Algoritmi euristici: definizioni

Sapienza Università di Roma

Indice

Perché gli algoritmi euristici?

Algoritmi euristici

I problemi e le ipotesi

Bound primali e duali

Classificazione degli algoritmi euristici

Rappresentazione delle soluzioni e spazio di ricerca

Intorni e ottimi locali

Intensificazione e diversificazione

Come valutare un algoritmo euristico

Errori tipici

In sintesi

Obiettivi

Alla fine di questo capitolo saremo in grado di:

- definire un algoritmo euristico e calcolarne il gap rispetto a un bound duale;
- distinguere i bound primali dai bound duali sulle quattro istanze di riferimento;
- classificare gli algoritmi euristici e descrivere la rappresentazione delle soluzioni e lo spazio di ricerca;
- definire intorni e ottimi locali, e spiegare intensificazione e diversificazione;
- valutare un algoritmo euristico in base a qualità, tempo di esecuzione ed esecuzioni ripetute.

Prerequisiti

- Dal modulo *Problemi di ottimizzazione combinatoria*, il problema dello zaino 0/1, il problema del bin packing, il problema del commesso viaggiatore e $P \,||\, C_{\max}$: istanze, soluzioni ammissibili e valori ottimi dei quattro problemi usati come esempi.

- Dal modulo *Programmazione lineare intera*, Capitolo 2 «Bound primali e duali»: il rilassamento lineare come fonte di bound duali.

- Qui ne diamo un breve richiamo per il problema dello zaino 0/1.

Perché gli algoritmi euristici?

Introduzione: Perché gli algoritmi euristici?

- Partiamo da una domanda pratica: che cosa possiamo fare quando un problema di ottimizzazione non si può risolvere all'ottimo nel tempo che abbiamo?
- Molti problemi di ottimizzazione di rilevanza pratica sono difficili, e gli algoritmi esatti possono essere troppo lenti
- Gli algoritmi euristici sono la risposta: danno buone soluzioni in poco tempo, senza dimostrare che sono ottime

Problemi di ottimizzazione e problemi difficili

- Sia $\mathscr{P}$ un **problema di ottimizzazione**.
- Per ogni **istanza** I di $\mathscr{P}$, sia $z(I)$ il **valore ottimo** di $\mathscr{P}$ su I.
- Molti problemi di ottimizzazione di rilevanza pratica sono **NP-hard**: non si conosce alcun algoritmo che risolva all'ottimo tutte le loro istanze in tempo polinomiale, ed è opinione diffusa che non ne esista nessuno.

Quando il tempo non basta

- I quattro problemi usati come esempi in questo capitolo sono tutti NP-hard: il problema dello zaino, il problema del bin packing, il problema del commesso viaggiatore e il problema di scheduling $P \,||\, C_{\max}$.
- Per questi problemi un algoritmo esatto può richiedere un tempo di esecuzione che cresce esponenzialmente con la dimensione dell'istanza.
- Quando questo tempo non è disponibile, accettiamo una soluzione *buona*, anche se non è dimostrato che sia ottima.

Riepilogo

La difficoltà:

- molti problemi di ottimizzazione di rilevanza pratica sono NP-hard
- un algoritmo esatto può richiedere un tempo che cresce esponenzialmente con la dimensione dell'istanza

La via d'uscita:

- quando il tempo non basta, accettiamo una soluzione buona, non dimostrata ottima
- è il compito degli algoritmi euristici

Algoritmi euristici

Introduzione: Algoritmi euristici

- In questa sezione definiamo che cos'è un algoritmo euristico e il valore $\mathscr{A}(I)$ che calcola
- Confrontiamo $\mathscr{A}(I)$ con il valore ottimo $z(I)$, per problemi di minimo e di massimo
- Vediamo dove si collocano gli algoritmi di approssimazione e il compromesso alla base di ogni euristica

Algoritmo euristico

Definizione 1: Algoritmo euristico

Dato un ***problema di ottimizzazione*** $\mathscr{P}$, *sia* $z(I)$ *il* ***valore ottimo*** *di* $\mathscr{P}$ *su un'****istanza*** I. *Un* ***algoritmo euristico*** *per* $\mathscr{P}$ *è un algoritmo che, per ogni istanza* I *di* $\mathscr{P}$, *calcola una soluzione ammissibile di* I, *senza garantire che sia ottima. Indichiamo con* $\mathscr{A}(I)$ *il valore della soluzione calcolata dall'algoritmo euristico* $\mathscr{A}$ *sull'istanza* I.

Valore euristico e valore ottimo

- Poiché la soluzione calcolata da $\mathscr{A}$ è ammissibile, per ogni istanza I

$$\begin{cases} \mathscr{A}(I) \geq z(I) & \text{se } \mathscr{P} \text{ è un problema di minimo,} \\ \mathscr{A}(I) \leq z(I) & \text{se } \mathscr{P} \text{ è un problema di massimo.} \end{cases}$$

- Il valore euristico non è mai migliore del valore ottimo: nel caso migliore è uguale.

Perché servono gli algoritmi euristici

- I quattro problemi usati come esempi sono tutti **NP-hard**:
  - il problema dello zaino 0/1;
  - il problema del bin packing;
  - il problema del commesso viaggiatore;
  - il problema di scheduling $P \,||\, C_{\max}$.

