Capítulo 1

Técnicas para el cálculo del costo de los vecinos de una solución

Explorar vecindades en el Problema de Enrutamiento de Vehículos (VRP) requiere enfrentar una complejidad asociada a la gran cantidad de soluciones que posee la vecindad. Por ejemplo, si se considera una vecindades que implica la selección de dos clientes y su reinserción en nuevas posiciones genera una complejidad del orden de $O(n^4)$ siendo n la cantidad de clientes del problema, lo que dificulta la exploración exhaustiva en problemas con un gran número de clientes.

Para superar la limitación que conlleva hacer una búsqueda exhaustiva, en este capítulo se proponen estrategias enfocadas a reducir el costo computacional asociado a la generación de datos necesarios para los modelos de optimización. Estos datos incluyen, entre otros, los costos de insertar o extraer clientes, que evalúan la calidad y factibilidad de las soluciones. Las estrategias desarrolladas permiten explorar la vecindad construyendo solamente $O(n^3)$ nuevas soluciones, logrando resultados favorables en cuanto al tiempo de ejecución, a los obtenidos con enfoques tradicionales como la búsqueda exhaustiva.

El contenido de este capítulo está organizado de la siguiente manera. Primero, en la sección 1.1, se define un criterio de vecindad que consiste en extraer dos clientes de sus posiciones y reinsertarlos en diferentes ubicaciones dentro de la solución. Luego, en la sección 1.2, se presenta un método para calcular los parámetros asociados a esta vecindad, con una complejidad computacional menor que la de una exploración exhaustiva. Este enfoque propone una alternativa eficiente para problemas de gran escala.

1.1. 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 $O(n^4)$ soluciones vecinas, siendo n la cantidad de clientes del problema. 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. A continuación, se ilustra el pseudocódigo de las operaciones que generan la vecindad.

```
(seleccionar-ruta r1)
(seleccionar-cliente c1 de r1)
(seleccionar-ruta r2)
(seleccionar-cliente c2 de r2)
(seleccionar-ruta r3)
(insertar-cliente c1 en r3)
(seleccionar-ruta r4)
(insertar-cliente c2 en r4)
```

Figura 1.1: Operaciones para generar la vecindad de reinserción de dos clientes.

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:

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

Cada uno de los clientes seleccionados puede reubicarse 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.

Como la solución de un CVRP puede expresarse como un grafo, se puede establecer una correspondencia entre las aristas de este grafo y las posiciones en las que puede insertarse un cliente.

La inserción de un cliente 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 eliminar la arista $\langle c_i, c_j \rangle$

para formar dos aristas $\langle c_i, c_x \rangle$ y $\langle c_x, c_j \rangle$ (Ver figura 1.2). De esta forma, cada arco $e = \langle c_i, c_j \rangle$ en una ruta representa una posición válida para insertar un cliente.

$r_1 : ... \rightarrow c_x \rightarrow ...$ $\qquad$ $r_1 : ... \rightarrow ...$

$r_2 : ... \rightarrow c_i \rightarrow c_j \rightarrow ...$ $\qquad$ $r_2 : ... \rightarrow c_i \rightarrow c_x \rightarrow c_j \rightarrow ...$

Figura 1.2: Reinserción del cliente c_x entre los vértices c_i, c_j. En rojo la arista que va a ocupar c_x y azul las nuevas que crea.

Si una solución posee r rutas y cada una posee m_k clientes, entonces el número de aristas en cada ruta es $m_k + 1$, 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

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

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, se puede reescribir la suma como:

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

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

Por lo tanto, dado que el número de aristas en el grafo de la solución 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 insertar cada cliente en cada posición de la solución 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.

En la siguiente sección se divide la vecindad en cinco regiones de menor complejidad computacional, para calcular el costo de los vecinos que pertenecen a estas regiones. El objetivo de dividir la vecindad en cinco regiones consiste en utilizar los datos que se calculan en cada una de las regiones, para calcular el resto de soluciones vecinas.

1.2. 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, se puede dividir la vecindad en cinco regiones que permitan calcular el costo de los vecinos de manera más eficiente que una búsqueda exhaustiva. Cada uno de estos escenarios presenta características que afectan tanto a la complejidad computacional de su exploración como a las técnicas utilizadas para generar las soluciones vecinas.

