Złożoność obliczeniowa algorytmów

Aproxymowalność

Kordian A. Smoliński

Wydział Fizyki i Informatyki Stosowanej

2024/2025

Aproxymowalność

Treść wykładu

1. Algorytmy aproxymacyjne
   - Problemy optymalizacyjne
   - Algorytmy ϵ-aproxymacyjne
   - Minimalne pokrycie wierzchołkowe
   - Problem komiwojażera
     - Metryczny problem komiwojażera
2. Aproxymowalność i złożoność
   - **MaxSNP**
   - **MaxSNP**-zupełność
   - Nieaproxymowalność

WYDZIAŁ FIZYKI I INFORMATYKI STOSOWANEJ
Uniwersytet Łódzki

Aproxymowalność

Wiele problemów o dużym znaczeniu praktycznym jest **NP**-zupełnych. Jeżeli $\mathbf{P} \neq \mathbf{NP}$, to żaden z nich nie ma efektywnego rozwiązania.

WYDZIAŁ FIZYKI I INFORMATYKI STOSOWANEJ
Uniwersytet Łódzki

Aproxymowalność

Wiele problemów o dużym znaczeniu praktycznym jest **NP**-zupełnych. Jeżeli $\mathbf{P} \neq \mathbf{NP}$, to żaden z nich nie ma efektywnego rozwiązania.
Możliwe sposoby częściowego poradzenia sobie z problemami **NP**-zupełnymi:

WYDZIAŁ FIZYKI I INFORMATYKI STOSOWANEJ
Uniwersytet Łódzki

Aproxymowalność

Wiele problemów o dużym znaczeniu praktycznym jest **NP**-zupełnych. Jeżeli $\mathbf{P} \neq \mathbf{NP}$, to żaden z nich nie ma efektywnego rozwiązania.
Możliwe sposoby częściowego poradzenia sobie z problemami **NP**-zupełnymi:

- poszukiwanie podproblemów, mających rozwiązania o wielomianowej złożoności czasowej;

WYDZIAŁ FIZYKI I INFORMATYKI STOSOWANEJ
Uniwersytet Łódzki

Aproxymowalność

Wiele problemów o dużym znaczeniu praktycznym jest **NP**-zupełnych. Jeżeli $\mathbf{P} \neq \mathbf{NP}$, to żaden z nich nie ma efektywnego rozwiązania.
Możliwe sposoby częściowego poradzenia sobie z problemami **NP**-zupełnymi:

- poszukiwanie podproblemów, mających rozwiązania o wielomianowej złożoności czasowej;
- konstruowanie algorytmów aproxymacyjnych, nierozwiazujących dokładnie problemu, ale wyznaczających rozwiązania przybliżone.

WYDZIAŁ FIZYKI I INFORMATYKI STOSOWANEJ
Uniwersytet Łódzki

Algorytmy aproxymacyjne

Rozwiązań problemów decyzyjnych nie można przybliżać — wymagają odpowiedzi „tak” albo „nie”.

WYDZIAŁ FIZYKI I INFORMATYKI STOSOWANEJ
Uniwersytet Łódzki

Algorytmy aproxymacyjne

Rozwiązań problemów decyzyjnych nie można przybliżać — wymagają odpowiedzi „tak” albo „nie”.
Możemy przybliżać rozwiązania problemów, dla których rozwiązaniem konkretnego przypadku jest pewien element ze zbioru rozwiązań dopuszczalnych.

WYDZIAŁ FIZYKI I INFORMATYKI STOSOWANEJ
Uniwersytet Łódzki

Algorytmy aproxymacyjne

Rozwiązań problemów decyzyjnych nie można przybliżać — wymagają odpowiedzi „tak” albo „nie”.
Możemy przybliżać rozwiązania problemów, dla których rozwiązaniem konkretnego przypadku jest pewien element ze zbioru rozwiązań dopuszczalnych.
Określając na elementach zbioru rozwiązań dopuszczalnych funkcję celu, możemy żądać, aby znaleziona odpowiedź minimalizowała bądź maxymalizowała wartość tej funkcji po wszystkich rozwiązaniach dopuszczalnych.

WYDZIAŁ FIZYKI I INFORMATYKI STOSOWANEJ
Uniwersytet Łódzki

Algorytmy aproxymacyjne

Rozwiązań problemów decyzyjnych nie można przybliżać — wymagają odpowiedzi „tak” albo „nie”.
Możemy przybliżać rozwiązania problemów, dla których rozwiązaniem konkretnego przypadku jest pewien element ze zbioru rozwiązań dopuszczalnych.
Określając na elementach zbioru rozwiązań dopuszczalnych funkcję celu, możemy żądać, aby znaleziona odpowiedź minimalizowała bądź maxymalizowała wartość tej funkcji po wszystkich rozwiązaniach dopuszczalnych.
Problemy o powyższej charakterystyce nazywamy problemami optymalizacyjnymi.

WYDZIAŁ FIZYKI I INFORMATYKI STOSOWANEJ
Uniwersytet Łódzki

Problemy optymalizacyjne

Definicja

W problemie optymalizacyjnym Π określamy:

Problemy optymalizacyjne

Definicja

W problemie optymalizacyjnym Π określamy:

zbiór przypadków D; $\forall_{x \in D}$ określamy

Problemy optymalizacyjne

Definicja

W problemie optymalizacyjnym Π określamy:

zbiór przypadków D; $\forall_{x \in D}$ określamy

zbiór rozwiązań dopuszczalnych $F(x)$; $\forall_{s \in F(x)}$ określamy

WYDZIAŁ FIZYKI i INFORMATYKI STOSOWANEJ
Uniwersytet Łódzki

Problemy optymalizacyjne

Definicja

W problemie optymalizacyjnym Π określamy:

zbiór przypadków D; $\forall_{x \in D}$ określamy

zbiór rozwiązań dopuszczalnych $F(x)$; $\forall_{s \in F(x)}$ określamy

koszt $c(s) \in \mathbb{N}$.