- Un algoritmo esatto può richiedere un tempo di esecuzione che cresce esponenzialmente con la dimensione dell'istanza.

Algoritmi di approssimazione

- Gli **algoritmi di approssimazione** sono algoritmi euristici con una garanzia *dimostrata* nel caso peggiore sul rapporto fra $\mathscr{A}(I)$ e $z(I)$.
- La maggior parte degli algoritmi euristici usati in pratica non ha questa garanzia.
- La loro qualità si valuta sulle istanze da risolvere, come descritto nella sezione «Come valutare un algoritmo euristico» di questo capitolo.

Qualità e tempo di esecuzione

Un algoritmo euristico scambia la **qualità** della soluzione con il **tempo di esecuzione**.

- Lo scopo di un'euristica è calcolare, in poco tempo, una soluzione il cui valore sia vicino a $z(I)$.

Riepilogo

Algoritmi euristici:

- calcolano una soluzione ammissibile di ogni istanza, senza garantire che sia ottima
- $\mathscr{A}(I) \geq z(I)$ per un problema di minimo, $\mathscr{A}(I) \leq z(I)$ per un problema di massimo

Algoritmi di approssimazione:

- algoritmi euristici con una garanzia dimostrata nel caso peggiore
- la maggior parte delle euristiche usate in pratica non ne ha: scambiano qualità con tempo di esecuzione

I problemi e le ipotesi

Introduzione: I problemi e le ipotesi

- Gli algoritmi euristici sono illustrati su quattro problemi classici di ottimizzazione combinatoria
- In questa sezione ne richiamiamo la notazione e le ipotesi sulle istanze
- Fissiamo un'istanza di riferimento per ogni problema, con valore ottimo noto: su queste la qualità di ogni soluzione euristica si può misurare esattamente

Quattro problemi classici

- Gli algoritmi euristici sono illustrati su quattro problemi classici di ottimizzazione combinatoria:
  - il problema dello zaino 0/1 (KP);
  - il problema del bin packing (BPP);
  - il problema del commesso viaggiatore (TSP);
  - $P \,||\, C_{\max}$.
- Le loro definizioni e ipotesi sono nelle rispettive pagine; la tabella della slide successiva richiama la notazione.

La notazione dei quattro problemi

problema	dati	soluzione	obiettivo
KP	oggetti $\mathscr{I}$, profitti p_j, pesi w_j, capacità c	sottoinsieme di oggetti	max profitto totale
BPP	oggetti $\mathscr{I}$, pesi w_j, capacità c	schemi di riempimento	min numero di contenitori
TSP	città $\mathscr{V}$, distanze d_{ij}	giro σ	min lunghezza $L(\sigma)$
$P \mid\mid C_{\max}$	lavori $\mathscr{J}$, tempi t_j, macchine m	insiemi $J_1, \ldots, J_m$	min makespan

Ipotesi

- Assumiamo che ogni istanza soddisfi le ipotesi del suo problema:
  - KP e BPP: $w_j \leq c$ per tutti gli oggetti e $\sum_j w_j > c$;
  - TSP: $n \geq 3$ e distanze simmetriche;
  - $P \,||\, C_{\max}$: $n > m$.
- Le quattro istanze seguenti si usano in tutte le note sugli algoritmi euristici.
- I loro valori ottimi sono noti, quindi la qualità di ogni soluzione euristica si può misurare esattamente.

Esempio: istanza di riferimento del KP

Uno zaino di capacità $c = 9$ e 4 oggetti con profitti e pesi

	$j = 1$	$j = 2$	$j = 3$	$j = 4$
p_j	10	7	6	4
w_j	5	4	3	3

- La soluzione ottima è $\tilde{S} = \{1, 2\}$, di peso 9 e profitto $z(I) = 17$.

Esempio: istanza di riferimento del BPP

Contenitori di capacità $c = 7$ e 6 oggetti con pesi

	$j = 1$	$j = 2$	$j = 3$	$j = 4$	$j = 5$	$j = 6$
w_j	4	4	5	3	2	3

- Il peso totale è $21 = 3c$, quindi servono almeno 3 contenitori.
- Gli schemi di riempimento $\{1, 4\}$, $\{2, 6\}$, $\{3, 5\}$ usano esattamente 3 contenitori, tutti pieni: $z(I) = 3$.

Esempio: istanza di riferimento del TSP

5 città con le distanze simmetriche

d_{ij}	1	2	3	4	5
1	—	5	2	2	9
2	5	—	4	3	4
3	2	4	—	4	7
4	2	3	4	—	7
5	9	4	7	7	—

- L'istanza è metrica.
- Fra i suoi 12 giri distinti, l'unico ottimo è $\sigma = (1, 3, 5, 2, 4)$, di lunghezza $z(I) = 18$.

Esempio: istanza di riferimento di $P \,||\, C_{\max}$

$m = 3$ macchine identiche e 7 lavori con tempi di lavorazione

	$j=1$	$j=2$	$j=3$	$j=4$	$j=5$	$j=6$	$j=7$
t_j	5	5	4	4	3	3	3

- Il tempo di lavorazione totale è 27, quindi il makespan è almeno $\lceil 27/3 \rceil = 9$.
- Gli insiemi $J_1 = \{1, 3\}$, $J_2 = \{2, 4\}$, $J_3 = \{5, 6, 7\}$ lo raggiungono: $z(I) = 9$.

