

Pokrycie wierzchołkowe kontratakuje

Marcin PILIPCZUK*

W *Delcie* 7/2009 Marek Cygan opisał pewien sposób radzenia sobie z tym, że dla niektórych *trudnych* problemów nie potrafimy znaleźć szybkiego algorytmu. Autor rozważał klasę problemów NP-trudnych – czyli takich, których prawdopodobnie nie można rozwiązać w czasie wielomianowym – i pokazywał, że dla wielu z nich można w efektywny sposób skonstruować nie dokładne, lecz *przybliżone* rozwiązanie.

W tym artykule spróbujemy ugryźć trudne problemy z drugiej strony. Wciąż poruszamy się w świecie algorytmów działających w czasie wielomianowym – te uznajemy za *szybkie*. Ale zamiast poszukiwać dokładnego rozwiązania, użyjemy innej sztuczki: będziemy szukali algorytmów, które starają się jak najbardziej zmniejszyć rozmiar instancji (czyli egzemplarza problemu, np. zadanego grafu), zachowując przy tym odpowiedź. Wówczas, dla zredukowanej instancji, możemy odpalić nasz ulubiony algorytm dokładny, np. rozpatrujący wszystkie możliwości. A jako że rozmiar egzemplarza będzie dużo mniejszy od początkowego, algorytm dokładny być może zadziała całkiem szybko. Przejdźmy do przykładu.

Pokryciem wierzchołkowym w grafie G nazwiemy taki zbiór wierzchołków P , że każda krawędź G ma co najmniej jeden koniec w P . Problem znalezienia liczności najmniejszego pokrycia wierzchołkowego w danym grafie jest NP-trudny. Marek Cygan w swoim artykule pokazywał, jak szybko znaleźć pokrycie wierzchołkowe, które jest co najwyżej dwa razy liczniejsze od optymalnego (czyli tzw. 2-aproksymację).

My będziemy rozważać wersję problemu pokrycia wierzchołkowego, w której odpowiedź jest binarna („TAK” lub „NIE”). Mianowicie, mając dany graf G oraz liczbę naturalną k , pytamy, czy w grafie G istnieje pokrycie wierzchołkowe zawierające co najwyżej k wierzchołków. Oczywiście, z punktu widzenia algorytmów wielomianowych obie wersje są równoważne: mając dany algorytm rozwiązujący wersję decyzyjną, możemy wyszukać najmniejsze k , dla którego algorytm ten daje odpowiedź „TAK”, i w ten sposób otrzymać rozmiar najmniejszego pokrycia wierzchołkowego.

Mamy więc dany graf G i liczbę k . Co możemy teraz zrobić? Ustalmy wierzchołek v . Jeśli v nie należy do pewnego pokrycia wierzchołkowego P , to wszyscy sąsiedzi v muszą należeć do P . Poszukujemy P o mocy co najwyżej k , więc jeśli v ma co najmniej $k + 1$ sąsiadów, to musimy wziąć v do P . W ten sposób otrzymaliśmy następującą *regulę redukcyjną*: *jeśli w grafie G jest wierzchołek v , który ma co najmniej $k + 1$ sąsiadów, wyrzuc z G wierzchołek v oraz incydentne z nim krawędzie oraz zmniejsz k o jeden*. W wyniku zastosowania tej reguły otrzymaliśmy równorzędną instancję problemu pokrycia wierzchołkowego: w zredukowanym grafie istnieje pokrycie wierzchołkowe rozmiaru co najwyżej $k - 1$ wtedy i tylko wtedy, gdy w oryginalnym grafie istniało pokrycie wierzchołkowe rozmiaru co najwyżej k .

A co się dzieje, jeśli nie możemy zastosować powyższej reguły? Każdy wierzchołek jest incydentny z co najwyżej k krawędziami, więc pokrycie wierzchołkowe P *pokrywa* co najwyżej $|P| \cdot k$ krawędzi. Czyli, jeśli nie możemy zastosować naszej reguły, a zostało nam więcej niż k^2 krawędzi, możemy śmiało odpowiedzieć „NIE”.

