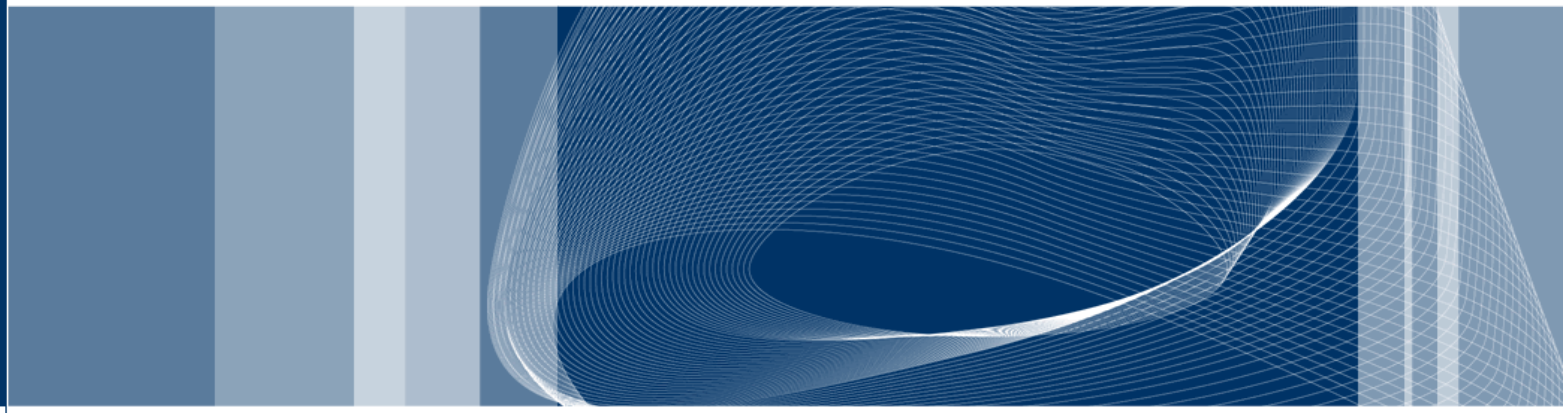
POLITECNICO DI MILANO



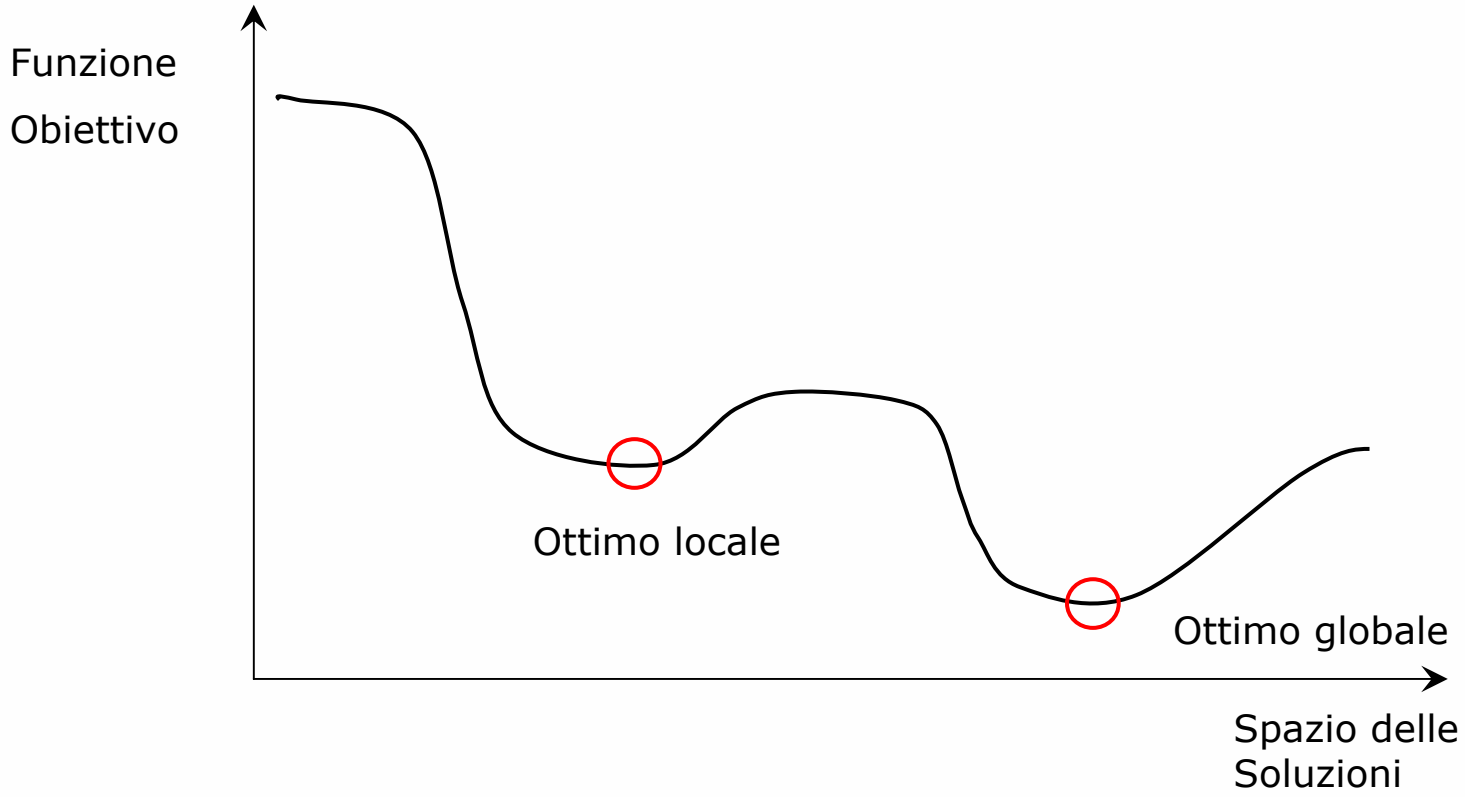
Algoritmi e Strutture Dati

Tabu Search

Daniele Loiacono

Tecniche Meta-euristiche

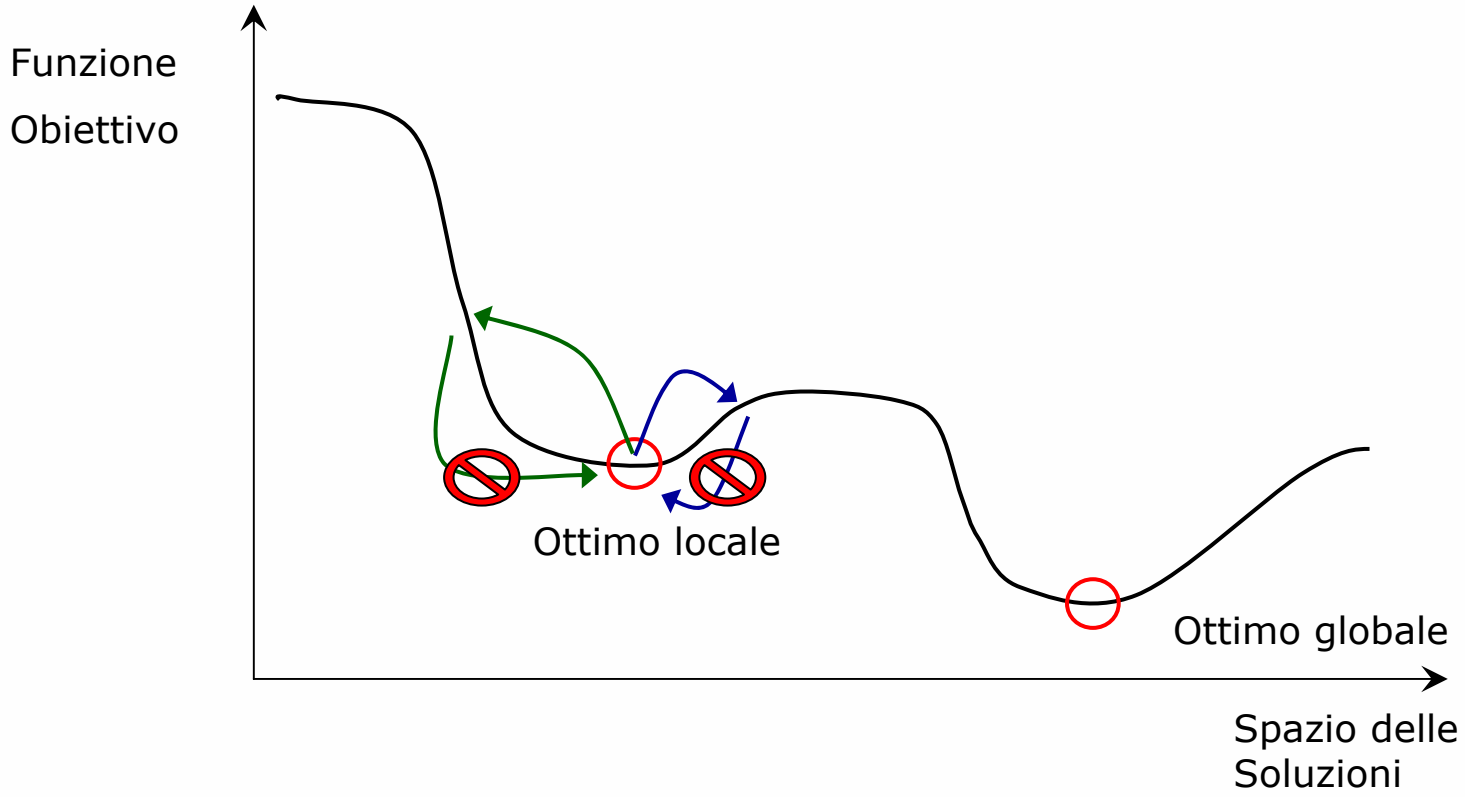
- Sono algoritmi di ottimizzazione basati su una ricerca locale
  - Partono da una soluzione e ne esplorano l'intorno
  - Si focalizzano sulla soluzione più promettente nell'intorno
- Hanno meccanismi pensati appositamente per evitare di rimanere bloccati in ottimi locali



