

IN47B, Aux 4: “Heurísticas de Ruteo”

Leonardo López H.

lelopez@ing.uchile.cl

28 de Marzo de 2008.

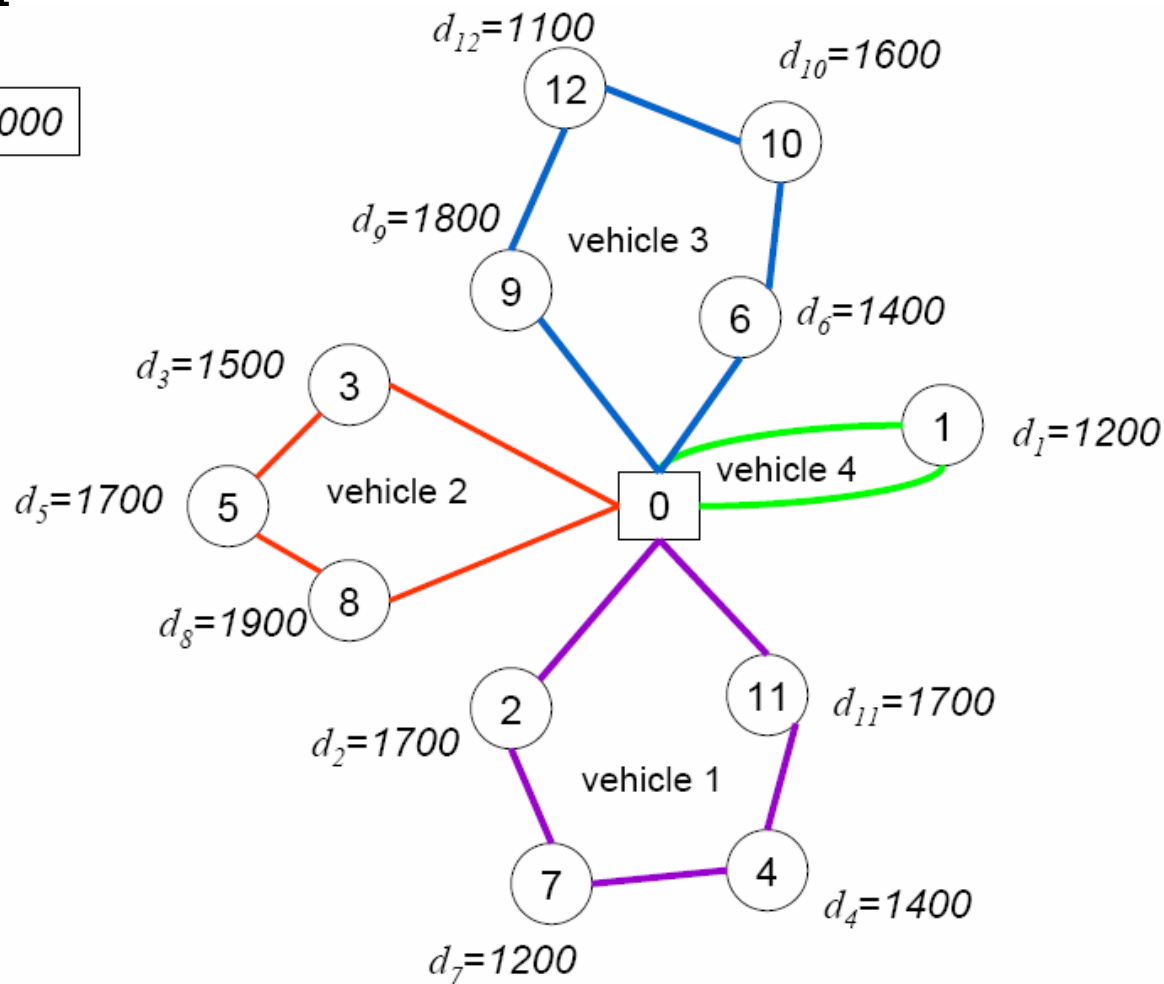
Tarea 2

- Tenemos:
 - Una única bodega central (depósito).
 - Clientes distribuidos en un plano, cada uno con cierta demanda por un único bien.
 - Vehículos, todos de una misma capacidad.
 - Se debe satisfacer la demanda de todos los clientes.
 - Los vehículos salen a repartir desde el depósito y luego deben retornar a él.
- Se debe encontrar un conjunto de rutas que utilice los menos vehículos posibles y al mismo tiempo tenga un costo (en términos de distancia recorrida) cercano al mínimo.
- Es decir, nos piden resolver de forma heurística el *Capacitated Vehicle Routing Problem* (CVRP), donde el número de vehículos no es fijo.

