Algorytmy aproksymacyjne i parametryzowane

Marek Cygan

Uniwersytet Warszawski

18 października 2012

Wstęp

- W algorytmice problemy dzielimy na obliczeniowo trudne (NP-trudne) oraz proste ($\in P$).

Wstęp

- W algorytmice problemy dzielimy na obliczeniowo trudne (NP-trudne) oraz proste ($\in P$).
- $2^{\mathrm{poly}(n)}$ vs $\mathrm{poly}(n)$.

Wstęp

- W algorytmice problemy dzielimy na obliczeniowo trudne (NP-trudne) oraz proste ($\in P$).
- $2^{\mathrm{poly}(n)}$ vs $\mathrm{poly}(n)$.
- Jeśli P $\neq$ NP, to co zrobić z problemami NP-trudnymi?

Wstęp

- W algorytmice problemy dzielimy na obliczeniowo trudne (NP-trudne) oraz proste ($\in P$).
- $2^{\mathrm{poly}(n)}$ vs $\mathrm{poly}(n)$.
- Jeśli P $\neq$ NP, to co zrobić z problemami NP-trudnymi?
- Dwa wyjścia: algorytmy aproksymacyjne, algorytmy parametryzowane.

Część I: Algorytmy aproksymacyjne

Algorytmy aproksymacyjne

Definicja

Algorytm ρ-**aproksymacyjny** dla problemu minimalizacyjnego w czasie $\mathrm{poly}(n)$ znajduje rozwiązanie o koszcie $\leqslant \rho \cdot \mathrm{OPT}$.

- Kompromis polega na zachowaniu czasu wielomianowego, kosztem dokładności rozwiązania.

Przykład problemu

- W problemie k-Center mamy dany zbiór miejscowości (V), które musimy obsłużyć otwierając k centrów obsługi klienta (COK).

Przykład problemu

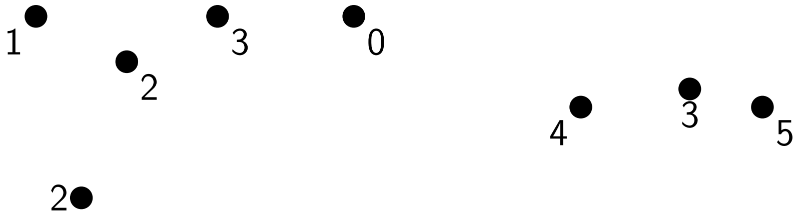
- W problemie k-Center mamy dany zbiór miejscowości (V), które musimy obsłużyć otwierając k centrów obsługi klienta (COK).
- Chcemy przypisać COK dla każdej miejscowości, tak aby zminimalizować najdalszą odległość pomiędzy klientem a jego COK.

Przykład problemu

- W problemie k-Center mamy dany zbiór miejscowości (V), które musimy obsłużyć otwierając k centrów obsługi klienta (COK).
- Chcemy przypisać COK dla każdej miejscowości, tak aby zminimalizować najdalszą odległość pomiędzy klientem a jego COK.
- Dodatkowo COK mają przepustowości — jeśli otworzymy COK w x, to może ono obsłużyć $L(x)$ miejscowości.

Definicja problemu

k-CENTER Z PRZEPUSTOWOŚCIAMI
Wejście: Metryka $\mathrm{d} : V \times V \to \mathbb{R}_+$, funkcja przepustowości $L : V \to \mathbb{N}$ oraz liczba k.
Wyjście: Zbiór $S \subseteq V$ rozmiaru k oraz funkcja $\phi : V \to S$, taka że dla każdego $u \in S$ mamy $|\phi^{-1}(u)| \leqslant L(u)$.
Cel: Zminimalizować $\max_{v \in V} \mathrm{d}(v, \phi(v))$.



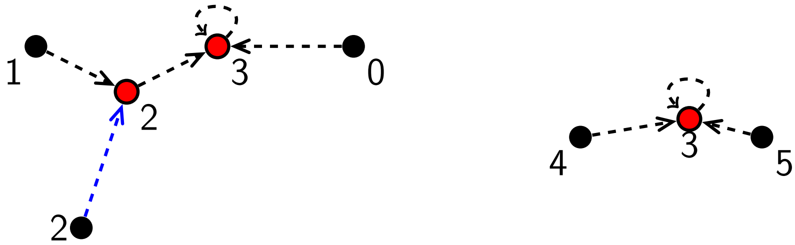
Definicja problemu

k-Center z przepustowościami

Wejście: Metryka $\mathrm{d} : V \times V \to \mathbb{R}_+$, funkcja przepustowości $L : V \to \mathbb{N}$ oraz liczba k.

