

¿Toda función en $\#P$ admite un FPRAS?

Proposición

Si una función f admite un FPRAS, entonces $L_f \in BPP$

Demostración: Suponga que $f : \Sigma^* \rightarrow \mathbb{N}$, y sea $\mathcal{A} : \Sigma^* \times \mathbb{R}^+ \rightarrow \mathbb{N}$ un FPRAS para f

Dado $w \in \Sigma^*$, considere el siguiente algoritmo aleatorizado $\mathcal{B}$ para verificar si $w \in L_f$:

1. Sea k el resultado de ejecutar $\mathcal{A}(w, \frac{1}{2})$
2. Si $k > 0$ entonces retorne **sí**, en caso contrario retorne **no**

Una demostración de la proposición

Necesitamos calcular la probabilidad de error del algoritmo $\mathcal{B}$

Suponga primero que $w \in L_f$, vale decir, $f(w) > 0$

Dado que $\mathcal{A}$ es un FPRAS para f , tenemos que:

$$\Pr(|\mathcal{A}(w, \frac{1}{2}) - f(w)| \leq \frac{1}{2} \cdot f(w)) \geq \frac{3}{4}$$

Lo cual es equivalente a:

$$\Pr(\frac{1}{2} \cdot f(w) \leq \mathcal{A}(w, \frac{1}{2}) \leq \frac{3}{2} \cdot f(w)) \geq \frac{3}{4}$$

Una demostración de la proposición

Concluimos entonces que:

$$\Pr\left(\frac{1}{2} \cdot f(w) \leq \mathcal{A}(w, \frac{1}{2})\right) \geq \frac{3}{4}$$

Dado que $f(w) > 0$, tenemos que $\frac{1}{2} \cdot f(w) > 0$, por lo que tenemos lo siguiente:

$$\Pr\left(0 < \mathcal{A}(w, \frac{1}{2})\right) \geq \frac{3}{4}$$

Vale decir, $\Pr(\mathcal{B} \text{ retorne sí}) \geq \frac{3}{4}$

Una demostración de la proposición

Suponga ahora que $w \notin L_f$, vale decir, $f(w) = 0$

Sabemos que:

$$\Pr\left(\frac{1}{2} \cdot f(w) \leq \mathcal{A}(w, \frac{1}{2}) \leq \frac{3}{2} \cdot f(w)\right) \geq \frac{3}{4}$$

Dado que $f(w) = 0$, concluimos entonces que $\Pr(\mathcal{A}(w, \frac{1}{2}) = 0) \geq \frac{3}{4}$

Vale decir, $\Pr(\mathcal{B} \text{ retorne no}) \geq \frac{3}{4}$

De los dos casos concluimos que $\Pr(\mathcal{B} \text{ retorne un resultado incorrecto}) \leq \frac{1}{4}$ $\square$

¿Qué funciones en $\#P$ tienen FPRAS?

Por la proposición anterior no esperamos tener FPRAS para $\#SAT$ y $\#CNF-SAT$

- ▶ A menos que $NP \subseteq BPP$

¿Cada función $f \in \#P$ tal que $L_f \in BPP$ admite un FPRAS?

- ▶ ¿Al menos esto se cumple para cada función $g \in \#P$ tal que $L_g \in PTIME$?

Para responder esta pregunta necesitamos considerar otros problemas $\#P$ -completos

Conjuntos independientes de un grafo

Dado un grafo $G = (N, A)$, decimos que $S \subseteq N$ es un conjunto independiente si para cada $a, b \in S$ se tiene que $(a, b) \notin A$

- Vale decir, los nodos mencionados en S no están conectados entre sí en el grafo

Sea $\text{GIS} = \{(G, k) \mid G \text{ tiene un conjunto independiente } S \text{ con } |S| \geq k\}$

Ejercicio

Demuestre que GIS es NP-completo

Contando el número de conjuntos independientes

Sea $\#GIS$ una función que, dado un grafo G y un número natural k , retorna el número de conjuntos independientes S de G tales que $|S| \geq k$

Proposición

$\#GIS$ es $\#P$ -completo bajo reducciones parsimoniosas.

Una función de conteo más simple

Sea $\#LIS$ una función que, dado un grafo G y un número natural k , retorna el número de conjuntos independientes S de G tales que $|S| \leq k$

Proposición

$\#LIS$ es $\#P$ -completo.