Tarea 2

- Ejemplo:

$$C=6000$$



¿Cómo resolvemos el CVRP?

- Buscamos métodos heurísticos para resolver el CVRP.
- Una forma intuitiva es: Agrupar y Rutear.
 - Primero asignar clientes a vehículos y luego rutear cada vehículo.
 - Es decir, encontramos un ruta o tour para cada vehículo sobre el conjunto de clientes que le fue asignado.
 - La demanda total del conjunto de clientes que visita un vehículo, no puede superar su capacidad.
 - ¿Cómo encontramos buenas rutas o tours para cada vehículo?
 - Resolviendo bien el *Traveling Salesman Problem* (TSP).

Tarea 2

- Es decir, para la tarea debemos ser capaces de resolver de forma heurística:
 - Primero el *Traveling Salesman Problem* (TSP).
 - Y luego *Capacitated Vehicle Routing Problem* (CVRP).

Formulación TSP

- Dado un Grafo $G=[N,E]$:

$$\min \sum_{(i,j) \in E} c_{ij} x_{ij}$$

s.a.

$$\sum_{(i,j) \in E} x_{ij} = 1 \quad \forall i \in N$$

$$\sum_{(i,j) \in E} x_{ij} = 1 \quad \forall j \in N$$

$$\sum_{\{(i,j) / i \in S, j \in S, i \neq j\}} x_{ij} \leq |S| - 1 \quad \forall S \subseteq N \text{ tal que } 2 \leq S \leq |N| - 2$$

$$x_{ij} \in \{0,1\} \quad \forall (i,j) \in E$$

Formulación CVRP

- Dado un Grafo $G=[N,E]$.
- Donde el nodo 0 es el depósito y el resto los clientes.
- Cada cliente tiene una demanda $d_i \quad \forall i \in N \setminus \{0\}$
- Cada vehículo con capacidad C .
- Hay K vehículos disponibles.
- Dado un conjunto S subconjunto de $N \setminus \{0\}$, vamos a definir $r(S)$ como el número mínimo de vehículos necesarios para servir a todos los clientes en S .

$$r(S) = \left\lceil \frac{\sum_{i \in S} d_i}{C} \right\rceil$$

- Notar que $r(V \setminus \{0\})$ determina el número mínimo de camiones con los que se puede atender a todos los clientes.
- Por último: $\delta(S) = \{(i, j) \in E / i \in S, j \in N \setminus S\}$

Formulación CVRP

$$\min \sum_{e \in E} c_e x_e$$

$$\sum_{e \in \delta(i)} x_e = 2 \quad \forall i \in V \setminus \{0\},$$

$$\sum_{e \in \delta(0)} x_e = 2K,$$

$$\sum_{e \in \delta(S)} x_e \geq 2r(S) \quad \forall S \subseteq V \setminus \{0\}, S \neq \emptyset,$$

$$x_e \in \{0, 1\} \quad \forall e \notin \delta(0),$$

$$x_e \in \{0, 1, 2\} \quad \forall e \in \delta(0).$$

Heurísticas para el TSP

- Consideremos dos tipos de heurísticas:
 - Construcción de rutas o tours:
 - Armar una ruta factible.
 - Ej: Vecino más cercano, método Greedy, Clarke y Wright.
 - Mejoramiento de rutas o tours:
 - Dada una ruta, realizar intercambios en el orden en que se visitan los clientes para mejorar la solución inicial.
 - Ej: 2-Opt, búsqueda tabú.

