Wojciech Wawrzyniak

Rozproszone algorytmy aproksymacyjne w analizie własności grafowych.

Rozprawa doktorska z Informatyki napisana
na Wydziale Matematyki i Informatyki
Uniwersytetu im. Adama Mickiewicza w Poznaniu

Promotor: **prof. dr hab. Michał Karoński**
Opiekun naukowy: **dr Michał Hańćkowiak**

Poznań, 2010

Składam serdeczne podziękowania:

prof. dr hab. Michałowi Karońskiemu oraz
dr Michałowi Hańćkowiakowi
za życzliwą pomoc, cenne rady i liczne wskazówki udzielone
w trakcie realizacji niniejszej pracy,

Rodzinie za wsparcie i pomoc w trudnych momentach.

Spis treści

Wstęp

Zastosowanie systemów wieloprocesorowych pozwala znacząco przyspieszyć proces obliczeń, co zaobserwowano już w początkowych latach rozwoju informatyki. Podstawowym problemem napotykanym podczas próby rozpraszania sekwencyjnych algorytmów jest ograniczenie wymiany komunikatów do niezbędnego minimum, których przesyłanie zajmuje najwięcej czasu. Prace nad tym zagadnieniem zaowocowały na początku lat 90, w artykule N. Liniala *"Locality in distributed graph algorithms" ([18])* próbą wprowadzenia formalnej definicji systemów rozproszonych. Pojęcie to stało się inspiracją do dalszych badań, a tym samym przyczyniło się do rozwoju algorytmów rozproszonych.

System rozproszony rozumiemy jako sieć komputerową składającą się z procesorów wykonujących niezależnie operacje i obliczenia. Dane pomiędzy połączonymi procesorami przesyłane są za pomocą tzw. komunikatów. W modelu tym algorytmy wykonywane są synchronicznie w podziale na rundy. W pojedynczym kroku każdy procesor może odebrać oraz wysłać wiadomości od/do wszystkich swoich sąsiadów oraz wykonać lokalne obliczenia na podstawie dotychczas zgromadzonych informacji. W analizie często sieć utożsamia się z pojęciem grafów, gdzie wierzchołkami są procesory, a jako krawędzie rozumiemy istniejące połączenia pomiędzy nimi.

W tak zdefiniowanym modelu możemy rozważać rozwiązywanie fundamentalnych problemów grafowych, takich jak: *minimalny zbiór dominujący, maksymalny zbiór niezależny, maksymalne skojarzenie czy też minimalne pokrycie wierzchołkowe*. Wyznaczanie powyższych zbiorów w sieciach komputerowych posiada praktyczne zastosowanie, np. w problemie optymalnego przydziału zadań, w którym korzysta się z wyznaczonego największego skojarzenia. Inny z algorytmów znajduje zastosowanie w sieciach "ad hoc", gdzie rozsyłanie wiadomości może odbywać się poprzez tzw. szkielet sieci zbudowany w oparciu o spójny minimalny zbiór dominujący. Ponadto w problemie rozsyłania wiadomości poprzez dużą liczbę rozłącznych wierzchołkowo ścieżek, można w oczywisty sposób wykorzystać rozwiązanie pakowania grafów.

Celem rozprawy doktorskiej jest przedstawienie i omówienie nowych zaproponowanych przeze mnie rozproszonych algorytmów grafowych rozwiązujących przybliżone rozwiązania podanych problemów.

W Rozdziale 1 rozprawy wprowadzone zostały podstawowe definicje i własności teorii grafów oraz formalny opis modelu obliczeń.

W Rozdziale 2 przedstawiony został algorytm rozwiązujący problem pakowania podanego grafu H w planarnym grafie G z pracy [9]. Algorytm ten zwraca przybliżone rozwiązanie

problemu o współczynniku aproksymacji dowolnie bliskim jeden, a jego złożoność czasowa wynosi $O(polilog\ n)$, gdzie n oznacza liczbę wierzchołków w grafie G. Problem ten, będący uogólnieniem problemu największego skojarzenia w grafie G, poruszany był w pracy [5] jednak dla innej klasy grafów, tzw. jednostkowych grafów dyskowych (UDG).

Z praktycznego punktu widzenia zastosowanie znajdują również algorytmy działające w znacząco krótszym czasie niż $O(polilog\ n)$ jednak z dużo gorszym współczynnikiem aproksymacji. W przypadku systemów on-line wymagamy szybkiej reakcji na zmieniające się dane, często nawet o stałym kwancie czasu (czyli niezależnym od wielkości sieci). Podejście to przedstawione zostało w Rozdziale 3, gdzie opisane zostały algorytmy ([11]) konstruujące $(2+\epsilon)$-aproksymację minimalnego pokrycia wierzchołkowego w grafach o ograniczonym stopniu oraz w grafach o ograniczonej lesistości w czasie stałym. W rozproszonym modelu obliczeń problem ten poprzez prostą 2-aproksymację maksymalnego skojarzenia do problemu pokrycia wierzchołkowego poruszany był w pracach [13] oraz [21], zawierających algorytmy działające w czasie $O(polilog\ n)$ oraz $O(\Delta + log^* n)$. Dopiero stosunkowo niedawno w pracach [23] oraz [24] (równolegle do wyniku uzyskanego w tej rozprawie) przedstawiono algorytm znajdujący 3-aproksymację oraz 2-aproksymację dla grafów o ograniczonym stopniu działający w czasie stałym. Procedura przeze mnie zaprezentowana posiada minimalnie gorszy współczynnik aproksymacji (o dowolnie małą $\delta > 0$), jednak jego budowa jest bardziej przejrzysta. W rozprawie wprowadzona została również modyfikacja tego algorytmu umożliwiająca wyznaczenie minimalnego pokrycia wierzchołkowego w grafach o ograniczonej lesistości z tym samym współczynnikiem aproksymacji oraz o stałym czasie działania.

Kompromisem pomiędzy złożonością czasową algorytmów, a ich współczynnikiem aproksymacji są algorytmy sublogarytmiczne ([10]) przedstawione w Rozdziale 4. Zwracają one dla grafów planarnych rozwiązanie będące $(1 \pm \epsilon)$-aproksymacją problemów: *minimalnego zbioru dominującego, największego skojarzenia, największego ważonego zbioru niezależnego* w czasie $O(log^* n)$ rund. Z prac [4], [8], [15] i [16] znane są algorytmy działające w czasie $O(polilog\ n)$ lub przeznaczone dla innych klas grafów. Najbliższy tematycznie wynik opisany został w pracy [17], gdzie autorzy pokazali procedurę wyznaczającą 72-aproksymację minimalnego zbioru dominującego w czasie stałym. Algorytmy o dowolnie dobrym współczynniku aproksymacji i działające w tak krótkim czasie nie zostały jednak wcześniej zaprezentowane dla klasy grafów planarnych. Co więcej z oszacowania dolnego udowodnionego w ostatnim Rozdziale 5 wynika, iż algorytmy te są optymalne pod względem złożoności czasowej.

1 Wprowadzenie.

Celem tego rozdziału jest wprowadzenie wymaganych definicji i opisu problemów omawianych w niniejszej rozprawie doktorskiej.

1.1 Podstawowe definicje i założenia.

Definicja 1.1. *Parę $G = (V, E)$ nazywamy* **grafem** *G, gdzie V jest zbiorem wierzchołków, a $E \subseteq \{\{u, v\} : u, v \in V(G), u \neq v\}$ zbiorem krawędzi. Liczbę wierzchołków w grafie $|V(G)|$ oznaczamy jako n, a liczbę krawędzi $|E(G)|$ w grafie oznaczamy jako e.*

Definicja 1.2. *Graf $G = (V, E)$ nazywamy* **planarnym**, *jeżeli można go przedstawić na płaszczyźnie w ten sposób, by żadne dwie krawędzie się ze sobą nie przecinały. Taką reprezentację grafu G na płaszczyźnie nazywamy* **grafem płaskim**.

Definicja 1.3. Odległością *pomiędzy wierzchołkami $u \in V(G)$ i $v \in V(G)$ nazywamy długość najkrótszej ścieżki pomiędzy tymi wierzchołkami i oznaczać będziemy jako $d_G(u, v)$.*

Definicja 1.4. Średnicą grafu *$G = (V, E)$ nazywamy najmniejszą liczbę δ taką, że $\underset{v,u \in V}{\forall}$ odległość $d_G(u, v) \leqslant \delta$ i oznaczamy jako $diam(G)$.*

Definicja 1.5. *Niech H będzie grafem, a H' jego podgrafem ($H' \subseteq H$).* **Słabą średnicą grafu** *H' nazywamy najmniejszą δ taką, że $\underset{v,u \in V(H')}{\forall}$ odległość $d_H(u, v) \leqslant \delta$ i oznaczamy jako $weak_H diam(H')$.*

Definicja 1.6. *Graf M, w którym dopuszczalne są krawędzie wielokrotne oraz pętle nazywamy* **multigrafem**.

Definicja 1.7. *Mamy dany graf $G = (V, E)$. Wartość $\max_{G' \subset G} \lceil \frac{|E(G')|}{|V(G')-1} \rceil$ nazywamy lesistością grafu G i oznaczać będziemy jako $arb(G)$.*

W rozprawie wybrany został tzw. model $\mathcal{LOCAL}$ ([18]) jako model obliczeń algorytmów rozproszonych.

Definicja 1.8. *Modelem obliczeń jest tzw.* **sieć komunikacyjna***:*

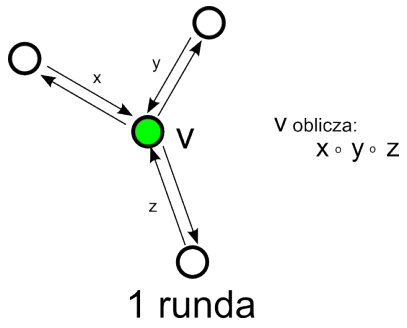
- *siecią jest nieskierowany graf $G = (V, E)$, gdzie wierzchołki V reprezentują procesory, a krawędzie E połączenia pomiędzy nimi;*

- *wykonywanie algorytmu podzielone jest na synchroniczne rundy;*

- *w jednej rundzie komunikacyjnej każdy wierzchołek może wysłać i odebrać wiadomości od/do swoich sąsiadów oraz na ich podstawie wykonać obliczenia;*

- *nie są nałożone żadne ograniczenia na złożoność wykonywanych obliczeń przez pojedynczy procesor;*

- *nie są nałożone żadne ograniczenia na długość przesyłanych komunikatów;*



Rysunek 1: Zadania wykonywane przez pojedynczy wierzchołek w modelu $\mathcal{L}OCAL$ w jednej synchronicznej rundzie.

Złożoność algorytmów. Według przyjętych zasad ([3]) przydatność algorytmów sekwencyjnych możemy rozpatrywać według następujących kryteriów:

- *Złożoność obliczeniowa* - jest to ilość wykonywanych obliczeń elementarnych dla pewnych danych wejściowych i wyrażana jest często, jako funkcja zależna on rozmiaru danych wejściowych n;

- *Złożoność pamięciowa* - jest to ilość potrzebnej pamięci do wykonania algorytmu dla pewnych danych wejściowych i wyrażana jest często w postaci funkcji o parametrze n, gdzie n to rozmiar danych wejściowych.

Przydatność danego algorytmu rozproszonego może być również rozpatrywana pod kątem złożoności komunikacyjnej (ilości przesyłanych komunikatów), a także ich długości.

Odzwierciedlenie tego podejścia znalazło w modelu $\mathcal{CONGEST}$. Nakłada on bowiem dodatkowe założenie, na długość pojedynczej wysyłanej wiadomości i wynosi maksymalnie $\log n$ bitów. W rozprawie tej z uwagi na wybrany model głównym kryterium branym pod uwagę będzie złożoność czasowa (pomimo, iż w przypadku algorytmów sublogarytmicznych modele te są równoważne).

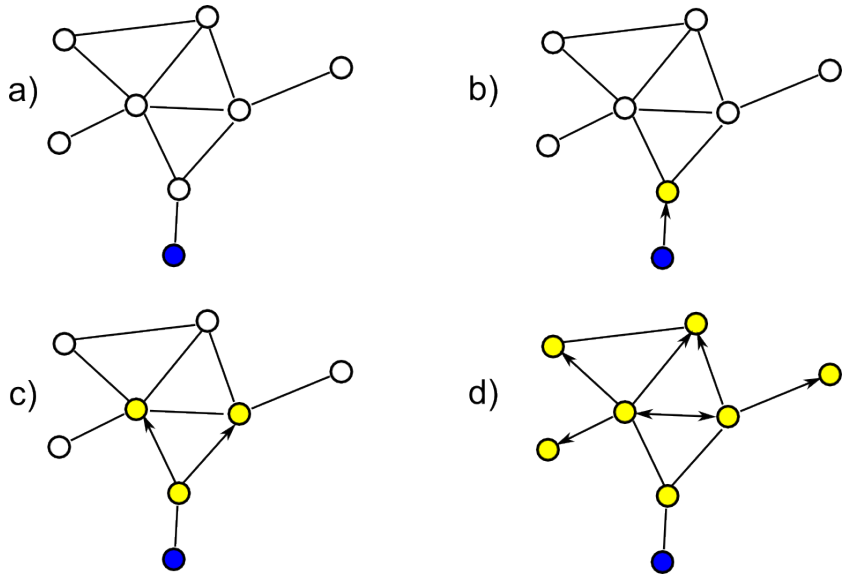
Do czasu, aż nie zostanie to określone inaczej, w pracy tej będą obowiązywały następujące założenia:

- modelem obliczeń jest model $\mathcal{LOCAL}$;
- każdy z wierzchołków (procesorów) posiada informację tylko o incydentnych do niego krawędziach (istniejących połączeniach), nie posiada natomiast żadnej informacji o strukturze całego grafu;
- dopuszczalna jest znajomość przez wierzchołki moc zbioru $V(G) = n$;
- procesory posiadają unikalne identyfikatory o długości $\log n$-bitów.

Rozpatrywany model obliczeń posiada kilka ważnych właściwości:

Własność 1.9. *Rozwiązanie dowolnego problemu w grafie G o stałej średnicy ($diam(G) = const$) możliwe jest w czasie $O(1)$.*

Własność ta otrzymywana jest "ad hoc" z założeń modelu. Zauważmy bowiem brak ograniczeń na długość przesyłanych komunikatów, dzięki czemu możliwe jest rozesłanie informacji o strukturze całej sieci w czasie $diam(G)$ rund. Ponieważ dowolne obliczenia procesor może wykonać w czasie $O(1)$, zatem wszystkie wierzchołki mogą obliczyć optymalne rozwiązanie w całym grafie G.



Rysunek 2: Przykład rozsyłania informacji posiadanych przez niebieski wierzchołek w kolejnych rundach algorytmu.

