

Makespan con set-up dipendenti dalla  
sequenza

$$1 / s_{jk} / C_{max}$$

$$1/s_{jk}/C_{max}$$

- Un tempo di riattrezzaggio (*set-up*)  $s_{jk}$  è richiesto fra il processamento di  $j$  e quello di  $k$ . In questo caso,  $C_{max}$  dipende dalla sequenza.

- In generale il problema è NP-hard in senso forte.

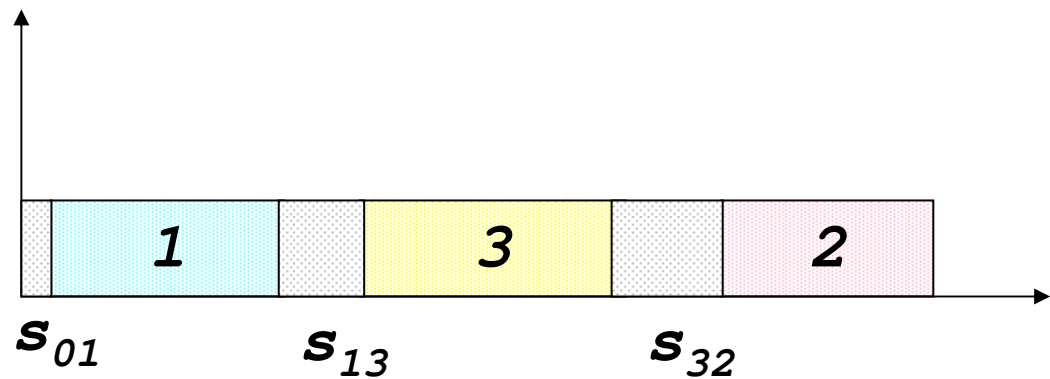
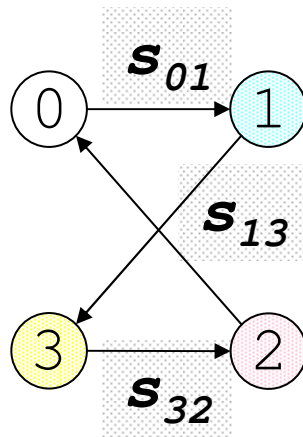
- **Caso speciale:** a ciascun job  $j$  sono associati due parametri  $a_j$  e  $b_j$ , ed il tempo di set-up è dato da  $s_{jk} = |a_k - b_j|$

**Motivazione:** dopo il processamento di  $j$  la macchina è lasciata nello stato  $b_j$  e, per poter processare  $k$ , deve essere portata nello stato  $a_k$ . Il tempo richiesto dipende dalla “distanza” fra i due stati.

# modello *TSP*

Ad ogni job corrisponde una città, con l'aggiunta di una città fittizia  $0$ , con parametri  $a_0, b_0$ . Si assume  $b_0 \leq b_1 \leq \dots \leq b_n$

Le sequenze di job corrispondono ai cicli hamiltoniani (*tour*)



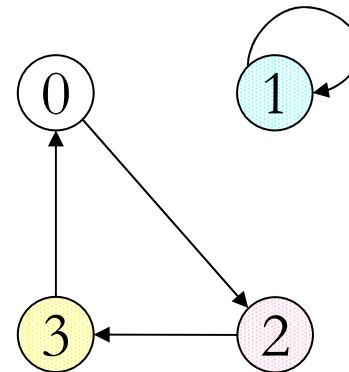
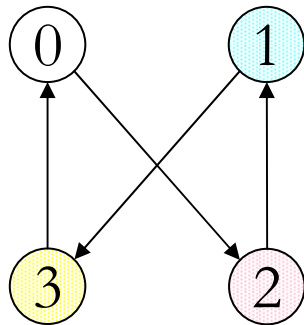
# modello *TSP*

*funzione permutazione*:  $\Phi: j \mapsto \varphi(j)$  associa al job (città)  $j$  il suo successore.

Non tutte le permutazioni corrispondono a tour:

$$\{0, 1, 2, 3\} \mapsto \{2, 1, 3, 0\}$$

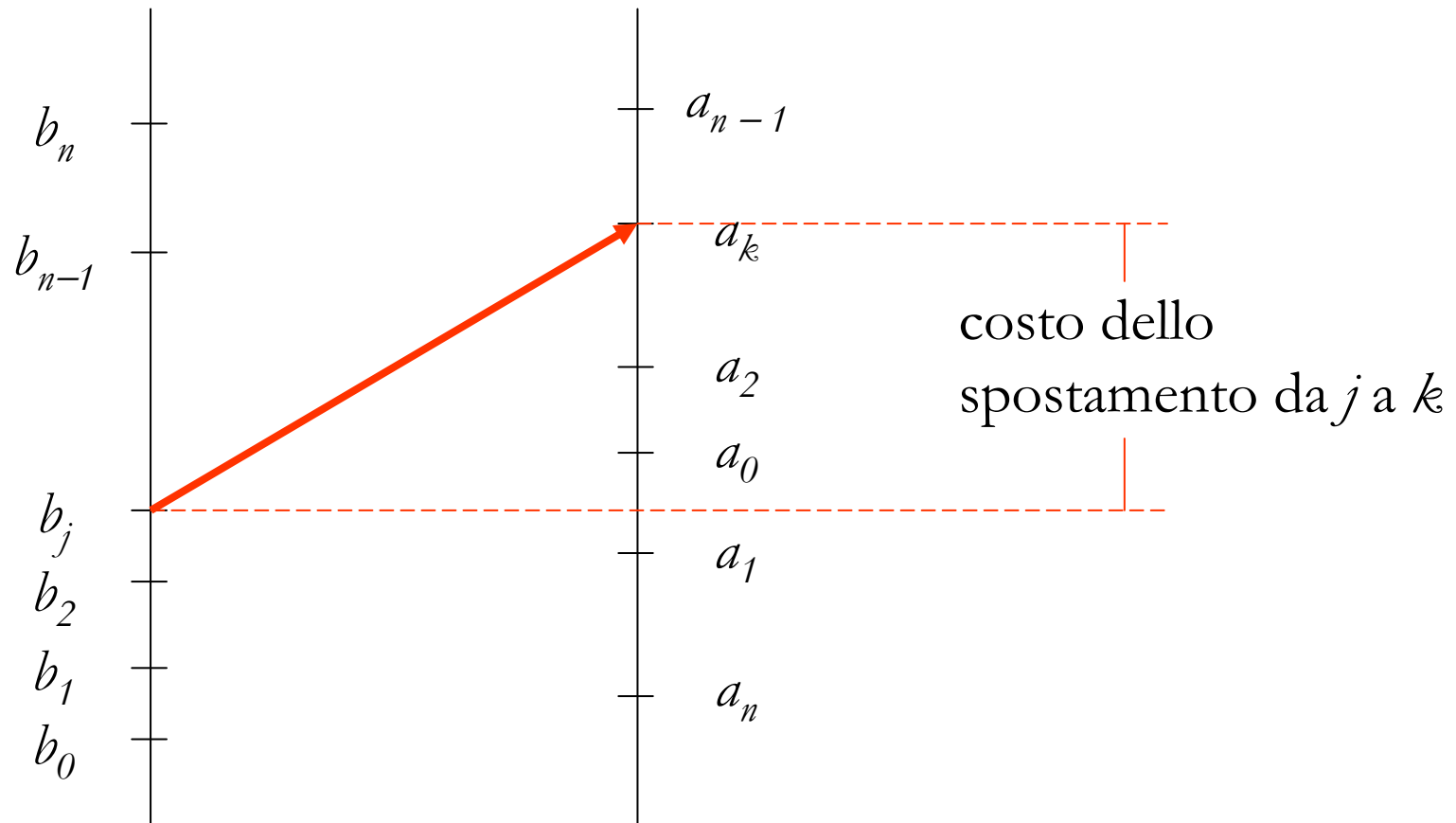
$$\{0, 1, 2, 3\} \mapsto \{2, 3, 0\} \{1\}$$



