

Los modelos de optimización requieren de un conjunto de datos que represente características de la solución que se posee. Estos datos son los costos asociados a una subruta o un cliente, los cuales garantizan la calidad y viabilidad de las soluciones obtenidas.

Existen vecindades que presentan una gran cantidad de soluciones posibles, por lo que realizar una exploración exhaustiva para generar todos los datos necesarios implica un alto costo computacional. Por ello, generar los datos de manera eficiente facilita la exploración del espacio de soluciones sin comprometer la calidad de los resultados obtenidos.

En este capítulo se abordan alternativas para la generación eficiente de los datos necesarios en los modelos de optimización. En la sección 1 se introduce un criterio de vecindad que consiste en la extracción de dos clientes de sus posiciones actuales en la solución y su posterior reinserción en diferentes ubicaciones. La exploración exhaustiva de las soluciones de este criterio para la generación de los datos, es del orden de $O(n^4)$ (siendo n la cantidad de clientes del problema) por lo que en la sección 2 se describe una forma de calcular los parámetros para esta vecindad en un orden menor al que se tuviera explorándola en su totalidad.

1. Análisis de la vecindad reinserción de dos clientes

El criterio de vecindad basado en la extracción y reinserción de dos clientes de sus posiciones actuales en la solución, es un criterio que produce una gran cantidad de nuevas soluciones. Este procedimiento consiste en seleccionar dos clientes, eliminarlos de sus respectivas rutas y reubicarlos en nuevas posiciones dentro de la misma o en otras rutas.

Este proceso da lugar a un espacio de búsqueda considerablemente grande, cuya complejidad depende del número de clientes que tenga el problema, debido a que determina la cantidad de combinaciones posibles para seleccionar dos clientes (j_1, j_2) a reubicar. Al seleccionar dos clientes de un conjunto de n , el número de combinaciones crece de manera cuadrática, ya que se consideran todas las parejas posibles de clientes. Este crecimiento se puede expresar mediante la siguiente fórmula:

$$\binom{n}{2} = \frac{n \cdot (n - 1)}{2}$$

Esta expresión muestra como el número de combinaciones aumenta con el tamaño del conjunto de clientes en orden de $O(n^2)$. Por ejemplo para un problema que posee 10 clientes, la cantidad total de combinaciones sería 45, sin embargo para uno de 20 serían 190.

La reinserción de los clientes seleccionados introduce una nueva capa de complejidad. Cada cliente puede ser reubicado en cualquier posición dentro de cualquier ruta. Esto implica que, para modelar adecuadamente las posibles ubicaciones de inserción, es necesario identificar todas las posiciones válidas donde un cliente puede ser colocado, las cuales son $n + r$, donde r es la cantidad de

rutas de la solución y n la cantidad de clientes. Demostremos el origen de esta cantidad.

Como el CVRP puede expresarse como un grafo, se puede establecer una correspondencia entre las aristas del grafo y las posiciones que puede tomar un cliente.

Supongamos que queremos reinsertar el cliente c_x en otra posición. La inserción de c_x en una ruta consiste en identificar dos vértices consecutivos c_i y c_j en la ruta, donde $c_i \neq c_x$ y $c_j \neq c_x$, y dividir la arista $\langle c_i, c_j \rangle$ para formar dos aristas $\langle c_i, c_x \rangle$ y $\langle c_x, c_j \rangle$. Entonces cada arco $e = \langle c_i, c_j \rangle$ en una ruta representa una posición válida para insertar un cliente. En caso de que $e = \langle c_i, c_x \rangle$ o $e = \langle c_x, c_j \rangle$, también se consideran posiciones válidas, solo que no se dividen las aristas, dado que al extraer c_x las dos aristas vuelven a ser el arco $\langle c_i, c_j \rangle$, en ese caso c_x es reinsertado en su misma posición.

