

# Formulaciones de Star Routing

Nahuel Nostrala Hatz

June 9, 2023

# 1 Star Routing

Descripción del Problema:

Se tiene un grafo dirigido completo  $G = (N, E)$ , con un conjunto de clientes  $S \subseteq N$  y un conjunto de vehículos  $K$ . Los nodos que no pertenecen a  $S$  se llamarán paradas intermedias y llamaremos  $T$  al conjunto de paradas intermedias, de manera que  $S \cup T = N$ . Cada nodo con un cliente puede tener varias paradas intermedias asociadas, llamamos  $\varrho(s)$  al conjunto de paradas asociadas a  $s$ , el cual nunca es vacío ya que  $s \in \varrho(s)$ . Una parada intermedia puede estar asociada a más de un cliente. Se busca el camino de menor costo que visite a todos los clientes, donde visitar significa pasar por el nodo o por alguna de las paradas intermedias asociadas a este nodo. El costo de cada camino está pre-computado y se refleja en  $w_{ij}$ . Los caminos empiezan y terminan en el depósito que es el nodo 1. Todos los vehículos tienen capacidad entera  $Q$  y cada cliente  $s$  se considera que tiene un peso igual a 1.

El modelo es:

$$\min \sum_{i \in N} \sum_{j \in N} (w_{ij} \sum_{k \in K} x_{ijk})$$

Sujeto a:

1. Cada vehículo sale del nodo en el que entra

$$\sum_{j \in N, (i,j) \in E} x_{ijk} = \sum_{j \in N, (j,i) \in E} x_{jik} \forall i \in N \forall k \in K$$

2. Cada cliente es servido por exactamente un vehículo

$$\sum_{k \in K} y_{sk} = 1 \forall s \in S$$

3. Cada vehículo en algún momento sale del depósito.

$$\sum_{j \in N, (j,1) \in E} x_{1jk} = 1 \forall k \in K$$

4. Un vehículo solamente puede servir clientes visitados. Se define  $\varrho(s)$  como el vecindario de  $s$ , es decir, los nodos a los que se puede ir caminando desde  $s$

$$y_{sk} \leq \sum_{i \in N} \sum_{v \in \varrho(s)} x_{ivk} \forall k \in K \forall s \in S$$

5. Restricción de capacidad

$$\sum_{s \in S} y_{sk} \leq Q \forall k \in K$$

6. Subtour Elimination Constraints (Miller-Tucker-Zemlin)

$$u_{ik} - u_{jk} + (|N| - 1)x_{ijk} \leq (|N| - 2) \forall i \in N : i \neq 1 \forall j \in N : j \neq 1, j \neq i \forall k \in K$$

$$1 \leq u_{ik} \leq |N| - 1 \forall i \in N : i \neq 1 \forall k \in K$$

Donde:

$x_{ijk}$  es una variable binaria que representa que en el camino óptimo, el vehículo  $k$  pasa por la arista que une el nodo  $i$  con el nodo  $j$ .

$y_{sk}$  es una variable binaria que indica que el vehículo  $k$  es el que atiende al cliente  $s$ . Se requiere esta variable ya que el camino de cierto vehículo podría pasar por el cliente pero no atenderlo, por ejemplo, por no tener suficiente capacidad.

$u_{ik}$  es una variable entera positiva utilizada en las condiciones MTZ, que representa aproximadamente en qué orden se visita cada nodo  $i$  por el vehículo  $k$ .

## 2 Set Partitioning

Vamos a definir  $\Omega$  como el conjunto de las rutas elementales que empiezan y terminan en el depósito y cumplen con las restricciones de capacidad. Vamos a usar los parámetros  $a_{irs}$  (variable binaria) que indica si la ruta  $r$  pasa por el nodo  $i$  y allí sirve al cliente  $s$ , y el parámetro  $c_r$  que es el costo de la ruta  $r$ . Si  $z_{ir} = \min(1, \sum_{s \in S} a_{irs})$  luego vale la igualdad  $c_r = \sum_{i \in N} z_{ir} w_{ij}$  con  $(i, j) \in r$ , esto es,  $j$  es el próximo nodo a visitar después de  $i$ .

### 2.1 Master Problem

El objetivo es:

$$\min \sum_{r \in \Omega} c_r \theta_r$$

Sujeto a:

1. Cada cliente  $s$  tiene exactamente una ruta que pasa por el.

$$\sum_{r \in \Omega} \sum_{i \in N} a_{irs} \theta_r = 1 \forall s \in S$$

2. Se utilizan exactamente  $|K|$  rutas

$$\sum_{r \in \Omega} \theta_r = |K|$$

Donde:

$\theta_r$  es una variable binaria que indica si se utiliza o no la ruta  $r$ , para algún vehículo.

