

Albo szybko, albo dokładnie. Opowieść o algorytmach aproksymacyjnych

Jakub Sokołowski

XII Ogólnopolska Matematyczna Konferencja Studentów OMatKo!!!
Politechnika Wrocławska
5-7 grudnia 2025

Problemy optymalizacyjne

- Mamy dany zbiór **rozwiązań dopuszczalnych** oraz **funkcję celu**
- Chcemy wybrać rozwiązanie, które minimalizuje/maksymalizuje funkcję celu
- Dla niektórych problemów znamy efektywne (wielomianowe) algorytmy
- Dla pewnych klas problemów takie algorytmy (raczej) nie istnieją
- Pomysł: pogorszyć dokładność rozwiązania na rzecz szybkości

Algorytm aproksymacyjny

Definicja

Algorytm A nazywamy **algorytmem α -aproksymacyjnym**, jeśli spełnia on następujące warunki:

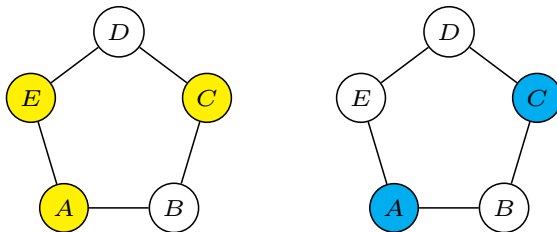
- działa w czasie wielomianowym,
- zawsze zwraca rozwiązanie dopuszczalne,
- dla każdego danych wejściowych x zachodzi $A(x) \leq \alpha \cdot OPT(x)$
(dla problemów maksymalizacji $A(x) \geq \alpha \cdot OPT(x)$)

Uwaga: dla problemów minimalizacji $\alpha > 1$, zaś dla problemów maksymalizacji $\alpha < 1$

Problem pokrycia wierzchołkowego

Definicja

Niech $G = (V, E)$ będzie grafem nieskierowanym. **Pokryciem wierzchołkowym** grafu G nazywamy dowolny podzbiór $C \subseteq V$, taki że każda krawędź ma przynajmniej jeden koniec w zbiorze C .



Żółty zbiór wierzchołków stanowi pokrycie wierzchołkowe. Niebieski zbiór nie jest pokryciem – nie pokrywa krawędzi $\{D, E\}$.

Problem pokrycia wierzchołkowego

- Mamy dany graf nieskierowany $G = (V, E)$
- Szukamy pokrycia wierzchołkowego o **minimalnej liczbie wierzchołków**
- Nie znamy żadnego algorytmu wielomianowego dla tego problemu
- Istnieje jednak algorytm 2-aproxymacyjny

Algorytm 2-aproksymacyjny

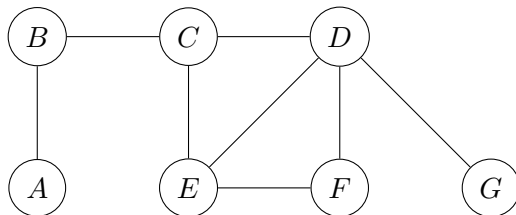
Dane wejściowe: Graf $G = (V, E)$

Wynik: Pokrycie wierzchołkowe C

```
1  $C = \emptyset$ ;  
2 while  $E \neq \emptyset$  do  
3   |   wybierz dowolną krawędź  $e = \{u, v\}$  z  $E$ ;  
4   |    $C = C \cup \{u, v\}$ ;  
5   |   usuń z  $E$  wszystkie krawędzie incydentne z  $u$  lub  $v$ ;  
6 end  
7 return  $C$ 
```