Antes de detallar como se calculan los parámetros en cada región, se ilustra **¿para qué?** la forma en la que se generan los posibles vecinos en la vecindad de extracción e inserción de dos clientes.

Para generar los primeros tipos de vecinos se seleccionan dos clientes que no estén en posiciones consecutivas dentro de la solución y se reubican en posiciones también no consecutivas en las rutas existentes (Figura 1.3 variante (a)).

El segundo tipo de vecino se forma mediante el intercambio de clientes, debido a que dos clientes se pueden extraer de sus posiciones originales y ocupar la posición del otro (Figura 1.3 variante (b)).

Un tercer tipo de vecino se genera al extraer pares de clientes consecutivos, y reubicándolos en posiciones separadas al momento de la inserción (Figura 1.3 variante (c)).

El cuarto tipo de vecino se pueden formar si se considera extraer los clientes de posiciones no consecutivas para luego insertarlos en posiciones consecutivas. En este caso, los clientes se tratan como una subruta y se colocan en una misma posición dentro de la solución (Figura 1.3 variante (d)).

Finalmente, el último tipo de vecino se pueden generar mediante la extracción e inserción conjunta de los clientes seleccionados, manteniendo su estructura como una subruta tanto al extraerlos como al reubicarlos (Figura 1.3 variante (e)). Los cinco tipos de vecinos propuestos permiten explorar todas las soluciones posibles dentro de la vecindad.

A continuación, se describen las cinco regiones principales en las que se divide la vecindad:

1. **Extracción e inserción separada**: En este caso los clientes se extraen de posiciones no consecutivas y se insertan en posiciones no consecutivas.

2. **Intercambio de clientes**: En este escenario se intercambian las posiciones de los clientes que se extraen.

$r_1 : 0 \to 2 \to 4 \to 6 \to 0$
$r_2 : 0 \to 3 \to 5 \to 1 \to 7 \to 0$

(a).

$r_1 : 0 \to 2 \to 4 \to 6 \to 0$
$r_2 : 0 \to 3 \to 5 \to 1 \to 7 \to 0$

(b).

$r_1 : 0 \to 2 \to 4 \to 6 \to 0$
$r_2 : 0 \to 3 \to 5 \to 1 \to 7 \to 0$

(c).

$r_1 : 0 \to 2 \to 4 \to 6 \to 0$
$r_2 : 0 \to 3 \to 5 \to 1 \to 7 \to 0$

(d).

$r_1 : 0 \to 2 \to 4 \to 6 \to 0$
$r_2 : 0 \to 3 \to 5 \to 1 \to 7 \to 0$

(e).

Figura 1.3: Ejemplos de generación de vecinos para la vecindad de extracción e inserción de dos clientes. Los clientes que se extraen, se insertan en las posiciones de su mismo color, o en la posición verde, si se insertan en la misma.

3. **Extracción conjunta e inserción separada**: Se extrae un par de clientes consecutivos, y se insertan en posiciones no consecutivas.

4. **Extracción separada e inserción en posiciones consecutivas**: Se extraen clientes no consecutivos, pero durante la inserción se tratan como una subruta.

5. **Extracción e inserción conjunta**: Ambos clientes se tratan como una subruta, tanto durante su extracción como en su reinserción.

Separar la vecindad en estos cinco escenarios permite calcular todos los parámetros con una complejidad computacional de $O(n^3)$ para cada escenario, lo cual mejora la complejidad de la búsqueda exhaustiva que es de $O(n^4)$.

En los siguientes epígrafes, se presentan técnicas para realizar el cálculo de los parámetros en cada uno de estos escenarios dentro de la vecindad de reinserción de dos clientes. Al describir las técnicas para realizar el calculo de los parámetros, siempre que se hace alusión a un cliente c_i, c_j indica que son el i-ésimo y j-ésimo clientes de una ruta. En caso de que los clientes aparezcan como un número (1, 2 o 6), se hace alusión a ese cliente específico.

