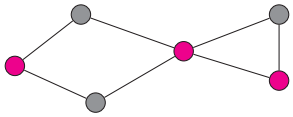
Programowanie półcałkowitoliczbowe

*Krzysztof KILJAN**

* student, Instytut Informatyki, Wydział Matematyki, Informatyki i Mechaniki, Uniwersytet Warszawski

W tym artykule zmierzymy się z problemem pokrycia wierzchołkowego (*Vertex Cover*). Poruszymy kilka kwestii na pograniczu optymalizacji liniowej, złożoności obliczeniowej oraz algorytmów parametryzowanych. Większość Czytelników może znać ten problem z różnych źródeł (między innymi pojawiał się on już w *Delcie* przed laty), ale na wszelki wypadek na początek przypomnę definicję.

Definicja 1. *Pokryciem wierzchołkowym grafu $G = (V, E)$ nazwiemy taki zbiór wierzchołków, że dla każdej krawędzi $(u, v) \in E$, występującej w grafie, przynajmniej jeden z wierzchołków u, v należy do pokrycia.*



Rys. 1. Graf z zaznaczonym kolorem jego minimalnym pokryciem wierzchołkowym

Minimalne pokrycie wierzchołkowe to, oczywiście, takie, które zawiera najmniejszą możliwą liczbę wierzchołków.

Z teorii złożoności wiemy już, że nasz problem jest NP-zupełny. Znaczy to, że jeśli potrafilibyśmy go szybko rozwiązać, to poprzez pewne redukcje umielibyśmy również szybko rozwiązać wszystkie problemy z klasy NP. W tym przypadku przez „szybko" rozumiemy rozwiązanie działające w czasie wielomianowym, czyli $O(n^c)$ dla pewnej stałej c. Jedna ze słynniejszych hipotez w informatyce mówi, że P $\neq$ NP, czyli że klasa problemów rozwiązywalnych w deterministycznym czasie wielomianowym jest mniejsza od tej rozpoznawanej w niedeterministycznym (czyli nie zachodzi NP $\subseteq$ P, zawieranie P $\subseteq$ NP jest oczywiste). Przy założeniu, że hipoteza ta jest prawdziwa, problemów NP-zupełnych nie da się rozwiązywać w deterministycznym czasie wielomianowym.

Jak możemy znaleźć minimalne pokrycie wierzchołkowe?

Pierwszym pomysłem, jaki możemy mieć, jest przejrzenie wszystkich podzbiorów wierzchołków naszego grafu i wybranie takiego, który jest pokryciem wierzchołkowym i ma najmniejszy rozmiar. Takich podzbiorów jest 2^n (gdzie n to liczba wierzchołków w naszym grafie).

Możemy też się zabrać za to inaczej – okazuje się, że bardzo często w problemach kombinatorycznych z pomocą przychodzi programowanie liniowe.

Dla niezaznajomionych z terminem programowania liniowego – jest to forma przedstawienia problemu za pomocą zbioru zmiennych, równań i nierówności liniowych na nich oraz pewnej funkcji liniowej tych zmiennych, którą minimalizujemy lub maksymalizujemy.

Tak też jest i w naszym przypadku. Dla każdego wierzchołka v w naszym grafie stworzymy zmienną całkowitą x_v oznaczającą, czy bierzemy v do naszego pokrycia, czy też nie. Oczywiście, będziemy chcieli zminimalizować liczbę wziętych wierzchołków. Wymaganie co do pokrycia krawędzi wyrazimy, tworząc dla każdej krawędzi (u, v) nierówność $x_u + x_v \geqslant 1$.

Podsumowując powyższe, dostajemy następujący program liniowy całkowitoliczbowy:

Program liniowy całkowitoliczbowy to taki, dla którego zmienne przyjmują wartości tylko ze zbioru liczb całkowitych.

$$\begin{aligned} &\text{Minimalizuj} \quad \textstyle\sum_{v \in V} x_v \\ &\text{z zachowaniem warunków: } x_u + x_v \geqslant 1 \quad \text{dla każdego } (u, v) \in E, \\ &\qquad\qquad\qquad\qquad x_w \in \{0, 1\} \quad \text{i dla każdego } w \in V. \end{aligned}$$

Potrafimy wyrazić więc problem NP-zupełny za pomocą ILP (*Integer Linear Programming* – programowanie całkowitoliczbowe). Oznacza to dokładnie, że pokazaliśmy w tym momencie NP-trudność problemu ILP. Tym samym nie oczekujemy, że dla ILP istnieje jakikolwiek algorytm działający w czasie wielomianowym.

Co się stanie, gdy zapomnimy o warunkach całkowitości dla naszych zmiennych?

