

O dolnych i górnych granicach złożoności obliczeniowych kilku problemów NP-zupełnych

Łukasz Bieniasz-Krzywiec

22.10.2009

Plan prezentacji

- 1 Wstęp
- 2 Złożoność obliczeniowa problemu CNF-SAT
- 3 Złożoność obliczeniowa CSP

Czy $P = NP$?

Czy algorytmy rozwiązujące problemy NP-zupełne muszą przeszukiwać przestrzeń wszystkich możliwych rozwiązań?

- L** — pewien problem NP-zupełny
 $O^*(2^n)$ — złożoność brutalnego algorytmu rozwiązującego **P**

*Czy dla problemu **L** istnieje algorytm
o złożoności obliczeniowej $O^*(2^{\delta n})$, gdzie δ — stała, $\delta < 1$?*

- CNF-SAT** — problem spełnialności formuł logicznych
w postaci CNF
- $O^*(2^n)$ — złożoność brutalnego algorytmu dla **CNF-SAT**,
 n – liczba zmiennych w formułach

*Czy istnieje algorytm dla **CNF-SAT**
o złożoności obliczeniowej $O^*(2^{\delta n})$, gdzie δ — stała, $\delta < 1$?*

Wyniki ankiety:



Da się – 10%



Nie da się – 90%

ZŁOŻONOŚĆ OBLICZENIOWA PROBLEMU CNF-SAT

Najszybszy znany algorytm dla **CNF-SAT** działa w czasie:

$$O^*\left(2^{n \cdot \left(1 - \frac{1}{\ln(m/n) + O(\ln \ln m)}\right)}\right)$$

gdzie n – liczba zmiennych, m – liczba klauzul

Pokażemy, że jeśli którą z poniższych hipotez jest prawdziwa, to istnieje szybszy algorytm dla **CNF-SAT**:

Hipoteza

Istnieje $k \geq 3$ i $\epsilon \in (0, k)$, takie że problem k -**DOMINATING SET** można rozwiązać w czasie $O(n^{k-\epsilon})$.

Hipoteza

Istnieje $\epsilon > 0$, taki że problem **2SAT+2CLAUSES** można rozwiązać w czasie $O((m+n)^{2-\epsilon})$.

Problem k -DOMINATING SET:

Wejście: graf nieskierowany o n wierzchołkach i m krawędziach

Wyjście: k -elementowy podzbiór zbioru wierzchołków dominujący wszystkie wierzchołki w grafie.

Definicja

Mówimy, że podzbiór $A \subset V$ dominuje wierzchołek $v \in V$, gdy $v \in A$ lub v jest sąsiadem pewnego wierzchołka z A .

Jak rozwiązać problem k -DOMINATING SET?

Algorytm bezpośredni — sprawdzić wszystkie k -elementowe podzbiory zbioru wierzchołków. Czas działania: $O(n^{k+1})$.

Da się szybciej?

Niech A będzie macierzą sąsiedztwa grafu G i niech $B = \overline{A} \cdot \overline{A}^T$.

Twierdzenie

Zbiór $\{v_i, v_j\}$ jest dominujący w grafie G wtw, gdy $B[i, j] = 0$.

Wniosek

2-DOMINATING SET można rozwiązać w czasie $O(n^\omega)$.

Rozszerzamy pomysł na dowolne k :

- rozpatrujemy wszystkie $k/2$ -elementowe podzbiory zbioru wierzchołków $S_1, S_2, \dots, S_{\binom{n}{k/2}}$

Rozszerzamy pomysł na dowolne k :

- rozpatrujemy wszystkie $k/2$ -elementowe podzbiory zbioru wierzchołków $S_1, S_2, \dots, S_{\binom{n}{k/2}}$
- definiujemy macierz A_k :
 $A_k[i, j] = 0$ wtw, gdy wierzchołek v_j jest dominowany przez S_i

Rozszerzamy pomysł na dowolne k :

- rozpatrujemy wszystkie $k/2$ -elementowe podzbiory zbioru wierzchołków $S_1, S_2, \dots, S_{\binom{n}{k/2}}$
- definiujemy macierz A_k :
 $A_k[i, j] = 0$ wtw, gdy wierzchołek v_j jest dominowany przez S_i
- obliczamy macierz $B_k = A_k \cdot A_k^T$ rozmiaru $\binom{n}{k/2} \times \binom{n}{k/2}$
(można to zrobić w czasie $O(n^{k+o(1)})$).