Si una solución posee r rutas y cada una posee m clientes, entonces el número de aristas en esa ruta es $m + 1$, debido a que se forma un ciclo que empieza y termina en el depósito. Como cada cliente se puede reinsertar en cualquier ruta la cantidad de posiciones posibles para insertar es:

$$\sum_{k=1}^r m_k + 1,$$

donde m_k representa el número de clientes en la k -ésima ruta. Como la suma de los clientes de todas las rutas es igual al total de clientes en la solución (n), y cada ruta incluye exactamente una arista adicional, podemos reescribir la suma como:

$$\sum_{k=1}^r m_k + 1 = n + r.$$

Esto asegura que estamos considerando todas las posibles aristas en la solución original.

Por lo tanto, dado que el número de aristas en el grafo del CVRP es $n + r$, la cantidad total de posibles posiciones para insertar cada cliente también es $n + r$. Además, considerando que la selección de los pares de clientes está en el orden de $O(n^2)$ y que evaluar todas las posiciones de inserción para cada cliente tiene una complejidad de $O(n^2)$, se obtiene una complejidad total de $O(n^4)$. Esto significa que, para una solución con 10 clientes, podrían generarse hasta 10,000 vecinos en el peor de los casos. Esto hace que realizar una exploración exhaustiva para generar todos los datos sea computacionalmente inviable.

Por lo tanto, en la siguiente sección se abordarán técnicas específicas para reducir la complejidad computacional asociada con la generación de datos en esta vecindad. Esto disminuye el tiempo requerido para la generación de datos y mejora la eficiencia global del modelo de optimización.

2. Generación Eficiente de Datos para la Vecindad de Reinserción de Dos Clientes

Para generar los datos de la vecindad basada en la extracción y reinserción de dos clientes, es necesario dividir la vecindad en cuatro escenarios específicos que permitan calcular los parámetros de manera más eficiente. Cada uno de estos escenarios presenta características particulares que afectan tanto la complejidad computacional de su exploración como las técnicas utilizadas para optimizar la generación de datos.

Estos escenarios se agrupan en los siguientes casos:

1. **Extracción e inserción separada:** Ambos clientes son extraídos e insertados en posiciones distintas.
2. **Extracción conjunta e inserción separada:** Se extraen pares de clientes consecutivos, y son insertados en posiciones separadas.
3. **Extracción separada e inserción conjunta:** Cada cliente se extrae de forma independiente, pero se insertan juntos en una misma posición.
4. **Extracción e inserción conjunta:** Ambos clientes se tratan como una subruta, tanto durante su extracción como en su reinserción.

Separar la vecindad en estos cuatro escenarios permite generar todos los datos en un orden menor a como lo haríamos con una exploración exhaustiva. Tanto la extracción como la inserción descritas en esos escenarios no modifican la solución actual, de esta manera se evitan los costos que conlleva la destrucción y la reparación de la solución.

En las siguientes subsecciones, se presentan técnicas específicas para abordar cada una de estos escenarios dentro de la vecindad de reinserción de dos clientes, con el objetivo de minimizar los costos computacionales sin comprometer la calidad de las soluciones obtenidas. En los distintos escenarios se hace una diferenciación entre arista y posición, nos referiremos a posición cuando un cliente pueda ser ubicado en ella y a arista para mostrar como se calculan los datos. En la figura 1 se muestra una solución al CVRP con 5 clientes y 2 rutas, esta solución se usará como base para los ejemplos de cada escenario.

$$\begin{aligned} r_1 : 0 &\rightarrow 2 \rightarrow 4 \rightarrow 9 \rightarrow 6 \rightarrow 0 \\ r_2 : 0 &\rightarrow 3 \rightarrow 5 \rightarrow 1 \rightarrow 7 \rightarrow 10 \rightarrow 8 \rightarrow 0 \end{aligned}$$

Figura 1: Grafo de solución de un CVRP de 10 clientes y 2 rutas

2.1. Extracción e inserción separadas

Este es el mas básico de los escenarios, y el que menor complejidad computacional presenta, debido a que los eventos de inserción y extracción del cliente

c_1 no afectan a los del cliente c_2 . De esta forma podemos considerar realizar solo el cálculo para cada cliente, en lugar de tomar las combinaciones de los mismos.

