Approximation preserving reductions

Cesare Lingua

Università degli Studi di Torino

La riduzione:

Idea:

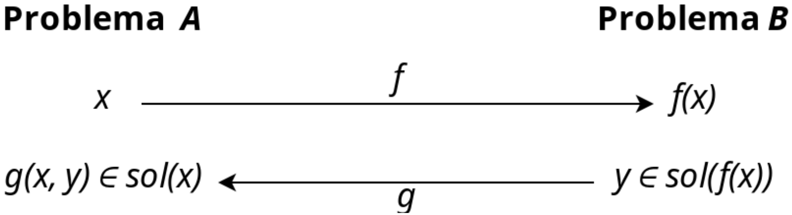
Trasformare un problema A in un altro problema B in modo tale che risolvendo B si possa ricavare anche una soluzione per A.

Vantaggi:

- trasferire tecniche di risoluzione da un problema all'altro;
- studiare la complessità dei problemi, cioè, la classe di complessità a cui appartengono.

Approximation preserving reductions:

- f mappa un istanza x del problema A in un istanza $f(x)$ del problema B
- g mappa una soluzione ammissibile y di B in una soluzione ammissibile $g(x)$ di A con la proprietà che $g(y)$ è una soluzione *approssimata* del problema A



Vantaggi dell'approximation preserving reductions:

- Permette di trasferire le tecniche di *approssimazione* da un problema B ad un problema A;
- Studiare limiti di approssimazione dei problemi, cioè, se sappiamo che il problema B non può essere approssimato oltre un certo limite, lo stesso vale per A;
- Differenziare le classi di complessità.

Problema di Ottimizzazione NP:

Definizione (NPO)

Un *problema di Ottimizzazione* $\Pi \in \mathbf{NP}$, abbreviato **NPO**, è una quadrupla $(I, Sol, \mathbf{m}, goal)$, tale che:

- I è l'insieme delle istanze, riconoscibili in tempo polinomiale, di Π;
- dato $x \in I$, $Sol(x)$ denota l'insieme delle *soluzioni ammissibili* di x; per ogni $y \in Sol(x)$, $|y|$ (la dimensione di y) è polinomiale in $|x|$ (la dimensione di x); dato una qualsiasi istanza x e qualsiasi soluzione y si può decidere in tempo polinomiale se $y \in Sol(x)$;
- data $x \in I$ e $y \in Sol(x)$, $\mathbf{m}(x, y)$ denota il valore di y è può essere caclolato in tempo polinomiale;
- $goal \in \{max, min\}$ denota il *tipo* di problema di ottimizzazione.

Algoritmo di approssimazione:

Dato $\Pi = (I, Sol, m, goal)$, problema **NPO**, una soluzione ottima di un istanza x di Π è indicata con $y^*(x)$ e il suo valore $\mathbf{m}(x, y^*(x))$ con $opt(x)$.

Definizione (Algoritmo di approssimazione)

Dato $\Pi = (I, Sol, m, goal)$, problema **NPO**, un *algorimo di approssimazione* A è un algoritmo che data un instanza x di Π restituisce una soluzione ammissibile $y \in Sol(x)$. Se A esegue in tempo polinomiale rispetto a $|x|$, si dice che A è un *algoritmo di approssimazione polinomiale* di Π.

La qualità della soluzione di un algoritmo di approssimazione A è misurata con il *rapporto di approssimazione* $\rho_A(x)$

$$\rho_A(x) = \max\left\{\frac{\mathbf{m}(x, A(x))}{opt(x)}, \frac{opt(x)}{\mathbf{m}(x, A(x))}\right\}$$

Classi dei problemi di ottimizzazione:

Definizione (APX)

Un problema **NPO** Π appartiene alla classe **APX** se esiste un *algoritmo di approssimazione polinomiale* A e un valore $r \in \mathbb{Q}$, tale che, data un'istanza x di Π, vale, $\rho_A(x) \leq r$. In questo caso A è detto algoritmo di $r-$approssimazione.