Dostaniemy zwykły program liniowy, a przecież do rozwiązywania takiego istnieją już algorytmy wielomianowe! Pytanie, czy wyniki tego programu mogą się nam do czegoś przydać. Mogłoby się wydawać, że nie – przykładowo dla cyklu o 3 wierzchołkach optymalnym rozwiązaniem będzie dać $x_1 = x_2 = x_3 = \frac{1}{2}$, które nam za wiele nie mówi.

Okazuje się jednak, że ta forma programu liniowego może być dla nas wciąż użyteczna! W dziedzinie algorytmów parametryzowanych istnieje dział zwany *kernelizacją*. Wykorzystamy rozwiązanie naszego LP (*Linear Programming*) właśnie w procesie kernelizacji. Najpierw jednak może warto wspomnieć, co kryje się pod tajemniczymi hasłami użytymi powyżej.

W skrócie algorytmy parametryzowane to takie, w których wybieramy sobie parametr opisujący w pewien sposób instancję naszego problemu, a następnie względem tego parametru będziemy optymalizować prędkość działania naszego algorytmu. Parametry te mogą w najróżniejszy sposób opisywać problem, z którym się mierzymy. Oczywiście, podstawowym parametrem może być wielkość naszego problemu wejściowego, jednak dużo z takiego parametru raczej nie wyciągniemy. Sensowniejszym przykładem dla problemów grafowych może być, na przykład, maksymalny stopień wierzchołka w rozważanym grafie. W ogólności, do każdego problemu da się wymyślić dużo różnych parametrów, ale wybranie tego właściwego może okazać się sporym wyzwaniem.

Drugie tajemnicze słowo, czyli „kernelizacja", można przetłumaczyć jako znajdowanie jądra problemu – chcemy w szybkim czasie z jakiejś instancji problemu NP-zupełnego wydobyć jego najtrudniejszą obliczeniowo część. Dzięki temu bardzo często jesteśmy w stanie mocno ograniczyć rozmiary problemu, z którym się borykamy, co znacząco przyspieszy nasze obliczenia (zostanie nam dużo mniej zadań, które musimy przetworzyć w czasie wykładniczym). Aby rozjaśnić te pojęcia, za chwilę zilustrujemy je na przykładzie pokrycia wierzchołkowego.

Skupimy się tym razem na wersji decyzyjnej problemu pokrycia wierzchołkowego – czy dla danego grafu G istnieje pokrycie zawierające co najwyżej k wierzchołków. Jako parametru użyjemy rozmiaru poszukiwanego rozwiązania (czyli liczby wierzchołków k w pokryciu).

Używając rozwiązania naszego programu liniowego

Minimalizuj $\sum_{v\in V} x_v$ z zachowaniem warunku

$$x_u + x_v \geqslant 1 \text{ dla każdego } (u,v) \in E,$$

podzielimy wierzchołki na 3 zbiory:

- $V_0 = \{v : 0 \leqslant x_v < \frac{1}{2}\}$
- $V_{1/2} = \{v : x_v = \frac{1}{2}\}$
- $V_1 = \{v : \frac{1}{2} < x_v \leqslant 1\}$.

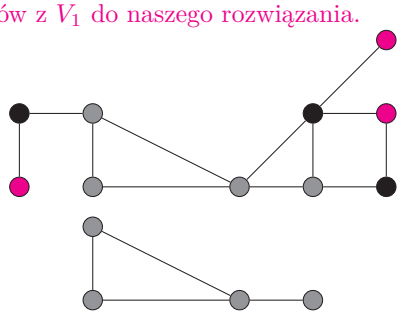
W tym momencie jasne się powinno stać, skąd w tytule wzięła się nazwa półcałkowite – przekształcamy rozwiązanie LP tak, żeby oprócz wartości całkowitych 0 i 1 osiągało ono trzecią wartość $\frac{1}{2}$. Okazuje się, że taka postać rozwiązania jest niezwykle użyteczna w znajdowaniu jądra. Pierwszym spostrzeżeniem, jakie możemy poczynić, jest to, że jeśli spojrzymy na którykolwiek wierzchołek $v \in V_0$, to musi mieć on sąsiada należącego do zbioru V_1. Można jednak udowodnić dużo ciekawsze twierdzenie.

Twierdzenie 2 (Nemhausera–Trottera)**.** *Istnieje takie minimalne pokrycie wierzchołkowe S w grafie G, że:*

$$V_1 \subseteq S \subseteq V_1 \cup V_{1/2}.$$

Dowód powyższego twierdzenia, oczywiście, pozostawiamy jako ćwiczenie dla Czytelnika. Skorzystamy z niego podczas następującej redukcji naszego problemu.