Przykład 1.10. *Mamy dany graf G jak pokazano na Rysunku 2 (w ogólności o ograniczonej średnicy). Rozwiązanie dowolnego problemu może przebiegać następująco:*

1. *Wierzchołki rozsyłają do wszystkich swoich sąsiadów informacje o znanych dla nich w danym momencie połączeniach pomiędzy wierzchołkami w grafie.*
2. *Jeżeli wykonano mniej niż $diam(G)$ rund to idź do punktu 1.*
3. *Wierzchołki obliczają rozwiązanie problemu (każdy niezależnie). Z uwagi na identyczne dane wejściowe wierzchołki otrzymują identyczne rozwiązanie.*

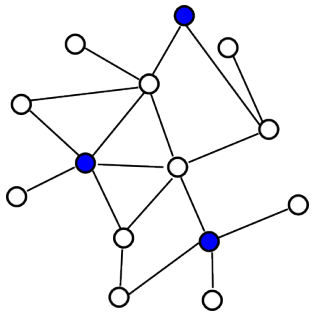
1.2 Problemy grafowe.

W rozprawie zajmuję się analizą fundamentalnych własności grafowych, takich jak: *minimalny zbiór dominujący, maksymalny zbiór niezależny, minimalne pokrycie wierzchołkowe, maksymalne skojarzenie* oraz *problem pakowania grafów* będący uogólnieniem *największego skojarzenia.*

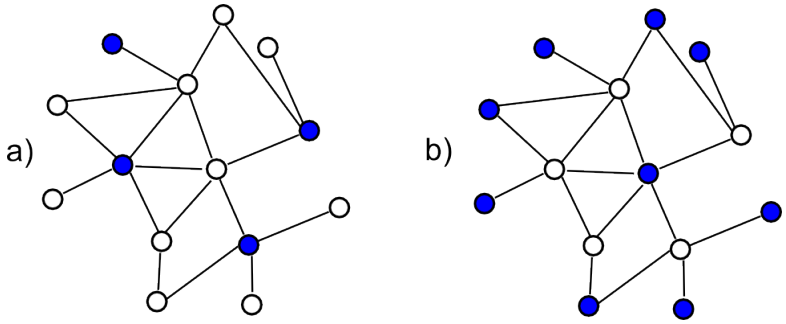
Definicja 1.11. Zbiorem niezależnym *nazywamy podzbiór wierzchołków* $I \subset V$ *grafu* $G = (V, E)$*, jeżeli*

$$\underset{v_i, v_j \in I}{\forall} v_i v_j \notin E.$$

Zbiór niezależny o największej możliwej mocy w grafie G *nazywamy* **największym zbiorem niezależnym** *i oznaczamy jako* I^**. Natomiast zbiór niezależny niebędący podzbiorem żadnego innego zbioru niezależnego nazywamy* **maksymalnym zbiorem niezależnym**.



Rysunek 3: Przykład zbioru niezależnego (wierzchołki ze zbioru zaznaczone zostały kolorem niebieskim).



Rysunek 4: Przykład: a) maksymalnego zbioru niezależnego, b) największego zbioru niezależnego

Problemy znajdowania zbiorów niezależnych w grafach są jednymi z najlepiej zbadanych. Już w końcówce lat osiemdziesiątych opublikowanych zostało wiele prac na ich temat w równoległym modelu obliczeń. Problem ten pomimo swojej prostej definicji jest z natury bardzo trudny, a jego optymalne rozwiązanie w modelu sekwencyjnym dla dowolnych grafów, w czasie wielomianowym implikuje równość $P = NP$.

Własność 1.12. *W sekwencyjnym modelu obliczeń znalezienie największego zbioru niezależnego jest problemem* NP*-zupełnym. Można łatwo udowodnić pokazując redukcje problemu spełnialności formuł logicznych (*$\mathcal{SAT}$*) do tego zagadnienia.*

Zagadnienie to można uznać za bardzo trudne do rozwiązania na jednym procesorze (praktycznie niemożliwe w czasie wielomianowym). W okresie rozwoju dziedziny algorytmów rozproszonych naturalnie postawionym pytaniem było: "jak trudne do rozwiązania jest zagadnienie w rozproszonym modelu obliczeń?". Odpowiedź została przedstawiona 1992 roku dla modelu $\mathcal{LOCAL}$.

Własność 1.13. *Rozwiązanie problemu MIS w modelu $\mathcal{LOCAL}$ wymaga co najmniej $O(\log^* n)$ rund (N.Linial 1992) ([18]).*

To dolne oszacowanie okazuje się asymptotycznie optymalne co do pewnej stałej c, z uwagi na algorytm zaproponowany w pracy R. Cole, U. Vishkin ([2]). Pokazali oni bowiem w jaki sposób w pojedynczej synchronicznej rundzie algorytmu zredukować z n do $\log n$ liczbę kolorów dla ukorzenionego drzewa. Iterując ich procedurę wielokrotnie, a następnie poddając znalezione kolorowanie pewnym modyfikacją, można uzyskać 3-kolorowanie wierzchołków w drzewie ukorzenionym w czasie $O(log^*n)$ rund.

Definicja 1.14. *Iterowany logarytm log^*n definiujemy następująco:*

$$log^*n = \begin{cases} 0 & jeżeli\ n \leqslant 1 \\ 1\ + \log^*(\log n) & jeżeli\ n > 1 \end{cases}.$$

Definicja 1.15. Zbiorem dominującym *nazywamy podzbiór wierzchołków $D \subset V$ grafu $G = (V, E)$, jeżeli*

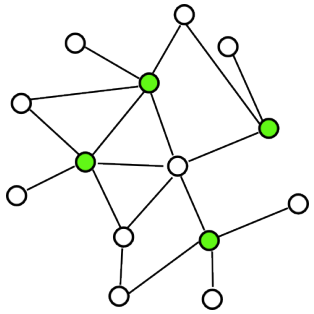
$$\underset{v_i \in V}{\forall}(v_i \in D \vee \underset{v_j \in D}{\exists} v_i v_j \in E).$$

Zbiór dominujący o najmniejszej możliwej mocy w grafie G nazywamy **minimalnym zbiorem dominującym (MDS)** *i oznaczamy jako D^*.*

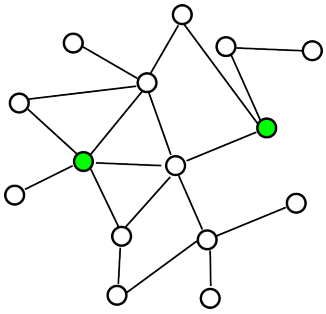
Problem ten oraz jego modyfikacja znajdują wiele zastosowań w życiu codziennym. W bezprzewodowych "ad hoc" sieciach komputerowych wykorzystuje się np. *spójne zbiory dominujące(CDS)*, które stają się szkieletem sieci i to przez nie odbywa się komunikacja pomiędzy wszystkimi wierzchołkami w grafie.

Definicja 1.16. Zbiorem l-dominującym, *gdzie $l \in \mathbb{N}_+$ nazywamy podzbiór wierzchołków $D \subset V$ grafu $G = (V, E)$, jeżeli*

$$\underset{v_i \in V}{\forall}\ \underset{v_j \in D}{\exists} d_G(v_i, v_j) \leqslant l.$$



Rysunek 5: Przykład zbioru dominującego (wierzchołki ze zbioru zaznaczone zostały kolorem zielonym).

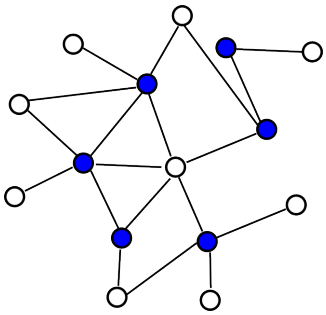


Rysunek 6: Przykład zbioru 2-dominującego (wierzchołki ze zbioru zaznaczone zostały kolorem zielonym).

Definicja 1.17. Pokryciem wierzchołkowym *w grafie* $G = (V, E)$ *nazywamy podzbiór wierzchołków* $C \in V$ *taki, że*

$$\underset{\{u,v\}\in E}{\forall} u \in C \vee v \in C.$$

Porycie wierzchołkowe o najmniejszej możliwej mocy nazywamy **minimalnym pokryciem wierzchołkowym(MVC)**.



Rysunek 7: Przykład pokrycia wierzchołkowego (wierzchołki ze zbioru zaznaczone zostały kolorem niebieskim).

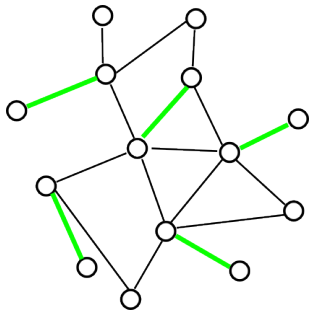
Własność 1.18. *W sekwencyjnym modelu obliczeń znalezienie minimalnego pokrycia wierzchołkowego jest problemem NP-zupełnym.*

Definicja 1.19. Skojarzeniem $M \subseteq E$ *nazywamy podzbiór krawędzi grafu* $G = (V, E)$ *taki, że żadne dwie krawędzie ze zbioru nie są do siebie przyległe, t.j.*

$$\underset{e_i, e_j \in M}{\forall} e_i \neq e_j \Rightarrow e_i \cap e_j = \emptyset.$$

Skojarzenie M *nazywamy* **maksymalnym skojarzeniem**, *jeżeli nie zawiera się w żadnym innym skojarzeniu.*

Skojarzenie M *o największej możliwej mocy nazywamy* **największym skojarzeniem** *i oznaczamy jako* M^*.



Rysunek 8: Przykład skojarzenia (krawędzie zaznaczone kolorem zielonym)

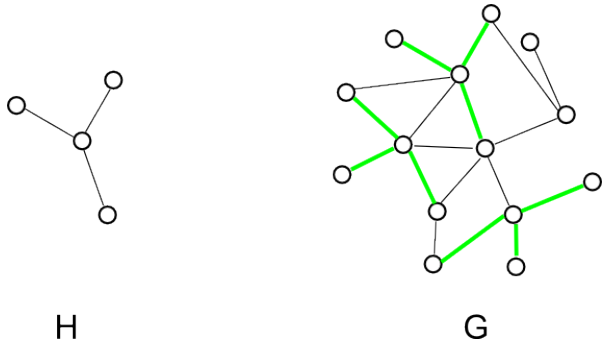
Zagadnienia maksymalnego i największego skojarzenia znajdują zastosowanie w wielu innych praktycznych problemach informatycznych, gdzie jednym z przykładów może być optymalny przydział zadań.

Własność 1.20. *W dowolnym grafie* $G = (V, E)$ *problem największego skojarzenia można rozwiązać w modelu sekwencyjnym w czasie wielomianowym (np. algorytmem z pracy [12]).*

Definicja 1.21. *Niech* H *i* G *będą dowolnymi grafami. Rodzinę grafów* $\{H_1, \ldots, H_l\}$ *nazywamy* l**-pakowaniem grafu** H **w grafie** G, *jeżeli dla każdego* $i \in \{1, \ldots, l\}$ *graf* H_i *jest izomorficzny do grafu* H. *Oraz grafy* H_i *są podgrafami grafu* G *o rozłącznych zbiorach wierzchołków (jeżeli* $i \neq j$ *to* $V(H_i) \cap V(H_j) = \emptyset$*).*

Zagadnienie pakowania grafów jest uogólnieniem problemu *największego skojarzenia*, bowiem za graf H możemy przyjąć dwa połączone krawędzią wierzchołki.

Definicja 1.22. *Niech* $G = (V, E)$ *będzie grafem planarnym.* **Funkcję wagi na wierzchołkach** *oznaczamy jako* $\omega : V \to R^+$.



Rysunek 9: Przykład 3-pakowania grafu H w grafie G. Na zielono zaznaczone zostało pakowanie.

Definicja 1.23. *Podział wierzchołków grafu $G = (V, E)$ na podzbiory $(P_1, \ldots, P_k)$ nazywamy* (α, β)**-podziałem** *(gdzie $\beta \in (0, 1)$) jeżeli spełnione są oba poniższe warunki:*

(a) $\sum_{i=1}^{k} \omega(\partial(P_i)) \leqslant \beta\omega(G)$;

(b) $\forall_i$ *diam*$(G[P_i]) \leqslant \alpha$, *gdzie diam oznacza średnicę grafu indukowanego zbiorem wierzchołków P_i,*

$\partial(P_i)$ oznacza podzbiór wierzchołków zbioru P_i posiadający sąsiedztwo w zbiorze $V \setminus P_i$.

1.3 Aproksymacja.

Dokładne rozwiązanie problemów optymalizacyjnych jest często zadaniem niewykonalnym w rozsądnym czasie (np. problemy *NP-zupełne*), z tego powodu rozpatruje się algorytmy znajdujące przybliżone rozwiązania zagadnień. Poprzez ich zastosowanie możliwe jest złamanie bariery czasowej, otrzymując rozwiązanie na przykład o mocy co najwyżej dwa razy większe niż optymalne.

Definicja 1.24. Algorytmem aproksymacyjnym *nazywamy algorytm znajdujący przybliżone rozwiązanie problemu optymalizacyjnego.*

Definicja 1.25. *Algorytm aproksymacyjny A zwracający wynik $A(x)$ dla problemu minimalizacyjnego, gdzie x jest dowolną daną wejściową, nazywamy* **algorytmem k-aproksymującym**, *jeżeli*

$$\mathrm{OPT} \leqslant A(x) \leqslant k \cdot \mathrm{OPT},$$

gdzie OPT oznacza optymalne rozwiązanie danego problemu.

Definicja 1.26. *Algorytm aproksymacyjny A zwracający wynik $A(x)$ dla problemu maksymalizacyjnego, gdzie x jest dowolną daną wejściową, nazywamy* **algorytmem k-aproksymującym**, *jeżeli*

$$k \cdot \mathrm{OPT} \leqslant A(x) \leqslant \mathrm{OPT},$$

gdzie OPT oznacza optymalne rozwiązanie danego problemu.

Definicja 1.27. *Rozproszony algorytm aproksymacyjny A nazywamy polilogarytmicznym (log-time approximation scheme), jeżeli $\underset{\epsilon>0}{\forall}$ w czasie $O(polilog)$ zwraca on poprawną $(1+\epsilon)$-aproksymacje problemu.*

Przykładem trywialnego algorytmu aproksymującego może być algorytm znajdujący minimalne pokrycie wierzchołkowe w grafie $G = (V, E)$.

W pierwszej kolejności w procedurze tej znajdujemy maksymalne skojarzenie M w grafie G, a następnie jako rozwiązanie zwracane są wszystkie wierzchołki należące do zbioru M. Algorytm ten jest algorytmem 2-aproksymującym, działającym w czasie wielomianowym pomimo, iż jest to problem NP-trudny ([22]).

2 Rozproszone algorytmy polilogarytmiczne.

W ostatnich latach zaproponowanych zostało wiele efektywnych algorytmów rozwiązujących fundamentalne problemy grafowe w pewnych klasach grafów np.: w grafach planarnych, jednostkowych grafach dyskowych(UDG)

Rozważane algorytmy wyznaczają następujące własności w grafach: *maksymalny zbiór niezależny* ([15],[25]), wielomianowa aproksymacja problemu *największego zbioru niezależnego* ([16]), *minimalny zbiór dominujący* ([4],[16]), *minimalny spójny zbiór dominujący* ([4]) oraz wiele innych.

W rozdziale tym przedstawię algorytm pakowania grafu H w planarnym grafie G.

