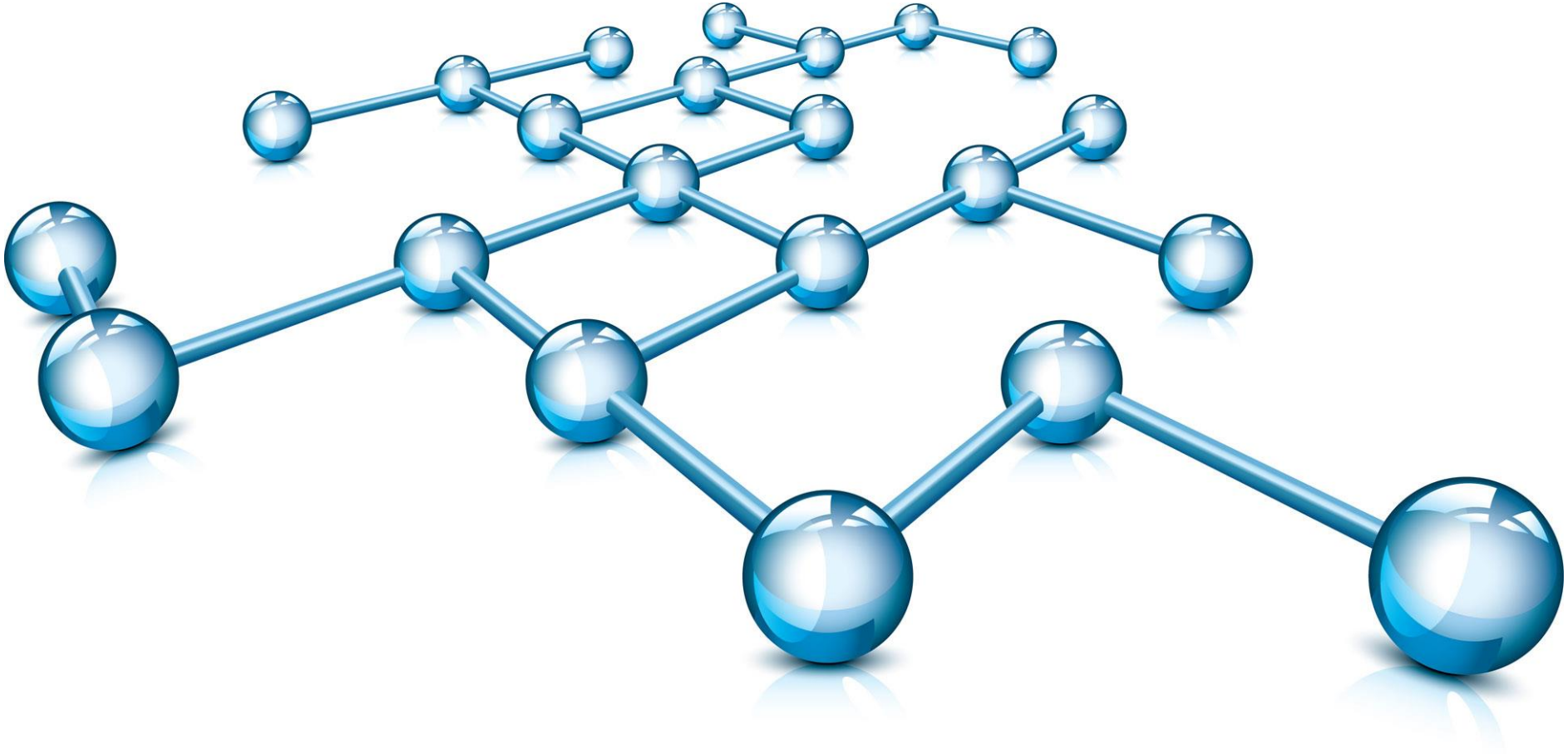


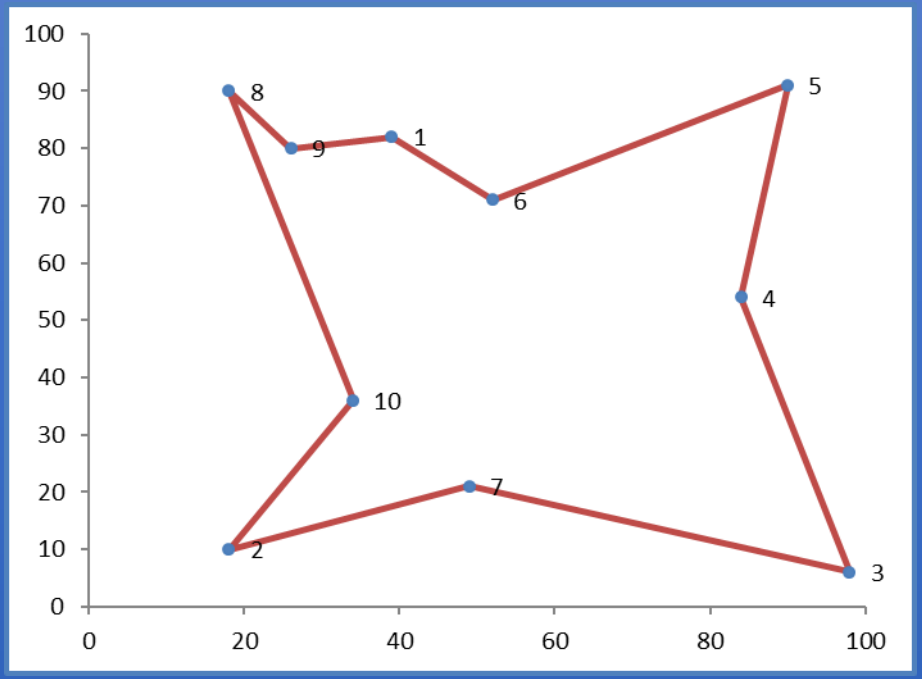
Modelos y algoritmos para logística y transporte

Agustín Pecorari

Rodrigo Maranzana

Problema del Viajante de Comercio

- Consideremos un viajante de comercio que debe recorrer n ciudades en el menor tiempo (o menor costo) posible



Ejemplo TSP

- *La solución para 13.500 ciudades de EEUU fue resuelto (con ciertas heurísticas) en 5 años de tiempo de cómputo.*

http://www.tsp.gatech.edu/

In Pursuit of the Traveling Salesman. W.J. Cook

TSP

Muchos problemas se relacionan con TSP:

- VRP.
- Asignación de tareas a una máquina con tiempos de recambio.
- Ordenamiento de almacén robótico.
- Soldadura 2D, 3D, con restricciones…
- Etc. etc. etc…..

TSP es NP-difícil

- En todo lo anterior tratamos de formular los problemas como problemas de FMC que eran polinomiales y por lo tanto resolubles con métodos polinomiales que en teoría son mejores que los NP-difíciles.