WYDZIAŁ FIZYKI I INFORMATYKI STOSOWANEJ
Uniwersytet Łódzki

Problemy optymalizacyjne

Definicja

W problemie optymalizacyjnym Π określamy:

zbiór przypadków D; $\forall_{x \in D}$ określamy

zbiór rozwiązań dopuszczalnych $F(x)$; $\forall_{s \in F(x)}$ określamy

koszt $c(s) \in \mathbb{N}$.

Definicja

Rozwiązaniem optymalnym minimalizacji (maxymalizacji) dla przypadku $x \in D$ problemu Π jest $s \in F(x)$ takie, że

$$c(s) = \mathrm{opt}(x) = \min_{s' \in F(x)} c(s') \quad \left(\max_{s' \in F(x)} c(s') \right).$$

Wartość $\mathrm{opt}(x)$ nazywamy optimum.

Problemy optymalizacyjne

Niech $|x|$ będzie rozmiarem przypadku $x \in D$ problemu optymalizacyjnego Π. Jeżeli:

Problemy optymalizacyjne

Niech $|x|$ będzie rozmiarem przypadku $x \in D$ problemu optymalizacyjnego Π. Jeżeli:

- $\forall_{s \in F(x)} |s|$ jest ograniczone wielomianowo w $|x|$;

Problemy optymalizacyjne

Niech $|x|$ będzie rozmiarem przypadku $x \in D$ problemu optymalizacyjnego Π. Jeżeli:

- $\forall_{s \in F(x)} |s|$ jest ograniczone wielomianowo w $|x|$;
- języki $\{x : x \in D\}$ i $\{y : y \in F(x)\}$ są rozstrzygalne w czasie wielomianowym;

WYDZIAŁ FIZYKI I INFORMATYKI STOSOWANEJ
Uniwersytet Łódzki

Problemy optymalizacyjne

Niech $|x|$ będzie rozmiarem przypadku $x \in D$ problemu optymalizacyjnego Π. Jeżeli:

- $\forall_{s \in F(x)} |s|$ jest ograniczone wielomianowo w $|x|$;
- języki $\{x : x \in D\}$ i $\{y : y \in F(x)\}$ są rozstrzygalne w czasie wielomianowym;
- $\forall_{x \in D} \forall_{y \in F(x)} c(y)$ można obliczyć w czasie wielomianowym;

Problemy optymalizacyjne

Niech $|x|$ będzie rozmiarem przypadku $x \in D$ problemu optymalizacyjnego Π. Jeżeli:

- $\forall_{s \in F(x)} |s|$ jest ograniczone wielomianowo w $|x|$;
- języki $\{x : x \in D\}$ i $\{y : y \in F(x)\}$ są rozstrzygalne w czasie wielomianowym;
- $\forall_{x \in D} \forall_{y \in F(x)} c(y)$ można obliczyć w czasie wielomianowym;

to Π jest problemem **NP**-optymalizacyjnym.

WYDZIAŁ FIZYKI I INFORMATYKI STOSOWANEJ
Uniwersytet Łódzki

Problemy optymalizacyjne

Z Π wiążemy problem decyzyjny:

Problem (Σ)

Czy dla przypadku $x \in D$ problemu Π i liczby $B \in \mathbb{R}_+$ istnieje rozwiązanie dopuszczalne $s \in F(x)$ dla którego $c(s) < B$ ($c(s) > B$)?

WYDZIAŁ FIZYKI I INFORMATYKI STOSOWANEJ
Uniwersytet Łódzki

Problemy optymalizacyjne

Z Π wiążemy problem decyzyjny:

Problem (Σ)

Czy dla przypadku $x \in D$ problemu Π i liczby $B \in \mathbb{R}_+$ istnieje rozwiązanie dopuszczalne $s \in F(x)$ dla którego $c(s) < B$ ($c(s) > B$)?

- Znając opt(x) można natychmiast podać odpowiedź dla problemu Σ.

WYDZIAŁ FIZYKI I INFORMATYKI STOSOWANEJ
Uniwersytet Łódzki

Problemy optymalizacyjne

Z Π wiążemy problem decyzyjny:

Problem (Σ)

Czy dla przypadku $x \in D$ problemu Π i liczby $B \in \mathbb{R}_+$ istnieje rozwiązanie dopuszczalne $s \in F(x)$ dla którego $c(s) < B$ ($c(s) > B$)?

- Znając opt(x) można natychmiast podać odpowiedź dla problemu Σ.
- Używając algorytmu rozwiązującego Σ można wyznaczyć opt(x) za pomocą wyszukiwania binarnego.

WYDZIAŁ FIZYKI i INFORMATYKI STOSOWANEJ
Uniwersytet Łódzki

Problemy optymalizacyjne

Z Π wiążemy problem decyzyjny:

Problem (Σ)

Czy dla przypadku $x \in D$ problemu Π i liczby $B \in \mathbb{R}_+$ istnieje rozwiązanie dopuszczalne $s \in F(x)$ dla którego $c(s) < B$ ($c(s) > B$)?

- Znając opt(x) można natychmiast podać odpowiedź dla problemu Σ.
- Używając algorytmu rozwiązującego Σ można wyznaczyć opt(x) za pomocą wyszukiwania binarnego.

Problem NP-trudny

Jeżeli problem decyzyjny Σ jest **NP**-zupełny, to skojarzony z nim problem optymalizacyjny Π jest **NP**-trudny.

Algorytmy ϵ-aproxymacyjne

Definicja

Niech $\mathcal{A}$ będzie algorytmem działającym w czasie wielomianowym, którego słowami wejściowymi są przypadki problemu Π. Jeżeli $\forall_{x \in D}\, \mathcal{A}(x) \in F(x)$, to nazywamy go algorytmem aproxymacyjnym dla problemu **NP**-optymalizacyjnego Π.

WYDZIAŁ FIZYKI I INFORMATYKI STOSOWANEJ
Uniwersytet Łódzki

Algorytmy ϵ-aproxymacyjne

Definicja

Niech $\mathcal{A}$ będzie algorytmem działającym w czasie wielomianowym, którego słowami wejściowymi są przypadki problemu Π. Jeżeli $\forall_{x \in D}\, \mathcal{A}(x) \in F(x)$, to nazywamy go algorytmem aproxymacyjnym dla problemu **NP**-optymalizacyjnego Π.

