

Enrutamiento de Vehículos. Problema de distribución de balones de gas licuado en La Habana

Claudia Olavarrieta Martínez

CLAUDIA.OLAVARRIETA@ESTUDIANTES.MATCOM.UH.CU

C-511

Marcos Adrián Valdivié Rodríguez

MARCOS.VALDIVIE@ESTUDIANTES.MATCOM.UH.CU

C-512

Palabras Clave: VRP, OR-Tools,

Tema: Enrutamiento de Vehículos, Carga y descarga simultánea, Restricciones de tiempo. OR-Tools

1. Introducción

Los problemas de enrutamiento de vehículos resultan de gran interés pues resuelven problemas prácticos a compañías que deban distribuir o recoger algún tipo de mercancía. En la actualidad resultan de gran interés a la comunidad científica donde se han trabajado desde diversos enfoques para hallarles una mejor solución.

En este trabajo se presenta un problema práctico de enrutamiento de vehículos con carga y descarga simultánea referente a la distribución de balones de gas licuado a los distintos puntos de ventas existentes en La Habana. Actualmente la planificación de las rutas se realiza de forma manual, lo cual está a expensas de errores humanos que quizás provoquen que no se optimice el tiempo de entrega a los puntos de venta, lo que provoca desabastecimiento en estos lugares y afecta directamente a la población.

El objetivo de este trabajo es utilizar el software Google OR-Tools para modelar el problema y encontrar una solución que permita optimizar los tiempos de las rutas de los camiones. Para esto se desea minimizar el tiempo de la mayor ruta de los camiones lo cual disminuye a la vez el tiempo de espera de los puntos de venta para recibir la carga. Se analizarán los resultados obtenidos y la factibilidad de emplear este tipo de soluciones de forma real para planificar la asignación de las rutas en la empresa de Gas Licuado.

2. Desarrollo

2.1 Definición del Problema

El problema a resolver consiste en escoger las rutas para los camiones distribuidores de balones de gas a los distintos puntos de venta de La Habana. En la definición del problema se tiene lo siguiente:

- Se cuenta con una única base, la cual sería el depósito de donde salen los vehículos.
- El producto a entregar es único.
- Los camiones cuentan con un límite de capacidad, es decir tienen un número máximo de balones de gas (vacíos o llenos) que pueden cargar.
- Los puntos de venta tienen una demanda de balones que deben recibir.
- Los puntos de venta tienen un conjunto de balones vacíos por entregar.
- Existe un tiempo para efectuar la descarga y la carga de los balones en los puntos de venta una vez arribe el camión.
- La carga y descarga en los puntos de ventas deben efectuarse en la misma visita.

Debido a que puede darse el caso que los puntos de venta comiencen la jornada laboral en cero, es decir, no tienen balas de gas para vender, se necesita que la ruta de los camiones sea de tal forma que se minimicen los tiempos. Es decir que se minimice el tiempo de la mayor ruta que tomen los camiones.

2.2 Modelación del Problema

El problema puede ser representado en términos de un Grafo $G(V, A)$ completo y dirigido. Donde los vértices representan a los puntos de venta, excepto el vértice 0 el cuál es el depósito y es común a todas las rutas, y

el conjunto A son los arcos los cuales representan el tiempo de transportación entre puntos. El valor de cada arista $\langle i, j \rangle$ sería el tiempo promedio que llevaría a un camión recorrer la ruta desde el punto de venta i al punto de venta j . Se tiene una flota de m vehículos con capacidades distintas. El plan de rutas consistiría en un conjunto de circuitos sobre G tal que:

- El vértice de inicio y final sea 0.
- El único vértice común entre las rutas es el 0.
- Cada vehículo es asignado a una ruta.
- La carga del vehículo durante su recorrido es menor o igual que su capacidad.
- Cada cliente tiene asociado un valor para la demanda y la entrega de bienes.
- Cada cliente tiene asociado un tiempo de carga y descarga de mercancía.

2.2.1 RESTRICCIONES

2.3 Solución

Para resolver el problema se utilizó la suite **Google OR-Tools**, el cual es un paquete de software gratuito y de código abierto desarrollado por Google para resolver problemas de programación lineal, programación entera mixta, programación de restricciones, enrutamiento de vehículos y problemas de optimización relacionados. OR-Tools es un conjunto de componentes escritos en C++, pero proporciona envolturas para Java, .NET y Python.

