UNIVERSIDAD NACIONAL AUTÓNOMA DE MÉXICO

FACULTAD DE CIENCIAS

COMPLEJIDAD DEL PROBLEMA DE RUTAS DE VEHÍCULOS

T E S I S

QUE PARA OBTENER EL TÍTULO DE:

LICENCIADO EN CIENCIAS DE LA COMPUTACIÓN

P R E S E N T A :

DANIEL MICHEL TAVERA

DIRECTORA DE TESIS:
DRA. MARÍA DE LUZ GASCA SOTO

Ciudad Universitaria, Cd. Mx., 2017

1.Datos del alumno
Michel
Tavera
Daniel
Teléfono - 26 50 43 40
Universidad Nacional Autónoma de México
Facultad de Ciencias
Ciencias de la Computación
Número de cuenta - 412055971

2. Datos del tutor
Dra.
María de Luz
Gasca
Soto

3. Datos del sinodal 1
Dr.
Sergio
Rajsbaum
Gorodezky

4. Datos del sinodal 2
Dr.
Armando
Castañeda
Rojano

5. Datos del sinodal 3
Dr.
José de Jesús
Galaviz
Casas

6. Datos del sinodal 4
Dra.
Elisa
Viso
Gurovich

7.Datos del trabajo escrito.
Complejidad del Problema de Rutas de Vehículos
71 páginas
2017

Introducción

El Problema de Rutas de Vehículos o VRP (por sus siglas en inglés, *Vehicle Routing Problem*) es un problema conocido cuya importancia en aplicaciones prácticas, desde el abastecimiento de productos para tiendas hasta recolección de basura y transporte escolar, amerita estudiarlo a profundidad.

El problema consiste en encontrar la mejor manera de asignar rutas a los vehículos de una flota, de tal forma que atiendan las demandas de un conjunto de clientes dispersos alrededor de un almacén o depósito central, minimizando tanto el número de vehículos a utilizar como la distancia total recorrida por los mismos.

Existen muchas variantes del VRP, algunas más generales y otras con restricciones adicionales para facilitar su manejo. El propósito de este trabajo es analizar la complejidad computacional de algunas de las principales variantes del VRP, cuyas definiciones se presentan más adelante:

- El Problema de Rutas de Vehículos Clásico o CVRP (por sus siglas en inglés, *Classic Vehicle Routing Problem)*
- El Problema de Rutas de Vehículos con Ventanas de Tiempo o VRPTW (por sus siglas en inglés, *Vehicle Routing Problem with Time Windows)*
- El Problema de Rutas de Vehículos con Flota Heterogénea o HFVRP (por sus siglas en inglés, *Heterogeneous Fleet Vehicle Routing Problem)*
- El Problema de Rutas de Vehículos con Depósitos Múltiples o MDVRP (por sus siglas en inglés, *Multiple Depot Vehicle Routing Problem)*

El resto de este trabajo está organizado de la siguiente manera:

En el primer capítulo se presentan antecedentes y definiciones de conceptos sobre complejidad que serán utilizados durante el desarrollo del trabajo.

En el segundo capítulo se describe el problema de rutas de vehículos, se determinan las variantes del mismo que serán abordadas y se define formalmente la primera de ellas, el CVRP, así como un problema adicional utilizado como apoyo durante el análisis de complejidad del CVRP.

En el tercer capítulo se demuestra que el CVRP está en NP con dos enfoques distintos.

En el cuarto capítulo se demuestra que el CVRP es NP-difícil, lo que aunado al resultado del capítulo anterior lleva a la conclusión de que el CVRP es un problema NP-completo.

En el quinto capítulo se utiliza el resultado obtenido en los capítulos anteriores como base para realizar el análisis de complejidad de tres variantes del problema de rutas de vehículos.

Finalmente, se presentan las conclusiones del trabajo y algunas implicaciones de los resultados obtenidos.

Adicionalmente, se tienen tres apéndices en los que se presentan las demostraciones de tres proposiciones: dos empleadas para facilitar el análisis de complejidad del CVRP en los capítulos tercero y cuarto y una utilizada para facilitar el análisis de complejidad del VRPTW en el quinto capítulo.

Índice general

Capítulo 1

Antecedentes y definiciones

A continuación se presentan breves definiciones de algunos conceptos utilizados en este trabajo

1.1. Problema de decisión

Un problema de decisión es aquel cuya solución consiste de un **sí** o un **no**; es decir, el problema puede plantearse como una pregunta cuya respuesta es afirmativa o negativa según el ejemplar.

Más formalmente, un problema de decisión consiste en determinar si el ejemplar satisface o no cierta propiedad dada.

Es posible construir versiones de decisión de un problema de optimización, en las que se busque una solución que satisfaga una cota determinada, en lugar de buscar la solución óptima.

En otras palabras, cualquier problema de optimización puede transformarse fácilmente en uno de decisión con tan solo establecer una cota C y hacer la pregunta:

$$\text{¿}\exists\ S \text{ solución del ejemplar, con valor menor o igual que } C?$$

para problemas de minimización, o

$$\text{¿}\exists\ S \text{ solución del ejemplar, con valor mayor o igual que } C?$$

para problemas de maximización.

La versión de decisión del problema consiste en determinar si la respuesta a dicha pregunta es **sí** o **no**.

La versión de optimización del problema siempre será al menos tan difícil de tratar como su versión de decisión, ya que cualquier algoritmo que resuelva la versión de optimización puede solucionar la versión de decisión, pero un algoritmo para la versión de decisión no necesariamente puede resolver la versión de optimización por sí solo.

1.2. Reducción

Una reducción es una forma de encontrar la solución de un problema, para el cual puede o no conocerse un algoritmo que lo resuelva, utilizando un método conocido para resolver otro problema; es decir, es una forma de construir un camino alternativo para llegar a la solución de un problema.

Formalmente, una reducción de un problema π en otro problema γ es un par de algoritmos A y B tales que:

- dado un ejemplar E de π, A construye un ejemplar E' de γ; y
- dada S' la solución para E' de γ, B construye la solución S para E de π.

Utilizando estos algoritmos, puede solucionarse un ejemplar E de π empleando un algoritmo diseñado para solucionar ejemplares de γ como paso intermedio, como se muestra en la Figura 1:

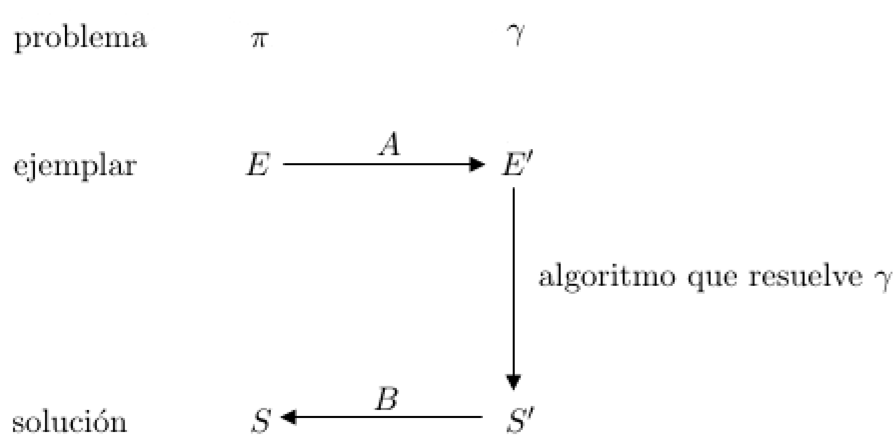

Figura 1: Reducción de π en γ

Si π y γ son problemas de decisión, basta que el algoritmo A construya el ejemplar E' de forma que la respuesta o solución S' sea afirmativa si y sólo si la solución S es afirmativa.

Se dice que una reducción es polinomial si tanto A como B tienen tiempos de ejecución acotados por un polinomio en n, el tamaño del ejemplar E.

Si existe una reducción de π en γ, se dice que π es reducible en γ.

Es importante mencionar que esta relación es transitiva; es decir, dados tres problemas π, γ y α, si π es reducible en γ, y γ es reducible en α, entonces π es reducible en α:

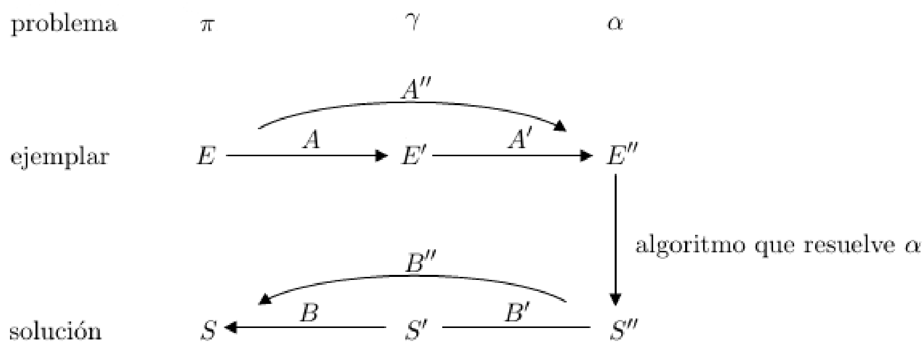

Figura 2: Reducción transitiva de π en α

Sea A el algoritmo para obtener el ejemplar E' de γ a partir del ejemplar E de π y sea A' el algoritmo para obtener el ejemplar E'' de α a partir de E', entonces el algoritmo A'' para obtener E'' a partir de E se conforma de la composición de A y A'; es decir, A'' consiste en aplicar A' al resultado de la aplicación de A.

El caso de los algoritmos para las soluciones es análogo.

1.3. Algoritmo no-determinista

Un algoritmo no-determinista es, en términos generales, un proceso que recibe como parte de su entrada un ejemplar de un problema de decisión y que, mediante el uso de no-determinismo y de un proceso de verificación, produce como salida un **sí** o un **no**.

El proceso de verificación debe garantizar que solamente se producirá como salida un sí si la respuesta del ejemplar recibido es sí, sin embargo, no se tiene una garantía similar para una respuesta negativa.

Esto quiere decir que si la salida del algoritmo no-determinista es sí, la respuesta del ejemplar recibido debe ser sí, pero si la salida del algoritmo no-determinista es un no, no es posible concluir nada respecto al ejemplar recibido.

Existen varias maneras de ver los algoritmos no-deterministas, que difieren en los detalles sobre la entrada y el funcionamiento del algoritmo, en particular sobre la forma en que éste utiliza el no-determinismo. En este trabajo se manejan dos de ellas, una utiliza un certificado y la otra emplea la primitiva ND-Choice.

Certificado

Dado un ejemplar E de un problema de decisión cuya respuesta es sí, un certificado C para E es un objeto sucinto que constata o certifica que la respuesta de E es afirmativa [11], de manera que dados E y C, es posible verificar C para validar que la respuesta de E es sí, sin tener la necesidad de resolver E.

En este contexto, sucinto significa que C no puede ser mucho más grande que E; es decir, si E tiene tamaño n, el tamaño de C está acotado superiormente por un polinomio en n.

Como el certificado constata que la respuesta del ejemplar es afirmativa, no puede existir un certificado válido para un ejemplar cuya respuesta sea negativa.

La forma del certificado es conocida de antemano, por lo que es posible construir un algoritmo que verifique el certificado para confirmar su validez.

En muchas ocasiones, la forma del certificado es muy similar a la forma de una solución del problema, por lo que el certificado suele definirse informalmente como una propuesta de solución a ser verificada [8].

En este trabajo, todos los certificados tendrán la forma de una solución.

Algoritmo no-determinista que utiliza un certificado

La primer forma de ver los algoritmos no-deterministas empleada en este trabajo es utilizando certificados [4] y consiste en lo siguiente:

Un algoritmo no-determinista es entendido como un proceso A tal que, dados un ejemplar E de un problema de decisión π y un certificado C para E, A determina la validez de C; es decir, A procesa C para verificar si en efecto constata que la respuesta de E es afirmativa.

Si el certificado es válido, la respuesta de E debe ser afirmativa y el algoritmo no-determinista regresa un **sí**; en cualquier otro caso la salida del algoritmo no-determinista será un **no**.

No se especifica la manera de obtener el certificado C; puede ser proporcionado por el usuario del algoritmo no-determinista, generado de forma pseudo aleatoria o proceder de alguna otra fuente.

Si A puede realizar tal verificación de C en un tiempo acotado por un polinomio del tamaño de E, entonces se dice que A es un algoritmo no-determinista polinomial.

Primitiva ND-Choice

La primitiva ND-Choice [9] es una operación que utiliza no-determinismo para hacer una elección entre varias opciones durante un proceso, ya sea pidiendo una entrada al usuario, usando el azar o de alguna otra forma.

No es necesario establecer la forma concreta en que opera, sino que solamente se asume que puede hacer la elección en tiempo constante.

Esta primitiva se emplea para construir una propuesta de solución que luego es verificada, por lo que puede verse al proceso de construcción de la solución propuesta como una especificación de la forma en que se obtiene un certificado, pero con la ventaja de que detallar la construcción concede mayor control sobre la forma de la propuesta de solución, permitiendo garantizar que se satisfaga una o varias características o propiedades, lo que facilita su verificación.

Sin embargo, debe cuidarse que la propuesta de solución construida siga siendo suficientemente general, ya que de lo contrario se estaría limitando a una fracción del espacio de soluciones del problema, por lo que el algoritmo solamente sería válido para una versión del problema con restricciones adicionales.

Algoritmo no-determinista que utiliza la primitiva ND-Choice

La segunda forma de ver los algoritmos no-deterministas utilizada en este trabajo, trabajando con la primitiva ND-Choice, consiste en lo siguiente:

Un algoritmo no-determinista es visto como un proceso que recibe un ejemplar E de un problema de decisión π y que consta de dos partes:

- Una parte no-determinista que genera una propuesta de solución para E mediante el uso de la primitiva ND-Choice
- Una parte de verificación que determina si la propuesta generada es en realidad una solución de E.

Si la propuesta de solución generada resulta ser una solución de E con respuesta afirmativa, significa que existe al menos una solución para E, por lo que el proceso regresa un **sí**; en cualquier otro caso, la salida del proceso es un **no** y no es posible concluir nada sobre E.

Visto de esta forma, se considera que el algoritmo no-determinista es polinomial si cada parte, tanto la no-determinista como la de verificación, requiere tiempo acotado por un polinomio del tamaño de E.

1.4. Clasificación de problemas

En esta sección se definen las clases de problemas NP, NP-difícil, y NP-completo.

NP
Se dice que un problema π está en la clase de problemas NP si existe una algoritmo no-determinista polinomial para π.

Por tanto, para probar que π está en la clase NP basta mostrar un algoritmo no-determinista polinomial para π. En este trabajo siempre se presentan dos de ellos, uno desde cada enfoque descrito en la sección anterior.

NP-difícil
Un problema π está en la clase de problemas NP-difíciles si para todo problema γ en NP, existe una reducción polinomial de γ en π. Si un problema π está en la clase de problemas NP-difíciles, se dice que π es NP-Difícil.

Una forma alternativa de mostrar que π es NP-difícil consiste en tomar otro problema α, que se sabe que es NP-difícil, y dar una reducción polinomial de α en π. De esta manera, aplicando la transitividad de las reducciones, se establece un método para construir la reducción de γ en π para cualquier γ en NP, ya que como α es NP-difícil, existe una reducción polinomial de γ en α, por lo que basta reducir γ en α y al ejemplar resultante aplicarle la reducción de α en π.

NP-Completo
Un problema π está en la clase de problemas NP-Completos si se cumplen dos condiciones:

- π es NP-difícil, y
- π está en NP.