sub-tour

# rappresentazione di una permutazione

insieme di frecce che connettono  $b_j, j=0, \dots, n$  ad  $a_k, k=0, \dots, n$



costo della permutazione: somma delle componenti verticali delle frecce

## swap

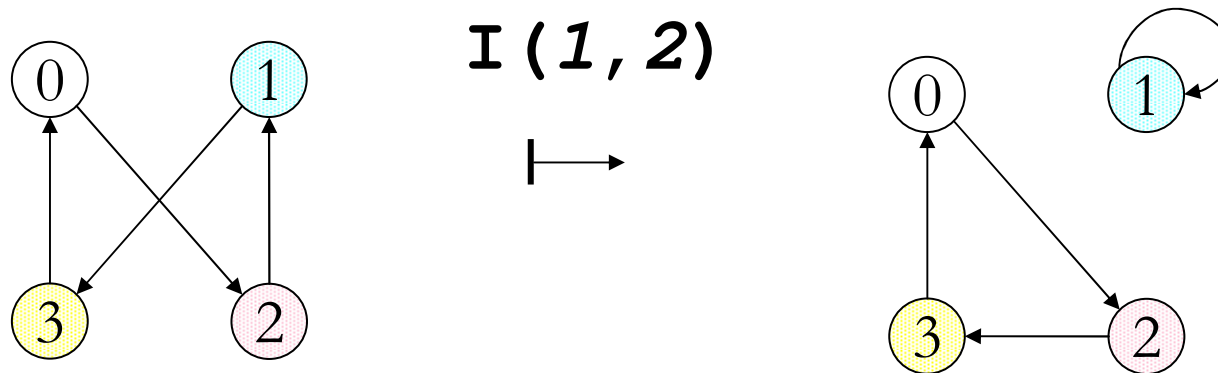
**Definizione.** Data una permutazione  $\Phi$ , uno swap  $I(j,k)$  produce una nuova permutazione  $\Phi' = \Phi I(j, k)$  tale che:

$$\begin{aligned}\varphi'(k) &= \varphi(j), \\ \varphi'(j) &= \varphi(k),\end{aligned}$$

$$\varphi'(l) = \varphi(l), \quad l \neq j, k$$

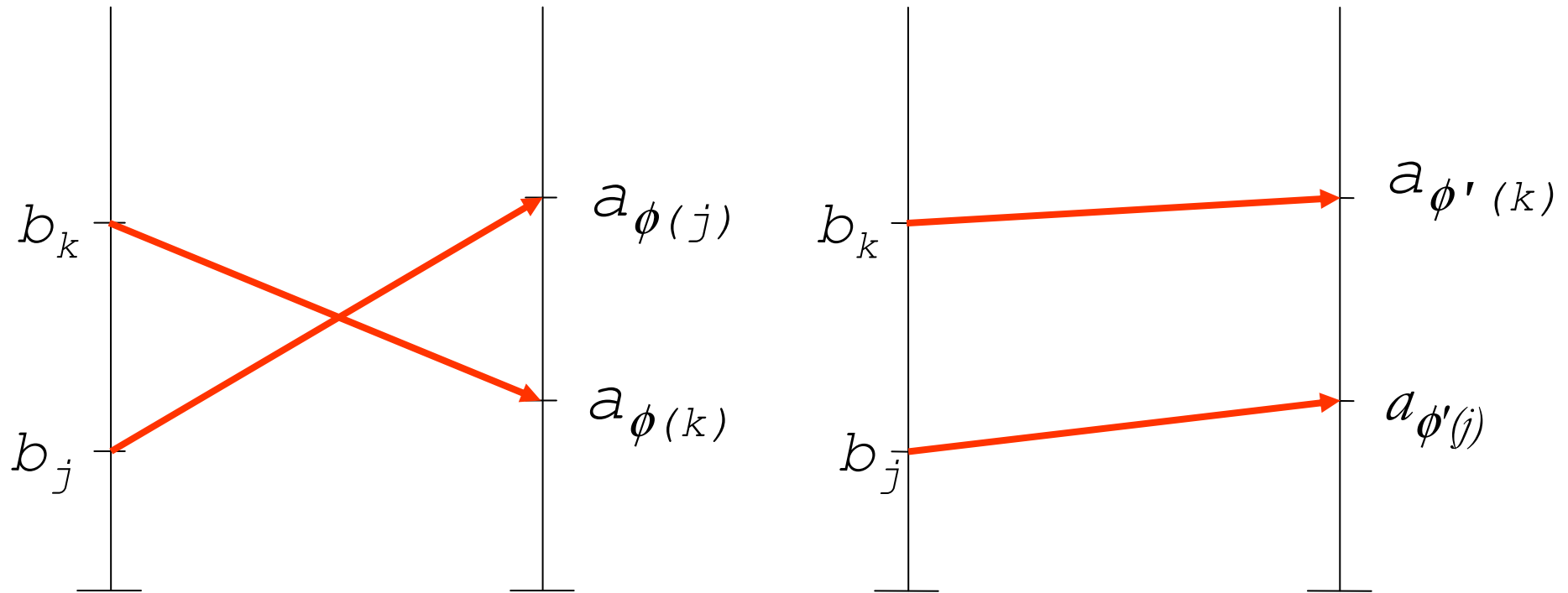
Uno swap  $I(j, k)$  scambia il successore di  $j$  con quello di  $k$  lasciando il resto della permutazione inalterata

## Effetto di uno swap



**Osservazione.** Se  $j$  e  $k$  appartengono allo stesso ciclo in  $\Phi$ , allora  $I(j, k)$  produce due sotto-cicli; altrimenti,  $I(j, k)$  fonde due i due sotto-cicli che contengono  $j$  e  $k$ .