- $\mathcal{A}$ znajduje rozwiązanie dopuszczalne dla $x \in D$.

WYDZIAŁ FIZYKI I INFORMATYKI STOSOWANEJ
Uniwersytet Łódzki

Algorytmy ϵ-aproxymacyjne

Definicja

Niech $\mathcal{A}$ będzie algorytmem działającym w czasie wielomianowym, którego słowami wejściowymi są przypadki problemu Π. Jeżeli $\forall_{x \in D}\, \mathcal{A}(x) \in F(x)$, to nazywamy go algorytmem aproxymacyjnym dla problemu **NP**-optymalizacyjnego Π.

- $\mathcal{A}$ znajduje rozwiązanie dopuszczalne dla $x \in D$.
- Interesują nas rozwiązania dopuszczalne będące przybliżeniami rozwiązania optymalnego.

WYDZIAŁ FIZYKI I INFORMATYKI STOSOWANEJ
Uniwersytet Łódzki

Algorytmy ϵ-aproxymacyjne

Definicja

Algorytm $\mathcal{A}$ jest ϵ-aproxymacyjny, jeżeli $\forall_{x \in D}\, \mathcal{A}(x) \in F(x)$ oraz

$$\exists \epsilon \geqslant 0 : \frac{|c(\mathcal{A}(x)) - \mathrm{opt}(x)|}{\max\{c(\mathcal{A}(x)), \mathrm{opt}(x)\}} \leqslant \epsilon.$$

WYDZIAŁ FIZYKI I INFORMATYKI STOSOWANEJ
Uniwersytet Łódzki

Algorytmy ϵ-aproxymacyjne

Definicja

Algorytm $\mathcal{A}$ jest ϵ-aproxymacyjny, jeżeli $\forall_{x \in D}\, \mathcal{A}(x) \in F(x)$ oraz

$$\exists \epsilon \geqslant 0 : \frac{|c(\mathcal{A}(x)) - \mathrm{opt}(x)|}{\max\{c(\mathcal{A}(x)), \mathrm{opt}(x)\}} \leqslant \epsilon.$$

- dla minimalizacji $c(\mathcal{A}(x)) \leqslant \dfrac{\mathrm{opt}(x)}{1-\epsilon}$;

WYDZIAŁ FIZYKI I INFORMATYKI STOSOWANEJ
Uniwersytet Łódzki

Algorytmy ϵ-aproxymacyjne

Definicja

Algorytm $\mathcal{A}$ jest ϵ-aproxymacyjny, jeżeli $\forall_{x \in D}\, \mathcal{A}(x) \in F(x)$ oraz

$$\exists \epsilon \geqslant 0 : \frac{|c(\mathcal{A}(x)) - \mathrm{opt}(x)|}{\max\{c(\mathcal{A}(x)), \mathrm{opt}(x)\}} \leqslant \epsilon.$$

- dla minimalizacji $c(\mathcal{A}(x)) \leqslant \dfrac{\mathrm{opt}(x)}{1-\epsilon}$;
- dla maxymalizacji $c(\mathcal{A}(x)) \geqslant (1-\epsilon)\,\mathrm{opt}(x)$;

WYDZIAŁ FIZYKI I INFORMATYKI STOSOWANEJ
Uniwersytet Łódzki

Algorytmy ϵ-aproxymacyjne

Definicja

Algorytm $\mathcal{A}$ jest ϵ-aproxymacyjny, jeżeli $\forall_{x \in D}\, \mathcal{A}(x) \in F(x)$ oraz

$$\exists \epsilon \geqslant 0 : \frac{|c(\mathcal{A}(x)) - \mathrm{opt}(x)|}{\max\{c(\mathcal{A}(x)), \mathrm{opt}(x)\}} \leqslant \epsilon.$$

- dla minimalizacji $c(\mathcal{A}(x)) \leqslant \dfrac{\mathrm{opt}(x)}{1-\epsilon}$;
- dla maxymalizacji $c(\mathcal{A}(x)) \geqslant (1-\epsilon)\,\mathrm{opt}(x)$;
- jeżeli $\epsilon = 0$, to $\mathcal{A}(x)$ jest rozwiązaniem optymalnym;

WYDZIAŁ FIZYKI i INFORMATYKI STOSOWANEJ
Uniwersytet Łódzki

Algorytmy ϵ-aproxymacyjne

Definicja

Algorytm $\mathcal{A}$ jest ϵ-aproxymacyjny, jeżeli $\forall_{x \in D}\, \mathcal{A}(x) \in F(x)$ oraz

$$\exists \epsilon \geqslant 0 : \frac{|c(\mathcal{A}(x)) - \mathrm{opt}(x)|}{\max\{c(\mathcal{A}(x)), \mathrm{opt}(x)\}} \leqslant \epsilon.$$

- dla minimalizacji $c(\mathcal{A}(x)) \leqslant \dfrac{\mathrm{opt}(x)}{1-\epsilon}$;
- dla maxymalizacji $c(\mathcal{A}(x)) \geqslant (1-\epsilon)\,\mathrm{opt}(x)$;
- jeżeli $\epsilon = 0$, to $\mathcal{A}(x)$ jest rozwiązaniem optymalnym;
- jeżeli $\epsilon = 1$, to $\mathcal{A}$ jest heurystyką, a $\mathcal{A}(x)$ może być dowolnie odległe od rozwiązania optymalnego.