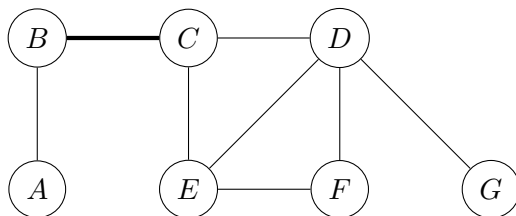
Czas działania: $O(|V| + |E|)$

Przykład



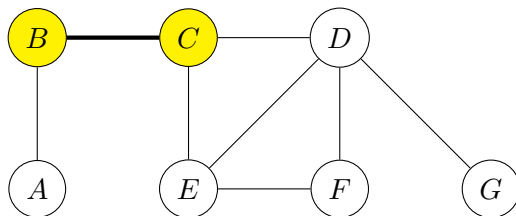
Przeanalizujemy działanie algorytmu na przykładzie powyższego grafu.

Przykład



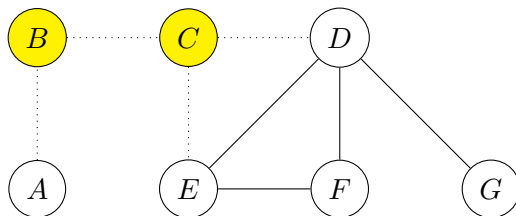
Wybieramy krawędź $\{B, C\}$.

Przykład



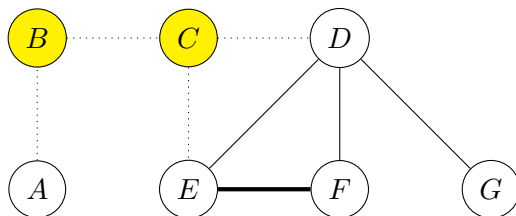
Wierzchołki B i C dodajemy do pokrycia.

Przykład



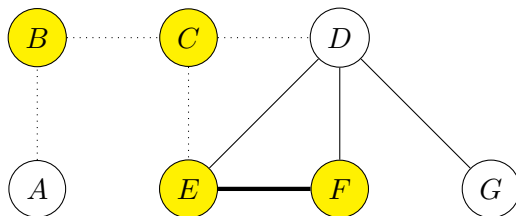
Usuwamy krawędzie incydentne z B lub C .

Przykład



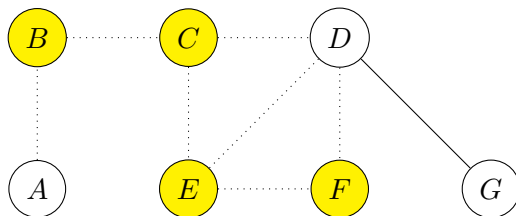
Wybieramy krawędź $\{E, F\}$.

Przykład



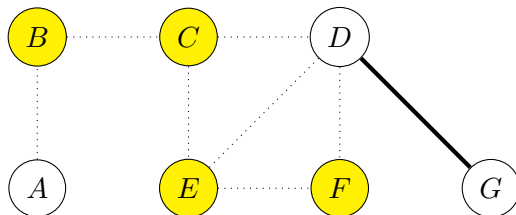
Wierzchołki E i F dodajemy do pokrycia.

Przykład



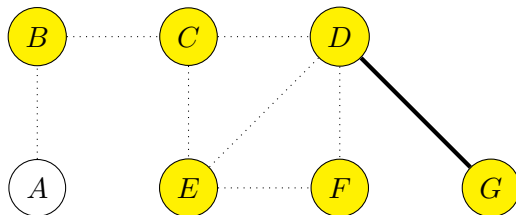
Usuwamy krawędzie incydentne z E lub F .

Przykład



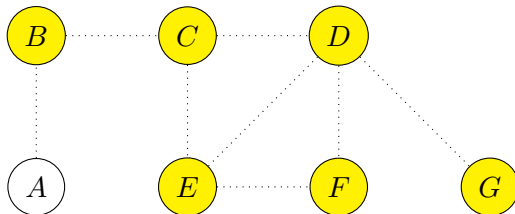
Wybieramy krawędź $\{D, G\}$.