Riepilogo

I quattro problemi:

- KP e BPP (oggetti, pesi, capacità), TSP (città, distanze), $P \,||\, C_{\max}$ (lavori, tempi, macchine)
- ogni istanza soddisfa le ipotesi del suo problema

Le istanze di riferimento:

- una piccola istanza per ogni problema, usata in tutte le note sugli algoritmi euristici
- valori ottimi: $z(I) = 17$ (KP), 3 (BPP), 18 (TSP), 9 ($P \,||\, C_{\max}$)

Bound primali e duali

Introduzione: Bound primali e duali

- Un'euristica dà una soluzione, ma quanto è buona? In questa sezione impariamo a rispondere anche quando il valore ottimo è ignoto
- Distinguiamo i bound primali, dati da soluzioni ammissibili, dai bound duali, dati da rilassamenti o da semplici proprietà del problema
- Definiamo il gap, dimostriamo che limita l'errore relativo e lo calcoliamo sulle quattro istanze di riferimento

Bound primali e bound duali

- Il valore di *qualunque* soluzione ammissibile limita il valore ottimo da un lato, e si chiama **bound primale**.
- Un algoritmo euristico è un metodo per calcolare bound primali.
- Un bound sul valore ottimo dall'*altro* lato, ottenuto senza calcolare una soluzione ammissibile, si chiama **bound duale**.
- I bound duali vengono da rilassamenti del problema, come il rilassamento lineare, o da semplici proprietà del problema.

Upper bound e lower bound

- Per un problema di minimo un bound primale è un **upper bound** UB e un bound duale è un **lower bound** LB; per un problema di massimo i ruoli si scambiano.

- In entrambi i casi il valore ottimo è compreso fra i due bound:

$$\begin{cases} \mathrm{LB}(I) \leq z(I) \leq \mathscr{A}(I) & \text{se } \mathscr{P} \text{ è un problema di minimo,} \\ \mathscr{A}(I) \leq z(I) \leq \mathrm{UB}(I) & \text{se } \mathscr{P} \text{ è un problema di massimo.} \end{cases}$$

Il gap

- Il **gap** di un valore euristico $\mathscr{A}(I)$ è il gap relativo $\frac{\mathrm{UB}-\mathrm{LB}}{\mathrm{LB}}$ definito nel modulo *Problemi di ottimizzazione combinatoria*, «Bound primali, bound duali e gap»: la differenza dei due bound divisa per il **minore**.

- Con il valore euristico al posto del bound primale diventa

$$gap(I) = \begin{cases} \dfrac{\mathscr{A}(I) - \mathrm{LB}(I)}{\mathrm{LB}(I)} & \text{se } \mathscr{P} \text{ è un problema di minimo,} \\ \dfrac{\mathrm{UB}(I) - \mathscr{A}(I)}{\mathscr{A}(I)} & \text{se } \mathscr{P} \text{ è un problema di massimo.} \end{cases}$$

L'errore relativo

- L'**errore relativo** di $\mathscr{A}$ su I è

$$\begin{cases} \dfrac{\mathscr{A}(I) - z(I)}{z(I)} & \text{per un problema di minimo,} \\ \dfrac{z(I) - \mathscr{A}(I)}{z(I)} & \text{per un problema di massimo.} \end{cases}$$

- Non si può calcolare quando $z(I)$ è ignoto, che è la situazione tipica.

- Il gap, invece, usa solo quantità calcolate.

Il gap limita l'errore relativo

Che relazione c'è fra il gap, che sappiamo calcolare, e l'errore relativo, che di solito non sappiamo calcolare?

Proposizione 1

Per ogni istanza I, l'errore relativo di $\mathscr{A}$ su I è al più $gap(I)$.

Dimostrazione (1/2)

Per un problema di minimo, $\mathrm{LB}(I) \leq z(I)$, quindi

$$\frac{\mathscr{A}(I)}{z(I)} \leq \frac{\mathscr{A}(I)}{\mathrm{LB}(I)}$$

e

$$\frac{\mathscr{A}(I) - z(I)}{z(I)} = \frac{\mathscr{A}(I)}{z(I)} - 1$$

$$\leq \frac{\mathscr{A}(I)}{\mathrm{LB}(I)} - 1$$

$$= gap(I).$$

Dimostrazione (2/2)

Per un problema di massimo, $\mathscr{A}(I) \leq z(I) \leq \mathrm{UB}(I)$, quindi

$$\frac{z(I) - \mathscr{A}(I)}{z(I)} \leq \frac{z(I) - \mathscr{A}(I)}{\mathscr{A}(I)}$$

$$\leq \frac{\mathrm{UB}(I) - \mathscr{A}(I)}{\mathscr{A}(I)}$$

$$= gap(I).$$

□

Il gap è un certificato

- Il gap è quindi un **certificato**:
  - se $gap(I) \leq 5\%$, la soluzione dista al più il 5% dall'ottimo, anche se l'ottimo è ignoto;
  - se $gap(I) = 0$, la soluzione è ottima.
- Calcoliamo ora bound primali, bound duali e gap sulle quattro istanze di riferimento.

Esempio: bound e gap sulle istanze di riferimento (1/5)

KP.