2.1 Pakowanie grafów.

Opis algorytmu

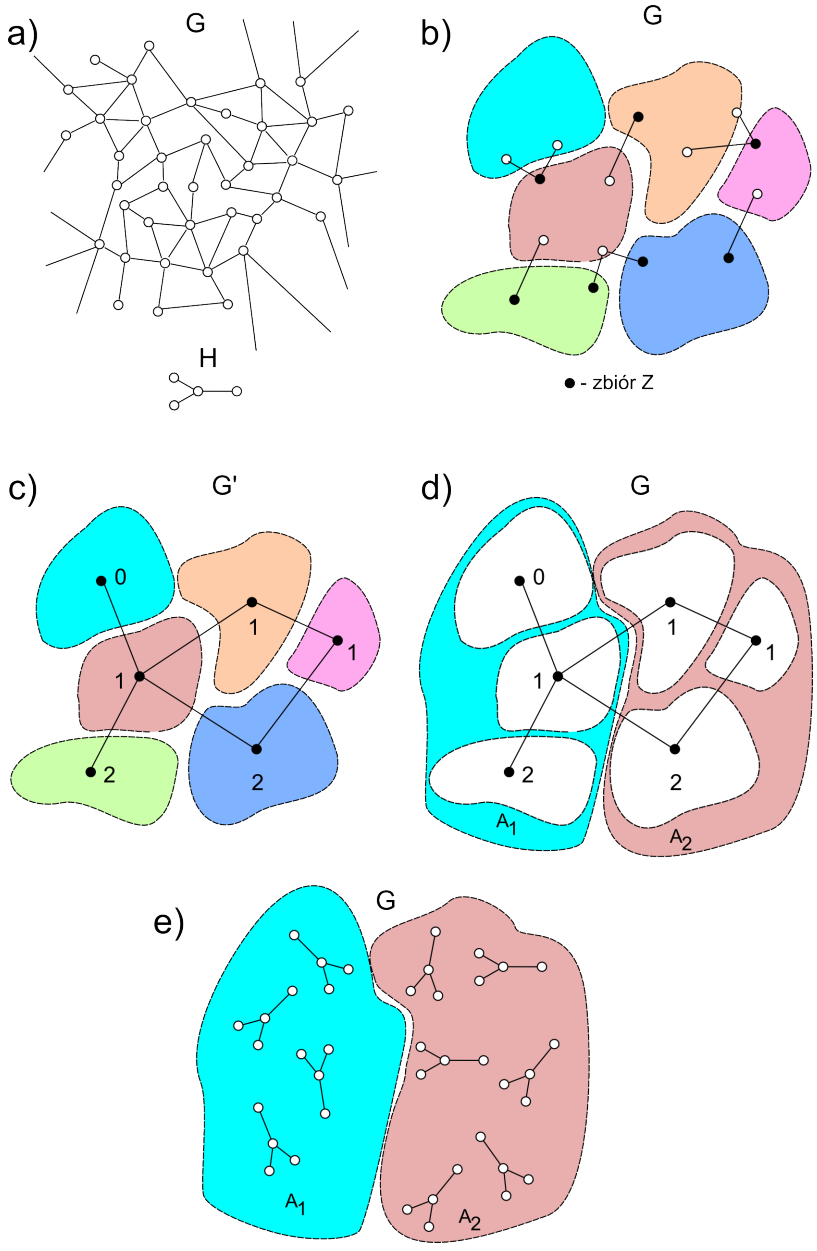
Poniżej opiszę ideę działania algorytmu pakowania. W pierwszym kroku algorytmu z grafu G usuwane są wszystkie wierzchołki, które nie mogą znaleźć się w żadnym upakowaniu grafu H w grafie G. Eliminacja ich niezbędna jest podczas oszacowania wielkości tworzonego zbioru panującego w stosunku do wielkości optymalnego pakowania grafu w PROCEDURZE KLASTROWANIA.

Następnie jak przedstawiono na Rysunku 10 wywoływana jest PROCEDURA KLASTROWANIA z parametrem $c = |H|$, wynikiem której jest podział wierzchołków z grafu G na zbiory $B = (B_1, B_2, ...B_q)$ oraz wynikowy zbiór Z. Podziały B zostały w tym kroku utworzone ze szczególną starannością, tak by w łatwy sposób można było oszacować ilość zewnętrznych wierzchołków we wszystkich podziałach razem.

W trzecim kroku procedury tworzony jest nowy graf G' poprzez ściśnięcie każdej ze składowych B_i podziału do pojedynczych wierzchołków b_i. W grafie tym każdemu w ten sposób utworzonemu wierzchołkowi przypisywana jest waga $\omega(b_i)$, będąca liczbą wierzchołków ze zbioru Z zawartą w składowej, z której wierzchołek został utworzony. Operacja ta w połączeniu z krokiem 4 ALGORYTMU PAKOWANIA służy utworzeniu nowego podziału grafu G, w ten sposób by ograniczyć z góry liczbę wierzchołków zewnętrznych w każdej składowej nowego podziału. W tym celu wykonywana jest procedura znalezienia
$O(polilog(|G|), 1/\log^{k+1}|G|)$-podziału w grafie ważonym wierzchołkowo G'.

W podziale A, skonstruowanym w tym kroku algorytmu, możliwe jest oszacowanie z góry liczby wierzchołków zewnętrznych, co więcej oszacowanie to jest odpowiednio małe. W ostatnim kroku w algorytmie znajdujemy optymalne rozwiązanie w każdej ze składo-

wych podziału A. Suma optymalnych rozwiązań jest aproksymacją optymalnego rozwiązania $\nu_H(G)$.



Rysunek 10: Przykład działania ALGORYTMU KLASTROWANIA.

Zastosowane techniki: pęki, procedura separacji, klastry ważone wierzchołkowo

Opisywanie wykorzystanych w dowodach struktur grafowych rozpocznę od pęków.

Pęki

Niech D będzie zbiorem *l-dominującym*, a rodziny zbiorów $\mathcal{P} = (P_1, P_2, \ldots, P_k)$, $\mathcal{C} = (C_1, C_2, \ldots, C_m)$ będą dwoma podziałami wierzchołków V grafu $G = (V, E)$ takimi, że:

(a) Elementy rodziny $\mathcal{C}$ (oraz $\mathcal{P}$) są parami rozłączne;

(b) $\underset{C_i \in \mathcal{C}}{\forall} \underset{P_j \in \mathcal{P}}{\exists} C_i \subset \mathcal{P}_j$;

(c) $\underset{C_i \in \mathcal{C}}{\forall}$ istnieje dokładnie jeden wierzchołek ze zbioru $d_i \in D$ oraz graf indukowany wierzchołkowo $G[C_i]$ jest spójny.

Zbiory C_i oraz P_i nazywamy odpowiednio *małymi* i *dużymi klastrami*.

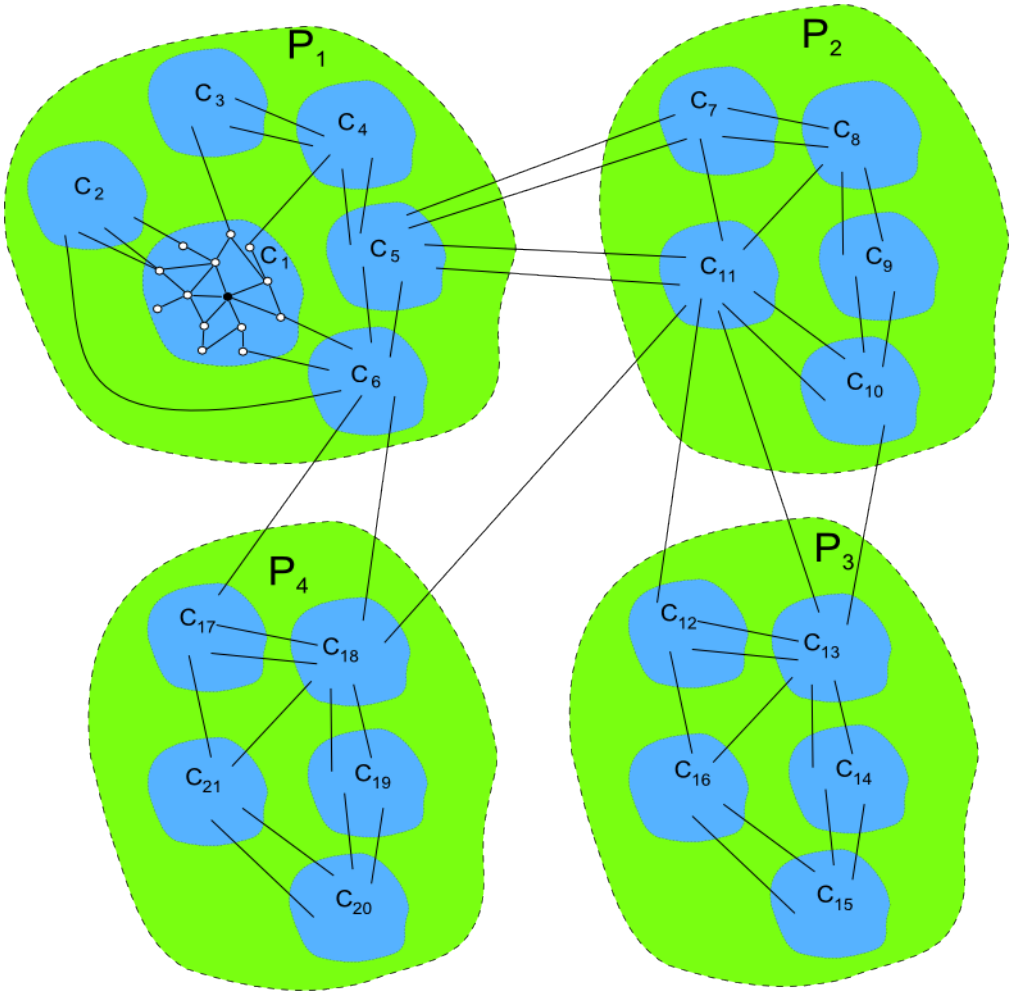
Dodatkowo zakładamy, iż w każdym podzbiorze C_i wyznaczone jest drzewo rozpinające T_i. Jego konstrukcja wymaga jedynie wywołania algorytmu przeszukiwania wszerz (tzw. *BFS*), o wierzchołku początkowym d_i.

Definicja 2.1. *Niech $d_i, d_j \in D$. Ścieżkę d_i-d_j nazywamy* **ścieżką specjalną**, *jeżeli jest postaci: $d_i P u v P' d_j$, gdzie $u \in C_i$, a $v \in C_j$. Ponadto ścieżka $P \subset T_i$ musi łączyć d_i z wierzchołkiem u, a ścieżka $P' \subset T_j$ musi łączyć d_j z wierzchołkiem v.*

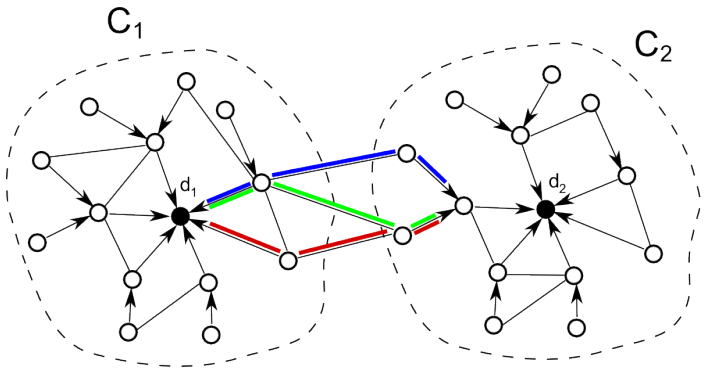
Zauważyć można, iż liczba *specjalnych ścieżek* pomiędzy wierzchołkami d_i, d_j równoliczna jest zbiorowi krawędzi pomiędzy małymi klastrami C_i i C_j. Własność ta wynika z faktu, iż dla każdej krawędzi $uv \in E(u \in C_i, v \in C_j)$ istnieje dokładnie jedna unikalna ścieżka z u do d_i w drzewie T_i oraz jedna unikalna ścieżka z v do d_j w drzewie T_j. Ścieżki specjalne nie muszą być parami rozłączne krawędziowo (jak pokazano na Rysunku 12).

Definicja 2.2. *Niech G będzie grafem płaskim zawartym w R^2. Maksymalny zbiór $f \in R^2 \setminus G$ taki, że dowolne dwa punkty w f mogą być połączone przez krzywą zawartą w f nazywamy* **ścianą grafu** *G.*

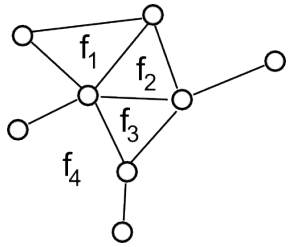
Własność 2.3. *Niech P i Q będą różnymi specjalnymi ścieżkami, pomiędzy wierzchołkami d_i oraz d_j. Dowolny podgraf indukowany grafu płaskiego $G[P \cup Q]$ zawiera dokładnie jedną ograniczoną ścianę.*



Rysunek 11: Przykład małych(C_i) oraz dużych(P_i) klastrów. W klastrze małym C_1 wierzchołek d_1 zaznaczony został kolorem czarnym.

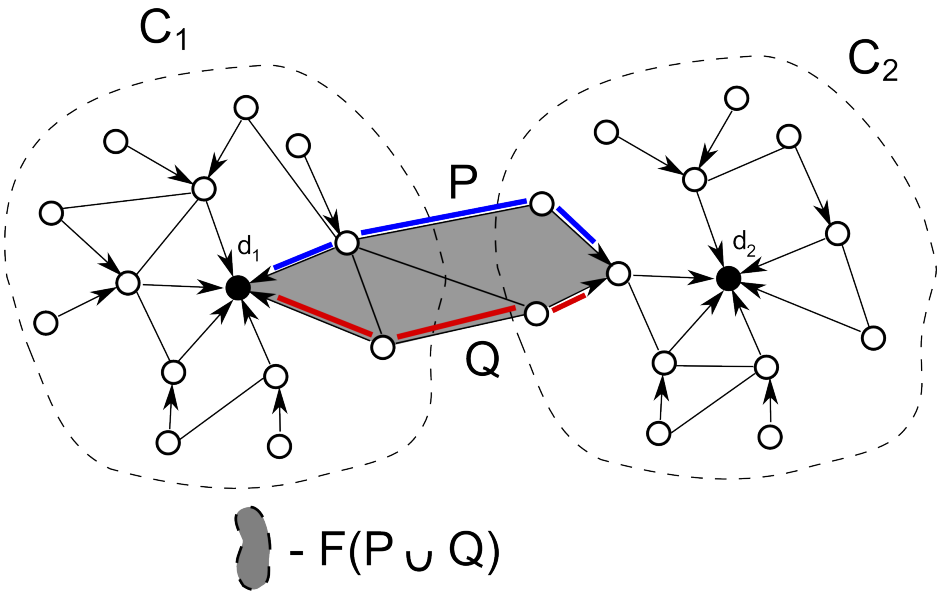


Rysunek 12: Przykład trzech ścieżek specjalnych pomiędzy wierzchołkami d_1 oraz d_2. Każda ścieżka zaznaczona została innym kolorem (odpowiednio zielonym, niebieskim i czerwonym).