En la figura 1.4 se muestra una solución para el CVRP con 7 clientes y 2 rutas. Esta solución se usará como base para los ejemplos de cada escenario.

$$r_1 : 0 \to 2 \to 4 \to 6 \to 0$$
$$r_2 : 0 \to 3 \to 5 \to 1 \to 7 \to 0$$

Figura 1.4: Grafo de solución de un CVRP de 7 clientes y 2 rutas.

A continuación se presenta el primer escenario, donde lo clientes se extraen y se insertan de forma no consecutiva.

1.2.1. Extracción e inserción separadas

En el escenario que se presenta en la sección, los clientes que se extraen no se encuentran en posiciones consecutivas, y para su inserción se seleccionan posiciones tales que los clientes no ocupen posiciones consecutivas tras la inserción (Caso (a) de la figura 1.3).

En este escenario los eventos de inserción y extracción del cliente c_i no afectan a los del cliente c_j. Esto significa que los cálculos necesarios para evaluar las operaciones de inserción y extracción pueden realizarse de manera independiente para cada cliente.

El costo asociado a la extracción de un cliente c_i de una ruta, se puede calcular como:

$$extract_{c_i} = C_{\langle c_{i-1}, c_i \rangle} + C_{\langle c_i, c_{i+1} \rangle} - C_{\langle c_{i-1}, c_{i+1} \rangle}, \tag{1.1}$$

donde C_a representa el costo de la arista a y $extract_{c_i}$ es el costo de extraer el cliente c_i.

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 de una ruta r, solo es necesario calcular el costo de las aristas $\langle c_p, c_i \rangle$ y $\langle c_i, c_{p+1} \rangle$ y restarle el costo de la arista $\langle c_p, c_{p+1} \rangle$ (Fórmula 1.2), donde c_p y c_{p+1} son los clientes que conforman el arco correspondiente a la posición p de la ruta r.

$$insert_{c_i p} = C_{\langle c_p, c_i \rangle} + C_{\langle c_i, c_{p+1} \rangle} - C_{\langle c_p, c_{p+1} \rangle}. \tag{1.2}$$

El dato $insert_{c_i p}$ representa la inserción del cliente c_i en la posición p.

La operación de inserción considera a cada cliente en cada posición de la solución 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 de insertar un cliente en cada posición de la solución sería de $O(n^2)$.

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

$$OperationCost_{c_ip} = insert_{c_ip} - extract_{c_i}. \tag{1.3}$$

Si el movimiento de un cliente consiste en una mejora, entonces $OperationCost_{c_ip} < 0$, en el caso de que se quiera insertar en la misma posición de la que se extrajo el cliente, entonces $OperationCost_{c_ip} = 0$.

En la figura 1.5 se muestra un ejemplo, donde se desea extraer los clientes 1 y 2 e insertarlos en las posiciones 2 de la ruta 1 y la 4 de la ruta 2, de la solución que se muestra en la figura 1.4.

$r_1 : 0 \to 2 \to 4 \to 6 \to 0$
$r_2 : 0 \to 3 \to 5 \to 1 \to 7 \to 0$

$r_1 : 0 \to 4 \to 1 \to 6 \to 0$
$r_2 : 0 \to 3 \to 5 \to 7 \to 2 \to 0$

Figura 1.5: Ejemplo de reinserción de los clientes 1 y 2. A la izquierda la solución de la figura 1.4.

Como se muestra en la figura 1.5, los clientes que se extraen no se encuentran en posiciones consecutivas ni se reinsertan juntos, por lo que el cálculo del *OperationCost* de uno no afecta al otro. En el siguiente epígrafe se trata el caso en que los clientes intercambian sus posiciones.

1.2.2. Intercambio de clientes

En este escenario solo se consideran extraer pares de clientes no consecutivos para realizar un intercambio de sus posiciones. Como los clientes que se extraen no están en posiciones consecutivas entonces para calcular el costo de extraerlos se pueden utilizar los datos que se calculan en la sección anterior.

Para calcular el costo de insertar los clientes se presenta la siguiente estrategia: Se seleccionan todos los pares de clientes c_i, c_j no consecutivos y solo se consideran para su inserción las posiciones que desaparecen al extraerlos las cuales son las correspondientes a las aristas que conectaban a los clientes en la solución ($\langle c_{i-1}, c_i\rangle$, $\langle c_i, c_{i+1}\rangle$, $\langle c_{j-1}, c_j\rangle$ y $\langle c_j, c_{j+1}\rangle$).