Otrzymaliśmy właśnie coś, co informatycy nazywają *jądrem* (ang. *kernel*) problemu pokrycia wierzchołkowego: pokazaliśmy szybki algorytm, który albo rozwiązuje problem pokrycia wierzchołkowego, albo znajduje równoważną instancję problemu o co najwyżej k^2 krawędziach. Zwróćmy uwagę na to, że rozmiar zredukowanego grafu zależy tylko od parametru k , a nie od rozmiaru wyjściowego grafu: początkowy graf mógł być olbrzymi.

A może da się lepiej? Zredukowaliśmy graf do k^2 krawędzi, ale może da się jeszcze ograniczyć liczbę wierzchołków? Zacznijmy od prostej obserwacji: z grafu G możemy wyrzucić wszystkie wierzchołki izolowane. Zostanie nam

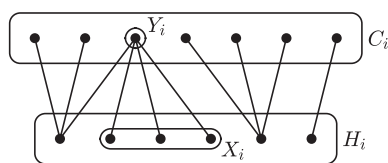
*Instytut Informatyki,
Uniwersytet Warszawski

co najwyżej $2k^2$ wierzchołków. Pokażemy teraz inną regułę redukcyjną – istotnie bardziej skomplikowaną od poprzedniej – która prowadzi do dużo mniejszego grafu: zredukujemy liczbę wierzchołków do $4k$. Założmy więc, że G ma więcej niż $4k$ wierzchołków.

Aproksymując pokrycie wierzchołkowe, Marek Cygan używał *skojarzeń*. My również ich użyjemy. Skojarzeniem w grafie nazywamy zbiór krawędzi, w którym żadne dwie nie mają wspólnego końca. Istnieje szybki algorytm znajdujący najliczniejsze skojarzenie w dowolnym grafie – założmy więc, że znamy M , najliczniejsze skojarzenie w grafie G . Zauważmy, iż w dowolnym pokryciu wierzchołkowym musi być co najmniej jeden koniec każdej krawędzi z M . Skoro te nie są incydentne, to jeśli $|M| > k$, możemy od razu odpowiedzieć „NIE”. Założmy więc dalej, że $|M| \leq k$.

Niech H_0 będzie zbiorem wierzchołków, które są końcami krawędzi w M , a C_0 niech będzie zbiorem pozostałych wierzchołków. Skoro $|M| \leq k$, to $|H_0| = 2 \cdot |M| \leq 2k$, czyli $|C_0| > 2k$. Co więcej, zauważmy, że w G nie ma żadnej krawędzi łączącej dwa wierzchołki w C_0 : gdyby była, moglibyśmy dodać ją do M , powiększając liczbę skojarzenia.

Uwaga, teraz będziemy robić coś skomplikowanego. Wyrzucmy z grafu G wszystkie krawędzie o obu końcach w H_0 . Otrzymamy wówczas graf dwudzielny G_0 , który z jednej strony będzie miał H_0 , a z drugiej C_0 . Sprawdźmy, czy w tym grafie istnieje skojarzenie rozmiaru $|H_0|$ (czyli największe możliwe). Jeśli nie istnieje, to, z twierdzenia Halla, istnieje zbiór $X_0 \subset H_0$ i zbiór jego sąsiadów $Y_0 \subset C_0$, taki że $|X_0| > |Y_0|$. Wyrzucmy z grafu X_0 i Y_0 , tj. weźmy $H_1 = H_0 \setminus X_0$ i $C_1 = C_0 \setminus Y_0$. Postępujemy tak do skutku, tj. mając dane H_i i C_i :



Rys. 1. Krok konstrukcyjny: wybór X_i oraz Y_i .

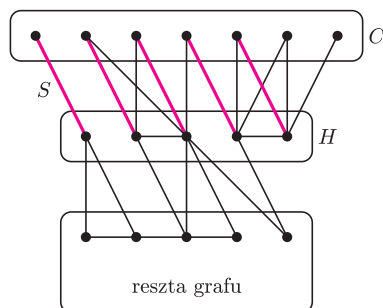
1. konstruujemy graf dwudzielny G_i , biorąc z grafu G tylko krawędzie łączące H_i z C_i ;
2. sprawdzamy, czy istnieje w tym grafie skojarzenie rozmiaru $|H_i|$ – jeśli tak, to koniec;
3. jeśli nie, to korzystając z twierdzenia Halla, znajdujemy zbiór $X_i \subset H_i$ i zbiór jego sąsiadów $Y_i \subset C_i$, taki że $|X_i| > |Y_i|$;
4. przypisujemy $H_{i+1} = H_i \setminus X_i$, $C_{i+1} = C_i \setminus Y_i$.