Rysunek 13: Przykład grafu płaskiego o czterech ścianach: f_1, f_2, f_3, f_4.

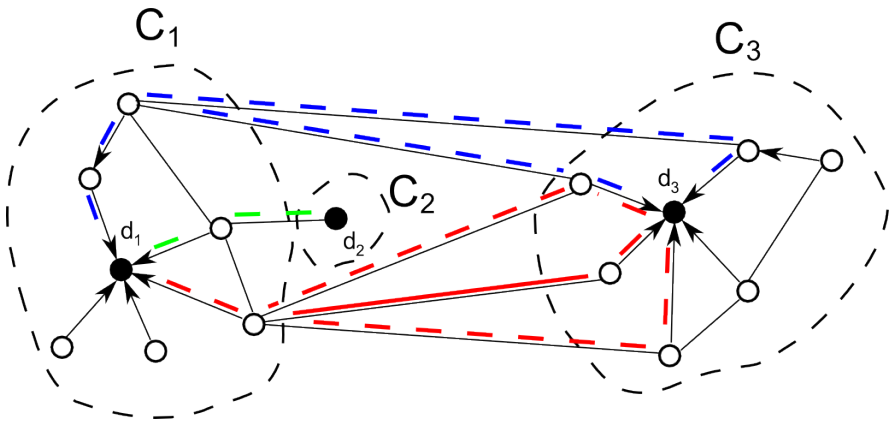
Definicja 2.4. *Oznaczmy* $F(P \cup Q) = f$ *oraz* $Reg[P \cup Q] = (P \cup Q) \cup f$, *gdzie* f *jest ograniczoną ścianą w podgrafie* $G[P \cup Q]$.



Rysunek 14: Przykład zbiorów $F(P \cup Q)$ oraz $Reg[P \cup Q]$.

Definicja 2.5. *Niech* $G = (V, E)$ *będzie grafem płaskim, ponadto niech* $d_i, d_j \in D$ *(gdzie* $i \neq j$*). Maksymalny zbiór* $\mathcal{B}$ *specjalnych ścieżek pomiędzy wierzchołkami* d_i, d_j *nazywamy* d_i**-**d_j **pękiem**, *jeżeli istnieją dwie (niekoniecznie rozłączne) ścieżki* $P, Q \in \mathcal{B}$ *takie, że wszystkie ścieżki ze zbioru* $\mathcal{B}$ *zawarte są w* $Reg[P \cup Q]$ *oraz żaden wierzchołek ze zbioru dominującego* D *nie jest zawarty w* $F(P \cup Q)$. *Ścieżki* P *i* Q *nazywamy* **zewnętrznymi specjalnymi ścieżkami** *w grafie* G.

Zauważmy, iż możliwe jest istnienie wielu pęków pomiędzy tymi samymi małymi klastrami.



Rysunek 15: Przykład trzech pęków oznaczonych kolorami: niebieskim, czerwonym i zielonym. *Specjalne ścieżki zewnętrzne* oznaczone zostały przerywanymi liniami.

Fakt 2.6. *Niech G będzie grafem płaskim, $d_i, d_j \in D$ $(i \neq j)$ oraz $\mathcal{A}$ i $\mathcal{B}$ będą dwoma d_i-d_j pękami. Jeżeli $\mathcal{A} \neq \mathcal{B}$ wtedy $\mathcal{A} \cap \mathcal{B} = \emptyset$ i dla każdej specjalnej ścieżki $P \in \mathcal{A}$ oraz każdej specjalnej ścieżki $Q \in \mathcal{B}$ istnieje wierzchołek ze zbioru D zawarty w $F(P \cup Q)$.*

Dowód:

Załóżmy, że $\mathcal{A} \neq \mathcal{B}$, wtedy z maksymalności pęków otrzymujemy $\mathcal{A} \cap \mathcal{B} = \emptyset$, bowiem w przeciwnym przypadku $\mathcal{A} = \mathcal{B}$. Ponieważ $P \in \mathcal{A}$ i $Q \in \mathcal{B}$ to ściana $F(P \cup Q)$ jest niepusta (w jej obszarze zwarty jest wierzchołek ze zbioru D), w przeciwnym przypadku z maksymalności pęków otrzymalibyśmy, że P i Q należą do tego samego pęku.

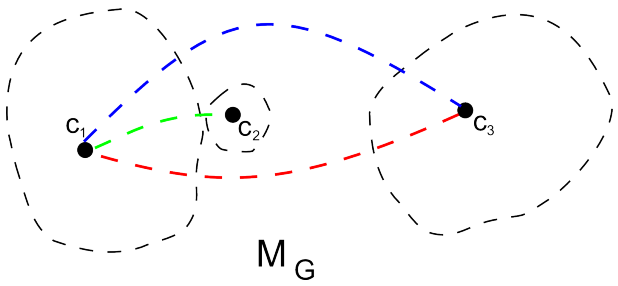
□

Niech **B** będzie zbiorem wszystkich pęków w grafie planarnym G. Multigraf otrzymany z grafu G przez ściśnięcie małych klastrów C_i do wierzchołków c_i oraz dodaniu krawędzi pomiędzy c_i, c_j dla każdego d_i-d_j pęku, oznaczmy jako M_G. Liczba wszystkich krawędzi w multigrafie M_G pomiędzy wierzchołkami c_i i c_j równa jest liczbie d_i-d_j pęków. W rezultacie otrzymujemy

$$|B| = |E(M_G)|. \tag{1}$$

Możemy łatwo zauważyć, że multigraf M_G jest zawsze planarny. Przyjmijmy dla uproszczenia, iż jest to płaski multigraf, z wierzchołkami znajdującymi się w tej samej pozycji w R^2 co wierzchołki d_i w grafie płaskim G. Przy takich założeniach prawdziwy jest następujący fakt:

Fakt 2.7. *Niech $e, e' \in E(M_G)$ będą dwom różnymi krawędziami pomiędzy wierzchołkami u i v w multigrafie M_G, wtedy istnieje wierzchołek $w \in V(M_G)$ zawarty w obszarze ograniczonym cyklem $ueve'u$.*

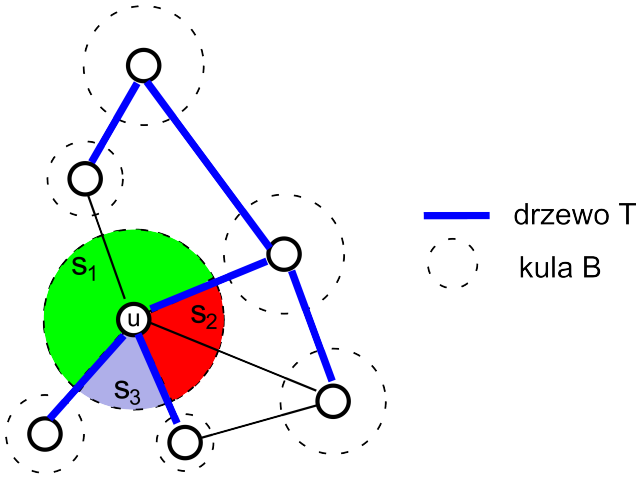


Rysunek 16: Przykład grafu M_G utworzonego z grafu G przedstawionego na Rysunku 15.

Fakt ten wynika bezpośrednio z faktu 2.6. Jako T oznaczmy dowolne drzewo rozpinające w multigrafie M_G.

Definicja 2.8. *Rozważmy płaską reprezentację grafu M_G i dla każdego wierzchołka $v \in V(T)$ o $\epsilon_v > 0$ oznaczmy jako B_v kulę wokół wierzchołka v o promieniu ϵ_v. Kule te wybieramy tak, by przecinały się tylko z krawędziami incydentnymi z v i nie posiadały punktów wspólnych z innymi kulami. Spójne regiony $B_v \setminus T \subseteq R^2$ nazywamy* **stronami wierzchołka** *v.*

Tak więc każda krawędź $uv \in E(M_G) \setminus E(T)$ leży po pewnej stronie s wierzchołka v.



Rysunek 17: Kolorami zielonym, czerwonym oraz niebieskim zaznaczone zostały trzy strony wierzchołka u odpowiednio s_1, s_2, s_3.

Lemat 2.9. *Niech $u, v \in V(M_G)$ wtedy istnieją co najwyżej dwie krawędzie $e, e' \in E(M_G) \setminus E(T)$ pomiędzy wierzchołkami u, v leżące po tej samej stronie wierzchołka v i wierzchołka u.*

Dowód:

Niech F będzie zbiorem $u - v$ krawędzi takim, że $F \subset E(M_G) \setminus E(T)$ oraz krawędzie leżą po tej samej stronie u i tej samej stronie v. Jeżeli wybierzemy dowolną krawędź $e \in F$, to $C = uTv + e$ jest cyklem w multigrafie M_G. Jeżeli $\bar{e} \in F$ zawiera się w ograniczonej ścianie cyklu C, to z faktu 2.7 otrzymujemy, że istnieje wierzchołek $z \in V(T)$ wewnątrz ograniczonej ściany $uev\bar{e}u$. Wierzchołek ten połączony jest z wierzchołkiem u albo z wierzchołkiem v. Bez straty ogólności możemy założyć, że istnieje połączenie zTu zawarte w ograniczonej ścianie $uev\bar{e}u$. Stąd e i $\bar{e}$ leżą po innych stronach wierzchołka u.

□

Fakt 2.10. *Niech G będzie spójnym grafem płaskim, a moc zbioru dominującego $|D| \geqslant 2$. Wtedy $|M_G| = |D|$ oraz $|E(M_G)| \leqslant 18|D| - 24$.*

Dowód:

Liczba wierzchołków w grafie M_G równa jest mocy zbioru D. Skoro graf G jest spójny to spójny jest również multigraf M_G, a tym samym posiada drzewo rozpinające T. Niech H będzie tzw. **supergrafem** T otrzymanym w następujący sposób:

- dla każdego wierzchołka $v \in V(T)$ dodajemy wierzchołek w_v po każdej stronie wierzchołka v i łączymy go z wierzchołkiem v;
- zastępujemy każdą krawędź $e \in E(M_G) \setminus E(T)$ leżącą po stronie wierzchołka w_v krawędzią kończącą się w wierzchołku w_v.

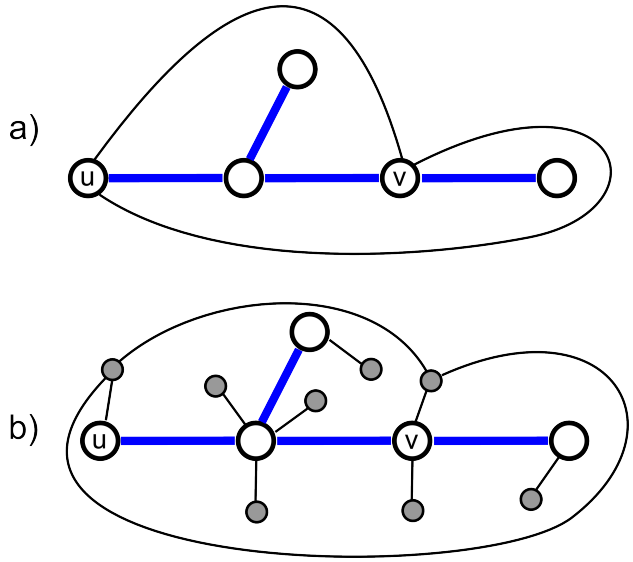
Wtedy H jest planarnym multigrafem o $|E(H)| = |E(M_G)|$. Z Lematu 2.9 otrzymujemy, że istnieją maksymalnie dwie krawędzie pomiędzy dowolnymi dwoma wierzchołkami w grafie H. Ponieważ $|V(H)| = |V(T)|$ to $\sum d_T(v) = 3|T| - 2 = 3|M_G| - 2$. Korzystając następnie ze wzoru Eulera otrzymujemy:

$$|E(M_G)| = |E(H)| \leqslant 6|H| - 12 = 18|M_G| - 24.$$

□

Procedura separacji

Przyjmijmy oznaczenia podziałów $\mathcal{P} = (P_1, P_2, \ldots, P_k), \mathcal{C} = (C_1, C_2, \ldots, C_m)$ oraz drzewa rozpinającego T zgodnie z poprzednim podrozdziałem. Celem działania procedury separacji, opisywanej w tym podrozdziale, jest znalezienie podziałów o pewnych specjalnych własnościach.



Rysunek 18: Przykład supergrafu H.

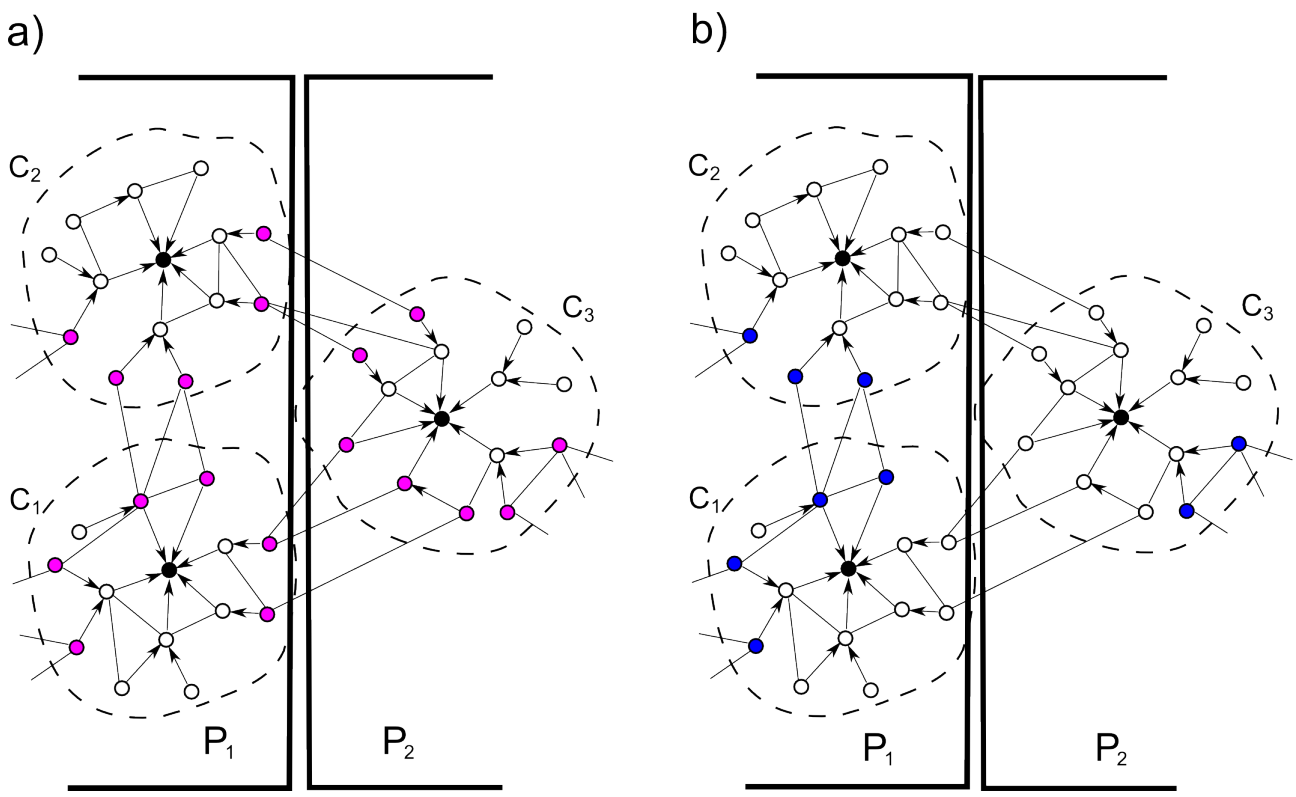
Definicja 2.11. *Niech $\mathcal{P} = (P_1, P_2, \ldots, P_k)$, $\mathcal{C} = (C_1, C_2, \ldots, C_m)$ będą dwoma podziałami wierzchołków. Załóżmy, że $C_i \subseteq P_j$, wtedy:*

(a) $\partial(C_i)$ oznacza zbiór wierzchołków w małym klastrze C_i takich, że posiadają one sąsiedztwo w zbiorze $V \setminus C_i$;

(b) $\partial_{IN}(C_i)$ oznacza podzbiór wierzchołków $\partial(C_i)$ takich, że posiadają one sąsiedztwo w zbiorze $P_j \setminus C_i$;

(c) $\partial_{OUT}(C_i)$ oznacza podzbiór wierzchołków $\partial(C_i)$ takich, że posiadają one sąsiedztwo w zbiorze $V \setminus P_j$;

(d) $\partial^*(P_j)$ oznacza rodzinę zbiorów $\{C_i | C_i \subseteq P_j$ i $\partial_{OUT}(C_i) \neq \emptyset\}$.



