Complementi ed Esercizi di Informatica Teorica II

Vincenzo Bonifaci

21 maggio 2008

4 Problemi di ottimizzazione: il Bin Packing

Il problema BIN PACKING è il seguente: dato un insieme di n oggetti di dimensioni intere $a_1, \ldots, a_n$ ed un intero b, determinare il minimo numero di contenitori di capacità b che possono contenere gli n oggetti.

Una soluzione a questo problema è rappresentata da una partizione degli n oggetti in contenitori tale che la dimensione totale degli oggetti contenuti in ciascun contenitore non superi b. Il costo della soluzione è dato dal numero di contenitori. Da queste definizioni possiamo facilmente vedere che si tratta di un problema in NPO.

4.1 L'algoritmo First Fit Decreasing

Si possono considerare vari algoritmi per questo problema. Il più semplice, *Next Fit*, costruisce la soluzione inserendo ogni oggetto nell'ultimo contenitore considerato, se c'è abbastanza spazio, altrimenti in un nuovo contenitore. Next Fit è un algoritmo 2-approssimato, il che mostra che BIN PACKING appartiene alla classe APX dei problemi approssimabili con un fattore costante. D'altra parte con la tecnica del gap è possibile mostrare che il problema non è approssimabile con un fattore migliore di $3/2$ a meno che P = NP, quindi non può neanche esistere uno schema di approssimazione polinomiale per questo problema (sempre a meno che P = NP).

Mostriamo che il problema è approssimabile con un fattore di $3/2$ a meno di un fattore additivo pari a 1. Per fare questo utilizziamo l'algoritmo *First Fit Decreasing* che ordina gli oggetti in ordine decrescente di dimensione e poi applica First Fit, ovvero pone ogni oggetto nel primo contenitore con abbastanza spazio tra quelli già usati, oppure se non ci sono contenitori adatti ne usa uno nuovo.

Sia $m(x)$ il numero di contenitori usati da First Fit Decreasing e sia $m^*(x)$ il numero di contenitori usati nella soluzione ottima. Mostriamo che

$$m(x) \leq 1.5m^*(x) + 1.$$

Dividiamo gli oggetti $a_1, \ldots, a_n$, a seconda delle dimensioni, in questi insiemi:

$$\begin{aligned} A &= \{i \,:\, a_i/b > 2/3\}, \\ B &= \{i \,:\, 2/3 \geq a_i/b > 1/2\}, \\ C &= \{i \,:\, 1/2 \geq a_i/b > 1/3\}, \\ D &= \{i \,:\, 1/3 \geq a_i/b\}. \end{aligned}$$

Consideriamo la soluzione trovata da First Fit Decreasing. Se, in questa soluzione, c'è un contenitore contenente solo oggetti dell'insieme D, allora escludendo il contenitore aperto per ultimo (che è sicuramente di questo tipo), ogni contenitore deve essere occupato per almeno 2/3 (altrimenti First Fit Decreasing avrebbe posto uno degli oggetti dell'ultimo contenitore in un contenitore precedente). Quindi $(2/3)(m(x) - 1) \leq \sum_i a_i/b \leq m^*(x)$ e si ha la tesi.

Consideriamo quindi il caso in cui nessun contenitore contenga solo elementi di D. Vogliamo mostrare che in questo caso First Fit Decreasing trova la soluzione ottima. Infatti, sia x' l'istanza ottenuta dall'istanza di partenza x eliminando tutti gli oggetti dell'insieme D. Dato che il numero di contenitori usati da First Fit Decreasing per x e x' è lo stesso, è sufficiente dimostrare l'ottimalità di First Fit Decreasing per x'. A questo scopo osserviamo che, in ogni soluzione ammissibile di x', gli oggetti dell'insieme A non possono condividere il proprio contenitore con nessun altro oggetto; inoltre, ogni contenitore contiene al più due oggetti (e di questi al massimo uno può appartenere all'insieme B). Quindi in ogni contenitore si possono avere solo le seguenti combinazioni di oggetti: A, B, C, BC, CC. La tesi segue osservando che First Fit Decreasing processa gli oggetti in ordine decrescente di dimensione. Quindi l'algoritmo pone ogni oggetto di C con il più grande oggetto di B con il quale possa condividere un contenitore. Questo implica che il numero di contenitori usati dalla soluzione ottima e da quella trovata da First Fit Decreasing è lo stesso.