Ejercicio

Muestre que $L_{\#LIS} \in PTIME$

- Es posible entonces que $\#LIS$ tenga un FPRAS

#LIS es #P-completo: demostración

Dado un grafo G con n nodos, y un número natural $k \geq 1$, se tiene que:

$$\#GIS(G, k) = \#LIS(G, n) - \#LIS(G, k - 1)$$

Concluimos entonces que $\#GIS \in FP^{\#LIS}$, vale decir, $\#GIS \leq_T^P \#LIS$

- ▶ Tenemos que $\#LIS$ es #P-hard
- ▶ Dado que $\#LIS \in \#P$, tenemos entonces que $\#LIS$ es #P-completo



¿Existe un FPRAS para $\#LIS$?

Vamos a dar una respuesta negativa a esta pregunta basada en una suposición de complejidad.

- ▶ Este resultado nos va a ayudar a identificar otros problemas que no admiten FPRAS

Teorema

Si existe un FPRAS para $\#LIS$, entonces $NP \subseteq BPP$

#LIS no admite un FPRAS: demostración

Vamos a demostrar que si existe un FPRAS para #LIS, entonces $\text{GIS} \in \text{BPP}$

- ▶ Dado que GIS es NP-completo, se concluye que $\text{NP} \subseteq \text{BPP}$

Sea $G = (N, A)$ un grafo con $|N| = n$, y sea $k \geq 1$ un número natural.

- ▶ (G, k) es una entrada de GIS

Consideramos un número natural r , cuyo valor definiremos más tarde, y para cada $v \in N$ definimos un nuevo conjunto de r nodos:

$$C_v = \{v_1, v_2, \dots, v_r\}$$

Técnica de demostración: construyendo un grafo aumentado

Usamos r para definir un nuevo grafo $G^r = (N^r, A^r)$:

$$N^r = \bigcup_{v \in N} C_v$$
$$A^r = \bigcup_{(u,v) \in A} \{(a, b) \mid a \in C_u \text{ y } b \in C_v\}$$

G^r es un grafo con $r \cdot n$ nodos, donde dos nodos son adyacentes si y sólo si sus nodos de origen en G son adyacentes.

La noción de testigo

Definición

Dados conjuntos independientes S de G y T de G^r , decimos que T es un testigo para S si:

$$S = \{v \in N \mid C_v \cap T \neq \emptyset\}$$

Un conjunto independiente de G^r es testigo de un único conjunto independiente de G

- ▶ Por otra parte, un conjunto independiente de G puede tener muchos testigos en G^r

La noción de testigo

La conexión fundamental entre los grafos:

$$(G, k) \in \text{GIS} \text{ si y sólo si } (G^r, k \cdot r) \in \text{GIS}$$

Utilizando esta conexión y eligiendo un valor adecuado para r vamos a construir un algoritmo aleatorizado de tiempo polinomial para GIS

- Teniendo como hipótesis la existencia de un FPRAS para $\#\text{LIS}$

El número de testigos

Definimos los conjuntos:

$$\begin{aligned} I^r(G, k) &= \{S \mid S \text{ es conjunto independiente de } G^r \text{ con } |S| = k \cdot r\} \\ M^r(G, k) &= \{S \mid S \text{ es conjunto independiente de } G^r \text{ con } |S| < k \cdot r\} \end{aligned}$$

Tenemos que $(G, k) \in \text{GIS}$ si y sólo si $(G^r, k \cdot r) \in \text{GIS}$ si y sólo si $I^r(G, k) \neq \emptyset$

- Suponiendo la existencia de un FPRAS para $\#\text{LIS}$, vamos a desarrollar un algoritmo aleatorizado de tiempo polinomial para verificar si $I^r(G, k) \neq \emptyset$

El número de testigos

Para $\ell \in \{0, 1, \dots, k\}$ definimos t_ℓ como la cantidad de testigos en G^r para un conjunto independiente de G con ℓ nodos.

Por la definición de G^r tenemos que $t_\ell = (2^r - 1)^\ell$

► ¿Por qué?