## Swap $\mathbf{I}(j, k)$



$$\begin{aligned} \varphi'(k) &= \varphi(k) & \mathbf{I}(j, k) &= \varphi(j) \\ \varphi'(j) &= \varphi(j) & \mathbf{I}(j, k) &= \varphi(k) \end{aligned}$$

**Osservazione.** Se le due frecce si intersecano in  $\Phi$ , allora non si intersecano in  $\Phi' = \Phi \mathbf{I}(j, k)$  e viceversa.



## Variazione di costo

$c_{\Phi} I(j, k)$  : variazione del costo della permutazione  $\Phi$  quando si applica lo swap  $I(j, k)$

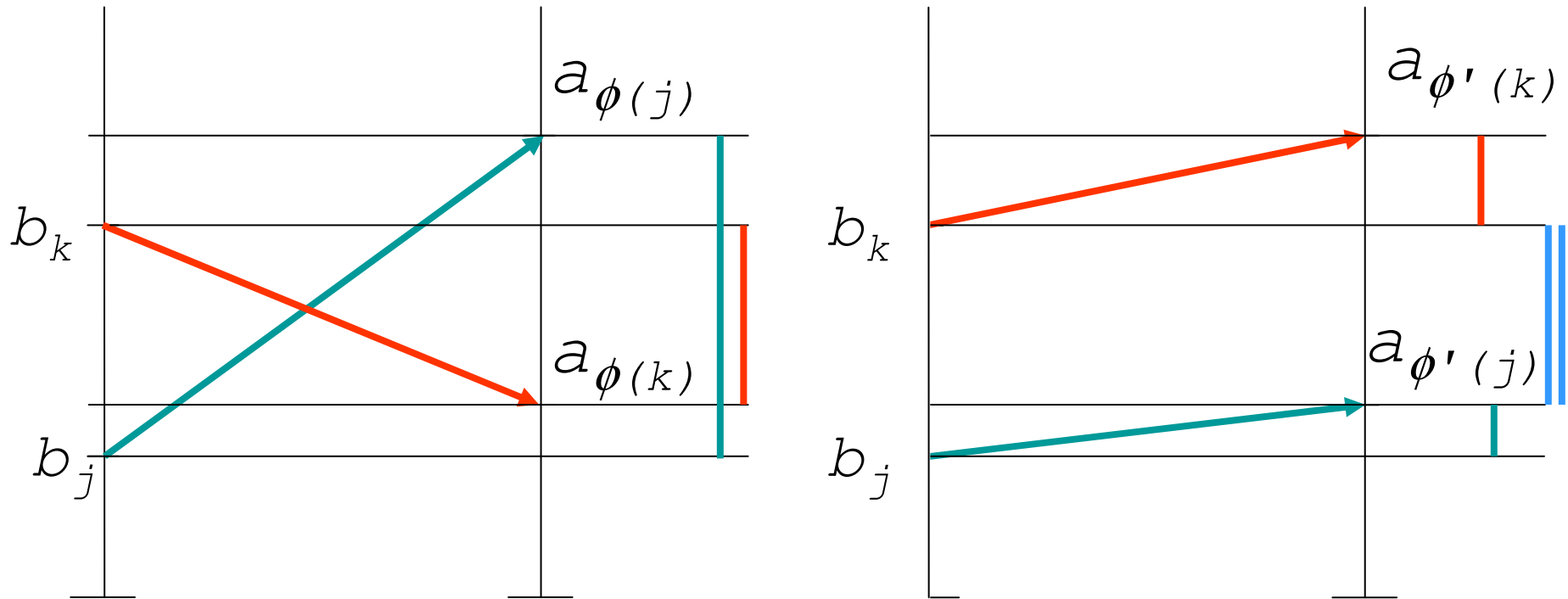
**Lemma 0.** Se lo swap  $I(j, k)$  fa intersecare due frecce non intersecantisi, allora il costo della permutazione aumenta e viceversa. Il valore di tale variazione è pari al doppio della lunghezza dell'intersezione degli intervalli  $[b_j, b_k]$  e  $[a_{\Phi(j)}, a_{\Phi(k)}]$ :

$$c_{\Phi} I(j, k) = \left\| [b_j, b_k] \cap [a_{\Phi(j)}, a_{\Phi(k)}] \right\|$$

in cui:

$$\left\| [a, b] \right\| = \begin{cases} 2(b - a) & \text{se } b \geq a \\ 2(a - b) & \text{se } b < a \end{cases}$$

## Variazione di costo

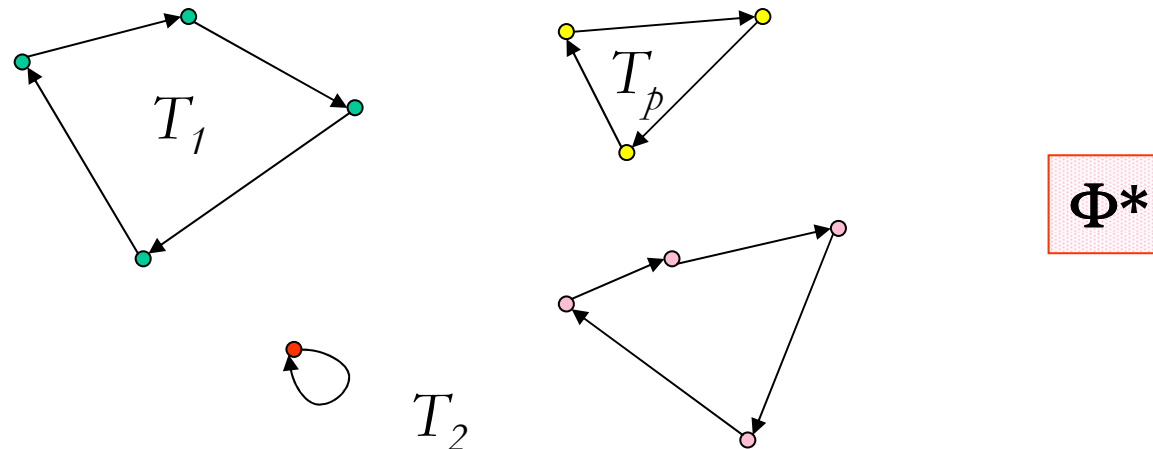


$$c_{\Phi} I(j, k) = \left\| [b_j, b_k] \cap [a_{\Phi(j)}, a_{\Phi(k)}] \right\|$$