- La soluzione $\{1, 3\}$ ha peso $8 \leq 9$ e profitto $\mathscr{A}(I) = 16$.
- Il rilassamento lineare, con gli oggetti ordinati per efficienza non crescente $(1, 3, 2, 4)$, inserisce gli oggetti 1 e 3 e una frazione $\frac{1}{4}$ dell'oggetto 2, quindi

$$z(I) \leq 10 + 6 + 7 \cdot \tfrac{1}{4} = 17.75$$

  e, poiché i profitti sono interi, $\mathrm{UB}(I) = 17$.

- Quindi

$$gap(I) = \frac{17 - 16}{16} = 6.25\%,$$

  mentre l'errore relativo è $\frac{17-16}{17} \approx 5.9\%$.

Esempio: bound e gap sulle istanze di riferimento (2/5)

BPP.

- Gli schemi di riempimento $\{1\}$, $\{2\}$, $\{3\}$, $\{4,5\}$, $\{6\}$ sono ammissibili e usano $\mathscr{A}(I) = 5$ contenitori.
- Il bound duale è
$$\mathrm{LB}(I) = \left\lceil \frac{21}{7} \right\rceil = 3.$$
- Quindi
$$gap(I) = \frac{5-3}{3} \approx 66.7\%.$$

Esempio: bound e gap sulle istanze di riferimento (3/5)

TSP.

- Il giro $(1, 3, 2, 4, 5)$ ha lunghezza

$$\mathscr{A}(I) = 2 + 4 + 3 + 7 + 9 = 25.$$

- Un bound duale è dato dalla proposizione sulle due distanze minime di ogni città, qualche slide più avanti: $\mathrm{LB}(I) = 17$.

- Quindi

$$gap(I) = \frac{25 - 17}{17} \approx 47.1\%,$$

mentre l'errore relativo è $\frac{25-18}{18} \approx 38.9\%$.

Esempio: bound e gap sulle istanze di riferimento (4/5)

$P \,||\, C_{\max}$.

- La soluzione con $J_1 = \{1, 5, 7\}$, $J_2 = \{2, 6\}$, $J_3 = \{3, 4\}$ ha makespan $\mathscr{A}(I) = 11$.
- Il bound duale è

$$\mathrm{LB}(I) = \max\{5, \lceil 27/3 \rceil\} = 9.$$

- Quindi

$$gap(I) = \frac{11 - 9}{9} \approx 22.2\%.$$

Esempio: bound e gap sulle istanze di riferimento (5/5)

- Sul BPP e su $P \,||\, C_{\max}$ il bound duale è uguale a $z(I)$, quindi il gap è uguale all'errore relativo.
- Anche sul KP il bound duale è uguale a $z(I)$, ma il gap divide per il valore minore $\mathscr{A}(I)$ ed è leggermente più grande.
- Sul TSP il bound duale è più debole, e il gap sovrastima l'errore relativo.

Un bound duale per il TSP

Da dove viene il bound duale 17 del TSP? Dai due spigoli più corti in ogni città.

Proposizione 2

Per ogni istanza I del TSP, siano a_i e b_i le due distanze minime dalla città i alle altre città. Allora

$$z(I) \geq \left\lceil \frac{1}{2} \sum_{i \in \mathcal{V}} (a_i + b_i) \right\rceil .$$

Dimostrazione

- In ogni giro ogni città è estremo di esattamente due spigoli, di lunghezza almeno a_i e b_i.
- Sommando le lunghezze dei due spigoli incidenti in ogni città si conta due volte ogni spigolo del giro, quindi

$$2\,L(\sigma) = \sum_{i \in \mathcal{V}} \big(\text{lunghezze dei due spigoli incidenti in } i\big)$$

$$\geq \sum_{i \in \mathcal{V}} (a_i + b_i)$$

  per ogni giro σ, e in particolare per uno ottimo.
- Poiché le distanze sono intere, $z(I)$ è un intero, e il bound si può arrotondare per eccesso.

□

Il bound sull'istanza di riferimento del TSP

- Le due distanze minime sono:
  - $(2, 2)$ per la città 1;
  - $(3, 4)$ per la città 2;
  - $(2, 4)$ per la città 3;
  - $(2, 3)$ per la città 4;
  - $(4, 7)$ per la città 5.

- Quindi

$$\mathrm{LB}(I) = \left\lceil \frac{4 + 7 + 6 + 5 + 11}{2} \right\rceil = \lceil 16.5 \rceil = 17.$$

Un lato solo non basta

Un algoritmo euristico dà solo un lato del gap.

- Senza un bound duale sull'altro lato, non si può dire nulla sulla qualità della soluzione, per quanto buona possa sembrare.