In effetti, attraverso un'analisi più dettagliata si può dimostrare il vincolo $m(x) \leq 11m^*(x)/9 + 7/9$. Nel caso in cui $m^*(x)$ sia grande, questo vincolo è molto migliore del precedente. Per vedere che il vincolo è stretto, si consideri, per n crescente, la seguente istanza, di $5n$ oggetti (con b un multiplo di 4): n oggetti di dimensione $b/2 + 1$, n oggetti di dimensione $b/4 + 2$, n oggetti di dimensione $b/4 + 1$, $2n$ oggetti di dimensione $b/4 - 2$. È possibile verificare che il rapporto tra il numero di contenitori usati da First Fit Decreasing e quelli usati dall'ottimo è almeno 11/9.

Esercizio 4.1. Verificare che, per l'istanza descritta, $m(x) = \frac{11}{6}n$ e $m^*(x) \leq \frac{3}{2}n$.

Un altro algoritmo per BIN PACKING è l'algoritmo *Best Fit Decreasing*. Come First Fit Decreasing, Best Fit Decreasing ordina inizialmente gli oggetti in ordine non crescente di dimensione e li considera poi sequenzialmente. La differenza tra i due algoritmi sta nella regola usata per scegliere il contenitore in cui inserire gli oggetti: Best Fit Decreasing sceglie il contenitore il cui spazio libero è minimo. In questo modo, l'algoritmo cerca di ridurre la frammentazione massimizzando il numero di contenitori con più spazio libero.

È possibile mostrare che il costo della soluzione trovata da First Fit Decreasing può essere inferiore a quello della soluzione trovata da Best Fit Decreasing. D'altra parte è possibile definire istanze per le quali Best Fit Decreasing trova una soluzione ottima mentre First Fit Decreasing ne trova una non ottima. In ogni caso, il fattore di approssimazione di 11/9 è stretto anche per Best Fit Decreasing come mostra la stessa istanza utilizzata nel caso di First Fit Decreasing.

Come ultima considerazione, notiamo che sia First Fit Decreasing che Best Fit Decreasing sono algoritmi *off-line* mentre Next Fit e First Fit sono algoritmi *on-line*. Infatti, questi ultimi due algoritmi assegnano ogni oggetto a_i ad un contenitore senza conoscere la dimensione degli oggetti a_j con $j > i$. D'altro canto, in un algoritmo off-line l'assegnazione degli oggetti ai contenitori inizia solo una volta che tutte le dimensioni sono note (come nel caso di First Fit Decreasing e Best Fit Decreasing, in cui il primo passo è quello di ordinare gli oggetti).

Esercizio 4.2. Si consideri la seguente istanza di BIN PACKING: $b = 27$ e oggetti di dimensioni 20,11,11,4,3,3,2. Mostrare che per questa istanza First Fit Decreasing usa meno contenitori di Best Fit Decreasing.

4.2 Uno schema di approssimazione asintotica

Abbiamo visto come non esistano schemi di approssimazione polinomiale per BIN PACKING a meno che P = NP. Vogliamo però mostrare l'esistenza di uno schema di approssimazione *asintotica*, ovvero di un algoritmo polinomiale A parametrizzato da un fattore r tale che per ogni $r > 1$, $A(x, r)$ restituisce una soluzione che ha un costo pari al più a $rm^*(x) + c$ per qualche costante c, dove con $m^*(x)$ indichiamo il costo di una soluzione ottima dell'istanza x. La classe dei problemi in NPO che ammettono un tale schema è la classe PTAS$^\infty$.

