Ottimizzazione Combinatoria

Euristica di Ricerca Locale

ANTONIO SASSANO

Università di Roma"La Sapienza"
Dipartimento di Informatica e Sistemistica

Roma, Dicembre 2009

Massimo Sottoinsieme Indipendente

$\max \{ c(F)\text{: } F$ **è un *indipendente* di** $\mathcal{S} \}$

Problema di ***Ottimizzazione Combinatoria*** con *funzione obiettivo lineare*

F^* *ottimo* $\Rightarrow c(F^*) \geq c(F) \quad F \in \mathcal{S}$

$h \in F^* \;\; \Rightarrow \;\; C_h \geq 0 \quad [$ se $C_h < 0 \Rightarrow c(F^* - \{h\}) > c(F^*)$ CONT. $]$

$\Rightarrow$ *possiamo* ***eliminare*** *gli elementi con* ***peso negativo***

(MIS)

$$\max c^T x$$
$$x \in P_R \cap \{0,1\}^{|\Gamma|}$$

(Rilassamento Lineare)

$$\max c^T x = UB$$
$$x \in P_R$$

$$c(F°) \leq c(F^*) \leq UB$$

Vogliamo trovare (euristica) una buona soluzione $F°$

Intorno di una soluzione ammissibile

- ***Insieme base*** $\Gamma = \{1,2,\ldots,n\}$ ***(eventi elementari)***
- ***Soluzioni ammissibili*** $\mathcal{S} = \{ F_1, F_2, \ldots, F_m \}$ $(F_i \subseteq \Gamma)$
- ***Intorno*** $\mathcal{N}(F)$ *di una soluzione ammissibile* F $\boxed{\mathcal{N}(F) \subseteq \mathcal{S}}$
- $\mathcal{N}(F) = \{ F_{j1}, F_{j2}, \ldots, F_{jm} \}$ ***vicini di (adiacenti a)*** F
- $\{ \mathcal{N}(F) : F \in \mathcal{S} \}$ ***Sistema di Intorni per*** $\mathcal{S}$

La scelta del ***sistema di intorni*** *dipende dal problema*

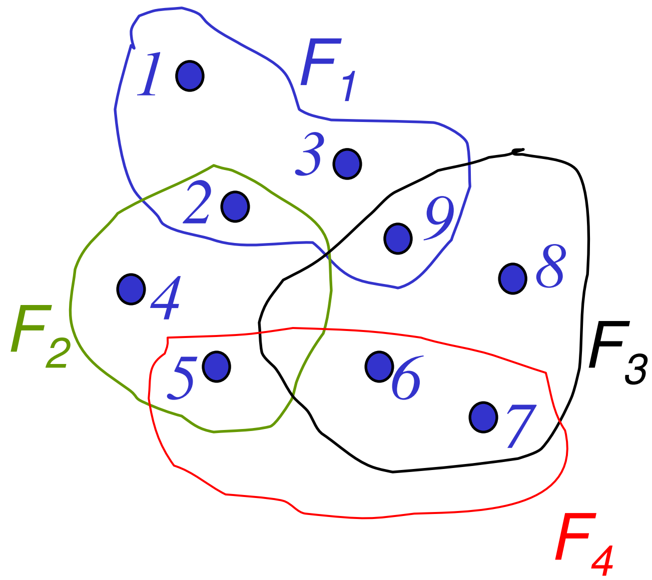
Esempio:

$$\mathcal{N}(F_1) = \{ F_2, F_4 \}$$

$$\mathcal{N}(F_2) = \{ F_1, F_4 \}$$

$$\mathcal{N}(F_3) = \{ F_2 \}$$

$$\mathcal{N}(F_4) = \{ F_1, F_2, F_3 \}$$



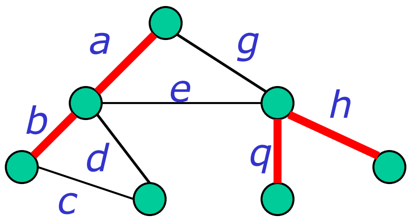
Sistemi di intorni elementari: greedy

Intorno "greedy"

$$F \in \mathcal{N}_+(F_i) \Leftrightarrow F = F_i \cup \{ j \} : j \notin F_i,\ F \in \mathcal{S}$$