Przykład



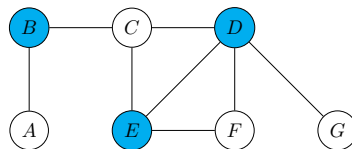
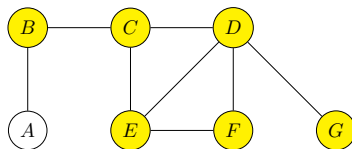
Wierzchołki D i G dodajemy do pokrycia.

Przykład



Usuwanie krawędzi incydentne z D lub G . Algorytm kończy działanie.

Przykład



Optymalne pokrycie zawiera tylko 3 wierzchołki.

Dowód poprawności

Twierdzenie

Przedstawiony algorytm zawsze zwraca pokrycie wierzchołkowe, które zawiera co najwyżej 2 razy więcej wierzchołków niż pokrycie optymalne.

Dowód

- Pętla jest wykonywana tak długo, jak w grafie są niepokryte krawędzie, więc utworzony zbiór jest pokryciem wierzchołkowym.
- Niech C^* – pokrycie optymalne, C – pokrycie znalezione przez algorytm, A – zbiór krawędzi wybranych przez algorytm.
- Krawędzie z A nie mają wspólnych wierzchołków, więc $|C^*| \geq |A|$.
- Każda krawędź dodaje do pokrycia C dwa wierzchołki: $|C| = 2|A|$.
- Stąd $|C| = 2|A| \leq 2|C^*|$. □

PTAS

Definicja

Algorytm **wielomianowym schematem aproksymacyjnym**, jeśli dla dowolnych danych wejściowych oraz dowolnego $\alpha > 1$ ($0 < \alpha < 1$) zwraca on rozwiązanie co najwyżej α razy gorsze od optymalnego oraz działa w czasie wielomianowym dla ustalonego α .

Przykładem problemu, który posiada wielomianowy schemat aproksymacyjny jest **problem plecakowy**.

Problem plecakowy

Dane wejściowe:

- wartości kolejnych przedmiotów: v_i , dla $i = 1, \dots, n$
- wagi kolejnych przedmiotów: w_i , dla $i = 1, \dots, n$
- pojemność plecaka: W

Cel: szukamy podzbioru $P \subseteq \{1, \dots, n\}$

- chcemy zmaksymalizować łączną wartość

$$\sum_{i \in P} v_i \rightarrow \max$$

- nie możemy przekroczyć pojemności plecaka

$$\sum_{i \in P} w_i \leq W$$

Algorytm dokładny

- Zamiast rozwiązywać cały problem od razu, będziemy rozwiązywać coraz większe podproblemy
- $T[i, j]$ – minimalna waga konieczna do osiągnięcia wartości j , gdy mamy do dyspozycji pierwsze i przedmiotów (lub ∞ , gdy się nie da)
- Korzystamy z zależności rekurencyjnej:

$$T[i, j] = \min\{T[i - 1, j], w_i + T[i - 1, j - v_i]\}$$

- Rozwiązanie problemu można odczytać z ostatniego wiersza tabeli

Przykład

Nr przedmiotu i	1	2	3
Wartość p_i	2 \$	1 \$	3 \$
Waga w_i	4 kg	2 kg	5 kg

$$W_{max} = 10 \text{ kg}$$

	0	1	2	3	4	5	6
0	0	∞	∞	∞	∞	∞	∞
1	0	4	4	∞	∞	∞	∞
2	0	2	4	6	∞	∞	∞
3	0	2	4	5	7	9	11

$$T[2, 1] = \min\{T[1, 1], w_2 + T[1, 0]\} = \min\{4, 2 + 0\} = 2$$

Rozwiązanie optymalne: weź przedmioty 1 i 3 (wartość 5, waga 9).

