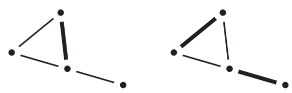
Skojarzenia…

Tomasz IDZIASZEK

Ten numer *Delty* jest zdominowany przez tematykę skojarzeń w grafach. Dla przypomnienia: graf nieskierowany to zbiór wierzchołków połączonych krawędziami, *skojarzenie* zaś w tym grafie to taki podzbiór krawędzi M, że każdy wierzchołek grafu jest incydentny z co najwyżej jedną krawędzią z M (rys. 1).



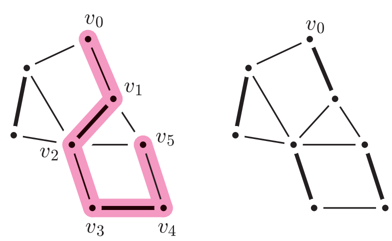
Rys. 1. Po lewej: przykładowe skojarzenie w grafie (pogrubiona krawędź). Po prawej: jedyne najliczniejsze skojarzenie w tym samym grafie.

W wielu zastosowaniach interesować nas będzie wyznaczenie *najliczniejszego* skojarzenia w grafie, tzn. skojarzenia, które zawiera możliwie najwięcej krawędzi. W tym artykule pokażemy, jak znajdować takie skojarzenia.

Nasze algorytmy będą działały metodą przyrostową: dla danego grafu G i wyróżnionego w nim skojarzenia M będą znajdować skojarzenie o jedną krawędź większe bądź też będą stwierdzać, że M jest już najliczniejszym skojarzeniem. W jaki sposób można powiększyć skojarzenie? Zauważmy, że jeśli w grafie istnieją dwa nieskojarzone wierzchołki połączone krawędzią, to można tę krawędź dodać do skojarzenia. Pomysł ten można uogólnić: jeśli w grafie istnieją dwa nieskojarzone wierzchołki v_0, v_{2k+1} połączone ścieżką, na której dokładnie co druga krawędź jest skojarzona, tzn. ścieżką

$$v_0 — v_1 — v_2 — \ldots — v_{2k+1}, \qquad v_i v_{i+1} \in M \text{ jeśli } i \text{ jest nieparzyste,}$$

to można wyrzucić ze skojarzenia k krawędzi $v_i v_{i+1}$ dla i nieparzystego i dorzucić do niego $k+1$ krawędzi $v_i v_{i+1}$ dla i parzystego (rys. 2). Taką ścieżkę, wzdłuż której powiększamy skojarzenie, nazywamy *ścieżką powiększającą* to skojarzenie. Jasne jest, że jeśli taka ścieżka w grafie istnieje, to możemy powiększyć skojarzenie. Okazuje się, że implikacja w przeciwną stronę również zachodzi i jest tezą twierdzenia sformułowanego i udowodnionego przez francuskiego matematyka Claude'a Berge'a:

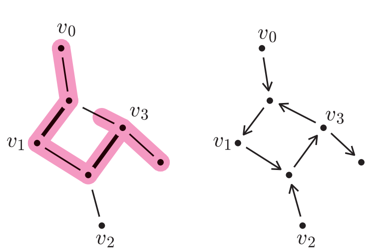


Rys. 2. Graf z wyróżnioną ścieżką powiększającą skojarzenie (kolor) oraz graf po powiększeniu skojarzenia wzdłuż tej ścieżki.

Skojarzenie w grafie jest najliczniejsze wtedy i tylko wtedy, gdy nie istnieje ścieżka powiększająca to skojarzenie.

Tak więc to, czego potrzebujemy, to procedura znajdowania ścieżki powiększającej lub stwierdzania, że takowa nie istnieje. Dla ustalenia uwagi możemy skupić się na poszukiwaniu ścieżki powiększającej zaczynającej się w ustalonym nieskojarzonym wierzchołku v_0 – wystarczy przejrzeć wszystkie takie wierzchołki. Oczywiście, moglibyśmy wyznaczyć wszystkie ścieżki wychodzące z tego wierzchołka, korzystając z przeszukiwania z nawrotami. Taka metoda będzie jednak wymagać czasu wykładniczego, zatem potrzebujemy czegoś sprytniejszego.

