Po angielsku problem komiwojażera nazywa się *travelling salesman problem*, w skrócie *TSP*.

*Instytut Informatyki, Uniwersytet Warszawski

Komiwojażer

Anna NIEWIAROWSKA *

Ogólny problem komiwojażera formułujemy następująco: dane jest n miast, a każde dwa z nich połączone są drogą o pewnej długości. W jednym z miast znajduje się komiwojażer, który chce odwiedzić wszystkie miasta w taki sposób, aby w każdym mieście znaleźć się dokładnie jeden raz, a na koniec wędrówki powrócić do miejsca startowego. Naszym celem jest znalezienie najkrótszej możliwej trasy dla komiwojażera.

Problem ten można łatwo przedstawić w języku teorii grafów. Dany jest graf pełny (klika) K_n z dodatnimi wagami na krawędziach, które odpowiadają odległościom między miastami. Niech $d(u,v)$ oznacza wagę krawędzi uv (odległość między miastami u i v). Chcemy znaleźć w tym grafie cykl Hamiltona (czyli cykl przechodzący przez każdy wierzchołek dokładnie jeden raz) o minimalnej sumie wag krawędzi. Ponieważ cykl musi przechodzić przez wszystkie wierzchołki, więc wybór wierzchołka początkowego nie ma znaczenia.

Nasuwającym się rozwiązaniem jest rozpatrzenie wszystkich możliwych cykli i wybranie najlepszego z nich. Jednak złożoność takiego rozwiązania jest wykładnicza względem liczby miast, więc już dla niedużych grafów komiwojażer nie doczeka się na wynik takiego obliczenia.

Często zdarza się, że odległości spełniają dodatkowe warunki, na przykład *nierówność trójkąta*:

$$d(u,v) \leqslant d(u,w) + d(w,v)$$

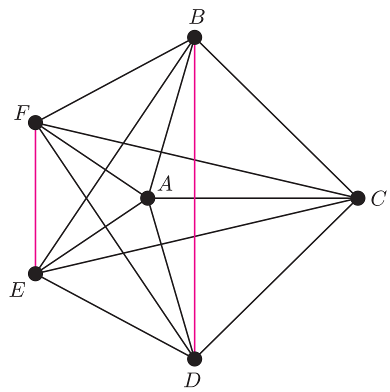
dla każdej trójki miast u, v, w. Jest tak na przykład – ale nie tylko! – wówczas, gdy miasta są położone na płaszczyźnie, a odległości mierzymy w metryce euklidesowej (to tak, jakby komiwojażer latał samolotem zawsze w linii prostej). Jeżeli wagi krawędzi spełniają nierówność trójkąta, to mówimy o *metrycznym problemie komiwojażera*.

Oba problemy – ogólny i metryczny – są NP-trudne, co oznacza, że żadne szybkie (wielomianowe) rozwiązania nie są znane i bardzo możliwe, że w ogóle nie istnieją. Czy w takim razie nie potrafimy wcale pomóc komiwojażerowi? Na szczęście nie jest tak źle. Jeżeli mamy do czynienia z problemem *metrycznym*, to możemy (tym razem efektywnie!) przygotować trasę, która będzie niewiele gorsza od trasy optymalnej – co najwyżej dwukrotnie dłuższa.

W rozwiązaniu będziemy korzystać z minimalnego drzewa rozpinającego grafu. Kilka słów wyjaśnienia dla niewtajemniczonych. Drzewem rozpinającym grafu G jest dowolny podgraf G, który jest drzewem i zawiera wszystkie wierzchołki grafu G. Jeśli krawędzie grafu mają przypisane wagi, to jest sens mówić o minimalnym drzewie rozpinającym – jest to po prostu drzewo rozpinające o minimalnej sumie wag krawędzi spośród wszystkich drzew rozpinających. W języku komiwojażera minimalne drzewo rozpinające to podzbiór wszystkich dróg, który umożliwia przejazd między każdymi dwoma miastami i ma minimalną łączną długość. Drzewo takie można łatwo skonstruować w czasie wielomianowym względem rozmiaru grafu, stosując strategię zachłanną.

