Proyecto Análisis de Algoritmos – Etapa 1

Sergio Oliveros, Omar Vargas y Stiven Viedman

Modelaje del Problema

Imaginemos que una empresa desea utilizar un vehículo eléctrico para repartir sus productos a un conjunto específico de clientes. El objetivo de la empresa es determinar la ruta de menor distancia que permita visitar a todos los clientes y regresar al punto de origen. Sin embargo, además de esta tarea, la empresa debe garantizar que se cumplan las siguientes restricciones:

1. La batería del vehículo eléctrico no debe agotarse en ningún punto del trayecto. Por lo que para ello este puede recargar su batería en un conjunto de estaciones determinado.
2. Cada cliente tan solo puede recibir su producto durante una ventana de tiempo específica. Por lo que el vehículo eléctrico debe visitar al cliente durante dicha ventana de tiempo.

Con base a lo anterior, el nombre formal que Roberti (2016) propone para este problema es *Electric Traveling Salesman Problem with Time Windows (E-TSPTW)*. Debido a que la base del problema el *Traveling Salesman Problem (TSP)*, pero con las restricciones adicionalmente descritas.

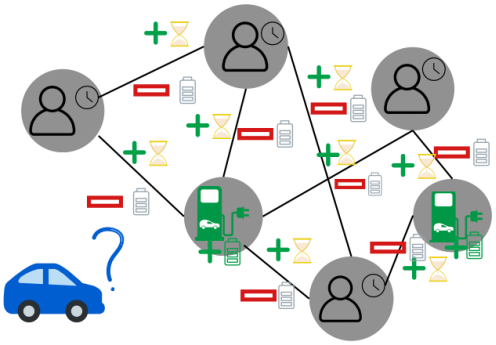

Fig. 1. Representación gráfica aproximada del problema

La formulación que Roberti (2016) propone para este problema es la siguiente:

- Sea V un conjunto de vértices cuya definición es $V = \{o, d\} \cup C \cup S$. Donde $\{o, d\}$ son los vértices de origen y destino, respectivamente, de manera que $o = d$. Sin embargo, para simplificar notación se hace la distinción. C es el conjunto de clientes a visitar. S es el conjunto de estaciones de recarga.
- Sea A un conjunto de ejes/arcos que está definido como $A = A_C \cup A_S$. Donde A_C es el conjunto de arcos entre los clientes, incluyendo los arcos desde el origen y hacia el destino. Y A_S es el conjunto de arcos desde los clientes hacia las

estaciones de recarga, desde las estaciones de recarga hacia los clientes y entre las estaciones de recarga.

- Cada vértice $i \in C \cup \{o, d\}$ tiene asociada una ventana de tiempo definida como $[e_i, l_i]$ donde $e_i, l_i \in \mathbb{Z}^+$. e_i indica el instante de tiempo más temprano en que se puede repartir un producto, mientras que l_i indica el instante de tiempo más tarde que se puede repartir un producto, por lo que $e_i \leq l_i$. Se hace la aclaración de que las estaciones de recarga no tienen ventana de tiempo en la cual pueden ser utilizadas.
- Cada $(i, j) \in A$ tiene asociada una distancia d_{ij}, un tiempo de recorrido t_{ij} y un consumo de batería q_{ij}. Con $d_{ij}, t_{ij}, q_{ij} \in \mathbb{Z}^+$.
- Finalmente, el vehículo eléctrico cuenta con una batería con capacidad $Q \in \mathbb{Z}^+$.

Entradas y salidas:

Con base a lo anterior se formulan las entradas y salidas del problema:

E/S	Nombre	Tipo	Descripción
E	$G(V, A)$	Grafo no dirigido	Grafo que contiene los vértices de origen y destino, los clientes a visitar, las estaciones de recarga y los arcos que se pueden tomar entre ellos. Los vértices y los arcos contienen las cantidades $e_i, l_i, d_{ij}, t_{ij}, q_{ij}$ intrínsecas al problema.
E	Q	Entero positivo	Capacidad de la batería del vehículo.
S	$Ruta$	Arreglo ordenado de vértices	El orden en el cual se deben recorrer los vértices de $G(V, A)$. Los vértices de S pueden ser visitados más de una vez.

Precondición:

La precondición especifica las condiciones que las entradas deben cumplir para ser válidas en el problema. En otras palabras, si la precondición no se cumple, la existencia de una solución no está garantizada. Por lo tanto, se detallan a continuación las condiciones que el grafo y la capacidad de la batería de entrada deben cumplir, primero en lenguaje natural y luego de manera formal.