Lo schema che consideriamo è strutturato nei seguenti passi:

1. Elimina gli oggetti "piccoli".
2. Raggruppa gli oggetti rimanenti in un numero costante gruppi a seconda della dimensione.
3. Trova una soluzione ottima dell'istanza risultante.
4. Ridivide gli oggetti.
5. Reinserisce gli oggetti "piccoli".

Descriviamo prima il passo 3, poi i passi 1 e 2 insieme con i loro "inversi" 5 e 4.

4.2.1 Bin packing ristretto

Per ogni costante intera c e per ogni costante razionale $\delta \leq 1$, consideriamo una versione ristretta di BIN PACKING in cui, per una qualsiasi istanza, sono presenti al più c diverse taglie di oggetti (ovvero l'insieme delle dimensioni degli

oggetti ha cardinalità c) e dove la dimensione di ciascun oggetto è almeno $\delta \cdot B$, dove B indica sempre la dimensione dei contenitori. Per chiarezza chiamiamo questo problema BIN PACKING (c, δ)-RISTRETTO. Osserviamo che un'istanza di questo problema può essere descritta dalla dimensione B dei contenitori e da un multinsieme $I = \{s_1 : n_1, s_2 : n_2, \ldots, s_c : n_c\}$ nel quale la coppia $s_i : n_i$, con $1 \leq i \leq c$, denota il numero n_i di oggetti aventi dimensione s_i.

Mostriamo ora come un'istanza di BIN PACKING (c, δ)-RISTRETTO può essere risolta in tempo $O(n^q)$ dove n è il numero di oggetti e q dipende solamente da c e δ. Facciamo cioé vedere che per ogni scelta delle costanti c e δ, BIN PACKING (c, δ)-RISTRETTO può essere risolto in tempo polinomiale.

Sia (I, B) un'istanza di BIN PACKING (c, δ)-RISTRETTO. Per una data soluzione di questa istanza, il *tipo* di un contenitore è un vettore $t = (t_1, \ldots, t_c)$ di interi con $0 \leq t_i \leq n_i$ tale che $\sum_{i=1}^{c} t_i s_i \leq B$. In altre parole, il tipo specifica, per ogni dimensione s_i tra quelle presenti, il numero di oggetti aventi dimensione s_i presenti nel contenitore. Ricordiamo adesso che per ogni categoria i, $s_i/(\delta B) \geq 1$ poiché la dimensione di ciascun oggetto è almeno δB per ipotesi. Quindi per ogni tipo t,

$$\sum_{i=1}^{c} t_i \leq \frac{1}{\delta} \sum_{i=1}^{c} t_i \frac{s_i}{B} \leq \frac{1}{\delta}.$$

Questo implica che il numero di tipi diversi di contenitori è limitato dal numero di modi in cui si possono scegliere c interi $(t_1, \ldots, t_c)$ la cui somma sia al più $\lfloor 1/\delta \rfloor$. Si può dimostrare che questo numero è pari a

$$q = \binom{c + \lfloor \frac{1}{\delta} \rfloor}{\lfloor \frac{1}{\delta} \rfloor}.$$

In ogni caso, a prescindere dalla particolare formula che determina q, il punto cruciale è che in una qualsiasi istanza di BIN PACKING (c, δ)-RISTRETTO il numero di tipi di contenitore possibili dipende da c e da δ ma *non* dipende dalla particolare istanza.

Dato che ci sono al più q tipi di contenitore, una soluzione ammissibile può ora essere descritta da un vettore a q componenti $(y_1, \ldots, y_q)$ dove y_i specifica, per il tipo di contenitore i-esimo, il numero di contenitori di quel tipo (notiamo che $0 \leq y_i \leq n$ poiché di certo n contenitori sono sufficienti a contenere n oggetti).

