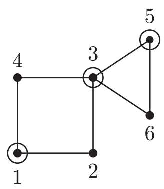
Algorytmy aproksymacyjne

Marek CYGAN[*]

Problemy, którymi zajmują się informatycy, możemy podzielić na kategorie w zależności od złożoności obliczeniowej. Niektóre zagadnienia umiemy rozwiązać w czasie liniowym, co oznacza, że aby znaleźć rozwiązanie, wystarczy wykonać liczbę operacji proporcjonalną do wielkości danych wejściowych. Szerszą grupę stanowią problemy, dla których znane są algorytmy wielomianowe (klasę takich problemów oznaczamy przez P). Istnieje jednakże wiele (często zasadniczych) problemów, dla których nie są znane algorytmy wielomianowe (bardzo istotną klasę takich problemów stanowią tzw. problemy NP-zupełne). Co więcej, panuje powszechne przekonanie, że algorytmy wielomianowe dla tych problemów w ogóle nie istnieją! Zadania, które będziemy chcieli rozwiązać w tym artykule, należą właśnie do kategorii tych trudnych problemów.

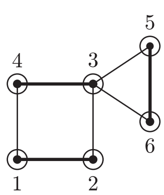
Zacznijmy od następującego zagadnienia. W grafie nieskierowanym $G = (V, E)$ pokryciem wierzchołkowym nazwiemy taki zbiór wierzchołków $V_0 \subseteq V$, że każda krawędź grafu G jest pokryta przez co najmniej jeden wierzchołek z V_0, tzn. co najmniej jeden z jej końców należy do V_0. Naszym celem jest znalezienie pokrycia wierzchołkowego o minimalnej liczności. Przykład 6-wierzchołkowego grafu wraz z najmniej licznym pokryciem wierzchołkowym $V_0 = \{1, 3, 5\}$ znajduje się na rysunku 1.



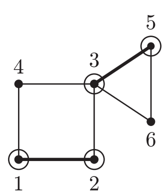
Rys. 1

Zgodnie ze wstępem, problem pokrycia wierzchołkowego należy do klasy tych trudnych (jest on NP-zupełny), więc zapewne nie uda nam się znaleźć algorytmu wielomianowego konstruującego optymalne pokrycie w tak krótkim artykule. Spróbujmy zatem skonstruować chociaż takie rozwiązanie, które nie będzie dużo gorsze od optymalnego. Powiemy, że nasze rozwiązanie będzie k-aproksymacją rozwiązania optymalnego, jeśli liczba wierzchołków przez nas wybranych będzie *dla każdego możliwego grafu* co najwyżej k razy większa od rozmiaru optymalnego pokrycia wierzchołkowego. Wartość k będziemy nazywać współczynnikiem aproksymacji, czyli stosunkiem rozmiaru naszego rozwiązania do rozmiaru rozwiązania optymalnego.

Wybierzmy dowolną krawędź grafu G; zgodnie z definicją, w optymalnym pokryciu wierzchołkowym co najmniej jeden z końców tej krawędzi musi zostać wybrany. Ponieważ nie szukamy optymalnego rozwiązania, możemy pozwolić sobie na to, żeby wybrać oba końce tej krawędzi do naszego zbioru V_0. Po zapamiętaniu tych dwóch wierzchołków możemy usunąć je z grafu, wraz ze wszystkimi krawędziami, które są z nimi incydentne. Następnie do zbioru V_0 wstawiamy oba końce dowolnej z pozostałych w grafie krawędzi, jednocześnie usuwając je z grafu wraz z krawędziami z nimi incydentnymi itd. Postępujemy w ten sposób, dopóki w grafie pozostaje jakakolwiek krawędź. Wyniki zastosowania tej procedury dla przykładowego grafu obrazują rysunki 2 (mniej udany wybór krawędzi powoduje zaliczenie wszystkich wierzchołków do V_0) i 3 (sprytniejszy wybór krawędzi prowadzi do $V_0 = \{1, 2, 3, 5\}$).