Mamy dany graf G oraz liczbę k (rozmiar pokrycia, który sprawdzamy). Jeśli $\sum_{v\in V} x_v > k$, to zwracamy odpowiedź nie. W przeciwnym przypadku zmniejszamy nasz graf poprzez wyrzucenie z niego wierzchołków należących do $V_0 \cup V_1$ oraz zmniejszenie wielkości poszukiwanego pokrycia w reszcie grafu do $k - |V_1|$. Odpowiada to zachłannemu wzięciu wszystkich wierzchołków z V_1 do naszego rozwiązania.



Rys. 2. U góry graf z odpowiednio zaznaczonymi kolorem wierzchołkami z V_0, szaro z $V_{1/2}$ i czarno z V_1. Na dole ten sam graf po wykonaniu na nim powyższej redukcji

Pozostało wykazać, że redukcja ta jest bezpieczna oraz sprawdzić jakiej wielkości jest jądro po jej zastosowaniu. Dalej przez (G', k') będę oznaczał instancję problemu, którą dostaniemy po zastosowaniu redukcji. Aby wykazać, że jest ona bezpieczna, wykażemy, że w (G', k') możemy znaleźć rozwiązanie wtedy i tylko wtedy, gdy mogliśmy je znaleźć w (G, k).

Zacznijmy od wykazania, że krok stwierdzający odpowiedź *nie* jest poprawny. Wiadomo, że wynik LP jest zawsze niewiększy niż ILP. Z tego też powodu, jeśli wynik LP będzie większy niż k, czyli stwierdzimy, że się nie da znaleźć rozwiązania w wersji bez warunku o całkowitości liczb, to tym bardziej z tym warunkiem by się nie dało.

Druga część bezpieczeństwa nie jest wiele trudniejsza do wykazania. Najpierw uzasadnijmy implikację

$$(G', k') \ tak \to (G, k) \ tak.$$

Jeśli S' to pokrycie grafu G', to na mocy wcześniejszej obserwacji, jeśli dołożymy wierzchołki z $V_1 \cup V_0$ do grafu, a V_1 do pokrycia, to dostaniemy dokładnie pokrycie grafu G co najwyżej o rozmiarze k.

W przeciwną stronę: aby wykazać

$$(G, k) \ tak \to (G', k') \ tak,$$

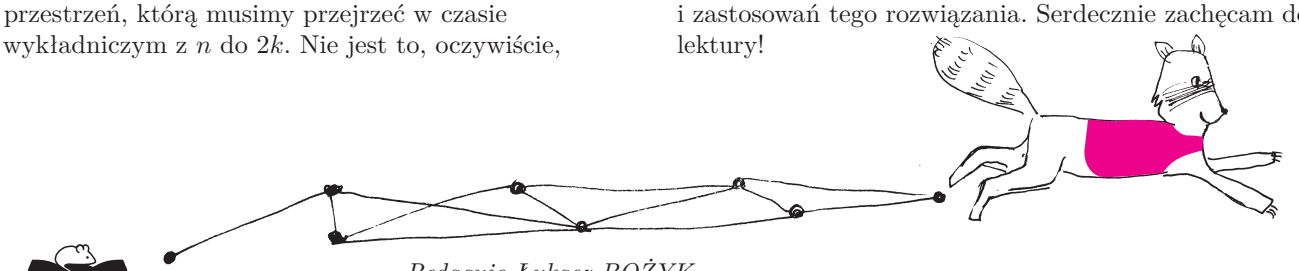
korzystamy z twierdzenia Nemhausera–Trottera przytoczonego powyżej. Weźmiemy wtedy S', będące częścią wspólną pokrycia S dla całego grafu, oraz $V_{1/2}$, aby otrzymać pokrycie dla G' o rozmiarze co najwyżej k'.

Ostatnią rzeczą jest wielkość jądra, które otrzymamy. Okazuje się, że rozmiar naszego nowego grafu G' będzie wynosił co najwyżej $2k$. Wystarczy zauważyć, że jest on tym samym, co liczba wierzchołków „połowicznych”, co z kolei wynosi nie więcej niż podwojona wielkość rozwiązania naszego LP, które musi być mniejsze niż k, gdyż w przeciwnym przypadku dostalibyśmy już odpowiedź *nie*. Powyższe zdanie matematycznie zapiszemy w postaci

$$|V(G')| = |V_{1/2}| = \sum_{v \in V_{1/2}} 2x_v \leqslant 2 \sum_{v \in V} x_v \leqslant 2k.$$

W ten oto prosty sposób udało nam się ograniczyć przestrzeń, którą musimy przejrzeć w czasie wykładniczym z n do $2k$. Nie jest to, oczywiście, jedyna redukcja, którą można zastosować do tego problemu. Okazuje się, że sam proces redukcji można jeszcze przyspieszyć, sprowadzając nasze LP do problemu przepływu w grafie. Po więcej szczegółów na temat algorytmów parametryzowanych (jak i tego sprowadzenia) można zajrzeć do książki *Parameterized Algorithms*, która na użytek własny jest dostępna za darmo w internecie.