Definizione (PTAS)

Un problema **NPO** Π appartiene alla classe **PTAS** se esiste uno *schema di approssimazione polinomiale* A_r, tale che, per ogni $r \in \mathbb{Q}, r \neq 1$ e una qualsiasi istanza x di Π, vale, $\rho_{A_r}(x) \leq r$.

Classi dei problemi di ottimizzazione:

<u>Definizione</u> (FPTAS)

Un problema **NPO** Π appartiene alla classe **FPTAS** se esiste uno <u>*schema*</u> *di approssimazione polinomiale* A_r, tale che, dato un qualsiasi valore $r \in \mathbb{Q}, r \neq 1$, e una qualsiasi istanza x di Π, vale, $\rho_{A_r}(x) \leq r$. Inoltre, esiste un polinomio q che limita il tempo di esecuzione di $A_r(x)$ a $q(x, 1/(r-1))$.

Sotto l'ipotesi di $\mathbf{P} \neq \mathbf{NP}$ le classi sopra descritta formano la segenete catena di inclusioni strette:
$\mathbf{FPTAS} \subset \mathbf{PTAS} \subset \mathbf{APX} \subset \mathbf{NPO}$

Riduzione base: R-riduzione

Definizione (R-riduzione)

Siano Π_1 e Π_2 due problemi **NPO**. Allora diciamo che Π_1 è R-riducibile a Π_2 e si scrive $\Pi_1 \leq_R \Pi_2$, se esistono due funzioni polinomiali f, g che soddisfano le seguenti proprietà:

- $f : I_{\Pi_1} \rightarrow I_{\Pi_2}$, tale che $\forall x_1 \in I_{\Pi_1}, f(x_1) \in I_{\Pi_2}$; in altre parole, data un istanza x_1 in Π_1, f permette di costruire un istanza $x_2 = f(x_1)$ in Π_2;
- $g : I_{\Pi_1} \times Sol_{\Pi_2} \rightarrow Sol_{\Pi_1}$, tale che, $\forall (x_1, y_2) \in (I_{\Pi_1} \times Sol_{\Pi_2}(f(x_1))), g(x_1, y_2) \in Sol_{\Pi_1}(x_1)$; in altre parole, partendo dalla soluzione y_2 dell'istanza x_2, g determina la soluzione $y_1 = g(x_1, y_2)$ dell'istanza iniziale x_1.

Riduzione Lineare: L-riduzione

Definizione (L-riduzione)

Siano Π e Π' due problemi in **NPO**. Allora, diciamo che Π è L-riducibile a Π', e si scrive $\Pi \leq_L \Pi'$, se esistono due funzioni f e g e due costanti $\alpha > 0$ e $\beta > 0$ tali che $\forall x \in I_\Pi$ e $\forall y' \in Sol_{\Pi'}(f(x))$:

- $opt_{\Pi'}(f(x)) \leq \alpha \; opt_\Pi(x)$;
- $|\mathbf{m}_\Pi(x, g(y')) - opt_\Pi(x)| \leq \beta \; |\mathbf{m}_{\Pi'}(f(x), y') - opt_{\Pi'}(f(x))|$.

Proprietà:

Dati due problemi Π e Π', se $\Pi \leq_L \Pi'$ e $\Pi' \in$ **PTAS**,allora, $\Pi \in$ **PTAS**. In altre parole, la L-riduzione preserva l'appartenenza a **PTAS**.

Esempio L-riduzione

MAX k-SAT

Dato un istanza φ composta da m clausole in CNF, ciascuna con k letterali l_i^k, con $i = 1, ..., m$, trovare un assegnamento di valori di verità che massimizzi il numero di clausole soddisfatte.

Teorema

$$MAX\ \text{3-SAT} \leq_L MAX\ \text{2-SAT}$$

Esempio L-riduzione

Dimostrazione

Data un istanza φ di MAX 3-SAT con m clausole C_i con $i = 1, ..., m$, siano $l_i^1 \vee l_i^2 \vee l_i^3$ le tre variabili dell'i-esima clausola, e l_i^1, l_i^2 e l_i^3 possono rappresentare o una variabile oppure la sua negazione.