Pomińmy tutaj problem, jak znajdować zbiory X_i i Y_i : wystarczy nam informacja, że można to zrobić w czasie wielomianowym. Zauważmy, że postępując według powyższej procedury, utrzymujemy następujące niezmienniki:

1. w G nie ma krawędzi łączącej dwa wierzchołki ze zbioru C_i , gdyż $C_i \subset C_0$, a ten warunek był spełniony już dla C_0 ;
2. $|H_i| < |C_i|$, bo $|H_0| < |C_0|$ i w każdym kroku $|X_i| > |Y_i|$;
3. zawsze $X_i \neq H_i$, gdyż zbiorem sąsiadów H_i jest całe C_i – żaden z wierzchołków C_i nie jest izolowany, bo te usunęliśmy. A zatem H_i nigdy nie stanie się puste;
4. H_i jest zbiorem sąsiadów C_i , gdyż za każdym razem usuwając z H_i zbiór X_i , ze zbioru C_i usuwamy Y_i , czyli wszystkich sąsiadów X_i .

W związku z tym, gdy nasza procedura zakończy działanie, otrzymamy niepuste zbiory wierzchołków $H = H_i$ oraz $C = C_i$ o następujących własnościach:

1. w G nie ma krawędzi łączącej dwa wierzchołki z C ;
2. istnieje skojarzenie S rozmiaru $|H|$, w którym każda krawędź łączy wierzchołek z H z wierzchołkiem z C ;
3. H jest zbiorem sąsiadów C .



Rys. 2. Rozkład koronny grafu G .

To, co właśnie otrzymaliśmy, nazywamy *rozkładem koronnym* (ang. *crown decomposition*) grafu: mamy koronę C leżącą na głowie H . Zauważmy, że w dowolnym pokryciu wierzchołkowym w grafie G musimy wybrać co najmniej $|S| = |H|$ wierzchołków spośród $C \cup H$ – musimy wybrać po końcu każdej krawędzi z S . Ale, z drugiej strony, biorąc do pokrycia wierzchołkowego całe H , pokrywamy wszystkie krawędzie incydentne z $C \cup H$. Możemy więc zachłannie wziąć do pokrycia wierzchołkowego całe H i usunąć z grafu wierzchołki $C \cup H$ oraz incydentne z nimi krawędzie. Tym samym zmniejszyliśmy rozmiar grafu.

Zastanówmy się, co właśnie osiągnęliśmy. Wyszliśmy od takiej instancji problemu pokrycia wierzchołkowego (grafu G oraz parametru k), że graf G miał więcej niż $4k$ wierzchołków. Przeraził Uważny Czytelnik zauważy także, że w powyższej konstrukcji po kryjomu założyliśmy $k > 0$. Możemy, dla danego grafu G z parametrem k , powtarzać powyższą redukcję, dopóki jest to możliwe. Zauważmy, iż przy redukcji parametr k może się zmieniać – ale zawsze będzie malał. Mamy trzy możliwe zakończenia tej procedury:

1. pewna redukcja odpowie „NIE”, gdyż rozmiar skojarzenia M będzie za duży – wówczas wiemy, że w początkowym grafie odpowiedź też była „NIE”;
2. otrzymamy w wyniku redukcji parametr $k = 0$ – wówczas odpowiedź brzmi „TAK”, jeśli w grafie nie pozostała żadna krawędź, lub „NIE” w przeciwnym przypadku;
3. otrzymamy nowy zredukowany graf G z nowym parametrem k , taki że graf G ma co najwyżej $4k$ wierzchołków.