Riepilogo (1/2)

Bound primali e duali:

- bound primale: il valore di una soluzione ammissibile, come $\mathscr{A}(I)$
- bound duale: da un rilassamento o da una semplice proprietà, senza una soluzione ammissibile
- il valore ottimo è compreso fra i due

Gap ed errore relativo:

- il gap divide la differenza dei due bound per il minore
- l'errore relativo richiede $z(I)$; il gap usa solo quantità calcolate, e lo limita

Riepilogo (2/2)

Sulle istanze di riferimento:

- gap 6.25% (KP), 66.7% (BPP), 47.1% (TSP), 22.2% ($P \,||\, C_{\max}$)
- il bound duale 17 del TSP viene dalle due distanze minime di ogni città

La lezione:

- un'euristica dà solo un lato del gap
- senza un bound duale, non si può dire nulla sulla qualità di una soluzione

Classificazione degli algoritmi euristici

Introduzione: Classificazione degli algoritmi euristici

- Gli algoritmi euristici sono molti: per orientarci li raggruppiamo in classi
- Le classi dipendono da come gli algoritmi producono e usano le soluzioni
- Ogni classe si studia in dettaglio più avanti nel corso

Quattro classi (1/2)

Gli algoritmi euristici si classificano in base a come producono e usano le soluzioni, in quattro classi.

classe	idea
euristiche costruttive	costruiscono una soluzione passo per passo, estendendo una soluzione parziale con un elemento alla volta; una scelta, una volta fatta, non viene mai rivista
ricerca locale	parte da una soluzione ammissibile e la sostituisce ripetutamente con una soluzione migliore ottenuta con una piccola modifica, finché nessuna piccola modifica la migliora

Quattro classi (2/2)

classe	idea
metaeuristiche	strategie generali che guidano una ricerca locale oltre il punto in cui si ferma, esplorando altre regioni delle soluzioni
euristiche basate sulla PLI	usano una formulazione di programmazione lineare intera mista, il suo rilassamento e un solver MIP per guidare la ricerca o per esplorare grandi insiemi di soluzioni

Le classi lavorano insieme

- Le classi non sono alternative:
  - una ricerca locale ha bisogno di una soluzione di partenza, di solito calcolata da un'euristica costruttiva;
  - una metaeuristica contiene una o più ricerche locali;
  - un'euristica basata sulla PLI può partire dalla soluzione di una qualunque delle altre, e migliorarla.

Riepilogo

Quattro classi:

- euristiche costruttive: una soluzione, costruita passo per passo, scelte mai riviste
- ricerca locale: piccole modifiche migliorative, finché nessuna migliora
- metaeuristiche: guidano una ricerca locale oltre il punto in cui si ferma
- euristiche basate sulla PLI: una formulazione, il suo rilassamento e un solver MIP

Non alternative:

- le classi si combinano: ognuna può partire dalle altre, o contenerle

Rappresentazione delle soluzioni e spazio di ricerca

Introduzione: Rappresentazione delle soluzioni e spazio di ricerca

- Prima di progettare un'euristica, dobbiamo decidere come memorizzare una soluzione in una struttura dati
- Questa scelta fissa lo spazio di ricerca, e quali modifiche di una soluzione sono facili da eseguire
- Fissiamo una rappresentazione per ognuno dei quattro problemi, e vediamo quanto sono grandi gli spazi di ricerca

Rappresentazione e spazio di ricerca

- Un algoritmo euristico manipola soluzioni, che vanno memorizzate in una struttura dati: la **rappresentazione** delle soluzioni.
- L'insieme di tutti gli oggetti che la rappresentazione può descrivere è lo **spazio di ricerca**.
- La rappresentazione è una scelta di progetto, e determina quali modifiche di una soluzione sono facili da eseguire.
- Per i quattro problemi usiamo le rappresentazioni della slide successiva.

Rappresentazioni dei quattro problemi

problema	rappresentazione	dimensione dello spazio di ricerca	ammissibile?
KP	vettore binario $x \in \{0,1\}^n$	2^n	non sempre: la capacità può essere superata
BPP	vettore *bin*, con bin_j il contenitore dell'oggetto j	al più n^n	non sempre: un contenitore può essere troppo pieno
TSP	permutazione σ delle città	$\frac{(n-1)!}{2}$ giri distinti	sempre
$P \mid\mid C_{\max}$	vettore *mach*, con $mach_j$ la macchina del lavoro j	m^n	sempre

Oggetti inammissibili e spazi di ricerca grandi

- Quando la rappresentazione può descrivere oggetti inammissibili, un'euristica deve limitarsi a quelli ammissibili oppure accettare quelli inammissibili e penalizzarli.

- Gli spazi di ricerca crescono molto rapidamente: un TSP con $n = 20$ città ha

$$\frac{19!}{2} \approx 6 \cdot 10^{16}$$

  giri distinti.
- Un'euristica esplora solo una minima parte dello spazio di ricerca, e la domanda è *quale* parte.

Riepilogo

Rappresentazione e spazio di ricerca:

- la rappresentazione è la struttura dati che memorizza una soluzione: una scelta di progetto
- lo spazio di ricerca è l'insieme di tutti gli oggetti che può descrivere

I quattro problemi:

- KP e BPP: la rappresentazione può descrivere oggetti inammissibili; TSP e $P \,||\, C_{\max}$: sempre ammissibili
- gli spazi di ricerca crescono molto rapidamente: un'euristica ne esplora solo una minima parte

Intorni e ottimi locali

Introduzione: Intorni e ottimi locali

- La ricerca locale e le metaeuristiche passano da una soluzione a una "vicina": in questa sezione rendiamo precisa questa idea
- Definiamo intorni, mosse e ottimi locali, e confrontiamo due intorni sull'istanza di riferimento del KP
- Vediamo perché, per i problemi NP-hard, un intorno è sempre un compromesso fra dimensione e qualità

Soluzioni ammissibili e loro valori

- Sia $\mathscr{F}(I)$ l'insieme delle soluzioni ammissibili dell'istanza I.
- Sia $f(S)$ il valore della soluzione $S \in \mathscr{F}(I)$.
- Allora $z(I)$ è il minimo (o il massimo) di f su $\mathscr{F}(I)$.