El coste asociado a la extracción de un cliente c_i de una ruta, donde c_i es el i -ésimo cliente de la ruta, se puede calcular como la suma de los costos de los arcos $\langle c_{i-1}, c_i \rangle$, $\langle c_i, c_{i+1} \rangle$ menos el costo del arco $\langle c_{i-1}, c_{i+1} \rangle$, debido a que, al extraer el cliente c_i los arcos asociados a él desaparecen, y para conectar nuevamente el grafo se crea el arco entre los clientes antecesor c_{i-1} y el sucesor c_{i+1} . Como esta operación se realiza para cada cliente, el costo total de la misma es $O(n)$.

Para calcular el costo de insertar un cliente c_i en una posición p solo es necesario calcular el costo de las aristas $\langle c_p, c_i \rangle$ y $\langle c_i, c_{p+1} \rangle$, donde c_p y c_{p+1} son los clientes que conforman el arco correspondiente a la posición p . En caso de que $c_i = c_p$ o $c_i = c_{p+1}$, el costo de insertar es cero debido a que ocupará la misma posición en la que se encontraba el cliente c_i .

La operación de inserción considera a cada cliente en cada posición del grafo por lo tanto su complejidad computacional sería $O(n(n+k))$ o lo que es lo mismo $O(n^2 + nk)$, como la cantidad de rutas en un CVRP usualmente es mucho más pequeña que la cantidad de clientes entonces la complejidad sería de $O(n^2)$ por la ley de la suma.

Ahora para calcular el costo total asociado a mover un cliente, se utiliza el valor obtenido de su extracción e inserción, es decir:

$$TotalCost_i = insert_i - extract_i.$$

Si el movimiento de un cliente consiste en una mejora, entonces $TotalCost_i < 0$, en el caso de que se quiera insertar en alguna de las dos posiciones en de las que ese cliente forma parte $TotalCost_i = 0$.

Veamos un ejemplo, donde se desean mover reinsertar los clientes c_1 y c_2 en la posición 2 de la ruta 1 y la 5 de la ruta 2, de la solución presente en la figura 1.

$$\begin{aligned} r_1 : 0 &\rightarrow 4 \rightarrow 1 \rightarrow 9 \rightarrow 6 \rightarrow 0 \\ r_2 : 0 &\rightarrow 3 \rightarrow 5 \rightarrow 7 \rightarrow 10 \rightarrow 2 \rightarrow 8 \rightarrow 0 \end{aligned}$$

Figura 2: Ejemplo de reinsertión de los clientes c_1 y c_2

Como se muestra en la figura 2, los clientes no son extraídos ni reinsertados juntos, por lo que el cálculo de uno no afecta al otro, sin embargo el querer intercambiarlos entre sí puede generar problemas debido a que no se esta modificando la solución actual, por tanto se presenta la siguiente alternativa para cuando los clientes vayan a ser intercambiados.

Se selecciona cada par de clientes no consecutivos, en las siguientes secciones se analizarán el resto de los casos. En el epígrafe anterior ya verificamos que esta selección posee una complejidad computacional de $O(n^2)$, y en lugar de insertarlos en cada posición, solo los insertaríamos en las posiciones problemáticas

es decir, en las posiciones que desaparecerían al extraerlos. El total de estas posiciones es 4, debido a que cada uno posee solo dos arcos ($\langle c_{i-1}, c_i \rangle$ y $\langle c_i, c_{i+1} \rangle$). Para calcular entonces el costo del intercambio entre c_i y c_j es necesario solo calcular la distancia de los arcos $\langle c_{i-1}, c_j \rangle$ y $\langle c_j, c_{i+1} \rangle$, de igual forma para ubicar c_i . El costo de extracción de los clientes se mantiene igual debido a que no son clientes consecutivos. Por tanto la complejidad de este enfoque es $O(n^2) \times O(4)$ que es lo mismo que $O(n^2)$.

Aunque este escenario modela la mayoría de los casos, puede que existan pares de clientes que necesitan ser separados debido al alto costo que conlleva mantenerlos juntos. En la siguiente sección se profundiza sobre este escenario.

2.2. Extracción conjunta e inserción separada

En este escenario se extraen pares de clientes consecutivos para luego ser insertados en posiciones diferentes uno del otro. A diferencia del anterior, al remover un cliente se afecta al siguiente (o anterior), por lo que se debe modelar una nueva forma de extracción. Para la inserción se puede considerar usar los datos obtenidos en la subsección anterior, con una modificación en el momento de insertar en alguna de las posiciones que desaparecen al ser removidos.

