Wnioskowania aproksymacyjne

w. 1, 19.02.13

w złożoności obliczeniowej

Problemy NP-trudne i ich aproksymacja

Christos Papadimitriou, Complexity, Addison Wesley 1994 [tłumaczenie polskie: Złożoność obliczeniowa, WNT, 2002, rozdział 13]

> Materiał do tej części wykładu jest w pliku Papadimitriou.pdf oraz pliku NPHardApprox.pdf (str. 435-438).

we wnioskowaniach indukcyjnych

Entropia

> Materiał tej części wykładu jest w pliku TeoriaInformacji.pdf

VC-dim

statystyczna teoria uczenia

> Materiał do tej części wykładu jest w pliku LearningTheory1.pdf oraz w pliku PrzykładyVC.pdf.

logiczne aspekty

- SAT
- modelowanie pojęć nieostrych
- wnioskowania niemonotoniczne
- logika [illegible]
- ...
- information flow

> materiał do tej części wykładu jest w plikach part1.pdf part4.pdf

Do każdej części wykładu w plikach pdf znajdują się też zeskanowane notatki.

Problem

Decyzyjny

- Zbiór przykładów (instancji, wystąpień)
- własność W

<u>Problem</u> (ustalony zbiór przykładów)

We: x - przykład

Wy: 1 jeśli x ma własność W

0 w p.p.

Przykłady

- SAT
- Cykl Hamiltona
- ...

Optymalizacyjny

- Zbiór przykładów
- Dla każdego przykładu x określony jest zbiór rozwiązań $F(x)$
- Funkcja kosztu c określona na rozwiązaniach z $F(x)$ [wartości c - nieujemne]

<u>Problem</u> (ustalone: zb. przykładów, F, c)

We: x - przykład

Wy: $s_0 \in F(x)$ o własności

$$c(s_0) = OPT(x) = \max_{s \in F(x)} c(s) \quad (\min)$$

<u>Przykłady</u>

- Problem komiwojażera (TSP)
- Problem plecakowy (KP)

TSP

We: $\begin{cases} G = (V, E) \\ d : E \to N_+ \end{cases}$ $x = (G, d)$

V – skończony zbiór wierzchołków np. $V = \{1, .., n\}$

$E \subseteq V \times V$,

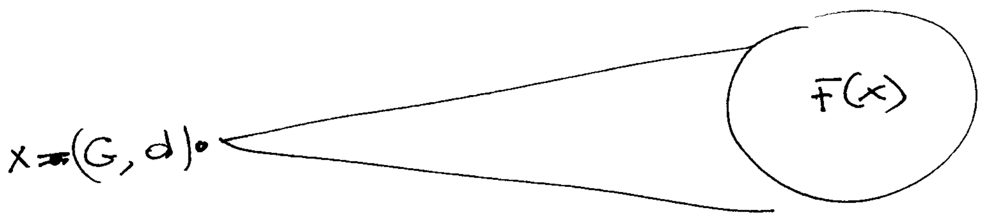
$d(i,j)$ – odległość 'miast' i, j $\quad (d(i,j) = d(j,i))$

π permutacja $\{1, .., n\}$ $\quad [\pi(n+1) = \pi(1)]$

$F(G, d)$ – zbiór permutacji $\{1, .., n\}$

$$C(\pi) = \sum_{i=1}^{n} d(\pi(i), \pi(i+1))$$

π_0 (π_{opt}) : $OPT(G, d) = C(\pi_{opt}) = \min_{\pi} C(\pi)$



$F(x)$ ma $n!$ elementów

!

KP

We : n – l. naturalna

$w_1, \ldots, w_n \in N$ (ciężary przedmiotów)

$v_1, \ldots, v_n \in N$ (wartości przedmiotów)

W – ograniczenie na ciężar

Wy : $S_{OPT} \subseteq \{1, \ldots, n\}$: $\sum_{i \in S_{OPT}} v_i = \max_Q \left\{ \sum_{i \in Q} v_i : \sum_{i \in Q} w_i \leq W \right\}$