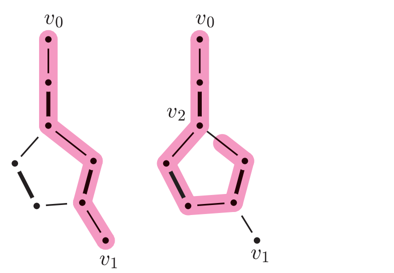
Na początek zajmiemy się grafami *dwudzielnymi*, tzn. takimi, których zbiór wierzchołków możemy podzielić na takie dwa podzbiory V_A, V_B, że każda krawędź łączy wierzchołki należące do różnych podzbiorów. Spróbujmy usprawnić przeszukiwanie z nawrotami, tak aby odwiedzić każdy wierzchołek co najwyżej raz. A konkretnie: zaczynając z wierzchołka $v_0 \in V_A$, wykonujemy przeszukiwanie grafu w głąb, z tym że w każdym wierzchołku na głębokości parzystej przeglądamy nieskojarzone krawędzie wychodzące z tego wierzchołka, w każdym zaś wierzchołku na głębokości nieparzystej idziemy krawędzią skojarzoną (może być co najwyżej jedna). Jeśli trafimy na wierzchołek nieskojarzony, to znaczy, że znaleźliśmy ścieżkę powiększającą (rys. 3). Co ciekawe, jeśli na niego nie trafiliśmy, to znaczy, że ścieżki powiększającej nie ma. Jak to udowodnić?



Rys. 3. Graf dwudzielny G z wyróżnionymi krawędziami odwiedzonymi podczas przeszukiwania w głąb oraz graf skierowany G'. W obu grafach podpisano wierzchołki ze zbioru V_A.

Rozważmy graf skierowany G' o zbiorze wierzchołków $V_A \cup V_B$. Dla każdej krawędzi $v_A v_B$, $v_A \in V_A$, $v_B \in V_B$, z naszego G dodajmy do G' skierowaną krawędź: $v_A \to v_B$, jeśli $v_A v_B \notin M$, albo $v_B \to v_A$, jeśli $v_A v_B \in M$. Zauważmy, że w grafie G istnieje ścieżka powiększająca wtedy i tylko wtedy, gdy w grafie G' da się dojść (zgodnie ze skierowaniem krawędzi) z wierzchołka v_0 do innego wierzchołka nieskojarzonego. A powyższy algorytm to nic innego jak zwykłe przeszukiwanie w głąb grafu G'.

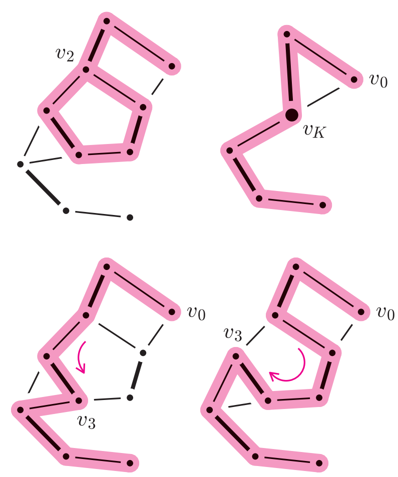
Przeszukiwanie to działa w czasie $O(m)$, zatem znalezienie ścieżki powiększającej zajmie czas $O(nm)$, jeśli zaczynamy przeszukiwanie z każdego wierzchołka nieskojarzonego od nowa, lub $O(m)$, jeśli zaczynamy naraz z wszystkich wierzchołków nieskojarzonych z V_A (dlaczego możemy tak zrobić?). Skojarzenie może być powiększone co najwyżej $n/2$ razy, zatem cały algorytm działa w czasie $O(nm)$. Dodajmy, że w praktyce nie opłaca się zaczynać z pustego skojarzenia, ale z jakiegoś skojarzenia znalezionego prostszą metodą (przykładowo, algorytm zachłannie dodający kolejne krawędzie do skojarzenia zawsze znajdzie skojarzenie o liczności równej co najmniej połowie liczności najliczniejszego skojarzenia).



Rys. 4. Po lewej: graf ze skojarzeniem (pogrubione krawędzie) i ścieżką powiększającą (kolor). Po prawej: ten sam graf z krawędziami odwiedzonymi przez algorytm.