Si un problema π está en la clase de problemas NP-Completos, se dice que π es NP-Completo.

Por lo tanto, para demostrar que π es NP-Completo, suele demostrarse cada una de las propiedades independientemente.

Para realizar dicha prueba, un camino común a seguir es:

- mostrar que π está en NP presentando un algoritmo no-determinista polinomial para π y, por separado
- mostrar que π es NP-difícil mediante una reducción polinomial de algún otro problema γ, que se sabe que es NP-Difícil, a π.

1.5. Definición formal de un problema

Para fines de este trabajo, la definición formal de un problema de optimización π significa la especificación de la forma y estructura de un ejemplar de π y de una solución para dicho ejemplar, así como la manera de determinar una solución óptima y de ver a π como un problema de decisión.

Capítulo 2

Marco teórico

A continuación se definen el problema principal a analizar, algunas de sus variantes y otro problema que será de utilidad para el análisis.

2.1. VRP y variantes

El problema de Rutas de Vehículos o VRP (por sus siglas en inglés, *Vehicle Routing Problem*) consiste en encontrar la mejor manera de asignar rutas a los vehículos de una flota, de tal forma que atiendan las demandas de un conjunto de clientes dispersos alrededor de un almacén o depósito central.

Se conocen de antemano las demandas de cada cliente y las distancias entre cada par de clientes, así como la distancia entre cada cliente y el depósito.

Los vehículos tienen una capacidad de carga fija y cada uno se utiliza en una sola ruta.

Cada ruta debe comenzar en el depósito, visitar a los clientes que vaya a servir y regresar al depósito. La demanda total de los clientes servidos por la ruta no puede exceder la capacidad de carga del vehículo.

Se busca un conjunto de rutas que satisfaga las demandas de todos los clientes, recorriendo la menor distancia total posible y utilizando el mínimo número de vehículos.

Estableciendo variables o restricciones adicionales, es posible definir una serie de variantes del problema, algunas de las cuales facilitan el estudio del mismo al enfocarse en algún caso particular, mientras que otras son generalizaciones que tienen una visión más amplia del problema pero pueden ser más difíciles de abordar.

De Jaegere et al. [5] publicaron un estudio sobre 144 artículos que tratan el VRP, en el cual entre otros análisis clasifican los artículos según la variante abordada del problema.

A continuación se listan algunas de las variantes del VRP más importantes según el estudio de De Jaegere et al. [5]:

- VRP con ventanas de tiempo o VRPTW (por sus siglas en inglés, *Vehicle Routing Problem with Time Windows*):

  Cada cliente tiene, además de su demanda, un intervalo de tiempo en el que debe ser atendido.

  La distancia entre cada par de clientes implica, además del costo de traslado que se busca minimizar, un tiempo necesario para ir de un cliente a otro; y se requiere cierto tiempo para atender al cliente una vez que el vehículo llega a él.

  En algunas interpretaciones del problema, el depósito también tiene una ventana de tiempo, en la que deben contenerse todas las operaciones de cada ruta; es decir, no se permite que los vehículos salgan del depósito ni lleguen a él fuera la ventana del mismo.

  En el apéndice C se muestra que las versiones del problema con y sin ventana de tiempo para el depósito son equivalentes.

  Finalmente, las restricciones de tiempo pueden ser suaves, si atender al cliente fuera de tiempo conlleva un costo o penalización; o duras, si no es posible atender al cliente fuera del intervalo de tiempo requerido.

- VRP con flota heterogénea o HFVRP (por sus siglas en inglés, *Heterogeneous Fleet Vehicle Routing Problem*):

  La capacidad de carga, así como el costo asociado a la distancia, puede variar de un vehículo a otro, por lo que debe considerarse qué vehículo es más conveniente utilizar en cada ruta.

  En contraste, en una flota homogénea todos los vehículos tienen la misma capacidad y el costo para recorrer la misma distancia es igual para todos los vehículos.

- VRP con depósitos múltiples o MDVRP (por sus siglas en inglés, *Multiple Depot Vehicle Routing Problem*):

  Existen varios depósitos en diferentes puntos, por lo que la distancia entre un cliente y cada depósito varía.

  Según el caso, las rutas pueden estar ligadas cada una a un depósito específico, o empezar y terminar en cualquier depósito; en ocasiones, con la restricción de que a cada depósito debe llegar el mismo número de vehículos que partieron de él inicialmente.

También existen variantes mixtas, como el HFVRPTW, en el que se tienen tanto intervalos de tiempo para visitar a cada cliente como una flota heterogénea con diferentes capacidades y costos para cada vehículo.

Sin embargo, en los resultados del estudio de De Jaegere et al. se destaca por su relevancia el VRP clásico, también llamado CVRP (por sus siglas en inglés, *Capacitated Vehicle Routing Problem*).

Cuando se habla del VRP, usualmente se refiere al CVRP, pero vale la pena hacer la distinción, ya que VRP también puede referirse al conjunto de problemas con todas sus variantes.

En este trabajo el enfoque se hace sobre el CVRP.

2.2. Descripción general del Problema de Rutas de Vehículos Clásico (CVRP)

Clarke y Wright [3] describen el problema de la siguiente forma:

> "...cómo atender a un número de clientes, dispersos geográficamente alrededor de un almacén central, utilizando una flota de vehículos..."

El problema consiste en encontrar un conjunto de rutas donde se satisfagan las demandas de todos los clientes, a la vez que se minimizan tanto la distancia total recorrida, como el número de vehículos a utilizar.

Las características del CVRP son:

- Se considera una flota de vehículos homogénea; es decir, todos los vehículos tienen las mismas características, en particular la misma capacidad de carga.
- Cada vehículo se utiliza en una sola ruta y la demanda total de todos los clientes visitados en la ruta no puede exceder dicha capacidad de carga.
- Cada cliente tiene una demanda menor o igual a la capacidad de un vehículo y es visitado exactamente una vez.
- Existe un único almacén central, cada ruta comienza y termina en él.

2.3. Ejemplo del CVRP

Supóngase que los vehículos tienen capacidad 5 y se desea atender los clientes A, B, C y D, con demandas y distancias como se muestra en la Figura 3, done los números dentro de los círculos representan la demanda de cada cliente, y los números sobre las líneas representan la distancia entre los clientes a los extremos de la línea.

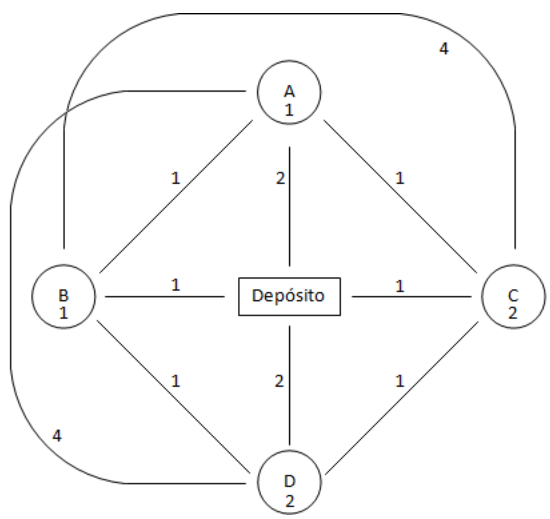

Figura 3: Ejemplar del CVRP

La demanda total es 6, por lo que un vehículo no bastará para atender a todos los clientes. Deben utilizarse al menos dos vehículos.

Una posible solución sería enviar un vehículo a satisfacer la mayor demanda posible, y luego un segundo vehículo a atender a los clientes que aún falten. En este caso, las dos rutas a utilizar serían:

- depósito $\rightarrow C \rightarrow D \rightarrow B \rightarrow$ depósito, atendiendo en total una demanda de 5, y recorriendo una distancia de 4; y
- depósito $\rightarrow A \rightarrow$ depósito, atendiendo en total una demanda de 1, y recorriendo una distancia de 4.

De esta forma, se atiende a todos los clientes utilizando 2 vehículos y se recorre una distancia total de 8.

Una mejor solución para este ejemplo es emplear las rutas

- depósito $\rightarrow C \rightarrow D \rightarrow$ depósito, atendiendo en total una demanda de 4, y recorriendo una distancia de 3; y
- depósito $\rightarrow A \rightarrow B \rightarrow$ depósito, atendiendo en total una demanda de 2, y recorriendo una distancia de 3.

De esta forma, se atiende a la totalidad de los clientes utilizando 2 vehículos y se recorre una distancia total de 6.

2.4. Definición formal del CVRP

En esta sección se presenta la definición formal del Problema de Rutas de Vehículos Clásico (CVRP) mediante la especificación detallada de la forma y características de un ejemplar del problema y de una solución para el ejemplar, así como las propiedades que caracterizan a un óptimo entre las soluciones.

Ejemplar:

- Un conjunto $C = \{c_0, c_1, c_2, ..., c_n\}$, de n clientes y un depósito c_0, con n natural
- Una constante k que representa la capacidad de los vehículos
- Una demanda o peso w_i, para cada $i = 0, 1, ..., n$, con $w_0 = 0$, y $0 < w_j \leq k$, para $j = 1, 2, ..., n$
- Distancias reales positivas d_{ij} para cada par (c_i, c_j), con $i \neq j$

Solución:

Un conjunto de m rutas $R = \{r_1, r_2, ..., r_m\}$, donde cada r_i es una lista ordenada de clientes $r_i = [c_0, c_{x_1}, c_{x_2}, ..., c_{x_{q_i}}, c_0]$, tal que

$$\sum_{c_j \in r_i} w_j \leq k.$$

Además, para cada r_i se tiene una longitud total denominada l_i, donde $l_i = d_{0x_1} + d_{x_1x_2} + ... + d_{x_{q_i}0}$, para $i = 1, 2, ..., m$.

Finalmente, debe cumplirse que cada cliente sea visitado por exactamente una ruta; esto es, $\forall c_i \in C,\ \exists! r_j \in R$ tal que $c_i \in r_j$.

Óptimo:

Si llamamos m al número de vehículos a utilizar y $L(R)$ a la longitud o costo total:

$$L(R) = \sum_{i=1}^{m} l_i$$

entonces el óptimo es la solución R con m y $L(R)$ mínimos.

Para facilitar la prueba, se utilizará una versión del problema con m fija. En el Apéndice A se muestra que el problema con esta modificación es equivalente al problema sin modificar.

De esta forma, el problema de decisión queda como sigue:
Dados un ejemplar del CVRP y un real positivo T, determinar si existe o no una solución con costo total menor o igual a T; es decir, determinar la respuesta a la pregunta

$$¿\exists R \text{ tal que } L(R) \leq T?$$

En el resto de este trabajo se analiza la complejidad del CVRP y de algunas otras variantes del VRP.

Primero, se presenta la definición del TSP, un problema NP-Difícil bien conocido.

En seguida, se proponen dos algoritmos no-deterministas polinomiales para el CVRP, uno que utiliza un certificado y otro que emplea la primitiva ND-Choice, lo que muestra que el problema está en NP.

Luego, se muestra que el CVRP es NP-difícil, dando una reducción polinomial del TSP al CVRP.

Después, teniendo que el CVRP está en NP y es NP-difícil, se concluye que es un problema NP-completo.

Finalmente, se utiliza este resultado para mostrar que otras variantes del VRP también pertenecen a la categoría de problemas NP-completos y se mencionan algunas implicaciones de estos resultados.

2.5. Descripción general del Problema del Agente Viajero (TSP)

El problema a utilizar en la reducción es el problema del agente viajero o TSP (por sus siglas en inglés, *Travelling Salesman Problem*), el cual consiste en lo siguiente:

Dado un conjunto de ciudades, con distancias conocidas entre ellas, se busca un ciclo que, iniciando en cualquier ciudad, pase por cada ciudad exactamente una vez y luego regrese a la ciudad inicial, recorriendo la menor distancia total posible.

Se sabe que el TSP es NP-completo [4], y en particular NP-difícil, por lo que al proporcionar una reducción polinomial al CVRP, se tendrá que este último es NP-difícil.

2.6. Ejemplo del TSP

Supóngase que se busca un ciclo que pase una vez por cada una de la ciudades A, B, C, D, y E, con distancias como se muestra en la Figura 4, donde el número sobre la línea que une dos ciudades representa la distancia entre las mismas.

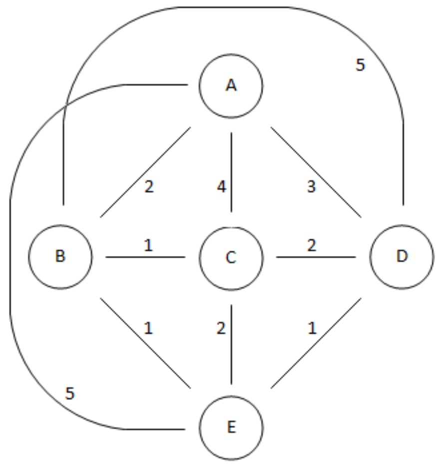

Figura 4: Ejemplar del TSP

Existen muchos ciclos posibles con distancia total 10; entre ellos el único ciclo que utiliza todas las rutas con distancia 1:

$$A \to C \to B \to E \to D \to A.$$

Pero existen mejores soluciones, como el ciclo con distancia total de 9

$$A \to B \to C \to E \to D \to A.$$

2.7. Definición formal del TSP

A continuación se define formalmente el Problema del Agente Viajero (TSP) mediante la especificación detallada de la forma y características de un ejemplar, de una solución y del óptimo.

Ejemplar:

Un conjunto $C = \{c_1, c_2, ..., c_n\}$ de n ciudades, con n natural, y distancias reales positivas d_{ij} para cada par (c_i, c_j), con $i \neq j$.

Solución:

Un ciclo $r = [c_{x_1}, c_{x_2}, ..., c_{x_n}, c_{x_1}]$, que pasa por todas las ciudades, con distancia total recorrida $l(r) = d_{x_1x_2} + d_{x_2x_3} + ... + d_{x_nx_1}$.

Óptimo:

r con $l(r)$ mínima.

Para facilitar la prueba, se utilizará una versión del problema con c_{x_1} fija; es decir, una versión en la que el ciclo debe iniciar y terminar en una ciudad específica. En el Apéndice B se muestra que el problema con esta modificación es equivalente al problema sin modificar.

El TSP como problema de decisión queda como sigue:

Dados un ejemplar del TSP y un real positivo T , determinar si existe o no una solución con distancia total recorrida menor o igual a T; es decir, determinar la respuesta a la pregunta

$$¿\exists r \text{ tal que } l(r) \leq T?$$

Contando con estas definiciones es posible proseguir con el análisis de complejidad del CVRP.

Capítulo 3

El CVRP está en NP

En este capítulo se demuestra que el CVRP está en la clase NP.

Se presentan dos algoritmos no-determinisas polinomiales para el problema, uno desde cada punto de vista mencionado en la definición de algoritmo no-determinista.

3.1. Algoritmo no-determinista de verificación para un certificado

Sea E un ejemplar del CVRP con C su correspondiente conjunto de clientes y una flota de tamaño fijo con m vehículos, cada uno de capacidad k.

Sea, además, R un certificado para E con la forma $R = \{r_1, r_2, ..., r_m\}$. Se busca verificar si R es una solución válida para E; es decir, verificar si R satisface las condiciones enunciadas al definir una solución del CVRP y, además, $L(R) \leq T$, con T una cota dada.

La verificación se realiza de la siguiente forma:

Se asignan etiquetas enteras a los clientes para saber si fueron visitados.
Inicialmente, $v_i = 0$, para cada $c_i \in C$.