- TSP es NP-difícil y NO es formulable como FMC.

Algoritmos Polinomiales y No Polinomiales

- Consideremos dos algoritmos que tardan $30.000\ n^3$ pasos y 2^n pasos, donde n es la cantidad de variables.
- Cuál es más rápido ??

- Supongamos que se pueden hacer 1.000.000.000 pasos por segundo (10^9 pasos/s)

# nodos	$30.000\ n^3$ pasos	2^n pasos
$n = 20,$	0,24 segundos	0,001 segundos
$n = 30,$	0,81 segundos	1,07 segundos
$n = 40,$	1,92 segundos	18,32 minutos
$n = 50,$	3,75 segundos	13,03 días
$n = 60,$	6,48 segundos	36 años

TSP como PAO

- Es posible formular el TSP como problema de asignación óptima:
- Pero no alcanza, por ?

$$\min \sum_{ij \in A} c_{ij} x_{ij}$$

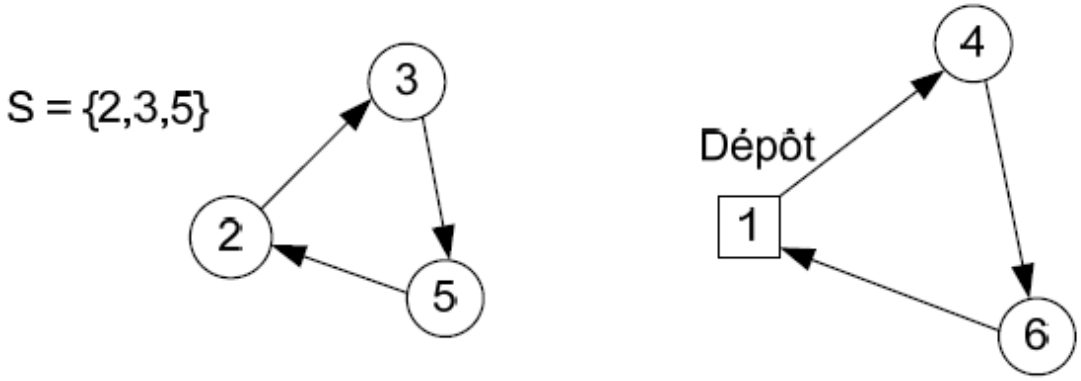
$$\sum_{j} x_{ij} = 1$$

$$\sum_{i} x_{ij} = 1$$

$$x_{ij} \geq 0 \quad \forall ij \in A$$

TSP. Sub-tours.