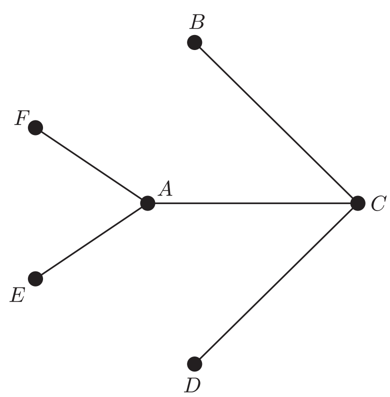
Patrz Cormen, Leiserson, Rivest, Stein *Wprowadzenie do algorytmów*, WNT, Warszawa 2005.

Teraz możemy przedstawić algorytm znajdowania trasy komiwojażera. Dla ułatwienia będziemy od razu pokazywać działanie tego algorytmu na przykładzie z sześcioma miastami, przedstawionym na rysunku (a). Początkowo komiwojażer znajduje się w mieście A.



(a) Kolorowe krawędzie mają koszt 5, czarne 3.

W pierwszym kroku znajdujemy minimalne drzewo rozpinające grafu (rysunek (b)).



(b)

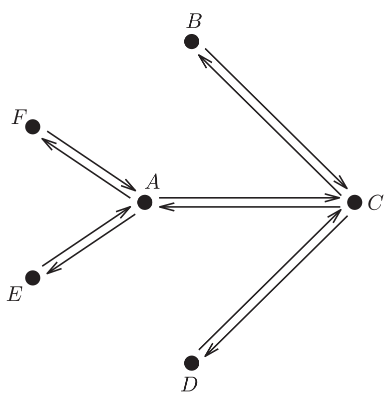
Zauważmy, że gdyby komiwojażer mógł wielokrotnie odwiedzać te same miasta, to jego trasa mogłaby składać się wyłącznie z krawędzi drzewa rozpinającego. Komiwojażer mógłby się wtedy poruszać po drzewie zgodnie z następującymi zasadami:

• jeśli z aktualnie odwiedzanego wierzchołka v prowadzi krawędź do pewnego nieodwiedzonego jeszcze wierzchołka w, to należy się tam udać,

• jeśli wszystkie wierzchołki połączone krawędziami z aktualnym wierzchołkiem v zostały już odwiedzone, to należy cofnąć się do wierzchołka, z którego weszliśmy do v,

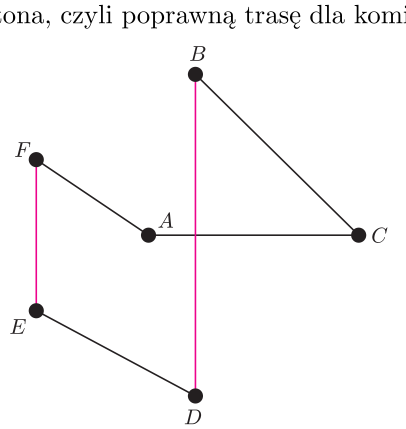
• jeśli aktualny wierzchołek jest wierzchołkiem początkowym i wszystkie sąsiadujące z nim wierzchołki zostały odwiedzone, to kończymy zadanie.

Łatwo zauważyć, że startując z dowolnego wierzchołka drzewa, odwiedzimy wszystkie pozostałe wierzchołki. Powyższy algorytm nazywamy algorytmem przechodzenia grafu (a w naszym przypadku drzewa) *w głąb*. Jeden ze sposobów przejścia przykładowego drzewa został zaprezentowany na rysunku (c). Odwiedzamy tu kolejno wierzchołki:
$A - C - B - C - D - C - A - E - A - F - A$.