En la figura 3, se muestra paso a paso el proceso de extracción para los clientes 4 y 9 de la solución de la figura 1 y como se modifica el valor total de la ruta en cada caso.

1. $r_1 : 0 \xrightarrow{2} 2 \xrightarrow{8} 4 \xrightarrow{4} 9 \xrightarrow{5} 6 \xrightarrow{2} 0 \text{ value} = 21$
 $r_2 : 0 \xrightarrow{3} 3 \xrightarrow{1} 5 \xrightarrow{6} 1 \xrightarrow{3} 7 \xrightarrow{1} 10 \xrightarrow{4} 8 \xrightarrow{7} 0 \text{ value} = 25$
2. $r_1 : 0 \xrightarrow{2} 2 \xrightarrow{15} 9 \xrightarrow{5} 6 \xrightarrow{2} 0 \text{ value} = 24$
 $r_2 : 0 \xrightarrow{3} 3 \xrightarrow{1} 5 \xrightarrow{6} 1 \xrightarrow{3} 7 \xrightarrow{1} 10 \xrightarrow{4} 8 \xrightarrow{7} 0 \text{ value} = 25$
3. $r_1 : 0 \xrightarrow{2} 2 \xrightarrow{6} 6 \xrightarrow{2} 0 \text{ value} = 10$
 $r_2 : 0 \xrightarrow{3} 3 \xrightarrow{1} 5 \xrightarrow{6} 1 \xrightarrow{3} 7 \xrightarrow{1} 10 \xrightarrow{4} 8 \xrightarrow{7} 0 \text{ value} = 25$

Figura 3: Ejemplo de extracción conjunta para la solución de la figura 1

Como se puede apreciar en la figura 3 el valor final después de la extracción de ambos clientes es el mismo que si se hubiesen extraído los dos en conjunto, esto se debe a que la arista entre el cliente 2 y el cliente 9 desaparece luego de que este último sea removido. Por tanto para calcular el coste de extracción de dos clientes consecutivos c_i, c_{i+1} en una solución del CVRP, se calcula el costo de eliminar las aristas $\langle c_{i-1}, c_i \rangle$, $\langle c_i, c_{i+1} \rangle$, $\langle c_{i+1}, c_{i+2} \rangle$, y de añadir la arista $\langle c_{i-1}, c_{i+2} \rangle$. La complejidad computacional de realizar todo este cálculo es de $O(n)$ porque evaluamos una cantidad fija de operaciones para cada uno de los n clientes en la solución.

Para la inserción de los clientes se puede aprovechar los cálculos realizados en la subsección anterior, realizando un ajuste al momento de reinsertarlos en su posición original. Supongamos que en nuestra solución de CVRP existe una solución con una ruta representada de la siguiente manera:

$$\dots \rightarrow c_{i-1} \xrightarrow{1} c_i \xrightarrow{2} c_{i+1} \xrightarrow{3} c_{i+2} \rightarrow \dots$$

En esta ruta se quieren reinsertar los clientes c_i, c_{i+1} en alguna de las posiciones marcadas como 1, 2 o 3. En caso de que a lo sumo uno de los dos clientes (c_i o c_{i+1}) se inserta en una de las tres posiciones, dicho cliente se mantiene en esa posición, mientras que el otro se inserta fuera de las mismas, en este caso basta con utilizar los datos generados previamente para saber a variación del costo, debido a que podemos utilizar el valor ya calculado del cliente que es reubicado fuera de estas tres posiciones.

En el caso de que ambos clientes son insertados en una de las tres posiciones entonces pueden suceder dos variantes, la primera es que la posición en la que será insertado c_{i+1} sea menor que la posición en la que se ubicará el cliente c_i ($p_{c_{i+1}} < p_{c_i}$) o que la posición de c_{i+1} sea mayor que la de c_i ($p_{c_{i+1}} > p_{c_i}$). Para calcular el costo en el primer caso, basta con utilizar los datos calculados previamente, debido a que este es el costo total de insertar el cliente c_i en la posición 3, o al cliente c_{i+1} en la posición 1. Para el segundo caso el valor del costo total no varía dado que se insertan en el mismo orden que poseen en la solución.

