Heurística para el problema de recogida de basuras

Maria Alejandra Soriano Castañeda

Universidad de los Andes
Cra. 1 #18a-12, Bogotá, Cundinamarca
ma.soriano@uniandes.edu.co

Abstract

Vehicle route planning has been considered advantageous, thanks to the wide variety of applications it has in various real-life situations, both in industrial and social settings. Therefore, the objective of this article has been to develop an optimization methodology that allows planning the routes for the garbage collection problem, minimizing the total distance traveled, subject to the capacity restriction of a single vehicle.

Index Terms—Routing problem, heuristics.

1 Introducción

El problema de enrutamiento de arco capacitado (CARP) es un problema de optimización combinatoria que puede ser categorizado como un problema de ruteo de vehículos con capacidad limitada (CVRP), así como, una variante del problema del agente viajero (TSP). El problema de enrutamiento de arco capacitado consiste en determinar los recorridos que deben realizar un conjunto de vehículos, de forma que se minimice la distancia total recorrida o los costes de arco en que se incurren al atravesar los arcos de una red de nodos sujeto a la capacidad de los vehículos. Este problema pertenece a la clase de problemas NP Hard, para los cuales la corrección de la solución no se puede comprobar en tiempo polinomial a optimalidad (Programador clic, 2018).

A lo largo de los años se han propuesto diversos algoritmos para hacer frente a las diferentes variaciones del problema de enrutamiento de arco capacitado puesto que son bastante funcionales, ya que se pueden aplicar a varias situaciones en el mundo real, así como: entrega de correos, mantenimiento de calles, rutas de autobuses, entre otros.

El objetivo de este artículo es presentar una heurística constructiva que encuentre una solución factible al problema de recolección de basuras sujeto a la restricción de capacidad de un único vehículo que deberá recorrer toda la red de arcos no dirigidos, dicha heurística constructiva está basada en el algoritmo heurístico propuesto por (Pearn,1989). Por otra parte, el rendimiento del algoritmo implementado se comparará con un método exacto y con las instancias proporcionadas por Golden (Golden & Wong,1981).

2 Descripción de la problemática

El problema a resolver en este artículo consiste en determinar la mejor ruta de distribución para un único vehículo capacitado que debe recolectar basura en las calles de una zona geográfica compuesta por un conjunto de 22 calles (arcos) unidas por 12 nodos, en los cuales se incluye un centro de distribución. El vehículo inicia su recorrido en el centro de distribución y debe recorrer toda la red de arcos no dirigidos para realizar la recolección de basura que se encuentra en cada calle, en este caso la red del problema posee una tonelada de basura en cada arco. De este modo, las calles pueden ser limpiadas una sola vez y en el momento que el vehículo llegue a su capacidad máxima de cinco toneladas tendrá que retornar al centro de distribución para realizar la descarga de la carga e iniciar una nueva ruta si aún no se ha recorrido toda la red. En este caso, el objetivo del problema es la minimización de la distancia total recorrida, considerando que cada ruta comienza desde el centro de distribución y termina en el mismo centro, y la capacidad del vehículo no debe excederse.

3 Formulación matemática

El problema puede formularse matemáticamente mediante programación lineal entera (PLE) como sigue Golden and Wong.

$$Minimizar \sum_{i=1}^{n}\sum_{j=1}^{n} C_{ij} \cdot X_{ij}$$

$$Maximize \sum_{n=1}^{N} V_n \cdot p_n \cdot Z_n + \sum_{q=1}^{Q} d_q \cdot Y_q$$

$$\sum_{n=1}^{N} Z_n = \sum_{q=1}^{Q} X_q$$

$$\sum_{q=1}^{Q} X_q = 0$$

$$X_q, Y_n, Z_n \in \{0,1]$$

Sujeto a

$$\sum_{k=1}^{n} X_{ik} - \sum_{k=1}^{n} X_{ki} = 0, \qquad \forall\, i \in n \tag{2}$$