A ciascuna delle m clausole associamo le seguenti dieci nuove clausole, composte al più da due letterali:
$l_i^1, l_i^2, l_i^3, l_i^4, \overline{l}_i^1 \vee \overline{l}_i^2, \overline{l}_i^1 \vee \overline{l}_i^3, \overline{l}_i^2 \vee \overline{l}_i^3, l_i^1 \vee \overline{l}_i^4, l_i^2 \vee \overline{l}_i^4, l_i^3 \vee \overline{l}_i^4$, dove l_i^4 è una nuova variabile.

Esempio L-riduzione

Sia C'_i la congiunzione delle dieci clausole derivanti da C_i.
Usando la formula $\varphi' = f(\varphi)$ generiamo la congiunzione di tutte le clausole C'_i con $i = 1, ..., m$. Quindi $\varphi' = f(\varphi) = \bigwedge_{i=1}^{m} C'_i$ dove φ' è un istanza di MAX 2-SAT.
Si osserva che:

- ogniqualvolta C_i è *vera*, allora si può trovare un valore di verità per l_i^4 in modo tale che esattamente **sette** clausole in C'_i siano *vere*:
  - se l_i^1, l_i^2, l_i^3 sono tutte *veri*, pongo $l_i^4 = vero$;
  - se esattemente un letterale l_i^1, l_i^2, l_i^3 è *vero*, pongo $l_i^4 = falso$;
  - altrimeni, l_i^4 può essere indifferentemente *vero* o $falso$.
- se C_i è *falsa*, quindi l_i^1, l_i^2, l_i^3 sono tutti *falsi*, allora ponendo l_i^4 a *falso* esattamente sei clausole C'_i sono *vere*.

Esempio L-riduzione

Quindi avremo che:

$$opt(\varphi') = 6m + opt(\varphi)$$

Inoltre, grazie al seguente lemma:

Lemma 1

Data una formula in CNF, almeno la metà delle clausole può sempre essere soddisfatta.

Avremo che, in quanto $\frac{1}{2}m \leq opt(\varphi) \Rightarrow m \leq 2opt(\varphi)$:

$$opt(\varphi') \leq opt(\varphi) + 12opt(\varphi) = 13opt(\varphi)$$

Esempio L-riduzione

Per ogni assegnamento *vero* τ' per le variabili di φ', la restrizione $g(\varphi,\tau') = \tau$, ottenuta eliminando le l_i^4, è tale che:

$$\begin{aligned} opt(\varphi) - \mathbf{m}(\varphi,\tau) &= opt(\varphi') - 6m - \mathbf{m}(\varphi,\tau) \\ &\leq opt(\varphi') - 6m - \mathbf{m}(\varphi',\tau') + 6m \qquad (1) \\ &= opt(\varphi') - \mathbf{m}(\varphi',\tau') \end{aligned}$$

Esempio L-riduzione

Mettendo insieme i risultati ottenuti avremo che:

- $opt(\varphi') \leq 13opt(\varphi)$
- $opt(\varphi) - \mathbf{m}(\varphi, \tau) \leq opt(\varphi') - \mathbf{m}(\varphi', \tau')$

Ponendo $\alpha = 13$ e $\beta = 1$ abbiamo si verificano le due disequanzioni necessarie per avere una L-riduzione

S-riduzione

L-riduzione non garantise un importante caratteristica, cioè che riducendo un problema Π_1 a un problema Π_2, la soluzione y_1 del problema di partenza Π_1 ottenuta grazie *mapping* di g sia almeno "buona" come la soluzione y_2 ottenuta da Π_2.
La *strict reduction* (S-riduzione) garantisce questa propietà

Definizione (S-riduzione)

Siano Π_1 e Π_2 due problemi **NPO**. Allora, diciamo che $\Pi_1 \leq_S \Pi_2$ se esistono due funzioni polinomiali calcolabili f, g che soddisfano la seguente proprietà:

$$\forall x \in I_{\Pi_1}, \forall y \in Sol_{\Pi_2}(f(x)), \rho_{\Pi_2}(f(x), y) \geq \rho_{\Pi_1}(x, g(x, y))$$