Sub-tours:



Eliminación de subtours

- Para eliminar subtours (Dantzig, Fulkerson, Johnson, 1954) se propone

$$\sum_{i,j \in S} x_{ij} \leq |S| - 1$$

Para todos los subconjuntos S que no contienen al 1 (nodo inicial o depósito).

Peligro !! La cantidad de restricciones crece con 2^n

Eliminación de subtours II

- En 1960 Miller-Tucker-Zemlin proponen agregar una nueva variable que representa el orden de visita y debe verificar

$$u_i - u_j + 1 \leq (n-1)\left(1 - x_{ij}\right)$$

Y para simplificar la resolución se puede hacer

$$u_1 = 1$$

$$2 \leq u_i \leq n \text{ si } i > 1$$

Eliminación de sub-tour III

- También hay una formulación sobre FMC con variables mixtas.
- Si se dualiza

Se puede descomponer el problemas en dos:

Uno de asignación y otro de flujo de mínimo costo.

$$\min \sum_{ij \in A} c_{ij} y_{ij}$$

$$\sum_{j} y_{ij} = 1$$

$$\sum_{i} y_{ij} = 1$$

$$Nx = b$$

$$x_{ij} \leq (n-1) y_{ij}$$

$$x_{ij} \geq 0$$

$$y_{ij} \in \{0,1\}$$

Eliminación de sub-tour IV

- ¿Qué restricciones son las mejores?
  - Número de restricciones Dantzig: 2^{n-1}, exponencial!
  - MTZ: $n.(n-1)$, polinomio (+ n variables y_i)
- De hecho, las restricciones Dantzig son mejores!
  - Podemos añadirlos gradualmente al modelo.
  - Incluso con sub-tours, el PL da una buena cota inferior.
- El modelo con restricciones MTZ:
  - Requiere las restricciones n (n-1) (si no es inviable).
  - El PL relajado da un límite inferior muy débil.

Un método óptimo para el TSP

```
construir PL de asignación sin restricciones Dantzig
repeat
   Resuelve el PL
   Si la solución contiene sub-tours
     Agregar a PL restricciones Dantzig solamente
     por estos sub-tours
   Fin si
until no hay sub-tours.
```

En general los sub-tours desaparecen rápidamente y usamos muy pocas restricciones Dantzig de las posibles 2^{n-1}.

Cotas inferiores del TSP

Sirven para evaluar las heurísticas.