Tabu Search

- È una generalizzazione della ricerca locale che consente di esplorare soluzioni "peggiorative"
- Utilizza una memoria a breve termine (**Tabu List**) per evitare di ritornare nelle ultime *t* soluzioni visitate:

$$T = \{x_{k-1}, x_{k-2}, \ldots, x_{k-t}\}$$



Quando applicare il Tabu Search

- Il Tabu Search può essere applicato ad un problema di ottimizzazione con i seguenti elementi
  - Lo spazio del problema ammette il concetto di intorno: data una soluzione x, esiste un insieme di soluzioni N(x) che rappresentano tutte le soluzioni raggiungibili da x con una sola *mossa*
  - disponibile una funzione di valutazione (o funzione obiettivo) che permette di calcolare facilmente il valore di una soluzione

Algoritmo base

- Pseudocodice dell'algoritmo base per un problema di minimizzazione
  - $z(\cdot)$ è la funzione obiettivo
  - $N(s)$ è l'intorno di s

- genera una soluzione iniziale s di valore $z(s)$
- $s^* = s$; $k := 1$; $T = \{s\}$;
- while not CRITERIO_TERMINAZIONE do
- genera l'intorno $N(s)\backslash T$ /* non tabu */
- trova la migliore soluzione $s' \in N(s)\backslash T$ rispetto a $z(\cdot)$
- if $z(s') < z(s^*)$ then $s^* := s'$; $k$best $:= k$
- $s := s'$
- $k := k+1$
- inserisci s' in T al posto della soluzione più vecchia
- endwhile

Tabu Search: criteri di terminazione

- Criteri di arresto possibili (`CRITERIO_TERMINAZIONE`):
  - $N(s) \setminus T = \emptyset$
  - $k > k_{max}$
  - Limite di tempo raggiunto
  - $k - k_{best} > k^*$ (numero massimo di iterazioni senza miglioramento)
  - s^* ottima (ad esempio pari ad un lower bound)

Lista Tabu

- T impedisce il verificarsi di cicli di lunghezza $\leq |T|$ ma…
- … memorizzare in T soluzioni complete può essere oneroso
- Esempio: problema di ottimizzazione in n variabili
  - ogni soluzione è un vettore di n elementi ;
  - confrontare due soluzioni costa $O(n)$;
  - verificare se una soluzione è tabu costa $O(n \cdot |T|)$
- Soluzioni
  - Memorizzare la tabu list T in una **hashtable**
  - Rappresentazione di T basata sulle mosse

Hashtable

- Senza perdere in generalità consideriamo un problema di ottimizzazione in cui le soluzioni sono vettori di **n** numeri interi definiti in **[a,b]**
- Memorizziamo la tabu list in una hashtable per agevolare la costruzione dell'intorno ammissibile
- Quale funzione di hash usare?