$$(L_{ij} + L_{ji}) = q_{ij}, \qquad \forall\,(i,j) \in G \tag{3}$$

$$X_{ij} \geq L_{ij}, \qquad \forall\,(i,j) \in G \tag{4}$$

$$\sum_{i=1}^{n}\sum_{j=1}^{n} L_{ij} \cdot q_{ij} \leq W \tag{5}$$

$$X_{ij},\ \ L_{ji} \in \{0,1] \tag{6}$$

La Ecuación 1 indica la función objetivo del problema que busca minimizar la distancia total recorrida, dependiendo de la longitud del arco C_{ij} y de la variable binaria X_{ij} la cual toma el valor de uno si el arco (i,j) fue atravesado o cero de lo contrario. La Ecuación 2 asegura la continuidad lógica de la ruta; es decir, si el arco(i,j) es atravesado esto significa que el arco (j,i) también se recorrió. La Ecuación 3 garantiza que cada arco fue atendido/limpiado exactamente una vez. La Ecuación 4 evita que un arco sea limpiado si este no fue atravesado. La Ecuación 5 restringe que la capacidad W del camión sea excedida. Finalmente, la Ecuación 6 habla sobre la naturaleza de las variables.

4 Metodología

El objetivo de la heurística que se va a desarrollar es construir exactamente una ruta, dado que se cuenta con un único vehículo con capacidad (W), de modo que cada arco o arista en la red sea visitado al menos una vez. Cada vez que el vehículo pasa por el depósito (nodo uno), la capacidad total del camión vuelve a estar disponible. Además, cada arco $e \in G$ tiene un costo asociado $C_{ij} > 0$ y una demanda q_{ij}.

Tomando como base el algoritmo de inserción de aumento propuesto por Pearn, se puede encontrar una solución factible al CARP como se presenta a continuación:

Paso 1. Definir los conjuntos G y G' siendo estos el conjunto de todos los arcos que componen el grafo y el conjunto de arcos que faltan por servicio de limpieza ($L_{ji} = 0$), respectivamente.

Paso 2. Sea *Dij* la suma de las longitudes de los arcos (deposito, *i*) y (*j*, deposito) que pertenecen a *G*, determinar *Dij* para cada arco del conjunto *G*.

$$D_{ij} = l_{1,i} + l_{j,1}$$

Paso 3. Seleccionar el arco (v_i, v_j) con mayor valor *Dij* que pertenezca al conjunto *G'*, es decir que aún no se haya servido.

Paso 4. Crear una ruta $C = SP_{0i} + (v_i, v_j) + SP_{j0}$ que comience y termine en el depósito (v_1), y ademas contenga el arco (v_i, v_j) con mayor valor *Dij* de forma que la distancia entre el arco y el depósito sea aquella que minimice la distancia total de la ruta. Inicialmente se considera que el arco (v_i, v_j) es el único arco servido de la ruta *C*.

Paso 5. Si el vehículo aún no ha superado su capacidad, ir modificando el estatus de los arcos que hacen parte de la ruta *C* y no han sido limpiados $(L_{ji} = 0)$ a limpiados $(L_{ji} = 1)$. Esto se debe realizar de manera consecutiva iniciando en el primer nodo que se atraviesa de la ruta *C* hasta que la restricción de capacidad vaya a ser violada. Cuando el vehículo vuelva al depósito el peso de la carga vuelve a cero.

Paso 6. Remover los arcos servidos del conjunto *G'*. El algoritmo se detiene en el momento que el conjunto *G'* quede vacío y todos los arcos hayan sido cubiertos. Si quedan arcos que no han sido limpiados, repetir el paso tres.

5 Análisis de resultados.

Considerando el caso del presente artículo, donde la capacidad del vehículo es de cinco toneladas y cada arco tiene una tonelada de demanda. El algoritmo de inserción de aumento para la primera instancia de Golden and Wong genero once rutas para la recolección de basura, cuyo costo fue de 509. Las rutas se presentan a continuación: [1, 12, 6, 7, 8, 11, 9, 2, 1], [1, 10, 11, 9, 2, 1], [1, 12, 6, 5, 11, 9, 2, 1], [1, 12, 6, 7, 8, 10, 1], [1, 12, 6, 5, 3, 4, 1], [1, 2, 9, 10, 1], [1, 2, 3, 5, 6, 12, 1], [1, 2, 4, 1], [1, 12, 6, 5, 12, 1], [1, 12, 6, 7, 12, 1], [1, 7, 6, 12, 1]