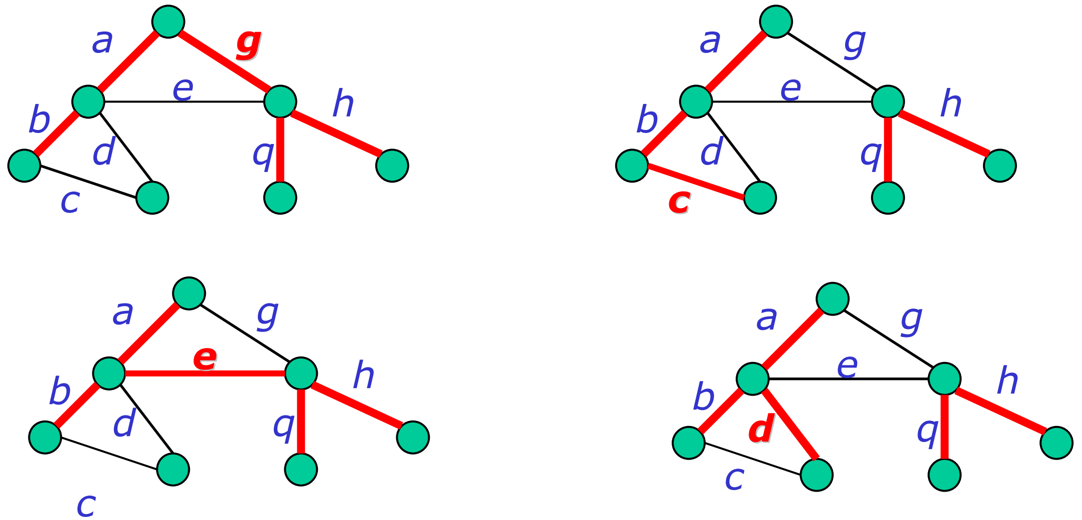
Esempio:

$\mathcal{S} = \{$ **foreste di un grafo** $G(V,E)$ $\}$



T_0

$\mathcal{N}_+(T_0)$



Foreste ***"adiacenti a"***

(o ***"nell'intorno greedy di"***) T_0

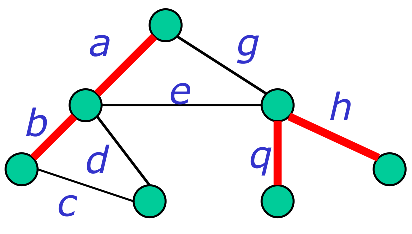
Sistemi di intorni elementari: reverse greedy

Intorno "reverse greedy"

$$F \in \mathcal{N}_{-}(F_i) \Leftrightarrow F = F_i - \{ k \} : k \in F_i$$

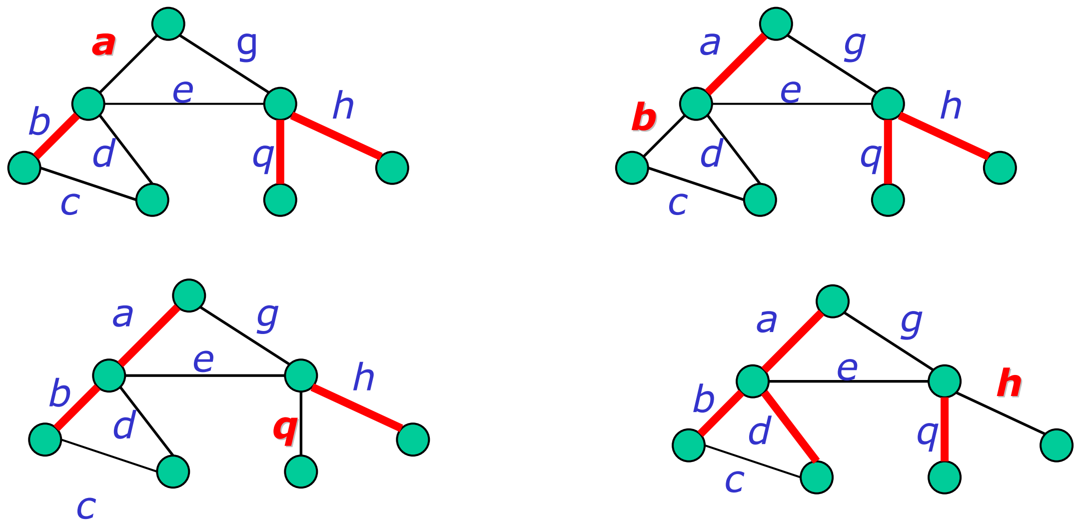
Esempio:

$\mathcal{S}$ = { **foreste di un grafo** $G(V,E)$ }



T_0

$\mathcal{N}_{-}(T_0)$



Foreste ***"adiacenti a"***

(o ***"nell'intorno reverse greedy di"***) T_0

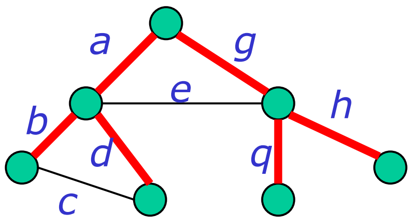
Sistemi di intorni elementari: scambio

Intorno di "scambio"

$$F \in \mathcal{N}_s(F_i) \Leftrightarrow F = F_i \cup \{ j \} - \{ k \} : k \in F_i,\ j \notin F_i,\ F \in \mathcal{S}$$