# Conseguenza

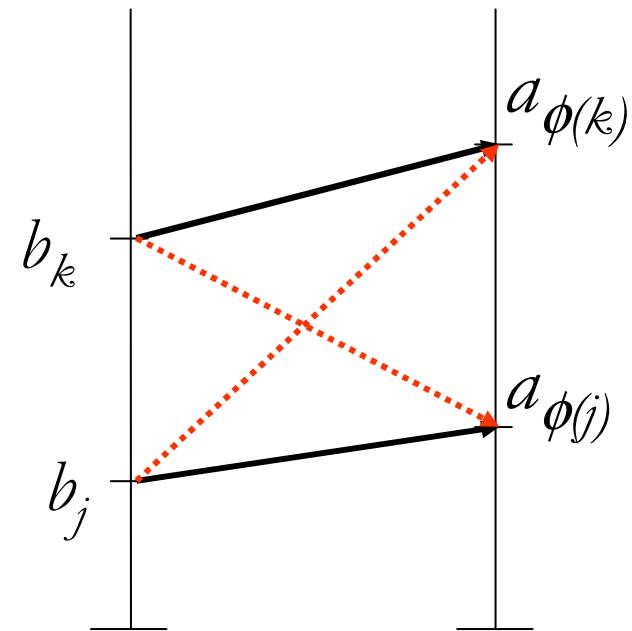
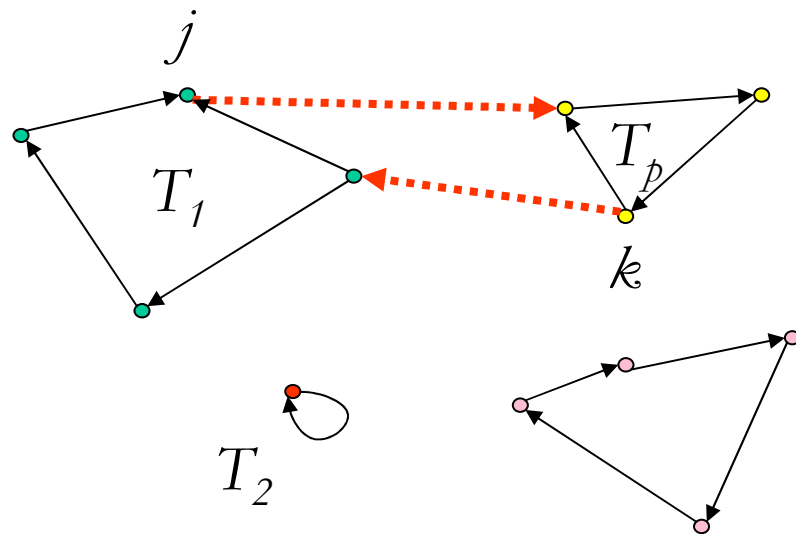
**Corollario.** Una permutazione ottima  $\Phi^*$  si ottiene quando  $b_j \leq b_k \Rightarrow a_{\Phi^*(j)} \leq a_{\Phi^*(k)}$ , cioè quando le frecce del diagramma sono tutte mutuamente non intersecantisi

La permutazione  $\Phi^*$  non corrisponde, in generale, ad un tour ammissibile, ma ad un insieme di sub-tour  $T_1, \dots, T_p$ . Tuttavia, il valore di  $\Phi^*$  è un lower bound del valore ottimo.



# Algoritmo

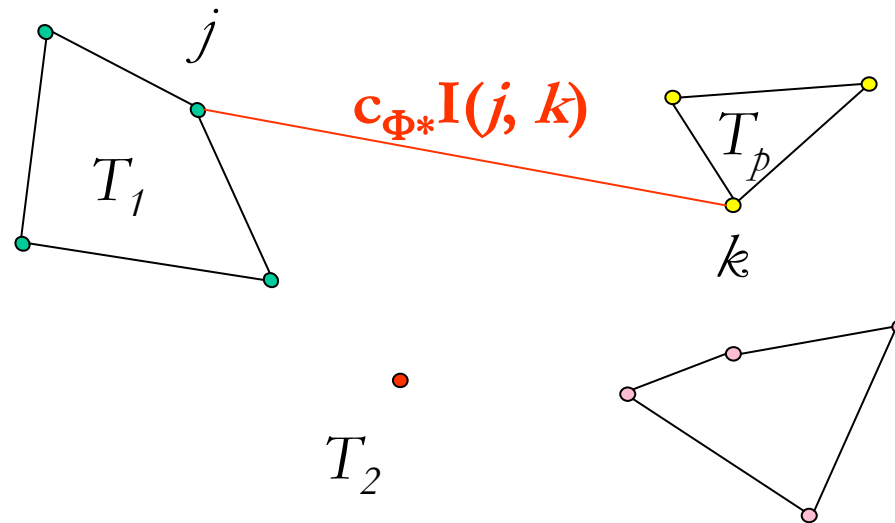
Applicando a  $\Phi^*$  uno swap  $I(j, k)$  in cui  $j$  e  $k$  appartengono a sub-tour distinti, i due sub-tour vengono fusi, ma il costo della permutazione aumenta (si rendono intersecantisi frecce che non lo erano).



Il metodo consiste nel cercare la sequenza di swap tale da costruire un tour ammissibile minimizzando l'aumento di costo rispetto a  $\Phi^*$

# Algoritmo

Consideriamo il grafo supporto (non orientato) corrispondente a  $\Phi^*$ . Connettere due sotto-cicli richiede l'inserimento di nuovi archi. A ciascun arco con estremi in cicli diversi si assegna un peso pari al costo del corrispondente swap.