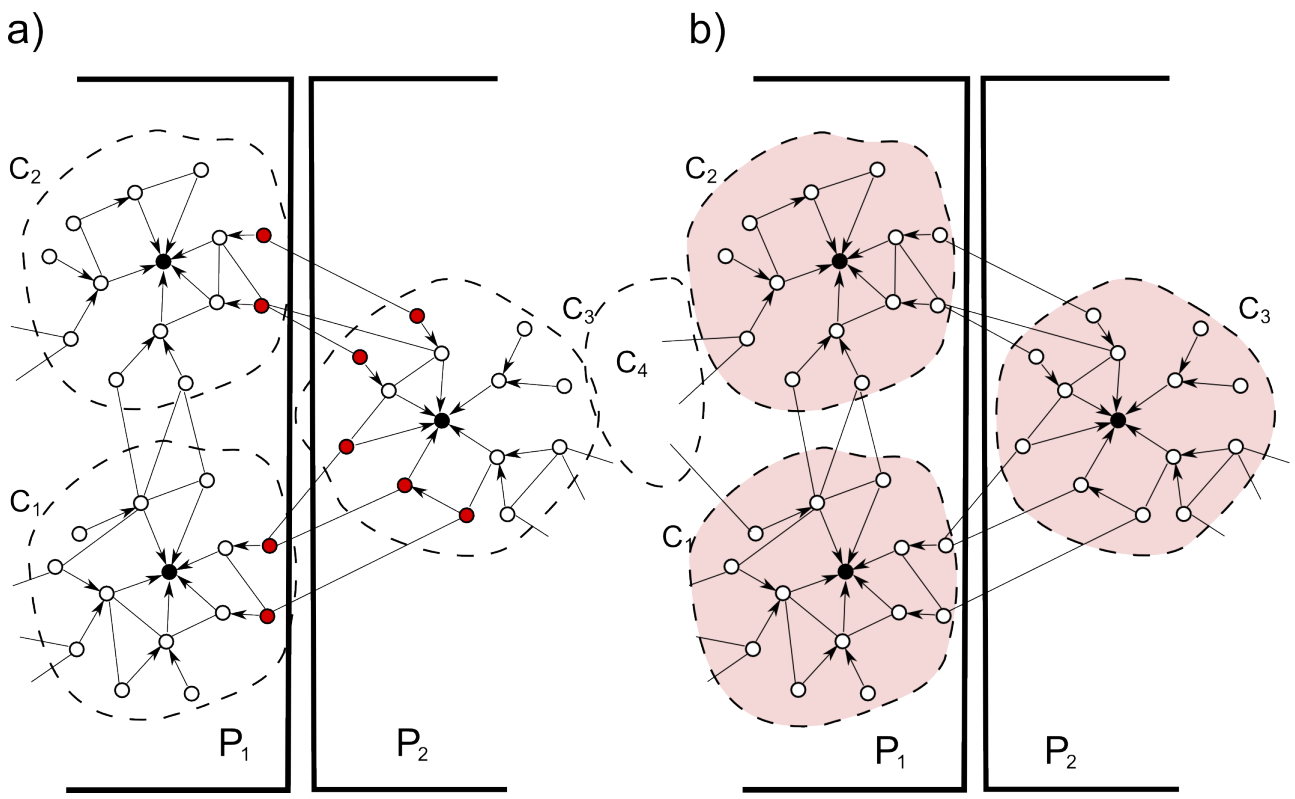
Rysunek 19: Przykład zbiorów: a) $\partial(C_i)$, b) $\partial_{IN}(C_i)$.

Procedura Separacji

Wejście: Spójny graf planarny G, zbiór l-dominujący $D = \{d_1, \ldots, d_l\}$, podziały $\mathcal{P} = (P_1, \ldots, P_k)$, $\mathcal{C} = (C_1, \ldots, C_l)$ i $(T_1, \ldots, T_l)$, gdzie T_i jest drzewem *BFS* w grafie indukowanym $G[C_i]$ o korzeniu d_i.

Wyjście: Zbiór $Z \subset V(G)$ oraz parami rozłączne podzbiory $(B_1, \ldots, B_m)$ takie, że $G[B_i]$ jest spójny.

1. $\forall\ v \in V(G) \setminus D$ ustaw zmienną $kolor[v] = biały$.
2. $\forall P_j$ i $\forall C_i \in \partial^*(P_j)$ wykonuj równolegle:
   (a) $\forall v \in (\partial_{IN}(C_i) \setminus \{d_i\})$ ustaw zmienną $kolor[v] = niebieski$. Ponadto, jeżeli wierzchołek w (taki, że $w \neq d_i$) leży na unikalnej ścieżce vT_id_i, to ustaw $kolor[w] = niebieski$.
   (b) $\forall v \in (\partial_{OUT}(C_i) \setminus \{d_i\})$ ustaw zmienną:
   $kolor[v] = zielony$, jeżeli $v \in \partial_{IN}(C_i) \cap \partial_{OUT}(C_i)$,
   $kolor[v] = czerwony$ w przeciwnym przypadku.



Rysunek 20: Przykład zbiorów: a) $\partial_{OUT}(C_i)$, b) $\partial^*(P_j)$.

Powtarzaj następujące korki w C_i do momentu ustabilizowania się kolorów w małym klastrze:

(ba) Jeżeli $kolor[v] = zielony$ oraz wierzchołek $w \neq d_i$ leży na unikalnej ścieżce vT_id_i, to ustaw $kolor[w] = zielony$.

(bb) Jeżeli $kolor[v] = biały$ i $\exists w \in N(v)$ taki, że $kolor[w] = czerwony$, to ustaw $kolor[v] = czerwony$.

(bc) Jeżeli $kolor[v] = niebieski$ i $\exists w \in N(v)$ taki, że $kolor[w] = czerwony$, to ustaw $kolor[v] = zielony$.

3. Niech Z będzie zbiorem wierzchołków v takich, że ich $kolor[v] = zielony$. Dla dużego klastra P_j niech Niebieskie$_j$ (Białe$_j$) będą zbiorem niebieskich(białych) wierzchołków ze zbioru P_j. Ponadto niech Czarne$_j$ będzie podzbiorem wierzchołków $(D \cap P_j) \setminus \partial^*(P_j)$, a $int(P_j) =$ Niebieskie$_j \cup$ Białe$_j \cup$ Czarne$_j$. Jako $int(P_j^1), \ldots, int(P_j^{l_j})$ oznaczmy spójne składowe w indukowanym podgrafie $G[int(P_j)]$.

4. Zwróć Z oraz $(int(P_1^1), \ldots, int(P_k^{l_k}))$.

Lemat 2.12. *Niech $\bar{G}$ będzie dowolną płaską reprezentacją grafu planarnego G. Jeżeli po zakończeniu działania* PROCEDURY SEPARACJI *wierzchołek $v \in V(G)$ posiada $kolor[v] =$ zielony, to leży on na zewnętrznej ścieżce specjalnej pewnego pęku $\mathcal{B}$ w grafie płaskim $\bar{G}$.*

Dowód:

Załóżmy nie wprost, że istnieje wierzchołek $v \in C_i \subset P_t$ o kolorze zielonym i nie leży on na żadnej *zewnętrznej ścieżce specjalnej*.

W pierwszej kolejności pokażemy, że wierzchołek v musi leżeć na ścieżce specjalnej. Skoro $kolor[v] = zielony$, to dla pewnego wierzchołka $v' \in C_i$, vT_iv' jest ścieżką (należy przy tym zauważyć, iż istnieje możliwość $v = v'$). Wynika to z faktu, iż wierzchołek v pokolorowany może być na kolor zielony tylko w następujących krokach algorytmu:

- gdy $v \in (\partial_{OUT}(C_i)\{d_i\})$ - punkcie 2(b) algorytmu;
- gdy v pokolorowany został jako część ścieżki w punkcie 2(ba);
- gdy v pokolorowany został w punkcie 2(bc), ale to oznacza, iż był koloru niebieskiego po kroku 2(a). Możliwe było to tylko wtedy, gdy istniał wierzchołek v' i ścieżka vT_iv'.

Rozpatrywać możemy dwa przypadki (1) $v' \in \partial_{IN}(C_i) \cap \partial_{OUT}(C_i)$ albo (2) $kolor[v'] =$ niebieski w kroku 2(a) PROCEDURY SEPARACJI. W pierwszym przypadku v' posiada sąsiada $w \in C_j \subset P_s$, gdzie $s \neq t$ i wtedy $d_iT_iv'wT_jd_j$ jest specjalną ścieżką zawierającą wierzchołek v. W drugim przypadku wierzchołek v' posiada sąsiada $w \in C_j \subset P_t$, gdzie $j \neq i$, a więc $d_iT_iv'wT_jd_j$ jest specjalną ścieżką zawierającą v.

Następnie udowodnię, że wierzchołek v leży na *zewnętrznej ścieżce specjalnej*. Załóżmy, że v leży na specjalnej $d_i - d_j$ ścieżce $P' \in \mathcal{B}$ i niech $P, Q \in \mathcal{B}$ będą dwiema zewnętrznymi ścieżkami pęku. Z założenia nie wprost wiemy, że $v \notin V(P) \cup V(Q)$, a skoro P, Q są zewnętrznymi ścieżkami to wierzchołek v zawarty jest w ograniczonym regionie $P \cup Q$. W rezultacie wszyscy sąsiedzi wierzchołka v leżą w $Reg[P \cup Q]$, tym samym żadna ścieżka postaci wT_id_i z $w \notin F[P \cup Q]$ nie zawiera v. W pierwszej kolejności załóżmy, że drugi koniec P' leży w innym dużym klastrze ($d_j \in P_s$, gdzie $s \neq t$). Jako, że $kolor[v] = zielony$, to istnieje wierzchołek w $F[P \cup Q] \cap C_i$ o kolorze niebieskim w kroku 2(a), a więc zbiór $F[P \cup Q] \cap D$ nie jest pusty. Rozumowanie to prowadzi do sprzeczności z definicją pęku, a więc w tym przypadku ścieżki P i Q nie mogą należeć do jednego pęku $\mathcal{B}$. Teraz załóżmy, że $d_j \in C_j \subseteq P_t$, wtedy wszystkie wierzchołki ze zbioru $(V(P) \cup V(Q)) \cap C_i$, poza wierzchołkiem d_i, kolorowane są na *niebiesko* w korku 2(a) i w konsekwencji wierzchołek v nigdy nie będzie czerwony. Stąd

wiemy, by wierzchołek v uzyskał kolor *zielony* wymagane jest istnienie wierzchołka czerwonego w zbiorze $F[P \cup Q]$, a więc $F[P \cup Q] \cap D$ nie jest puste. Uzyskujemy więc sprzeczność z definicją pęku, co kończy dowód.

□

Poprawność lematu po zrozumieniu idei działania procedury separacji nie budzi żadnych wątpliwości. Algorytm ten w pierwszym kroku koloruje bowiem wierzchołki leżące na *ścieżkach specjalnych* pomiędzy małymi klastrami, w obrębie jednego P_j, kolorem niebieskim. Następnie wszystkie wierzchołki z małego klastra połączone z innym dużym klastrem kolorowane są na kolor czerwony. Kolor czerwony "rozprzestrzenia" się na wszystkie sąsiednie wierzchołki do czasu "dotknięcia" koloru niebieskiego. Styk dwóch kolorów oznaczany jest kolorem zielonym, a więc należy on do pewnej ścieżki specjalnej. Co więcej stykiem może być jedynie *zewnętrzna ścieżka specjalna*.