## 2.2 Dual

Dado el modelo de Set Partitioning que ya fue definido, queda determinado el problema dual. La solución del problema dual nos servirá más adelante para resolver el subproblema de Pricing. Las constantes tienen la misma definición que en el Master Problem, y en este caso estamos optimizando sobre las variables  $\lambda_i$ . Notemos que hay una variable  $\lambda_i$  por cada restricción del Master Problem, o sea, por cada  $i \in S$  y una adicional para la última restricción, la cual se llama  $\lambda_0$ .

El objetivo es:

$$\max \sum_{s \in S} \lambda_s + |K| \lambda_0$$

Sujeto a:

$$\lambda_0 + \sum_{s \in S} \sum_{i \in N} a_{isr} \lambda_s \leq c_r \forall r \in \Omega$$

$$\lambda_i \text{ libre } \forall i \in S$$

$$\lambda_0 \text{ libre}$$

## 2.3 Pricing

Para resolver el problema de Pricing, primero nos tenemos que enfocar en calcular los costos reducidos. Una vez que se resuelve el problema dual restringido, llamamos  $\lambda_0^*, \lambda_i^*$  a la solución óptima. Los costos reducidos se calculan de la siguiente manera:

$$\bar{c}_r = c_r - \lambda_0^* - \sum_{s \in S} \sum_{i \in N} a_{isr} \lambda_s^*$$

Aquí se quiere encontrar si existe un costo reducido negativo, es decir, ver si el mínimo es menor que 0 y en este caso descubrir qué ruta lo genera. Dado que a esta altura no tenemos un algoritmo de labeling, alcanza con tener definidas las variables  $a_{isr}$  para describir la ruta.

Definimos  $z_{ir} = \min(1, \sum_{s \in \tilde{S}} a_{isr})$ , es decir  $z_{ir}$  indica si la ruta  $r$  pasa por el nodo  $i$ . Esto nos obliga a definir el conjunto  $\tilde{S} = S \cup \{s_0\}$ , donde  $s_0$  es un cliente fantasma que se utiliza para indicar el caso en el que la ruta  $r$  pasa por el nodo  $i$  pero no atiende a ningún cliente, en cuyo caso  $a_{is_0r} = 1, a_{isr} = 0 \forall s \in S$ . Además, definimos la variable  $u_{ij} \in \{0, 1\}$  que indica si en la ruta  $r$  se utiliza el eje entre  $i$  y  $j$ .

De esta manera, es posible calcular  $c_r = \sum_{i \in N} \sum_{j \in N} u_{ij} w_{ij}$

Con estas variables estamos en condiciones de definir el modelo:

### 2.3.1 Modelo con Programación Lineal Entera

Las variables  $x_{is}$  de este modelo son las que nos van a permitir reconstruir los valores de  $a_{isr}$ . Este modelo usa las variables  $x_{is} \in \{0, 1\}$  y  $z_i \in \{0, 1\}$

La función objetivo es:

$$\min \sum_{i \in N} \sum_{j \in N} u_{ij} w_{ij} - \sum_{s \in S} \sum_{i \in N} x_{is} \lambda_s^* - \lambda_0^*$$

Sujeto a:

1. Se debe cumplir  $z_{ir} = \min(1, \sum_{s \in \tilde{S}} a_{isr})$ .

$$z_i \geq \frac{1}{|\tilde{S}|} \sum_{s \in \tilde{S}} x_{is}$$

2. Cada cliente es atendido en a lo sumo una parada.

$$\sum_{i \in N} x_{is} \leq 1 \forall s \in S$$

3. El camino contiene al nodo depósito (nodo  $i = 1$ )

$$z_1 = 1$$

4. Un cliente solo se puede atender desde una parada en su vecindario, o sea, si el nodo  $i$  no está en la zona caminable de  $s$ , no lo puede atender

$$\sum_{i \notin \varrho(s)} x_{is} = 0 \forall s \in S$$

5. Restricción de capacidad:

$$\sum_{i \in N} \sum_{s \in S} x_{is} q_s \leq Q$$

6.  $x_{is_0} = 1$  implica que  $\sum_{s \in S} x_{is} = 0$ . Esto es porque  $s_0$  modela que se pasa por el nodo  $i$  sin atender a nadie.

$$\sum_{s \in S} x_{is} \leq |\tilde{S}|(1 - x_{is_0}) \forall i \in N$$

7.  $u_{ij}$  se puede utilizar solamente si se pasa por  $i$  y  $j$  (no es obligatorio).

$$u_{ij} \leq z_i + z_j - 1 \forall (i, j) \in E$$

8. Cada nodo  $i$  del camino debe tener exactamente un arco que salga de allí

$$z_i = \sum_{(i,j) \in E} u_{ij} = \sum_{(j,i) \in E} u_{ji} \forall i \in N$$

9. Subtour Elimination Constraints

$$v_i - v_j + (|N| - 1)u_{ij} \leq (|N| - 2) \forall i \in N : i \neq 1 \forall j \in N : j \neq 1, j \neq i$$

$$1 \leq v_i \leq |N| - 1 \forall i \in N : i \neq 1$$