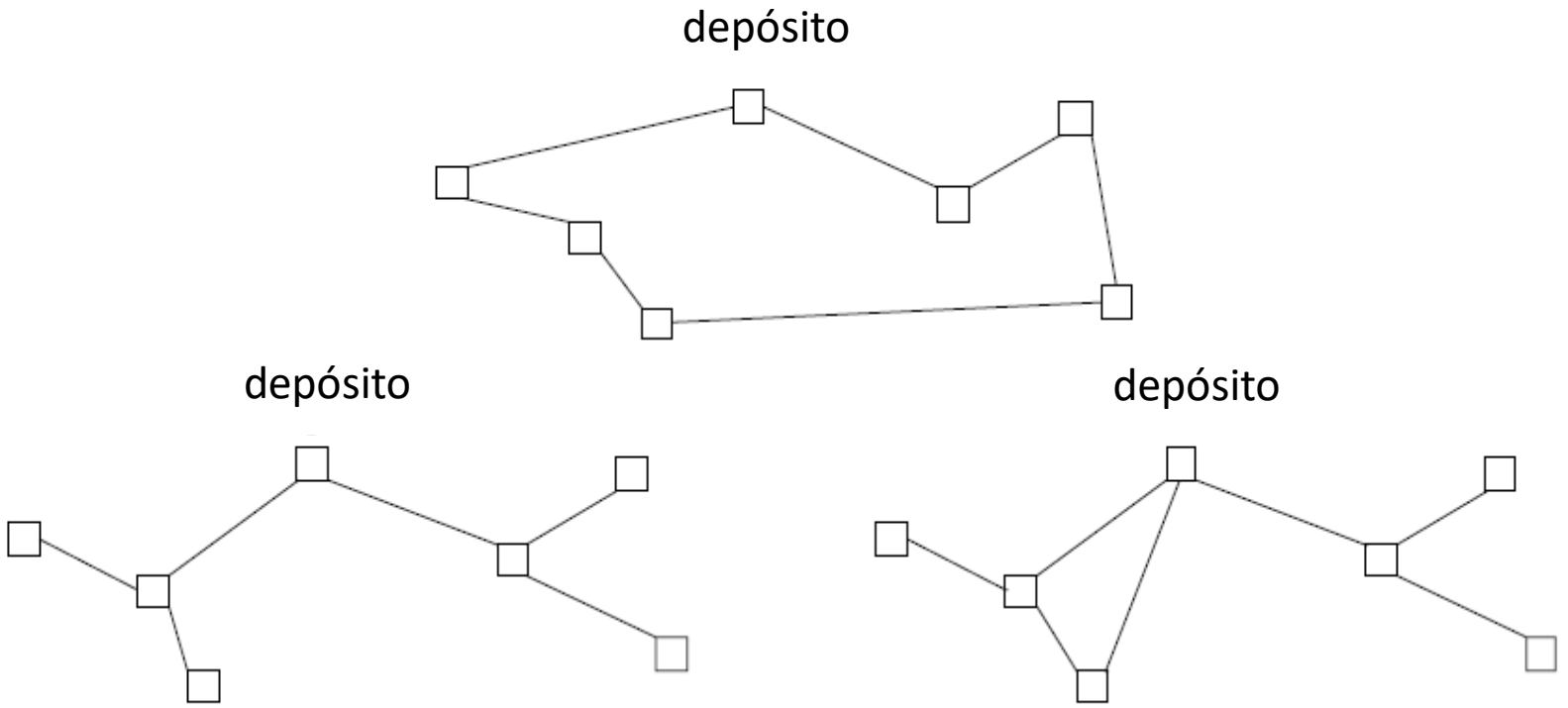
- **TSP "orientado" (asimétrico)**

  La solución de asignación PL da un cota inferior. Si no tiene sub-tours entonces es una solución óptima para el TSP.

- **TSP "no dirigido" (simétrico)**

  Un árbol es un grafo conectado sin ciclos. Un árbol de recubrimiento de un grafo H está formado por aristas de H y conecta todos los nodos de H.

Cotas inferiores del TSP



Ciclo óptimo (arriba), árbol de recubrimiento mínimo (izquierda) y 1-árbol óptimo (derecha)

Cotas inferiores del TSP

De hecho, si sacamos una arista de un tour óptimo T*, obtenemos un árbol de cobertura A particular (no ramificado), y C (A *) ≤ C (A) ≤ C (T*).

Mejora de la cota inferior de árboles: los 1-árboles.

Un 1-árbol es un árbol de cobertura al que se agrega una arista incidente al nodo del depósito, lo que crea un solo ciclo.

Un 1 árbol B* de costo mínimo da una cota mínima, porque cualquier recorrido es un 1-árbol particular, con todos los nodos en el ciclo único.

Cotas inferiores del TSP

El algoritmo de Prim calcula un árbol de cobertura A* de costo mínimo. Obtenemos un 1-árbol óptimo B* agregando a A* la arista incidente al depósito de menor valor.

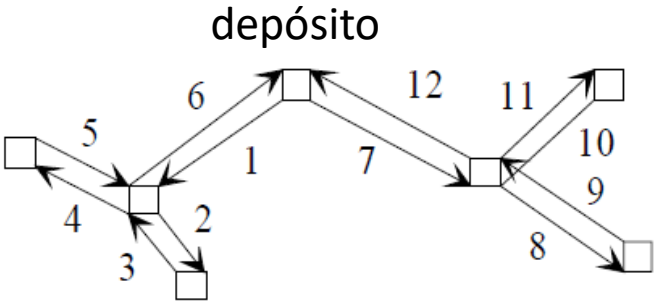
Partir de un árbol A reducido al depósito.

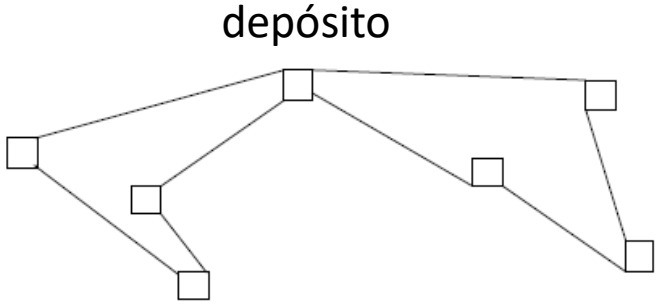
```
costo := 0
for contar := 1 a nc do
   busque la arista más pequeña (i,j) tal que
   i pertenezca A y j no pertenezca A
   añadir (i, j) a A
   costo := costo + D(i, j)
endfor
```

Podemos implementar este algoritmo en O (n^2).

Método de Christofides

Cálculo de un árbol de cobertura mínima (algoritmo de Prim).
Luego “tour del árbol" y eliminación de nodos ya visitados.