Przykład

Nr przedmiotu i	1	2	3
Wartość p_i	2 \$	1 \$	3 \$
Waga w_i	4 kg	2 kg	5 kg

$$W_{max} = 10 \text{ kg}$$

	0	1	2	3	4	5	6
0	0	∞	∞	∞	∞	∞	∞
1	0	4	4	∞	∞	∞	∞
2	0	2	4	6	∞	∞	∞
3	0	2	4	5	7	9	11

Czas działania: $O(n \sum v_i) = O(n^2 v_{max}) = O(n^2 2^{\log v_{max}})$

Schemat aproksymacyjny

- Ustalamy dokładność aproksymacji $\varepsilon > 0$
- Wyliczamy parametr zaokrąglania $b = \frac{\varepsilon v_{max}}{n}$
- Wartości przedmiotów będziemy zaokrąglać:

$$v'_i = \left\lfloor \frac{v_i}{b} \right\rfloor, \quad \text{stąd} \quad v'_{max} = \left\lfloor \frac{n}{\varepsilon} \right\rfloor$$

- Używamy algorytmu dokładnego do problemu z wagami v'_i
- Czas działania będzie wynosił $O(n^2 \left\lfloor \frac{n}{\varepsilon} \right\rfloor) = O(\frac{n^3}{\varepsilon})$ – wielomian dla ustalonego ε
- Traktujemy uzyskany zbiór przedmiotów jako rozwiązanie wyjściowego problemu

Schemat aproksymacyjny

Twierdzenie

Wynik zwrócony przez opisany wyżej algorytm stanowi co najmniej $(1 - \varepsilon)$ wartości rozwiązania optymalnego.

Dowód

- Niech D – optimum dla wag v_i , A – optimum dla wag v'_i
- Przyjęliśmy $v'_i = \lfloor \frac{v_i}{b} \rfloor$
- Gdybyśmy wzięli $v''_i = \lfloor \frac{v_i}{b} \rfloor b$, to uzyskany zbiór byłby taki sam
- Zachodzi $v_i - b \leq v''_i \leq v_i$

Schemat aproksymacyjny

- Stąd poniższy ciąg nierówności:

$$\sum_{i \in A} v_i \geq$$

$$\text{bo } v_i \geq v_i''$$

$$\sum_{i \in A} v_i'' \geq$$

$$\text{bo } A \text{ jest optimum dla wag } v_i''$$

$$\sum_{i \in D} v_i'' \geq$$

$$\text{bo } v_i'' \geq v_i - b$$

$$\sum_{i \in D} (v_i - b) \geq$$

$$\text{bo } |D| \leq n$$

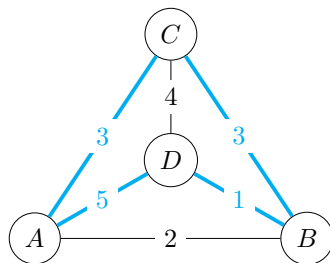
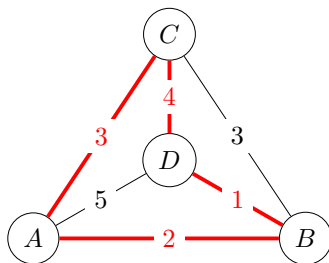
$$\sum_{i \in D} v_i - nb = \sum_{i \in D} v_i - \varepsilon v_{max} \geq$$

$$\text{bo } v_{max} \leq \sum_{i \in D} v_i$$

$$\sum_{i \in D} v_i - \varepsilon \sum_{i \in D} v_i = (1 - \varepsilon) \sum_{i \in D} v_i \quad \square$$

Problem komiwojażera

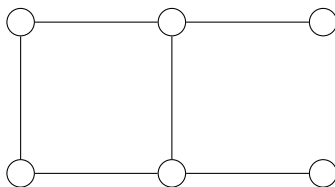
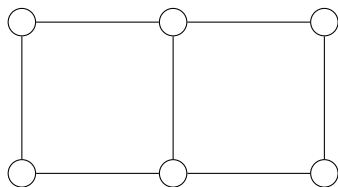
Mamy dany graf pełny wraz z wagami na jego krawędziach. Szukamy cyklu Hamiltona o najmniejszej sumie wag.