È chiaro che il numero di soluzioni ammissibili è limitato da $O(n^q)$, quindi il problema può essere risolto in tempo $O(n^q p(n))$, per un polinomio adatto p, semplicemente generando esaustivamente tutte le soluzioni ammissibili.

Esercizio 4.3. Si consideri l'istanza di BIN PACKING $(3, 3/8)$-RISTRETTO:

$$I = \{3 : 4, 5 : 2, 7 : 1\} \text{ e } B = 8.$$

Quanto valgono c, δ e q in questo caso? Quanti sono i tipi di contenitori ammissibili?

4.2.2 Raggruppamento degli oggetti

Data un'istanza x di BIN PACKING, assumiamo che gli n oggetti siano ordinati in ordine decrescente di dimensione. Se a_i è la dimensione dell'i-esimo oggetto, abbiamo quindi

$$a_1 \geq a_2 \geq \ldots \geq a_n.$$

Fissato un intero $k \leq n$, raggruppiamo gli oggetti in gruppi di k partendo dai più grandi fino ad arrivare ai più piccoli. Siano $G_1, \ldots, G_{m+1}$ i gruppi formati in questo modo, dove $m = \lfloor n/k \rfloor$.

Definiamo poi una nuova istanza x_g di BIN PACKING con la stessa dimensione B dei contenitori e che, per $i = 2, 3, \ldots, m+1$, contiene gli "stessi" oggetti del gruppo G_i dove però ciascun oggetto è stato portato alla dimensione del più grande del suo gruppo. Ad esempio se $x = \langle B, (9, 8, 7, 5, 3, 2, 1)\rangle$ e $k = 2$, abbiamo che $x_g = \langle B, (7, 7, 3, 3, 1)\rangle$. Data una soluzione per l'istanza x_g, possiamo ottenere una soluzione per l'istanza x semplicemente aggiungendo k contenitori dove porre i primi k oggetti. D'altro canto, data una soluzione per l'istanza x, possiamo ottenere una soluzione per l'istanza x_g ponendo ogni oggetto di x_g nel contenitore in cui era stato posto il corrispondente oggetto di categoria superiore di x. Abbiamo quindi

$$m^*(x_g) \leq m^*(x) \leq m^*(x_g) + k$$

ovvero data una soluzione ottimale per x_g possiamo trovare una soluzione per x con errore assoluto al più k.

Esercizio 4.4. Si consideri l'istanza x formata da 11 oggetti le cui dimensioni siano 9, 9, 8, 7, 6, 6, 5, 4, 3, 3 e 3, rispettivamente, e sia $k = 3$. Quali sono i corrispondenti gruppi G_i? Qual è la corrispondente istanza x_g?

4.2.3 Gestione degli oggetti piccoli

Sia x un'istanza di BIN PACKING e, per ogni costante razionale $\delta \in (0, 1/2]$, sia x_δ l'istanza ottenuta eliminando da x tutti gli oggetti con dimensione minore di δB. Data una partizione di x_δ in M contenitori, possiamo usare l'approccio di First Fit per reinserire gli oggetti piccoli. Ovvero, inseriamo ognuno di questi oggetti nel primo contenitore che può contenerlo; se non entra in nessuno dei contenitori disponibili, ricorriamo ad un contenitore nuovo.

Alla fine di questa procedura si possono avere due casi.

1. Nessun nuovo contenitore è stato creato e i contenitori usati sono rimasti M.

2. M' nuovi contenitori sono stati creati ($M' \geq 1$). In questo caso, usando un'analisi simile a quella condotta nel caso di First Fit Decreasing, possiamo dimostrare che tutti i contenitori, eccetto al più uno, hanno spazio libero pari al più a δB. Questo implica che

$$(1 - \delta)(M + M' - 1) \leq \frac{\sum_{i=1}^{n} a_i}{B} \leq m^*(x),$$

ovvero,

$$M + M' \leq \frac{1}{1-\delta} m^*(x) + 1 \leq (1+2\delta)m^*(x) + 1.$$

In conclusione, data una partizione di x_δ in M contenitori, possiamo trovare in tempo polinomiale una soluzione all'istanza x che usa al massimo un numero di contenitori pari a