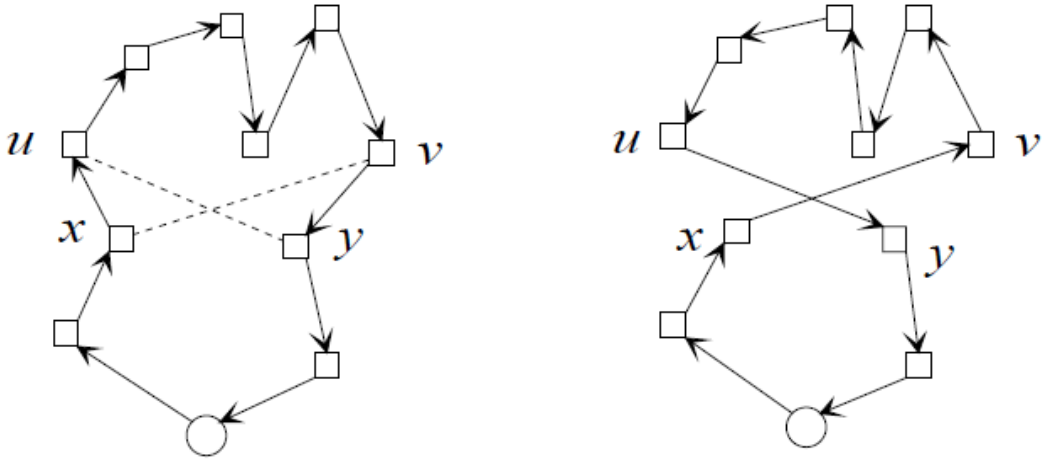


Resultado mediocre en promedio, pero podemos probar que nunca es más del doble de la óptima: ella calcula un recorrido T tal que $C(T) / C(T^*) \leq 2$.

Búsquedas locales

Poco a poco mejorar una solución de partida, calculada, por ejemplo, por una heurística simple. Ellas construyen un conjunto de soluciones sucesivas de costos decrecientes.

Transformación o "movimiento" 2-OPT:

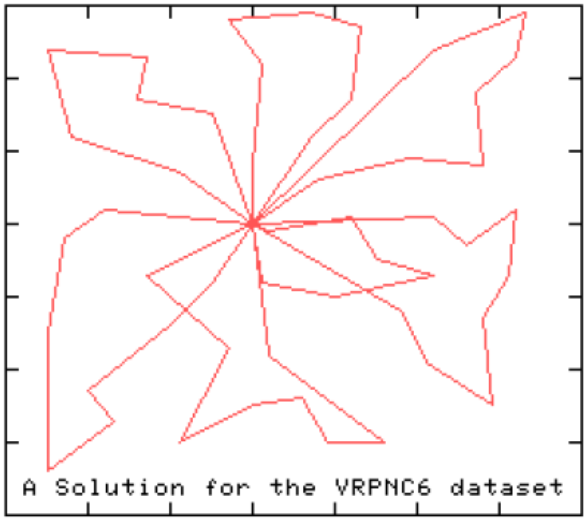


Implementación de 2-OPT

```
calcular una solución inicial S
repeat
   Δmin := ∞
   foreach par de aristas no consecutivas ((u, x), (v, y))
      Δ = D(x, v) + D(u, y) - D(x, u) - D(v, y)
      if Δ < Δmin then
         Δmin := Δ; p := ((u, x), (v, y))
      endif
   endfor
   if Δmin < 0 then
      cruzar las aristas de la pareja p en S
   endif
until Δmin = ∞
```

Ruteo de vehículos (VRP)

- El VRP (problema de enrutamiento del vehículo) extiende el TSP a varios excursiones. Es aún más difícil: no se conoce el óptimo para algunos problemas de 75 clientes.
- Cualquier heurística de TSP se puede utilizar para construir recorridos de VRP uno por uno. Creamos un recorrido cuando la capacidad del vehículo se agota. Este es el modo secuencial.



Ruteo de vehículos (VRP)

Si se impone *nv*, usamos el modo paralelo:

- Construimos nv rondas T_1, T_2, ..., T_{nv} en paralelo.
- Cada iteración de la heurística determina las mejores decisiones para T_1, T_2, ..., T_{nv}.
- Ejemplo con PPV: determinamos el próximo cliente para T_1, luego T_2, etc. hasta T_{nv} y comenzamos de nuevo.