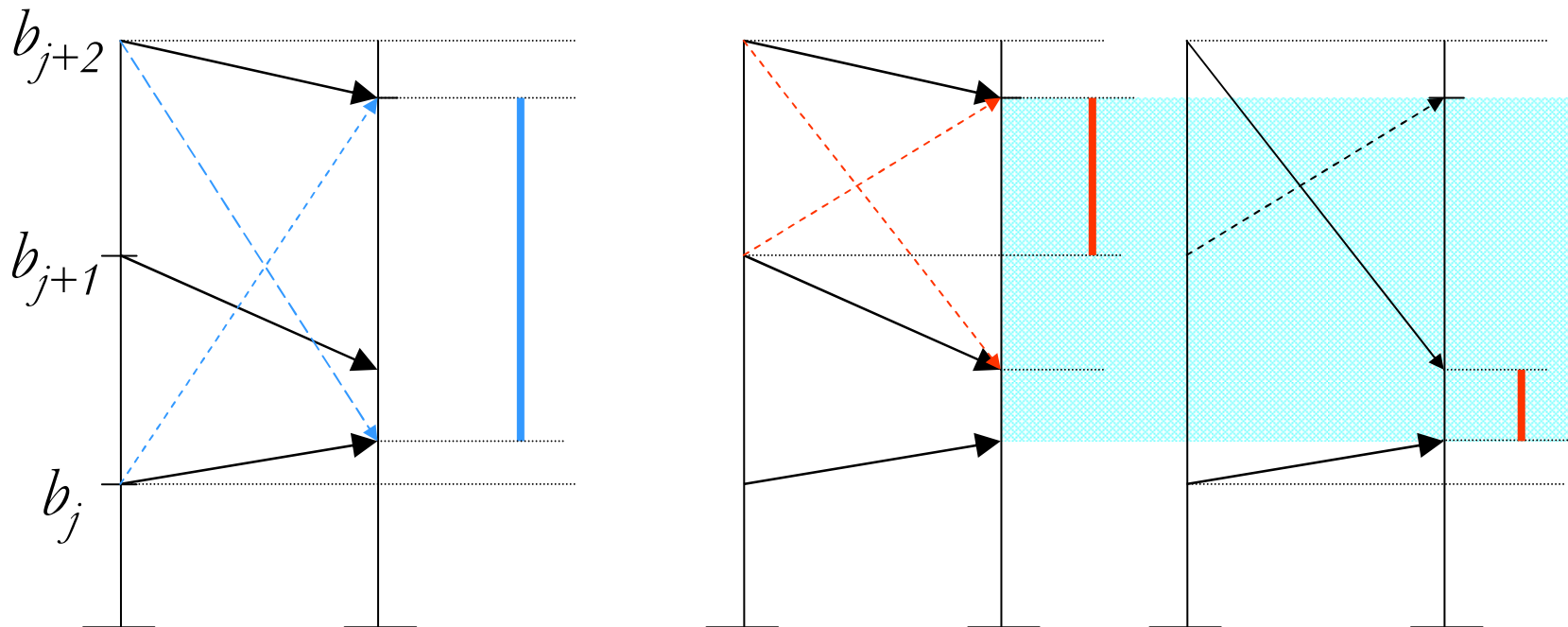


Il grafo connesso (cioè il tour ammissibile) di costo minimo si ottiene scegliendo iterativamente l'arco di costo minimo che connette cicli distinti (algoritmo *greedy*)

# Scelta degli archi

**Lemma 1.** Una collezione di archi che connette il grafo supportato a costo minimo contiene solo archi del tipo  $(j, j+1)$

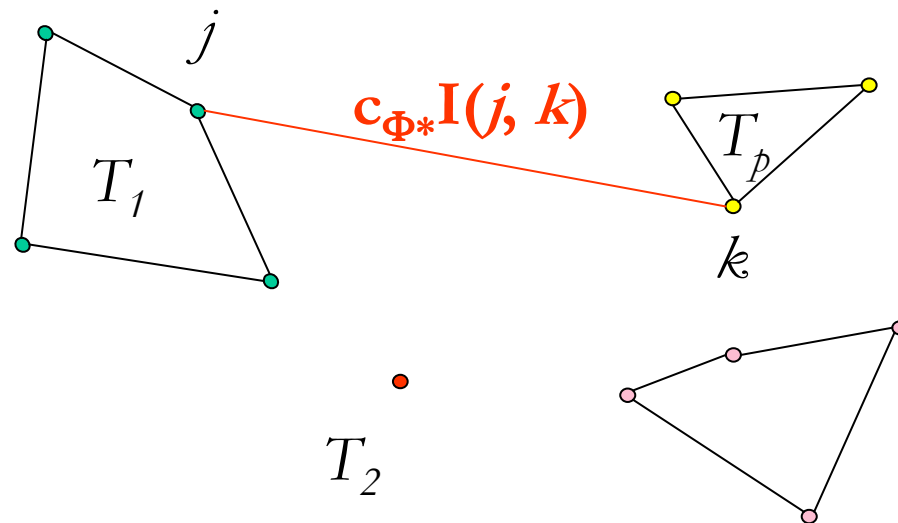
**Dimostrazione.** Il costo degli archi è calcolato dal Lemma 0 rispetto a  $\Phi^*$ . Il costo di swap di due frecce non adiacenti è almeno pari alla somma dei costi di swap di tutte le frecce comprese fra esse:



# Difficoltà

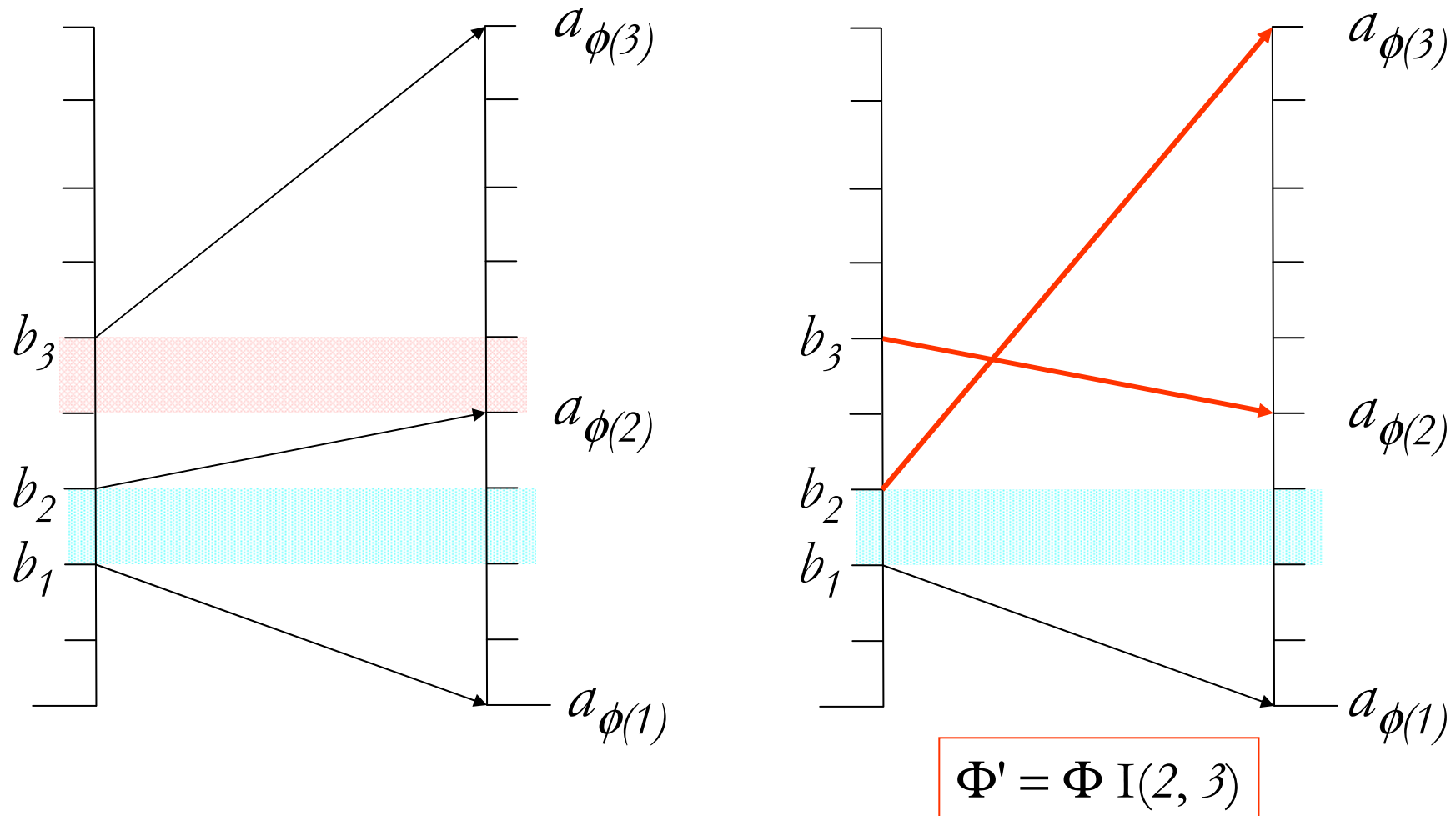
I costi  $c_{\Phi^*} I(j, k)$  assegnati agli archi inter-ciclo del grafo supporto sono stati calcolati con l'assunzione che i corrispondenti swap fossero applicati a  $\Phi^*$ , in cui non ci sono intersezioni fra frecce.