WYDZIAŁ FIZYKI I INFORMATYKI STOSOWANEJ
Uniwersytet Łódzki

Algorytmy ϵ-aproxymacyjne

Definicja

Algorytm $\mathcal{A}$ jest ϵ-aproxymacyjny, jeżeli $\forall_{x \in D}\, \mathcal{A}(x) \in F(x)$ oraz

$$\exists \epsilon \geqslant 0 : \frac{|c(\mathcal{A}(x)) - \mathrm{opt}(x)|}{\max\{c(\mathcal{A}(x)), \mathrm{opt}(x)\}} \leqslant \epsilon.$$

- dla minimalizacji $c(\mathcal{A}(x)) \leqslant \dfrac{\mathrm{opt}(x)}{1-\epsilon}$;
- dla maxymalizacji $c(\mathcal{A}(x)) \geqslant (1-\epsilon)\,\mathrm{opt}(x)$;
- jeżeli $\epsilon = 0$, to $\mathcal{A}(x)$ jest rozwiązaniem optymalnym;
- jeżeli $\epsilon = 1$, to $\mathcal{A}$ jest heurystyką, a $\mathcal{A}(x)$ może być dowolnie odległe od rozwiązania optymalnego.

Definicja

$\inf\{\epsilon > 0 : \exists$ algorytm ϵ-aproxymacyjny dla $\Pi\}$ to próg aproxymacji problemu Π.

Minimalne pokrycie wierzchołkowe

Problem (MINIMUM NODE COVER)

Minimalne pokrycie wierzchołkowe

Problem (MINIMUM NODE COVER)

Wejście: $G(V, E)$

Minimalne pokrycie wierzchołkowe

Problem (MINIMUM NODE COVER)

Wejście: $G(V, E)$

Wyjście: $C \subseteq V$ *takie, że* $\text{opt}(G) = |C| = \min_{V' \subseteq V}\{|V'| : V' \textit{ jest pokryciem wierzchołkowym } G\}$

Minimalne pokrycie wierzchołkowe

Problem (MINIMUM NODE COVER)

Wejście: $G(V, E)$
Wyjście: $C \subseteq V$ *takie, że* $\text{opt}(G) = |C| = \min_{V' \subseteq V}\{|V'| : V' \textit{ jest pokryciem wierzchołkowym } G\}$

- dopuszczalność rozwiązania można sprawdzić w czasie liniowym od rozmiaru grafu;

Minimalne pokrycie wierzchołkowe

Problem (MINIMUM NODE COVER)

Wejście: $G(V, E)$
Wyjście: $C \subseteq V$ *takie, że* $\text{opt}(G) = |C| = \min_{V' \subseteq V}\{|V'| : V' \textit{ jest pokryciem wierzchołkowym } G\}$

- dopuszczalność rozwiązania można sprawdzić w czasie liniowym od rozmiaru grafu;
- funkcję celu można obliczyć w czasie liniowym od rozmiaru grafu.

WYDZIAŁ FIZYKI I INFORMATYKI STOSOWANEJ
Uniwersytet Łódzki

Minimalne pokrycie wierzchołkowe

Problem (MINIMUM NODE COVER)

Wejście: $G(V, E)$
Wyjście: $C \subseteq V$ *takie, że* $\text{opt}(G) = |C| = \min_{V' \subseteq V}\{|V'| : V' \textit{ jest pokryciem wierzchołkowym } G\}$

- dopuszczalność rozwiązania można sprawdzić w czasie liniowym od rozmiaru grafu;
- funkcję celu można obliczyć w czasie liniowym od rozmiaru grafu.

Pełne przeszukanie

Algorytm przeprowadzający pełne przeszukanie działa w czasie $2^{|C|}|V|^{O(1)}$.

WYDZIAŁ FIZYKI i INFORMATYKI STOSOWANEJ
Uniwersytet Łódzki

Minimalne pokrycie wierzchołkowe

Zadanie równoważne do MINIMUM NODE COVER:

Minimalne pokrycie wierzchołkowe

Zadanie równoważne do MINIMUM NODE COVER:

Przykład (programowanie liniowe binarne)

Dla grafu $G = (V, E)$ znajdź

$$c = \min \sum_{v \in V} x_v$$

przy warunku

$$\forall_{\{v,w\} \in E} \, x_v + x_w \geqslant 1$$

gdzie $\forall_{v \in V} \, x_v \in \{0, 1\}$.

WYDZIAŁ FIZYKI I INFORMATYKI STOSOWANEJ
Uniwersytet Łódzki

Minimalne pokrycie wierzchołkowe

Algorytm zachłanny

$\tilde{C} \leftarrow \emptyset$
while $|E| > 0$ **do**
    wybierz wierzchołek $v \in V$ o największym stopniu
    $\tilde{C} \leftarrow \tilde{C} \cup \{v\}$
    $G \leftarrow G \setminus \{v\}$
end while

Heurystyka

Powyższy algorytm nie jest algorytmem ϵ-aproxymacyjnym dla żadnego $\epsilon < 1$ — współczynnik błędu rośnie jak $\log |V|$.

WYDZIAŁ FIZYKI I INFORMATYKI STOSOWANEJ
Uniwersytet Łódzki

Minimalne pokrycie wierzchołkowe

Algorytm skojarzeniowy

$\tilde{C} \leftarrow \emptyset$
while $|E| > 0$ **do**
    wybierz dowolną krawędź $\{v, w\} \in E$
    $\tilde{C} \leftarrow \tilde{C} \cup \{v, w\}$
    $G \leftarrow G \setminus \{v, w\}$
end while

Lemat

Podany algorytm tworzy skojarzenie maxymalne.

Dowód.

Każda para dodana do $\tilde{C}$ jest jedyna: algorytm nie doda krawędzi $\{v, w\}$ i $\{v, u\}$, bo gdy trafi na $\{v, w\}$ doda v i w, więc krawędź $\{v, u\}$ jest pokryta przez wierzchołek v. Zatem algorytm tworzy w G skojarzenie i jest ono maxymalne, bo iterowanie trwa dopóki są niepokryte krawędzie. □

Minimalne pokrycie wierzchołkowe

Lemat

Dla dowolnego skojarzenia maxymalnego M w grafie G

$$\mathrm{opt}(G) \geqslant |M|.$$

Minimalne pokrycie wierzchołkowe

Lemat

Dla dowolnego skojarzenia maxymalnego M w grafie G

$$\mathrm{opt}(G) \geqslant |M|.$$

Dowód.