$$h(x) = \sum_{i=1}^{n} z_i x_i$$

  - Dove **z** è un vettore di numeri pseudocasuali definiti in [1,m]
- Caratteristiche
  - La probabilità che due vettori abbiano la stessa hash decresce con l'aumentare di n, m e b-a
  - È facile da calcolare per elementi nell'intorno: se x e y sono identici a meno dell'i-esima componente

$$h(y) = h(x) + z_i(y_i - x_i)$$

Mosse e Lista Tabu

- In diversi problemi di ottimizzazione, l'intorno di una soluzione e la la lista tabu possono essere definiti efficacemente con il concetto di mossa
- Una **mossa *m*** è un operazione elementare (e.g. scambio di due elementi di una soluzione) per ottenere una soluzione *s'* dalla soluzione corrente *s:*

$$s' = s \oplus m$$

- Possiamo allora definire l'intorno di s come:

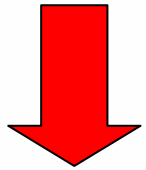
$$N(s) := \{s' : \exists\ m \text{ tale che } s' = s \oplus m\ \}$$

- La tabu list può essere definita come l'insieme di **mosse inverse** alle ultime *t* mosse effettuate nell'esplorazione dello spazio delle soluzioni

Mosse e Lista Tabu: esempio

- Siano le soluzioni di un problema tutte le possibili terne di elementi distinti dell'insieme {a,b,c,d,e}

Soluzione	Mossa	Tabu List
abc	c → d	d → c
abd	b → c	d → c c → b
acd	d → b	d → c c → b b → d



acb=abc

Limitazione delle mosse inverse

- La tabu list T^I che rappresenta le mosse inverse è molto più restrittiva di T che rappresenta le soluzioni precedenti

$$R := \{x' : \exists\, m \in T^I \text{ tale che } x' = x \oplus m\}$$

- Risulta che
- $$| N(x) \backslash R | \leq | N(x) \backslash T | \text{ (spesso } << \text{)}$$

- Tuttavia T^I non garantisce che non si abbiano cicli di periodo $\leq | T^I |$

Criteri di Aspirazione

- Tecnica utilizzata per sopperire alle condizioni troppo restrittive imposte dalle mosse inverse
- Una mossa, anche se tabu, può essere comunque effettuata se questa conduce ad una “buona soluzione”
- La bontà della soluzione si valuta con un opportuno contributo alla funzione obiettivo

Intensificazione e diversificazione

- La tabu list costituisce la memoria di breve periodo
- Una ricerca efficace necessita anche di memoria di medio/lungo periodo
- Il Tabu Search utilizza a questo scopo i seguenti meccanismi
  - **intensificazione della ricerca**
    - non è consentito spostarsi troppo dalla parte di regione che si sta visitando
    - preferire mosse con caratteristiche in comune con una buona soluzione recentemente incontrata
    - penalità da aggiungere a z(·) per le mosse che alterano tale caratteristica
  - **diversificazione della ricerca**
    - per spostarsi verso altre zone (inesplorate) dello spazio delle soluzioni
    - penalità da aggiungere a z(·) per le soluzioni troppo vicine a quella corrente

Tabu search: vincoli

- Se il problema è fortemente vincolato la cardinalità di $N(s)$ può essere molto piccola (è facile che $N(x) \setminus T = \emptyset$)
- Si rilassano alcuni vincoli aggiungendo ad $f(x)$ una penalità proporzionale alla violazione dei vincoli in x
- La ricerca si muove anche attraverso soluzioni non ammissibili
- Aggiustamento adattativo della penalità:
  - La penalità può crescere se da molte iterazioni non si incontrano soluzioni ammissibili
  - La penalità può decrescere se da molte iterazioni si incontrano soluzioni tutte ammissibili

Tabu Tenure

- Finora abbiamo ipotizzato la **tabu tenure** (lunghezza della lista tabu) una costante pari a t
- Nella pratica può essere modificata dinamicamente (es. ogni h iterazioni)
  - **Intensificazione:**
    se s^* è stata migliorata → $t := \min \{t\text{min}, t\text{-}1\}$
  - **Diversificazione:**
    se s^* rimane immutata → $t := \max \{t\text{max}, t\text{+}1\}$
- In alternativa può essere scelta casualmente in $[t\text{min}, t\text{max}]$
- È molto importante scegliere I valori di t, $t$min, $t$max, h