Al contrario, man mano che gli swap sono eseguiti, tale ipotesi non è più valida



## Esempio

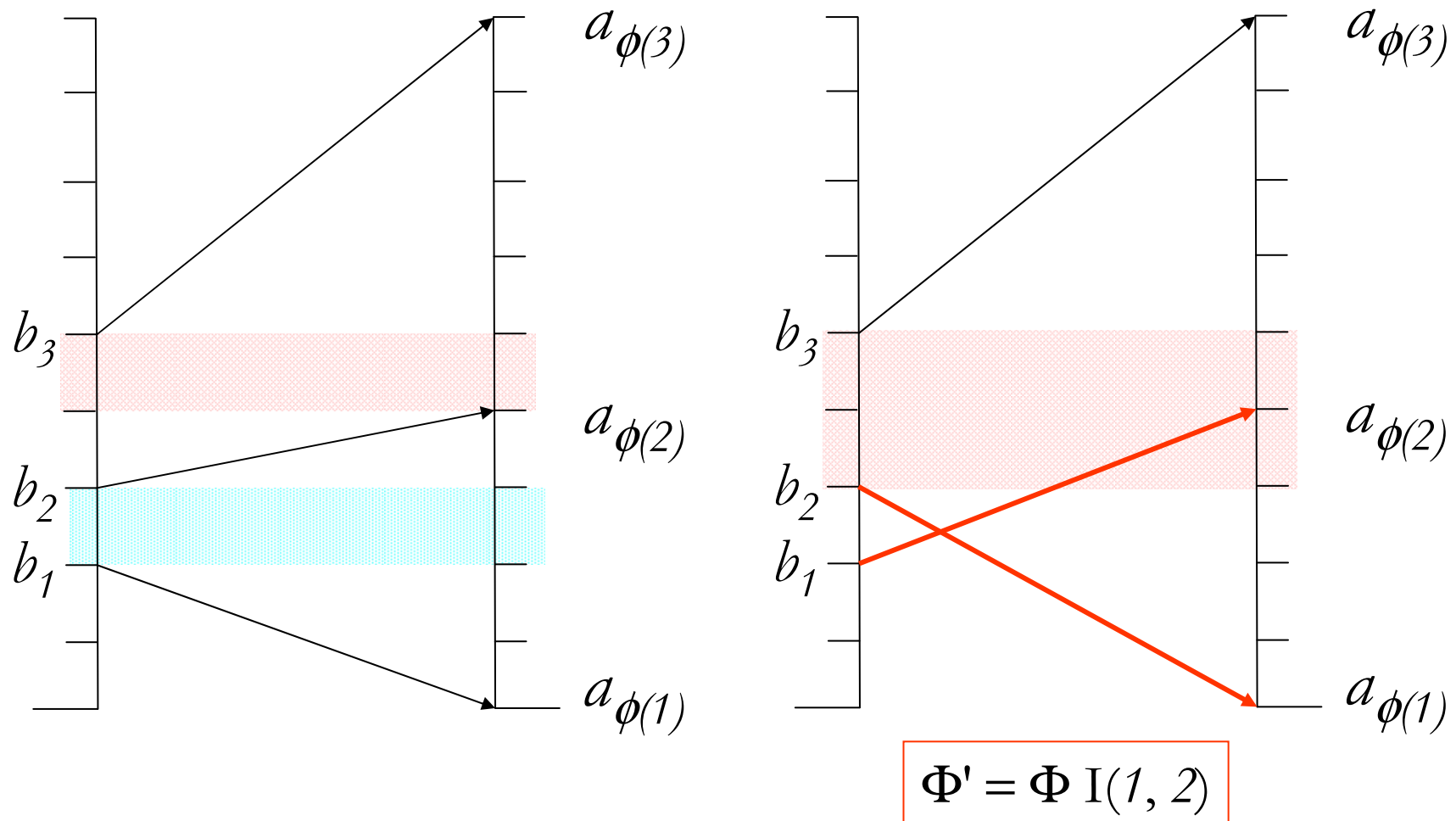
Nella situazione di figura si ha  $c_{\Phi}I(1, 2)=2$  e  $c_{\Phi}I(2, 3)=2$ . Se eseguiamo prima lo swap  $I(2, 3)$  e poi  $I(1, 2)$ , il costo di quest'ultimo in  $\Phi'$  rimane pari a  $c_{\Phi}I(1, 2)$ .