- Para empezar, es necesario asegurar la existencia de una ruta que permita visitar a todos los clientes. La manera más sencilla de garantizar esta propiedad es asumir que el grafo de entrada es fuertemente conexo.
- Adicionalmente, dado que se cuenta con ventanas de tiempo, resulta necesario garantizar que es posible visitar a los clientes en sus ventanas de tiempo. Por lo que se define una nueva variable $z_i \in \mathbb{Z}^+$ que especifica el instante de tiempo en que se visita el vértice $i \in \{o, d\} \cup C$. De manera que se debe cumplir que exista una ruta de vértices tales que: $e_i \leq z_i \leq l_i$.

- Por último, también se debe garantizar que es posible recorrer la ruta que cumpla las condiciones anteriores sin que se acabe la batería del vehículo eléctrico. Para ello se define una nueva variable $y_i \in \mathbb{Z}^+$ que cuantifica el nivel de batería que tiene el vehículo eléctrico al visitar el vértice de la posición i de la ruta de respuesta. De manera que el nivel de batería entre dos vértices de la ruta debe estar en el rango de $[1, Q]$.

Para poder expresar las condiciones anteriores de manera formal se utilizan las siguientes definiciones:

- $Conexo\big(G(V,A)\big) = (\forall i, j \in V |\, i \neq j : \exists a \in A \,\big|\, a = (i,j)\big)$
- $ATiempo(ruta) = (\forall i \in ruta \mid i \in C : e_i \leq z_i \leq l_i)$
- $ConBateria(ruta) = (\forall i \in ruta \mid 1 \leq y_i \leq Q) \wedge y_0 = Q$

Finalmente, la precondición se expresaría como:

- $\hat{V}$ es el conjunto de todas las posibles rutas que se pueden realizar dentro del grafo. Aclarando que es posible revisitar vértices que hagan parte de S.

$$Q: \{Conexo\big(G(V,A)\big) \wedge (\exists\, ruta \in \hat{V} |\, ATiempo(ruta) \wedge ConBateria(ruta)) \}$$

Postcondición:

De manera similar, la postcondición define los criterios que la salida del problema debe cumplir para ser considerada una solución correcta. Además de satisfacer las precondiciones, se introduce el requisito de optimalidad, el cual exige que la ruta encontrada sea la de menor distancia posible. Para ello se utiliza la siguiente definición:

- $Distancia(ruta) = \sum_{i \in [1, ruta.length)} d_{ruta[i-1], ruta[i]}$

$$R: \{(Ruta[0] = Ruta[Ruta.length - 1]) \wedge ATiempo(Ruta) \wedge ConBateria(Ruta) \wedge \big(\forall\, ruta \in \hat{V} | Distancia(Ruta\big) \leq Distanica(ruta) \}$$

Casos de aplicación:

En el contexto de la transición energética, numerosas empresas han adoptado vehículos de transporte eléctricos para la distribución de sus productos. Por lo tanto, cualquier compañía que utilice vehículos eléctricos en sus operaciones logísticas de distribución de materiales se beneficiaría de la solución a este problema. La planificación de rutas para la distribución de productos es un problema frecuente para las empresas. Por lo que, determinar la ruta óptima en términos de distancia permite, a largo plazo, una reducción significativa de los costos operativos.

Algoritmos de solución:

Dos algoritmos para la solución del E-TSPTW se presentan a continuación. El primero es un algoritmo de tres etapas implementado por Roberti (2016) basado en el algoritmo para el TSPTW propuesto por da Silva y Urrutia (2010). El segundo algoritmo (Küçükoğlu,

Dewil & Cattrysse, 2019) emplea una técnica hibrida entra búsqueda tabú y recocido simulado.

El primer algoritmo se ejecuta en tres etapas descritas de la siguiente forma.

Etapa 0 – inicialización: satisfacción de restricción de ventanas de tiempo: esta parte establece las soluciones iniciales y los parámetros que controlan la exploración y convergencia del algoritmo. Se inicializan las mejores soluciones encontradas (x^* para TSPTW y y^* para E-TSPTW) en nil y se define $iterBest = 0$. Luego se genera un recorrido aleatorio (x) que puede no cumplir con las restricciones de tiempo y batería. A continuación, el algoritmo entra en el bucle principal donde se satisfacen las restricciones de ventana de tiempo y carga.

Etapa 1 – satisfacción de restricción de ventanas de tiempo: en esta primera etapa, el recorrido generado x se ajusta para satisfacer las restricciones de ventanas de tiempo mediante un algoritmo de Búsqueda de Entorno Variable (VND). La función objetivo minimiza la suma de las diferencias positivas entre el tiempo de llegada a cada cliente y el final de su ventana de servicio. Para tratar la infactibilidad, se aplican secuencialmente tres tipos de movimientos: reubicar clientes hacia atrás, reubicar clientes hacia adelante e intercambiar pares de clientes. Estos movimientos se repiten mientras sea posible disminuir la penalización por incumplimiento de las ventanas de tiempo o hasta lograr una solución completamente factible en este aspecto.