Vecino más cercano

- Para construir un tour del TSP.
- Añade en cada paso el vértice más cercano al actual.

Inicialización

Seleccionar un vértice j al azar.

Hacer $t = j$ y $W = V \setminus \{j\}$.

Mientras ($W \neq \emptyset$)

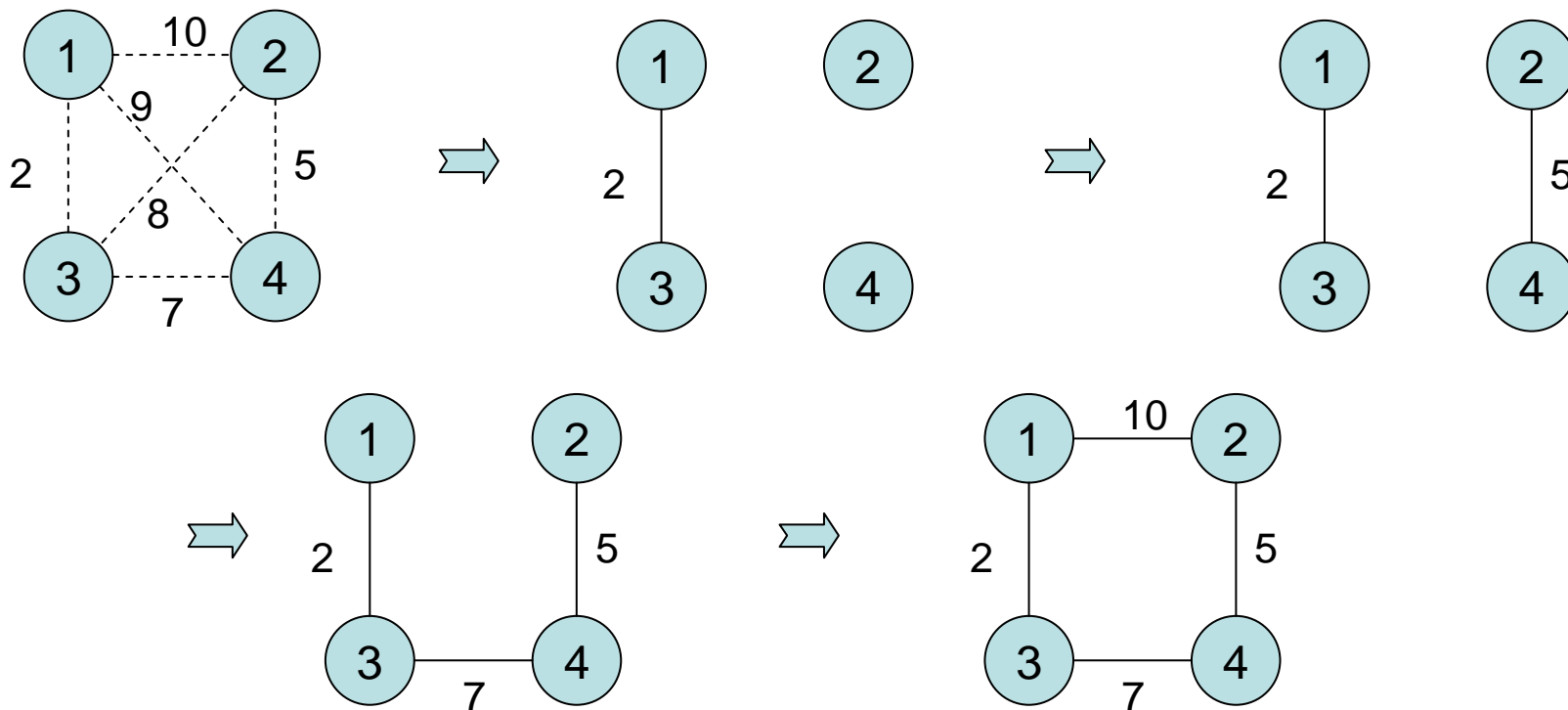
Tomar j de W / $c_{tj} = \min \{c_{ti} / i \text{ en } W\}$

Conectar t a j

Hacer $W = W \setminus \{j\}$ y $t = j$.