S-riduzione

Proprietà 1

Dati due problemi Π_1 e Π_2, se $\Pi_1 \leq_S \Pi_2$ e $\Pi_2 \in$ **APX**, allora $\Pi_1 \in$ **APX**

Proprietà 2

Dati due problemi Π_1 e Π_2, se $\Pi_1 \leq_S \Pi_2$ e $\Pi_2 \in$ **PTAS**, allora $\Pi_1 \in$ **PTAS**

Esempio S-riduzione

Teorema

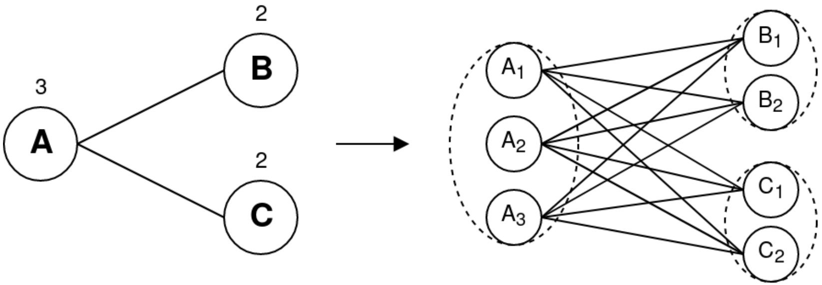
MIN WEIGHTED VERTEX COVER $\leq_S$ MIN VERTEX COVER

Dimostrazione

Consideriamo un istanza $(G(V, E), \vec{w})$ di MIN WEIGHTED VERTEX COVER in cui i pesi dei vertici sono limitati da un polinomio $p(n)$ e mostriamo come trasformarla in un instanza $G'(V', E')$ di MIN VERTEX COVER. Procediamo con i seguenti passaggi:

1. per ogni vertice $v_i \in V$ con peso w_i costruiamo un insieme indipendente W_i con w_i nodi;
2. per ogni arco $(v_i, v_j) \in E$ costruiamo un grafo bipartito completo tra i vertici dei due insiemi indipendenti W_i e W_j in G'

Esempio S-riduzione



Questa operazione è polinomiale poiche il grafo risultante G' ha $\sum_{i=1}^{n} w_i \leq np(n)$ vertici.

Esempio S-riduzione

Consideriamo adesso una copertura minima C' di G' e dimostriamo che C' è composto da l inisiemi indipendenti W_i i cui nodi sono tutti inclusi o esclusi.
Assumiamo per assurdo il contratio:

- consideriamo un insime W_k solo parzialmente incluso in C';
- consideriamo anche gli insiemi indipendenti W_p, connsessi ai vertici di W_k, che possono essere interamente o parzialmente inclusi in C'.

Esempio S-riduzione

Si possono verificare due casi:

1. tutti gli insiemi W_p hanno i loro vertici inclusi in C'; in questo caso C' non sarebbe più una copertura minimale;
2. tra gli inemi W_p esiste almeno un insieme W_q i cui vertici W'_q sono solo parzialmente inclusi in C'.

Nel secondo caso poichè il sottografo di G' prodotto da $W_k \cup W_q$ è bipartito completo, gli archi che connettono i vertici di $W_p \backslash W'_p$ con quelli di $W_q \backslash W'_q$ non sono inclusi nella copertura di C', contraddicendo l'assunzione precedente.
Allora possiamo definire una funzione g nel seguente modo:
Se C' è una copertura di G' e se W_i, $i = 1, ..., l$ sono insiemi indipendenti di C', allora una copertura di C contiene tutti i corrispondenti vertici $v_1, ..., v_l$ di V. La funzione g è polinomiale.

Bibliografia

- Ausiello, Giorgio. (2005). Approximability preserving reduction.
- P. Crescenzi. (1997). A Short Guide to Approximation Preserving Reductions