Etapa 2 – mejoramiento del costo total: esta etapa tomo el recorrido con las restricciones de tiempo satisfechas de la etapa anterior e intenta reducir la distancia total recorrida. De nuevo, se aplica una Búsqueda de Entorno Variable (VND), explorando iterativamente cuatro tipos de vecindarios: reubicar clientes hacia atrás, reubicar clientes hacia adelante, aplicar el operador *two-opt* e intercambiar clientes. En cada iteración, se ejecuta el mejor movimiento disponible que mantenga la factibilidad con la restricción de ventanas de tiempo. El proceso finaliza cuando no se pueden encontrar más mejoras.

Etapa 3 – obtención de una solución válida para la restricción de batería: por último, se emplea un algoritmo de Programación Dinámica para insertar paradas en estaciones de recarga y garantizar que el recorrido cumpla con las restricciones de batería. Se mantiene el orden de visita de los clientes y se aplican etiquetas para representar el estado del recorrido en cada punto, considerando el tiempo de llegada, el nivel de batería y la distancia acumulada. A partir de la solución del TSPTW obtenida en dos las fases anteriores, se expande iterativamente cada camino con dos opciones: continuar directamente al siguiente cliente o detenerse en una estación de recarga. Se selecciona la mejor combinación de paradas para minimizar el costo total del recorrido, asegurando que la batería permanezca dentro de los límites operativos en todo momento.

El segundo algoritmo de desarrolla de la siguiente manera.

Preprocesamiento: antes de comenzar la búsqueda de soluciones, el algoritmo realiza una limpieza del problema por lo cual se descartan aquellas rutas donde el tiempo de llegada a un cliente excede el límite permitido y se aseguran de que los caminos generados sean posibles en términos de consumo energético, descartando aquellos donde la batería no alcanza.

Inicialización: el algoritmo crea una solución inicial organizando los vértices a visitar de forma que se respeten los tiempos máximos permitidos, siguiendo un orden ascendente. Sin embargo, esta solución no siempre garantiza que el recorrido o tour sea posible de realizar o que sea óptimo. Para ajustarla, el algoritmo realiza 3 movimientos que permiten reorganizar la secuencia de visitas de 3 diferentes maneras:

1. Se toma un nodo a visitar y se coloca en una posición diferente de la ruta, evaluando si el cambio mejora el recorrido. (1-SHIFT) Ej. ; A → B → C → D → E pasa a D → A → B → C →E
2. Se elige una parte del trayecto y se invierte el orden de visita de varios nodos por visitar consecutivos para eliminar cruces innecesarios o caminos largos. (2-OPT) Ej.: A → B → C → D → E pasa a A → D → C → B → E
3. 3.Se intercambien las posiciones de dos nodos a visitar en la ruta, lo que puede resultar en una configuración más eficiente. (SWAP). A → B → C → D → E pasa a A → D → C → B → E

Estas operaciones se realizan varias veces hasta obtener una solución que respete las restricciones de las ventanas de tiempo. A partir de ahí, se generan rutas con puntos de recarga, insertando estaciones donde el vehículo lo necesite para asegurar que tenga suficiente batería en cada tramo.

Búsqueda Local y Parada: Una vez obtenida una solución inicial factible, el algoritmo busca mejoras seleccionando aleatoriamente uno de los 3 movimientos mencionados anteriormente y evalúa si el nuevo recorrido es mejor que el anterior. Si la nueva ruta es mejor, se actualiza la solución actual; de lo contrario, prueba con otro movimiento. Cuando una solución parece válida, el algoritmo verifica si el vehículo puede llegar a los destinos sin quedarse sin batería. Si es necesario, inserta puntos de recarga en la ruta, utilizando programación dinámica, Al principio el algoritmo está dispuesto a aceptar soluciones peores que las que ya encontró, pero cada vez se vuelve más estricto (similar a un Learning Rate que cada vez se reduce más) y solo acepta soluciones que realmente mejoren la óptima ruta actual. Por último, la condición de parada se da al terminar cierto número de iteraciones o no encuentre mejoras dado un número de iteraciones parecido a una exploración y explotación.

Referencias:

1. Roberti, R., & Wen, M. (2016). The Electric Traveling Salesman Problem with Time Windows. *Transportation Research Part E: Logistics and Transportation Review*, *89*. https://doi.org/10.1016/j.tre.2016.01.010

2. da Silva, R. F., & Urrutia, S. (2010). A General VNS heuristic for the traveling salesman problem with time windows. Discrete Optimization, 7(4), 203–211. https://doi.org/10.1016/j.disopt.2010.04.002
3. Küçükoğlu, İ., Dewil, R., & Cattrysse, D. (2019). Hybrid simulated annealing and tabu search method for the electric travelling salesman problem with time windows and mixed charging rates. Expert Systems with Applications, 134. https://doi.org/10.1016/j.eswa.2019.05.037