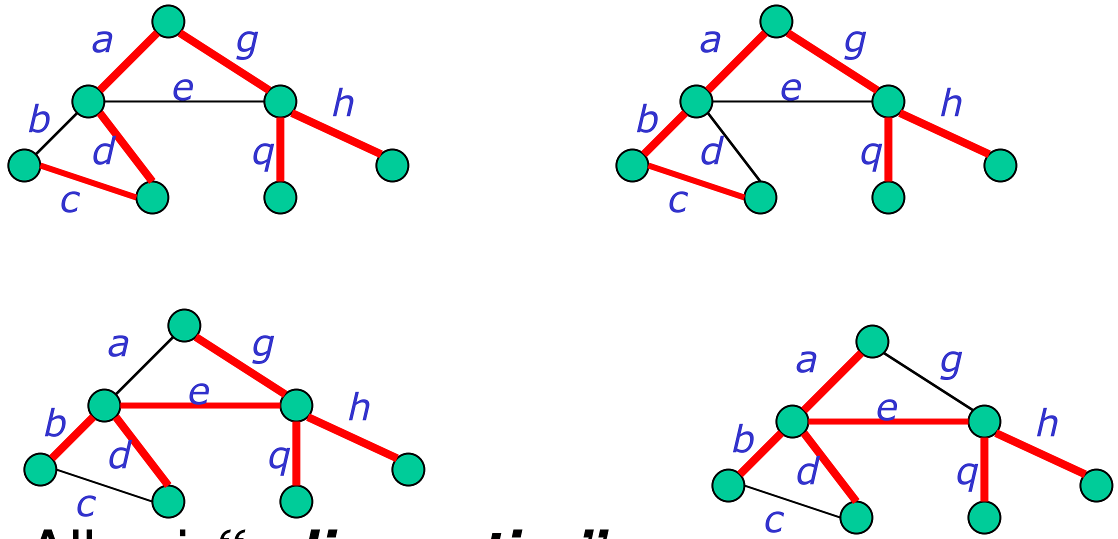
Esempio:

$\mathcal{S} = \{$ **foreste di un grafo** $G(V,E)$ $\}$



T_0

$\mathcal{N}_s(T_0)$



Alberi ***"adiacenti a"***
(o ***"nell'intorno di scambio di"***) T_0

Sistema di intorni – "scambio generalizzato"

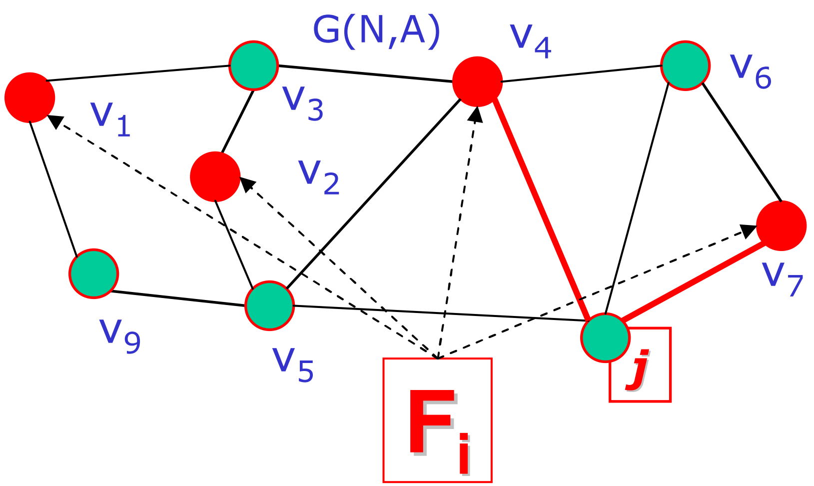
Dato $X \subseteq \Gamma$ definiamo: $\mathcal{C}(X)=\{C \subseteq X$ **circuito di** $\mathcal{S}\}$