Se llevan también contadores para las rutas de la solución propuesta que registren la demanda total de los clientes visitados en la ruta.
Inicialmente, $h_i = 0$ para cada $r_i \in R$.

También se inicializan en cero las distancias l_i de cada ruta r_i y la distancia total de la solución $L(R)$.

La idea es verificar que las rutas de R sean correctas, entre todas visiten a todos los clientes y en total no excedan la cota de distancia T.

La estructura general del algoritmo es la siguiente:

Se recorre cada ruta r_i de R, calculando h_i y l_i, marcando cada cliente visitado y verificando en cada paso si la solución sigue siendo válida o se ha violado alguna de las condiciones establecidas.

A la vez que se van recorriendo las rutas, se calcula también $L(R)$ y se verifica que no exceda el valor de la cota T.

Si se terminan de recorrer todas las rutas y no se ha violado ninguna condición sobre ellas, se verifica que todos los clientes hayan sido visitados.

Si todas las rutas son válidas, todos los clientes fueron visitados y se sigue cumpliendo que $L(R) \leq T$, entonces R es una solución válida para E.

A continuación se muestra con mayor detalle el proceso de verificación:

Inicializar en cero los valores de las etiquetas v_i para cada $c_i \in C$, y h_i para cada $r_i \in R$, así como las distancias $L(R)$ y l_i para cada $r_i \in R$.

Recorrer cada ruta $r_i \in R$ y realizar lo siguiente para cada uno de los elementos de r_i:

> Sea x el elemento que se está verificando, x es un cliente c_j.
>
> Si x es el primer elemento de r_i:
>
> > Si $j \neq 0$; es decir, si x es el primer elemento de la ruta y no es el depósito c_0, la ruta no comienza en el depósito, por lo que R no es una solución válida y la verificación termina con respuesta negativa.
> >
> > Si, por el contrario, $j = 0$, la ruta comienza en el depósito, por lo que se termina la verificación de x y se procede a verificar el siguiente elemento de r_i.
>
> Sea c_z el elemento anterior a x en la ruta.
>
> Sumar la distancia entre c_z y x, d_{zj}, a l_i y a $L(R)$.
>
> Si $L(R) > T$, se ha sobrepasado la cota de la distancia total, por lo que R no es una solución válida y la verificación termina con respuesta negativa.
>
> Si x es el último elemento de r_i y no es el depósito c_0, la ruta no termina en el depósito, por lo que R no es una solución válida y la verificación termina con respuesta negativa.
>
> Si x no es el primer ni el último elemento de la ruta r_i:

Si $j = 0$; es decir, si x es el depósito c_0, la ruta regresa al depósito más de una vez , por lo que R no es una solución válida y la verificación termina con respuesta negativa.

Sumar 1 a v_j.

Si $v_j > 1$, se ha visitado al cliente j más de una vez, por lo que R no es una solución válida y la verificación termina con respuesta negativa.

Sumar la demanda de x, w_j, a h_i.

Si $h_i > k$, se ha excedido la capacidad de carga del vehículo, por lo que R no es una solución válida y la verificación termina con respuesta negativa.

Si se recorren todas las rutas sin obtener una respuesta negativa, se procede a verificar que todos los clientes hayan sido visitados; esto es, que $v_i = 1, \forall c_i \in C, i \neq 0$.

Para esto, basta con recorrer C y revisar la etiqueta de cada cliente. Si alguna etiqueta es cero, ese cliente no fue visitado, por lo que R no es una solución válida y la verificación termina con respuesta negativa.

Si en este punto sigue sin tenerse respuesta negativa, esto quiere decir que todas las rutas son válidas y en su conjunto visitan a todos los clientes exactamente una vez sin exceder la cota de distancia recorrida T. Por lo tanto, R es una solución válida de E y la verificación termina con respuesta afirmativa.

Procedimiento 1: Verificación de un certificado para el CVRP.

Si una ruta tiene más elementos que el número de clientes, tendrá que visitar más de una vez a algún cliente, lo que provocaría una terminación con respuesta negativa en el paso 3.2, o bien pasar por el depósito en algún momento además del inicio y el fin de la ruta, lo que provocaría una terminación con respuesta negativa en el paso 3.1.

De este modo, en el peor caso la verificación recorre todos los elementos de cada ruta, efectuando por cada uno a lo más cinco pasos, cada uno de los cuales realiza un número constante de operaciones; y después recorre todos los clientes, realizando una sola comparación por cada uno. Así, si el número de clientes $|C|$ es n, como se tienen m rutas, la verificación realiza a lo más un número de operaciones $O(m + n)$ en el peor caso; es decir, la verificación tiene complejidad acotada por un polinomio del tamaño de E.

Por lo tanto, la verificación es polinomial y el CVRP está en la clase NP.

■

3.2. Algoritmo no-determinista que utiliza la primitiva ND-Choice

Sea E un ejemplar del CVRP con C su correspondiente conjunto de n clientes y una flota de tamaño fijo con m vehículos de capacidad k.

Primero, en la parte no determinista, se construirá una solución colocando cada cliente en una ruta aleatoria, de forma que se tengan m rutas, que entre todas visiten a cada cliente exactamente una vez.

Luego, en la parte de verificación, se revisará cada ruta para constatar que no se sobrepase la capacidad de carga k del vehículo, a la vez que se calcula la distancia total recorrida para compararla con la cota T dada.

La parte no determinista trabaja de la siguiente forma:

> Crear un conjunto de m rutas, $R = \{r_1, r_2, ..., r_m\}$.
>
> Inicializar las m rutas, de forma que cada una consista únicamente del depósito c_0, que es el punto de partida para cualquier ruta válida.
>
> Hacer una copia C' del conjunto de clientes C, y aplicar n veces la primitiva ND-Choice, que en este caso realiza lo siguiente:
>
> > Tomar un cliente x de C', elegido de forma aleatoria para evitar que el orden en el que están listados los clientes en C tenga influencia sobre el resultado.
> >
> > Elegir una ruta aleatoria e insertar x al final de la misma.
> >
> > Eliminar x de C'.
>
> Finalmente, agregar nuevamente el depósito c_0 al final de cada ruta.

Procedimiento 2: Algoritmo no determinista para el CVRP.

Este proceso es polinomial, ya que realiza un número constante de operaciones para cada uno de los n clientes, y otro número constante de operaciones para cada una de las m rutas, por lo que en total requiere un número de operaciones $O(n + m)$.

La parte de verificación opera como sigue:

Asignar como cero el valor inicial de las distancias $L(R)$ y l_i, $\forall i = 1, 2, ..., m$.

Para cada $r_i \in R$, se realizar lo siguiente:

> Sea q el número de clientes visitados en la ruta, de modo que se tiene $r_i = [x_0, x_1, x_2, ..., x_{q-1}, x_q]$, con $x_0 = c_0$ y $x_q = c_0$.
>
> Inicializar en cero un contador h_i de demanda cubierta por la ruta.
>
> Recorrer los elementos x_j de r_i, desde $j = 1$ hasta $j = q$, y para cada uno:
>
> > Sumar a h_i la demanda del cliente actual.
> >
> > Si $h_i > k$, se ha excedido la capacidad del vehículo, por lo que R no es una solución válida y el proceso termina con respuesta negativa.
> >
> > Sean $c_y = x_{j-1}$ y $c_z = x_j$ los clientes anterior y actual, respectivamente.
> >
> > Sumar la distancia entre el cliente anterior y el actual, d_{yz}, a l_i y a $L(R)$.
> >
> > Si $L(R) > T$, se ha excedido la cota sobre la distancia total recorrida, por lo que la solución R no satisface las condiciones del problema y el proceso termina con respuesta negativa.
>
> Si no se obtiene respuesta negativa para ninguno de los elementos x_j, significa que la ruta r_i es válida y se prosigue a verificar la siguiente ruta.

Si se verifican todas las rutas sin obtener respuesta negativa, entonces la solución R es correcta y satisface las condiciones del problema, por lo que el proceso termina con respuesta afirmativa.

Procedimiento 3: Algoritmo de verificación para el CVRP.

Esta verificación es polinomial, ya que solamente realiza un número constante de operaciones por cada elemento de cada ruta, y por construcción se garantiza que en su conjunto las rutas solamente visitan a cada cliente una vez, y al depósito dos veces por ruta, por lo que en total se requiere de un número de operaciones $O(n + m)$.

Por lo tanto, al ser tanto la parte no-determinista como la parte de verificación polinomiales, el algoritmo es polinomial y el CVRP está en la clase NP.

■

Capítulo 4

El CVRP es NP-difícil

A continuación se muestra que el CVRP es NP-difícil mediante la siguiente reducción de TSP a CVRP:

Sea E un ejemplar del TSP, con $C = \{c_1, c_2, ..., c_n\}$ un conjunto de n ciudades, y d_{ij} la distancia entre c_i y c_j, como se describió en la definición del TSP.

Supóngase, sin pérdida de generalidad, que se busca una solución para E cuyo ciclo empiece y termine en c_n.

Se construye un ejemplar E' del CVRP como sigue:

Sea C' el conjunto de clientes, con el depósito siendo c_n, y las $n-1$ ciudades restantes los clientes.

Utilizar para C' las mismas distancias entre los elementos definidas para C.

Asignar demanda w_i unitaria a cada cliente $c_i \in C'$; es decir:

$$w_i = 1, \ \forall i = 1, 2, ..., (n-1).$$

Establecer la capacidad de carga del vehículo igual al número de clientes, esto es:

$$k = n - 1.$$

Finalmente, utilizar un solo vehículo; es decir, fijar:

$$m = 1.$$

Procedimiento 4: Reducción del TSP al CVRP.

Esta construcción es $O(n)$, con n el número de ciudades del ejemplar E, ya que solamente requiere asignar $n+1$ valores: $n-1$ demandas, 1 capacidad de carga y 1 número de vehículos; es decir, la reducción es polinomial.

A continuación se muestra que existe una solución para el ejemplar E con distancia total recorrida menor o igual a T, si y solamente si existe una solución para el ejemplar E' con costo total menor o igual a la misma T:

Primero supóngase que existe una solución para E con distancia total menor o igual a T; es decir,

$$\exists r \text{ tal que } l(r) \leq T, \text{ con } r \text{ iniciando y terminando en } c_n.$$

Entonces podemos trazar una ruta r_1 en C', que siga los mismos pasos que r en C; es decir, r_1 inicia y termina en el depósito c_n, y pasa por todos los otros elementos de C' exactamente una vez.

De esta forma se tiene que $\forall c_i \in C', c_i \in r_1$; además, como $w_n = 0$, por ser c_n el depósito, y $w_i = 1, \ \forall i = 1, 2, ..., (n-1)$; se tiene que

$$\sum_{c_j \in r_1} w_j = \sum_{j=1}^{n-1} w_j = \sum_{j=1}^{n-1} 1 = n - 1 = k.$$

Por tanto $R = \{r_1\}$ es una solución de E', el ejemplar del TSP.

Por otro lado, como las distancias en C' son las mismas que en C, la longitud total l_1 de r_1 será igual a $l(r)$, y como $m = 1$, se tiene que

$$L(R) = \sum_{i=1}^{m} l_i = l_1 = l(r).$$

Por lo tanto, $\exists R$ tal que $L(R) \leq T$; es decir, existe una solución para E' con costo total menor o igual a T.

$\square$

Supóngase ahora que existe una solución para E' con costo total menor o igual a T; esto es,

$$\exists R \text{ tal que } L(R) \leq T.$$

Dado que $m = 1$, R debe consistir de una sola ruta r_1 con costo $l_1 = L(R)$.
Entonces, como r_1 comienza y termina en el depósito c_n, podemos construir un ciclo r en C siguiendo los mismos pasos que r_1 en C'; es decir, r inicia y termina en c_n, además, pasa exactamente una vez por cada uno de los otros elementos de C.

Por lo tanto, r es una solución de E.

Como las distancias en C y C' son las mismas, se tiene que $l(r) = l_1 = L(R)$.

Por lo tanto, $\exists r$ tal que $l(r) \leq T$; es decir, existe una solución para E con costo total menor o igual a T.

$\square$

De esta forma, podemos concluir que si E tiene solución con distancia menor o igual a T, entonces E' también, e, inversamente, si E' tiene solución con costo menor o igual a T, entonces E también la tiene; es decir, existe una solución para E con distancia total recorrida menor o igual a T si y solamente si existe una solución para E' con costo total menor o igual a la misma T.

Por lo tanto, la reducción presentada es correcta y el CVRP es NP-difícil.

$\blacksquare$

Habiendo establecido que el CVRP está en la clase NP y es NP-difícil, se concluye que el CVRP es un problema NP-completo.

Capítulo 5

Análisis de otras variantes del VRP

A continuación se presenta el análisis de complejidad de algunas variantes adicionales del VRP.

Las variantes a analizar son las tres mencionadas previamente:

- El VRP con Ventanas de Tiempo o VRPTW,
- El VRP con Flota Heterogénea o HFVRP y
- El VRP con Depósitos Múltiples o MDVRP.

Para cada variante, el análisis es similar al realizado para el CVRP:

- Primero se define formalmente el problema;
- luego se muestra que está en la clase de problemas NP mediante algoritmos no-deterministas polinomiales con dos enfoques distintos;
- después se muestra que el problema es NP-difícil mediante una reducción de otro problema NP-difícil a la variante tratada;
- finalmente, habiendo probado que la variante está en la clase NP y es un problema NP-difícil, se concluye que el problema analizado es NP-completo.

En los tres casos, el problema NP-difícil a usar en la reducción es el CVRP; es decir, las reducciones son del CVRP a la variante en cuestión.

5.1. El VRPTW es NP-completo

El CVRP puede verse como un caso particular del Problema de Rutas de Vehículos con Ventanas de Tiempo (VRPTW) en el que todos los clientes tienen ventanas de tiempo para atenderlos lo suficientemente amplias para que puedan ser visitados en cualquier orden y en cualquier tiempo.

La prueba de que el VRPTW es NP-completo se basa en la prueba para el CVRP:

Solamente se agregan variables para las ventanas de tiempo de cada cliente, la cantidad de tiempo necesaria para atender a cada cliente y para viajar entre cada par de clientes, así como la cantidad total de tiempo utilizada en la solución.

Para esta sección se consideran restricciones de tiempo duras; es decir, no es posible atender a un cliente en un momento fuera de la ventana de tiempo correspondiente.

Para facilitar la prueba, no se considera una ventana de tiempo para el depósito. En el apéndice C se muestra que el problema considerando una ventana de tiempo para el depósito es equivalente.

Definición formal del VRPTW

Ejemplar:

- Un conjunto $C = \{c_0, c_1, c_2, ..., c_n\}$, de n clientes y un depósito c_0, con n natural.
- Una constante k que representa la capacidad de los vehículos.
- Una demanda o peso w_i para cada $i = 0, 1, ..., n$, con $w_0 = 0$ y $0 < w_j \leq k$, para $j = 1, 2, ..., n$.
- Un intervalo o ventana de tiempo $[y_i, z_i]$, para cada $i = 1, 2, ..., n$, con $0 < y_i < z_i$.
- Distancias reales positivas d_{ij} para cada par (c_i, c_j), con $i \neq j$.
- Para cada par (c_i, c_j), un real positivo t_{ij} que define el tiempo necesario para ir de c_i a c_j, cuando $i \neq j$, y el tempo requerido para atender al cliente c_i cuando $i = j$.