Do skojarzenia maxymalnego nie można dodać więcej krawędzi (z definicji), więc każda pozostała krawędź sąsiaduje z przynajmniej jedną krawędzią z M. Jeżeli weźmiemy oba końce wszystkich krawędzi w M jako pokrycie wierzchołkowe V', to wszystkie krawędzie wewnątrz V' są pokryte, a wszystkie pozostałe krawędzie dochodzą do V' (bo sąsiadują z M). Zatem $|V'| \geqslant |M|$, bo V' musi włączać przynajmniej jeden z końców wszystkich krawędzi w M, aby pokryć wszystkie krawędzie w G. □

WYDZIAŁ FIZYKI i INFORMATYKI STOSOWANEJ
Uniwersytet Łódzki

Minimalne pokrycie wierzchołkowe

Fakt

Próg aproxymacji problemu MINIMUM NODE COVER *nie mniejszy niż* $\frac{1}{2}$.

Minimalne pokrycie wierzchołkowe

Fakt

Próg aproxymacji problemu MINIMUM NODE COVER *nie mniejszy niż* $\frac{1}{2}$.

Dowód.

Z algorytmu i 1. Lematu wnioskujemy, że $|\mathcal{A}(G)| = 2|M|$ (bierzemy oba wierzchołki skojarzenia). Zestawiając to z wynikiem 2. Lematu, mamy

$$\mathrm{opt}(G) \geqslant |M| = \frac{1}{2}|\mathcal{A}(G)|,$$

czyli

$$\frac{|c(\mathcal{A}(G)) - \mathrm{opt}(G)|}{|c(\mathcal{A}(G))|} \leqslant \frac{1}{2},$$

tzn. rozmiar pokrycia wierzchołkowego otrzymanego z algorytmu jest mniej niż dwukrotnie większy od rozmiaru minimalnego pokrycia wierzchołkowego. □

Problem komiwojażera

TSP (*Travelling Salesman Problem*)

Problem (TSP)

Problem komiwojażera

TSP (*Travelling Salesman Problem*)

Problem (TSP)

Wejście: $G(V, E)$, $w : E \to \mathbb{N}$

Problem komiwojażera

TSP (*Travelling Salesman Problem*)

Problem (TSP)

Wejście: $G(V, E)$, $w : E \to \mathbb{N}$

Wyjście: *cykl Hamiltona* C *w* G *taki, że*
$c(C) = \min_{C' \in G}\{c(C') : C'$ *cykl Hamiltona w* $G\}$, *gdzie*
$c(C') = \sum_{e \in E(C')} w(e)$

WYDZIAŁ FIZYKI I INFORMATYKI STOSOWANEJ
Uniwersytet Łódzki

Problem komiwojażera

TSP (*Travelling Salesman Problem*)

Problem (TSP)

Wejście: $G(V, E)$, $w\colon E \to \mathbb{N}$

Wyjście: *cykl Hamiltona* C *w* G *taki, że*
$c(C) = \min_{C' \in G}\{c(C')\colon C'$ *cykl Hamiltona w* $G\}$, *gdzie*
$c(C') = \sum_{e \in E(C')} w(e)$

Twierdzenie

Jeżeli $\mathbf{P} \neq \mathbf{NP}$, *to próg aproxymacyjny dla* TSP *wynosi* 1.

WYDZIAŁ FIZYKI i INFORMATYKI STOSOWANEJ
Uniwersytet Łódzki

Problem komiwojażera

Dowód.

Przypuśćmy, że dla pewnego $\epsilon < 1$ istnieje wielomianowy algorytm ϵ-aproxymacyjny $\mathcal{A}$ dla TSP. $\mathcal{A}$ mógłby zostać użyty do rozwiązania HAMILTONIAN CYCLE w czasie wielomianowym.

Problem komiwojażera

Dowód.

Przypuśćmy, że dla pewnego $\epsilon < 1$ istnieje wielomianowy algorytm ϵ-aproxymacyjny $\mathcal{A}$ dla TSP. $\mathcal{A}$ mógłby zostać użyty do rozwiązania HAMILTONIAN CYCLE w czasie wielomianowym.
Redukujemy HAMILTONIAN CYCLE dla grafu $G = (V, E)$ do TSP na grafie pełnym $G' = (V, V \times V)$ z

$$w(\{u, v\}) = \begin{cases} 1 & \{u, v\} \in E \\ \frac{|V|}{1-\epsilon} & \{u, v\} \notin E. \end{cases}$$

Problem komiwojażera

Dowód.

Przypuśćmy, że dla pewnego $\epsilon < 1$ istnieje wielomianowy algorytm ϵ-aproxymacyjny $\mathcal{A}$ dla TSP. $\mathcal{A}$ mógłby zostać użyty do rozwiązania HAMILTONIAN CYCLE w czasie wielomianowym.
Redukujemy HAMILTONIAN CYCLE dla grafu $G = (V, E)$ do TSP na grafie pełnym $G' = (V, V \times V)$ z

$$w(\{u, v\}) = \begin{cases} 1 & \{u, v\} \in E \\ \frac{|V|}{1-\epsilon} & \{u, v\} \notin E. \end{cases}$$

Uruchamiamy $\mathcal{A}$ na G'. Jeżeli G ma cykl Hamitona, to $c(\mathcal{A}(G')) = |V|$. Jeżeli G nie ma cyklu Hamiltona, to każdy cykl Hamiltona w G' zawiera przynajmniej jedną krawędź o wadze $\frac{|V|}{1-\epsilon}$, więc $c(\mathcal{A}(G')) > \frac{|V|}{1-\epsilon}$. W drugą stronę — ponieważ $\mathcal{A}$ jest ϵ-aproxymacyjny, więc wnioskujemy, że $\text{opt}(G) > V$ i G nie może mieć cyklu Hamiltona.

Problem komiwojażera

Dowód.