$$\boxed{X \in \mathcal{S} \Rightarrow \mathcal{C}(X)=\emptyset}$$

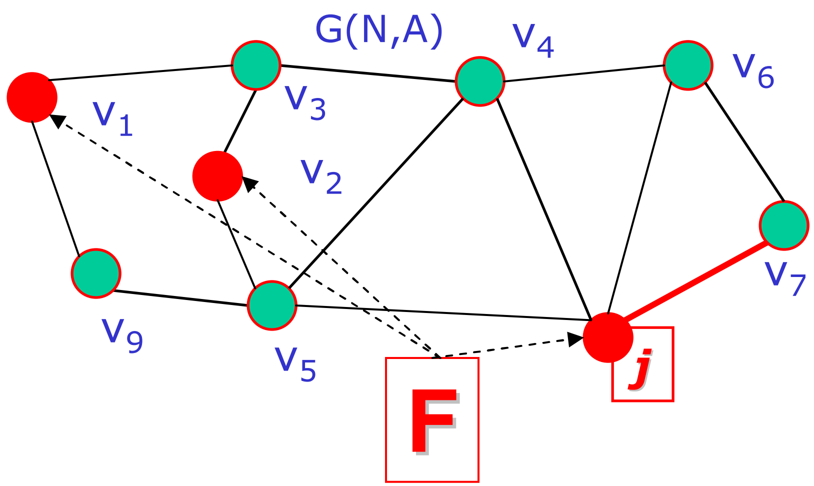
$$\boxed{F \in \mathcal{N}_G(F_i) \Leftrightarrow F = F_i \cup \{ j \} - K: \ j \notin F_i}$$

dove K ha la proprietà che: $\boxed{K \cap C \neq \emptyset \quad \forall C \in \mathcal{C}(F_i \cup \{ j \})}$

Esempio: $\mathcal{S} = \{$ ***insiemi stabili un grafo*** *G(V,E)* $\}$



$\mathcal{C}(F_i \cup \{ j \})= \{ jv_4, jv_7\}$



$K= \{ v_4, v_7\}$

$F= F_i \cup \{ j \} - \{ v_4, v_7\}$

Sistemi di Intorni

"greedy" e *"scambio"* **sono** *casi particolari* **dello** *"scambio generalizzato"*

$$F \in \mathcal{N}_G(F_i) \Leftrightarrow F = F_i \cup \{ j \} - K:\ j \notin F_i$$

$$K \cap C \neq \varnothing \quad \forall C \in \mathcal{C}(F_i \cup \{ j \})$$

$$F \in \mathcal{N}_+(F_i) \Leftrightarrow F = F_i \cup j : j \notin F_i,\ F \in \mathcal{S}$$

Intorno *"greedy"* $(K = \varnothing)$

DIM: $\mathcal{C}(F_i \cup \{ j \}) = \mathcal{C}(F)$ ..ma $F \in \mathcal{S} \Rightarrow \mathcal{C}(F) = \varnothing \Rightarrow K = \varnothing$

$$F \in \mathcal{N}_S(F_i) \Leftrightarrow F = F_i \cup \{ j \} - \{k\} : k \in F_i,\ j \notin F_i,\ F \in \mathcal{S}$$

Intorno di *"scambio"* $(K = \{k\})$

DIM: $F = F_i \cup \{ j \} - \{k\} \in \mathcal{S} \Rightarrow$

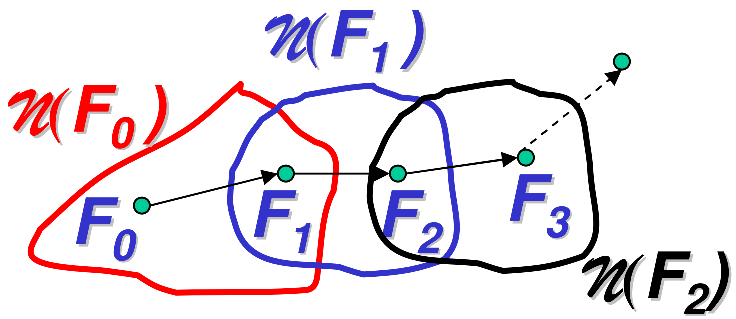
$\Rightarrow$ **ogni circuito di** $\mathcal{C}(F_i \cup \{ j \})$ *contiene* k

$\Rightarrow K = \{k\}$

Algoritmo di Ricerca Locale ("Local Search")

Euristica per il problema: $\max \{ c(F): F \in \mathcal{S} \}$

Idea base



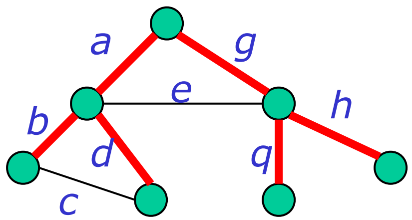
Costruire una ***sequenza*** di soluzioni ammissibili $F_0, F_1, F_2, F_3 \ldots$:

a. **a partire** da una soluzione ammissibile F_0

b. **individua**, al passo k, la **soluzione di massimo peso** F_k appartenente all'intorno $\mathcal{N}(F_{k-1})$ della soluzione corrente F_{k-1}