Para calcular el costo de insertar al cliente c_i en la posición de c_j solo se suma los costos de los arcos $\langle c_{j-1}, c_i\rangle$ y $\langle c_i, c_{j+1}\rangle$ y se resta el costo del arco $\langle c_{i-1}, c_{i+1}\rangle$ como muestra la fórmula 1.4.

De igual forma se calcula el costo de insertar el cliente c_j (Fórmula 1.4). El costo de extracción de los clientes se mantiene igual debido a que no son clientes consecutivos por lo que se puede usar la fórmula 1.1.

$$insert_{c_i j} = C_{\langle c_{j-1}, c_i \rangle} + C_{\langle c_i, c_{j+1} \rangle} - C_{\langle c_{i-1}, c_{i+1} \rangle}. \tag{1.4}$$

En la formula 1.4, el dato $insert_{c_i j}$ representa el costo de insertar el cliente c_i en la posición del cliente c_j. Para calcular el costo de la operación de intercambiar dos clientes se utiliza la formula:

$$OperationCostc_i c_j = insert_{c_i j} + insert_{c_j i} - extract_{c_j} - extract_{c_i}. \tag{1.5}$$

Si el intercambio de los clientes consiste en una mejora, entonces $OperationCost_{c_i c_j} < 0$, en el caso de que se quiera insertar en la misma posición de la que se extrajo el cliente, entonces $OperationCost_{c_i c_j} = 0$.

En este escenario se seleccionan todos los pares de clientes no consecutivos, lo cual, como se analizo en la sección 1.1, presenta un complejidad computacional de $O(n^2)$. Luego para la inserción de los clientes que se seleccionaron, solo se consideran las cuatro posiciones a las que ellos pertenecen. Por lo tanto el costo de intercambiar a dos clientes es el costo de extraerlos ($O(n^2)$) multiplicado por el costo de insertarlos $O(4)$ es decir: $O(n^2) \times O(4)$ lo cual es $O(n^2)$.

Con la forma en la que se modela la extracción en las dos secciones anteriores, no se consideran los casos en que se extraigan dos clientes consecutivos. Sin embargo, existen soluciones que implican la extracción de clientes consecutivos. En la siguiente sección se profundiza sobre este escenario.

1.2.3. Extracción conjunta e inserción separada

En este escenario se extraen pares de clientes consecutivos y luego se insertan en posiciones no consecutivas (Figura 1.3 variante (c)). A diferencia del escenario anterior, al extraer un cliente se afecta el costo de extracción del cliente siguiente (o anterior), por lo que se debe crear una nueva fórmula que represente la extracción de dos clientes consecutivos. Para la inserción se pueden usar los datos obtenidos con la fórmula 1.2 en el epígrafe 1.2.1.

En la figura 1.6, se muestra paso a paso el proceso de extracción para los clientes 5 y 1 de la solución de la figura 1.3 y cómo se modifica el valor total de la ruta en cada caso. Los números encima de cada arco es la distancia entre cada par de clientes.

Como se puede apreciar en la figura 1.6, el valor final después de la extracción de ambos clientes es el mismo que si se hubiesen extraído los dos en conjunto, esto

1. $r_1 : 0 \xrightarrow{2} 2 \xrightarrow{8} 4 \xrightarrow{4} 6 \xrightarrow{2} 0$
   $value_{r_1} = 16$
   $r_2 : 0 \xrightarrow{3} 3 \xrightarrow{1} 5 \xrightarrow{6} 1 \xrightarrow{3} 7 \xrightarrow{7} 0$
   $value_{r_2} = 20$

2. $r_1 : 0 \xrightarrow{2} 2 \xrightarrow{8} 4 \xrightarrow{4} 6 \xrightarrow{2} 0$
   $value_{r_1} = 16$
   $r_2 : 0 \xrightarrow{3} 3 \xrightarrow{20} 1 \xrightarrow{3} 7 \xrightarrow{7} 0$
   $value_{r_2} = 33$