Lemat 2.13. *Niech G będzie spójnym grafem planarnym, a D będzie zbiorem l-dominującym o mocy $|D| \geqslant 2$, wtedy zbiór Z zwracany przez* PROCEDURĘ SEPARACJI *posiada moc:*

$$|Z| \leqslant 72 \cdot l|D| - 96 \cdot l.$$

Dowód:

Ponieważ D jest zbiorem l-dominującym, to każda specjalna ścieżka posiada co najwyżej $2l$ wierzchołków. W płaskiej reprezentacji grafu G dowolny pęk posiada co najwyżej dwie zewnętrzne ścieżki specjalne, a z Lematu 2.12 wiemy, że każdy zielony wierzchołek leży na ścieżce zewnętrznej. Otrzymujemy stąd, że liczba zielonych wierzchołków wynosi co najwyżej $4l|\mathbf{B}|$, gdzie $\mathbf{B}$ jest zbiorem wszystkich pęków w płaskiej reprezentacji grafu G. Z faktu 2.10 i równania (1) wynika, że

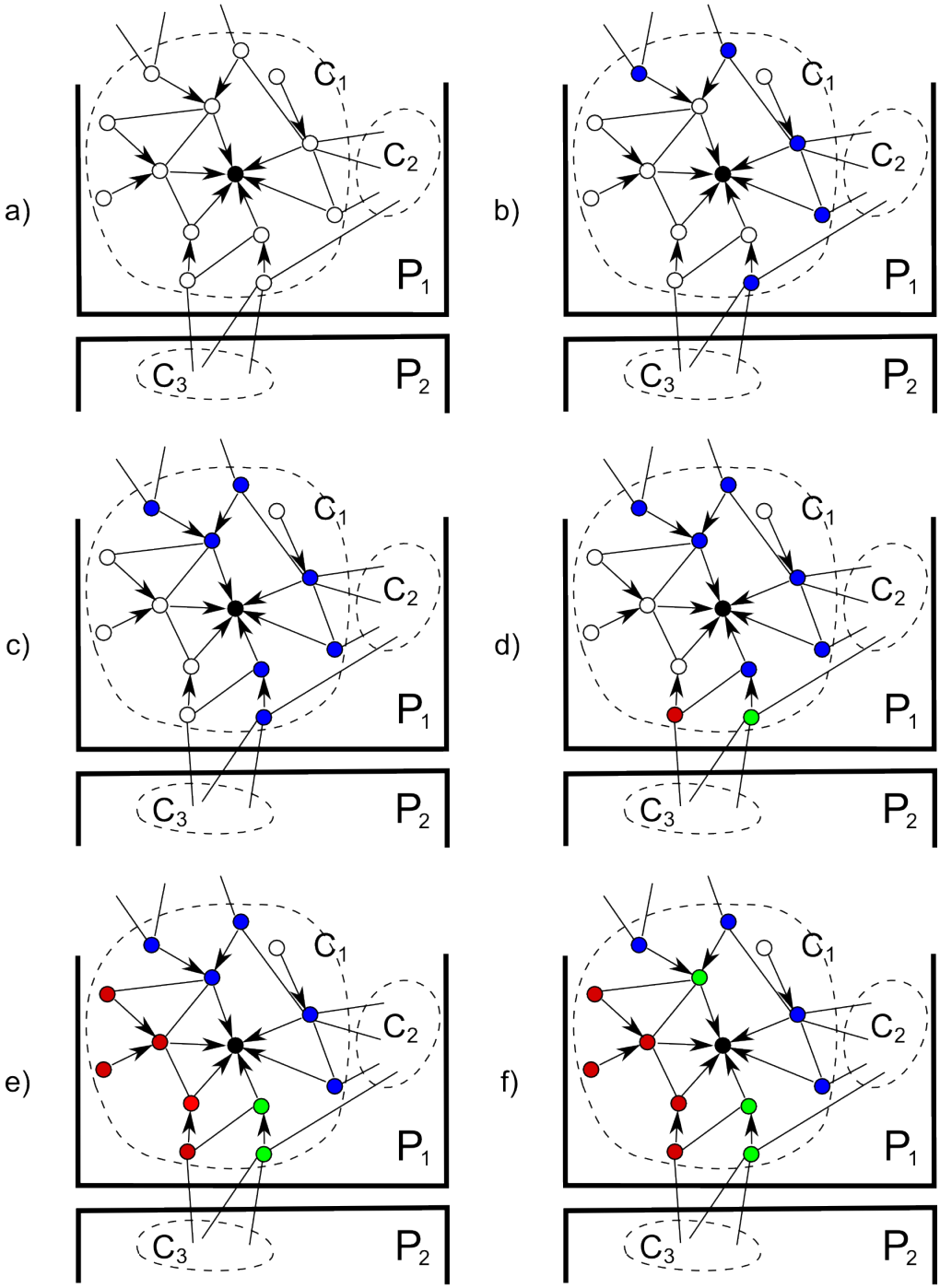
$$|Z| \leqslant 4l(18|D| - 24) = 72 \cdot l|D| - 96 \cdot l.$$

□

Lemat 2.14. *Każda krawędź e mająca dokładnie jeden koniec w zbiorze wierzchołków $int(P_j)$ drugi koniec posiada w zbiorze $Z \cup D$.*

Dowód:

Niech $e = \{u, v\}$ będzie taką krawędzią, że $v \in int(P_j)$, a $u \notin int(P_j)$. Jeżeli wierzchołek $v \notin \partial^*(P_j)$, to wtedy $u \in C_i$ dla pewnego $C_i \in \partial^*(P_j)$. W kroku 2(a) PROCEDURY SEPARACJI



Rysunek 21: Przykład działania PROCEDURY SEPARACJI.

albo $u = d_i$, albo $kolor[u] =$ niebieski. Skoro niebieskie wierzchołki należą do zbioru $int(P_j)$ to kolor wierzchołka u nie może być niebieski po zakończeniu działania procedury, w takim razie musi być zielony. Otrzymujemy stąd, że $u \in Z \cup D$.

W drugim przypadku, gdy wierzchołek $v \in \partial^*(P_j)$ z definicji algorytmu wiemy, że po

zakończeniu działania procedury $kolor[v]$ = niebieski lub $kolor[v]$ = biały. Wierzchołek v nie może wobec tego posiadać sąsiada w kolorze czerwonym, w przeciwnym przypadku zmieniłby swój kolor w kroku 2(b) procedury. Ponieważ wierzchołek $u \notin int(P_j)$, to $kolor[u]$ nie może być ani biały ani niebieski, ponadto nie może być też czerwony, zatem wierzchołek u jest koloru zielonego lub czarnego. Po połączeniu obu rozpatrywanych przypadków otrzymujemy, że $u \in Z \cup D$.

Klastry ważone wierzchołkowo

Jedną ze znanych technik aproksymacyjnego rozwiązywania różnych problemów optymalizacyjnych jest podział grafu na klastry.

Polega ona na odpowiednim podziale wierzchołków w grafie na rozłączne wierzchołkowo podzbiory, w skrócie nazywane klastrami. Następnie w każdym podzbiorze z osobna wykonywana jest procedura znajdująca optymalne rozwiązanie problemu, a suma rozwiązań, po ewentualnych poprawkach jest aproksymacją optymalnego rozwiązania problemu. Własność tę uzyskuje się poprzez wymuszenie pewnych własności w budowanych klastrach.

Modyfikacja owej techniki została wykorzystana również w rozprawie.

Definicja 2.15. *Podzbiór wierzchołków $S \subset V$ nazywany jest* (a, b)**-zbiorem panującym** *jeżeli jest zbiorem b-dominującym oraz $\forall_{v,v' \in S}$, gdzie $v \neq v'$ odległość pomiędzy dwoma wierzchołkami wynosi co najmniej a, czyli $d(v, v') \geqslant a$.*

W literaturze znana jest procedura ([1]) znajdująca $(k, O(k \log |G|))$-zbiór panujący w $O(k \log |G|)$ rundach. Otrzymany zbiór S użyty został do konstrukcji podziału zbioru wierzchołków V na podzbiory $\mathcal{C}$ opisywane we wcześniejszym podrozdziale. Każdy wierzchołek ze zbioru $V \setminus S$ "dołącza się" do najbliższego wierzchołka ze zbioru S tworząc klaster. W sytuacji, gdy kilka wierzchołków leży w tej samej odległości wybierany jest dowolny wierzchołek (przyjmijmy ten o najniższym *ID*).

Każdy z otrzymanych w ten sposób podzbiorów $(C_1, \ldots, C_{|S|})$ zawiera dokładnie jeden wierzchołek ze zbioru S. Dodatkowo dla każdego zbioru wierzchołków C_i indukowany wierzchołkowo podgraf $G[C_i]$ jest spójny oraz odległość dowolnego wierzchołka $v \in C_i$ od wierzchołka S wynosi co najwyżej b.

Skonstruowanie podziału $\mathcal{P}$ wymaga dodatkowo użycia dwóch procedur z prac [4] oraz [7], których opis zostanie tutaj pominięty.

Definicja 2.16. *Niech $G = (V, E)$ będzie grafem planarnym.* **Funkcją wagi** *nazywamy funkcję $\omega : V \to R^+ \cup \{0\}$.*

Definicja 2.17. *Niech $G = (V, E, \omega)$ będzie grafem planarnym, gdzie ω jest funkcją wagi. Podział $(P_1, \ldots, P_k)$ wierzchołków V nazywamy (α, β)-**podziałem ważonym wierzchołkowo** (w skrócie podziałem) jeżeli:*

(a) $\sum_{i=1}^{k} \omega(\partial(P_i)) \leqslant \beta\omega(V(G))$ *oraz*

(b) $\forall_i$ *słaba średnica* $weak_G diam(G[P_i])$ *grafu indukowanego wierzchołkowo* $G[P_i]$ *wynosi co najwyżej* α *i podgraf ten jest spójny.*

W pracy [7] pokazano, że w grafie planarnym $G = (V, E)$ o ograniczonej liczbie wierzchołków przez m ($|V| \leqslant m$) można otrzymać $(polilog(m), 1/\log^{\Theta(1)} m)$-podział ważony wierzchołkowo w czasie $polilog(m)$ rund. Ponadto z pracy [4] wiemy, że jeżeli $\omega(v) \in \{0, 1\}$ to $(O(1), 1/2)$-podział skonstruować można w czasie polilogarytmicznym ($polilog(m)$) i jednocześnie zagwarantować, by średnica każdego indukowanego wierzchołkowo grafu wynosiła co najwyżej α.

Procedura Klastrowania

Wejście: Planarny graf G oraz $c \geqslant 1$.

Wyjście: Podział $\mathcal{B}$ wierzchołków grafu $V(G)$ oraz podzbiór wierzchołków $Z \subset V(G)$.

1. Znajdź $(2c+1, O(c \log |G|))$-zbiór panujący D w grafie G i oznacz $D^{(0)} = D, G^{(0)} = G$.
2. Od $a = 1$ do $O(\log |G|)$ wykonaj:
   (a) Znajdź podziały grafu $G^{(a-1)}$ na małe klastry $(C_1^{(a)}, \ldots, C_{l_a}^{(a)})$ wokół zbioru $D^{(a-1)}$. Następnie ściśnij każdy mały klaster C_i do pojedynczego wierzchołka c_i w celu uzyskania grafu $F^{(a-1)}$. Ustal wagę każdego wierzchołka $v \in F^{(a-1)}$ równą 1.
   (b) Wywołaj procedurę z pracy [4] znajdującą $(O(1), 1/2)$-podział $\bar{\mathcal{P}}$ w grafie $F^{(a-1)}$. Niech $(P_1^{(a)}, \ldots, P_{s_a}^{(a)})$ będzie podziałem grafu $G^{(a-1)}$ otrzymanego z $\bar{\mathcal{P}}$ za pomocą operacji odwrotnej do ściskania wierzchołków ze zbioru $F^{(a-1)}$.
   (c) Użyj procedury separacji na podziałach $(P_1^{(a)}, \ldots P_{s_a}^{(a)})$, $(C_1^{(a)}, \ldots, C_{l_a}^{(a)})$ oraz drzewach BFS $T_i^{(a)}$ ukorzenionych w wierzchołkach $d_i \in D^{(a-1)} \cap C_i^{(a)}$. Procedura separacji zwraca zbiór $Z^{(a)}$ i rodzinę zbiorów $\{B_i^{(a)}\}$.
   (d) Niech $V = V \setminus \bigcup_i B_i^{(a)}$, $G^{(a)} = G[V]$, $D^{(a)} = D^{(a-1)} \cap V$.
3. Niech $Z = \bigcup(Z^{(a)} \cup D^{(a)})$. Zwróć Z i kolekcję wszystkich zbiorów $B_i^{(a)}$ dla wszystkich a oraz i.

Lemat 2.18. *Po każdej iteracji pętli z kroku drugiego* PROCEDURY KLASTROWANIA *spełnione są następujące warunki:*

(a) każda krawędź w grafie $G^{(a)}$ o dokładnie jednym końcu w zbiorze wierzchołków $B_i^{(a)}$ posiada drugi koniec w zbiorze Z;

(b) $|Z^{(a)}| = O(c|D|\log|G|/2^{a-1})$;

(c) zbiór $D^{(a)}$ jest $O(c\log|G|)$-dominującym zbiorem w grafie $G^{(a)}$.

Dowód:

Podpunkt (a) wynika natychmiast z zastosowania Lematu 2.14. W podpunkcie (b) należy natomiast zauważyć, że $|D^{(a+1)}| = |\partial(\bar{\mathcal{P}})| \leqslant |D^{(a)}|/2$, a stąd $|D^{(a+1)}| \leqslant |D|/2^{a+1}$. Podstawiając za l wartość $O(c\log|G|)$ w Lemacie 2.13 otrzymujemy:

$$|Z^{(a)}| = O(c\log|G||D^{(a-1)}|) = O(c|D|\log|G|/2^{a-1}).$$

W celu udowodnienia podpunktu (c) pokażę, że $D^{(a)}$ jest $O(c\log|G|)$-dominującym zbiorem w grafie indukowanym przez zielone i czerwone wierzchołki. Załóżmy więc nie wprost, że wierzchołek $v \in G^{(a)}$ nie jest $O(c\log|G|)$-zdominowany przez wierzchołek d_i. Wiadomo, że wierzchołek $kolor[v] \in \{\text{zielony, czerwony}\}$, w przeciwnym przypadku byłby koloru czarnego i dominowałby sam siebie.

Z faktu, iż wierzchołek v nie jest zdominowany wiemy, że istnieje wierzchołek $w \in vT_id_i$, którego $kolor[w] \in \{\text{biały, niebieski}\}$. Jako w oznaczmy najbliżej leżący wierzchołek o kolorze białym lub niebieskim na ścieżce vT_id_i, a wierzchołek go poprzedzający oznaczamy jako u, zatem $kolor[u] \in \{\text{zielony, czerwony}\}$. Zauważyć można, że kolor wierzchołka w nie może być ani biały, ani niebieski, ponieważ zostałby w takim przypadku pokolorowany podczas wykonywania PROCEDURY SEPARACJI na kolor zielony lub czerwony. Otrzymujemy więc sprzeczność z założeniem, że wierzchołek $w \notin G^{(a)}$.

□

Lemat 2.19. *Niech $G = (V, E)$ będzie wejściowym grafem w* PROCEDURZE KLASTROWANIA *oraz niech $(B_1, \ldots, B_q)$ będzie rodziną zbiorów zwróconą w kroku trzecim tej procedury, wówczas spełnione są następujące warunki:*

(a) $(B_1, \ldots, B_q)$ jest podziałem wierzchołków V;

(b) zbiór Z jest pokryciem wierzchołkowym wszystkich krawędzi pomiędzy zbiorami $(B_1, \ldots, B_q)$;

(c) $|Z| = O(c|D| \log |G|)$, gdzie D jest zbiorem z pierwszego kroku PROCEDURY KLASTROWANIA.

Dowód:

W celu udowodnienia podpunktu (a) oszacujmy liczbę małych klastrów $C_i^{(a)}$ zawartych w podziale skonstruowanym w korku 2(b) PROCEDURY KLASTROWANIA. Ponieważ

$$|\{\mathcal{C} \in \bigcup_j \partial^*(P_j^{(a)})\}| \leqslant l_a/2,$$

to w a-tej iteracji procedury moc pozostałego zbioru l-dominującego wynosi $|D^{(a)}| \leqslant |D|/2^a$. Po wykonaniu $O(\log |G|)$ iteracji zbiór V jest pusty. Ponadto każdy wierzchołek z grafu G należy do jednego ze zbiorów $B_i^{(a)}$ co oznacza, że jest to poprawny podział wierzchołków.

Podpunkt (b) wymaga jedynie zauważenia, iż każda krawędź pomiędzy zbiorami $(B_1, \ldots, B_q)$ jest krawędzią posiadającą dokładnie jeden koniec w zbiorze $B_i^{(a)}$, natomiast drugi jej koniec z Lematu 2.18 (a) leży w zbiorze Z. Tak więc zbiór Z jest pokryciem wierzchołkowym wszystkich takich krawędzi.