Intorno

Definizione 2: Intorno

*Un **intorno** è una funzione N che associa a ogni soluzione ammissibile $S \in \mathscr{F}(I)$ un insieme $N(S) \subseteq \mathscr{F}(I)$ di soluzioni ammissibili, i **vicini** di S. Una **mossa** è la trasformazione di S in uno dei suoi vicini.*

- Un intorno si definisce di solito con una famiglia di piccole modifiche della rappresentazione: cambiare una componente del vettore x, spostare un lavoro su un'altra macchina, sostituire due spigoli di un giro.

Ottimo locale

Definizione 3: Ottimo locale

Una soluzione ammissibile $S \in \mathscr{F}(I)$ è un ***ottimo locale*** *rispetto all'intorno N se nessun vicino di S è migliore di S, cioè per ogni $S' \in N(S)$:*

$$\begin{cases} f(S) \leq f(S') & \text{se } \mathscr{P} \text{ è un problema di minimo,} \\ f(S) \geq f(S') & \text{se } \mathscr{P} \text{ è un problema di massimo.} \end{cases}$$

Una soluzione ottima si chiama anche ***ottimo globale****.*

Ottimi locali e ottimi globali

- Un ottimo globale è un ottimo locale per ogni intorno.
- Un ottimo locale non è necessariamente un ottimo globale.
- Inoltre, **essere un ottimo locale dipende dall'intorno**: la stessa soluzione può essere un ottimo locale per un intorno e non per un altro.
- L'esempio seguente lo mostra sull'istanza di riferimento del KP.

Esempio: due intorni per il KP (1/2)

Sull'istanza di riferimento del KP ($c = 9$), consideriamo la soluzione $x = (1, 0, 1, 0)$, cioè gli oggetti $\{1, 3\}$, di peso 8 e profitto 16.

L'intorno **flip** cambia una componente di x, cioè aggiunge o toglie un oggetto:

vicino	peso	profitto	ammissibile
$(0, 0, 1, 0)$	3	6	sì
$(1, 1, 1, 0)$	12	—	no
$(1, 0, 0, 0)$	5	10	sì
$(1, 0, 1, 1)$	11	—	no

I vicini ammissibili hanno profitto 6 e 10, minore di 16: x è un **ottimo locale** rispetto all'intorno flip.

Esempio: due intorni per il KP (2/2)

L'intorno **swap** toglie un oggetto scelto e aggiunge un oggetto non scelto:

vicino	peso	profitto	ammissibile
$(0,1,1,0)$: esce 1, entra 2	7	13	sì
$(0,0,1,1)$: esce 1, entra 4	6	10	sì
$(1,1,0,0)$: esce 3, entra 2	9	17	sì
$(1,0,0,1)$: esce 3, entra 4	8	14	sì

Il vicino $(1,1,0,0)$ ha profitto $17 > 16$: x **non** è un ottimo locale rispetto all'intorno swap, e la mossa porta all'ottimo globale.

Intorno esatto

Definizione 4: Intorno esatto

*Un intorno N è **esatto** se ogni ottimo locale rispetto a N è un ottimo globale.*

- Gli intorni esatti sono rari, ed esistono per problemi risolvibili in tempo polinomiale:
  - l'intorno del metodo del simplesso, che si sposta fra vertici adiacenti del poliedro, è esatto per la programmazione lineare;
  - l'intorno che scambia uno spigolo entrante e uno uscente è esatto per il problema dell'albero ricoprente di costo minimo.

Dimensione e qualità

- Per i problemi NP-hard gli intorni piccoli non sono esatti.
- Gli intorni esatti, come quello che contiene tutte le soluzioni ammissibili, sono troppo grandi per essere esplorati.
- Il progetto di un intorno è un compromesso fra:
  - la sua **dimensione**, che determina il tempo necessario per esplorarlo;
  - la **qualità** dei suoi ottimi locali.

Riepilogo

Intorni:

- un intorno $N(S)$ è un insieme di soluzioni ammissibili vicine a S; una mossa porta a una di esse
- di solito definito da piccole modifiche della rappresentazione

Ottimi locali:

- nessun vicino è migliore; essere un ottimo locale dipende dall'intorno (flip e swap sul KP)
- gli intorni esatti sono rari: dimensione contro qualità degli ottimi locali