$x = (n, w_1, \ldots, w_n, W, v_1, \ldots, v_n)$

$F(x) = \{ Q \subseteq \{1, \ldots, n\} : \sum_{i \in Q} w_i \leq W \}$

$C(Q) = \sum_{i \in Q} v_i$; $OPT(x) = \max_{Q \in F(x)} C(Q)$

Definicja algorytmu ε – aproksymacyjnego dla problemu optymalizacyjnego A

$\varepsilon \in (0,1)$

$x \longrightarrow F(x)$

$c \qquad c(s) \geq 0$

$$OPT(x) = \underset{s \in F(x)}{\min\ (\max)}\ c(s)$$

M – algorytm przekształcający x na $M(x) \in F(x)$

→ deterministyczny, wielomianowy

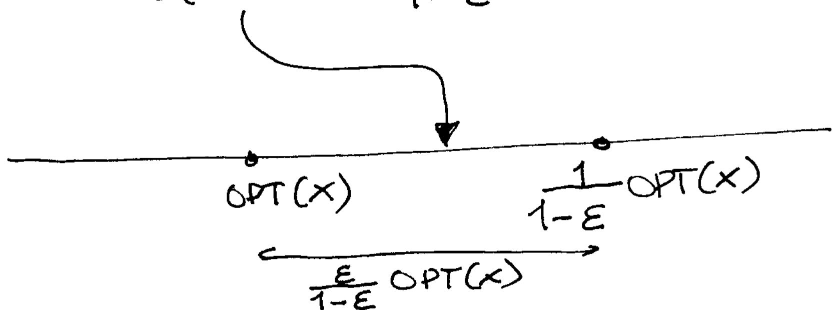
→ dla każdego przykładu x

$$\frac{|c(M(x)) - OPT(x)|}{\max\{OPT(x), c(M(x))\}} \leq \varepsilon$$

min:

$$\frac{c(M(x)) - OPT(x)}{c(M(x))} \leq \varepsilon$$

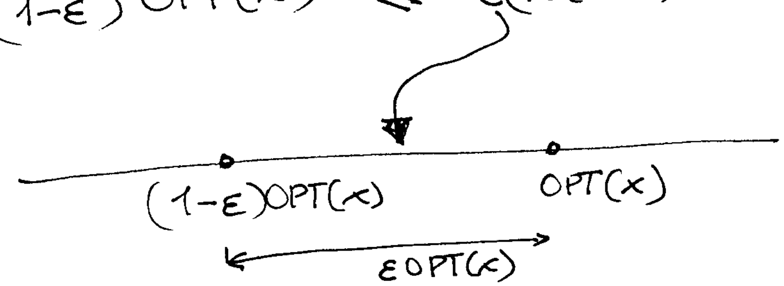
$$c(M(x)) \leq \frac{1}{1-\varepsilon} OPT(x)$$



max:

$$\frac{OPT(x) - c(M(x))}{OPT(x)} \leq \varepsilon$$

$$(1-\varepsilon)\, OPT(x) \leq c(M(x))$$



<u>Definicja</u> progu aproksymacji dla problemu optymalizacyjnego A :

kres dolny wszystkich $\varepsilon > 0$ takich, że istnieje (wielomianowy w czasie) algorytm ε-aproksymacyjny dla A

Dla problemów minimalizacji A: (maksymalizacji)

jeśli próg jest 0 to możliwa jest aproksymacja z dowolną dokładnością

jeśli próg jest 1 to problem nie jest aproksymowalny tzn. nie istnieje $0<\varepsilon<1$ takie, że dla tego ε istnieje algorytm M ε-aproksymacyjny dla A

<u>Uwaga</u> Redukcja jednych problemów do innych nie zachowuje optymalności!

Twierdzenie Próg aproksymacji dla TSP jest 1, chyba, że P = NP.

Dowód Przypuśćmy, że istnieje $\varepsilon < 1$ oraz algorytm ε-aproksymacyjny M dla TSP.
z założenia wielomianowy w czasie

Pokażemy, że istnieją wtedy wielomianowy w czasie algorytm dla problemu NP-zupełnego CYKL HAMILTONA (HC)

$$G = (V, E) \longrightarrow x = (G, d), \quad G = (V, E)$$

przykład dla HC $\qquad$ przykład dla TSP

$$d(i,j) = \begin{cases} 1 & \text{jeśli } (i,j) \in E \\ \dfrac{|V|}{1-\varepsilon} & \text{jeśli } (i,j) \notin E \end{cases}$$

Zastosujmy teraz M do tak skonstruowanego przykładu (G, d) dla TSP.

Możliwe są dwa przypadki:

1. M wyznacza drogę o koszcie $|V|$ tzn. drogę przechodzącą przez wszystkie wierzchołki (wykorzystującą tylko krawędzie z G)

   wtedy G ma cykl Hamiltona

2. M wyznacza drogę o koszcie $> |V|$. wtedy ta droga prowadzi przynajmniej przez jedną krawędź $(i,j) \notin E$ a więc koszt całkowity jest $> \dfrac{|V|}{1-\varepsilon}$

$$\left.\begin{aligned} c(M(x)) &> \frac{|V|}{1-\varepsilon} \\ \text{cyli}\quad (1-\varepsilon)\, c(M(x)) &\le OPT(x) \end{aligned}\right\} \longrightarrow OPT(x) > |V|$$

z drugiej strony $\qquad$ a więc G nie ma cyklu Hamiltona.