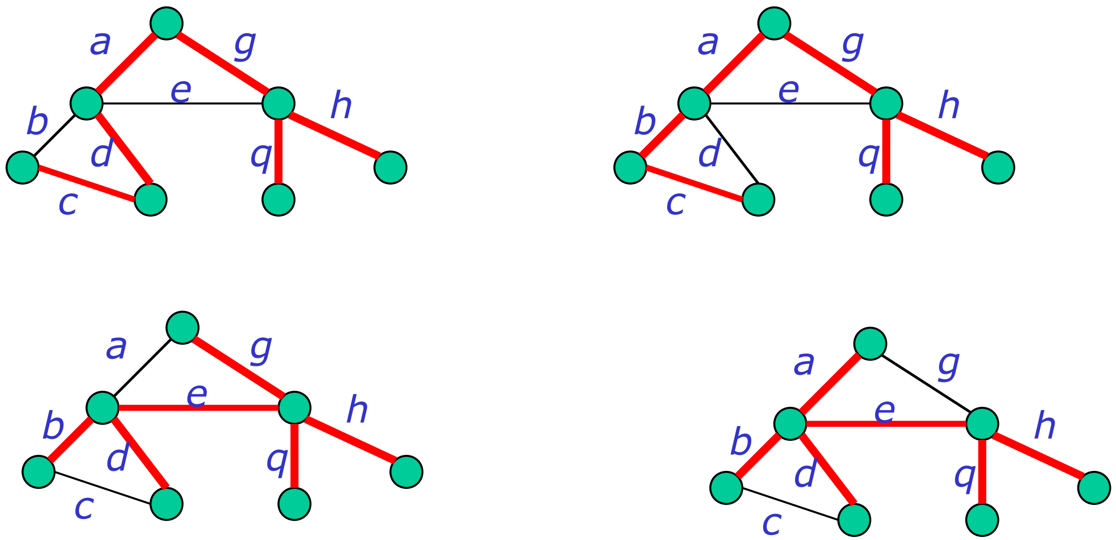
c. **STOP** se ogni soluzione appartenente all'intorno $\mathcal{N}(F_{k-1})$ ha un valore della funzione obiettivo **minore di** $c(F_{k-1})$