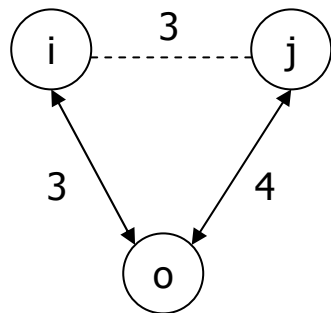
Método Greedy

- Para construir un tour del TSP.
- En cada iteración se busca el arco con menor costo y es agregado al tour, cumpliendo que:
 - Cada nodo no puede tener más de dos arcos incidentes en el tour y que no se produzcan subtours.

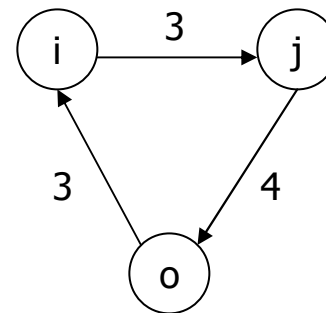


Clarke y Wright

- Esta heurística nos sirve para construir rutas directamente para el CVRP.
 - Paso 0: Asignar un vehículo por cliente, con costos por cliente $c_{oi} + c_{io}$.
 - Paso 1: Combinar dos clientes para un vehículo tal que $c_{oi} + c_{ij} + c_{jo} < c_{oi} + c_{io} + c_{oj} + c_{jo}$, con una ganancia $s_{ij} = c_{io} + c_{oj} - c_{ij}$. Se combinan los clientes con mejor s_{ij} .
 - Paso 2: Los clientes i y j se consideran ahora como un solo cliente. De esta manera se siguen juntando clientes hasta llegar a una solución factible.



$$\text{Costo} = 3 + 3 + 4 + 4 = 14$$



$$\text{Costo} = 3 + 3 + 4 = 10$$

Clarke y Wright

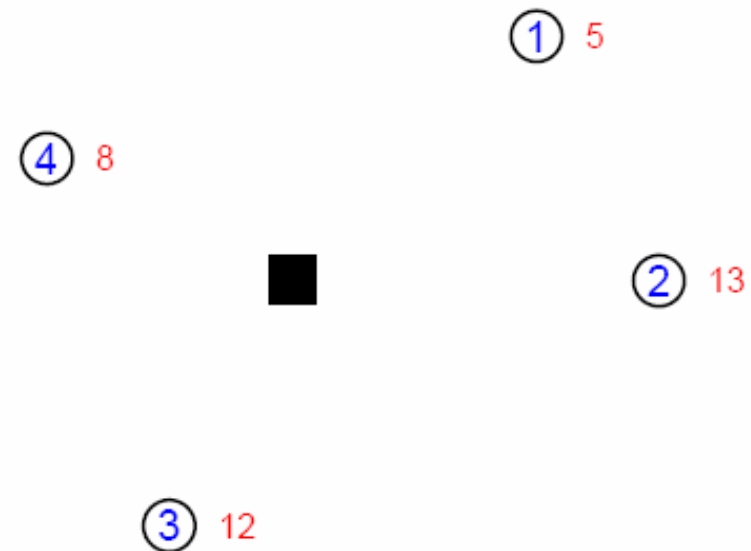
■ Ejemplo:

Input Data:

- $n = 4$ customers
- 0 depot
- $d_i = (0, 5, 13, 12, 8)$
demands
- Cost Matrix= $\{c_{i,j}\} =$