Wyjście: Zbiór $S \subseteq V$ rozmiaru k oraz funkcja $\phi : V \to S$, taka że dla każdego $u \in S$ mamy $|\phi^{-1}(u)| \leqslant L(u)$.

Cel: Zminimalizować $\max_{v \in V} \mathrm{d}(v, \phi(v))$.



Historia problemu oraz główny wynik

Poprzednio:

- Nawet bez przepustowości problem k-center jest NP-trudny. Istnieje 2-aproksymacja ('85) oraz wiadomo że nie da się otrzymać $\rho = (2 - \epsilon)$.
- Dla identycznych przepustowości istnieje 6-aproksymacja (Khuller and Sussman '96).

C., Hajiaghayi, Khuller 2012

Algorytm ze stałym współczynnikiem aproksymacji dla dowolnych przepustowości.

Narzędzia: LP + kombinatoryczna metoda zaokrąglania.

Część II: Algorytmy parametryzowane

Algorytmy parametryzowane

- Instancja wyposażona jest w dodatkową wartość — parametr (ozn. k).
- Parametr ma za zadanie odzwierciedlać trudność instancji.

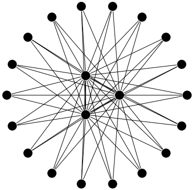
Definicja

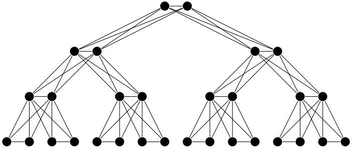
Algorytm **parametryzowany** rozwiązuje problem (dokładnie) w czasie $f(k)\mathrm{poly}(n)$.

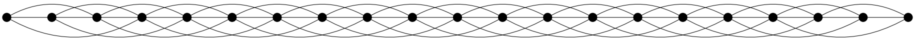
Przykładowo algorytm o złożoności $2^k n$ może być użyteczny dla $k = 20$, $n = 10^4$.

Czym jest szerokość drzewiasta?

- Szerokość drzewiasta (ang. treewidth) - parametr odzwierciedlający „trudność” grafu; mierzy stopień podobieństwa do drzewa.

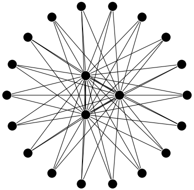


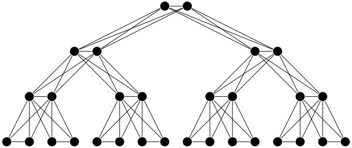


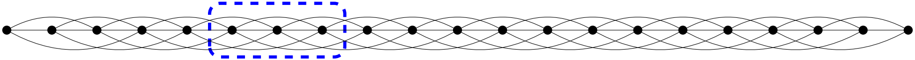


Czym jest szerokość drzewiasta?

- Szerokość drzewiasta (ang. treewidth) - parametr odzwierciedlający „trudność” grafu; mierzy stopień podobieństwa do drzewa.

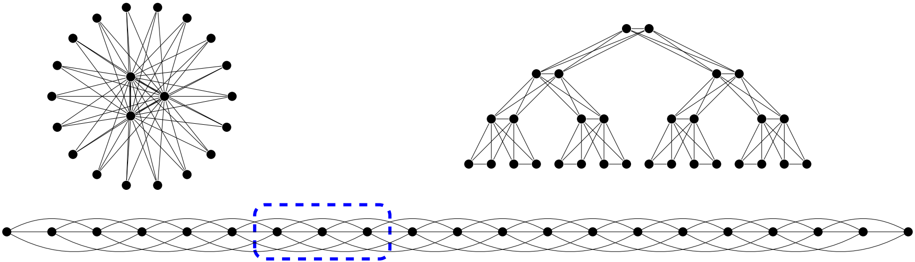






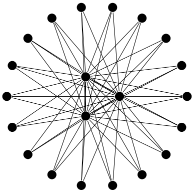
Czym jest szerokość drzewiasta?

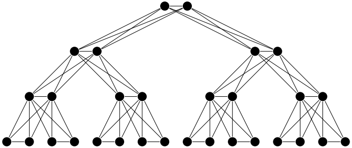
- Szerokość drzewiasta (ang. treewidth) - parametr odzwierciedlający „trudność” grafu; mierzy stopień podobieństwa do drzewa.

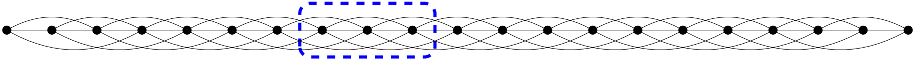


Czym jest szerokość drzewiasta?

- Szerokość drzewiasta (ang. treewidth) - parametr odzwierciedlający „trudność" grafu; mierzy stopień podobieństwa do drzewa.