Dowodząc ostatni podpunkt (c) lematu zauważmy, że skoro wykonywanych jest $O(\log |G|)$ iteracji, to z Lematu 2.18,

$$|Z| = O(c|D| \log |G| \sum_{a \geqslant 0} 2^{-a}).$$

Sumę szeregu występującego w równaniu możemy w prosty sposób oszacować z góry przez wartość 1, a stąd otrzymujemy

$$|Z| = O(c|D| \log |G|).$$

□

Algorytm pakowania grafów.

W tym miejscu opisany zostanie algorytm pakowania grafu H w grafie G. Korzystać on będzie z opisywanych we wcześniejszych podrozdziałach procedur.

ALGORYTM PAKOWANIA

Wejście: Spójny graf H, planarny graf G, dodatnia liczba całkowita $k \in \mathbb{Z}$.

Wyjście: Aproksymacja pakowania grafu H w grafie G.

1. Każdy wierzchołek v z grafu G, dla którego nie istnieje podgraf grafu G izomorficzny z grafem H i zawierający niego samego usuń z grafu G.

2. Wywołaj PROCEDURĘ KLASTROWANIA z parametrem $c = |H|$ i niech $B = (B_1, \ldots, B_q)$ będzie podziałem, a zbiór Z zbiorem zwróconym przez tę procedurę.

3. Niech G' będzie ważonym minorem grafu G otrzymanym przez ściśnięcie każdego podzbioru B_i do pojedynczego wierzchołka b_i i ustawieniem funkcji wagi $\omega(b_i) = |B_i \cap Z|$.

4. Znajdź $O(polilog(|G|), 1/\log^{k+1}|G|)$-podział A' grafu G' używając algorytmu z pracy [7] i niech $A = (A_1, \ldots, A_t)$ będzie podziałem grafu G otrzymanym z A' przez operację odwrotną do ściskania wierzchołków b_i.

5. Znajdź optymalne pakowanie grafu H w każdym z podgrafów indukowanych wierzchołkowo $G[A_i]$ i zwróć sumę zbiorów. Zwracany zbiór oznacz jako $\mathcal{H}_{ALG}$.

Twierdzenie 2.20. *Niech $\nu_H(G)$ będzie wartością optymalnego pakowania grafu H w planarnym grafie G i niech k będzie dowolną ustaloną stałą dodatnią.* PROCEDURA PAKOWANIA *mając dane grafy H, G oraz wartość k zwraca właściwe pakowanie grafu H w grafie G, którego moc wynosi co najmniej*

$$(1 - O(|H|/\log^k |G|))\nu_H(G),$$

gdzie $\nu_H(G)$ oznacza moc optymalnego pakowania grafu H w grafie G. Czas wykonywania algorytmu jest polilogarytmiczny $polilog(|V(G)|)$ dla grafu H o mocy $|V(H)| = O(polilog(|G|))$ i wartości $k = O(1)$.

Dowód:

Oznaczmy jako $\mathcal{H}$ optymalne pakowanie grafu H, czyli takie, którego moc wynosi $\nu(H)$. Niech $\mathcal{H}_{IN}$ będzie podzbiorem pakowania $\mathcal{H}$ takim, że:

$$\underset{H \in \mathcal{H}_{IN}}{\forall} \underset{i}{\exists} H \subset G[A_i].$$

Skoro w piątym kroku algorytmu znajdujemy optymalne pakownie każdej ze składowych $G[A_i]$, to moc rozwiązania zwróconego przez ALGORYTM PAKOWANIA wynosi co najmniej $|\mathcal{H}_{IN}|$, zatem $|\mathcal{H}_{ALG}| \geqslant |\mathcal{H}_{IN}|$.

Jako $\mathcal{H}_{OUT}$ oznaczmy upakowane grafy ze zbioru $\mathcal{H}$ nie należące do $\mathcal{H}_{IN}$, a więc $\mathcal{H}_{OUT} = \mathcal{H} \setminus \mathcal{H}_{IN}$. Oczywiste jest, iż grafy ze zbioru $\mathcal{H}$ są wierzchołkowo rozłączne. Oznacza to, iż moc zbioru $|\mathcal{H}_{OUT}|$ wynosi co najwyżej mocy największego skojarzenia w grafie

$G[\partial(A)]$ (o krawędziach pomiędzy zbiorami A_i). Wynika to z faktu, że każdy upakowany graf ze zbioru $\mathcal{H}_{OUT}$ musi zawierać co najmniej jedną taką krawędź, a upakowania muszą być z założenia problemu rozłączne wierzchołkowo.

Zauważyć również można, że każda krawędź ze zbioru $G[\partial(A)]$ posiada jeden koniec w zbiorze Z, a drugi w B. Na mocy Lematu 2.19 (b) zbiór Z jest wierzchołkowym pokryciem $G[\partial(B)]$. Oznaczmy jako Z' podzbiór zbioru Z ($Z' \subset Z$) będący pokryciem wierzchołkowym zbioru $G[\partial(A)]$.

Stosując Lemat 2.19(c) otrzymujemy, że:

$$|Z'| \leqslant \omega(G')/\log^{k+1}|G| = |Z|/\log^{k+1}|G| \leqslant c|D|/\log^{k}|G|,$$

gdzie $c = |H|$. Skoro wielkość maksymalnego skojarzenia wynosi co najwyżej moc minimalnego pokrycia otrzymujemy:

$$|\mathcal{H}_{OUT}| = O(c|D|/\log^{k}|G|).$$

Dodatkowo zachodzi $|D| \leqslant \nu_H(G)$ gdyż dowolne dwa ustalone wierzchołki w D są w odległości co najwyżej $2c+1$ i każdy wierzchołek w grafie zawiera kopię grafu H. Na tej podstawie moc rozwiązania zwróconego przez algorytm wynosi co najmniej

$$\nu_H(G) - O(\nu_H(G)c/\log^{k}|G|) = \left(1 - \frac{|H|}{\log^{k}|G|}\right)\nu_H(G).$$

□

3 Rozproszone algorytmy lokalne.

Jednym z głównych nurtów rozwoju algorytmów rozproszonych są tzw. algorytmy lokalne, czyli o stałej złożoności czasowej. Wierzchołki występujące w grafie podejmują w nich działanie na podstawie pewnych wejściowych parametrów oraz informacji zebranych od ograniczonego sąsiedztwa, stąd też pochodzi ich nazwa. Wszelkie obliczenia prowadzone są praktycznie na danych "prawie" lokalnych.

3.1 Minimalne pokrycie wierzchołkowe.

Z wcześniejszych rozdziałów rozprawy znany jest algorytm aproksymacyjny rozwiązujący problem minimalnego pokrycia wierzchołkowego za pomocą maksymalnego skojarzenia. Podejście to, pozwala na otrzymanie 2-aproksymującego algorytmu w różnych podklasach grafów. W pracy [23] pokazano, że istnieje lokalny algorytm znajdujący 3-aproksymację problemu MVC w grafach o ograniczonym stopniu.

W rozprawie przedstawiony zostanie algorytm znajdujący $(2+\epsilon)$-aproksymację tego problemu w tej samej klasie grafów, rezultat ten otrzymany został równolegle z pracą [24]. Pomimo minimalnie gorszego współczynnika aproksymacji (dowolnie mała stała $\epsilon > 0$) wart jest uwagi ze względu na jego prostą budowę. W rozdziale tym zaprezentowane zostanie również podejście umożliwiające rozszerzenie algorytmu na klasę grafów o stałej lesistości.

Zastosowane techniki

Pierwsza z procedur wykorzystana w rozproszonym algorytmie znajdującym $(2+\epsilon)$- aproksymacje MVC została zaprezentowana w pracy [13]. Pozwala na znalezienie w czasie stałym $O(K)$ w grafie o ograniczonym stopniu $(d(v) \leqslant K)$ maksymalnego skojarzenia. Z powodu jej wielokrotnego wykonywania w naszym algorytmie oraz jej prostoty zostanie poniżej przedstawiona w celu pełniejszego opisu działania naszego algorytmu.

Procedura Maksymalnego-skojarzenia-dla-grafów-dwudzielnych

Wejście: Dwudzielny graf $G = (V, E)$, gdzie $V_1 \cup V_2 = V, V_1 \cap V_2 = \emptyset$ są rozłącznymi podzbiorami wierzchołków w grafie dwudzielnym.

Wyjście: Maksymalne skojarzenie M w grafie G.

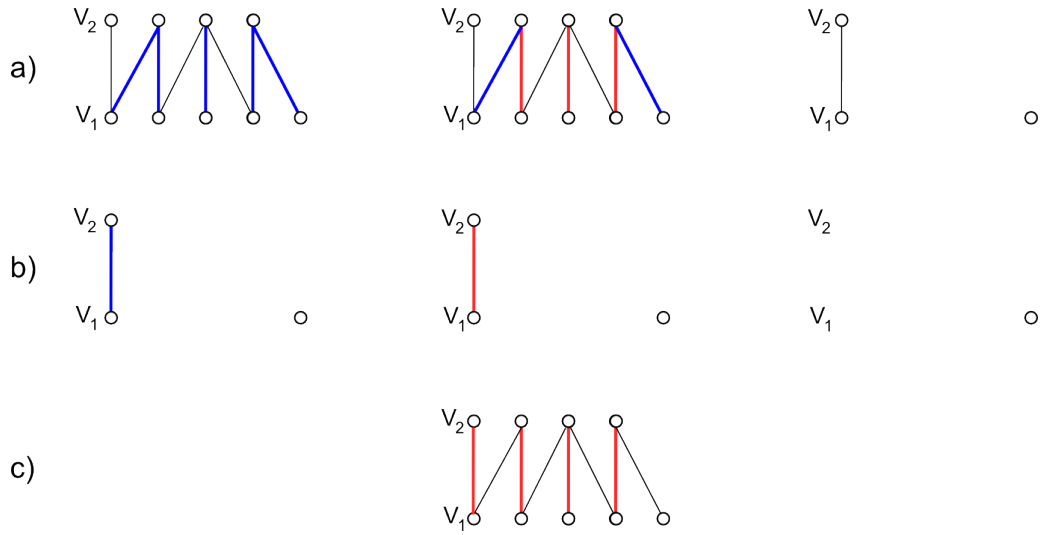
1. Niech $M = \emptyset$.
2. Iteruj K razy:
   a) dla każdego wierzchołka v ze zbioru V_1 wybierz(zaznacz) jedną krawędź incydentną do niego;
   b) dla każdego wierzchołka u ze zbioru V_2 wybierz jedną (o ile istnieje) zaznaczoną już krawędź incydentną do niego;
   c) dodaj wszystkie krawędzie wybrane przez wierzchołki ze zbioru V_2 do M. Wszystkie krawędzie ze zbioru M oraz do nich przyległe wierzchołki usuń z grafu G. Z grafu G usuń wierzchołki izolowane;
3. Zwróć zbiór M.

Procedura ta, jak można łatwo zauważyć zwraca szukane maksymalne skojarzenie, a czas działania z uwagi na redukcje frakcji krawędzi ($\geqslant \frac{1}{K}$) w pojedynczej rundzie wynosi $O(K)$. Przykład działania procedury przedstawiony został na Rysunku 22.

Drugą procedurą (MVC-3-aproksymacja), z której skorzystaliśmy, pochodzi ze wspomnianej już pracy [23]. Polega ona na podziale grafu G o ograniczonym stopniu na ścieżki i cykle w ten sposób, by każda krawędź z grafu G posiadała przynajmniej jeden koniec w jakimś cyklu lub ścieżce.

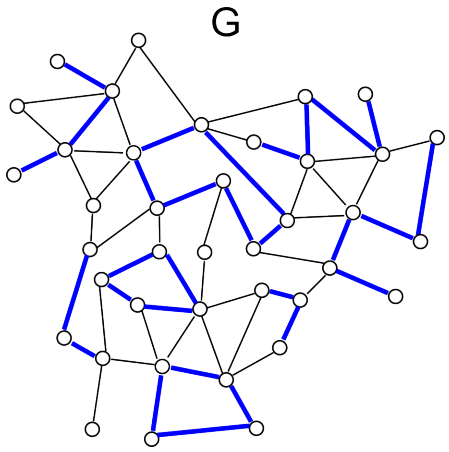
Fakt 3.1. *Niech G będzie grafem takim, że $\Delta(G) \leqslant K$, wtedy algorytm* MVC-3-aproksymacja *w $O(K)$ rundach znajduje podgraf H grafu G ($H \subseteq G$) o następujących własnościach:*

- *każda składowa grafu H jest nietrywialną ścieżką, cyklem oraz*
- *każda krawędź grafu G posiada przynajmniej jeden koniec w zbiorze wierzchołków $V(H)$.*



Rysunek 22: Przykład działania PROCEDURY MAKSYMALNEGO-SKOJARZENIA-DLA-GRAFÓW-DWUDZIELNYCH.

W wyniku wybrania wszystkich wierzchołków $V(H)$ z grafu H, jako pokrycia wierzchołkowego grafu G, otrzymujemy 3-aproksymację optymalnego rozwiązania problemu. Najgorszy współczynnik aproksymacji uzyskuje się podczas "pokrywania" krawędzi ścieżki długości 2 (o trzech wierzchołkach).



Rysunek 23: Przykład działania PROCEDURY MVC-3-APROKSYMACJI. Krawędzie w grafie H zaznaczone zostały kolorem niebieskim.

Algorytm $(2+\epsilon)$-MVC w grafie o ograniczonym stopniu

Możliwe jest uzyskanie algorytmu aproksymującego minimalne pokrycie wierzchołkowe o lepszym współczynniku aproksymacji poprzez ograniczenie ilości krótkich ścieżek parzystych

w grafie H. W podrozdziale tym przedstawiony zostanie zaproponowany przez nas algorytm.

Na wstępie wprowadźmy wymagane definicje i terminy.

Definicja 3.2. *Mówimy, że długość ścieżki P o k krawędziach wynosi k i oznaczamy jako $length(P) = k$.*

Dla rodziny zbioru wszystkich ścieżek $\mathcal{P}$ wierzchołki $v \in \mathcal{P}$ o stopniu równym 1 ($d(v) = 1$, czyli końce ścieżek) oznaczamy jako $ends(\mathcal{P})$. Dodatkowo jako $\mathcal{P}^{(k)}$ oznaczamy zbiór ścieżek $P \in \mathcal{P}$ o długości co najmniej k. Algorytm MVC-$(2+\epsilon)$-APROKSYMACJI podzielony został na dwie części. Pierwsza z nich odpowiedzialna jest za utworzeniu grafu H o pewnych dodatkowych własnościach, natomiast faza druga wykonywana jest w celu łączenia krótkich parzystych ścieżek ze sobą.

FAZA 1: MVC-$(2+\epsilon)$-APROKSYMACJI

Wejście: Graf G o ograniczonym stopniu ($\Delta(G) \leqslant K$), dodatnie liczby całkowite K oraz L, gdzie $L \geqslant 2$.

Wyjście: Zbiór ścieżek $\mathcal{P}$ i wierzchołków należących do składowych utworzonego grafu H.