(i,j)	0	1	2	3	4
0	0	2	3	2	2
1	2	0	2	4	4
2	3	2	0	4.5	5
3	2	4	4.5	0	3
4	2	4	5	3	0

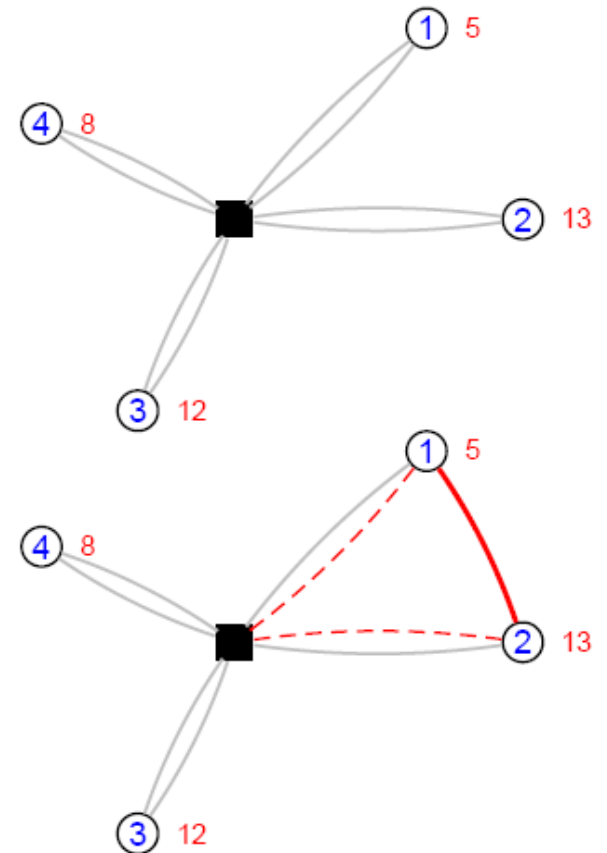
- $Q = 20$ vehicle capacity



Clarke y Wright

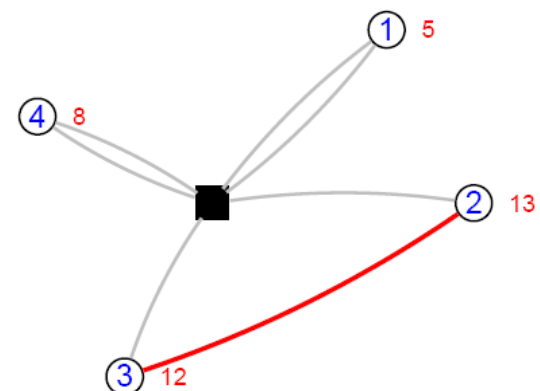
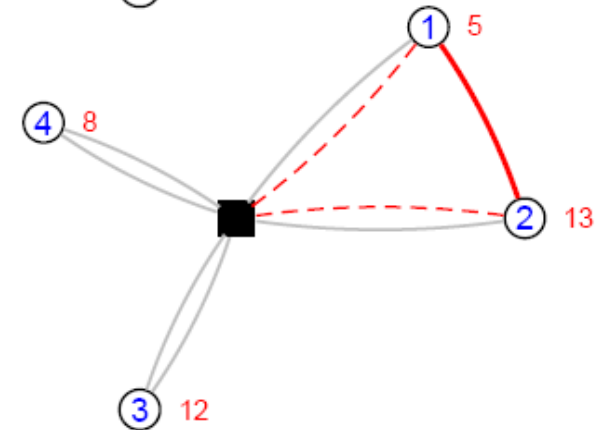
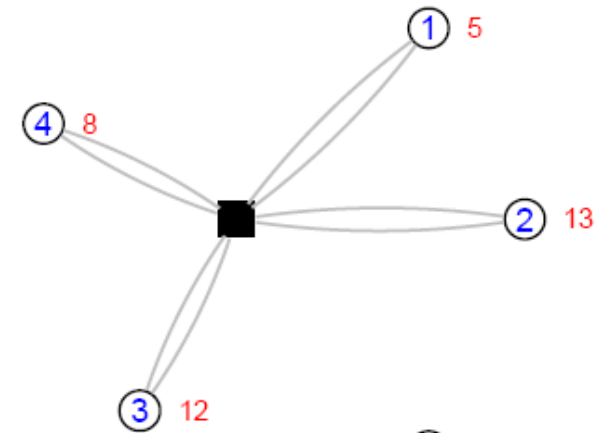
■ Ejemplo:

- Inicialmente, cada cliente es servido por un único vehículo.
- Podemos combinar las rutas de los clientes 1 y 2, lo que nos entregaría un ahorro de:
 $s(1,2) = c(1,0) + c(0,2) - c(1,2)$
 $= 2 + 3 - 2 = 3.$