Przypuśćmy, że dla pewnego $\epsilon < 1$ istnieje wielomianowy algorytm ϵ-aproxymacyjny $\mathcal{A}$ dla TSP. $\mathcal{A}$ mógłby zostać użyty do rozwiązania HAMILTONIAN CYCLE w czasie wielomianowym.
Redukujemy HAMILTONIAN CYCLE dla grafu $G = (V, E)$ do TSP na grafie pełnym $G' = (V, V \times V)$ z

$$w(\{u, v\}) = \begin{cases} 1 & \{u, v\} \in E \\ \frac{|V|}{1-\epsilon} & \{u, v\} \notin E. \end{cases}$$

Uruchamiamy $\mathcal{A}$ na G'. Jeżeli G ma cykl Hamitona, to $c(\mathcal{A}(G')) = |V|$. Jeżeli G nie ma cyklu Hamiltona, to każdy cykl Hamiltona w G' zawiera przynajmniej jedną krawędź o wadze $\frac{|V|}{1-\epsilon}$, więc $c(\mathcal{A}(G')) > \frac{|V|}{1-\epsilon}$. W drugą stronę — ponieważ $\mathcal{A}$ jest ϵ-aproxymacyjny, więc wnioskujemy, że $\text{opt}(G) > V$ i G nie może mieć cyklu Hamiltona.
G ma cykl Hamiltona wtedy i tylko wtedy, gdy $c(\mathcal{A}(G')) = |V|$, co można sprawdzić w czasie wielomianowym. Ponieważ HAMILTONIAN CYCLE jest **NP**-zupełny, więc **P** = **NP**, co przeczyłoby założeniu, lub algorytm $\mathcal{A}$ nie istnieje. □

Metryczny problem komiwojażera

Δ-TSP

Problem (Δ-TSP)

Metryczny problem komiwojażera

Δ-TSP

Problem (Δ-TSP)

Wejście: $G(V, E)$, $w : V \times V \to \mathbb{N}$ *spełniająca nierówność trójkąta:* $\forall u, v, z \in V : w(u, v) \leqslant w(u, z) + w(v, z)$

Metryczny problem komiwojażera

Δ-TSP

Problem (Δ-TSP)

Wejście: *$G(V, E)$, $w : V \times V \to \mathbb{N}$ spełniająca nierówność trójkąta: $\forall u, v, z \in V : w(u, v) \leqslant w(u, z) + w(v, z)$*

Wyjście: *cykl Hamiltona C w G taki, że $w(C) = \min_{C' \in G}\{w(C') : C'$ cykl Hamiltona w $G\}$, gdzie $w(C') = \sum_{e \in E(C')} w(e)$ $(w(\{u, v\}) \equiv w(u, v))$*

WYDZIAŁ FIZYKI I INFORMATYKI STOSOWANEJ
Uniwersytet Łódzki

Metryczny problem komiwojażera

Δ-TSP

Problem (Δ-TSP)

Wejście: *$G(V, E)$, $w\colon V \times V \to \mathbb{N}$ spełniająca nierówność trójkąta: $\forall u, v, z \in V\colon w(u, v) \leqslant w(u, z) + w(v, z)$*

Wyjście: *cykl Hamiltona C w G taki, że $w(C) = \min_{C' \in G}\{w(C')\colon C' \text{ cykl Hamiltona w } G\}$, gdzie $w(C') = \sum_{e \in E(C')} w(e)$ $(w(\{u, v\}) \equiv w(u, v))$*

Fakt

Istnieje algorytm $\frac{1}{2}$-aproxymacyjny dla Δ-TSP.

WYDZIAŁ FIZYKI I INFORMATYKI STOSOWANEJ
Uniwersytet Łódzki

Aproxymowalność i złożoność

„Hierarchia'' rozwiązań problemów optymalizacyjnych

1. dokładny algorytm wielomianowy

Aproxymowalność i złożoność

„Hierarchia” rozwiązań problemów optymalizacyjnych

1. dokładny algorytm wielomianowy
2. wielomianowy algorytm aproxymacyjny

WYDZIAŁ FIZYKI I INFORMATYKI STOSOWANEJ
Uniwersytet Łódzki

Aproxymowalność i złożoność

„Hierarchia'' rozwiązań problemów optymalizacyjnych

1. dokładny algorytm wielomianowy
2. wielomianowy algorytm aproxymacyjny

Problem

Dla jakich problemów **NP***-optymalizacyjnych istnieją wielomianowe algorytmy aproxymacyjne?*

WYDZIAŁ FIZYKI I INFORMATYKI STOSOWANEJ
Uniwersytet Łódzki

MaxSNP

Definicje

Formuły logiki rzędu

MaxSNP

Definicje

Formuły logiki rzędu

0 zawierają zmienne zdaniowe i spójniki zdaniowe — rachunek zdań — np. $x \in P \vee x \notin P$;

MaxSNP

Definicje

Formuły logiki rzędu

0 zawierają zmienne zdaniowe i spójniki zdaniowe — rachunek zdań — np. $x \in P \vee x \notin P$;

1 zawierają ponadto kwantyfikatory oparte na elementach — rachunek kwantyfikatorów — $\forall_x(x \in P \vee x \notin P)$

WYDZIAŁ FIZYKI I INFORMATYKI STOSOWANEJ
Uniwersytet Łódzki

MaxSNP

Definicje

Formuły logiki rzędu

0. zawierają zmienne zdaniowe i spójniki zdaniowe — rachunek zdań — np. $x \in P \vee x \notin P$;
1. zawierają ponadto kwantyfikatory oparte na elementach — rachunek kwantyfikatorów — $\forall_x(x \in P \vee x \notin P)$
2. zawierają ponadto kwantyfikatory oparte na relacjach — $\forall_P \forall_x(x \in P \vee x \notin P)$

WYDZIAŁ FIZYKI i INFORMATYKI STOSOWANEJ
Uniwersytet Łódzki

MaxSNP

Definicje

Formuły logiki rzędu

0 zawierają zmienne zdaniowe i spójniki zdaniowe — rachunek zdań — np. $x \in P \vee x \notin P$;

1 zawierają ponadto kwantyfikatory oparte na elementach — rachunek kwantyfikatorów — $\forall_x(x \in P \vee x \notin P)$

2 zawierają ponadto kwantyfikatory oparte na relacjach — $\forall_P \forall_x(x \in P \vee x \notin P)$

Egzystencjalna logika 2. rzędu **SO∃** — klasa formuł, w których wszystkie kwantyfikatory 2. rzędu są egzystencjalne (tj. $\exists$).