KP

Twierdzenie Próg aproksymacji dla KP wynosi 0.

$$V = \max\{v_1, \dots, v_n\}$$

Dla $0 \le i \le n$ oraz $0 \le v \le nV$

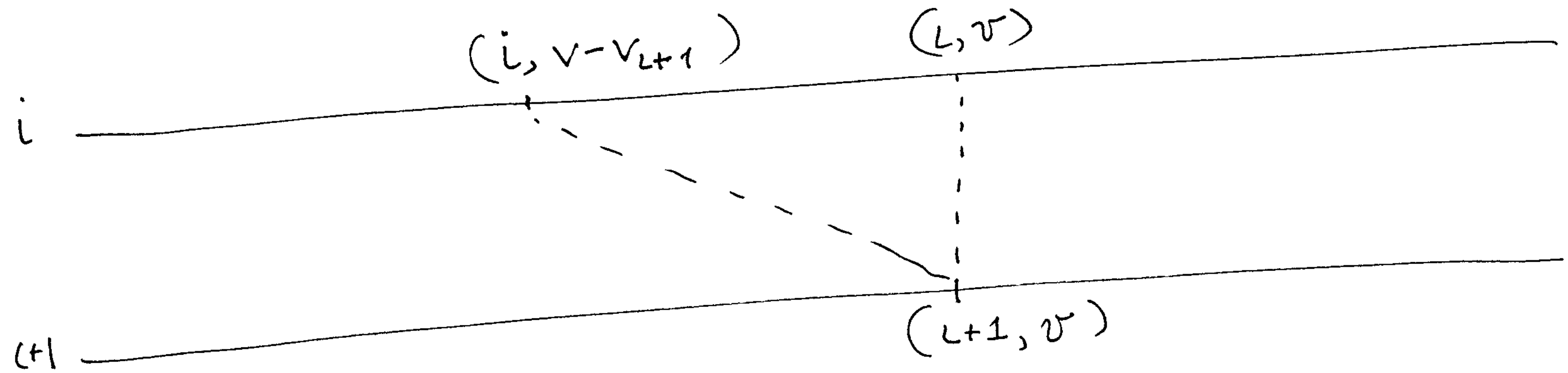
$W(i,v)$ = minimalny ciężar potrzebny do zdobycia wartości v z pewnych z pierwszych i-przedmiotów

$i \backslash v$	0	1	2	. . . . .	nV
0	0	∞	∞		∞
1	0				
$\vdots$					
n	0				

$W(0, v) = \infty$ dla $v > 0$

$W(i, 0) = 0$ $\quad 0 \leq i \leq n$

$$W(i+1, v) = \min\{W(i, v),\ W(i, v - v_{i+1}) + w_{i+1}\}$$



- Na końcu wybieramy największe v takie, że $W(n, v) \leq W$

$$\boxed{O(n^2 V)}$$

$$x = (n, w_1, \dots, w_n, W, v_1, \dots, v_n)$$

kodowanie

1

bin

$$|x|_1 = |\underbrace{1 \dots 1}_{n}\, 0\, \underbrace{1 \dots 1}_{w_1}\, 0\, \underbrace{1 \dots 1}_{w_2}\, 0 \dots 0\, \underbrace{1 \dots 1}_{w_n}\, 0\, \underbrace{1 \dots 1}_{W}\, 0\, \underbrace{1 \dots 1}_{v_1}\, 0 \dots 0\, \underbrace{1 \dots 1}_{v_n}|$$

$$= O(3n + 1 + nV + (n+1)W)$$

$$O(n^2 V)$$

$$C n^2 V \leq C |x|_1^3$$

bin:

$$\underbrace{\dots}_{\log n} \# \underbrace{\dots}_{\log w_1} \# \dots \# \underbrace{\dots}_{\log w_n} \# \underbrace{\dots}_{\log W} \# \underbrace{\dots}_{\log v_1} \# \dots \# \underbrace{\dots}_{\log v_n} \|$$

$$|x|_{bin} \leq C(\log n + 2n + 1 + n \log V + (n+1) \log W)$$

$$O(n^2 V) = O(\underbrace{n^2 2^{\log_2 V}}_{\text{nie jest wielomianem od } |x|_{bin}})$$

