

Euristiche per VRP

(40)

Due classi principali:

(Cap. 5
Toth-Vigo)

1) euristiche classiche
(largamente usate)

2) metaeuristiche: esplorazione profonda
dello spazio delle soluzioni "più promettente";
qualità maggiore ma maggiore costo
computazionale

Euristiche classiche:

1) costruttive: costruiscono gradualmente
una soluzione ammissibile, senza fasi
di miglioramento

2) a due fasi:

fase I: clusterizzazione dei nodi in
rotte ammissibili

fase II: costruzione effettiva delle rotte

A loro volta divise in:

- "cluster-first, route-second"
- "route-first, cluster-second"

3) euristiche di miglioramento: eseguono passi per migliorare la soluzione corrente (es. scambio di nodi/archi entro una rotta o tre rotte)

OSS: per alcune euristiche il numero K di veicoli $\bar{k}$ da determinare ($\bar{k}$ è un dato di output)

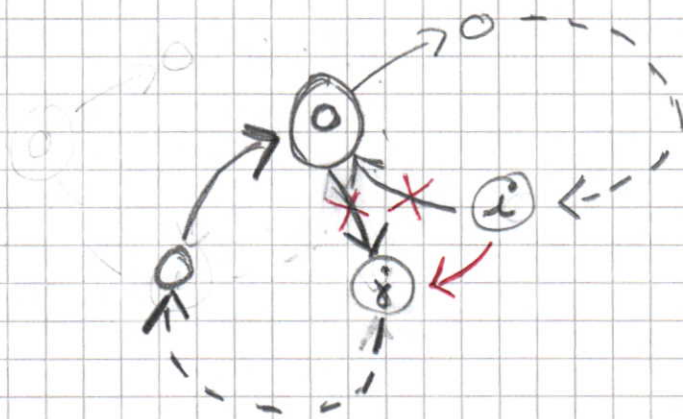
Algoritmo di "savings" di Clarke e Wright

(5.2.1 Toth-Vigo)

• Esempio di euristica costruttiva

Idea base:

date due rotte "ammissibili":



se tali rotte possono essere fuse ("merge")
in un'unica rotte "ammissibile" nel
modo indicato, i termini di costo si
ha un "risparmio" (saving):

$$S_{ij} = C_{io} + C_{oj} - C_{ij}$$

Algoritmo:

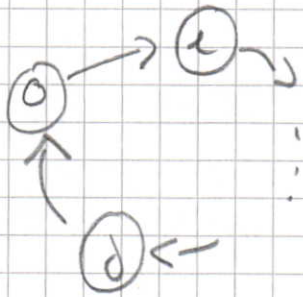
Passo 1 (calcolo dei saving):

- calcola $s_{ij} = c_{i0} + c_{0j} - c_{ij} \quad \forall i, j = 1, \dots, n, i \neq j$
- ordina $\{s_{ij}\}$ in modo non crescente
- crea n rotte iniziali del tipo:

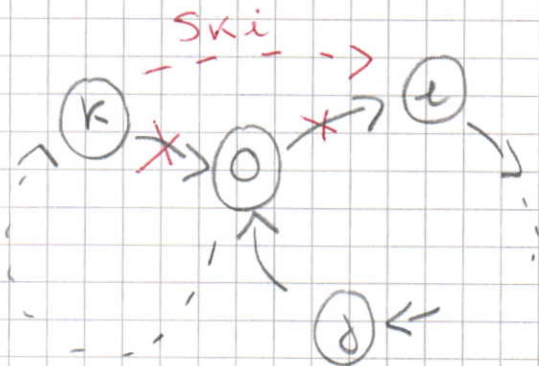


Passo 2 (fusione delle rotte):

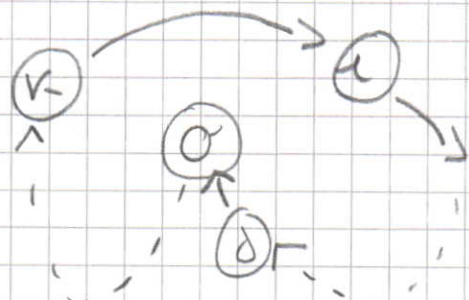
- per ogni rotte:



- determino il primo (nell'ordine) s_{ki} tale

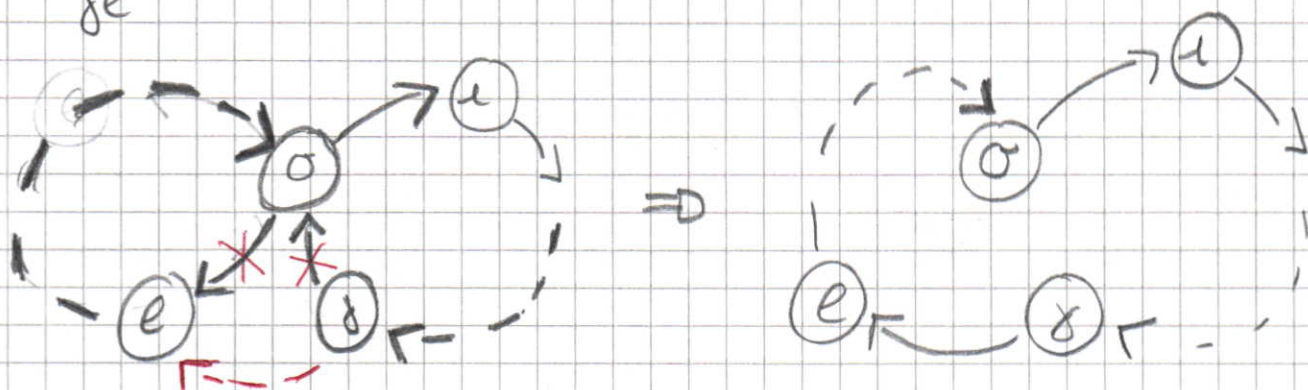


$\Rightarrow$



rotte "ammissibile"

- oppure determina il primo (nell'ordine) $s_{\gamma e}$ t.c.d.



$s_{\gamma e}$

rotta "ammissibile"

- implementa la fusione ("merge") ed itera sulla rotta corrente; se nessuna fusione genera una rotta "ammissibile", considera un'altra rotta e ripeti il

Passo 2.

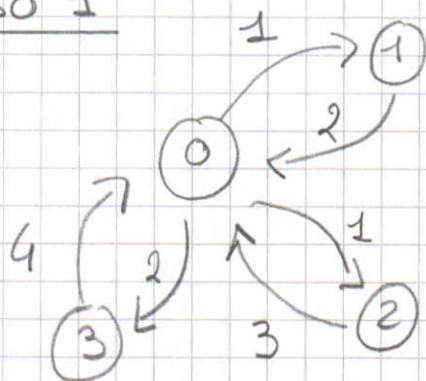
STOP quando nessuna fusione è ammissibile (varianti in funzione di K e riduzione costo)

(oppure: sono state determinate K rotte)

esempio

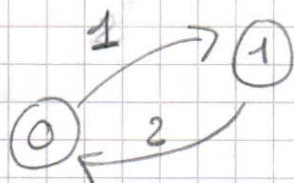
$n = 3$

Passo 1



calcolo $\{s_{ij}\}$

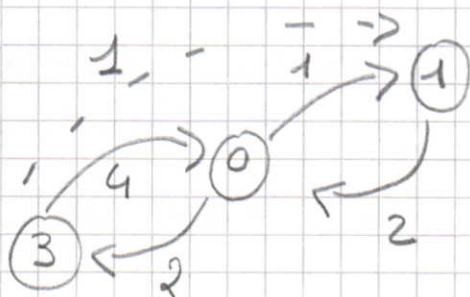
Passo 2 : consideriamo la rete $\rightarrow$



consideriamo i nodi "adiacenti" il deposito (nell'esempio ①), e determiniamo il

massimo s_{k1} oppure s_{1k} che permette una fusione di rete

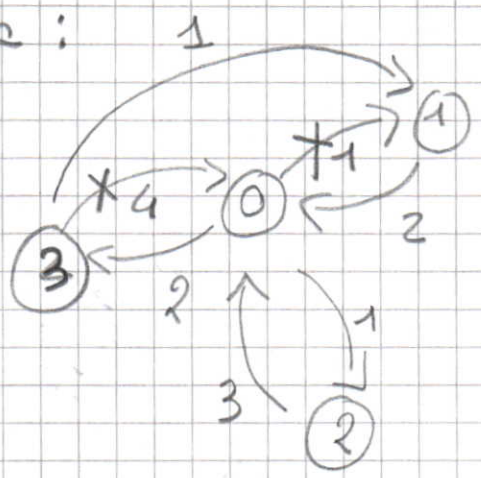
se fosse



$$\begin{aligned} s_{31} &= c_{30} + c_{01} - c_{31} \\ &= 4 + 1 - 1 = 4 \end{aligned}$$

NB : vale la disuguaglianza triangolare

albero :



$S_{31} = 4$ rappresenta il risparmio totale
($\equiv$ riduzione del costo totale) derivante
dall'operazione di fusione

Se ci fossero requisiti di ammissibilità
(es. capacità) vanno verificati prima di
eseguire la fusione

- o
- o itero
- o
- o

Euristiche di inserzione sequenziale di Mole e Gamberoni

(5.2.4.1 Toth-Vigo)