Rozwiązanie zadania F 761.
Bilans energii ma postać

$$q_1 \lambda + c q_1 (t_1 - t_2) = c q_2 (t_x - t_2).$$

Stąd

$$t_x = \frac{q_1 \lambda}{q_2 c} + t_2 \left(1 - \frac{q_1}{q_2}\right) + \frac{q_1}{q_2} t_1 = 72^\circ \text{C}.$$

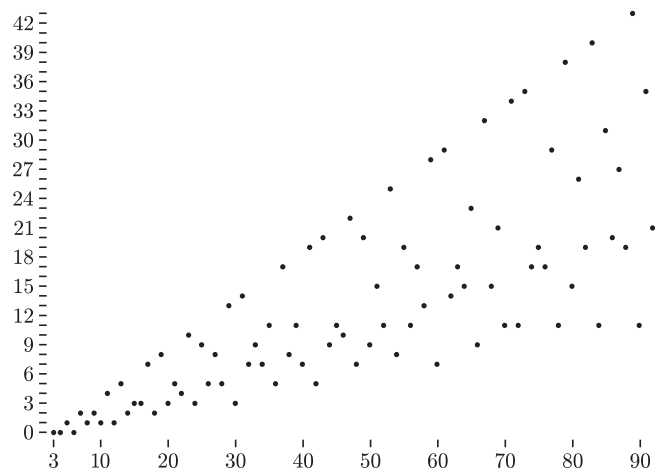
Pokazaliśmy właśnie, że algorytm polegający na aplikowaniu opisanej redukcji, dopóki jest to możliwe, prowadzi do zredukowania wyjściowego grafu do rozmiaru co najwyżej $4k$, gdzie k jest parametrem zredukowanej instancji, a więc jest nie większy od parametru oryginalnej instancji.

Wykonując powyższą konstrukcję trochę uważniej, można otrzymać jądro o $3k$ wierzchołkach, a komplikując dużo bardziej, da się dojść do algorytmu redukującego graf do $2k$ wierzchołków. Z drugiej strony, ostatnio udowodniono, że przy pewnych, rozsądnych założeniach z teorii złożoności nie istnieje jądro o k^γ krawędziach dla żadnego $\gamma < 2$.

Wiele NP-trudnych problemów ma niewielkie jądra. Bardzo często algorytmy redukujące są proste, opierają się na kombinatorycznych spostrzeżeniach, a nie na wielkiej teorii. Można by całą *Deltę* wypełnić opisami algorytmów jak ten powyżej, ale może jednak tego nie róbmy.

Ta sama funkcja trzema sposobami

Funkcję na ogół określa się za pomocą wykresu, tabelki lub wzoru. Oto ta sama funkcja $\#$ w tych trzech postaciach.



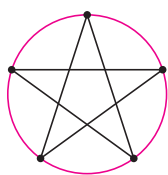
n	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20
$\#$	0	0	1	0	2	1	2	1	4	1	5	2	3	3	7	2	8	3
n	21	22	23	24	25	26	27	28	29	30	31	32	33	34	35	36	37	38
$\#$	5	4	10	3	9	5	8	5	13	3	14	7	9	7	11	5	17	8
n	39	40	41	42	43	44	45	46	47	48	49	50	51	52	53	54	55	56
$\#$	11	7	19	5	20	9	11	10	22	7	20	9	15	11	25	8	19	11
n	57	58	59	60	61	62	63	64	65	66	67	68	69	70	71	72	73	74
$\#$	17	13	28	7	29	14	17	15	23	9	32	15	21	11	34	11	35	17
n	75	76	77	78	79	80	81	82	83	84	85	86	87	88	89	90	91	92
$\#$	19	17	29	11	38	15	26	19	40	11	31	20	27	19	43	11	35	21

$$\#(n) = \left(\frac{n}{2} \prod_{p_i | n} \left(1 - \frac{1}{p_i}\right) \right) - 1,$$

gdzie p_i przebiegają wszystkie dzielniki pierwsze liczby n ; na przykład $\#(100) = 19$.

Proszę sprawdzić, że funkcja ta oblicza, ile jest różnych wielokątów foremnych gwiazdzistych o n wierzchołkach.

Wielokąt foremny gwiazdzisty o n wierzchołkach to łamana zamknięta wpisana w okrąg, mająca wszystkie odcinki równej długości, większej od boku zwykłego n -kąta foremnego. Wynika z tego, że łamana taka ma samoprzecięcia. Najbardziej znanym przykładem jest, mający 5 wierzchołków, pitagorejski pentagram.



M. K.