Por tanto el costo total para estos datos se puede modelar de manera general con la ecuación:

$$TotalCost_{i,j} = insert_i + insert_j - extract_{i,j}$$

Si la extracción conlleva a una mejora entonces $TotalCost_{i,j} < 0$, en el caso que ambos clientes se inserten con su orden en su posición original $TotalCost_{i,j} = 0$, y en el caso de que solo uno se mantenga en su posición, se utiliza los datos ya obtenidos.

El costo computacional que conlleva la generación de estos datos es de $O(n^3)$, debido a que el costo de extraer los pares de clientes es $O(n)$ y luego para cada cliente en cada par, ubicarlos en todas las posiciones posibles, lo que posee una complejidad de $O(n^2)$, entonces $O(n^2) \times O(n) = O(n^3)$.

Existen pares de clientes que debido a sus características como la distancia entre ellos, son buenas opciones a estar unidos en una solución, sin embargo la forma en la que se generan los datos en estas dos secciones, no modelan esta característica. En la próxima subsección se profundiza en las técnicas para el cálculo de este escenario.

2.3. Extracción separada e inserción conjunta

En este escenario los clientes que se extraen no poseen relación entre ellos, por lo que no son clientes consecutivos, aunque posteriormente serán insertados en la misma posición. Con esta estrategia se permite analizar los casos en que dos

clientes mejoren su costo al ser insertados juntos. Como el proceso de extracción de cada cliente no posee relación entre sí, se puede aprovechar los datos obtenidos en la sección 1.

Para calcular los datos de la inserción conjunta de dos clientes (c_i, c_j) , es necesario separar el proceso de ubicación en dos casos. El primer caso es cuando la posición en la que se van a insertar corresponde a alguna de las aristas directamente conectadas a los clientes a insertar, es decir, $\langle c_{i-1}, c_i \rangle$, $\langle c_i, c_{i+1} \rangle$, $\langle c_{j-1}, c_j \rangle$ o $\langle c_j, c_{j+1} \rangle$. El segundo caso es cuando la posición a insertar no corresponde a ninguna de las del primer caso. Como los clientes no se ubican al mismo tiempo sino que el primero que se extrae es el primero en insertarse, por cada posición en la que se pueda insertar hay que considerar el caso en que se inserten c_i, c_j y el caso en que se inserten c_j, c_i .

Para generar los datos del primer caso, se identifica a cuál de los clientes corresponde la arista seleccionada, supongamos que le pertenece a c_i . Una vez determinada esta relación, se utilizan los datos previamente generados para ubicar al cliente c_j en la posición correspondiente, esto sucede debido a que c_i sera reinsertado en su misma posición. Para modelar el caso que falta para esta posición, se utiliza el dato de insertar el cliente c_j en la posición correspondiente a la otra arista de c_i . El mismo enfoque se utiliza para insertar c_i .

En el segundo caso, el coste de inserción de los clientes c_i, c_j en la posición p , se puede calcular como el coste de añadir las aristas $\langle c_p, c_i \rangle$, $\langle c_i, c_j \rangle$, $\langle c_j, c_{p+1} \rangle$, donde c_p, c_{p+1} son los clientes que conforman la arista de la posición p , y eliminando la arista correspondiente a la posición. Para modelar el otro enfoque, se alternan entre c_i y c_j .

A partir de la solución de la figura 1, evaluaremos el proceso de extracción e inserción conjunta para diferentes pares de clientes.