Czerwony cykl w K_4 ma łączny koszt 10, niebieski ma koszt 12.

TSP i aproksymacje

- TSP jest blisko powiązany z problemem cyklu Hamiltona



- Problem cyklu Hamiltona jest NP -zupełny, więc podejrzewamy, że nie można go rozwiązać w czasie wielomianowym
- Można pokazać, że jakikolwiek algorytm aproksymacyjny dla TSP pozwoliłby rozwiązać problem cyklu Hamiltona

TSP i aproksymacje

Twierdzenie

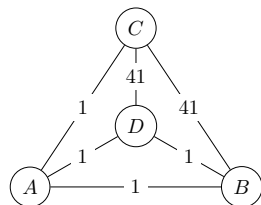
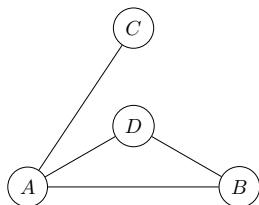
Jeśli dla problemu cyklu Hamiltona nie istnieje algorytm wielomianowy, to dla problemu komiwożacza nie istnieje algorytm α -aproksymacyjny dla żadnego $\alpha > 1$.

Dowód

- Załóżmy, że dla pewnego $\alpha > 1$ istnieje algorytm α -aproksymacyjny dla problemu komiwożacza
- Pokażemy, że z pomocą tego algorytmu moglibyśmy znajdować cykle Hamiltona w czasie wielomianowym

Dowód – c.d.

- Niech $G = (V, E)$ – graf, w którym chcemy znaleźć cykl Hamiltona
- Tworzymy graf **pełny** $G' = (V, E')$
- Krawędziom z pierwotnego grafu nadajemy wagę 1
- Nowym krawędziom nadajemy wagę $\alpha|V| + 1$



Przykład dla $\alpha = 10$

Dowód – c.d.

- Jeśli graf G zawiera cykl Hamiltona, to w G' na pewno istnieje cykl o koszcie $|V|$
- Wtedy algorytm aproksymacyjny na pewno zwróci cykl o koszcie nie większym niż $\alpha|V|$
- Jeśli graf G nie zawiera cyklu Hamiltona, to każdy cykl w G' ma koszt co najmniej

$$(\alpha|V| + 1) + (|V| - 1) = \alpha|V| + |V| > \alpha|V|$$

- Sprawdzając, czy zwrócony koszt jest większy czy mniejszy od $\alpha|V|$ moglibyśmy w czasie wielomianowym odszukać cykl Hamiltona. $\square$

Podsumowanie

- Poznaliśmy tylko kilka podstawowych pojęć z zakresu algorytmów aproksymacyjnych
- Jest to bardzo rozległa i intensywnie badana dziedzina algorytmiki
- Istnieje wiele ciekawych technik projektowania algorytmów aproksymacyjnych
- Programowanie liniowe, przeszukiwanie lokalne, randomizacja czy algorytmy zachłanne to niektóre z tych technik

Bibliografia

- [1] Thomas H. Cormen i in. *Wprowadzenie do algorytmów*. Wydawnictwo Naukowe PWN SA, 2012.
- [2] Jon Kleinberg i Éva Tardos. *Algorithm Design*. Pearson Education, Inc., 2006.
- [3] Christos H. Papadimitriou. *Złożoność obliczeniowa*. Wydawnictwo Helion, 2012.
- [4] David P. Williamson i David B. Shmoys. *The Design of Approximation Algorithms*. Cambridge University Press, 2011.
- [5] Kevin Wayne. *11. Approximation Algorithms*. 2005. URL: <https://www.cs.princeton.edu/~wayne/kleinberg-tardos/pdf/11ApproximationAlgorithms.pdf> (term. wiz. 27.11.2025).

Dziękuję za uwagę!