Albero Ricoprente: Intorno di Scambio

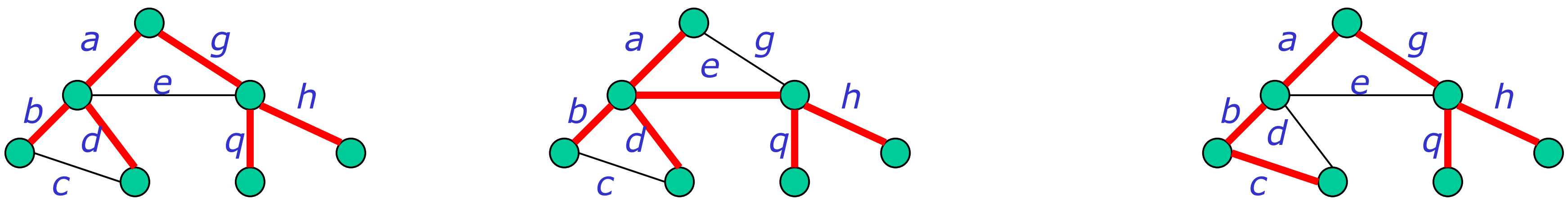


F_0

$\mathcal{N}_s(F_0)$



Alberi ***“adiacenti a”***
(o ***“nell’intorno di scambio di”***) F_0

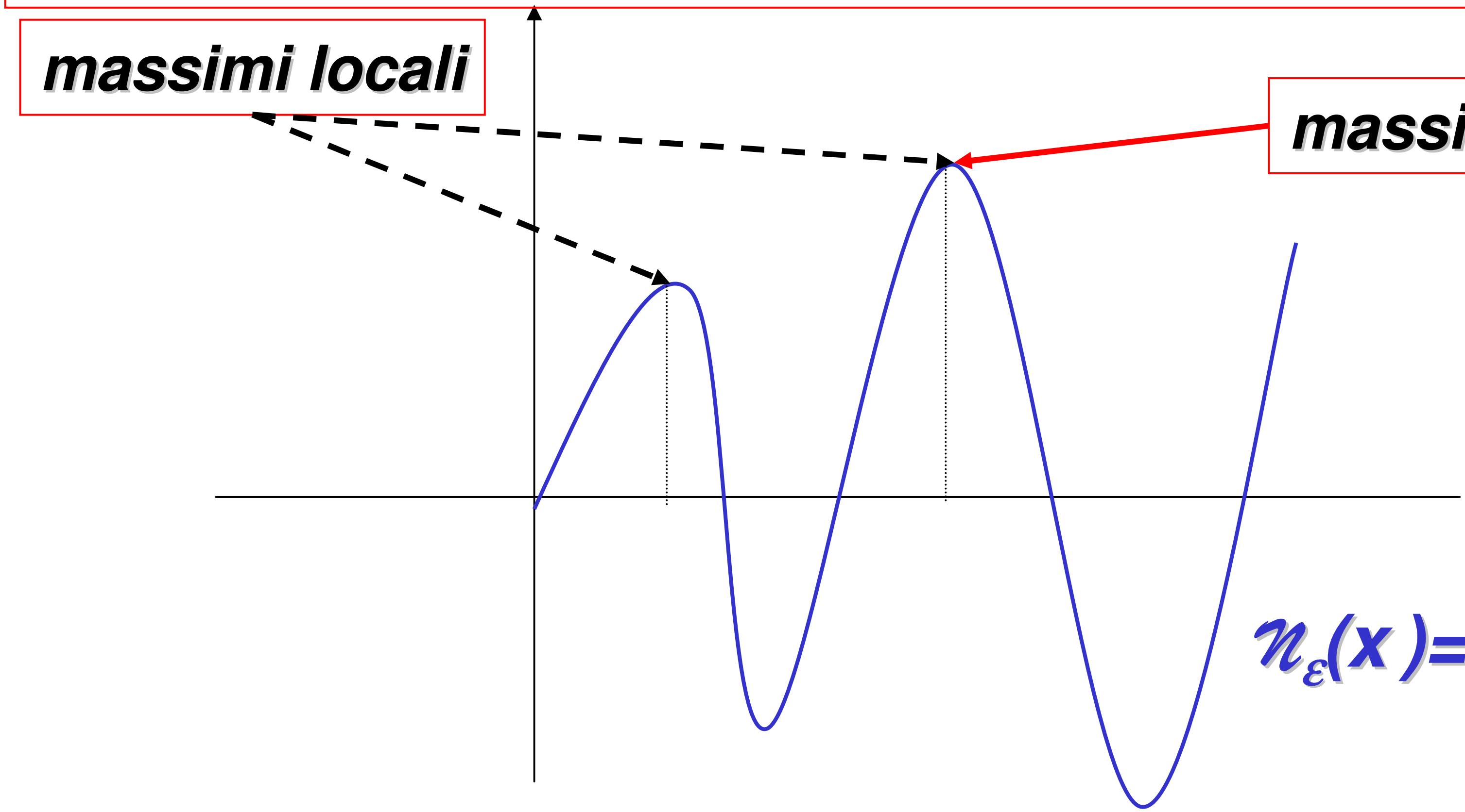


$F_0 \longrightarrow F_1 = F_0 \cup \{e\} - \{g\} \longrightarrow F_2 = F_1 \cup \{c\} - \{d\}$

Massimi Locali e Massimi Globali

F^* **massimo globale** $\Leftrightarrow$ $c(F^*) \geq c(F)$ per ogni $F \in \mathcal{S}$