Aproksymacja KP

$$x = (w_1, \dots, w_n, W, v_1, \dots, v_n) \longrightarrow x' = (w_1, \dots, w_n, W, v_1', \dots, v_n')$$

$$v_i' = 2^b \left\lfloor \frac{v_i}{2^b} \right\rfloor$$

S – rozwiązanie optymalne dla x

S' " " dla x'

$$\left(\begin{array}{l} \frac{v_i}{2^b} - 1 < \left\lfloor \frac{v_i}{2^b} \right\rfloor \leq \frac{v_i}{2^b} \\ v_i - 2^b < 2^b \left\lfloor \frac{v_i}{2^b} \right\rfloor \leq v_i \end{array} \right)$$

$$\sum_{i \in S} v_i \underset{\substack{\text{bo } S \\ \text{optymalne} \\ \text{dla } x}}{\geq} \sum_{i \in S'} v_i \underset{v_i \geq v_i'}{\geq} \sum_{i \in S'} v_i' \underset{\substack{\text{bo } S' \\ \text{opt. dla} \\ x'}}{\geq} \sum_{i \in S} v_i' \geq \sum_{i \in S} (v_i - 2^b) \geq \sum_{i \in S} v_i - n 2^b$$

V – ograniczenie dolne na wartość rozwiązania optymalnego (można tak założyć, inaczej można odrzucić v_i, w_i dla których $w_i > W$ i wtedy V jest osiągalne)

M dostarcza rozwiązanie optymalne dla x' ze złożonością czasową $O(n^2 \frac{V}{2^b})$

$\geq OPT(x) - 2^b \cdot n$

$$\frac{|OPT(x) - c(M(x))|}{\max\{OPT(x), c(M(x))\}} = \frac{OPT(x) - c(M(x))}{OPT(x)} \leq \frac{n \cdot 2^b}{V}$$

aby zapewnić $\frac{n \cdot 2^b}{V} \leq \varepsilon$

Dla $\varepsilon > 0$ wybieramy $b = \left\lceil \log \frac{\varepsilon V}{n} \right\rceil$ bitów i wyznaczamy rozwiązanie aproksymacyjne w czasie

$$O\left(\frac{n^2 V}{2^b}\right) = O\left(\frac{n^2 V}{\frac{\varepsilon V}{n}}\right) = O\left(\frac{n^3}{\varepsilon}\right)$$

<u>Uwagi</u>

1. Możemy założyć $\frac{\varepsilon \bar{V}}{n} > 1$ bo inaczej $\bar{V} \leqslant \frac{n}{\varepsilon}$ i wtedy $O(n^2 \bar{V}) = O(n^3/\varepsilon)$ bez aproksymacji

2. $\log \frac{\varepsilon \bar{V}}{n} \leqslant \left\lceil \log \frac{\varepsilon \bar{V}}{n} \right\rceil < \log \frac{\varepsilon \bar{V}}{n} + 1$

$$\frac{n^2 \bar{V}}{2^b} \leqslant \frac{n^2 \bar{V}}{\frac{\varepsilon \bar{V}}{n}} = \frac{n^3}{\varepsilon}$$

Problem MAX-3SAT

We: φ - formuła rachunku zdań w postaci 3-CNF

Wy: $$\text{MAX-3SAT}(\varphi) = \max_{v} \text{MAX-3SAT}(\varphi, v)$$

gdzie $$\text{MAX-3SAT}(\varphi, v) = \frac{\text{liczba klauzul (czynników) } \varphi \text{ spełnionych przez } v}{\text{liczba klauzul (czynników) w } \varphi}$$

v - wartościowanie zmiennych występujących w φ w $\{0,1\}$.

Uwaga: Żądamy również podania v_{opt} takiego, że $\text{MAX-3SAT}(\varphi, v_{opt}) = \text{MAX-3SAT}(\varphi)$.

Schemat aproksymacji

Wielomianowy schemat aproksymacji dla problemu optymalizacyjnego A zwraca to algorytm, który dla każdego $\varepsilon > 0$ i przykładu x problemu A zwraca rozwiązanie o błędzie względnym $\leq \varepsilon$ w czasie ograniczonym przez wielomian od długości x oraz zależny od ε (np. $O(\frac{n^3}{\varepsilon})$ dla problemu KP).

Problem 3-SAT

$(p \vee \bar{q} \vee r) \wedge \ldots\ldots \wedge (\quad\quad)$

każda klauzula ma 3 (zmienne bądź ich negacje) literały

We: α – formuła r. zdań w postaci 3-~~SAT~~CNF

Wy: 1 α jest spełnialna [tzn. istnieje wartościowanie v zmiennych α takie, że $v(\alpha)=1$]

0 w pp.