Rys. 2



Rys. 3

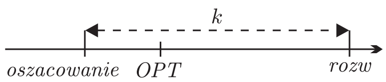
Wprost z zastosowanej metody konstrukcji możemy wywnioskować, że wynikowy zbiór V_0 będzie zawsze pokryciem wierzchołkowym G. Z przykładu wynika jednakże, że otrzymane przez nas rozwiązanie może być gorsze od optymalnego. Ale o ile gorsze? Zauważmy, że każdej z par wierzchołków dodanych do zbioru V_0 możemy przypisać jeden wierzchołek z pewnego optymalnego pokrycia wierzchołkowego (będzie to ten z końców łączącej je krawędzi, który należy do optymalnego rozwiązania). Argument ten pokazuje, że skonstruowany przez nas zbiór V_0 jest co najwyżej dwa razy większy od rozwiązania optymalnego, co oznacza, że skonstruowaliśmy algorytm 2-aproksymacyjny.

Tym razem udało nam się stosunkowo szybko wyznaczyć współczynnik aproksymacji rozwiązania, jednakże zazwyczaj nie jest to tak łatwą sprawą. Faktycznie, oszacowanie jakości rozwiązania aproksymacyjnego wymaga nie tylko rozpatrzenia *wszystkich* możliwych danych wejściowych algorytmu, ale, co nawet gorsze, umiejętności porównania jego działania z *rozwiązaniem optymalnym*, którego przecież nie znamy! (Dla ścisłości: jakieś rozwiązanie zazwyczaj znamy, ale jest to wówczas rozwiązanie wykładnicze i najczęściej nie pozwala wyciągnąć pożytecznych wniosków na temat struktury problemu). Aby lepiej zrozumieć, w jaki sposób udało nam się tym razem oszacować współczynnik aproksymacji, a zarazem aby wyciągnąć wnioski na przyszłość, spójrzmy na nasz algorytm

[*] Instytut Informatyki, Uniwersytet Warszawski

z innej strony. Zbiór krawędzi, które zostały w nim wybrane, tworzy tak zwane skojarzenie, co oznacza, że żadna para krawędzi nie ma wspólnego końca. W rozwiązaniu skorzystaliśmy z tego, że każde pokrycie wierzchołkowe musi mieć rozmiar nie mniejszy niż liczność dowolnego skojarzenia, gdyż z każdej krawędzi skojarzenia musi zostać wybrany co najmniej jeden wierzchołek. Nasze rozwiązanie ma rozmiar będący dwukrotnością rozmiaru pewnego skojarzenia, gdyż z każdej wybranej krawędzi do zbioru V_0 dodawaliśmy oba jej końce. W ten sposób współczynnik aproksymacji naszego algorytmu oszacowaliśmy w dwóch krokach:

- w pierwszym kroku znaleźliśmy obiekt (skojarzenie), którego rozmiar jest dolnym ograniczeniem rozmiaru optymalnego rozwiązania;
- w drugim kroku pokazaliśmy, że rozmiar skonstruowanego przez nas rozwiązania jest co najwyżej dwa razy większy od rozmiaru skojarzenia, czyli obiektu będącego ograniczeniem dolnym rozwiązania optymalnego; stąd jest on również co najwyżej dwa razy większy od rozmiaru samego rozwiązania optymalnego.



Rys. 4. Na osi zaznaczone są oszacowanie dolne na wielkość rozwiązania (czyli skojarzenia) i rozmiary rozwiązań: optymalnego (OPT) oraz znalezionego przez nasz algorytm ($rozw$). Wartość k jest stosunkiem wielkości naszego rozwiązania oraz oszacowania, stąd k jest też współczynnikiem aproksymacji.

Rozważmy kolejny przykład: problem pakowania plecaka. Mamy do dyspozycji plecak, który może pomieścić przedmioty o sumarycznej masie nie większej niż M, a także n przedmiotów, z których każdy charakteryzuje się dodatnią masą m_i oraz nieujemną wartością w_i. Wiemy, że każdy przedmiot pojedynczo zmieści się do plecaka, czyli $m_i \leqslant M$. Nasze zadanie polega na wybraniu przedmiotów, których suma mas jest nie większa niż M (czyli mieszczących się razem do plecaka) i których sumaryczna wartość jest jak największa.