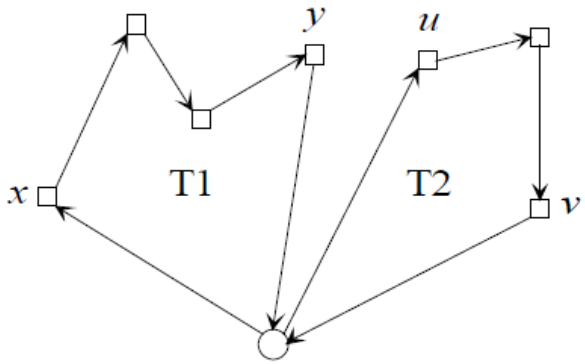
Este modo paralelo da mejores giras equilibradas.

Ruteo de vehículos (VRP)

Heurística de Clarke y Wright
O método de la margarita.

Principio:

- inicializar la margarita (n vueltas con un cliente cada una)
- Fusiones de 2 turnos siempre que proporcione una ganancia.
- En cada paso, la ganancia máxima se fusiona.



Ruteo de vehículos (VRP)

Si el vehículo del tour T_1 puede hacer T_2 sin regresar al depósito (demanda total ≤ W), la ganancia de la fusión es:

e (y, u) = D (y, s) + D (s, u) - D (y, u).

NB: 4 posibles fusiones si no hay red dirigida.

Una iteración del algoritmo prueba todos los pares de rondas y los 4 casos por par. Realiza la fusión de ganancia > 0 máximo, si existe. De lo contrario, es el final de algoritmo.
Cada iteración reduce el costo y también ahorra un recorrido, el método se utiliza para minimizar tanto el costo como la cantidad de vehículos utilizados.

Ruteo de vehículos (VRP)

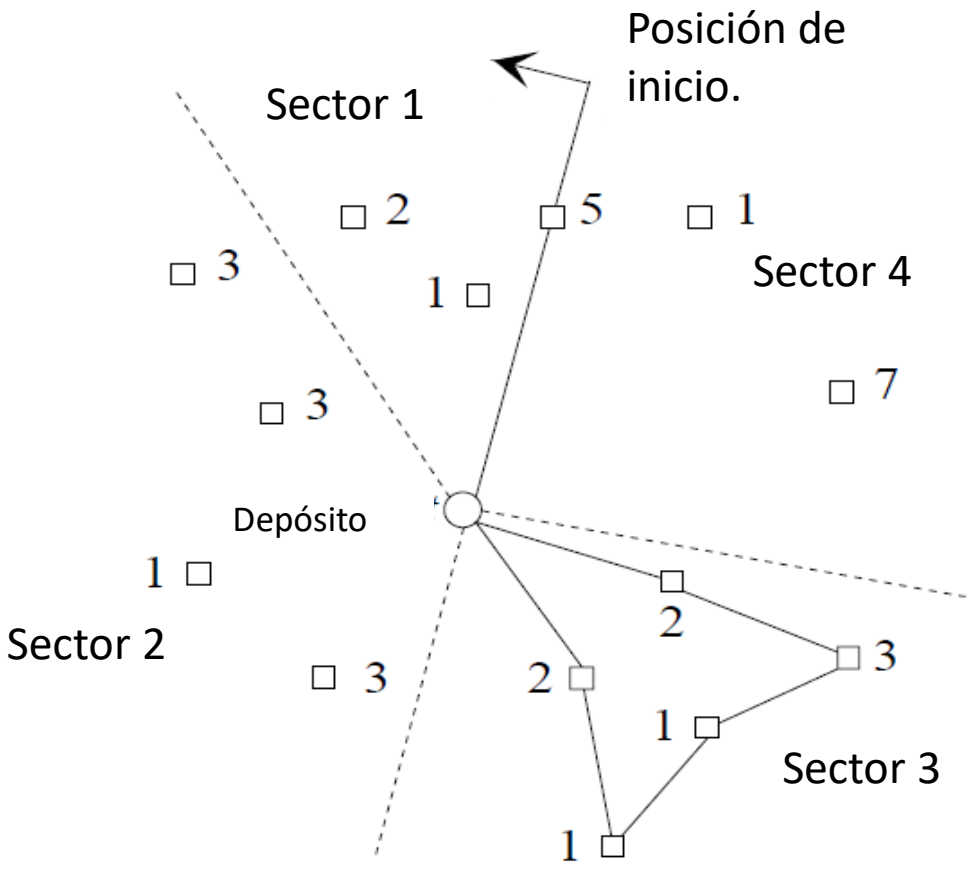
Heurística de Gillett y Miller

Método de cluster primero-ruta segunda: construimos grupos de clientes y calculamos un recorrido en cada grupo.