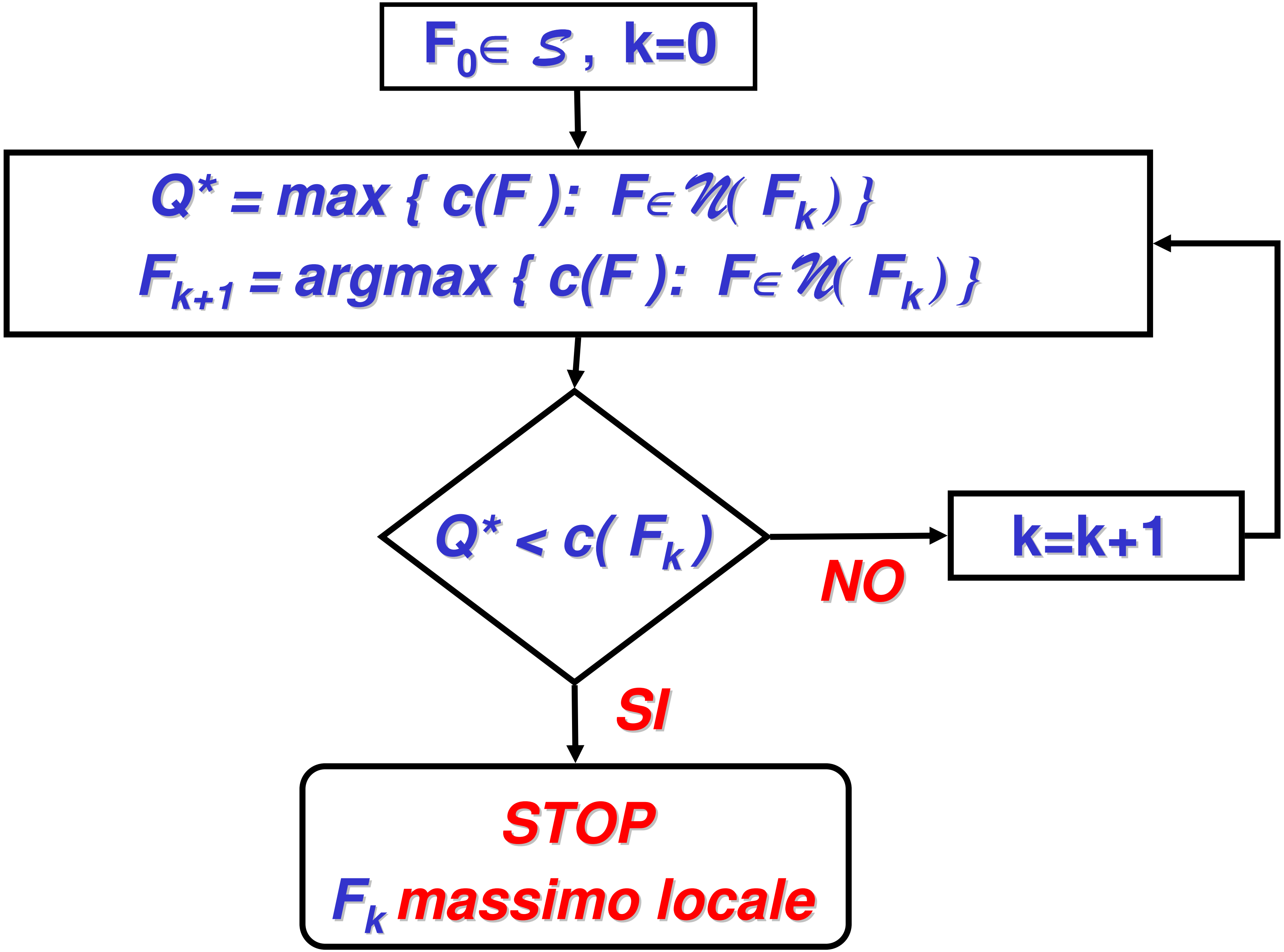
F^* **massimo locale** $\Leftrightarrow$ $c(F^*) \geq c(F)$ per ogni $F \in \mathcal{N}(F^*)$



$\mathcal{N}_\varepsilon(x) = \{y: |x-y| < \varepsilon\}$

L'algoritmo di **ricerca locale** individua un **massimo locale**

Ricerca Locale ("Local Search")

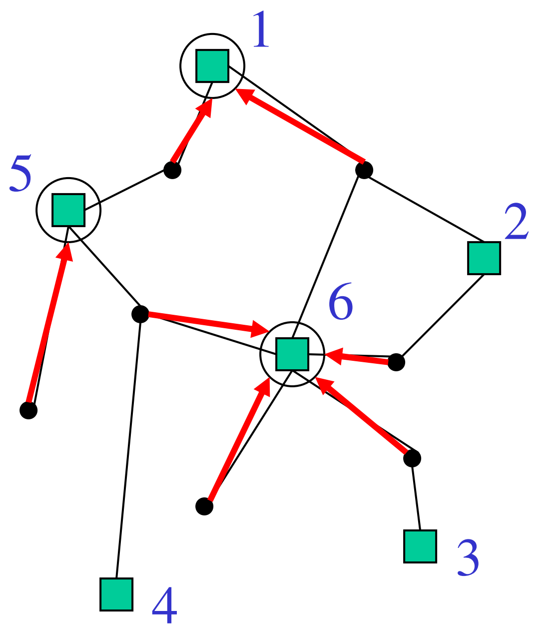


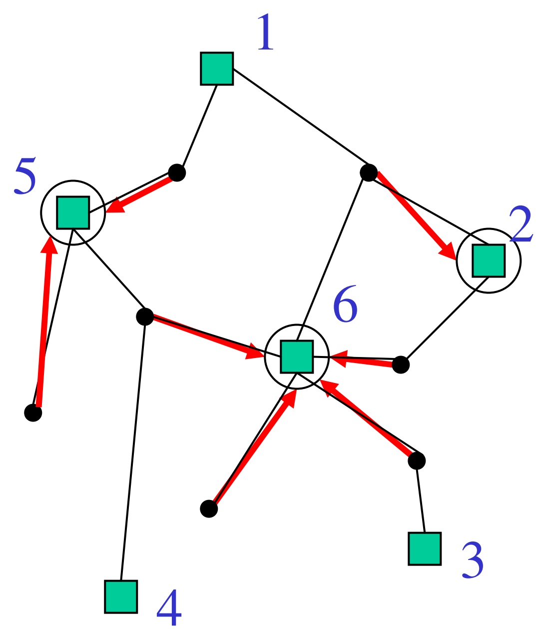
Esempio "Local Search" - Localizzazione

Attenzione! Problema di ***minimo***

Intorno di "scambio"

$$\mathcal{N}_s\,(F_i) = \{\, F_i - \{k\} \cup \{j\} \in \mathcal{S} :\ k \in F_i,\ j \notin F_i \}$$