Klasyczne rozwiązanie tego problemu jest zastosowaniem techniki programowania dynamicznego (można o nim poczytać w praktycznie każdej książce o algorytmach). Jest ono całkiem efektywne, ale tylko wtedy, gdy wszystkie masy bądź wszystkie wartości przedmiotów są stosunkowo niedużymi liczbami całkowitymi. W tym artykule rozważymy ogólniejszą wersję tego problemu, w której wszystkie te wielkości mogą być dowolnie duże, co więcej, mogą być dowolnymi liczbami rzeczywistymi. Stąd jedynym parametrem, jaki nas będzie interesował w oszacowaniu złożoności czasowej algorytmu, będzie n.

Na początku posortujmy przedmioty nierosnąco względem wartości w przeliczeniu na kilogram, czyli stosunku wartości do masy, tak aby $\frac{w_i}{m_i} \geqslant \frac{w_{i+1}}{m_{i+1}}$. Niech j będzie największym takim indeksem, że przedmioty $\{1, \ldots, j\}$ mieszczą się w plecaku (czyli $\sum_{i=1}^{j} m_i \leqslant M$). Jeśli wszystkie przedmioty mieszczą się w plecaku ($j = n$), to pakujemy wszystko. W przeciwnym przypadku wybieramy bardziej wartościowe upakowanie spośród dwóch możliwości:

- przedmioty o indeksach $\{1, \ldots, j\}$,
- pojedynczy przedmiot o indeksie $j + 1$.

Opisany algorytm może wydawać się nieco dziwny – ciekawe jest już chociażby to, że w drugiej z powyższych możliwości ograniczamy się zawsze do jednego (!) przedmiotu; spróbujmy się tym jednak nie zrażać i zastanówmy się, jaki jest współczynnik aproksymacji tego zaskakującego algorytmu. Zauważmy mianowicie, że sumaryczna wartość W przedmiotów o pierwszych $j + 1$ numerach nie może być mniejsza od wartości przedmiotów W_{OPT} w optymalnym upakowaniu. Wynika to stąd, że przedmioty te stanowią „czołówkę” w kategorii *wartość w przeliczeniu na kilogram* i gdyby dało się zapakować je wszystkie do plecaka, to w ten sposób otrzymalibyśmy na pewno rozwiązanie o największej sumarycznej wartości (dokładniejsze uzasadnienie tej obserwacji pozostawiam jako ćwiczenie). Jako że wartością naszego rozwiązania jest maksimum z dwóch liczb, których suma jest równa W, to mamy gwarancję, że wartość wybranego przez nas upakowania jest nie mniejsza niż $W/2$, co dowodzi, że nasze rozwiązanie jest warte nie mniej niż połowę W_{OPT}, czyli jest 2-aproksymacją.

W problemie pakowania plecaka, odwrotnie niż w pokryciu wierzchołkowym, chcieliśmy zmaksymalizować wartość konstruowanego rozwiązania, jednakże metody szacowania współczynnika aproksymacji pozostały takie same – jedyną różnicą jest zmiana kierunku osi z rysunku 4.

Na koniec jako pouczające ćwiczenie zachęcam Cię, Drogi Czytelniku, do skonstruowania przykładów danych, dla których jeden z dwóch kandydatów do upakowania konstruowanego przez nasz algorytm aproksymacyjny jest dowolnie mało wart w stosunku do rozwiązania optymalnego.

Rozwiązanie zadania F 743.
Podczas zamarzania wody o masie m_1 wydzieli się ciepło λm_1, które pozwoli wyparować wodzie o masie m_2 tak, że

$$\lambda m_1 = r m_2.$$

Ale $m_1 + m_2 = m$ to masa całej wody w kolbie, stąd wyparuje

$$\frac{m_2}{m} = \frac{\lambda}{\lambda + r} = 0{,}123$$

całej wody.