WYDZIAŁ FIZYKI i INFORMATYKI STOSOWANEJ
Uniwersytet Łódzki

MaxSNP

Definicje

Formuły logiki rzędu

0. zawierają zmienne zdaniowe i spójniki zdaniowe — rachunek zdań — np. $x \in P \vee x \notin P$;
1. zawierają ponadto kwantyfikatory oparte na elementach — rachunek kwantyfikatorów — $\forall_x(x \in P \vee x \notin P)$
2. zawierają ponadto kwantyfikatory oparte na relacjach — $\forall_P \forall_x(x \in P \vee x \notin P)$

Egzystencjalna logika 2. rzędu **SO∃** — klasa formuł, w których wszystkie kwantyfikatory 2. rzędu są egzystencjalne (tj. ∃).

Twierdzenie (Fagin)

$$\mathbf{NP} = \mathbf{SO\exists}.$$

MaxSNP

Definicja

SNP (**StrictNP**) — klasa formuł z **SO∃**, w których wszystkie kwantyfikatory 1. rzędu są uniwersalne (tj. ∀).

MaxSNP

Definicja

SNP (**StrictNP**) — klasa formuł z **SO∃**, w których wszystkie kwantyfikatory 1. rzędu są uniwersalne (tj. ∀).

Przykład (formuła z **SNP**)

$$\exists_S \forall_{x_1} \forall_{x_2} \ldots \forall_{x_k} \phi$$

WYDZIAŁ FIZYKI I INFORMATYKI STOSOWANEJ
Uniwersytet Łódzki

MaxSNP

Definicja

SNP (**StrictNP**) — klasa formuł z **SO∃**, w których wszystkie kwantyfikatory 1. rzędu są uniwersalne (tj. ∀).

Przykład (formuła z **SNP**)

$$\exists_S \forall_{x_1} \forall_{x_2} \ldots \forall_{x_k} \phi$$

Definicja

MaxSNP$_0$ — klasa problemów optymalizacyjnych określanych przez formułę

$$\max_S |\{(x_1, x_2, \ldots, x_k) : \phi\}|$$

WYDZIAŁ FIZYKI i INFORMATYKI STOSOWANEJ
Uniwersytet Łódzki

MaxSNP

Definicja

SNP (**StrictNP**) — klasa formuł z **SO∃**, w których wszystkie kwantyfikatory 1. rzędu są uniwersalne (tj. ∀).

Przykład (formuła z **SNP**)

$$\exists_S \forall_{x_1} \forall_{x_2} \ldots \forall_{x_k} \phi$$

Definicja

MaxSNP$_0$ — klasa problemów optymalizacyjnych określanych przez formułę

$$\max_S |\{(x_1, x_2, \ldots, x_k) : \phi\}|$$

Szukamy takiej relacji S, że formuła ϕ zachodzi dla tak wielu krotek $(x_1, x_2, \ldots, x_k)$, jak to tylko możliwe.

WYDZIAŁ FIZYKI i INFORMATYKI STOSOWANEJ Uniwersytet Łódzki

MaxSNP

Definicja

Niech Π_1 i Π_2 będą problemami optymalizacyjnymi z kosztami odpowiednio c_1 i c_2. Para funkcji f i g tworzy L-redukcję jeżeli zachodzą następujące warunki:

MaxSNP

Definicja

Niech Π_1 i Π_2 będą problemami optymalizacyjnymi z kosztami odpowiednio c_1 i c_2. Para funkcji f i g tworzy L-redukcję jeżeli zachodzą następujące warunki:

- f i g są obliczalne w czasie wielomianowym;

MaxSNP

Definicja

Niech Π_1 i Π_2 będą problemami optymalizacyjnymi z kosztami odpowiednio c_1 i c_2. Para funkcji f i g tworzy L-redukcję jeżeli zachodzą następujące warunki:

- f i g są obliczalne w czasie wielomianowym;
- $x \in D_1 \implies f(x) \in D_2$;

MaxSNP

Definicja

Niech Π_1 i Π_2 będą problemami optymalizacyjnymi z kosztami odpowiednio c_1 i c_2. Para funkcji f i g tworzy L-redukcję jeżeli zachodzą następujące warunki:

- f i g są obliczalne w czasie wielomianowym;
- $x \in D_1 \implies f(x) \in D_2$;
- $s \in F_2(f(x)) \implies g(s) \in F_1(x)$;

WYDZIAŁ FIZYKI I INFORMATYKI STOSOWANEJ
Uniwersytet Łódzki

MaxSNP

Definicja

Niech Π_1 i Π_2 będą problemami optymalizacyjnymi z kosztami odpowiednio c_1 i c_2. Para funkcji f i g tworzy L-redukcję jeżeli zachodzą następujące warunki:

- f i g są obliczalne w czasie wielomianowym;
- $x \in D_1 \implies f(x) \in D_2$;
- $s \in F_2(f(x)) \implies g(s) \in F_1(x)$;
- $\exists_{\alpha>0}\ \mathrm{opt}_2(f(x)) \leqslant \alpha\, \mathrm{opt}_1(x)$;

WYDZIAŁ FIZYKI I INFORMATYKI STOSOWANEJ
Uniwersytet Łódzki

MaxSNP

Definicja

Niech Π_1 i Π_2 będą problemami optymalizacyjnymi z kosztami odpowiednio c_1 i c_2. Para funkcji f i g tworzy L-redukcję jeżeli zachodzą następujące warunki:

- f i g są obliczalne w czasie wielomianowym;
- $x \in D_1 \implies f(x) \in D_2$;
- $s \in F_2(f(x)) \implies g(s) \in F_1(x)$;
- $\exists_{\alpha>0}\ \mathrm{opt}_2(f(x)) \leqslant \alpha\, \mathrm{opt}_1(x)$;
- $\exists_{\beta>0}\forall_{s\in F_2(f(x))}\, |\mathrm{opt}_1(x) - c_1(g(s))| \leqslant \beta |\mathrm{opt}_2(f(x)) - c_2(s)|$