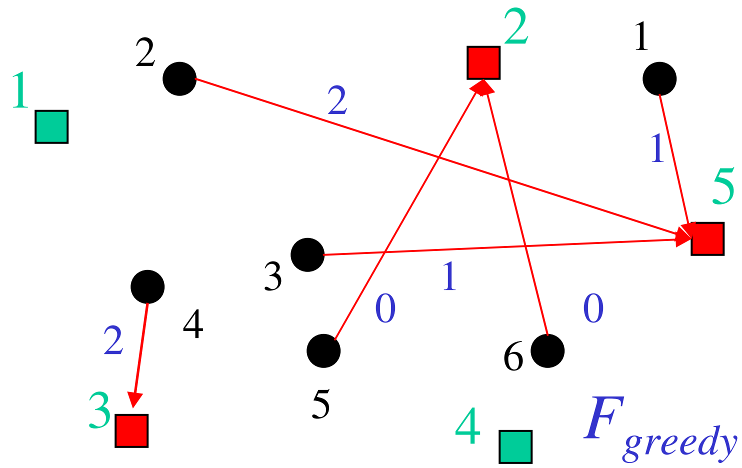
$F_i=\{1,5,6\}$

$F_k=\{1,5,6\} - \{1\} \cup \{2\} \in \mathcal{N}_s(F_i)$

$\mathcal{N}_s(F_i) = \{\{2,5,6\}, \{3,5,6\}, \{4,5,6\}, \{1,4,6\}, \ldots$

$$
\begin{array}{ccccc}
1 & 2 & 3 & 4 & 5
\end{array}
$$

$$
c = \begin{bmatrix}
12 & 13 & 6 & 0 & 1 \\
8 & 4 & 9 & 1 & 2 \\
2 & 6 & 6 & 0 & 1 \\
3 & 5 & 2 & 10 & 8 \\
8 & 0 & 5 & 10 & 8 \\
2 & 0 & 3 & 4 & 1
\end{bmatrix}
$$



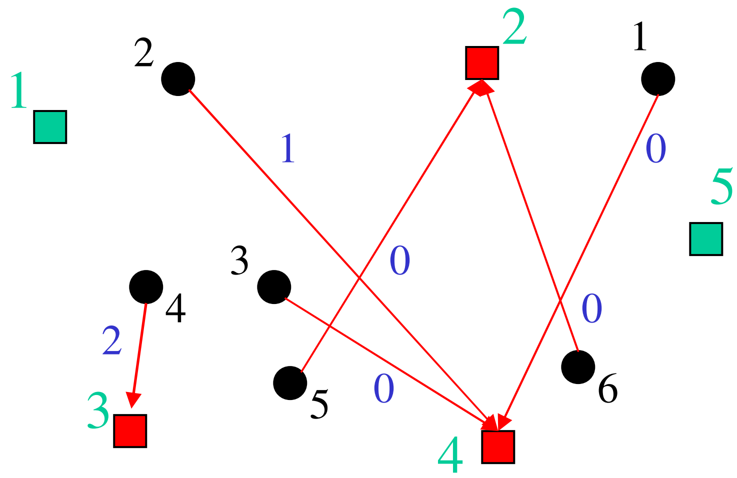
$$F_{greedy} = \{2,3,5\}$$

$$f = [4 \quad \boxed{3} \quad \boxed{1} \quad 4 \quad \boxed{7}]$$

$$\mathcal{N}_s(F_{greedy}) = \{\{2,3,1\},\ \boxed{\{2,3,4\}},\ \{1,3,5\},\ \{4,3,5\},\ \{2,1,5\},\ \{2,4,5\}\}$$

$$
\begin{array}{ccccc}
1 & 2 & 3 & 4 & 5
\end{array}
$$

$$
c = \begin{bmatrix}
12 & 13 & 6 & 0 & 1 \\
8 & 4 & 9 & 1 & 2 \\
2 & 6 & 6 & 0 & 1 \\
3 & 5 & 2 & 10 & 8 \\
8 & 0 & 5 & 10 & 8 \\
2 & 0 & 3 & 4 & 1
\end{bmatrix}
$$



$$f = [4 \quad \boxed{3} \quad \boxed{1} \quad \boxed{4} \quad 7]$$