Además, G tiene a lo más $\binom{n}{\ell}$ conjuntos independientes con ℓ nodos

El número de testigos

A partir de lo anterior concluimos que:

$$\begin{aligned} |M^r(G, k)| &\leq \sum_{\ell=0}^{k-1} \binom{n}{\ell} t_{\ell} \\ &\leq \sum_{\ell=0}^{k-1} \binom{n}{\ell} t_{k-1} \\ &\leq t_{k-1} \sum_{\ell=0}^n \binom{n}{\ell} \\ &= 2^n \cdot t_{k-1} \\ &= 2^n \cdot (2^r - 1)^{k-1} \end{aligned}$$

Técnica de demostración: aumentando las diferencias

Por otra parte, si $(G, k) \in \text{GIS}$ tenemos que:

$$|I^r(G, k)| \geq t_k = (2^r - 1)^k$$

Concluimos que:

- ▶ Si $(G, k) \notin \text{GIS}$: $\# \text{LIS}(G^r, k \cdot r) = |M^r(G, k)| \leq 2^n(2^r - 1)^{k-1}$
- ▶ Si $(G, k) \in \text{GIS}$: $\# \text{LIS}(G^r, k \cdot r) \geq |I^r(G, k)| \geq (2^r - 1)^k$

En particular, eligiendo $r = n + 3$:

- ▶ Si $(G, k) \notin \text{GIS}$: $\# \text{LIS}(G^{n+3}, k \cdot (n + 3)) \leq 2^n(2^{n+3} - 1)^{k-1}$
- ▶ Si $(G, k) \in \text{GIS}$: $\# \text{LIS}(G^{n+3}, k \cdot (n + 3)) \geq (2^{n+3} - 1)^k$

Un algoritmo aleatorizado para GIS

Suponemos que existe un FPRAS $\mathcal{A}$ para $\#GIS$, y mostramos como esto puede ser utilizado para obtener que $GIS \in BPP$

Definimos el siguiente algoritmo aleatorizado $\mathcal{B}$ para GIS:

1. Dada la entrada (G, k) , donde G es un grafo con n nodos, si $k = 0$ retorne **sí**, en otro caso vaya al paso 2
2. Genere el grafo G^{n+3}
3. Sea s el resultado de ejecutar $\mathcal{A}(G^{n+3}, k \cdot (n + 3), \frac{1}{2})$
4. Si $s > (1 + \frac{1}{2}) \cdot 2^n (2^{n+3} - 1)^{k-1}$, entonces retorne **sí**, en caso contrario retorne **no**

Nótese que $\mathcal{B}$ funciona en tiempo polinomial en el tamaño de la entrada (G, k)

- Dado que el tamaño de G^{n+3} es polinomial en el tamaño de G (G^{n+3} tiene $n \cdot (n + 3)$ nodos), $\mathcal{A}$ es un FPRAS y $\varepsilon = \frac{1}{2}$ cuando se ejecuta $\mathcal{A}$

La probabilidad de error del algoritmo

Si $(G, k) \notin \text{GIS}$ obtenemos que $\Pr(\mathcal{B}(G, k) \text{ retorne sí})$ es igual a:

$$\begin{aligned} & \Pr(\mathcal{A}(G^{n+3}, k \cdot (n+3), \frac{1}{2}) > (1 + \frac{1}{2}) \cdot 2^n (2^{n+3} - 1)^{k-1}) \\ & \leq \Pr(\mathcal{A}(G^{n+3}, k \cdot (n+3), \frac{1}{2}) > (1 + \frac{1}{2}) \cdot \#\text{LIS}(G^{n+3}, k \cdot (n+3))) \\ & \leq \Pr(\mathcal{A}(G^{n+3}, k \cdot (n+3), \frac{1}{2}) > (1 + \frac{1}{2}) \cdot \#\text{LIS}(G^{n+3}, k \cdot (n+3)) \vee \\ & \quad \mathcal{A}(G^{n+3}, k \cdot (n+3), \frac{1}{2}) < (1 - \frac{1}{2}) \cdot \#\text{LIS}(G^{n+3}, k \cdot (n+3))) \\ & = 1 - \Pr(\mathcal{A}(G^{n+3}, k \cdot (n+3), \frac{1}{2}) \leq (1 + \frac{1}{2}) \cdot \#\text{LIS}(G^{n+3}, k \cdot (n+3)) \wedge \\ & \quad \mathcal{A}(G^{n+3}, k \cdot (n+3), \frac{1}{2}) \geq (1 - \frac{1}{2}) \cdot \#\text{LIS}(G^{n+3}, k \cdot (n+3))) \\ & = 1 - \Pr(|\mathcal{A}(G^{n+3}, k \cdot (n+3), \frac{1}{2}) - \#\text{LIS}(G^{n+3}, k \cdot (n+3))| \leq \\ & \quad \frac{1}{2} \cdot \#\text{LIS}(G^{n+3}, k \cdot (n+3))) \\ & \leq 1 - \frac{3}{4} = \frac{1}{4} \end{aligned}$$