3. $r_1 : 0 \xrightarrow{2} 2 \xrightarrow{8} 4 \xrightarrow{4} 6 \xrightarrow{2} 0$
   $value_{r_1} = 16$
   $r_2 : 0 \xrightarrow{3} 3 \xrightarrow{2} 7 \xrightarrow{7} 0$
   $value_{r_2} = 12$

1. $r_1 : 0 \xrightarrow{2} 2 \xrightarrow{8} 4 \xrightarrow{4} 6 \xrightarrow{2} 0$
   $value_{r_1} = 16$
   $r_2 : 0 \xrightarrow{3} 3 \xrightarrow{1} 5 \xrightarrow{6} 1 \xrightarrow{3} 7 \xrightarrow{7} 0$
   $value_{r_2} = 20$

2. $r_1 : 0 \xrightarrow{2} 2 \xrightarrow{8} 4 \xrightarrow{4} 6 \xrightarrow{2} 0$
   $value_{r_1} = 16$
   $r_2 : 0 \xrightarrow{3} 3 \xrightarrow{2} 7 \xrightarrow{7} 0$
   $value_{r_2} = 12$

Figura 1.6: Ejemplo de extracción conjunta para la solución de la figura 1.4. A la izquierda si los clientes se extraen uno a uno, a la derecha si se extraen en conjunto.

se debe a que la arista entre el cliente 3 y el cliente 1 desaparece luego de que este último se elimina. Por tanto para calcular el costo 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 $\langle c_{i-1}, c_{i+2}\rangle$ (Fórmula 1.6).

$$extract_{c_i c_{i+1}} = C_{\langle c_{i-1}, c_i\rangle} + C_{\langle c_i, c_{i+1}\rangle} + C_{\langle c_{i+1}, c_{i+2}\rangle} - C_{\langle c_{i-1}, c_{i+2}\rangle} \tag{1.6}$$

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 pueden utilizar los datos que se calculan en la sección 1.2.1 por la fórmula 1.2. Existe un caso en que el proceso de inserción se simplifica, debido a que uno de los clientes puede mantener su posición original.

Si se considera que en la solución de CVRP existe una ruta como la que se

representa a continuación:

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

y se desea reinsertar el cliente c_i en alguna de las posiciones marcadas como 1, 2 o 3, el cliente c_i mantiene su posición actual. Por lo que el cliente c_{i+1} se debe insertar fuera de una de las tres posiciones marcadas. Si c_{i+1} se insertara en alguna de las posiciones marcadas en la solución anterior, entonces ambos clientes se insertarían de forma consecutiva, y este vecino no pertenece al escenario que se está analizando en esta sección.

Por lo tanto, al insertar c_{i+1} en alguna posición distinta a las marcadas en la solución basta con utilizar los datos generados previamente por la fórmula 1.3 para determinar la variación del costo que provoca mover solo este cliente. El mismo análisis se aplica para c_i

Por tanto, el costo total de extraer dos clientes consecutivos e insertarlos en posiciones no consecutivas, se puede calcular de manera general con la ecuación:

$$OperationCost_{c_ip_i,c_jp_j} = insert_{c_ip_i} + insert_{c_jp_j} - extract_{i,j} \tag{1.7}$$

En este caso los valores de $insert_{c_ip_i}$ e $insert_{c_jp_j}$ es el costo de insertar los clientes c_i y c_j en las posiciones p_i y p_j. Estos datos fueron calculados por la formula 1.2 en la sección 1.2.1. El parámetro $extract_{i,j}$ se calculó mediante la fórmula 1.6.

Si la extracción conlleva a una mejora, entonces $OperationCost_{c_ip_i,c_jp_j} < 0$. En el caso que ambos clientes se inserten con su orden en su posición original $OperationCost_{c_ip_i,c_jp_j} = 0$.

El costo computacional que conlleva la generación de los datos que se calculan en esta sección es de $O(n)$, debido a que el costo de extraer los pares de clientes es $O(n)$ y luego para cada cliente en cada par, se utilizan los datos que se generaron en la sección 1.2.1 para calcular el costo de la inserción mediante la fórmula 1.2.

La forma en la que se generan los datos en las dos secciones analizadas no modelan la creación de vecinos en la que se extraen clientes no consecutivos para luego ser insertados de manera consecutiva. En la próxima sección se profundiza en las técnicas para el cálculo de los costos de los vecinos que pertenecen al escenario de extracción separada e inserción conjunta.