Intensificazione e diversificazione

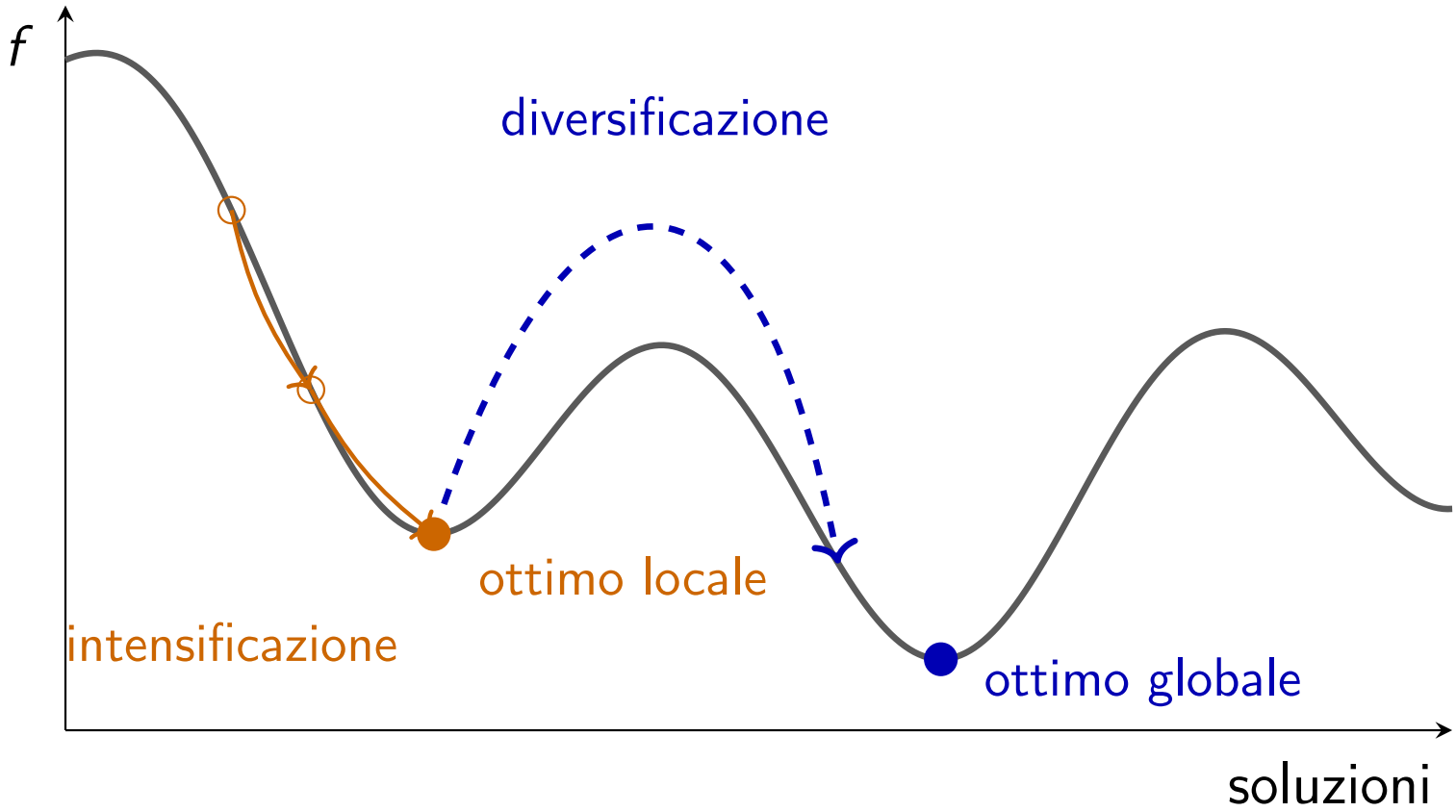
Introduzione: Intensificazione e diversificazione

- Un'immagine aiuta a vedere come un'euristica si muove fra le soluzioni: il paesaggio dei loro valori
- Su di esso vediamo le due forze di ogni ricerca: scendere in una valle, e lasciarla per trovarne una più profonda
- Bilanciarle è l'idea chiave delle metaeuristiche, più avanti nel corso

Il paesaggio

- I valori delle soluzioni, visti come funzione sullo spazio di ricerca, formano il **paesaggio** esplorato da un'euristica.
- Per un problema di minimo gli ottimi locali sono i fondi delle valli.
- L'ottimo globale è il fondo della valle più profonda.

Il paesaggio di un problema di minimo



Intensificazione e diversificazione

- L'**intensificazione** è la ricerca di soluzioni migliori nella regione dove sono già state trovate buone soluzioni: si scende verso il fondo della valle corrente.
- La ricerca locale è pura intensificazione, e si ferma al primo ottimo locale che raggiunge.
- La **diversificazione** è lo spostamento verso regioni dello spazio di ricerca non ancora esplorate: si lascia la valle corrente, eventualmente passando per soluzioni peggiori, per raggiungerne un'altra.

Bilanciare le due

- Troppa intensificazione intrappola la ricerca nella prima valle.
- Troppa diversificazione la fa vagare senza mai esplorare una valle fino in fondo.
- Ogni metaeuristica è un modo di bilanciare le due.

Riepilogo

Il paesaggio:

- i valori delle soluzioni sullo spazio di ricerca
- gli ottimi locali sui fondi delle valli, l'ottimo globale sul fondo della più profonda

Due forze:

- intensificazione: giù nella valle corrente (la ricerca locale è pura intensificazione)
- diversificazione: fuori dalla valle, verso regioni inesplorate
- ogni metaeuristica bilancia le due

Come valutare un algoritmo euristico

Introduzione: Come valutare un algoritmo euristico

- La maggior parte delle euristiche non ha garanzie dimostrate: come decidiamo se una è buona?
- In questa sezione vediamo i due criteri, qualità e tempo di esecuzione, e come misurarli
- Vediamo perché una sola esecuzione su una sola istanza non dimostra nulla, e come trattare le scelte casuali

Qualità e tempo di esecuzione

- Un algoritmo euristico si valuta in base alla **qualità** delle sue soluzioni e al suo **tempo di esecuzione**, che sono obiettivi in conflitto.
- La **qualità** si misura con il gap rispetto al miglior bound duale disponibile, o con l'errore relativo quando il valore ottimo è noto, come sulle istanze di riferimento di questo capitolo.