Twierdzenie

Zbiór $S_i \cup S_j$ jest dominujący w grafie G wtw, gdy $B_k[i, j] = 0$.

Wniosek

***k -DOMINATING SET** można rozwiązać w czasie $O(n^{k+o(1)})$.*

Udowodnimy teraz następujące twierdzenie:

Twierdzenie

Jeśli dla pewnego k problem k -DOMINATING SET można rozwiązać w czasie $O(n^{f(k)})$, to dla CNF-SAT istnieje algorytm o złożoności $O((m + k2^{n/k})^{f(k)})$.

Jeśli weźmiemy $f(k) = k - \epsilon$, to otrzymamy lepszy algorytm dla CNF-SAT.

Problem 2SAT+2CLAUSES:

Wejście: koniunkcja formuły w postaci 2CNF z
dwoma klauzulami o dowolnej długości

Wyjście: spełnialność formuły

Jak rozwiązać problem 2SAT+2CLAUSES?

Twierdzenie

Problem **2SAT+2CLAUSES** można rozwiązać w czasie $O(mn + n^2)$, gdzie n – liczba zmiennych, m – liczba wszystkich klauzul.

Czy istnieje algorytm działający w czasie $O((m + n)^{2-\epsilon})$?

Udowodnimy następujące twierdzenie:

Twierdzenie

Jeśli problem **2SAT+2CLAUSES** można rozwiązać w czasie $O((m+n)^{2-\epsilon})$, to dla **CNF-SAT** istnieje algorytm o złożoności $O(2^{(1-\frac{\epsilon}{2})n})$.

ZŁOŻONOŚĆ OBLICZENIOWA CSP

Problem spełnialności więzów (d, k) -CSP

Wejście:

- Σ — dziedzina zmiennych, $|\Sigma| = d$,
- $\mathcal{C}$ — zbiór więzów postaci

$$C = \{x_1 \neq s_1, x_2 \neq s_2, \dots\}$$

gdzie $|C| \leq k$, x_i — zmienne, $s_i \in \Sigma$ wartości.

Wyjście:

Wartościowanie zmiennych $a : \text{Var}(\mathcal{C}) \mapsto \Sigma$ spełniające więzy.

Przykłady:

- $(2, k)$ -**CSP** to problem k -**SAT**

Przykłady:

- $(2, k)$ -**CSP** to problem k -**SAT**
- $(d, 2)$ -**CSP** to problem d -**GRAPH COLORABILITY**

Jaka jest dolna granica złożoności obliczeniowej (d, k) -**CSP**?

Definicja

$c_{d,k} = \inf\{c : (d, k)\text{-}\mathbf{CSP} \text{ można rozwiązać w czasie } 2^{cn} \text{poly}(n)\}$

Pokażemy, że prawdziwe jest twierdzenie:

Twierdzenie

Dla dowolnych stałych $d' > d$ i k :

$$c_{d',k} \geq \lfloor \log_d(d') \rfloor \cdot c_{d,k}$$

W szczególności przy założeniu następującej hipotezy:

Exponential Time Hypothesis (ETH)

$$c_{2,3} > 0$$

otrzymujemy, że:

$$c_{d,3} \in O(\log d)$$

czyli złożoność obliczeniowa $(d, 3)$ -**CSP** to $\Omega^*(2^{\log(d)n})$.

Bibliografia



Mihai Patrascu, Ryan Williams
“On the possibility of faster SAT algorithms”



Patric Traxler
“The Time Complexity of Constraint Satisfaction”