Problem MAX-3-SAT

We: α – formuła r. zdań w postaci 3-CNF

Wy: wartościowanie v przy którym iloraz

$$\frac{\text{liczba klauzul } \alpha \text{ spełnionych przez } v}{\text{liczba wszystkich klauzul } \alpha}$$

osiąga maksimum

MAX-3SAT(α) = OPT(α) oznacza maksymalną wartość tego ilorazu.

Problem: Czy próg aproksymacji MAX-3-SAT jest 0 ?

Twierdzenie. Jeśli istnieje schemat aproksymacyjny (wielomianowy) dla MAX-3SAT to $P = NP$.

Dowód tego twierdzenia okazał się możliwy dzięki nowej charakteryzacji klasy NP za pomocą tzw. weryfikatorów probabilistycznych.

Weryfikatory :

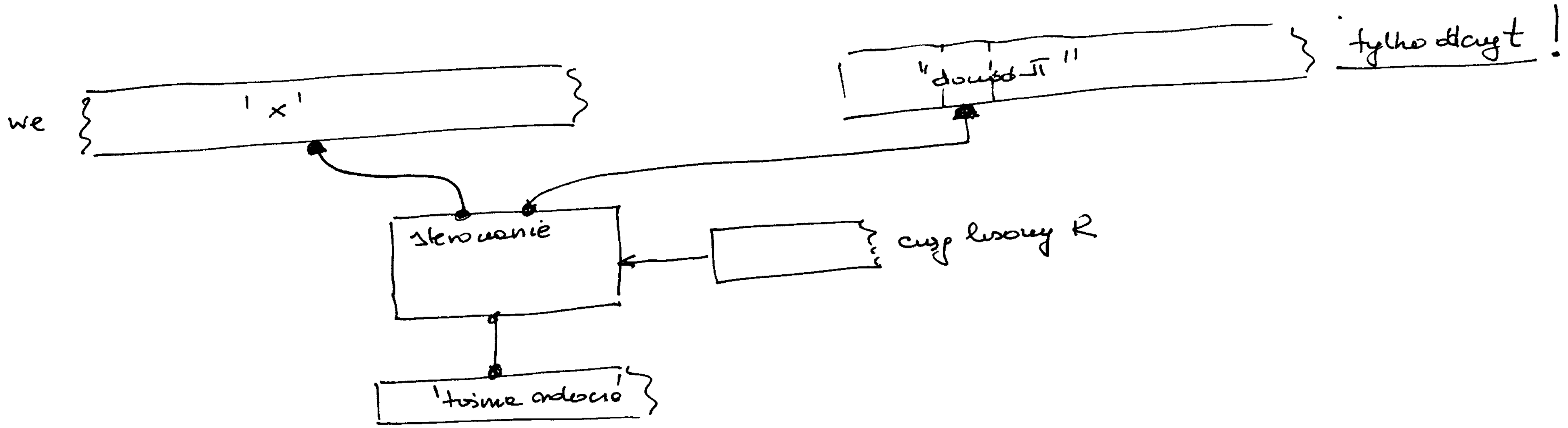
NDTM probabilistyczne, wielomianowe w czasie

M

- taśma we
- " robocza
- -"- bitów losowych
- -"- tylko do odczytu : Π (taśma dowodu)

zawartość taśmy

maszyna ma dostęp do Π poprzez adresy miejsc, w których π jest zapisane i musi odczytywać bit, który jest zawartością wskazanego adresu



$$M^{\pi}(x, R) = \begin{cases} 1 & M \text{ akceptuje } x \text{ mając dostęp do } \pi \text{ i ciągu losowego } R \\ 0 & \text{w p.p.} \end{cases}$$

Df. Weryfikator M rozpoznaje język L, jeśli

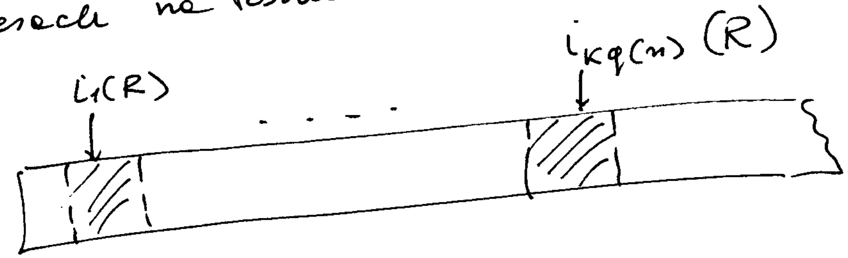
- dla każdego $x \in L$ istnieje π taki, że M akceptuje x przy każdym ciągu losowym R
- dla każdego $x \notin L$ dla jakiegokolwiek dowodu π : π jest "odrzucany" z prawdopodobieństwem $\geq \frac{1}{2}$ tzn.