1.2.4. Extracción separada e inserción conjunta

En el escenario que se aborda en esta sección los clientes que se extraen no se encuentran en posiciones consecutivas, aunque posteriormente serán insertados

en una misma posición, puesto que al insertarlos se tratan como una subruta (Variante (c) de la figura 1.3). Como el proceso de extracción de un cliente no posee relación con la extracción de otro cliente, se pueden aprovechar los datos que se obtuvieron en la sección 1.2.1.

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 que conectan a los clientes en la solución, es decir, $\langle c_{i-1}, c_i \rangle$, $\langle c_i, c_{i+1} \rangle$, $\langle c_{j-1}, c_j \rangle$ o $\langle c_j, c_{j+1} \rangle$, en la figura 1.7 se muestran dichas posiciones de inserción, para los clientes 2 y 5.

$$r_1 : 0 \rightarrow 2 \rightarrow 4 \rightarrow 6 \rightarrow 0$$
$$r_2 : 0 \rightarrow 3 \rightarrow 5 \rightarrow 1 \rightarrow 7 \rightarrow 0$$

Figura 1.7: Posiciones de inserción para los clientes 2 y 5.

El segundo caso es cuando la posición a insertar no corresponde a ninguna de las aristas a las que pertenecen los clientes que se extraerán. Por cada posición en la que se pueda insertar hay que considerar el caso en que se inserten en el orden c_i, c_j y el caso en que se inserten en el orden c_j, c_i. En el ejemplo de la figura 1.8, se muestran las dos formas en las que se pueden insertar dos clientes c_i y c_j en una posición p.

$$r_1 : ... \rightarrow c_p \rightarrow c_i \rightarrow c_j \rightarrow c_{p+1} \rightarrow ... \qquad r_1 : ... \rightarrow c_p \rightarrow c_j \rightarrow c_i \rightarrow c_{p+1} \rightarrow ...$$

Figura 1.8: Formas de inserción de los clientes c_i, c_j

Para generar los datos en que se insertarán los clientes c_i, c_j en una arista a la que pertenece uno de los dos, se identifica a cuál de los clientes corresponde la arista seleccionada. Si el cliente c_i pertenece a la arista seleccionada, se utilizan los datos que se generaron en la sección 1.2.1 por la fórmula 1.3 para ubicar al cliente c_j en la posición correspondiente. Se pueden utilizar los datos de la sección 1.2.1 debido a que c_i se reinsertará en su misma posición. Para modelar el caso en que se inserte primero c_j, 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 caso que los clientes c_i, c_j se inserten en alguna arista a la que no pertenecen, 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, como se muestra en la fórmula

$$insert_{c_i,c_j,p} = C_{\langle c_p,c_i\rangle} + C_{\langle c_i,c_j\rangle} + C_{\langle c_j,c_{p+1}\rangle} - C_{\langle c_p,c_{p+1}\rangle}. \tag{1.8}$$

Para modelar el caso en que se inserte primero c_j, se alternan entre c_i y c_j.

A partir de la solución de la figura 1.4, se evalúa el proceso de extracción e inserción conjunta para diferentes pares de clientes.

Se supone que se quiere insertar conjuntamente los clientes 4 y 7 en una posición correspondiente a las aristas directamente conectadas a ellos. Si seleccionamos la arista $\langle 2, 4\rangle$, el cliente 4 mantiene su posición original y 7 se inserta inmediatamente después de 2, dando como resultado la ruta:

$$r_1 : 0 \to 2 \to 7 \to 4 \to 6 \to 0$$

Alternativamente, 7 puede insertarse después de 4, resultando en:

$$r_1 : 0 \to 2 \to 4 \to 7 \to 6 \to 0$$

Ahora se considera insertar conjuntamente 4 y 7 en una posición arbitraria en la ruta r_2. Por ejemplo, entre los clientes 5 y 1. El resultado de insertar 4 seguido de 7 sería:

$$r_2 : 0 \to 3 \to 5 \to 4 \to 7 \to 1 \to 0$$

Si se alterna el orden y primero se inserta 9 seguido de 4, se obtiene:

$$r_2 : 0 \to 3 \to 5 \to 7 \to 4 \to 1 \to 0$$

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

Cada parámetro se puede generar mediante la fórmula:

$$OperationCost_{c_i,c_j,p} = insert_{c_i,c_j,p} - extract_{c_i} - extract_{c_j} \tag{1.9}$$