- Altro esempio di euristica costruttiva

h e μ parametri

- $\alpha(i, k, \delta) = c_{ik} + c_{k\delta} - h c_{i\delta}$
- $\beta(i, k, \delta) = \mu c_{ok} - \alpha(i, k, \delta)$

Passo 1 (Inizializzazione): costruire una
rotta (o, k, o) , con k nodo non appartenente
mente ad alcuna rotta



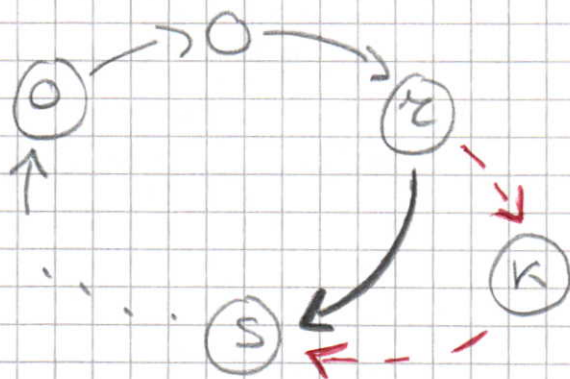
Passo 2 (Estensione):

- per ogni nodo k non appartenente
ad alcuna rotta e "adiacente" alla
rotta in fase di costruzione determinare

$$\alpha^*(i_k, k, j_k) =$$

$$= \min \{ \alpha(i, k, s) \}$$

miglior costo
di inserzione
di k



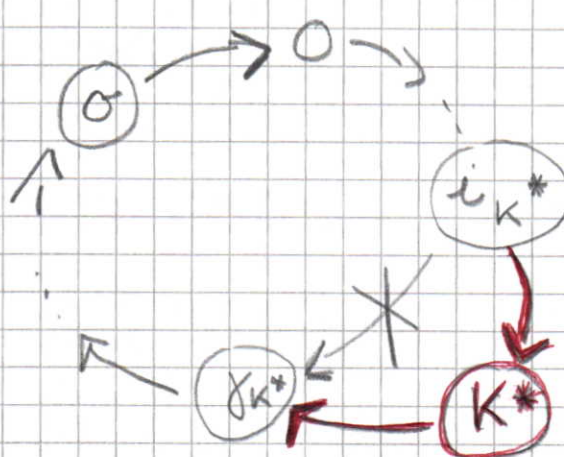
$$\alpha(i, k, s) =$$

$$c_{ik} + c_{ks} - \cancel{c_{is}}$$

NB: in presenza di requisiti di ammissibilità,
va determinato il miglior costo di
inserzione ammissibile

• Inserisci k^* tale che

$$\beta^*(i_{k^*}, k^*, j_{k^*}) = \max \{ \beta(i_k, k, j_k) \}$$



$$\beta(i_k, k, j_k) =$$

$$MC_{ok} - \alpha(i_k, k, j_k)$$

- se nessun nodo K^* può essere inserito nella rete corrente, vai al Passo 1 (Inizializzazione nuova rete)

Passo 3 (Ottimizzazione): "ottimizza"

la rete corrente e vai al Passo 2;

STOP quando tutti i nodi appartengono a una rete

Possibili scelte dei parametri;

- 1) $\lambda = 1, \mu = 0$: l'algoritmo inserisce il nodo K^* che produce il minimo costo (distanza) extra
- 2) $\lambda = \mu = 0$: l'algoritmo inserisce il nodo K^* avente la minima somma delle distanze dai nodi tra cui è inserito
- 3) $\mu = +\infty$ e $\lambda > 0$: si inserisce il nodo più lontano dal deposito

Algoritmo di Bramel e

45

Simchi-Levi

(5.3.1.3 Toth-Vigo)

• Esempio di euristica a due fasi

I^o FASE (Individuazione di cluster $\equiv$ insieme di nodi serviti da uno stesso veicolo)

Si risolve un problema di p -median

con $p = K$ (numero veicoli) e $I = J$

(con l'aggiunta del vincolo di capacità C per ogni "deposito" = "cluster")

esempio

$K = 2$, $C = 5$, $n = 6$

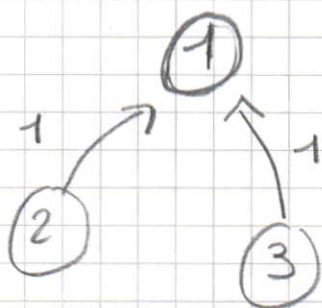
	1	2	3	4	5	6
1	0					
2	1	0				
3	1		0			
4				0		
5				1	0	
6				1		0

1	2	3	4	5	6
1	2	2	3	1	1

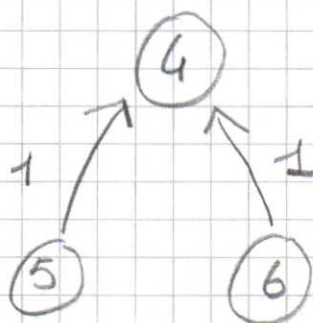
d

$C_{ij} = 2$ altrimenti

Una soluzione ottima:



I° cluster



II° cluster

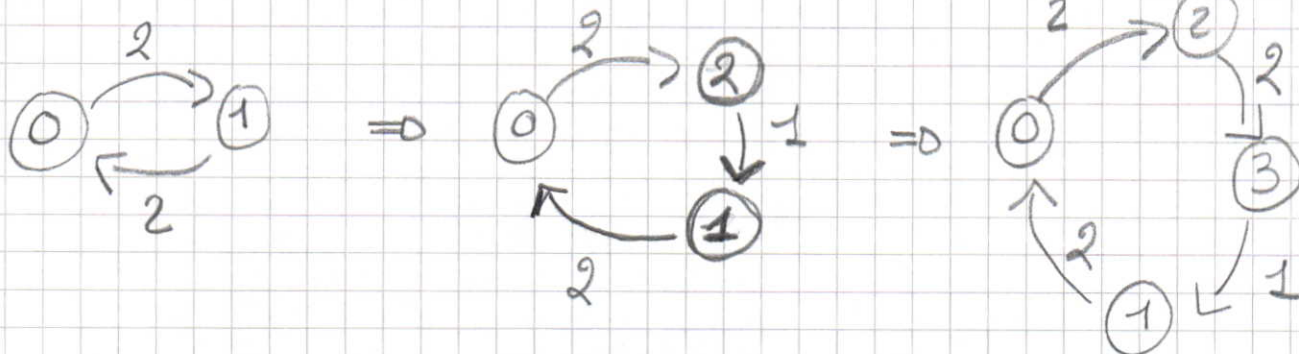
$$(d_1 + d_2 + d_3 \leq c = 5) \quad (d_4 + d_5 + d_6 \leq c = 5)$$

II° FASE (Individuazione di una zotte per ogni cluster)

es mediante e' euristico di misurazione sequenziale, caso $\lambda = 0, \mu = 0$ (diversità rispetto a Toth - Vigo)

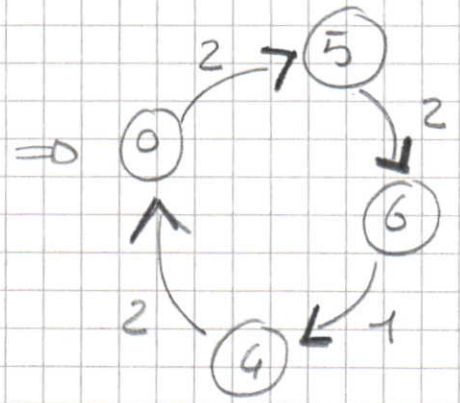
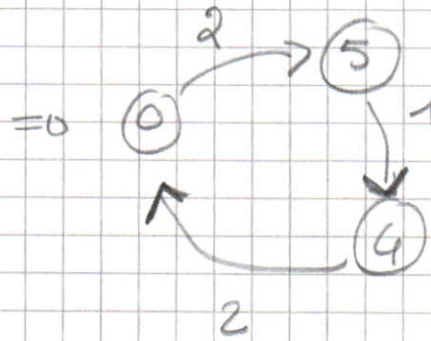
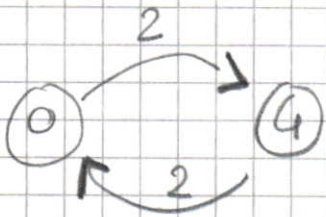
esempio (cont.) : nota : dentro ci gioca il deposito

I° cluster



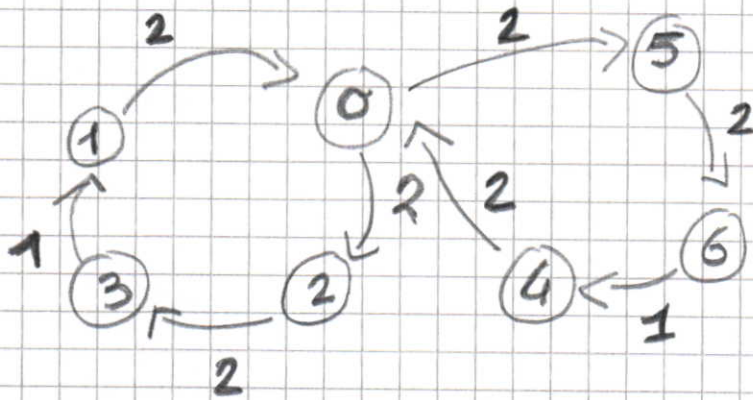
II° cluster

(analogo)



46

Soluzioni restituite:



costo: 14