WYDZIAŁ FIZYKI I INFORMATYKI STOSOWANEJ
Uniwersytet Łódzki

MaxSNP

Definicja

Niech Π_1 i Π_2 będą problemami optymalizacyjnymi z kosztami odpowiednio c_1 i c_2. Para funkcji f i g tworzy L-redukcję jeżeli zachodzą następujące warunki:

- f i g są obliczalne w czasie wielomianowym;
- $x \in D_1 \implies f(x) \in D_2$;
- $s \in F_2(f(x)) \implies g(s) \in F_1(x)$;
- $\exists_{\alpha>0}\ \mathrm{opt}_2(f(x)) \leqslant \alpha\, \mathrm{opt}_1(x)$;
- $\exists_{\beta>0}\forall_{s\in F_2(f(x))}\, |\mathrm{opt}_1(x) - c_1(g(s))| \leqslant \beta |\mathrm{opt}_2(f(x)) - c_2(s)|$

Definicja

MaxSNP to klasa problemów optymalizacyjnych L-redukowalnych do problemów z $\mathbf{MaxSNP}_0$.

MaxSNP

Przykład (MAX CUT)

MAX CUT $\in$ **MaxSNP**$_0$ więc i MAX CUT $\in$ **MaxSNP**:

$$\max_{S \subseteq V} |\{(v, w) : \{v, w\} \in E \wedge v \in S \wedge w \notin S\}|.$$

WYDZIAŁ FIZYKI I INFORMATYKI STOSOWANEJ
Uniwersytet Łódzki

MaxSNP

Przykład (MAX CUT)

MAX CUT $\in$ **MaxSNP**$_0$ więc i MAX CUT $\in$ **MaxSNP**:

$$\max_{S \subseteq V} |\{(v, w) : \{v, w\} \in E \wedge v \in S \wedge w \notin S\}|.$$

Twierdzenie

Jeżeli $\Pi \in$ **MaxSNP**$_0$ *jest postaci* $\max_S |\{(x_1, \ldots, x_k) : \phi\}|$, *to istnieje dla* Π *algorytm* $(1 - 2^{-k})$*-aproxymacyjny, gdzie* k *jest liczbą tych formuł atomowych w formule* ϕ, *które zawierają* S.

WYDZIAŁ FIZYKI i INFORMATYKI STOSOWANEJ
Uniwersytet Łódzki

MaxSNP-zupełność

Definicja

Problem $\Pi \in$ **MaxSNP** jest **MaxSNP**-zupełny, jeżeli dowolny problem z **MaxSNP** L-redukuje się do niego.

MaxSNP-zupełność

Definicja

Problem $\Pi \in$ **MaxSNP** jest **MaxSNP**-zupełny, jeżeli dowolny problem z **MaxSNP** L-redukuje się do niego.

Fakt

MAX CUT *jest* **MaxSNP**-*zupełny.*

WYDZIAŁ FIZYKI I INFORMATYKI STOSOWANEJ
Uniwersytet Łódzki

MaxSNP-zupełność

Definicja

Problem $\Pi \in$ **MaxSNP** jest **MaxSNP**-zupełny, jeżeli dowolny problem z **MaxSNP** L-redukuje się do niego.

Fakt

MAX CUT *jest* **MaxSNP**-*zupełny.*

Problem

WYDZIAŁ FIZYKI I INFORMATYKI STOSOWANEJ
Uniwersytet Łódzki

MaxSNP-zupełność

Definicja

Problem $\Pi \in$ **MaxSNP** jest **MaxSNP**-zupełny, jeżeli dowolny problem z **MaxSNP** L-redukuje się do niego.

Fakt

MAX CUT *jest* **MaxSNP**-*zupełny.*

Problem

Wejście: ϕ — *formuła logiczna z co najwyżej 3 zmiennymi na klauzulę*

WYDZIAŁ FIZYKI i INFORMATYKI STOSOWANEJ
Uniwersytet Łódzki

MaxSNP-zupełność

Definicja

Problem $\Pi \in$ **MaxSNP** jest **MaxSNP**-zupełny, jeżeli dowolny problem z **MaxSNP** L-redukuje się do niego.

Fakt

MAX CUT *jest* **MaxSNP**-*zupełny.*

Problem

Wejście: *ϕ — formuła logiczna z co najwyżej 3 zmiennymi na klauzulę*

Wyjście: *T – wartościowanie zmiennych spełniające największą liczbę klauzul*

WYDZIAŁ FIZYKI I INFORMATYKI STOSOWANEJ
Uniwersytet Łódzki

MaxSNP-zupełność

Definicja

Problem $\Pi \in$ **MaxSNP** jest **MaxSNP**-zupełny, jeżeli dowolny problem z **MaxSNP** L-redukuje się do niego.

Fakt

MAX CUT *jest* **MaxSNP**-*zupełny.*

Problem

Wejście: *ϕ — formuła logiczna z co najwyżej 3 zmiennymi na klauzulę*

Wyjście: *T – wartościowanie zmiennych spełniające największą liczbę klauzul*

Fakt

MAX3SAT *jest* **MaxSNP**-*zupełny.*

Nieaproxymowalność

Twierdzenie

Jeżeli istnieje wielomianowy schemat aproxymacji dla MAX3SAT, *to* $\mathbf{P} = \mathbf{NP}$.

Nieaproxymowalność

Twierdzenie

Jeżeli istnieje wielomianowy schemat aproxymacji dla MAX3SAT, *to* $\mathbf{P} = \mathbf{NP}$.

Wniosek

Żaden problem **MaxSNP**-*zupełny nie może mieć wielomianowego schematu aproxymacji, chyba że* $\mathbf{P} = \mathbf{NP}$.

WYDZIAŁ FIZYKI I INFORMATYKI STOSOWANEJ
Uniwersytet Łódzki