Principio:

- Clasificar a los clientes aumentando el ángulo polar / depósito.
- Escanearlos desde un cliente inicial cualquiera.
- Esto da sectores compatibles con la capacidad.
- Calcular una ronda por sector con heurística TSP.
- Mejorar cada recorrido con 2-OPT.
- Buenos resultados en el campo o en gran escala (departamento) y con un depósito central.

Ruteo de vehículos (VRP)



Ruteo de vehículos (VRP)

Heurística de Beasley
Ruta primero - segundo método de agrupación: calculamos un recorrido gigante visitando a todos los clientes (podemos usar cualquier método para TSP), luego dividimos este recorrido en recorridos factibles.

El corte se puede hacer de manera óptima, cf. siguiente ejemplo con un tour gigante S = (a, b, c, d, e) a nc = 5 clientes.

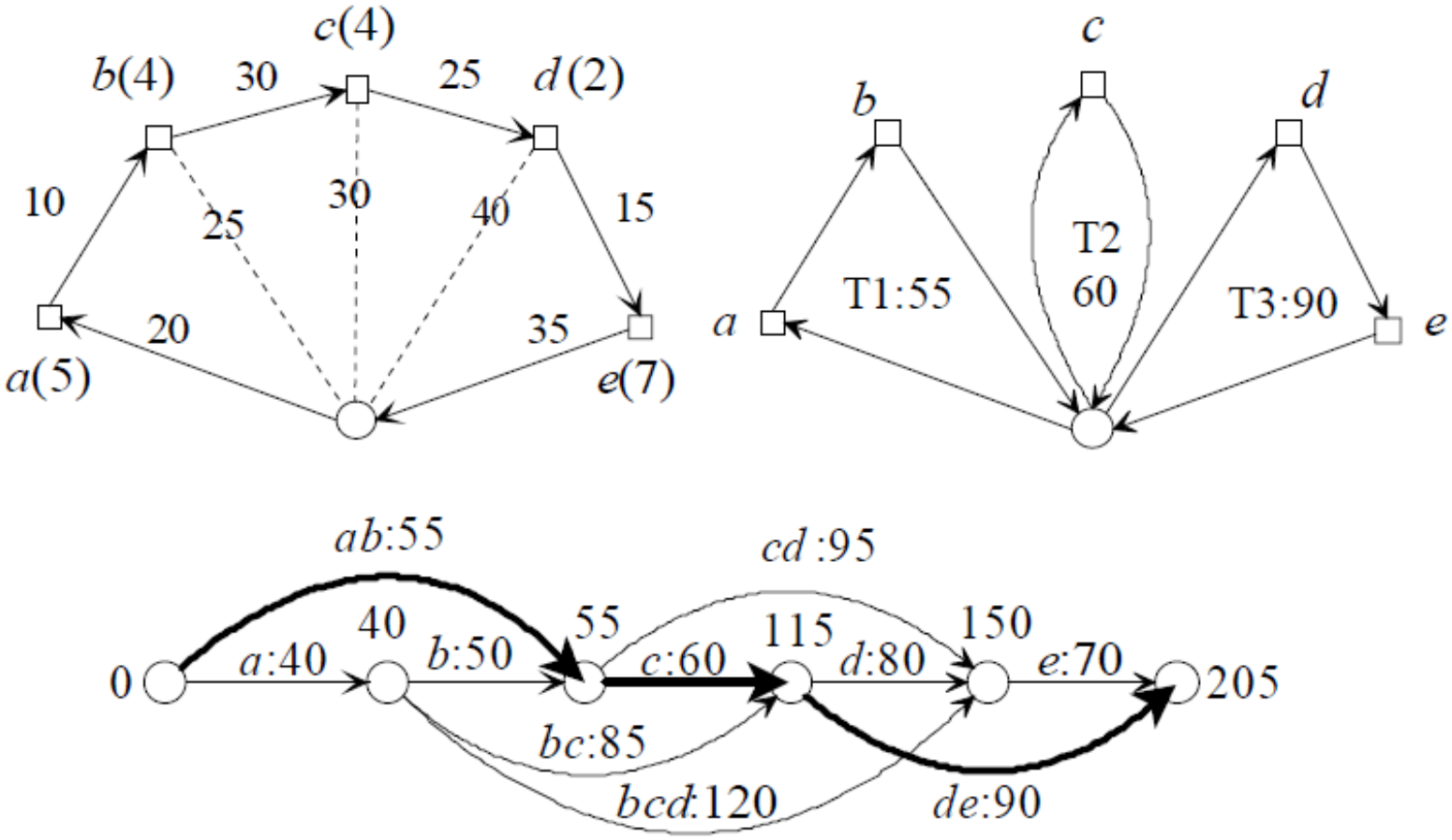
Las demandas están entre paréntesis. Los posibles viajes de ida y vuelta al depósito están punteados. La capacidad de los vehículos es W = 10.