En la primera iteración del algoritmo constructivo, durante el paso tres, se seleccionó el arco (8,10) y en el paso cuatro se determinó la ruta *C* la cual se compone de (1) el recorrido que representa menor costo para llegar al nodo ocho desde el depósito [1, 12, 6, 7, 8] y (2) el recorrido que representa menor costo para ir del nodo 11 hasta el depósito [11, 9, 2, 1]. En este paso, se asume que únicamente se limpiara el arco seleccionado en el paso tres. Posteriormente, en el paso cinco, dado que el vehículo aun cuenta con capacidad de almacenamiento, se procedió a ir limpiando los arcos, que no han sido servidos, de manera consecutiva iniciando desde el depósito hasta antes de exceder la capacidad del vehículo. De esta forma, en esta primera ruta se limpiaron los arcos (1, 12), (12, 6), (6,7), (7, 8) y (8, 11). Así sucesivamente se repiten los pasos presentados en la sección metodología.

Cabe resaltar que la heurística constructiva tiene la ventaja de tener un bajo costo computacional, sin embargo, al aplicarla a instancias de gran escala los resultados pueden ser ineficientes. El algoritmo heurístico se diseñó e implementó en el lenguaje de programación de Python para agilizar el análisis experimental.

Analizando los datos de la tabla 1, donde se presentan los resultados del algoritmo realizado por Golden, el algoritmo del método exacto propuesto por Belenguer y los resultados de la heurística aplicada en este artículo. Podemos afirmar que se obtuvo una brecha promedio del 75.1% entre el valor de la función objetivo del CARP al aplicar la heurística constructiva desarrollada en este artículo, basado en el método de Pearn, y el valor objetivo de las primeras siete instancias obtenido con el primer método (IMP1) presentado en Golden and Wong. Por otro lado, la brecha promedio comparando la solución encontrada (heurística constructiva) con los resultados óptimos del algoritmo de plano de corte (CPA2) propuesto por Belenguer es del 71.8%.

Tabla 1: Resultados de la heurística constructiva versus los resultados de los algoritmos de Golden y Belenguer.

Instancia	**IMP1**	**LB**	**CPA2**	**Heurística constructiva**	**GAP (IMP1)**	**GAP (CPA2)**
1	351	310	316	509	64.2%	61.1%
2	394	339	339	646	90.6%	90.6%
3	316	275	275	456	65.8%	65.8%
4	316	274	287	472	72.3%	64.5%
5	429	370	377	673	81.9%	78.5%
6	340	295	298	538	82.4%	80.5%
7	325	312	325	525	68.3%	61.5%
GAP Promedio					75.1%	71.8%

Considerando el costo de las siete instancias para cada uno de los métodos usados, se puede observar que tanto el primer método usado por Golden como el método exacto de Belenguer obtienen menores costos y por ende una solución más acertada para el CARP, el cual busca minimizar los costos. Los grandes valores del GAP pueden ser ocasionados debido a que para el desarrollo de la heurística presentada se replicó únicamente la primera parte del método propuesto por Pern, el método de aumento.

6 Conclusión

Los resultados obtenidos intentando replicar los primeros pasos del algoritmo augment-insert para resolver el problema de enrutamiento de arco capacitado (CARP), con restricciones de capacidad para un único vehículo, permiten concluir que la aplicación de la heurística presentada puede exhibir un comportamiento uniforme que permite obtener soluciones factibles ante situaciones del mundo real para el caso de ruteo de vehículos.

Los anteriores resultados pueden llevar a concluir que esta alternativa podría resultar conveniente en la práctica con instancias de pequeña escala; sin embargo, se demostró que existen otros algoritmos más eficientes, como lo es el algoritmo de Golden and Wong y el método exacto de Belenguer, para resolver problemas de ruteo de arcos capacitados. Finalmente, se puede afirmar que el objetivo de implementar un algoritmo que permita encontrar una ruta de transporte factible al problema de recolección de basura se cumple a cabalidad.

References

Belenguer, J.M. & E. Benavent (1998). *A cutting-plane algorithm for the capacitated arc routing problem*. Working Paper, Dept. of Stats and OR, University of Valencia, Spain

Golden BL, Wong R. (1981). *Capacitated arc routing problems*. Networks,11(3), 305–15.

Pearn, W. L. (1989). *Approximate solutions for the capacitated arc routing problem*. Computers and Operations Research, 16(6), 589–600.

Programador clic. (2018). *[Turn] La diferencia entre NP-Hard y NP-Complete - programador clic*. Programmerclick.com. Recuperado el 3 de septiembre de 2022 de https://programmerclick.com/article/86831692464/