$$Pr_R \{ M^{\pi}(x, R) = 1 \} \geq \frac{1}{2}$$

Weryfikator M jest $(r(n), q(n))$-ograniczony jeśli na każdym wejściu o rozmiarze n weryfikator używa co najwyżej $O(r(n))$ bitów losowych w realizowanym obliczeniu i pyta o zawartość co najwyżej $O(q(n))$ miejsc (adresów) na taśmie dowodów tzn. istnieją stałe c, k takie, że weryfikator czyta z taśmy losowej $c\,r(n)$ bitów; ~~Na~~ na wejściu o długości n czyta ciąg losowy R i oblicza $kq(n)$ adresów!

$$i_1(R) \quad \dots \quad i_{kq(n)}(R)$$

oraz pyta o zawartość miejsc o tych adresach na taśmie dowodu



Dalej w zależności od tych zawartości kontynuuje (niekoniecznie w czasie działania) i akceptuje lub odrzuca X.

Df. $L \in PCP(r(n), q(n))$ jeśli istnieje weryfikator $(r(n), q(n))$-ograniczony rozpoznający L.

Fakt. $NP = PCP(0, poly(n))$

TW. $PCP(\log n, 1) = NP$

probabilistically checkable proofs

Łatwa część dowodu $PCP(\log n, 1) \subseteq NP$

Dowód Należy pokazać, że jeśli $L \in PCP(\log n, 1)$ to $L \in NP$

NDTM zgaduje Π dla x o dł. n i sprawdza czy przy każdym ciągu losowym R o długości $c \log n$ weryfikator wspierający L akceptuje x. Różnych R jest $2^{c \log n} = n^c$ – wielomianowa względem n liczba.

(dowód Arora)

Trudna część dowodu: $NP \subseteq PCP(\log n, 1)$

Tw. (D. S. Hochbaum: Approximation algorithms for NP-hard problems PWS Publishing Company 1997, Boston)

Następujące warunki są równoważne!

~~Istnieje ε > 0 oraz~~

(I) $NP = PCP(\log n, 1)$

(II) Istnieje $\varepsilon > 0$ oraz wielomianowa w czasie redukcja τ z SAT do MAX-3SAT taka, że dla każdej formuły φ

a) jeśli $\varphi \in SAT$ to $\text{MAX-3SAT}(\tau(\varphi)) = 1$

b) jeśli $\varphi \notin SAT$ to $\text{MAX-3SAT}(\tau(\varphi)) < \frac{1}{1+\varepsilon}$

Wobec tego osiągnięcie ~~aproksymacji~~ współczynnika aproksymacji $\frac{\varepsilon}{1+\varepsilon}$ jest NP-trudne

W związku z tym, że $NP = PCP(\log n, 1)$ otrzymujemy następujące twierdzenie:

<u>Twierdzenie</u> Istnieje $\varepsilon > 0$ oraz wielomianowa w czasie redukcja τ z SAT do MAX-3SAT taka, że dla każdej formuły rachunku zdań φ

a) jeśli $\varphi \in SAT$ to $\text{MAX-3SAT}(\tau(\varphi)) = 1$

b) jeśli $\varphi \notin SAT$ to $\text{MAX-3SAT}(\tau(\varphi)) < \frac{1}{1+\varepsilon}$.

Twierdzenie to stwierdza w istocie, że osiągnięcie współczynnika aproksymacji $\frac{\varepsilon}{1+\varepsilon}$ jest NP-trudne tzn. jeśli istnieje wielomianowy algorytm $\mathcal{A}$ ~~dla~~ aproksymacyjny dla MAX-3SAT ze współczynnikiem $\frac{\varepsilon}{1+\varepsilon}$ to P = NP.

Przypuśćmy, że taki algorytm $\mathcal{A}$ istnieje. Niech φ – formuła r.z. (przykład dla SAT). Przekształcamy φ za pomocą redukcji τ i otrzymujemy $\tau(\varphi)$ – przykład dla MAX-3SAT.