Solución:

- Un conjunto de m rutas $R = \{r_1, r_2, ..., r_m\}$, donde cada r_i es una lista ordenada de clientes de la forma $r_i = [c_0, c_{x_1}, c_{x_2}, ..., c_{x_{q_i}}, c_0]$, tal que

$$\sum_{c_j \in r_i} w_j \leq k.$$

- Una longitud total l_i para cada r_i, con $l_i = d_{0x_1} + d_{x_1x_2} + ... + d_{x_{q_i}0}$, tal que

$$\forall c_i \in C,\ \exists! r_j \in R \text{ tal que } c_i \in r_j.$$

- Un conjunto de m cronogramas $S = \{s_1, s_2, ..., s_m\}$, donde cada s_i es una lista de tiempos asociada a la ruta r_i de la forma $s_i = [p_0, p_1, ..., p_{q_i}, p_{q_i+1}]$, tal que
  - el cronograma inicia en el momento 0:

    $$p_0 = 0.$$

  - al llegar al j-ésimo cliente, hay tiempo suficiente para atenderlo antes de que termine su ventana de tiempo:

    $$y_{x_j} \leq p_j < p_j + t_{x_jx_j} \leq z_{x_j} \quad \forall j, j = 1, 2, ..., q_i.$$

  - al llegar al j-ésimo cliente, hay tiempo suficiente para atenderlo y trasladarse antes del momento de llegar al siguiente cliente:

    $$p_j + t_{x_jx_j} + t_{x_jx_{(j+1)}} \leq p_{(j+1)} \quad \forall j, j = 0, 1, ..., q_i.$$

Óptimo:

Si llamamos m al número de vehículos a utilizar, $L(R)$ a la longitud o costo total:

$$L(R) = \sum_{i=1}^{m} l_i$$

y L(S) al momento en el que el último vehículo regresa al depósito; es decir, el tiempo total que requiere el conjunto de rutas:

$$L(S) = \text{máx}\{p_{q_i} \forall i, i = 1, 2, ..., m\}.$$

entonces el óptimo es la solución de la forma (R, S) con m, $L(R)$ y $L(S)$ son mínimos, donde R es el conjunto de rutas y S su conjunto asociado de cronogramas válidos. .

Como problema de decisión:

Para construir la versión de decisión del VRPTW, establecemos cotas tanto para $L(R)$ como para $L(S)$.

El VRPTW como problema de decisión queda de la siguiente forma:

Dados un ejemplar del VRPTW y dos reales positivos T y U, determinar si existe o no una solución con costo total menor o igual a T y tiempo total menor o igual a U; es decir, responder la pregunta

$$¿\exists(R, S) \text{ tal que } L(R) \leq T \text{ y } L(S) \leq U?$$

El VRPTW está en la clase NP

Para mostrar que el VRP con Ventanas de Tiempo está en NP, se proponen dos algoritmos no-deterministas polinomiales.

El primer algoritmo utiliza un certificado de la siguiente forma:

Sea E un ejemplar del VRPTW con C su correspondiente conjunto de clientes y una flota de tamaño fijo de m vehículos, cada uno de capacidad k.

Sea, además, (R, S) un certificado para el ejemplar E, donde $R = \{r_1, r_2, ..., r_m\}$ y $S = \{s_1, s_2, ..., s_m\}$. Se busca verificar si (R, S) es una solución válida para E; es decir, validar si R y S satisfacen las condiciones enunciadas al definir una solución del VRPTW y, además, $L(R) \leq T$ y $L(S) \leq U$, con T y U las cotas dadas.

La verificación se realiza como sigue:

Se asignan etiquetas enteras a los clientes para saber si fueron visitados, inicialmente $v_i = 0$, para cada $c_i \in C$.

Se llevan también contadores para las rutas de la solución propuesta que registren la demanda total de los clientes visitados en la ruta. Inicialmente, $h_i = 0$ para cada $r_i \in R$.

También se inicializan en cero las distancias l_i de cada ruta r_i, la distancia total de la solución $L(R)$ y el tiempo total requerido $L(S)$.

La idea es verificar que las rutas de R sean correctas, entre todas visiten a todos los clientes, y en total no excedan la cota de distancia T, a la vez que se verifica que los tiempos de cada $s_i \in S$ sean correctos y que no se exceda la cota de tiempo U.

El proceso de verificación es el siguiente:

Inicializar en cero los valores de las etiquetas v_i para cada $c_i \in C$, y h_i para cada $r_i \in R$, así como las distancias $L(R)$ y l_i para cada $r_i \in R$ y el tiempo requerido $L(S)$.

Recorrer cada ruta $r_i \in R$, realizando lo siguiente para cada uno de los elementos de r_i:

> Sean x el elemento que se está verificando y a su elemento correspondiente en S, x es un cliente c_j.
>
> Si x es el primer elemento de r_i:
>
> > Si $j \neq 0$; es decir, si x es el primer elemento de la ruta y no es el depósito c_0, la ruta no comienza en el depósito, por lo que R no es un conjunto de rutas válido y la verificación termina con respuesta negativa.
> >
> > Si $a \neq 0$, el vehículo no está partiendo en el momento inicial, por lo que s_i no es un cronograma válido y la verificación termina con respuesta negativa.
>
> Si, por el contrario, x no es el primer elemento de r_i:
>
> > Sean c_v el elemento anterior a x en la ruta y b su elemento correspondiente en S:
> >
> > Sumar la distancia entre c_v y x, d_{vj}, a l_i y a $L(R)$.
> >
> > Si $L(R) > T$, se ha sobrepasado la cota de la distancia total, por lo que (R, S) no es una solución válida y la verificación termina con respuesta negativa.
> >
> > Si $b + t_{vv} + t_{vj} > a$, no es posible atender al cliente c_v y llegar a x a tiempo para cumplir el cronograma s_i, por lo que éste no es un cronograma válido y la verificación termina con respuesta negativa.
>
> Si x es el último elemento de r_i:
>
> > Si x no es el depósito c_0, la ruta no termina en el depósito, por lo que (R, S) no es una solución válida y la verificación termina con respuesta negativa.
> >
> > Si $a > U$, el cronograma s_i sobrepasa la cota de tiempo establecida, por lo que la solución no es satisfactoria y la verificación termina con respuesta negativa. Si $a > L(S)$ asignar a $L(S)$ el valor de a.
>
> Si x no es el primer ni el último elemento de r_i:
>
> > Si $j = 0$; es decir, si x es el depósito c_0, la ruta regresa al depósito más de una vez , por lo que R no es conjunto de rutas válido y la verificación termina con respuesta negativa.
> >
> > Sumar 1 a v_j.

Si $v_j > 1$, se ha visitado al cliente j más de una vez, por lo que (R, S) no es una solución válida y la verificación termina con respuesta negativa.

Sumar la demanda de x, w_j, a h_i.

Si $h_i > k$, se ha excedido la capacidad de carga del vehículo, por lo que R no es un conjunto de rutas válido y la verificación termina con respuesta negativa.

Si $y_j > a$ o $z_j < a + t_{jj}$, el cliente es atendido fuera de la ventana de tiempo permitida, por lo que s_i no es un cronograma válido y la verificación termina con respuesta negativa.

Si se recorren todas las rutas sin obtener una respuesta negativa, se procede a verificar que todos los clientes hayan sido visitados; esto es, que $v_i = 1, \forall c_i \in C, i \neq 0$.

Para esto, basta con recorrer C y revisar la etiqueta de cada cliente. Si alguna etiqueta es cero, ese cliente no fue visitado, por lo que (R, S) no es una solución válida y la verificación termina con respuesta negativa.

Si en este punto sigue sin tenerse respuesta negativa, quiere decir que todas las rutas son válidas y en su conjunto visitan a todos los clientes una vez sin exceder la cota de distancia recorrida T ni la de tiempo requerido U. Por lo tanto, R y S constituyen una solución válida de E y la verificación termina con respuesta afirmativa.

Procedimiento 5: Verificación de un certificado para el VRPTW.

Si una ruta tiene más elementos que el número de clientes, tendrá que visitar más de una vez a algún cliente, lo que provocaría una terminación con respuesta negativa en el paso 3.2, o bien pasar por el depósito en algún momento además del inicio y el fin de la ruta, lo que provocaría una terminación con respuesta negativa en el paso 3.1.

Así, en el peor caso la verificación recorre todos los elementos de cada ruta, efectuando por cada uno a lo más ocho pasos, cada uno de los cuales realiza un número constante de operaciones; y después recorre todos los clientes, realizando una sola comparación por cada uno. De esta forma, si el número de clientes $|C|$ es n, como se tienen m rutas, la verificación realiza a lo más un número de operaciones $O(m+n)$; es decir, la verificación tiene complejidad acotada por un polinomio en el tamaño del ejemplar E.

Por lo tanto, la verificación es polinomial y el VRPTW está en NP.

■

El segundo algoritmo propuesto emplea la primitiva ND-Choice como sigue:

Sea E un ejemplar del VRPTW con C su correspondiente conjunto de n clientes y una flota de tamaño fijo de m vehículos, cada uno de capacidad k.

Primero, en la parte no determinista, se construye una solución colocando cada cliente en una ruta aleatoria, de forma que se tengan m rutas, que entre todas visiten a cada cliente exactamente una vez.

Luego, en la parte de verificación, se revisa cada ruta para constatar que la demanda cubierta por el vehículo i, h_i, no sobrepase la capacidad de carga k. Paralelamente se calculan la distancia total recorrida y el tiempo necesario, para compararlos con las cotas dadas de distancia T y tiempo U, respectivamente.

La parte no determinista trabaja de la siguiente forma:

Crear un conjunto de m rutas, $R = \{r_1, r_2, ..., r_m\}$, y un conjunto de m cronogramas, $S = \{s_1, s_2, ..., s_m\}$.

Inicializar las m rutas de forma que cada una consista únicamente del depósito c_0, que es el punto de partida para cualquier ruta válida, y los m cronogramas de manera que cada uno contenga únicamente un elemento $p_0 = 0$, que es el momento inicial en el que el vehículo se encuentra en el depósito.

Hacer una copia C' del conjunto de clientes C, y aplicar n veces la primitiva ND-Choice, que en este caso realiza lo siguiente:

> Tomar un cliente c_x de C', elegido de forma aleatoria.
>
> Elegir una ruta aleatoria e insertar c_x al final de la misma.
>
> Determinar el elemento inmediato anterior de la ruta, c_a, y su correspondiente entrada en el cronograma, p_i.
>
> Agregar al cronograma correspondiente a la ruta un elemento nuevo $p_{i+1} = max\{y_x, p_i + t_{aa} + t_{ax}\}$.
>
> Eliminar x de C'.

Finalmente, agregar nuevamente el depósito c_0 al final de cada ruta, así como un elemento $p_q = p_{q-1} + t_{aa} + t_{a0}$ al cronograma correspondiente, donde q es la cantidad de elementos de la ruta y c_a es el último cliente visitado en la misma antes de regresar al depósito.

Procedimiento 6: Algoritmo no determinista para el VRPTW.

De este modo, se tiene una posible solución para E, generada de forma no determinista.

Este proceso es polinomial, ya que realiza un número constante de operaciones para cada cliente y otro número constante de operaciones para cada ruta, por lo que en total requiere un número de operaciones $O(n+m)$.

La parte de verificación opera como sigue:

Inicializar en cero las distancias $L(R)$ y l_i, $\forall i = 1, 2, ..., m$, así como el tiempo total requerido $L(S)$.

Para cada $r_i \in R$, realizar lo siguiente:

> Inicializar en cero un contador h_i de demanda cubierta por la ruta.
>
> Sea q el número de clientes visitados en la ruta, de modo que se tiene $r_i = [x_0, x_1, x_2, ..., x_{q-1}, x_q]$, con $x_0 = c_0$ y $x_q = c_0$.
>
> Recorrer los elementos x_j de r_i, desde $j = 1$ hasta $j = q$ y para cada uno:
>
> > Sumar a h_i la demanda del cliente actual.
> >
> > Si $h_i > k$, se ha excedido la capacidad del vehículo, por lo que R no es una solución válida y el proceso termina con respuesta negativa.
> >
> > Sean $c_a = x_{j-1}$ y $c_b = x_j$ los clientes anterior y actual, respectivamente.
> >
> > Sumar la distancia entre el cliente anterior y el actual, d_{ab}, a l_i y a $L(R)$.
> >
> > Si $L(R) > T$, se ha excedido la cota sobre la distancia total recorrida, por lo que R no satisface las condiciones del problema y el proceso termina con respuesta negativa.
> >
> > Verificar que el cliente x_j sea atendido dentro de la ventana de tiempo permitida, comprobando si $p_j + t_{bb} \leq z_b$. De no ser así, el cronograma s_i no es válido y el proceso termina con respuesta negativa.
>
> Finalmente, verificar si $p_q > L(S)$ y, de ser así, asignar a $L(S)$ el valor de p_q y comparar a $L(S)$ con la cota U, si $L(S)$ es mayor, se ha excedido la cota de tiempo y el proceso termina con respuesta negativa.
>
> Si no se obtiene respuesta negativa para ninguno de los elementos x_j ni se excede la cota de tiempo, significa que la ruta r_i y su correspondiente cronograma s_i son válidos, por lo que se prosigue a verificar la siguiente ruta.

Si se verifican todas las rutas sin obtener respuesta negativa, entonces la solución (R, S) es correcta y satisface las condiciones del problema, por lo que el proceso termina con respuesta afirmativa.

Procedimiento 7: Algoritmo de verificación para el VRPTW.

Esta verificación es polinomial, ya que solamente realiza un número constante de operaciones por cada elemento de cada ruta y de cada cronograma, además de que por construcción se garantiza que en su conjunto las rutas solamente visitan a cada cliente una vez y al depósito dos veces por ruta, por lo que en total se requiere de un número de operaciones $O(n+m)$.

Por lo tanto, al ser tanto la parte no-determinista como la parte de verificación polinomiales, el algoritmo es polinomial y el VRPTW está en NP.

■

El VRPTW es NP-difícil

A continuación se muestra que VRPTW es NP-difícil mediante la siguiente reducción del CVRP al VRPTW:

Sea E un ejemplar del CVRP, con $C = \{c_0, c_1, c_2, ..., c_n\}$ un conjunto con el depósito c_0 y n clientes, d_{ij} la distancia entre c_i y c_j, m el número de vehículos y k la capacidad de los mismos.

Se construye un ejemplar E' del VRPTW como sigue:

Utilizar los mismos valores definidos para E

Agregar tiempos de recorrido y atención unitarios:

$$t_{ij} = 1, \ \forall i, j, \text{ con } i \text{ y } j = 1, 2, ..., n.$$

Agregar también ventanas de tiempo que inician en el momento 0 y tienen duración $2n$:

$$y_i = 0, z_i = 2n, \ \forall i, i = 1, 2, ..., n.$$

Procedimiento 8: Reducción del CVRP al VRPTW.

Esta construcción es polinomial sobre el número n de clientes en E, ya que solamente requiere asignar $n^2 + 2n$ valores: uno para cada extremo de las n ventanas de tiempo y uno para el tiempo de recorrido o atención correspondiente a cada una de las $n \times n$ parejas de clientes; es decir, en total se requiere un número de operaciones $O(n^2)$.

A continuación se muestra que existe una solución para E con distancia total recorrida menor o igual a T, si y solamente si existe una solución para E' con costo total menor o igual a la misma T y con tiempo total requerido menor o igual a $2n + 1$.