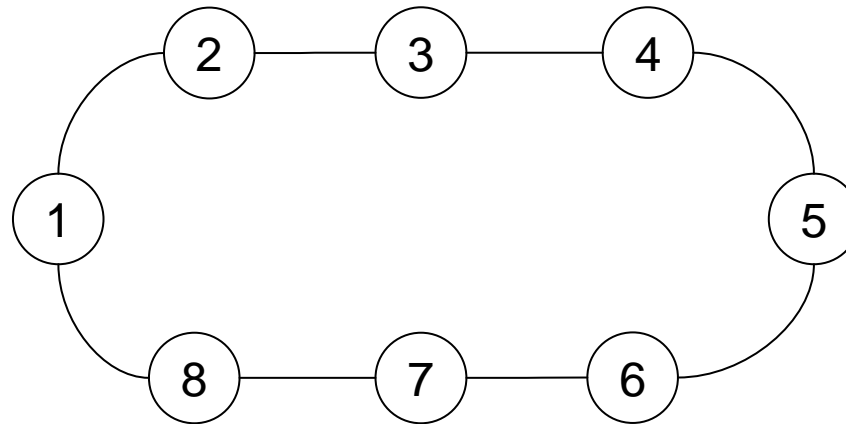
Clarke y Wright

- Ejemplo:
 - Inicialmente, cada cliente es servido por un único vehículo.
 - Podemos combinar las rutas de los clientes 1 y 2, lo que nos entregaría un ahorro de: $s(1,2) = c(1,0) + c(0,2) - c(1,2) = 2 + 3 - 2 = 3$.
 - Convinar 2 y 3, no es factible pues no la demanda de 2 y 3 es 25 que es mayor a la capacidad de cada vehículo.
- Detalles de la heurística:
 - Ver archivo **ClarkeWright.pdf** en **U-Cursos**.



2-Opt

- 2-Opt es un heurística de mejoramiento de tours.
- Consideremos un tour inicial factible para el TSP, representado en un grafo no dirigido. Por ejemplo:

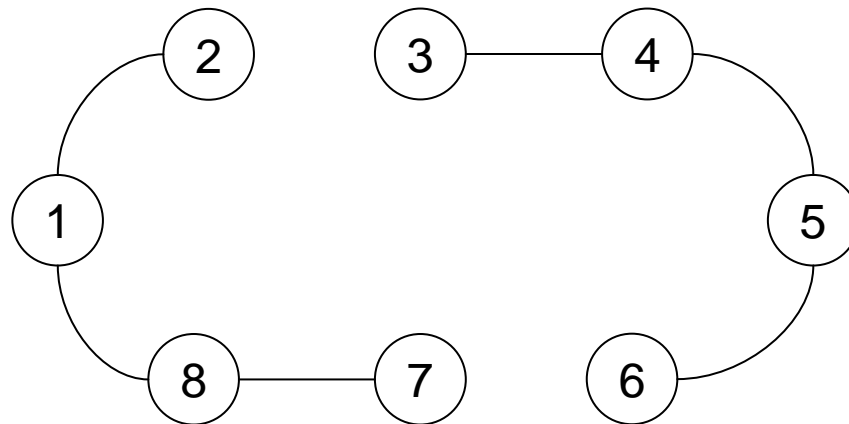


- Cualquier tour factible para el TSP, lo podemos representar en un arreglo escribiendo la secuencia de nodos. Por ejemplo:

1	2	3	4	5	6	7	8
---	---	---	---	---	---	---	---

2-Opt

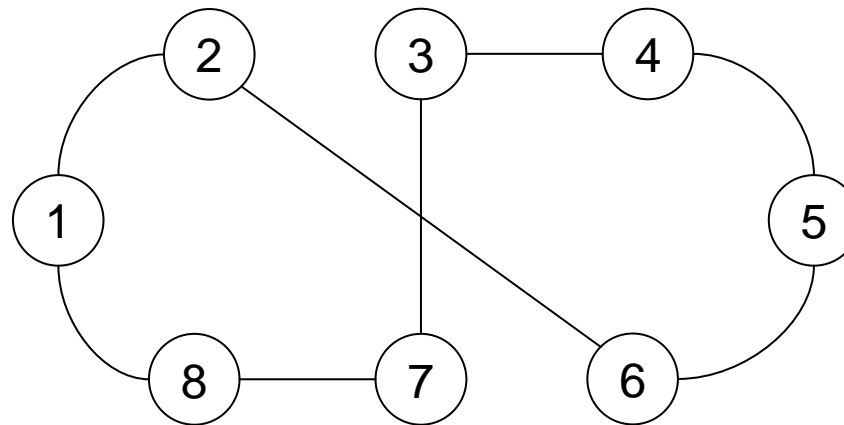
- **2-Opt toma dos aristas no continuas del tour y las elimina. Luego, existe una única forma reconectar los nodos de forma de obtener una nueva ruta factible para el TSP. Si este nuevo tour tiene un costo menor que el inicial, entonces actualizo el tour. Si no, me quedo con el tour inicial.**
- En el ejemplo, podríamos eliminar las aristas (2,3) y (6,7):



- ¿Qué aristas agregamos para formar un nuevo tour factible?

2-Opt

- Hay una única forma de construir un nuevo tour:
 - No podemos agregar (2,3) y (6,7) pues son las aristas eliminadas.
 - Si agregamos las arista (2,7) y (3,6), obtenemos dos sub-tours!!!
 - Luego, sólo nos queda la opción de agregar las aristas (2,6) y (3,7).



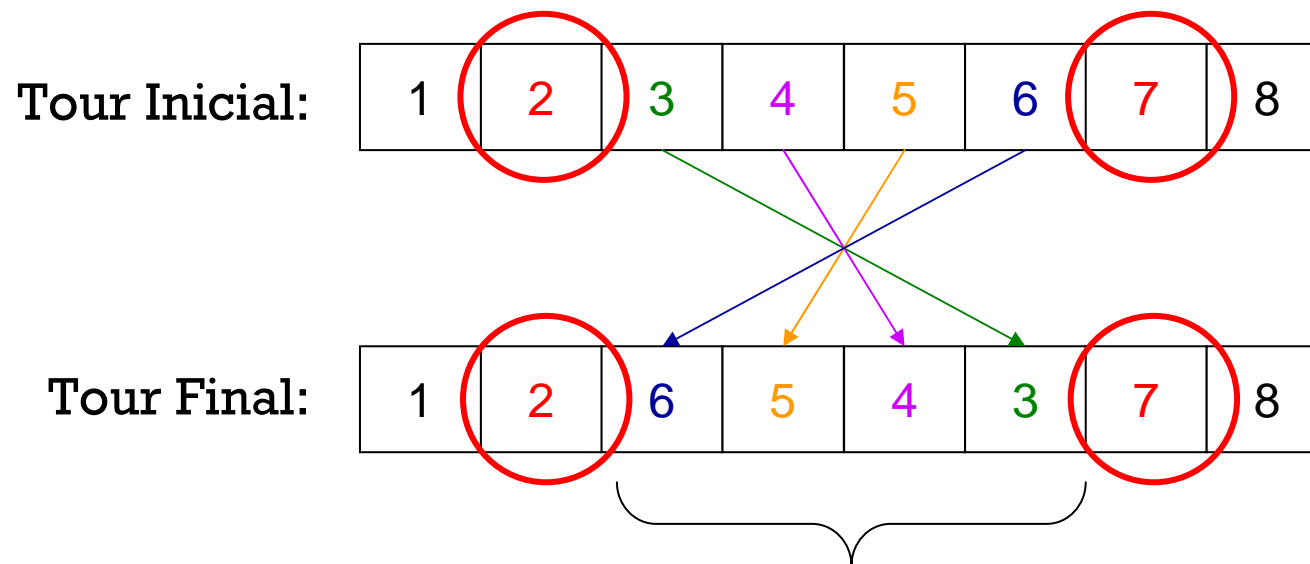
- Si este nuevo tour tiene un costo menor que el anterior, nos quedamos con este. Si no, nos quedamos con el inicial.