Mamy:

$$\underbrace{\left(1-\frac{\varepsilon}{1+\varepsilon}\right)}_{\frac{1}{1+\varepsilon}} \text{MAX-3SAT}(\varphi) \leq c(\mathcal{A}(\tau(\varphi))) \leq \text{MAX-3SAT}(\tau(\varphi)) \qquad (*)$$

gdzie c – funkcja kosztu wartościowania dostarczonego przez $\mathcal{A}$

Sprawdzamy czy $c(\mathcal{A}(\tau(\varphi))) \geq \frac{1}{1+\varepsilon}$?

T: $\varphi \in SAT$

(w przeciwnym przypadku $\varphi \notin SAT$ i wtedy $c(\mathcal{A}(\tau(\varphi))) \leq \text{MAX-3SAT}(\tau(\varphi)) < \frac{1}{1+\varepsilon}$ a więc sprzeczność z założeniem.)

N: $\varphi \notin SAT$

(w przeciwnym przypadku $\varphi \in SAT$ a więc z $(*)$

$\frac{1}{1+\varepsilon} \cdot 1 \leq c(\mathcal{A}(\tau(\varphi)))$

a więc sprzeczność z założeniem)

Twierdzenie. Następujące warunki są równoważne:

① $NP \subseteq PCP(\log n, 1)$

② Istnieje wielomianowa (w czasie) redukcja τ z SAT do MAX-3SAT taka, że dla pewnego $\varepsilon > 0$ oraz dla każdej formuły rachunku zdań φ

(i) $\varphi \in SAT \longrightarrow MAX\text{-}3SAT(\tau(\varphi)) = 1$

(ii) $\varphi \notin SAT \longrightarrow MAX\text{-}3SAT(\tau(\varphi)) < \frac{1}{1+\varepsilon}$

Dowód. ② → ①

Niech $L \in NP$ i niech x będzie słowem wejściowym. Redukujemy L do SAT za pomocą pewnego obliczalnego w czasie wielomianowym od $|x|$ odwzorowania redukującego f a następnie za pomocą τ do MAX-3SAT.

Jak działa weryfikator?

- Spodziewamy się ~~wartościowania~~ 'dowodu' należenia zmiennych wartościowania dla przykładu MAX-3SAT opisanego wyżej $\varphi = \tau(f(x))$.
- Aby sprawdzić ten dowód wybierana jest losowo klauzula z $\varphi = \tau(f(x))$. Odczytywane są z losowymi dowodu wartości zmiennych z tej klauzuli (potrzeba tylko 3 bitów!) i następuje akceptacja jeśli wartości te spełniają klauzulę.

a. Jeśli $x \in L$ to oczywiście istnieje wartościowanie (dowód) który jest akceptowany z prawdopodobieństwem 1 (tzn. dla każdej wylosowanej klauzuli φ).

b. Jeśli $x \notin L$ to każdy (dowód) wartościowanie jest odrzucone z prawdopodobieństwem $\geq 1 - \frac{1}{1+\varepsilon}$

Powtarzając sprawdzenie $O(\frac{1}{\varepsilon})$ razy (zwiększamy dł. ciągu losowego $O(\frac{1}{\varepsilon})$ razy) prawdopodobieństwo to $\geq \frac{1}{2}$.

koniunkcja n^c funkcji bool.

$$\varphi \rightsquigarrow \{f_R\}_R \rightsquigarrow \bigwedge_R f_R$$

$\varphi \in SAT \longrightarrow \{f_R\}_R$ spełnialny (n^c funkcji) $\longrightarrow$ spełnialne (*)

$\varphi \notin SAT \longrightarrow$ ułamek niespełnialnych funkcji z $\{f_R\}_R$ przy dowolnym wartościowaniu $\geqslant \frac{1}{2}$ (spełnialnych $< \frac{1}{2}$) $\rightsquigarrow$ ułamek czynników (klauzul) niespełnialnych $\geqslant \dfrac{\frac{1}{2}n^c}{n^c}$ (**)

$$\bigwedge_R f_R \rightsquigarrow \bigwedge_R \bigwedge_{j=1}^{2^k} C_{R,j} \longrightarrow \bigwedge_R \bigwedge_{j=1}^{2^k} \bigwedge_{l=1}^{k-1} C^{(l)}_{R,j}$$

każdą funkcję f_R może być przedstawiona w postaci 2^k koniunkcji klauzul (por. przykład dalej)

(*) $\longrightarrow$ spełnialne

(**) $\longrightarrow$ ułamek czynników niespełnialnych $\geqslant \dfrac{\frac{1}{2}n^c}{2^k n^c} = \dfrac{1}{2^{k+1}}$

każdą klauzulę z k literałami może być przedstawiona w postaci $k-1$ klauzul z 3-ma literałami (por. przykład dalej)