La probabilidad de error del algoritmo

Consideramos ahora el caso en que $(G, k) \in \text{GIS}$

Primero debemos demostrar que:

$$(1 - \frac{1}{2}) \cdot \#\text{LIS}(G^{n+3}, k \cdot (n+3)) > (1 + \frac{1}{2}) \cdot 2^n (2^{n+3} - 1)^{k-1}$$

Dado que $\#\text{LIS}(G^{n+3}, k \cdot (n+3)) \geq (2^{n+3} - 1)^k$, basta demostrar que:

$$(1 - \frac{1}{2}) \cdot (2^{n+3} - 1)^k > (1 + \frac{1}{2}) \cdot 2^n (2^{n+3} - 1)^{k-1}$$

La probabilidad de error del algoritmo

Tenemos que:

$$\begin{aligned} & \left(1 - \frac{1}{2}\right) \cdot (2^{n+3} - 1)^k > \left(1 + \frac{1}{2}\right) \cdot 2^n (2^{n+3} - 1)^{k-1} \\ \Leftrightarrow & \frac{1}{2} \cdot (2^{n+3} - 1)^k > \frac{3}{2} \cdot 2^n (2^{n+3} - 1)^{k-1} \\ \Leftrightarrow & \frac{1}{2} \cdot (2^{n+3} - 1) > \frac{3}{2} \cdot 2^n \\ \Leftrightarrow & 2^{n+3} - 1 > 3 \cdot 2^n \\ \Leftrightarrow & 2^n (2^3 - 3) - 1 > 0 \\ \Leftrightarrow & 5 \cdot 2^n - 1 > 0 \end{aligned}$$

Esta última condición es cierta para todo $n \geq 0$, de lo cual concluimos que $\left(1 - \frac{1}{2}\right) \cdot \# \text{LIS}(G^{n+3}, k \cdot (n+3)) > \left(1 + \frac{1}{2}\right) \cdot 2^n (2^{n+3} - 1)^{k-1}$

La probabilidad de error del algoritmo

Si $(G, k) \in \text{GIS}$ obtenemos que $\Pr(\mathcal{B}(G, k) \text{ retorne no})$ es igual a:

$$\begin{aligned} \Pr(\mathcal{A}(G^{n+3}, k \cdot (n+3), \frac{1}{2}) \leq (1 + \frac{1}{2}) \cdot 2^n (2^{n+3} - 1)^{k-1}) \\ \leq \Pr(\mathcal{A}(G^{n+3}, k \cdot (n+3), \frac{1}{2}) < (1 - \frac{1}{2}) \cdot \# \text{LIS}(G^{n+3}, k \cdot (n+3))) \\ \leq \Pr(\mathcal{A}(G^{n+3}, k \cdot (n+3), \frac{1}{2}) < (1 - \frac{1}{2}) \cdot \# \text{LIS}(G^{n+3}, k \cdot (n+3)) \vee \\ \mathcal{A}(G^{n+3}, k \cdot (n+3), \frac{1}{2}) > (1 + \frac{1}{2}) \cdot \# \text{LIS}(G^{n+3}, k \cdot (n+3))) \\ = 1 - \Pr(\mathcal{A}(G^{n+3}, k \cdot (n+3), \frac{1}{2}) \geq (1 - \frac{1}{2}) \cdot \# \text{LIS}(G^{n+3}, k \cdot (n+3)) \wedge \\ \mathcal{A}(G^{n+3}, k \cdot (n+3), \frac{1}{2}) \leq (1 + \frac{1}{2}) \cdot \# \text{LIS}(G^{n+3}, k \cdot (n+3))) \\ = 1 - \Pr(|\mathcal{A}(G^{n+3}, k \cdot (n+3), \frac{1}{2}) - \# \text{LIS}(G^{n+3}, k \cdot (n+3))| \leq \\ \frac{1}{2} \cdot \# \text{LIS}(G^{n+3}, k \cdot (n+3))) \\ \leq 1 - \frac{3}{4} = \frac{1}{4} \end{aligned}$$

#LIS no admite un FPRAS: conclusión

Tenemos que en ambos casos el error del algoritmo $\mathcal{B}$ está acotado superiormente por $\frac{1}{4}$

Concluimos entonces que $\text{GIS} \in \text{BPP}$

► Por lo tanto $\text{NP} \subseteq \text{BPP}$



Utilizando la técnica anterior en otros problemas

Sea $\#IS$ una función que, dado un grafo G , retorna el número de conjuntos independientes de G

Teorema

Si existe un FPRAS para $\#IS$, entonces $NP \subseteq BPP$

Ejercicio

Demuestre el teorema utilizando la técnica usada para el caso de $\#LIS$