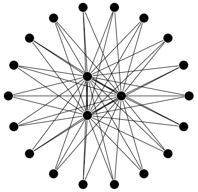


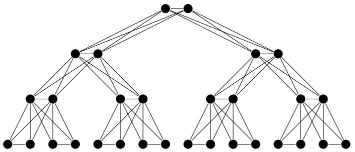


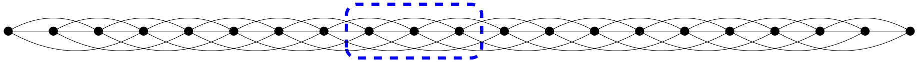


Czym jest szerokość drzewiasta?

- Szerokość drzewiasta (ang. treewidth) - parametr odzwierciedlający „trudność” grafu; mierzy stopień podobieństwa do drzewa.

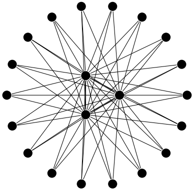


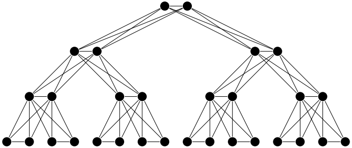


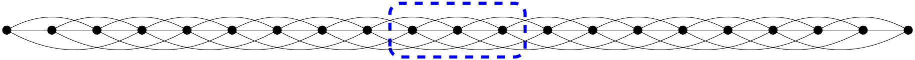


Czym jest szerokość drzewiasta?

- Szerokość drzewiasta (ang. treewidth) - parametr odzwierciedlający „trudność” grafu; mierzy stopień podobieństwa do drzewa.

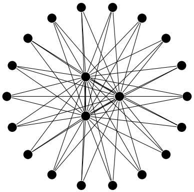


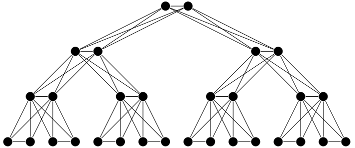


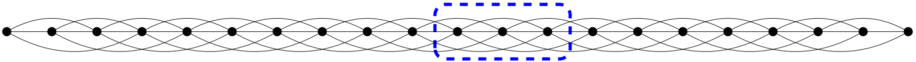


Czym jest szerokość drzewiasta?

- Szerokość drzewiasta (ang. treewidth) - parametr odzwierciedlający „trudność” grafu; mierzy stopień podobieństwa do drzewa.

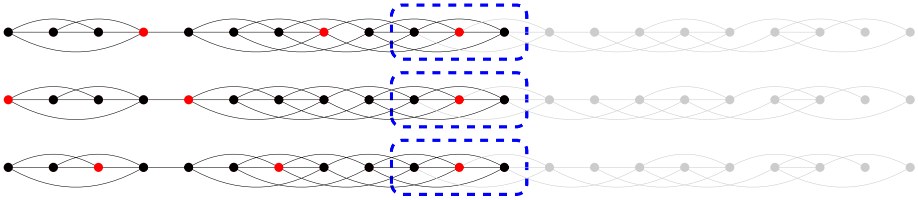






Znany algorytm dla zbioru niezależnego

- Problemu najliczniejszego zbioru niezależnego.
- Równoważność częściowych rozwiązań pozwala na rozwiązanie przez programowanie dynamiczne w czasie $2^t \mathrm{poly}(n)$.



Problemy lokalne i globalne

- Dla wielu „lokalnych" problemów istnieją algorytmy $c^t \mathrm{poly}(n)$: zbiór niezależny (2^t), zbiór dominujący (3^t), zliczanie skojarzeń (2^t).
- Jednakże dla problemów z warunkiem globalnym (np. spójność) znano jedynie algorytmy $c^{t \log t} \mathrm{poly}(n)$: najdłuższa ścieżka (ścieżka Hamiltona), drzewo Steinera, zbiór rozcyklający (ang. FVS), spójne pokrycie wierzchołkowe, spójny zbiór dominujący, pokrycie najmniejszą liczbą cykli, pakowanie cykli.

Technika „tnij i zliczaj”

C., Nederlof, Pilipczuk, Pilipczuk, van Rooij, Wojtaszczyk 2011

Opracowaliśmy technikę **„tnij i zliczaj”**, która pozwala na uzyskanie algorytmów Monte Carlo o złożoności $c^t\mathrm{poly}(n)$, dla problemów gdzie liczba spójnych składowych jest minimalizowana:

- najdłuższa ścieżka,
- drzewo Steinera,
- spójne pokrycie wierzchołkowe,
- spójny zbiór dominujący,
- pokrycie najmniejszą liczbą cykli,
- zbiór rozcyklający,
- ...