Supongamos que queremos insertar conjuntamente los clientes c_4 y c_9 en una posición correspondiente a las aristas directamente conectadas a ellos. Si seleccionamos la arista $\langle 2, 4 \rangle$, el cliente c_4 mantiene su posición original y c_9 se inserta inmediatamente después de 2, dando como resultado la ruta:

$$r_1 : 0 \rightarrow 2 \rightarrow 9 \rightarrow 4 \rightarrow 6 \rightarrow 0$$

Alternativamente, c_9 puede insertarse después de 4, resultando en:

$$r_1 : 0 \rightarrow 2 \rightarrow 4 \rightarrow 9 \rightarrow 6 \rightarrow 0$$

Ahora consideremos insertar conjuntamente c_4 y c_9 en una posición arbitraria en la ruta r_2 . Por ejemplo, entre los clientes 5 y 1. El resultado de insertar c_4 seguido de c_9 sería:

$$r_2 : 0 \rightarrow 3 \rightarrow 5 \rightarrow 4 \rightarrow 9 \rightarrow 1 \rightarrow 7 \rightarrow 10 \rightarrow 8 \rightarrow 0$$

Si alternamos el orden y primero insertamos c_9 seguido de c_4 , obtenemos:

$$r_2 : 0 \rightarrow 3 \rightarrow 5 \rightarrow 9 \rightarrow 4 \rightarrow 1 \rightarrow 7 \rightarrow 10 \rightarrow 8 \rightarrow 0$$

Estos ejemplos permiten visualizar cómo se calculan y modelan los diferentes escenarios de inserción conjunta en una solución de CVRP.

Cada dato se puede generar mediante la formula:

$$TotalCost_{i,j} = insert_{i,j} - extract_i - extract_j$$

Si los clientes son una buena opción a unir entonces $TotalCost_{i,j} < 0$.

La complejidad temporal total de las operaciones en este escenario es $O(n^3)$ debido a que se seleccionan todas las combinaciones de clientes, lo cual posee una complejidad de $O(n^2)$, y por cada combinación analizan si se colocaran en cada posición que se encuentran en el orden de $O(n)$, por lo que el costo total seria de $O(n^2) \times O(n) = O(n^3)$.

Hasta este punto los escenarios contemplan la mayoría de los casos en los que se puede extraer e insertar dos clientes, sin embargo aun no se modela el escenario en que se extraen clientes consecutivos y se insertan en la misma posición. En la siguiente subsección se profundiza en este escenario.

2.4. Extracción e inserción conjunta

Para generar los datos en este escenario, se utilizan la misma idea de extracción e inserción conjunta visto en las etapas anteriores. La diferencia entre este y el resto es que aquí existen tres posiciones de interés para la inserción.

Sean los clientes c_i, c_{i+1} los clientes de una ruta de una solución representada de la siguiente forma:

$$\dots \rightarrow c_{i-1} \xrightarrow{1} c_i \xrightarrow{2} c_{i+1} \xrightarrow{3} c_{i+2} \rightarrow \dots$$

Este escenario considera el caso en que la posición donde será insertado c_{i+1} es igual a la posición en la que se ubicará el cliente c_i ($p_{c_{i+1}} = p_{c_i}$). En caso de que se seleccione alguna de las posiciones 1,2 o 3, la solución se mantiene igual y este dato sera igual a 0. Como se debe modelar el caso en que c_{i+1} es insertado primero en la solución, entonces se puede utilizar el dato de extraer a c_{i+1} e insertarlo en la posición 1.

En caso de que se seleccionen otras posiciones se mantiene el enfoque visto en la subsección anterior para generar los datos considerando la opción en la que los clientes mantienen su orden original y en el que se pueden invertir.

Se puede generar los datos asociados a esta etapa de la siguiente manera:

$$TotalCost_{i,j} = insert_{i,j} - retrieve_{i,j}$$

Como en los escenarios anteriores, si $TotalCost_{i,j} < 0$, entonces consiste en una mejora de la solución.

El costo computacional que conlleva realizar todas las operaciones de este escenario es $O(n^2)$, debido a que la extracción de los pares consecutivos es $O(n)$ y para la inserción se consideran todas las posiciones, realizando dos operaciones en ellas, el costo es de $O(n)$, por lo que el costo total sería de $O(n^2)$.

Como se puede apreciar, el costo computacional asociado a estas operaciones es de $O(n^3)$, lo que representa una mejora en comparación con la exploración exhaustiva, cuyo costo es del orden de $O(n^4)$. Este avance permite un análisis

más eficiente del espacio de soluciones sin comprometer la calidad de los resultados. A continuación, se presentan los resultados obtenidos, destacando el impacto de estas optimizaciones en el desempeño del modelo y la calidad de las soluciones.