Analisi del caso peggiore e analisi sperimentale

- L'**analisi del caso peggiore**, come per gli algoritmi di approssimazione, dà garanzie valide per tutte le istanze, ma spesso lontane dal comportamento tipico.
- L'**analisi sperimentale** misura i gap e i tempi di esecuzione su un insieme di istanze di benchmark.
- L'insieme di benchmark deve essere abbastanza grande e vario da evitare conclusioni valide solo per poche istanze.

Una singola istanza non dimostra nulla

- Una singola istanza non dimostra nulla: sull'istanza di riferimento del TSP la stessa regola, applicata da città di partenza diverse, dà giri di lunghezza fra 18 e 25.
- I confronti fra euristiche devono usare le stesse istanze e riportare la distribuzione dei risultati, non solo il migliore.

Scelte casuali e semi

- Per le euristiche che fanno scelte casuali, i risultati dipendono dai numeri casuali.
- Ogni istanza va risolta più volte, con diversi **semi** del generatore di numeri casuali.
- Si riportano la media e la dispersione dei risultati.

Che cosa rende buona un'euristica

Un algoritmo euristico è buono se,
sulle istanze a cui è destinato,
trova con regolarità soluzioni con un gap piccolo
nel tempo disponibile.

Riepilogo

Due criteri in conflitto:

- qualità (gap, o errore relativo quando $z(I)$ è noto) e tempo di esecuzione
- l'analisi del caso peggiore dà garanzie; l'analisi sperimentale misura il comportamento tipico

Esperimenti corretti:

- molte istanze varie, le stesse per tutte le euristiche, l'intera distribuzione dei risultati
- metodi casuali: più semi, con la media e la dispersione

Errori tipici

Introduzione: Errori tipici

- Raccogliamo gli errori più frequenti con gli algoritmi euristici e i loro bound
- Ognuno viene da una definizione o da un risultato di questo capitolo
- Controllateli prima di risolvere gli esercizi

Errori tipici (1/3)

Osservazione 1: Giudicare un'euristica da una sola esecuzione o da una sola istanza

La stessa regola, da città di partenza diverse, dà giri di lunghezza fra 18 *e* 25 *sull'istanza di riferimento del TSP.*

Osservazione 2: Confondere il gap con l'errore relativo

Il gap si misura rispetto a un bound duale, quindi non è mai minore dell'errore relativo; in un problema di minimo coincidono solo quando il bound duale è uguale a $z(I)$.

Errori tipici (2/3)

Osservazione 3: Usare il denominatore sbagliato

Il gap divide sempre per il bound minore: per un problema di massimo è

$$\frac{\mathrm{UB}(I) - \mathscr{A}(I)}{\mathscr{A}(I)},$$

per un problema di minimo

$$\frac{\mathscr{A}(I) - \mathrm{LB}(I)}{\mathrm{LB}(I)}.$$

Errori tipici (3/3)

Osservazione 4: Dimenticare di arrotondare il bound duale

Con dati interi il bound 17.75 *del KP si arrotonda per difetto a* 17*, e il bound* 16.5 *del TSP si arrotonda per eccesso a* 17*.*

Osservazione 5: Fidarsi di una soluzione che sembra buona

Senza un bound duale sull'altro lato, non si può dire nulla sulla qualità di una soluzione.

Riepilogo

Valutare un'euristica:

- mai da una sola esecuzione o da una sola istanza
- senza un bound duale, una soluzione che sembra buona non dimostra nulla

Calcolare il gap:

- il gap non è l'errore relativo: non è mai minore
- si divide per il bound minore, e con dati interi si arrotonda il bound duale

In sintesi

Introduzione: In sintesi

- Raccogliamo le idee chiave del capitolo sugli algoritmi euristici
- Il quadro riassuntivo raccoglie i bound trovati sulle quattro istanze di riferimento

Riepilogo (1/2)

- Un algoritmo euristico restituisce una soluzione ammissibile senza garantirne l'ottimalità: il suo valore è un bound *primale*.
- Il gap confronta il valore euristico con il miglior bound duale, ed è un upper bound sull'errore relativo.
- Gli algoritmi euristici sono costruttivi, di ricerca locale, metaeuristiche o basati sulla PLI; un intorno definisce gli ottimi locali, e la ricerca bilancia intensificazione e diversificazione.
- La qualità si valuta su molte istanze e, per i metodi casuali, su molti semi.

Riepilogo (2/2)

Quadro riassuntivo delle istanze di riferimento. Soluzioni euristiche di questo capitolo e loro bound duali:

problema	bound primale	bound duale	gap	$z(I)$
KP (max)	16	$\mathrm{UB}(I) = 17$	6.25%	17
BPP (min)	5	$\mathrm{LB}(I) = 3$	66.7%	3
TSP (min)	25	$\mathrm{LB}(I) = 17$	47.1%	18
$P \mid\mid C_{\max}$ (min)	11	$\mathrm{LB}(I) = 9$	22.2%	9