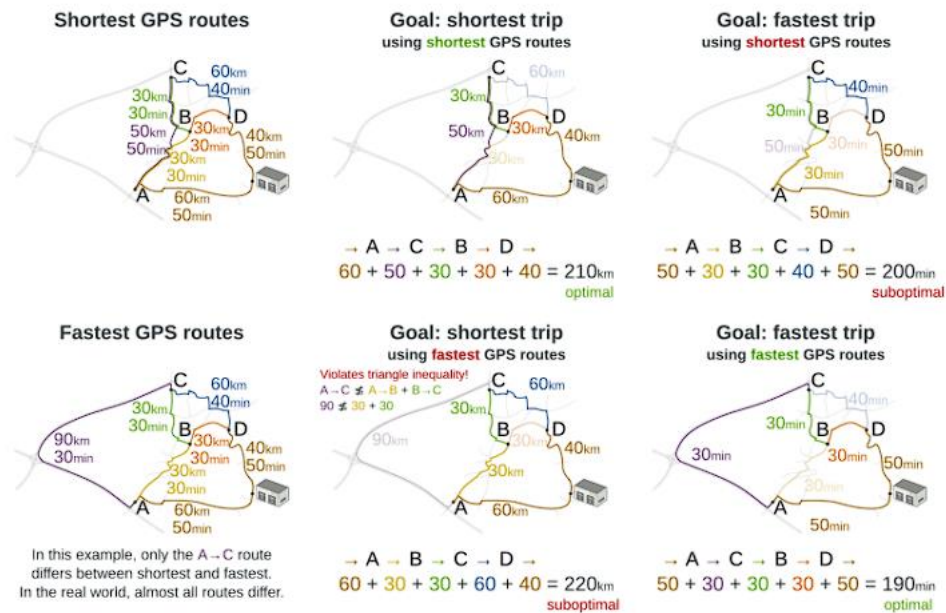
Can we optimize for air distances, when we need road distances or driving times?



Para el algoritmo de optimización, esto no importa mucho, siempre que la distancia entre dos puntos se pueda buscar (y preferiblemente se calcule previamente). El costo del camino ni siquiera necesita ser una distancia, también puede ser el tiempo de viaje, el costo del combustible o una función ponderada de estos. Hay varias tecnologías disponibles para precalcular los costos de las carreteras, como GraphHopper (motor Java incrustable y fuera de línea), Open MapQuest (servicio web) y Google Maps Client API (servicio web).

Road distance triangle inequality

Routes and trips must optimize the same property to avoid suboptimal solutions.



Contrariamente a la creencia popular, la mayoría de los usuarios no quieren la ruta más corta: prefieren la ruta más rápida. Prefieren las carreteras a las carreteras normales. Prefieren caminos normales a caminos de tierra. En el mundo real, la ruta más rápida y la más corta rara vez son iguales.

Para más información pueden revisar:

- La documentación de OptaPlanner sobre el problema de VRP
 - <https://docs.optaplanner.org/8.33.0.Final/optaplanner-docs/pdf/optaplanner-docs.pdf>
- Ejemplo de OptaPlanner para VRP
 - <https://www.youtube.com/watch?v=3Bgt4xC4a1M>

II. Métodos de solución

Aquí se enumeran las técnicas más utilizadas para resolver problemas de generación de rutas para vehículos. Casi todos ellos son heurísticos y metaheurísticos porque no se puede garantizar que ningún algoritmo exacto encuentre recorridos óptimos dentro de un tiempo de cálculo razonable cuando el número de ciudades es grande. Esto se debe a la dureza NP del problema. A continuación, podemos encontrar una clasificación de las técnicas de solución que hemos considerado:

2.1 Enfoques exactos

Como sugiere el nombre, este enfoque propone calcular todas las soluciones posibles hasta llegar a una de las mejores.

- Rama y atado
- Rama y corte
- Heurística

Los métodos heurísticos realizan una exploración relativamente limitada del espacio de búsqueda y, por lo general, producen soluciones de buena calidad en tiempos de computación modestos.

2.2 Métodos constructivos

Construya gradualmente una solución factible mientras vigila el costo de la solución, pero no contenga una fase de mejora per se.

- Ahorros: Clark y Wright
- Basado en coincidencias
- Heurística de mejora de múltiples rutas
- Thompson y Psaraftis
- Van Breedam
- Kinderwater y Savelsbergh

2.3 Algoritmo de 2 fases

El problema se descompone en sus dos componentes naturales: (1) agrupación de vértices en rutas factibles y (2) construcción de ruta real, con posibles bucles de retroalimentación entre las dos etapas.