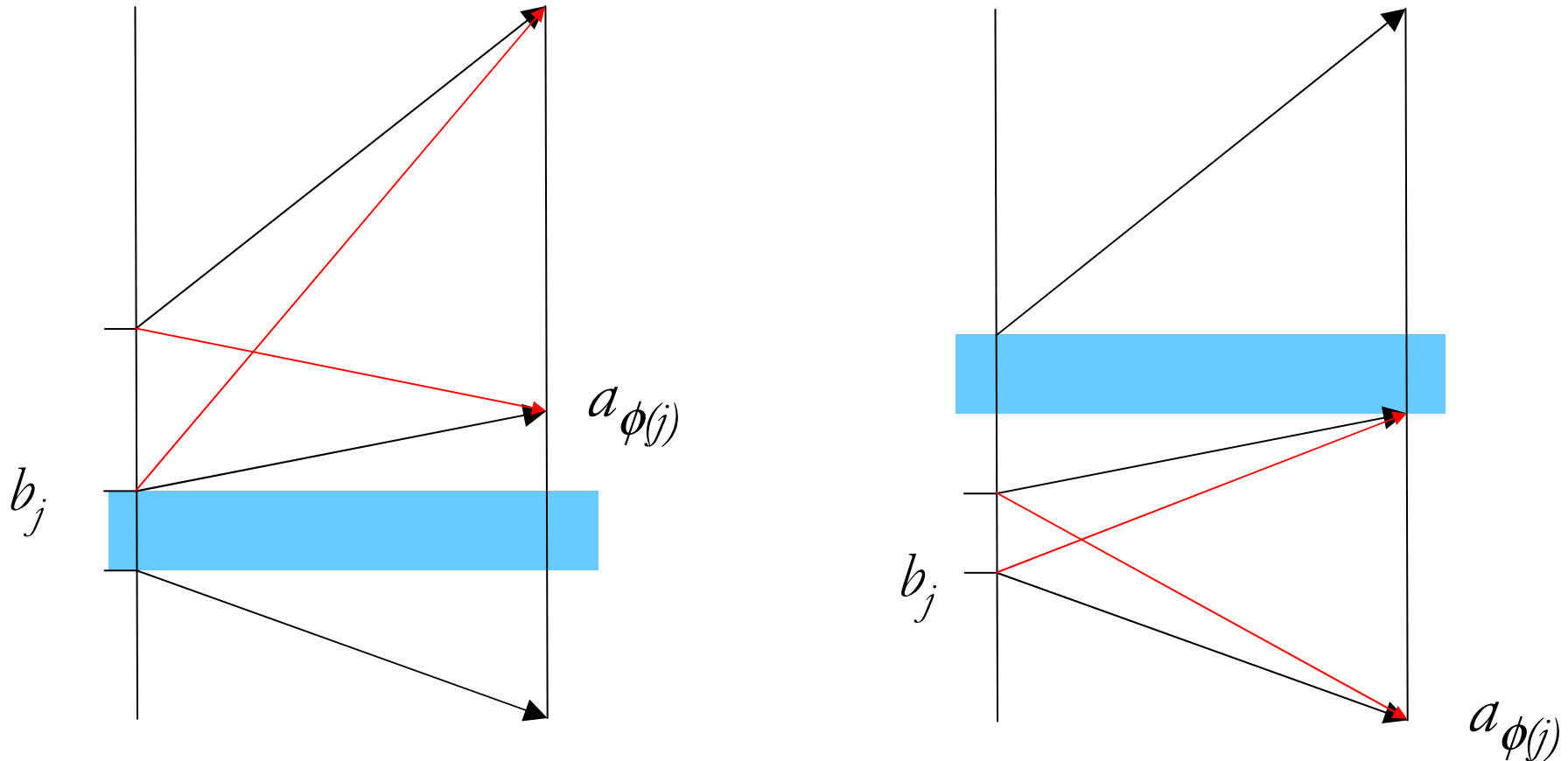
# Esempio

Al contrario, se eseguiamo prima lo swap  $I(1, 2)$  e poi  $I(2, 3)$ , il costo di quest'ultimo cambia:  $c_{\Phi'} I(2, 3) = 4$ .



## Scelta della sequenza di swap

Informalmente, quando si esegue (a partire dalla permutazione ottima) uno swap la cui freccia inferiore “punta in su” (punta in giù) il costo degli swap inferiori (superiori) non cambia.



# Classificazione degli swap

**Definizione.** Un nodo  $j$  è detto di

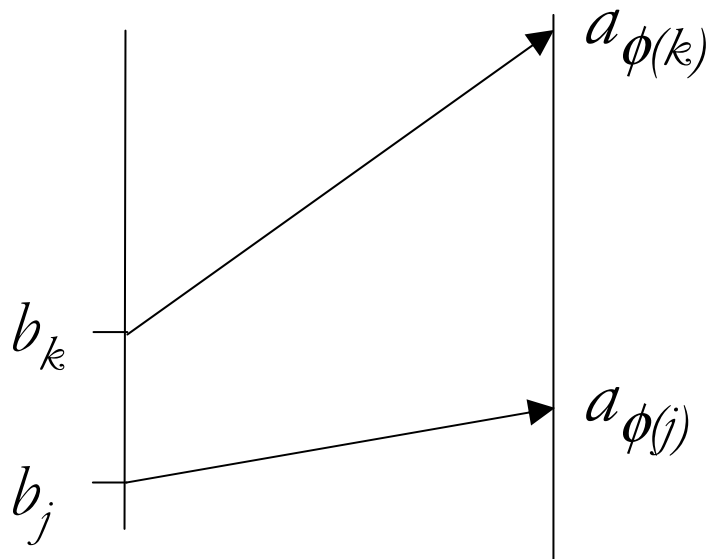
Tipo 1 se  $b_j \leq a_{\phi(j)}$

Tipo 2 se  $b_j > a_{\phi(j)}$

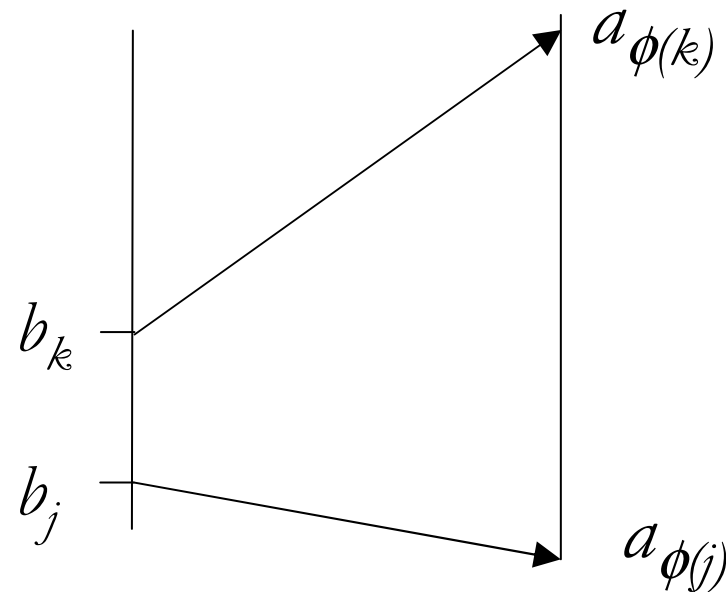
Uno swap è detto di

Tipo 1 se il suo nodo inferiore è di tipo 1

Tipo 2 se il suo nodo inferiore è di tipo 2



**Tipo 1**



**Tipo 2**

## Scelta della sequenza di swap

È possibile ottenere un tour ammissibile senza modificare i costi degli archi inter-ciclo  $c_{\Phi^* I(j, k)}$  mediante la seguente:

**Regola:** eseguire gli swap di tipo 1 in ordine decrescente di indici seguiti dagli swap di tipo 2 in ordine crescente di indici

# Esempio

job	0	1	2	3	4	5	6
$b_j$	1	15	26	40	3	19	31
$a_j$	7	16	22	18	4	45	34

*stato iniziale*

$b_0$

*stato finale*

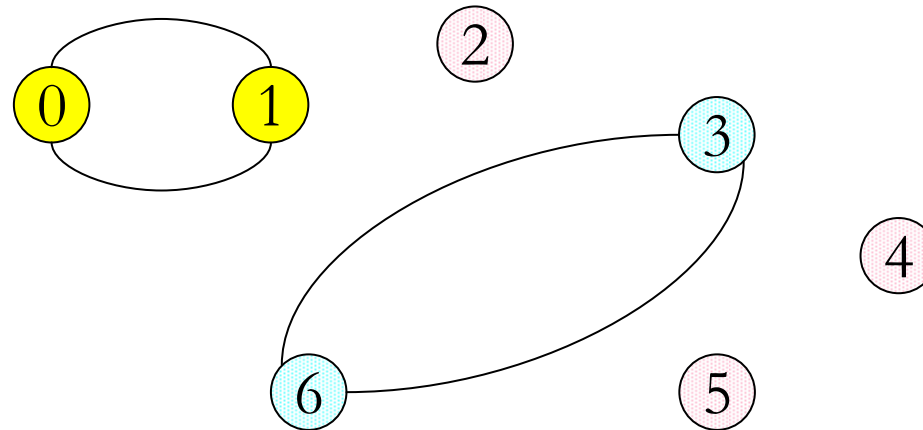
$a_0$

Step 1:

job	0	1	2	3	4	5	6
$b_j$	1	3	15	19	26	31	40
$a_j$	7	4	16	45	22	34	18
$a_{\phi^*(j)}$	4	7	16	18	22	34	45
$\phi^*(j)$	1	0	2	6	4	5	3

# Esempio

Step 2:

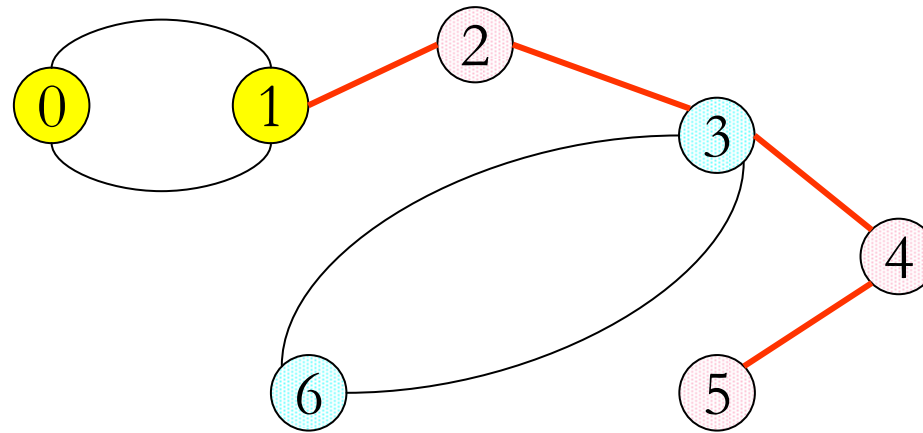


Step 3:

arco	(1,2)	(2,3)	(3,4)	(4,5)	(5,6)
$b_j$	3	15	19	26	31
$b_{j+1}$	15	19	26	31	40
$a_{\Phi^*(j)}$	7	16	18	22	34
$a_{\Phi^*(j+1)}$	16	18	22	34	45
$c_{\Phi^*}I(j, j+1)$	16	4	6	10	12

# Esempio

Step 4:



Step 5:

arco	(1,2)	(2,3)	(3,4)	(4,5)
$b_j$	3	15	19	26
$a_{\phi^*(j)}$	7	16	18	22
<i>tipo</i>	1	1	2	2

# Esempio

Step 6:

arco	(1,2)	(2,3)	(3,4)	(4,5)
$b_j$	3	15	19	26
$a_{\Phi^*(j)}$	7	16	18	22
tipo	1	1	2	2

$$j_1 = 2, j_2 = 1$$

$$k_1 = 3, k_2 = 4$$

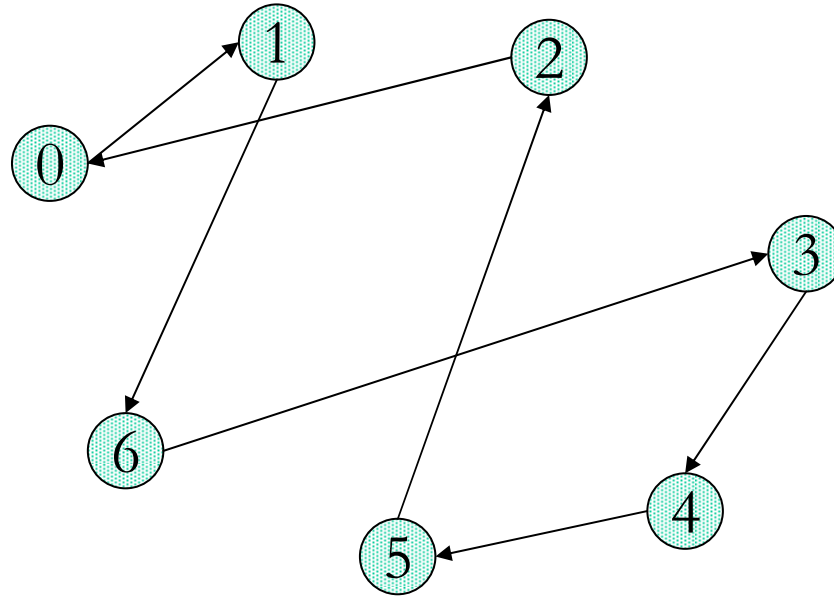
Step 7:

$$\Phi^{**} = \Phi^* I(2, 3) I(1, 2) I(3, 4) I(4, 5)$$

città	0	1	2	3	4	5	6
$\phi^*(j)$	1	0	2	6	4	5	3
$\phi^*(j)I(2,3)$	1	0	6	2	4	5	3
$\phi^*(j)I(2,3)I(1,2)$	1	6	0	2	4	5	3
$\phi^*(j)I(2,3)I(1,2)I(3,4)$	1	6	0	4	2	5	3
$\phi^*(j)I(2,3)I(1,2)I(3,4)I(4,5)$	1	6	0	4	5	2	3



# Tour ottimo



città	0	1	2	3	4	5	6
$\phi^*(j)$	1	6	0	4	5	2	3
$I(2, 3)$							
$I(1, 2)$							
$I(3, 4)$							
$I(4, 5)$							