Oczywiście, nie jest to też jedyne zastosowanie programowania półcałkowitoliczbowego. W internecie można znaleźć pracę *Half-integrality, LP-branching and FPT Algorithms*, w której jest więcej szczegółów, jak i zastosowań tego rozwiązania. Serdecznie zachęcam do lektury!



Zadania

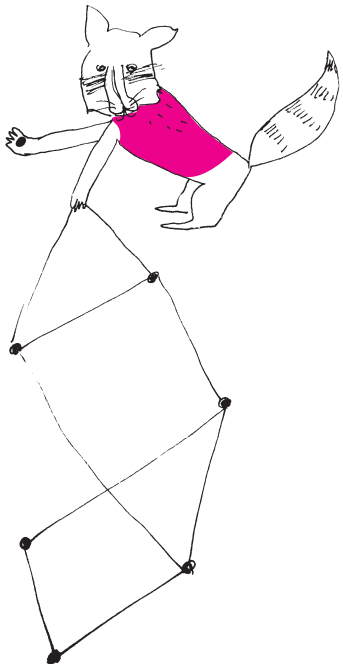
Redaguje Łukasz BOŻYK

M 1558. Czy istnieją liczby całkowite a, b, c o tej własności, że każdy z trójmianów kwadratowych

$$ax^2 + bx + c \quad \text{oraz} \quad (a+1)x^2 + (b+1)x + (c+1)$$

ma obydwa pierwiastki całkowite?
Rozwiązanie na str. 22

M 1559. Pierwsza ćwiartka płaszczyzny z kartezjańskim układem współrzędnych jest podzielona prostymi o równaniach $x = n$ oraz $y = n$ dla $n \in \{1, 2, 3, \ldots\}$ na kwadraty jednostkowe, zwane dalej *polami*. Czy można wyróżnić niektóre pola w taki sposób, że:
• każdy kwadrat o bokach całkowitej długości równoległych do osi układu i jednym z wierzchołków w punkcie $(0, 0)$ zawiera więcej pól wyróżnionych niż niewyróżnionych;
• każda prosta równoległa do prostej $y = x$ przecina wnętrze tylko skończenie wielu wyróżnionych pól?
Rozwiązanie na str. 11



M 1560. Wyznaczyć iloczyn długości wszystkich boków i przekątnych n-kąta foremnego wpisanego w okrąg o promieniu 1.
Rozwiązanie na str. 22

Przygotował Andrzej MAJHOFER

F 947. Doświadczenia z rozpraszaniem cząstek naładowanych na jądrach atomowych wskazują, że promień jądra R rośnie z liczbą A nukleonów jak $R = r_0 A^{1/3}$, gdzie $r_0 \approx 1,25 \cdot 10^{-15}\,\mathrm{m}$. Korzystając z zasady nieoznaczoności, oszacuj na tej podstawie średnią energię E_B wiązania nukleonu w jądrze. Masa nukleonu $M \approx 940\,\mathrm{MeV/c^2}$, a iloczyn $\hbar$, tj. stałej Plancka podzielonej przez 2π i prędkości światła c wynosi $\hbar c = 197 \cdot 10^{-15}\,\mathrm{MeV \cdot m}$.
Rozwiązanie na str. 18

F 948. Jony uranu U, znajdujące się w ciekłym cyrkonie $ZrSiO_4$, mieszają się z nim, podstawiając jony Zr, a jony ołowiu Pb nie reagują z cyrkonem i szybko dyfundują poza jego objętość. Po zestaleniu cyrkonu, dyfuzja praktycznie ustaje. W kryształach cyrkonu, znajdujących się w skale znalezionej w Zimbabwe, zmierzono, że stosunek liczby atomów ołowiu ^{206}Pb do liczby atomów uranu ^{238}U wynosi $x = 0,5$. Jaki jest wiek tej skały, jeśli w serii kolejnych rozpadów alfa i beta, ^{238}U rozpada się do ^{206}Pb z czasem połowicznego zaniku $t_{1/2} = 4,47 \cdot 10^9$ lat?
Rozwiązanie na str. 18