En 1.9 los datos $extract_{c_i}$ y $extract_{c_j}$ ya se obtuvieron por la fórmula 1.1 de la sección 1.2.1. Si la inserción de manera consecutiva de los clientes conlleva una mejora entonces $OperationCost_{c_i,c_j,p} < 0$.

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

Con los escenarios descritos en las secciones 1.2.1, 1.2.3 y 1.2.4, se contemplan los casos en los que se puede extraer e insertar dos clientes siempre que una de las dos, o la inserción o la extracción sean consecutivas. Por lo que aún no se modela el escenario en que se extraen clientes consecutivos y se insertan de manera consecutiva. La siguiente subsección tratará de este escenario.

1.2.5. Extracción e inserción conjunta

Para generar los datos cuando los clientes son tratados como una subruta de dos clientes, se utiliza la misma idea de extracción e inserción conjunta que se propone en las secciones 1.2.3 y 1.2.4. Se puede generar los datos de la siguiente manera:

$$OperationCost_{c_i,c_j,p} = insert_{c_i,c_j,p} - retrieve_{c_i,c_j} \tag{1.10}$$

El dato $insert_{c_i,c_j,p}$ representa los costos de insertar a los clientes c_i y c_j en la posición p y se calcula siguiendo la fórmula 1.8. El dato $retrieve_{c_i,c_j}$ se calculó en la sección 1.2.3 mediante la fórmula 1.6 y representa el costo de extraer los clientes c_i y c_j como un par de su posición original. Como en los escenarios anteriores, si $OperationCost_{i,j} < 0$, entonces consiste en una mejora de la solución.

Existen tres posiciones en la solución, en las cuales al insertar dos clientes c_i, c_{i+1}, pueden mantener su posición original (c_i, c_{i+1}), o se invierte la subruta (c_{i+1}, c_i), por lo tanto no se puede utilizar la fórmula 1.10 para calcular este parámetro. A continuación se muestran las posiciones y cómo se calcula el dato en caso de insertarlos en una de las tres posiciones.

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

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

Las posiciones marcadas como 1, 2 y 3, son las posiciones a las que pertenecen el par de clientes que se desea extraer (c_i, c_{i+1}). En caso de que se seleccione alguna de las posiciones marcadas para realizar la inserción y la subruta mantiene su forma original (c_i, c_{i+1}), la solución se mantiene igual y $OperationCost_{c_i,c_j,p} = 0$.

Al modelar el caso en que se desea insertar la subruta invirtiendo el orden de sus clientes (c_{i+1}, c_i), puesto que se puede insertar primero el cliente c_{i+1} y luego c_i en alguna de las posiciones marcadas, se utiliza el dato para insertar el cliente c_i en la posición 3 (o para insertar el cliente c_{i+1} en la posición 1) que se calculó con la fórmula 1.3.

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 al cálculo de los *OperationCost* en los escenarios 1.2.1, 1.2.3, 1.2.4, 1.2.5 es de $O(n^3)$, lo que representa una mejora en comparación con la exploración exhaustiva, cuyo costo es del orden de $O(n^4)$.

La generación de los parámetros dividiendo una vecindad en escenarios con una complejidad de orden menor que la complejidad de la vecindad, y que además modelen cada manera de generar los vecinos, permite un análisis más eficiente del espacio de soluciones con respecto a la búsqueda exhaustiva. Si se generan los parámetros siguiendo las fórmulas de las secciones 1.2.1, 1.2.3, 1.2.4 y 1.2.5 no se comprometen la calidad de los resultados debido a que se generan todos los vecinos de una solución para la vecindad de extracción e inserción de dos clientes.

A continuación, se presentan los resultados obtenidos por los modelos presentados en el capítulo anterior, destacando el impacto de los cálculos de los datos propuestos en este capítulo para mejorar la eficiencia de los modelos de optimización y la calidad de las soluciones.