2-Opt

- ¿Cómo queda el arreglo?

1	2	6	5	4	3	7	8
---	---	---	---	---	---	---	---

- ¿Qué paso en el arreglo?



Invertimos el orden de los nodos

Implementando 2-Opt

- Sea $T[]$ un arreglo con un tour y $C(T[])$ el costo del tour.
- Sea $TI[]$ un arreglo con un tour inicial.
- Sea $TA[]$ un arreglo auxiliar tal que $TA[] = TI[]$.
- Si con 2-Opt, queremos eliminar las sgtes. 2 aristas:
 - 1) La que une el nodo de la posición k del arreglo con el nodo de la posición $k+1$. Con $0 \leq k < (n-1)$.
 - 2) Y la que une el nodo de la posición j del arreglo con el nodo de la posición $j+1$. Con $k < j < n$.
- Generamos una nueva ruta en el arreglo auxiliar haciendo:

```
for(int i=0; i<=j-k; i++){  
    TA[k+i]=TI[j-i];  
}
```

- En $TA[]$ tenemos ahora el nuevo tour.
- Si $C(TA[]) < C(TI[])$, entonces cambiamos de tour. Ahora nuestro nuevo tour es $TA[]$.

Búsqueda Tabú

- Metaheurística de mejoramiento.
- La podemos utilizar para el TSP y para el CVRP.
- Problema: 2-Opt nos puede llevar a óptimos locales.
- Búsqueda tabú busca escapar de óptimos locales, utilizando memorias flexibles.
 - Corto plazo: La llamaremos **Lista Tabú** y almacena la historia de los últimos movimientos. La idea es tener un registro de los movimientos prohibidos (movimientos tabú) que pueden hacernos regresar a un óptimo local de una iteración anterior. El largo de la Lista Tabú es un parámetro a decidir.
 - Mediano plazo: se registran los atributos más comunes de un conjunto de soluciones , para poder explorar dicha zona del espacio de búsqueda. Esta memoria es utilizada por la estrategia de intensificación .
 - Largo plazo: para diversificar la búsqueda sobre regiones que no han sido exploradas aún. Esta memoria es utilizada por la estrategia de diversificación.

Búsqueda Tabú

- Para implementar Búsqueda Tabú necesitamos:
 - Encontrar una solución inicial.
 - Definición de la Lista Tabú.
 - Definición de vecindad de una solución.
 - Definición de un criterio de aspiración.
 - Definición de una estrategia de intensificación.
 - Definición de una estrategia de diversificación.
 - Definición de un criterio de parada.

Búsqueda Tabú para Ruteo

- Solución inicial:
 - Utilizar alguna heurística para construir rutas (del TSP o del CVRP).
- Lista Tabú:
 - Por ejemplo, que los clientes cambiados en el orden de una ruta (TSP) o los clientes intercambiados de ruta (CVRP) no puedan ser escogidos nuevamente.
- Vecindad:
 - A partir de una solución podemos encontrar nuevas soluciones utilizando heurísticas de mejoramiento.
 - TSP: Por ejemplo, 2-Opt para mejorar un tour.
 - CVRP: Por ejemplo,
 - » Dentro de una ruta, usar estrategias para el TSP.
 - » Utilizar heurísticas de mutación, de inserción, etc, para intercambiar clientes de ruta.
- Criterio de aspiración:
 - Por ejemplo, aceptar nuevas soluciones sólo si son mejores que las anteriores.

Búsqueda Tabú para Ruteo

- Estrategia de intensificación:
 - Por ejemplo, intensificar la exploración de vecindarios de las mejores soluciones encontradas.
- Estrategia de diversificación:
 - Por ejemplo, repetir el proceso de búsqueda pero partiendo de una nueva solución inicial:
 - Por ejemplo, utilizando otra heurística de construcción de rutas ó una nueva aleatoria.
 - Si se obtienen mejores soluciones, se ejecuta el proceso de intensificación.
- Criterio de parada:
 - Por ejemplo, un número determinado de iteraciones.