$$\max(M, (1+2\delta)m^*(x) + 1).$$

Con questo abbiamo completato la descrizione dei cinque passi dell'algoritmo per BIN PACKING. Dobbiamo ancora descrivere come mettere insieme questi passi e come scegliere le costanti δ e k. Notiamo che nel caso in cui $r \geq 2$, è sufficiente applicare l'algoritmo Next Fit per ottenere l'approssimazione desiderata, che nel nostro caso è:

$$m(x) \leq rm^*(x) + 1.$$

Assumiamo quindi, senza perdita di generalità, che il valore di r sia minore di 2. Lo schema di approssimazione è il seguente.

Schema di approssimazione asintotica per BIN PACKING

input Istanza x di BIN PACKING, numero razionale r con $1 < r < 2$;
output Soluzione il cui costo è al più $rm^*(x) + 1$;
begin
$\delta := (r-1)/2$;
Sia x_δ l'istanza ottenuta rimuovendo gli oggetti di dimensione minore di δB;
$k := \lceil (r-1)^2 n'/2 \rceil$ dove n' è il numero di oggetti di x_δ;
Sia $x_{\delta,g}$ l'istanza ottenuta raggruppando gli oggetti di x_δ in gruppi di k;
Trova una soluzione ottima di $x_{\delta,g}$ di costo $m^*(x_{\delta,g})$;
Inserisci i primi k oggetti di x_δ in k nuovi contenitori;
Applica First Fit per reinserire gli oggetti piccoli;
return la partizione così ottenuta
end

Mostriamo intanto che per ogni $r < 2$, questo algoritmo usa tempo polinomiale. È sufficiente mostrare che la soluzione ottima di $x_{\delta,g}$ può essere trovata in tempo polinomiale. Ma $x_{\delta,g}$ è un'istanza di BIN PACKING $(\lfloor n'/k \rfloor, \delta)$-RISTRETTO: dato che, per ogni r fissato, sia $\lfloor n'/k \rfloor$ che δ sono costanti, abbiamo che una soluzione ottima di $x_{\delta,g}$ può essere calcolata in tempo $O(n^q p(n))$ dove q dipende solo da r e p è un polinomio.

Verifichiamo ora il fattore di approssimazione della soluzione trovata. A questo scopo, osserviamo innanzitutto che il costo della soluzione trovata dallo schema prima di applicare First Fit è $m^*(x_{\delta,g}) + k$. Dato che ogni oggetto in x_δ ha dimensione almeno δB, ne segue che $\delta n' \leq m^*(x_\delta)$ e, quindi,

$$k \leq \frac{(r-1)^2}{2} n' + 1 = (r-1)\delta n' + 1 \leq (r-1)m^*(x_\delta) + 1.$$

Dall'analisi svolta sul raggruppamento degli oggetti, abbiamo che l'algoritmo divide gli oggetti di x_δ in un numero di contenitori pari al più a

$$m^*(x_{\delta,g}) + k \leq m^*(x_\delta) + (r-1)m^*(x_\delta) + 1 = rm^*(x_\delta) + 1.$$

Infine, usando il fatto che $r = (1 + 2\delta)$ e l'analisi sul reinserimento degli oggetti piccoli, otteniamo che il numero totale di contenitori usati è al più

$$\max(rm^*(x_\delta) + 1, rm^*(x) + 1) \leq rm^*(x) + 1,$$

il che dimostra il vincolo richiesto.

Abbiamo quindi dimostrato che BIN PACKING appartiene alla classe PTAS$^\infty$. In effetti è possibile dimostrare un risultato più forte, e precisamente che esiste uno schema di approssimazione asintotica che usa un tempo polinomiale sia nella lunghezza dell'input che in $1/(r-1)$.