Okazuje się, że w przypadku grafów, które nie są dwudzielne, sprawa się komplikuje, a nasz algorytm nie działa. Istotnie, spójrzmy na graf z rysunku 4. Pomimo tego, że istnieje w nim ścieżka powiększająca z wierzchołka v_0 do wierzchołka v_1, to algorytm nigdy jej nie znajdzie, jeśli zacznie obchodzić cykl od lewej strony (w szczególności nigdy nie odwiedzi wierzchołka v_1).

Cały kłopot jest powodowany przez cykle, które znajdujemy, przeszukując graf. Problematyczne okazują się mianowicie cykle *nieparzystej* długości, które są połączone z wierzchołkiem v_0 ścieżką o parzystej długości. Taki cykl nazwiemy *kielichem* (ang. *blossom*), a ścieżkę *łodygą* (ang. *stem*). Kanadyjski matematyk Jack Edmonds podał następujący sposób, w jaki można sobie z nimi radzić: ściągnąć, czyli zastąpić jednym wierzchołkiem, a następnie rekurencyjnie poszukać ścieżki powiększającej w mniejszym grafie. Przyjrzyjmy się temu pomysłowi bliżej.



Rys. 5. Na górze po lewej: graf G z zaznaczonym kielichem. Na górze po prawej: graf G_K z zaznaczoną ścieżką powiększającą. Na dole: dwie możliwości uzupełnienia ścieżki powiększającej w grafie G.

Operacja ściągnięcia kielicha wygląda następująco (rys. 5): zastępujemy wszystkie wierzchołki kielicha jednym wierzchołkiem v_K, który jest połączony krawędziami z wszystkimi sąsiadami kielicha, otrzymując graf G_K. Zauważmy, że na rysunku wierzchołek v_2, który łączył kielich z łodygą, był jedynym wierzchołkiem spośród wierzchołków kielicha skojarzonym z wierzchołkiem poza kielichem, zatem wierzchołek v_K będzie również skojarzony z dokładnie jednym wierzchołkiem. Wynika z tego, że po ściągnięciu nowo utworzone skojarzenie M_K nadal jest poprawne. Okazuje się, że prawdziwy jest następujący lemat:

W grafie G istnieje ścieżka powiększająca skojarzenie M wtedy i tylko wtedy, gdy w grafie G_K istnieje ścieżka powiększająca skojarzenie M_K.

Dla dowodu lematu załóżmy, że w grafie G_K znaleźliśmy ścieżkę powiększającą S. Pokażemy, jak dzięki niej znaleźć ścieżkę w grafie G. Jeśli ścieżka S nie zawiera wierzchołka v_K, to w oczywisty sposób jest ona również ścieżką powiększającą w G. W przeciwnym przypadku można ją uzupełnić krawędziami z kielicha, przechodząc go w lewo lub w prawo, w zależności od położenia wierzchołka v_3, którym wchodzi druga krawędź ścieżki (rys. 5). Ponadto każda ścieżka powiększająca w G musi być takiej postaci (tzn. jeśli przecina kielich, to wchodzi do niego krawędzią łodygi), zatem jest również ścieżką powiększającą w G_K.

Jako zadanie dla Czytelnika pozostawiamy wykazanie, że jeśli w G istnieje ścieżka powiększająca, to nasz algorytm wyszukujący ścieżki powiększające albo ją znajdzie, albo znajdzie kielich, który można ściągnąć.

Znalezienie ścieżki powiększającej z ustalonego wierzchołka zajmuje czas $O(nm)$, gdyż potencjalnie aż $n/2$ razy będziemy musieli wykonać ściągnięcie kielicha. Zatem cały algorytm działa w czasie $O(n^3m)$.

Na koniec zaznaczmy, że opisane algorytmy nie są najlepszymi znanymi algorytmami znajdującymi najliczniejsze skojarzenia w grafach, są za to niezbyt skomplikowane. Przykładowo algorytm Hopcrofta–Karpa dla grafów dwudzielnych oraz algorytm Micalego–Vaziraniego dla grafów dowolnych działają w czasie $O(\sqrt{n}m)$, zaś randomizowany algorytm Muchy–Sankowskiego sprowadza znajdowanie skojarzenia w grafie dowolnym do problemu mnożenia macierzy i działa w czasie $O(n^{2{,}37})$.