Primero supóngase que existe una solución para E con distancia total menor o igual a T; es decir,

$$\exists R, R = \{r_1, r_2, ..., r_m\} \text{ tal que } L(R) \leq T.$$

Para E' se utilizan las mismas rutas que para E, por lo que no se sobrepasa la capacidad de ningún vehículo ni la cota de distancia total recorrida.

Como los tiempos de recorrido y atención son todos unitarios, entonces una ruta más larga o que atienda a más clientes necesariamente requerirá más tiempo.

El máximo tamaño posible para una ruta r_i es $n + 1$, lo cual solamente ocurre si en la ruta r_i se atiende a todos los clientes; es decir, si $r_i = \{c_0, c_{x_1}, c_{x_2}, ..., c_{x_n}, c_0\}$.

En este caso, es posible construir un cronograma válido s_i en el que se comience en el depósito en el momento 0, se alcance el primer cliente en el momento 1, se alcance a cada cliente subsecuente dos unidades de tiempo después (una para atender al cliente actual y una para el recorrido de un cliente al otro), se atienda al último cliente en una unidad de tiempo y, finalmente, se use una unidad de tiempo más para regresar al depósito:

$$s_i = \{0, 1, 3, 5, 7, ..., 2n - 1, 2n + 1\}.$$

Así, aún si existe una ruta de tamaño máximo, es posible construir un cronograma para ella que requiera de $2n + 1$ unidades de tiempo, por lo que no sobrepasa la cota de tiempo.

Como se estableció que una ruta más larga requiere mayor tiempo, y una ruta de tamaño máximo no sobrepasa la cota de tiempo, entonces para cualquier ruta es posible construir un cronograma válido que no sobrepase la cota de tiempo U.

De esta forma, ya que el tiempo total requerido $L(S)$ corresponde al mayor tiempo requerido por una ruta individual, se tiene que $L(S)$ es, a lo más, igual a la cota U.

Por lo tanto, existe una solución para E' tal que $L(R) \leq T$ y $L(S) \leq U$.

$\square$

Supóngase ahora que existe una solución para E' con costo total menor o igual a T y tiempo total menor o igual a U; es decir,

$$\exists (R, S) \text{ tal que } L(R) \leq T \text{ y } L(S) \leq U.$$

Como el conjunto de clientes, las distancias entre los mismos, el número y capacidad de vehículos son iguales para los ejemplares E y E', utilizando el mismo conjunto de rutas R de la solución para E' se tiene una solución válida para E, ya que las rutas siguen siendo válidas y se sabe que $L(R) \leq T$.

Por lo tanto, existe una solución para E con distancia o costo total menor o igual a T.

$\square$

De esta forma, se tiene que si E tiene solución con distancia menor o igual a T, entonces E' tiene solución con distancia total menor o igual a T y tiempo requerido menor o igual a U, y si E' tiene solución con costo menor o igual a T, entonces E también la tiene. Esto significa que existe una solución para el ejemplar E con distancia total recorrida menor o igual a T, si y solamente si existe una solución para el ejemplar E' con costo total menor o igual a la misma T y tiempo requerido menor o igual a U. Por lo tanto, la reducción presentada es correcta y el VRPTW es NP-difícil.

$\blacksquare$

Habiendo establecido que el VRPTW está en la clase NP y es NP-difícil, se tiene que el VRPTW es un problema NP-completo.

5.2. El HFVRP es NP-completo

El CVRP es un caso particular del Problema de Rutas de Vehículos con Flota Heterogénea (HFVRP) en el que todos los vehículos utilizados tienen la misma capacidad y los mismos costos de operación.

La prueba de que el HFVRP es NP-completo se basa en la prueba para el CVRP:

Solamente se reemplazan las variables de capacidad de carga y de costo por m variables cada una, donde m es el número de vehículos a utilizar.

Definición formal del HFVRP

Ejemplar:

- Un conjunto $C = \{c_0, c_1, c_2, ..., c_n\}$, de n clientes y un depósito c_0, con n natural.
- Una demanda o peso w_i, para cada $i = 0, 1, ..., n$, con $w_0 = 0$ y $w_j > 0$, para $j = 1, 2, ..., n$.
- Un conjunto $G = \{g_1, g_2, ..., g_m\}$, de m vehículos, con m natural.
- Para cada vehículo g_i, una constante k_i que representa su capacidad y una función d_i de costos o distancias reales positivas, donde $d_i(a, b)$ es el costo de trasladar el vehículo g_i desde c_a hasta c_b, para $a \neq b$.

Solución:

- Un conjunto $R = \{r_1, r_2, ..., r_m\}$ de m rutas, cada una de las cuales tiene la forma $r_i = [c_0, c_{x_1}, c_{x_2}, ..., c_{x_{q_i}}, c_0]$, tal que $\forall c_i \in C$, $\exists! r_j \in R$ tal que $c_i \in r_j$, donde r_i es una lista ordenada de clientes que representa la ruta que utiliza el vehículo g_i y satisface
$$\sum_{c_j \in r_i} w_j \leq k_i.$$
- Para cada r_i, una longitud total $l_i = d_i(0, x_1) + d_i(x_1, x_2) + ... + d_i(x_{q_i}, 0)$.

Óptimo:

Si llamamos $L(R)$ a la longitud o costo total:

$$L(R) = \sum_{i=1}^{m} l_i$$

entonces el óptimo es la solución R con $L(R)$ mínima.

Como problema de decisión:

Dados un ejemplar del HFVRP y un real positivo T, determinar si existe o no una solución con costo total menor o igual a T; es decir, responder la pregunta

$$¿\exists R \text{ tal que } L(R) \leq T?$$

El HFVRP está en la clase NP

Para mostrar que el VRP con Flota Heterogénea está en la clase NP, se proponen dos algoritmos no-deterministas polinomiales.

El primer algoritmo utiliza un certificado de la siguiente forma:

Sea E un ejemplar del HFVRP con C su correspondiente conjunto de clientes y una flota G de m vehículos, cada vehículo g_i de capacidad k_i.

Sea además R un certificado para E, de la forma $R = \{r_1, r_2, ..., r_m\}$. Se busca verificar si R es una solución válida para E; es decir, determinar si R satisface las condiciones enunciadas al definir una solución del HFVRP y, además, $L(R) \leq T$, con T una cota dada.

La verificación se realiza como sigue:

Se asignan etiquetas enteras a los clientes para saber si fueron visitados, inicialmente $v_i = 0$, para cada $c_i \in C$.

Se llevan también contadores para las rutas de la solución propuesta que registren la demanda total de los clientes visitados en la ruta, inicialmente $h_i = 0$ para cada $r_i \in R$.

También se da el valor inicial de cero a las distancias l_i de cada ruta r_i y la distancia total de la solución $L(R)$.

La idea es verificar que las rutas de R sean correctas, entre todas visiten a todos los clientes, y en total no excedan la cota de distancia T.

El proceso de verificación es el siguiente:

Inicializar en cero los valores de las etiquetas v_i para cada $c_i \in C$, y h_i para cada $r_i \in R$, así como las distancias $L(R)$ y l_i para cada $r_i \in R$.

Recorrer cada ruta $r_i \in R$, realizando lo siguiente para cada uno de los elementos de r_i:

> Sea x el elemento que se está verificando, x es un cliente c_j.
>
> Si x es el primer elemento de r_i:
>
> > Si $j \neq 0$; es decir, si x es el primer elemento de la ruta y no es el depósito c_0, la ruta no comienza en el depósito, por lo que R no es una solución válida y la verificación termina con respuesta negativa.
>
> Si x no es el primer elemento de r_i:
>
> > Sea c_z el elemento anterior a x en la ruta.
> >
> > Sumar la distancia entre c_z y x, $d_i(z, j)$, a l_i y a $L(R)$.
> >
> > Si $L(R) > T$, se ha sobrepasado la cota de la distancia total, por lo que R no es una solución válida y la verificación termina con respuesta negativa.
>
> Si x es el último elemento de r_i y no es el depósito c_0, la ruta no termina en el depósito, por lo que R no es una solución válida y la verificación termina con respuesta negativa.
>
> Si x no es el primer ni el último elemento de r_i:
>
> > Si $j = 0$; es decir, si x es el depósito c_0, la ruta regresa al depósito más de una vez , por lo que R no es una solución válida y la verificación termina con respuesta negativa.
> >
> > Sumar 1 a v_j.
> >
> > Si $v_j > 1$, se ha visitado al cliente j más de una vez, por lo que R no es una solución válida y la verificación termina con respuesta negativa.
> >
> > Sumar la demanda de x, w_j, a h_i.

Si $h_i > k_i$, se ha excedido la capacidad de carga del vehículo, por lo que R no es una solución válida y la verificación termina con respuesta negativa.

Si se recorren todas las rutas sin obtener una respuesta negativa, proceder a verificar que todos los clientes hayan sido visitados; esto es, que $v_i = 1, \forall c_i \in C, i \neq 0$.

Para esto, basta recorrer C y revisar la etiqueta de cada cliente. Si alguna etiqueta es cero, ese cliente no fue visitado, por lo que R no es una solución válida y la verificación termina con respuesta negativa.

Si en este punto sigue sin tenerse respuesta negativa, esto quiere decir que todas las rutas son válidas y en su conjunto visitan a todos los clientes una vez sin exceder la cota de distancia recorrida T. Por lo tanto, R es una solución válida de E y la verificación termina con respuesta afirmativa.

Procedimiento 9: Verificación de un certificado para el HFVRP.

Si una ruta tiene más elementos que el número de clientes, tendrá que visitar más de una vez a algún cliente, lo que provocaría una terminación con respuesta negativa en el paso 3.2, o bien pasar por el depósito en algún momento además del inicio y el fin de la ruta, lo que provocaría una terminación con respuesta negativa en el paso 3.1.

Por tanto, en el peor de los casos esta verificación recorre todos los elementos de cada ruta, efectuando por cada uno a lo más cinco pasos, cada uno de los cuales realiza un número constante de operaciones; y después recorre todos los clientes, realizando una sola comparación por cada uno.

Así, si el número de clientes $|C|$ es n, como se tienen m rutas, la verificación realiza a lo más un número de operaciones $O(m + n)$; es decir, la verificación tiene complejidad acotada por un polinomio en el tamaño de E.

Se tiene entonces que la verificación es polinomial y el HFVRP está en NP.

■

El segundo algoritmo propuesto emplea la primitiva ND-Choice como sigue:

Sean T una cota real positiva y E un ejemplar del HFVRP con C su correspondiente conjunto de clientes y una flota G de m vehículos, cada vehículo g_i de capacidad k_i.

Primero, en la parte no determinista, se construirá una solución R colocando cada cliente en una ruta aleatoria, de forma que entre las m rutas visiten a cada cliente exactamente una vez.

Luego, en la parte de verificación, se revisará cada ruta r_i para constatar que no se sobrepase la capacidad de carga k_i del vehículo correspondiente g_i, a la vez que se calcula la distancia total recorrida para compararla con la cota establecida T.

La parte no determinista trabaja de la siguiente forma:

Crear un conjunto de m rutas, $R = \{r_1, r_2, ..., r_m\}$.

Inicializar las m rutas, de forma que cada una consista únicamente del depósito c_0, que es el punto de partida para cualquier ruta válida.

Hacer una copia C' del conjunto de clientes C, y aplicar n veces la primitiva ND-Choice, que en este caso realiza lo siguiente:

> Tomar un cliente x de C', elegido de forma aleatoria para evitar que el orden en el que están listados los clientes en C tenga influencia sobre el resultado.
>
> Elegir una ruta aleatoria e inserta x al final de la misma.
>
> Eliminar x de C'.

Finalmente, agregar nuevamente el depósito c_0 al final de cada ruta, de forma que, si no se sobrepasa la capacidad del vehículo, la ruta sea válida.

Procedimiento 10: Algoritmo no determinista para el HFVRP.

De este modo, se tiene una posible solución para E, generada de forma no determinista.

Este proceso es polinomial, ya que realiza un número constante de operaciones para cada uno de los n clientes, y otro número constante de operaciones para cada una de las m rutas, por lo que en total realiza un número de operaciones $O(n + m)$.

La parte de verificación opera como sigue:

Inicializar en cero las distancias $L(R)$ y l_i, $\forall i = 1, 2, ..., m$.

Para cada $r_i \in R$, realizar lo siguiente:

> Inicializar en cero un contador h_i de demanda cubierta por la ruta.

Sea q el número de clientes visitados en la ruta, de modo que $r_i = [x_0, x_1, x_2, ..., x_{q-1}, x_q]$, con $x_0 = c_0$ y $x_q = c_0$.

Recorrer los elementos x_j de r_i, desde $j = 1$ hasta $j = q$, y para cada uno:

Sumar a h_i la demanda del cliente actual.

Si $h_i > k_i$, se ha excedido la capacidad del vehículo, por lo que R no es una solución válida y el proceso termina con respuesta negativa.

Asignar $c_y = x_{j-1}$ y $c_z = x_j$ los clientes anterior y actual, respectivamente, y sumar la distancia entre el cliente anterior y el actual, $d_i(y, z)$, a l_i y a $L(R)$.

Si $L(R) > T$, se ha excedido la cota sobre la distancia total recorrida, por lo que la solución R no satisface las condiciones del problema y el proceso termina con respuesta negativa.

Si no se obtiene respuesta negativa para ninguno de los elementos x_j, significa que la ruta r_i es válida, y se prosigue a verificar la siguiente ruta.

Si se verifican todas las rutas sin obtener respuesta negativa, entonces la solución R es correcta y satisface las condiciones del problema, por lo que el proceso termina con respuesta afirmativa.

Procedimiento 11: Algoritmo de verificación para el HFVRP.

Esta verificación es polinomial, ya que solamente realiza un número constante de operaciones por cada elemento de cada ruta, y por construcción se garantiza que en su conjunto las rutas solamente visitan a cada cliente una vez, y al depósito dos veces por ruta, por lo que en total se requiere de un número de operaciones $O(n + m)$.

Por lo tanto, al ser tanto la parte no-determinista como la parte de verificación polinomiales, el algoritmo es polinomial y el HFVRP está en la clase NP.

■

El HFVRP es NP-difícil

A continuación se muestra que el HFVRP es NP-difícil mediante la siguiente reducción del CVRP al HFVRP:

Sea E un ejemplar del CVRP, con $C = \{c_0, c_1, c_2, ..., c_n\}$ un conjunto con el depósito c_0 y n clientes, d_{ij} la distancia entre c_i y c_j, m el número de vehículos y k la capacidad de los mismos.

Se construye un ejemplar E' del HFVRP como sigue:

Utilizar el mismo conjunto de clientes C.

Preservar también las distancias entre los clientes, de modo que para cada vehículo v_i el costo para ir desde un cliente c_a a otro cliente c_b es:

$$d_i(a, b) = d_{ab} \;\; \forall i, i = 1, 2, ..., m.$$

Construir el conjunto de m vehículos $V = \{v_1, v_2, ...v_m\}$, asignando a cada vehículo v_i capacidad k:

$$k_i = k \;\; \forall i, i = 1, 2, ..., m$$

Procedimiento 12: Reducción del CVRP al HFVRP.

Esta construcción es polinomial sobre el número n de clientes y el tamaño m de la flota en el ejemplar E, ya que solamente requiere asignar n clientes, m capacidades y nm distancias, por lo que en total se requieren $O(mn)$ operaciones.