Ruteo de vehículos (VRP)

Un grafo auxiliar H se construye con *nc* + 1 nodos indexados de 0 a *nc*. Si el subconjunto de clientes S_i ... S_j puede constituir un recorrido factible, se modela en H mediante un arco (*i-1, j*), evaluado por el costo del recorrido.

La división óptima en rondas corresponde a una ruta óptima desde el primer hasta el último nodo en H (en negrita). Conseguimos 3 tours, por un costo total de 205.

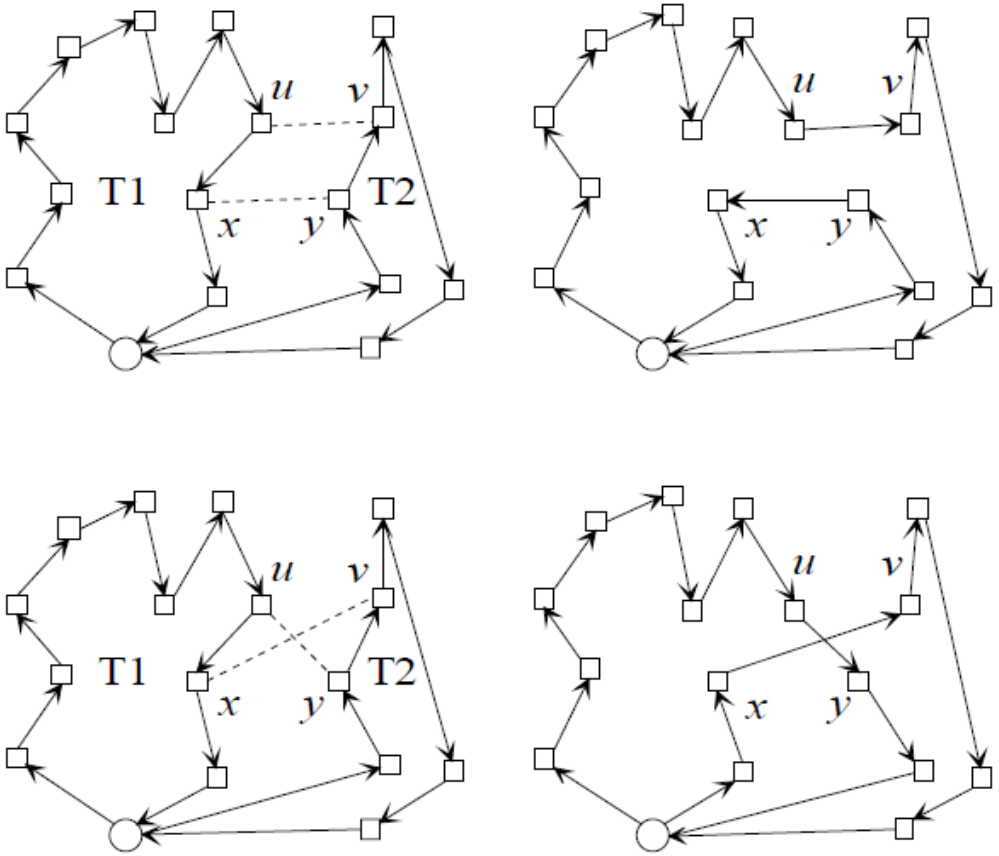
Ruteo de vehículos (VRP)



Ruteo de vehículos (VRP)

Búsquedas locales
Las búsquedas locales de PVC (2-OPT) se pueden aplicar recorrido por recorrido.
Pero también podemos aplicarlos a dos recorridos.

Ejemplo de 2-OPT en dos recorridos (2 casos):



Ruteo de vehículos (VRP)

Algunas complicaciones

Depósitos multiples
Si no hay restricciones de capacidad de depósito, cada cliente se asigna al depósito más cercano y se resuelve un VRP por depósito.

De lo contrario, asignamos clientes por PL, para minimizar la suma de las distancias a los depósitos respetando las capacidades de los depósitos (para formular esta PL como ejercicio).

Ruteo de vehículos (VRP)

Flota heterogénea
Disponemos de varios tipos de vehículos con diferentes capacidades. Se conoce el número de vehículos de cada tipo. La práctica demuestra que a menudo es mejor usar vehículos grandes primero.

Restricciones de acceso
Si ciertos arcos de la red están prohibidos para un tipo de vehículo (peso o altura limitados), es necesario construir para este tipo una distancia específica, calculando las RMC que no toman prestados arcos prohibidos. Esto se puede hacer eliminando temporalmente estos arcos de la red.