- Algoritmos de clúster primero, segundo de ruta
- Fisher y Jaikumar
- El algoritmo del pétalo
- El algoritmo de barrido
- Taillard
- Algoritmos de ruta primero, grupo segundo

2. 4 Metaheurísticas

- Algoritmos de hormigas
- Programación de restricciones
- recocido determinista
- Algoritmos genéticos
- Recocido simulado
- Búsqueda tabú
- Tabú granular
- El procedimiento de la memoria adaptativa
- kelly y xu

III. Qué necesita enviar

Su trabajo en esta tarea es desarrollar una solución en Python que resuelve inteligentemente la programación de rutas de vehículos (VRP). Su programa debe contar con las siguientes características:

Restricciones duras:

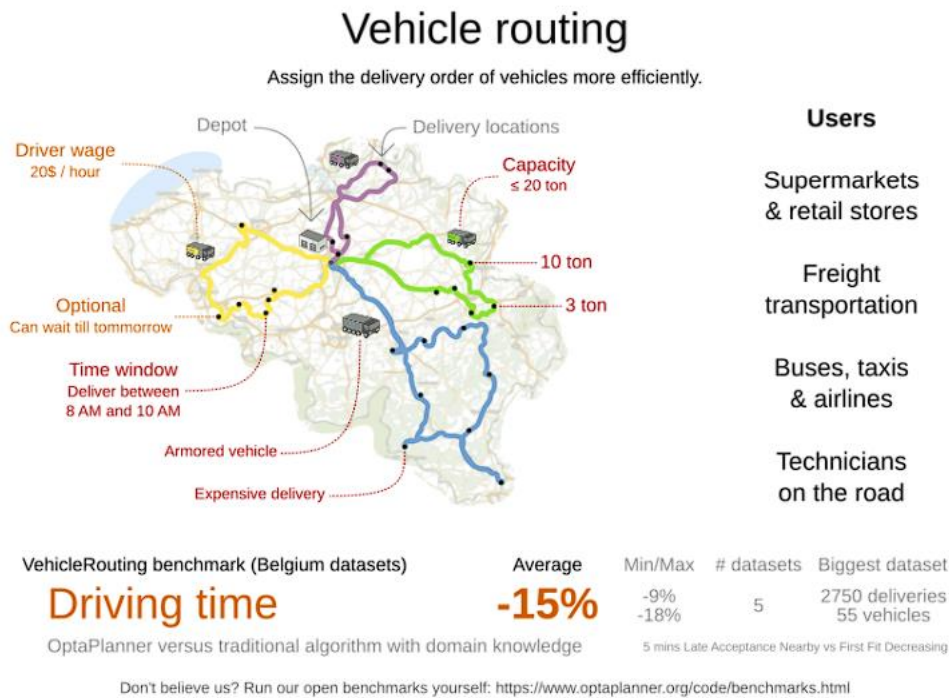
- Capacidad del vehículo: un vehículo no puede transportar más artículos que su capacidad.
- Ventanas de tiempo (solo en CVRPTW):
 - Tiempo de viaje: viajar de un lugar a otro lleva tiempo.
 - Duración del servicio al cliente: un vehículo debe permanecer en el cliente durante la duración del servicio.
 - Hora de preparación del cliente: un vehículo puede llegar antes de la hora de preparación del cliente, pero debe esperar hasta la hora de preparación antes de realizar el servicio.
 - Hora de vencimiento del cliente: un vehículo debe llegar a tiempo, antes de la hora de vencimiento del cliente.

Restricciones blandas:

- Distancia total: minimizar la distancia total recorrida (consumo de combustible) de todos los vehículos.

El problema de enrutamiento de vehículos capacitados (CVRP) y su variante de ventana de tiempo (CVRPTW) están definidos por la web VRP

- Su código debe ser hecho desde cero en Python en un cuaderno de Jupyter.
- Debe seleccionar uno de los algoritmos mencionados en el capítulo 2.



VI. Requerimientos de la entrega

En la entrega de este proyecto debe detallar:

- Recursos locales emplearán.
- Recursos en la nube usarán.
- Archivos cartográficos sin formatear de la provincia seleccionada.
- Arquitectura del sistema que proponen como solución.
- Diagrama de flujo de los datos y proceso.
- Procedimiento realizado para resolver el problema.
- Formato de salida propuesto de la información.
- Rol de los miembros del equipo y trabajo realizado.
- Costos de la propuesta.
- Métricas de efectividad del sistema propuesto.
- Dificultades al realizar el proyecto.
- Posibles mejoras de la propuesta.
- Riesgos y limitaciones.
- Código fuente.

NOTA:

Este proyecto **debe ser explicado en clase** por el grupo en la fecha indicada por el maestro, **de lo contrario se perderán todos los puntos** del mismo.