A continuación se muestra que existe una solución para E con distancia total recorrida menor o igual a T, si y solamente si existe una solución para E' con costo total menor o igual a la misma T:

Primero supóngase que existe una solución para E con distancia total menor o igual a T; es decir,

$$\exists R = \{r_1, r_2, ..., r_m\} \text{ tal que } L(R) \leq T.$$

Tomando las mismas rutas para E', se tiene que ninguna excederá la capacidad del vehículo, ya que $k_i = k, \; \forall i, i = 1, 2, ..., m$.

De esta forma, se tiene que R es un conjunto de rutas válido para E'.

Además, como los costos en el ejemplar E' son iguales a las distancias en el ejemplar E, se tiene que para cada ruta r_i, su costo l_i también se mantiene, por lo que el costo total para E' sigue siendo la misma $L(R)$.

Por lo tanto, existe una solución R para E' con costo total $L(R)$ menor o igual a T.

$\square$

Supóngase ahora que existe una solución para el ejemplar E' con costo total menor o igual a T; es decir,

$$\exists R = \{r_1, r_2, ..., r_m\} \text{ tal que } L(R) \leq T.$$

Tomando las mismas rutas para E, se tiene que ninguna excederá la capacidad del vehículo, ya que $k_i = k, \ \forall i, i = 1, 2, ..., m$.

Además, como los costos en E' son iguales a las distancias en E, se tiene que para cada r_i, su distancia l_i también se mantiene, por lo que la distancia total para E' sigue siendo la misma $L(R)$.

Por lo tanto, existe una solución para E con distancia total recorrida menor o igual a T.

$\square$

De esta forma, se tiene que si E tiene solución con distancia menor o igual a T, entonces E' tiene solución con costo total menor o igual a T, y si E' tiene solución con costo menor o igual a T, entonces E también la tiene. Esto significa que existe una solución para el ejemplar E con distancia total recorrida menor o igual a T, si y solamente si existe una solución para el ejemplar E' con costo total menor o igual a la misma T.

Por lo tanto, la reducción presentada es correcta y el HFVRP es NP-difícil.

$\blacksquare$

Habiendo establecido que el HFVRP está en la clase NP y es NP-difícil, se tiene que el HFVRP es un problema NP-completo.

5.3. El MDVRP es NP-completo

El CVRP es un caso particular del Problema de Rutas de Vehículos con Depósitos Múltiples (MDVRP) en el que todos los vehículos parten de un solo depósito.

La prueba de que el MDVRP es NP-completo se basa en la prueba para el CVRP:
Solamente se reemplaza el depósito único por un conjunto de depósitos, y se consideran distancias entre cada depósito y cada cliente. No es necesario utilizar todos los depósitos; puede haber alguno del que no salgan vehículos.

Para esta sección se considera una versión del MDVRP en la que cada vehículo debe regresar al depósito del que partió originalmente; es decir, cada ruta de la solución debe iniciar y terminar en el mismo depósito.

Definición formal del MDVRP

Ejemplar:

- Un conjunto $C = \{c_1, c_2, ..., c_n\}$ de n clientes, con n natural.
- Un conjunto $F = \{f_1, f_2, ..., f_u\}$ de u depósitos.
- Una demanda o peso $w_i > 0$, para cada cliente c_i, $i = 1, 2, ..., n$.
- Una constante k que representa la capacidad de los vehículos.
- Distancias reales positivas d_{ij} para cada par de clientes (c_i, c_j), con $i \neq j$, d'_{ij} para cada par depósito-cliente (f_i, c_j) y d''_{ij} para cada par cliente-depósito (c_i, f_j).

Solución:

- Un conjunto R de m rutas, $R = \{r_1, r_2, ..., r_m\}$, donde cada ruta r_i es una lista ordenada de clientes de la forma $r_i = [f_{x_0}, c_{x_1}, c_{x_2}, ..., c_{x_{q_i}}, f_{x_0}]$, tal que la demanda cubierta por la ruta no sobrepase la capacidad del vehículo; esto es,

$$\sum_{c_j \in r_i} w_j \leq k$$

  y se cumple que $\forall c_i \in C$, $\exists! r_j \in R$ tal que $c_i \in r_j$.
- Una longitud total $l_i = d'_{x_0 x_1} + d_{x_1 x_2} + d_{x_2 x_3} + ... + d_{x_{q_i - 1} x_{q_i}} + d''_{x_{q_i} x_0}$ para cada r_i.

Óptimo:

Si llamamos m al número de vehículos a utilizar y $L(R)$ a la longitud o costo total:

$$L(R) = \sum_{i=1}^{m} l_i$$

entonces el óptimo es la solución R con m y $L(R)$ mínimos.

Como problema de decisión:

Dados un ejemplar del CVRP y un real positivo T , determinar si existe o no una solución con costo total menor o igual a T; es decir, determinar la respuesta a la pregunta

$$¿\exists R \text{ tal que } L(R) \leq T?$$

El MDVRP está en la clase NP

Para mostrar que el VRP con Depósitos Múltiples está en NP, se proponen dos algoritmos no-deterministas polinomiales.

El primer algoritmo utiliza un certificado de la siguiente forma:

Sea E un ejemplar del MDVRP con C su correspondiente conjunto de clientes, F su conjunto de depósitos y una flota de m vehículos, cada uno de capacidad k.

Además, sea R un certificado para E, de la forma $R = \{r_1, r_2, ..., r_m\}$. Se busca verificar si R es una solución válida para E; es decir, determinar si R satisface las condiciones enunciadas al definir una solución del MDVRP y, además, $L(R) \leq T$, con T una cota dada.

La verificación se realiza como sigue:

Se asignan etiquetas enteras a los clientes para saber si fueron visitados. Inicialmente, $v_i = 0$, para cada $c_i \in C$.

Se llevan también contadores para las rutas de la solución propuesta que registren la demanda total de los clientes visitados en la ruta, inicialmente $h_i = 0$ para cada $r_i \in R$.

También se inicializan en cero las distancias l_i de cada ruta r_i y la distancia total de la solución $L(R)$.

La idea es verificar que las rutas de R sean correctas, entre todas visiten a todos los clientes, y en total no excedan la cota de distancia T.

El proceso de verificación es el siguiente:

> Inicializar en cero los valores de las etiquetas v_i para cada $c_i \in C$, y h_i para cada $r_i \in R$, así como las distancias $L(R)$ y l_i para cada $r_i \in R$.
>
> Recorrer cada ruta $r_i \in R$, realizando lo siguiente para cada uno de los elementos de r_i:
>
> > Sea x el elemento que se está verificando, x es un cliente c_j o un depósito f_j.
> >
> > Si x es un depósito:
> >
> > > Si x no es el primer ni el último elemento de r_i, la ruta regresa al depósito más de una vez o pasa por más de un depósito, por lo que R no es una solución válida y la verificación termina con respuesta negativa.
> > >
> > > Si x es el último elemento de la ruta:
> > >
> > > > Sea f_z el primer elemento de la ruta.
> > > >
> > > > Si $z \neq j$; es decir, si el depósito de inicio y fin de la ruta son distintos, la ruta no cumple con las condiciones establecidas, por lo que R no es una solución válida y la verificación termina con respuesta negativa.
> > > >
> > > > Si x no es el segundo elemento de la ruta; esto es, si se atiende a algún cliente en la ruta:
> > > >
> > > > > Sea c_z el elemento anterior a x en la ruta; es decir, el último cliente de la ruta.
> > > > >
> > > > > Sumar la distancia entre c_z y x, d''_{jz}, a l_i y a $L(R)$.
> > > > >
> > > > > Si $L(R) > T$, se ha sobrepasado la cota de la distancia total, por lo que R no es una solución válida y la verificación termina con respuesta negativa.
> >
> > Si x es el segundo elemento de la ruta; lo que significa que x es el primer cliente de r_i:

Sumar 1 a v_j.

Si $v_j > 1$, se ha visitado al cliente j más de una vez, por lo que R no es una solución válida y la verificación termina con respuesta negativa.

Sumar la demanda de x, w_j, a h_i.

Si $h_i > k$, se ha excedido la capacidad de carga del vehículo, por lo que R no es una solución válida y la verificación termina con respuesta negativa.

Sea f_z el elemento anterior a x en la ruta; es decir, el depósito de la ruta.

Sumar la distancia entre f_z y x, d'_{zj}, a l_i y a $L(R)$.

Si $L(R) > T$, se ha sobrepasado la cota de la distancia total, por lo que R no es una solución válida y la verificación termina con respuesta negativa.

Si x no es el primer, segundo ni último elemento de r_i:

Sumar 1 a v_j.

Si $v_j > 1$, se ha visitado al cliente j más de una vez, por lo que R no es una solución válida y la verificación termina con respuesta negativa.

Sumar la demanda de x, w_j, a h_i.

Si $h_i > k$, se ha excedido la capacidad de carga del vehículo, por lo que R no es una solución válida y la verificación termina con respuesta negativa.

Sea c_z el elemento anterior a x en la ruta.

Sumar la distancia entre c_z y x, d_{zj}, a l_i y a $L(R)$.

Si $L(R) > T$, se ha sobrepasado la cota de la distancia total, por lo que R no es una solución válida y la verificación termina con respuesta negativa.

Si se recorren todas las rutas sin obtener una respuesta negativa, entonces proceder a verificar que todos los clientes hayan sido visitados; esto es, que $v_i = 1, \forall c_i \in C, i \neq 0$.

Para esto, basta con recorrer C y revisar la etiqueta de cada cliente. Si alguna etiqueta es cero, ese cliente no fue visitado, por lo que R no es una solución válida y la verificación termina con respuesta negativa.

Si en este punto sigue sin tenerse respuesta negativa, esto quiere decir que todas las rutas son válidas y en su conjunto visitan a todos los clientes una vez sin exceder la cota de distancia recorrida T. Por lo tanto, R es una solución válida de E y la verificación termina con respuesta afirmativa.

Procedimiento 13: Verificación de un certificado para el MDVRP.

Si una ruta tiene más elementos que el número de clientes, tendrá que visitar más de una vez a algún cliente, lo que provocaría una terminación con respuesta negativa en el paso 3.2, o bien pasar por un depósito en algún momento además del inicio y el fin de la ruta, lo que provocaría una terminación con respuesta negativa en el paso 1.1.

De esta forma, en el peor caso esta verificación recorre todos los elementos de cada ruta, efectuando por cada uno a lo más cuatro pasos, cada uno de los cuales realiza un número constante de operaciones; y después recorre todos los clientes, realizando una sola comparación por cada uno. Así, si el número de clientes $|C|$ es n, como se tienen m rutas, la verificación realiza a lo más un número de operaciones $O(m + n)$; es decir, la verificación tiene complejidad acotada por un polinomio del tamaño de E.

Por lo tanto, la verificación es polinomial y el MDVRP está en NP.

■

El segundo algoritmo propuesto emplea la primitiva ND-Choice como sigue:

Sea E un ejemplar del MDVRP con C su correspondiente conjunto de clientes, F su conjunto de depósitos y una flota de m vehículos, cada uno de capacidad k.

Primero, en la parte no determinista, se construirá una solución R colocando cada cliente en una ruta aleatoria, de forma que entre las m rutas visiten a cada cliente exactamente una vez.

Después, en la parte de verificación, se revisará cada ruta r_i para constatar que la demanda atendida en ella, h_i, no sobrepase la capacidad de carga k del vehículo, a la vez que se calcula la distancia total recorrida para compararla con la cota T dada.

La parte no determinista trabaja de la siguiente forma:

Crear un conjunto de m rutas, $R = \{r_1, r_2, ..., r_m\}$.

Inicializar las m rutas, de forma que cada una consista únicamente un depósito f_u elegido de forma aleatoria del conjunto F, de modo que se cuenta con un punto de partida para cada ruta.

Hacer una copia C' del conjunto de clientes C, y aplicar n veces la primitiva ND-Choice, que en este caso realiza lo siguiente:

Tomar un cliente x de C', elegido de forma aleatoria para evitar que el orden en el que están listados los clientes en C tenga influencia sobre el resultado.

Elegir una ruta aleatoria e insertar x al final de la misma.

Eliminar x de C'.

Finalmente, agrega al final de cada ruta el mismo depósito f_u en el que la ruta comienza, de forma que, si no se sobrepasa la capacidad del vehículo, la ruta se válida.

Procedimiento 14: Algoritmo no determinista para el MDVRP.

De este modo, se tiene una posible solución para E, generada de forma no determinista.

Este proceso es polinomial, ya que realiza un número constante de operaciones para cada cliente y otro número constante de operaciones para cada ruta, por lo que en total requiere un número de operaciones $O(n+m)$.

La parte de verificación opera como sigue:

Inicializar en cero las distancias $L(R)$ y l_i, $\forall i = 1, 2, ..., m$.

Para cada $r_i \in R$, realizar lo siguiente:

Inicializar en cero un contador h_i de demanda cubierta por la ruta.

Sea q el número de clientes visitados en la ruta, de modo que $r_i = [x_0, x_1, x_2, ..., x_{q-1}, x_q]$, con $x_0 = x_q = f_u$.

Recorrer los elementos x_j de r_i, desde $j = 1$ hasta $j = q - 1$, y para cada uno:

Sumar a h_i la demanda del cliente actual.

Si $h_i > k$, se ha excedido la capacidad del vehículo, por lo que R no es una solución válida y el proceso termina con respuesta negativa.

Si $j > 1$; es decir, si hay un cliente antes de x_j en la ruta.

Sean $c_y = x_{j-1}$ y $c_z = x_j$ los clientes anterior y actual, respectivamente.

Sumar la distancia entre el cliente anterior y el actual, d_{yz}, a l_i y a $L(R)$.

Si $L(R) > T$, se ha excedido la cota sobre la distancia total recorrida, por lo que la solución R no satisface las condiciones del problema y el proceso termina con respuesta negativa.

Si no se obtiene respuesta negativa para ninguno de los clientes, proceder a sumar las distancias entre el depósito y los clientes a los extremos de la ruta:

Sean c_y el primer cliente de la ruta y c_z el último.

Sumar d'_{uy} y d''_{zu} a l_i y a $L(R)$.

Si $L(R) > T$, se ha excedido la cota sobre la distancia total recorrida, por lo que la solución R no satisface las condiciones del problema y el proceso termina con respuesta negativa.

Si tras esta última suma sigue sin obtenerse respuesta negativa, significa que la ruta r_i es válida, proseguir a verificar la siguiente ruta.

Si se verifican todas las rutas sin obtener respuesta negativa, entonces la solución R es correcta y satisface las condiciones del problema, por lo que el proceso termina con respuesta afirmativa.

Procedimiento 15: Algoritmo de verificación para el MDVRP.

Esta verificación es polinomial, ya que solamente realiza un número constante de operaciones por cada elemento de cada ruta, y por construcción se garantiza que en su conjunto las rutas solamente visitan a cada cliente una vez, y al depósito dos veces por ruta, por lo que en total se requiere de un número de operaciones $O(n + m)$.

Por lo tanto, al ser tanto la parte no-determinista como la parte de verificación polinomiales, el algoritmo es polinomial y el MDVRP está en la clase NP.

■

El MDVRP es NP-difícil

A continuación se muestra que MDVRP es NP-difícil mediante la siguiente reducción del CVRP al MDVRP:

Sea E un ejemplar del CVRP, con $C = \{c_1, c_2, ..., c_n\}$ un conjunto de n clientes, d_{ij} la distancia entre c_i y c_j, m el número de vehículos y k la capacidad de los mismos.