(*) $\longrightarrow$ spełnialna

(**) $\longrightarrow$ ułamek czynników niespełnialnych $\geqslant \dfrac{\frac{1}{2}n^c}{(k-1)2^k n^c}$

Więc ułamek klauzul spełnialnych $< 1 - \dfrac{1}{(k-1)2^{k+1}}$. Wystarczy przyjąć $\dfrac{1}{1+\varepsilon} = 1 - \dfrac{1}{(k-1)2^{k+1}}$.

x_1	x_2	x_3	f	$\bar{f}$
0	0	0	1	0
0	0	1	0	1
0	1	0	1	0
0	1	1	1	0
1	0	0	0	1
1	0	1	0	1
1	1	0	0	1
1	1	1	0	1

Jeśli $f : \{0,1\}^k \to \{0,1\}$ to f może być zapisane w postaci CNF za pomocą formuły z 2^k klauzulami

$$f = \bar{x}_1\bar{x}_2\bar{x}_3 \vee \bar{x}_1 x_2 \bar{x}_3 \vee \bar{x}_1 x_2 x_3$$

$$\bar{f} = \bar{x}_1\bar{x}_2 x_3 \vee x_1\bar{x}_2\bar{x}_3 \vee x_1\bar{x}_2 x_3 \vee x_1 x_2 \bar{x}_3 \vee x_1 x_2 x_3$$

$$f = \bar{\bar{f}} = \overline{\bar{x}_1\bar{x}_2 x_3 \vee x_1\bar{x}_2\bar{x}_3 \vee x_1\bar{x}_2 x_3 \vee x_1 x_2 \bar{x}_3 \vee x_1 x_2 x_3} =$$

$$= (x_1 \vee x_2 \vee \bar{x}_3) \wedge (\bar{x}_1 \vee x_2 \vee x_3) \wedge (\bar{x}_1 \vee x_2 \vee \bar{x}_3) \wedge (\bar{x}_1 \vee \bar{x}_2 \vee x_3) \wedge (\bar{x}_1 \vee \bar{x}_2 \vee \bar{x}_3)$$

$(k>3)$

Lemat. Dla każdej klauzuli $l_1 \vee \dots \vee l_k$ istnieją zmienne zdaniowe $z_1, \dots, z_{k-2}$ oraz formuła α w postaci 3-CNF z $(k-1)$ klauzulami i z literałami ze zbioru $\{l_1, \dots, l_k, z_1, \dots, z_{k-2}, \bar{z}_1, \dots, \bar{z}_{k-2}\}$

taka, że $l_1 \vee \dots \vee l_k$ jest spełniona przez wartościowanie v zmiennych występujących w $l_1 \vee \dots \vee l_k$ wtw α jest spełniona przez pewne rozszerzenie (na zmienne $z_1, \dots, z_{k-2}$) wartościowania v

$$(l_5 \vee l_6 \vee \bar{z}_4)$$

$k=6$

$$(l_1 \vee l_2 \vee z_1) \wedge$$
$$(\bar{z}_1 \vee l_3 \vee z_2) \wedge$$
$$(\bar{z}_2 \vee l_4 \vee z_3) \wedge$$
$$(\bar{z}_3 \vee l_5 \vee z_4) \wedge$$

1. $v(l_1) = \dots = v(l_6) = 0 \longrightarrow \alpha$ nie jest spełniona przy żadnym rozszerzeniu wartościowania v

2. $v(l_i) = 1 \longrightarrow v(l_1 \vee \dots \vee l_6) = 1$

np. $i=3$

wtedy $v^*(\bar{z}_1 \vee l_3 \vee z_2) = 1$ dla każdego rozszerzenia v^* wartościowania v

tzn. wartości $\bar{z}_1, z_2$ nie wpływają na wartość $\bar{z}_1 \vee l_3 \vee z_2$

więc też możemy przyjąć $v^*(z_1) = 1$ a wtedy $v^*(l_1 \vee l_2 \vee z_1) = 1$

$v^*(z_2) = 0$ " $v^*(\bar{z}_2 \vee l_4 \vee z_3) = 1$

niezależnie od tego jaka jest wartość z_3 przy tym wartościowaniu

możemy więc przyjąć $v^*(z_3) = 0 \longrightarrow v^*(\bar{z}_3 \vee l_5 \vee z_4) = 1$

i znowu może przyjąć $v^*(z_4) = 0 \longrightarrow v^*(l_5 \vee l_6 \vee \bar{z}_4) = 1$