2.3.1 TIEMPOS DE CARGA/DESCARGA

Para que la solución obtenida tuviese en cuenta los tiempos de carga y descarga en los clientes se decidió crear un grafo $G'(V', A')$, el cual consiste en, para cada nodo v que pertenece a un cliente, duplicarlo en un nodo v' y colocar una arista desde v a v' con costo igual al tiempo de carga/descarga del cliente v . Además, toda arista $u \rightarrow v$ del grafo original se elimina y se coloca una arista en el nuevo grafo desde u' hasta v . De esta forma se asegura que cuando una ruta para un vehículo dado pase por un nodo v , tenga en cuenta en el costo de la ruta final el valor de la arista $v \rightarrow v'$ que corresponde al tiempo de carga/descarga de dicho nodo.

```
dist = [[oo for _ in range(2 * nodes - 1)] for _ in range(2 * nodes - 1)]
        for i in range(nodes):
            for j in range(nodes):
                u = i if i == 0 else 2 * i
                v = j if j == 0 else 2 * j - 1
                dist[u][v] = distances[i][j]
            if i != 0: dist[2 * i - 1][2 * i] = load_time[i]
```

2.3.2 DEMANDAS DE LOS CLIENTES

Para obtener una solución que satisfaga la demanda de balones de gas de los clientes se añadió una dimensión, utilizando el método *AddDimensionWithVehicleCapacity* de OR-Tools. A esta dimensión se le especificó que la cantidad de balones entregados en todo momento de la ruta debe ser menor igual que la capacidad del vehículo y se le proporcionó un valor de false para el flag *fix_start_cumul_to_zero*, para de esta forma simular la cantidad de balones con los que parte inicialmente el vehículo.

```
routing.AddDimensionWithVehicleCapacity(deliveries_callback_index, 0,
                                         data['vehicle_capacities'], False, deliveries_str)
```

2.3.3 DEMANDAS DE ENTREGA DE BALONES VACÍOS DE LOS CLIENTES

Satisfacer las demandas de balones vacíos fue un poco más complicado, ya que se necesitaba además que la cantidad de balones vacíos y llenos que llevara el vehículo en cualquier momento fuera menor o igual que su capacidad, por tanto esta restricción no podía ser independiente de la de demanda de balones llenos. Para solucionarlo se añadió una dimensión para asegurar que la cantidad de balones vacíos recogidos menos la cantidad de balones entregados más la carga inicial sea siempre menor igual que la capacidad del vehículo. Esta dimensión se añadió con el flag *fix_start_cumul_to_zero* en false para simular el valor de la carga inicial.

```
routing.AddDimensionWithVehicleCapacity(pickups_callback_index, 0,
                                         data['vehicle_capacities'], False, pickups_str)
```

Además se añadió una restricción al modelo para asegurar que el valor inicial de la cantidad de balones que carga cada vehículo sea la misma para ambas dimensiones.

```
for idx in range(manager.GetNumberOfVehicles()):
    index = routing.Start(idx)
    routing.solver().Add(deliveries_dimension.CumulVar(index)
                        == pickups_dimension.CumulVar(index))
```

Demostremos que las dimensiones añadidas resuelven el problema de las demandas, sea E_i y R_i la cantidad de balones entregados y recogidos por un vehículo en una ruta cualquiera hasta el nodo i , sea C la capacidad de dicho vehículo y K la cantidad de balones que carga al iniciar la ruta:

- La primera dimensión es bastante evidente, basta con ver que esta restringe que $0 \leq K - E_i \leq C$, lo que indica que la cantidad de balones de gas entregados en todo momento será menor o igual que la capacidad del vehículo.
- La segunda dimensión asegura el cumplimiento de la restricción $0 \leq K + R_i - E_i \leq C$, esta restricción indica que la cantidad de balones que tenía inicialmente el vehículo, menos los entregados, más los recogidos, es siempre menor igual que la capacidad del vehículo, asegurando así que el vehículo tenga una cantidad de balones acorde a su capacidad.

3. Implementación de la solución

El código de la solución del problema puede encontrarse en el siguiente [repositorio de GitHub](#).

4. Conclusiones

El problema de enrutamiento de vehículos para la distribución de balones de gas licuado posee grandes dificultades para su solución debido a la cantidad de restricciones. En este trabajo se logró dar solución a una versión del mismo, con algunas de las cualidades más importantes, haciendo uso del software OR-Tools. Las ideas detrás de la solución antes expuesta pueden ser usadas para solucionar variaciones más completas del mismo, como puede ser incorporar ventanas de tiempo o múltiples viajes para un mismo vehículo.