Se construye un ejemplar E' del MDVRP como sigue:

Utilizar un conjunto de clientes C' igual a C salvo por la exclusión del depósito c_0; es decir,

$$C' = \{c_1, c_2, ..., c_n\}.$$

Construir el conjunto de m depósitos:

$$F = \{f_1, f_2, ..., f_m\}.$$

Preservar las distancias, de modo que la distancia entre los clientes c_i y c_j es d_{ij}, la distancia entre el depósito f_a y el cliente c_i es:

$$d'_{ai} = d_{0i}$$

y la distancia entre el cliente c_i y el depósito f_a es:

$$d''_{ia} = d_{i0},$$

para todo par de clientes c_i y c_j en C' y para todo depósito f_a en F.

Procedimiento 16: Reducción del CVRP al MDVRP.

Esta construcción es polinomial sobre el número n de clientes en el ejemplar E, ya que solamente requiere asignar n clientes, m depósitos y $n^2 + 2nm$ distancias, pero como en cada ruta debe atenderse al menos un cliente podemos afirmar que $m \leq n$, por lo que en total se necesita un número de operaciones $O(n^2)$.

A continuación, se muestra que existe una solución para E con distancia total recorrida menor o igual a T, si y solamente si existe una solución para E' con distancia total recorrida menor o igual a la misma T:

Primero supóngase que existe una solución para E con distancia total menor o igual a T; es decir,

$$\exists R, R = \{r_1, r_2, ..., r_m\} \text{ tal que } L(R) \leq T.$$

Sea $R' = \{r'_1, r'_2, ..., r'_m\}$ un conjunto de rutas tal que cada r'_i se obtiene tomando r_i y reemplazando su primer y último elementos por f_i.

Como sólo se reemplazaron el primer y último elementos, que eran ambos c_0, las rutas siguen estando correctamente construidas, comenzando en un depósito, atendiendo a un número de clientes y terminando en el depósito en el que iniciaron. Además, la demanda total atendida en cada ruta no cambia, por lo que sigue sin excederse la capacidad de carga de los vehículos.

Por lo tanto, R' es un conjunto de rutas válido para E'.

Además, como las distancias en E' son iguales a las de E, se tiene que para cada r'_i, su costo sigue siendo l_i, por lo que la distancia total recorrida en R' es la misma $L(R)$.

Por lo tanto, existe una solución R' para E' cuya distancia total recorrida $L(R') = L(R)$ es menor o igual a T.

$\square$

Supóngase ahora que existe una solución para E' con costo total menor o igual a T; es decir,

$$\exists R = \{r_1, r_2, ..., r_m\} \text{ tal que } L(R) \leq T.$$

Aplicando un proceso análogo al del caso anterior, se construye un conjunto de rutas $R' = \{r'_1, r'_2, ..., r'_m\}$ tal que cada r'_i se obtiene tomando r_i y reemplazando su primer y último elementos por c_0.

Como solo se reemplazaron el primer y último elementos de cada ruta, las rutas siguen estando correctamente construidas, comenzando en el depósito, atendiendo a un número de clientes y terminando en el mismo depósito c_0. Además, la demanda total atendida en cada ruta no cambia, por lo que sigue sin excederse la capacidad de carga de los vehículos.

Por lo tanto, R' es un conjunto de rutas válido para E.

Además, como las distancias en E' son iguales a las de E, se tiene que para cada r'_i, su costo sigue siendo l_i, por lo que la distancia total recorrida en R' es la misma $L(R)$.

Por lo tanto, existe una solución R' para E cuya distancia total recorrida $L(R') = L(R)$ es menor o igual a T.

$\square$

De esta forma, se tiene que si E tiene solución con distancia total menor o igual a T, entonces E' también tiene solución con distancia total menor o igual a T, y si E' tiene solución con distancia total menor o igual a T, entonces E también la tiene.

Esto significa que existe una solución para el ejemplar E con distancia total recorrida menor o igual a T, si y solamente si existe una solución para el ejemplar E' con distancia total recorrida menor o igual a la misma T.

Por lo tanto, la reducción presentada es correcta y MDVRP es NP-difícil.

■

Habiendo establecido que el MDVRP está en la clase NP y es NP-difícil, se tiene que el MDVRP es un problema NP-completo.

5.4. Estrategia general para una variante del VRP

En este capítulo se han presentado una serie de pruebas que utilizan estrategias similares para mostrar que cierto tipo de variantes del VRP son también problemas NP-completos.

Estas pruebas se basan en la técnica de restricción [7], la cual consiste en lo siguiente:

Sea π un problema que se busca probar que es NP-difícil y sea π' un caso particular de π que se sabe que es NP-difícil. Sean Y y X el conjunto de ejemplares de π' y de π, respectivamente.

Es posible mostrar que π es NP-difícil mediante los siguientes pasos:

- Especificar el conjunto Z de restricciones a las variables de π, necesarias para definir Y en términos de X; es decir, especificar Z de forma que se cumpla $Y = \{e \in X \mid e \text{ satisface } Z\}$.
- Z induce una reducción R de π' a π, la cual consiste en tomar un ejemplar E' de π' y construir un ejemplar E de π asignando a las variables los valores indicados según Z y E'.
- La relación entre π y π' y la construcción de R aseguran que E tendrá respuesta afirmativa si y sólo si E' tiene respuesta afirmativa. Además, si el tamaño de Z es polinomial respecto al tamaño de E', la construcción de E puede realizarse en tiempo acotado por un polinomio en el tamaño de E'.

- Por lo tanto, siempre que $|Z| \in O(P(E))$, para algún polinomio P, la reducción R construida es una reducción polinomial de π' en π, lo cual, como se sabe que π' es NP-difícil, prueba que π es NP-difícil.

Con la finalidad de servir como apoyo en el análisis de variantes adicionales, a continuación se detalla la forma en que se aplicó la técnica de restricción a las variantes del VRP, destacando los pasos seguidos durante el desarrollo de las demostraciones presentadas en este capítulo.

Sea X una variante del VRP, con variables adicionales respecto al CVRP; es decir, X es una generalización del CVRP y, equivalentemente, el CVRP es un caso particular de X.

Primero, se define formalmente a X con el fin de encontrar las variables que lo diferencian del CVRP; es decir, las variables que, de restringirse a ciertos valores, llevan a X al caso particular del CVRP.

- En el VRPTW, estas son el inicio y fin de la ventana de tiempo y tiempo de atención para cada cliente, el tiempo de recorrido entre cada par de clientes, los cronogramas para cada ruta y la cota de tiempo total.
- En el caso del HFVRP, son las capacidades y costos de recorrido de cada vehículo.
- Para el MDVRP, las variables adicionales son las distancias entre cada depósito y cada cliente, en ambas direcciones.

Una vez determinadas las variables adicionales, se prueba de que X está en NP, utilizando como base el algoritmo no-determinista para el CVRP y agregándole verificaciones para validar los valores de las variables adicionales, que arrojen respuestas negativas si algún valor es inválido o si se sobrepasa una cota adicional del problema.

Ya sea que se consideren algoritmos de validación para certificados o algoritmos no-deterministas que utilicen la primitiva ND-Choice, los cambios realizados son muy parecidos.

- Para el VRPTW, se agregan verificaciones para cada elemento de cada ruta, para validar si el cliente es atendido dentro de la ventana de tiempo correspondiente y si la cota de tiempo no se ha sobrepasado.

- En el caso del HFVRP, basta modificar las verificaciones existentes para que consideren el costo y capacidad del vehículo correspondiente a la ruta.
- Para el MDVRP, debe cuidarse que se esté considerando la distancia entre el cliente y el depósito correcto, además de que la ruta comience y termine en el mismo depósito.

Para mostrar que el problema X es NP-difícil, se presenta una reducción del CVRP a X, la cual consiste en tomar un ejemplar E del CVRP y agregar de forma explícita las variables adicionales con los valores particulares correspondientes al CVRP, manteniendo los demás valores de E iguales, para construir un ejemplar E' de X.

- En el VRPTW, se agregan ventanas de tiempo bastante amplias y tiempos de atención cortos a cada cliente, además de una cota de tiempo suficientemente grande.
- Para el HFVRP y el MDVRP, basta preservar las distancias de E, asignándoles valores iguales para todos los vehículos y depósitos.

Si el número de valores asignados está acotado por un polinomio en el tamaño de E, la reducción será polinomial.

Solo queda mostrar que E tiene solución para una cota T si y sólo si E' tiene solución para la misma T y valores adecuados para otras cotas propias de la variante X.

Para lograr esto, primero se supone que existe una solución S para E y se muestra que el mismo conjunto de rutas, junto con valores adecuados para variables adicionales, constituyen una solución válida S' de E'.

- En el caso del VRPTW, se agregan cronogramas simples para cada ruta, los cuales se sabe que son válidos y no exceden la cota de tiempo por la construcción de E' y de los cronogramas.
- En el HFVRP, se asigna un índice de vehículo distinto a cada ruta.
- Para el MDVRP, se especifica el depósito utilizado en cada ruta.

Luego, se supone la existencia de una solución S' de E' y se muestra que el conjunto de rutas utilizado en S', descartando la información de variables adicionales, constituye una solución válida S para E.

- En el VRPTW, la distancia recorrida y el tiempo son independientes, por lo que la cota de distancia se sigue satisfaciendo sin importar los cronogramas.
- Para el HFVRP, por la construcción de E' se sabe que todos los vehículos tienen costos de traslado y capacidades iguales, así que omitir el índice del vehículo en la ruta no afecta la validez de la solución.
- En el caso del MDVRP ocurre algo similar, ya que la construcción de E' garantiza que todos los depósitos tienen igual distancia a cada cliente, así que las rutas siguen siendo válidas aunque se omita el índice de depósito.

De esta forma se tiene, por la construcción de E', que la existencia de S implica la de S' y la existencia de S' implica a su vez la de S, por lo que la reducción es válida y X es NP-difícil.

Finalmente, habiendo mostrado que X está en NP y es NP-difícil, se concluye que X es un problema NP-completo.

Conclusiones

En este trabajo se ha mostrado que la versión de decisión del Problema de Rutas de Vehículos Clásico (CVRP) es un problema NP-completo, como también lo son las versiones de decisión de sus principales variantes: el Problema de Rutas de Vehículos con Ventanas de Tiempo (VRPTW), el Problema de Rutas de Vehículos con Flota Heterogénea (HFVRP) y el Problema de Rutas de Vehículos con Depósitos Múltiples (MDVRP).

Además, se ha descrito un método general en el que se utiliza la técnica de restricción para mostrar que otras variantes con características similares, que pueden verse como generalizaciones de las variantes ya tratadas, se encuentran en la misma categoría de complejidad.

Este método puede utilizarse, como se hizo para generalizar el TSP al CVRP y el CVRP a las otras variantes trabajadas, para generalizar éstas a las variantes mixtas del problema, como el HFVRPTW o incluso el MDHFVRPTW, y mostrar que todas ellas son problemas NP-completos.

Cabe recordar que en el apéndice A se muestra la equivalencia entre el CVRP con y sin tamaño de flota fijo, en el apéndice B se prueba la equivalencia entre el TSP con y sin ciudad inicial fija y en el apéndice C se muestra la equivalencia entre el VRPTW con y sin ventana de tiempo para el depósito.

Considerando las pruebas de los apéndices se tiene que, al ser equivalentes a las variantes utilizadas en este trabajo y que se demostró son problemas NP-completos, las otras versiones mencionadas también están en esa categoría de complejidad.

Dado que estos problemas de decisión son NP-completos, no es práctico abordarlos directamente con algoritmos tradicionales, ya que no existe un algoritmo que los resuelva en tiempo polinomial, a no ser que las clases de problemas P y NP sean iguales.

Dado que las versiones de optimización de los problemas son al menos tan difíciles como las de decisión, el tiempo necesario para solucionarlas es tan grande como el requerido para su versión de decisión, o incluso mayor, por lo que tampoco es posible resolver en tiempo polinomial las versiones de optimización de los problemas abordados en este trabajo.

Por lo tanto es necesario encontrar otras formas de tratar el problema:

Una de esas formas puede ser enfocarse en variantes menos generales, con restricciones adicionales que faciliten el manejo de los datos y el cálculo de la solución óptima.

Un ejemplo de tal variante es el SDP (*Skip Delivery Problem*), el cual puede resolverse en tiempo polinomial para vehículos con capacidad 1 o 2 [1].

Sin embargo, no siempre es posible encontrar una variante menos compleja del problema que se ajuste a un caso particular en la práctica.

Otra opción para abordar el problema es utilizar métodos de aproximación o heurísticas, para obtener buenas soluciones en tiempos razonables, pero debe tenerse claro que estas técnicas no garantizan la obtención de la solución óptima.

Bibliografía

[1] C. Archetti, R. Mansini, and M. G. Speranza. Complexity and reducibility of the skip delivery problem. *Transportation Science*, 39:182–187, 2005.

[2] Claudia Archetti, Martin W. P. Savelsbergh, and M. Grazia Speranza. Worst-case analysis for split delivery vehicle routing problems. *Transportation Science*, 40:226–234, 2006.

[3] G Clarke and J. W. Wright. Scheduling of vehicles from a central depot to a number of delivery points. *Operations Research*, 12(4):568–582, 1964.

[4] Thomas H. Cormen, Charles E. Leiserson, Ronald L. Rivest, and Clifford Stein. *Introduction to algorithms*. McGraw-Hill, Boston, 2001.

[5] Nathalie De Jaegere, Mieke Defraeye, and Inneke Van Nieuwenhyuse. The vehicle routing problem: State of the art classification and review. 2014.

[6] Moshe Dror. Note on the complexity of the shortest path models for column generation in VRPTW. *Operations Research*, 42:977–978, 1994.

[7] Michael R. Garey and David S. Johnson. *Computers and Intractability. A Guide to the Theory of NP-Completeness*. W. H. Freeman and Company, New York, 1979.

[8] Jon Kleinberg and Éva Tardos. *Algorithm Design*. Pearson Education, Inc., 2006.

[9] Udi Manber. *Introduction to Algorithms. A Creative Approach*. Addison-Wesley, Reading, Massachusetts, 1989.

[10] Christos H. Papadimitriou. *Computational Complexity*. Addison Wesley Longman, 1994.

[11] Christos H. Papadimitriou and Kenneth Steiglitz. *Combinatorial Optimization: Algorithms and Complexity*. Dover Publications, Inc., Mineola, New York, 1998.

Apéndices

Apéndice A: equivalencia entre CVRP y CVRP con m fija

A continuación se muestra que la versión modificada del CVRP empleada en este trabajo, en la cual se fija el número m de vehículos a utilizar, es equivalente al CVRP sin modificar:

Sean E un ejemplar del CVRP, T una cota para la distancia, m una cota para el número de vehículos a utilizar y E' un ejemplar del CVRP modificado, igual a E pero con el número de vehículos fijo en m

Se mostrará que existe una solución para E, con distancia total menor o igual a T y utilizando a lo más m vehículos, si y sólo si existe una solución para E', con distancia total menor o igual a T:

Primero supóngase que existe una solución R para E', con $L(R) \leq T$.

Como E' tiene el número de vehículos fijo en m, R deberá constar de m rutas, ya que es una solución válida de E'.

Además, como E' es igual a E con el tamaño de flota fijo, E' tiene el mismo conjunto de clientes, e iguales demandas y distancias entre clientes, por lo que R es aplicable a E y la distancia total recorrida será la misma $L(R)$.

Así, como R es una solución para E que consta de m rutas, con $L(R) \leq T$, se tiene que existe una solución para E, con distancia total menor o igual a T y utilizando a lo más m vehículos.

$\square$

Supóngase ahora que existe una solución R para E, con $L(R) \leq T$ y con a lo más m rutas.