Niewielkie stałe i ograniczenia dolne

Problem	złożoność	ograniczenie dolne (SETH)
drzewo Steinera	3^t	$(3-\epsilon)^t$
zbiór rozcyklający	3^t	$(3-\epsilon)^t$
spójne pokrycie wierzchołkowe	3^t	$(3-\epsilon)^t$
spójny zbiór dominujący	4^t	$(4-\epsilon)^t$
spójny zbiór rozcyklający	4^t	$(4-\epsilon)^t$
(skier.) najdłuższa ścieżka	(6^t) 4^t	
(skier.) najmniejsze pokrycie cyklowe	(6^t) 4^t	
...		

Uzupełniający wynik

Maksymalizacja liczby spójnych składowych jest trudna.

Twierdzenie

Przy założeniu hipotezy ETH nie istnieje algorytm o złożoności $2^{o(t \log t)}\mathrm{poly}(n)$ dla problemu pakowania cykli (i innych).

Następstwa uzyskanych wyników

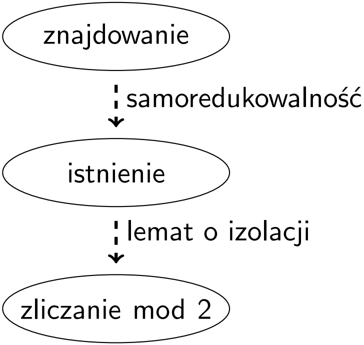
Szybsze algorytmy parametryzowane wielkością rozwiązania:

- $3.83^k \mathrm{poly}(n) \to 3^k \mathrm{poly}(n)$ dla zbioru rozcyklającego,
- $2.49^k \mathrm{poly}(n) \to 2^k \mathrm{poly}(n)$ dla spójnego pokrycia wierzchołkowego,
- $46.2^k \mathrm{poly}(n) \to 3^k \mathrm{poly}(n)$ dla spójnego zbioru rozcyklającego.

Następstwa uzyskanych wyników

- Szybsze algorytmy parametryzowane, wykładnicze i aproksymacyjne w grafach planarnych i z zabronionym minorem.
- Cykl Hamiltona w grafach kubicznych: $O(1.251^n) \rightarrow O(1.201^n)$.

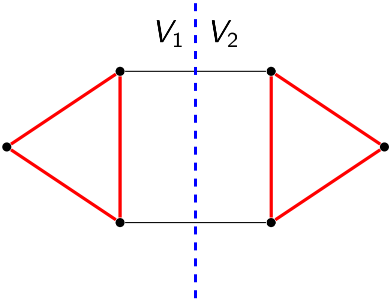
Technika „tnij i zliczaj”

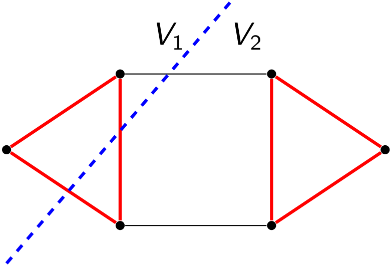


- Nowy cel: znaleźć #cykli Hamiltona modulo 2.

Technika „tnij i zliczaj”

- Przez cięcie (V_1, V_2) oznaczamy podział $V = V_1 \uplus V_2$.
- Pokrycie cyklowe $X \subseteq E$ nazwiemy *zgodnym* z cięciem (V_1, V_2) wtw $X \cap E(V_1, V_2) = \emptyset$.



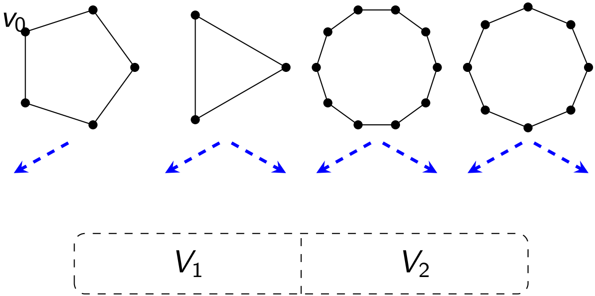


Technika „tnij i zliczaj”

Główny pomysł

Zamiast zliczać cykle Hamiltona będziemy zliczać pary (pokrycie cyklowe, zgodne cięcie). Takie obiekty się łatwo zlicza standardowymi technikami.

- Pokrycie o a cyklach jest zgodne z dokładnie 2^{a-1} cięciami.



Pozostałe zainteresowania

- Dokładne algorytmy wykładnicze.
- Teoria grafów.
- Skojarzenia.
- Algorytmy online.

Pytania

Dziękuję za uwagę.