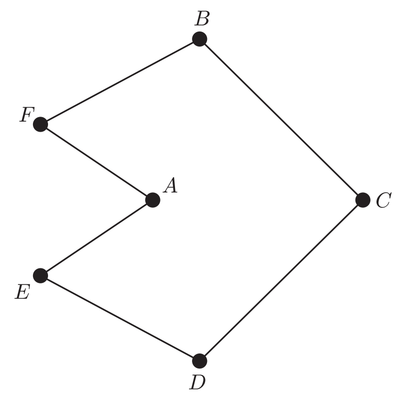
(c)

Okazuje się, że możemy łatwo zmodyfikować znalezioną trasę tak, aby komiwojażer odwiedzał każde miasto dokładnie jeden raz. Wystarczy w tym celu „przeskakiwać" te wierzchołki, które zostały już odwiedzone, a na koniec wrócić do miejsca początkowego. W naszym przykładzie, zaczynając z wierzchołka A, odwiedzimy najpierw wierzchołki C i B. Następnie, przeskakując odwiedzony już wierzchołek C, przejdziemy do D. Dalej odwiedzimy wierzchołki E i F, a na koniec wrócimy do wierzchołka A. Otrzymaliśmy więc przedstawioną na rysunku (d) trasę $A - C - B - D - E - F - A$. Postępując w opisany powyżej sposób, zawsze uzyskamy cykl Hamiltona, czyli poprawną trasę dla komiwojażera.



(d)

Zauważmy, że trasa ta może być gorsza od optymalnej. W naszym przykładzie koszt znalezionego rozwiązania wynosi 22, a koszt optymalnego rozwiązania (przedstawionego na rysunku (e)) wynosi 18. Oczywiście, gdybyśmy znaleźli inne minimalne drzewo rozpinające lub przechodzili drzewo w nieco innym porządku (zgodnym z przedstawionym powyżej opisem), to moglibyśmy otrzymać cykl o innym koszcie.



(e)

Pokażemy, że cykl znajdowany przez przedstawiony algorytm nie może być dużo gorszy od optymalnego cyklu – jego łączny koszt może być co najwyżej dwukrotnie większy.

Optymalnym rozwiązaniem jest pewien cykl Hamiltona C w klice K_n. Usuwając dowolną krawędź tego cyklu, otrzymamy pewne drzewo rozpinające tej kliki. Wynika stąd, że koszt cyklu C jest większy od kosztu pewnego drzewa rozpinającego K_n, więc tym bardziej jest większy niż koszt minimalnego drzewa rozpinającego.

Gdybyśmy po prostu przechodzili w głąb minimalne drzewo rozpinające (jak na rysunku (c)), to koszt trasy byłby dwukrotnie większy od sumy wag krawędzi tego drzewa (każdą krawędzią przechodzimy dwukrotnie). Omijanie odwiedzonych wierzchołków nie zwiększa kosztu (nierówność trójkąta!), więc koszt znalezionego rozwiązania również jest co najwyżej dwukrotnie większy od sumy wag krawędzi minimalnego drzewa rozpinającego. Reasumując ostatnie dwa akapity, stwierdzamy, że:

koszt znalezionej trasy jest nie większy niż dwukrotny koszt minimalnego drzewa rozpinającego i mniejszy niż dwukrotny koszt najtańszego cyklu Hamiltona.

Czytelnikowi pozostawiamy następujące zadanie: skonstruować rodzinę grafów, dla której najgorsze rozwiązania, znajdowane przez algorytm, mają (asymptotycznie) koszt dwukrotnie większy od kosztu optymalnego.

Mówiąc fachowo, znaleźliśmy dla metrycznego problemu komiwojażera *algorytm 2-aproksymacyjny*, tzn. szybki (wielomianowy) algorytm przybliżony, dający zawsze wynik o koszcie co najwyżej 2 razy większym od optymalnego. Dla tego problemu istnieje także nieco bardziej skomplikowany algorytm $\frac{3}{2}$-aproksymacyjny i jest to najlepszy współczynnik, jaki udało się dotychczas uzyskać. Dla ogólnego problemu komiwojażera nie istnieje żaden algorytm aproksymacyjny (o ile $P \neq NP$).

Algorytmy aproksymacyjne są aktywnie badane, ponieważ są jednym ze sposobów atakowania trudnych problemów, których nie potrafimy efektywnie rozwiązywać dokładnie.