Sean $a \leq m$ el número de rutas utilizadas en R y $b = m - a$ el número de rutas que le faltan a R para alcanzar las m rutas que requiere una solución de E'.

Se define una ruta vacía como una ruta r que comienza y termina en el depósito c_0 y no pasa por ningún cliente; es decir, una ruta r de la forma

$$r = [c_0, c_0]$$

Las rutas vacías siguen cumpliendo las características de una ruta; es decir, comienzan y terminan en el depósito, no pasan por un cliente más de una vez y no sobrepasan la capacidad de carga del vehículo. Además, no aumentan la distancia total recorrida, ya que $d_{00} = 0$.

Por lo tanto, la solución R' obtenida al tomar R y agregar b rutas vacías, seguirá siendo una solución válida para E.

Como R' consiste de las a rutas de R y b rutas vacías, en total tiene $a + b = m$ rutas, y por tanto es una solución válida para E'.

Además, como las rutas vacías no aumentan la distancia total recorrida, se tiene que $L(R') = L(R) \leq T$.

De esta forma se tiene que existe una solución R' para E', con distancia total menor o igual a T.

$\square$

Por lo tanto, existe una solución para E, con distancia total menor o igual a T y utilizando a lo más m vehículos, si y sólo si existe una solución para E', con distancia total menor o igual a T.

Por lo tanto, el CVRP y el CVRP con tamaño de flota fijo son equivalentes.

$\blacksquare$

Apéndice B: equivalencia entre TSP y TSP con ciudad inicial fija

A continuación se muestra que la versión modificada del TSP utilizada en este trabajo, en la cual se fija la ciudad inicial del ciclo, es equivalente al TSP sin modificar:

Sean E un ejemplar del TSP con n ciudades, T una cota para la distancia total recorrida, y E' un ejemplar del TSP modificado, igual a E pero con la ciudad inicial fija en c_1.

Se mostrará que existe una solución para E, con distancia total menor o igual a T, si y sólo si existe una solución para E', con distancia total menor o igual a T.

Primero supóngase que existe una solución r para E', con $l(r) \leq T$.

Como r es un ciclo que pasa por todas las ciudades exactamente una vez y regresa a la ciudad inicial c_1, recorriendo en total una distancia $l(r) \leq T$, r es una solución para E, con distancia total recorrida menor o igual a T.

Así, existe una solución para E, con distancia total menor o igual a T.

$\square$

Supóngase ahora que existe una solución r para E, con $l(r) \leq T$.

Se sabe que r es un ciclo que comienza en alguna ciudad c_x, pasa por todas las demás ciudades, exactamente una vez, y regresa a c_x.

Si $c_x = c_1$, se tendrá que r es una solución para E' con $l(r) \leq T$.

Supóngase entonces que $c_x \neq c_1$, y que c_1 es la i-ésima ciudad en r, se construye un nuevo ciclo r' como sigue:

El ciclo r' comienza en c_1, recorre las últimas $n-i$ ciudades de r en el mismo orden que r y luego recorre las primeras i ciudades de r en el mismo orden, terminando en c_1; es decir, si r es de la forma

$$r = [c_x, c_{x_2}, c_{x_3} ..., c_{x_{i-1}}, c_1, c_{x_{i+1}}, c_{x_{i+2}}, ..., c_{x_n}, c_x],$$

entonces r' será de la forma:

$$r' = [c_1, c_{x_{i+1}}, c_{x_{i+2}}, ..., c_{x_n}, c_x, c_{x_2}, c_{x_3} ..., c_{x_{i-1}}, c_1].$$

Así, el ciclo r' tiene los mismos elementos que r, pero c_x solamente aparece una vez y c_1 aparece dos veces; es decir, r' es una lista en la que la ciudad inicial y final son la misma, y todas las otras ciudades aparecen exactamente una vez.

Por lo tanto, r' es una ruta cíclica que define correctamente una solución para E y para E'.

Además, si se denota c_x como c_{x_1} y c_1 como c_{x_i}, se tiene que:

$$l(r) = d_{x_1x_2} + d_{x_2x_3} + ... + d_{x_{i-1}x_i} + d_{x_ix_{i+1}} + d_{x_{i+1}x_{i+2}} + ... + d_{x_{n-1}x_n} + d_{x_nx_1}$$
$$\Rightarrow l(r) = \sum_{j=1}^{n-1}(d_{x_jx_{j+1}}) + d_{x_nx_1}$$

y, por otro lado, se tiene que:

$$l(r') = d_{x_ix_{i+1}} + d_{x_{i+1}x_{i+2}} + ... + d_{x_{n-1}x_n} + d_{x_nx_1} + d_{x_1x_2} + d_{x_2x_3} + ... + d_{x_{i-1}x_i}$$
$$\Rightarrow l(r) = \sum_{j=i}^{n-1}(d_{x_jx_{j+1}}) + d_{x_nx_1} + \sum_{j=1}^{i-1}(d_{x_jx_{j+1}})$$
$$= \sum_{j=1}^{i-1}(d_{x_jx_{j+1}}) + \sum_{j=i}^{n-1}(d_{x_jx_{j+1}}) + d_{x_nx_1}$$
$$= \sum_{j=1}^{n-1}(d_{x_jx_{j+1}}) + d_{x_nx_1} = l(r)$$

Entonces, r' es una solución para E' y $l(r') = l(r) \leq T$.
De esta forma se tiene que existe una solución r' para E', con distancia total menor o igual a T.

$\square$

Por lo tanto, existe una solución para E, con distancia total menor o igual a T, si y sólo si existe una solución para E', con distancia total menor o igual a T.

Por lo tanto, el TSP y el TSP con ciudad inicial fija son equivalentes.

$\blacksquare$

Apéndice C: equivalencia entre VRPTW con y sin ventana de tiempo para el depósito

A continuación se muestra que la versión del VRPTW utilizada en este trabajo, en la cual no se considera ventana de tiempo para el depósito, es equivalente al VRPTW con ventana de tiempo en el depósito:

Sean E un ejemplar del VRPTW con ventana de tiempo para el depósito, $[y_0, z_0]$ la ventana de tiempo del depósito, T una cota para la distancia, U una cota para el tiempo y E' un ejemplar del VRPTW sin ventana de tiempo para el depósito, igual a E a excepción de que todas las ventanas de tiempo están anticipadas y_0 unidades; es decir:

$$y'_i = y_i - y_0 \text{ y } z'_i = z_i - z_0,\ \forall i,\ i = 1, 2, ..., n.$$

Se mostrará que existe una solución para E, con distancia total menor o igual a T y tiempo total menor o igual a U, si y sólo si existe una solución para E' con la misma cotas de distancia T y con tiempo total requerido menor o igual a una cota $U' = \text{mín}\{U, z_0 - y_0\}$:

Primero supóngase que existe una solución (R, S') para E', con $L(R) \leq T$ y $L(S') \leq U'$.

Como E' tiene el mismo conjunto de clientes, e iguales demandas y distancias entre clientes que E y dado que R es independiente de las ventanas de tiempo, R es aplicable a E y la distancia total recorrida será la misma $L(R)$.

Sea $S = \{s_1, s_2, ..., s_m\}$ un cronograma igual a S' pero postergado y_0 unidades de tiempo; es decir,

$$s_i = [p_0, p_1, ..., p_{q_i}, p_{q_i+1}],\ \forall i,\ i = 1, 2, ..., m,$$
$$\text{donde } p_j = p'_j + y_0,\ \forall j,\ j = 0, 1, ..., q_i, q_i + 1.$$

Como S' es un conjunto de cronogramas válido para E', satisface las ventanas de tiempo de los clientes; es decir, para cada ruta $r_i \in R,\ r_i = [c_0, c_{x_1}, c_{x_1}, ..., c_{x_{q_i}}, c_0]$, el cronograma correspondiente de S', s'_i, cumple que

$$p'_j + t_{x_j x_{j+1}} < p'_{j+1},\ \forall j,\ j = 0, 2, ..., q_i$$

y que

$$y'_{x_j} < p'_j < p'_j + t_{x_j x_j} < z'_{x_j},\ \forall j,\ j = 1, 2, ..., q_i.$$

De donde, como las ventanas de tiempo en E' fueron anticipadas y los cronogramas en S postergados, se tiene que

$$s_i = [p'_0 + y_0, p'_1 + y_0, ..., p'_{q_i} + y_0, p'_{q_i+1} + y_0],\ \forall i,\ i = 1, 2, ..., m,$$

por lo que se tienen las siguientes propiedades:

- $p'_j + t_{x_j x_{j+1}} < p'_{j+1},\ \forall j, j = 0, 2, ..., q_i$
  $\Rightarrow (p'_j + y_0) + t_{x_j x_{j+1}} < (p'_{j+1} + y_0)$
  $\Rightarrow p_j + t_{x_j x_{j+1}} < p_{j+1},\ \forall j, j = 0, 2, ..., q_i$
- $y'_{x_j} < p'_j < p'_j + t_{x_j x_j} < z'_{x_j},\ \forall j,\ j = 1, 2, ..., q_i$
  $\Rightarrow (y'_{x_j} + y_0) < (p'_j + y_0) < (p'_j + y_0) + t_{x_j x_j} < (z'_{x_j} + y_0)$
  $\Rightarrow ((y_{x_j} - y_0) + y_0) < p_j < p_j + t_{x_j x_j} < ((z_{x_j} - y_0) + y_0)$
  $\Rightarrow y_{x_j} < p_j < p_j + t_{x_j x_j} < z_{x_j},\ \forall j,\ j = 1, 2, ..., q_i$

Por lo tanto, S satisface las ventanas de tiempo de los clientes en E.

Por otro lado, nótese que $L(S')$ representa, por la forma en que se definió, el momento en el que el último vehículo regresa al depósito, por lo que para obtener el tiempo total transcurrido hasta ese momento, hace falta restarle el momento inicial.

Sea ω' el momento final dado por $L(S')$.

Como para E' el momento inicial es cero, el tiempo total requerido por S' es $\omega' - 0 = L(S')$. Sin embargo, como los cronogramas de S se postergaron por y_0 unidades, el momento inicial para S es $0 + y_0 = y_0$ y el momento final para S es $\omega = \omega' + y_0$, por lo que el tiempo total requerido por S es:

$$L(S) = \omega - y_0 = \omega' + y_0 - y_0 = \omega' = L(S').$$

Así, como el momento inicial de S es y_0, ninguno de los cronogramas de S tendrá acciones previas al inicio de la ventana de tiempo del depósito. Además, como $L(S) = L(S') \leq U' \leq z_0 - y_0$, la última acción realizada en cualquiera de los cronogramas de S será, como máximo, en el momento z_0. Por lo tanto, en los cronogramas de S en ningún momento se operará fuera de la ventana de tiempo del depósito.

De esta forma se tiene que S es un conjunto de cronogramas válido para E y, por consiguiente, (R, S) es una solución válida para E.

Por lo tanto, como (R, S) es una solución para E, con $L(R) \leq T$ y $L(S) \leq U' \leq U$, se tiene que existe una solución para E, con distancia total menor o igual a T tiempo total menor o igual a U.

$\square$

Supóngase ahora que existe una solución (R, S) para E, con $L(R) \leq T$ y $L(S) \leq U$.

Sea $S' = \{s'_1, s'_2, ..., s'_m\}$ un cronograma igual a S pero anticipado y_0 unidades de tiempo; es decir,

$$s'_i = [p'_0, p'_1, ..., p'_{q_i}, p'_{q_i+1}],\ \forall i,\ i = 1, 2, ..., m,$$
$$\text{donde } p'_j = p_j - y_0,\ \forall j,\ j = 0, 1, ..., q_i, q_i + 1.$$

Como S es un conjunto de cronogramas válido para E, satisface las ventanas de tiempo de los clientes; es decir, para cada ruta $r_i \in R,\ r_i = [c_0, c_{x_1}, c_{x_1}, ..., c_{x_{q_i}}, c_0]$, el cronograma correspondiente de S, s_i, cumple que

$$p_j + t_{x_j x_{j+1}} < p_{j+1} \quad \forall j, j = 0, 2, ..., q_i$$

y que

$$y_{x_j} < p_j < p_j + t_{x_j x_j} < z_{x_j} \quad \forall j, j = 1, 2, ..., q_i.$$

Por lo tanto, como tanto las ventanas de tiempo en E' como los cronogramas en S' fueron anticipados, se tiene que

$$s'_i = [p_0 - y_0, p_1 - y_0, ..., p_{q_i} - y_0, p_{q_i+1} - y_0],\ \forall i,\ i = 1, 2, ..., m,$$

de donde se derivan las siguientes propiedades:

- $p_j + t_{x_j x_{j+1}} < p_{j+1},\ \forall j,\ j = 0, 2, ..., q_i$
  $\Rightarrow (p_j - y_0) + t_{x_j x_{j+1}} < (p_{j+1} - y_0)$
  $\Rightarrow p'_j + t_{x_j x_{j+1}} < p'_{j+1},\ \forall j,\ j = 0, 2, ..., q_i$

- $y_{x_j} < p_j < p_j + t_{x_jx_j} < z_{x_j},\ \forall j,\ j = 1, 2, ..., q_i$
  $\Rightarrow (y_{x_j} - y_0) < (p_j - y_0) < (p_j - y_0) + t_{x_jx_j} < (z_{x_j} - y_0)$
  $\Rightarrow (y'_{x_j} < p'_j < p'_j + t_{x_jx_j} < z'_{x_j},\ \forall j,\ j = 1, 2, ..., q_i$

Por lo tanto, S' satisface las ventanas de tiempo de los clientes en E'.

Por otro lado, sean $\omega \leq z_0$ el momento en el que el último vehículo regresa al depósito según S y $U' = \text{mín}\{U, z_0 - y_0\}$ una cota de tiempo.

Como el momento inicial de S es y_0, ninguno de los cronogramas de S tendrá acciones previas a ese momento, por lo que ninguno de los cronogramas en S' tendrá acciones previas al momento $y_0 - y_0 = 0$.

Además, como S es un conjunto de cronogramas válido para E, se tiene que el tiempo total requerido por S, $L(S) = \omega - y_0$, es menor o igual a la cota de tiempo U y que ningún cronograma en S tendrá acciones posteriores al momento ω, por lo que ningún cronograma de S' tendrá acciones posteriores al momento $\omega' = \omega - y_0 \leq z_0 - y_0$.

Así, la última acción realizada en cualquiera de los cronogramas de S será, como máximo, en el momento ω', de donde se obtiene que el tiempo total requerido para S', $L(S')$, cumple que:

$$L(S') = \omega' - 0 = \omega' = \omega - y_0 \leq U \text{ y } L(S') = \omega' - 0 = \omega' \leq y_0 - z_0;$$

es decir, $L(S') = \omega' \leq U'$.

De esta forma se tiene que S' es un conjunto de cronogramas válido para E', con tiempo total requerido menor o igual a la cota U', por lo que (R, S) es una solución válida para E.

Así, como (R, S') es una solución para E', con $L(R) \leq T$ y $L(S') \leq U'$, se tiene que existe una solución para E', con distancia total menor o igual a T y tiempo total menor o igual a U'.

□

Por lo tanto, existe una solución para E con distancia total menor o igual a T y tiempo total menor o igual a U, si y sólo si existe una solución para E' con distancia total menor o igual a la misma T y tiempo total menor o igual a $U' = \text{mín}\{U, y_0 - z_0\}$; es decir, los dos problemas, el VRPTW con ventana de tiempo para el depósito y el VRPTW sin ventana de tiempo para el depósito, son equivalentes.

■