1. Wywołaj PROCEDURĘ MVC-3-APROKSYMACJA w celu znalezienia podgrafu H w grafie G. Jako $\mathcal{P}$ oznaczmy zbiór wszystkich parzystych ścieżek zawartych w grafie H o długości co najwyżej L, a przez $\mathcal{Q}$ oznaczmy pozostałe składowe grafu H.

2. Niech $U = ends(\mathcal{P})$ oraz niech W będzie zbiorem wierzchołków w grafie $V(G) \setminus V(H)$, które posiadają przynajmniej jednego sąsiada w zbiorze U.

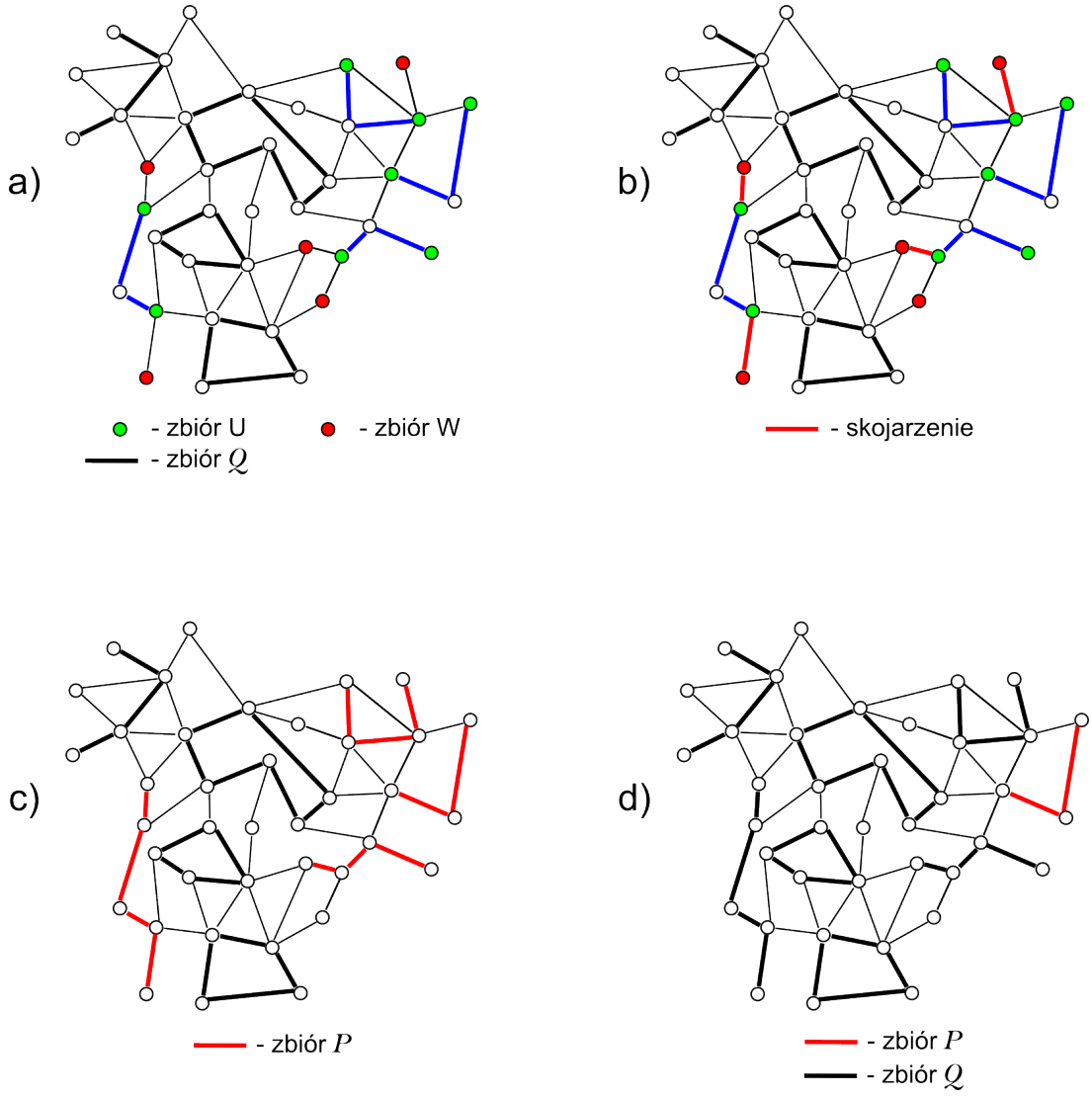
3. Wywołaj PROCEDURĘ MAKSYMALNEGO-SKOJARZENIA-DLA-GRAFÓW-DWUDZIELNYCH, aby znaleźć maksymalne skojarzenie M w dwudzielnym grafie $G[U, W]$.

4. Przedłuż ścieżki ze zbioru $\mathcal{P}$ używając krawędzie ze zbioru M. Nowy otrzymany podzbiór ścieżek oznaczmy jako $\mathcal{P}$.

5. Dla każdej ścieżki $P \in \mathcal{P}$, jeżeli $length(P)$ jest nieparzysta przenieś P z $\mathcal{P}$ do $\mathcal{Q}$.

6. Zwróć $\mathcal{P}$ oraz $\mathcal{Q}$.

Lemat 3.3. *Niech G będzie grafem o ograniczonym stopniu $\Delta(G) \leqslant K$. Niech $\mathcal{P}$ i $\mathcal{Q}$ będą zbiorami zwróconymi przez* FAZĘ 1 MVC-$(2+\epsilon)$-APROKSYMACJI*, a V_P będzie zbiorem*



Rysunek 24: Przykład działania FAZY 1: MVC-$(2+\epsilon)$-APROKSYMACJI a) po kroku 2, b) po kroku 3, c) po kroku 4, d)po kroku 6.

wierzchołków leżących na ścieżce $\mathcal{P}$, a V_Q zbiorem wierzchołków należących do zbioru $\mathcal{Q}$. Wtedy:

a) długość każdej ścieżki P ze zbioru ścieżek $\mathcal{P}$ wynosi $2 \leqslant l \leqslant L+2$. Natomiast zbiór składowych $\mathcal{Q}$ zawiera rozłączne wierzchołkowo grafy, z których każdy jest cyklem, nietrywialną ścieżka o nieparzystej długości lub ścieżką o długości większej niż L;

b) każda krawędź $e = \{u, v\} \in E(G)$ o jednym końcu $v \notin V_P \cup V_Q$ posiada drugi koniec u w zbiorze V_Q lub u jest wewnętrznym wierzchołkiem pewnej ścieżki ze zbioru $\mathcal{P}$.

Czas działania fazy 1 wynosi $O(K+L)$ rund.

Dowód:

Każda ścieżka P należąca do zbioru $\mathcal{P}$ utworzonego w pierwszym kroku procedury jest nietrywialna oraz posiada parzystą długość. Ponieważ wydłużona, w kroku 4 tej procedury, może zostać jedynie o co najwyżej dwie krawędzie, to długość jej jest ograniczona przez wartość $length(P) \leqslant L + 2$. Natomiast w przypadku, gdy dołączona zostanie do niej tylko jedna krawędź, to w kroku 5 procedury zostanie ścieżka ta przeniesiona do zbioru składowych Q ze zbioru $\mathcal{P}$. W obu przypadkach otrzymujemy prawdziwość podpunktu a) lematu.

W celu udowodnienia podpunktu b) wybierzmy dowolną krawędź $e = \{u, v\} \in E(G)$ taką, że jeden z jej końców nie należy do wierzchołków ze składowej grafu H (spełniającą zależność $v \notin V(H) = V_P \cup V_Q$). Z faktu 3.1 wynika, że drugi koniec wybranej krawędzi znajduje się w zbiorze wierzchołków grafu H, czyli $u \in V(H)$.

Załóżmy, że $u \notin V_Q$, wtedy wierzchołek u musi należeć do zbioru V_P. Może on być końcem krawędzi P lub wierzchołkiem wewnętrznym. W drugim przypadku otrzymujemy prawdziwość podpunktu b) lematu. Załóżmy, że $u \in ends(\mathcal{P})$. W rezultacie v musiał znaleźć się w zbiorze W w kroku 2 algorytmu, ponieważ sąsiadem jego jest wierzchołek ze zbioru U. Z maksymalności skojarzenia M w grafie $G[U, W]$ wiemy, że istnieje krawędź $f \in M$ taka, że $f \cap e = \{u\}$, w przeciwnym przypadku krawędź e należałaby do skojarzenia M. Zauważmy, że wszystkie początkowe ścieżki w kroku 1 algorytmu są nietrywialne, a więc u jest wierzchołkiem wewnętrznym pewnej ścieżki $P' \in \mathcal{P}$ zwróconym w korku 6.

Z faktu 3.1 algorytm wymaga $O(K)$ rund, aby skonstruować graf H. Ponadto konstrukcje zbiorów V_Q oraz $\mathcal{P}$ pochłaniają $O(L)$ rund (czas wymagany do sprawdzenia czy $length(P) \leqslant L$), a znalezienie skojarzenia M wymaga $O(K)$ rund. Sumując podane złożoności otrzymujemy prawdziwość ostatniego założenia lematu.

□

W drugiej fazie algorytmu zamierzamy usunąć wszystkie możliwe krótkie ścieżki o parzystej długości z istniejącego zbioru $\mathcal{P}$, tak by zbiór $V_P \cup V_Q \setminus ends(P)$ był aproksymacją rozwiązania problemu minimalnego pokrycia wierzchołkowego o żądanym współczynniku aproksymacji.

FAZA 2: MVC-$(2+\epsilon)$-APROKSYMACJI

Wejście: $\mathcal{P}$, $\mathcal{Q}$ - zbiory zwrócone w FAZIE 1 PROCEDURY MVC-$(2+\epsilon)$-APROKSYMACJI.

Wyjście: Minimalne pokrycie wierzchołkowe D.

1. Iteruj od $i = 1$ do $(L + 1)/2$:

(A) Dla każdej krawędzi $e \in \mathcal{P}$ ustaw $kolor[e] =$ czarny, a pozostałe krawędzie ustaw jako niepokolorowane. Dla każdej ścieżki $P \in \mathcal{P}^{(2i)}$ dodaj jeden z jej końców do zbioru U_{Left}, a drugi do zbioru U_{Right};

(B) Użyj algorytmu MVC 3-APROKSYMACJI w grafie $G[U_{Left}]$. Niech H' będzie grafem zwróconym przez tę procedurę. Dla każdej krawędzi $e \in E(H')$ ustaw $kolor[e] =$ niebieski;

  (a) Każdy wierzchołek $u \in U_{Left}$ incydentny z dwoma niebieskimi krawędziami usuń ze ścieżki zawierającej $P \in \mathcal{P}$ i następnie przenieś ścieżkę P do zbioru $\mathcal{Q}$;

  (b) Połącz ścieżki $P', P'' \in \mathcal{P}$ za pomocą niebieskich krawędzi (o ile to możliwe). Jeżeli długość l w ten sposób powstałej ścieżki wynosi $l \geqslant L + 1$ to przenieś ją do zbioru $\mathcal{Q}$, w przeciwnym przypadku krawędzie pokoloruj na czarno. Dodatkowo do tego zbioru dodaj wszystkie niebieskie cykle, a ich krawędzie ustaw jako niepokolorowane;

(C) Wszystkie wierzchołki $u \in U_{Right} \cup U_{Left}$ leżące na ścieżce, która została zmodyfikowana w kroku (B) usuń ze zbioru $U_{Right} \cup U_{Left}$. Wykonaj krok (B) w grafie $G[U_{Right}]$;

(D) Wszystkie wierzchołki $u \in U_{Right} \cup U_{Left}$ leżące na ścieżce, która została zmodyfikowana w kroku (C) usuń ze zbioru $U_{Right} \cup U_{Left}$;

(E) Wywołaj PROCEDURĘ MAKSYMALNEGO-SKOJARZENIA-DLA-GRAFÓW -DWUDZIELNYCH w grafie dwudzielnym $G[U_{Right}, U_{Left}]$. Niech M będzie znalezionym skojarzeniem, a każdą krawędź $e \in M$ pokoloruj na czarno. Następnie rozważmy graf indukowany przez wszystkie czarne krawędzie: cykle, ścieżki o długości $l \geqslant L + 1$ oraz ścieżki nieparzyste przenieś do $\mathcal{Q}$(usuwając automatycznie ich kolorowanie);

2. Zwróć $D = V_Q \cup \mathcal{P} \setminus ends(\mathcal{P})$.

Lemat 3.4. *Niech $\mathcal{P}$, $\mathcal{Q}$ będą zbiorami otrzymanymi po zakończeniu i-tej iteracji* FAZY 2 MVC-(2+ϵ)-APROKSYMACJI, *natomiast $\mathcal{P}_*, \mathcal{Q}_*$ będą zbiorami przed rozpoczęciem i-tej iteracji. Wtedy*

(a) każdy wierzchołek incydentny do niebieskiej krawędzi (w i-tej iteracji) nie należy do zbioru $ends(\mathcal{P})$ *oraz spełniona jest zależność* $ends(\mathcal{P}) \subseteq ends(\mathcal{P}_*)$.

(b) rodzina $\mathcal{Q}$ *zawiera wierzchołkowo rozłączne grafy, z których każdy jest cyklem, nietrywialną ścieżką o nieparzystej długości lub ścieżką o długości większej niż* L.

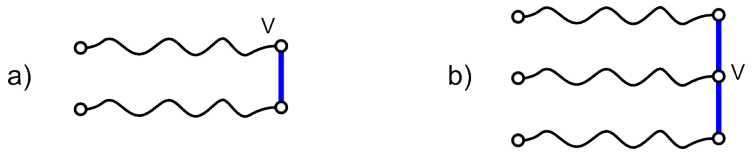
(c) zbiór $ends(\mathcal{P}^{(2i)})$ *jest niezależny w grafie* G.

Dodatkowo zbiór D *zwrócony w kroku 3 jest wierzchołkowym pokryciem, a czas działania procedury wynosi* $O(L(K+L))$ *rund.*

Dowód:

Zauważmy, że dowolna ścieżka $P \in \mathcal{P}_*$ w i-tej iteracji posiada co najwyżej jeden koniec incydentny do niebieskich krawędzi. Wystąpienie takiej sytuacji prowadzi do zmodyfikowania ścieżki.

W celu udowodnienia podpunktu (a) wykażemy, że każdy końcowy wierzchołek v staje się wierzchołkiem wewnętrznym pewnej ścieżki lub cyklu, a tym samym nie należy do $ends(\mathcal{P})$. Wierzchołek v może być incydentny do jednej lub dwóch krawędzi niebieskich, co zostało to przedstawione na Rysunku 25.



Rysunek 25: Przykład wierzchołka incydentego do niebieskich krawędzi podczas działania FAZY 2 MVC-$(2+\epsilon)$-APROKSYMACJI.

W pierwszym przypadku, w kroku 1(B)(b) ścieżki zostaną połączone, a wierzchołek znajdzie się w środku połączonych ścieżek. Natomiast w drugim przypadku w kroku 1(B)(a) zostanie on odłączony od "środkowej" ścieżki, a w kroku 1(B)(b) połączy on inne dwie ścieżki. W obu przypadkach wierzchołek ten po zakończeniu iteracji nie będzie więc należał do zbioru $ends(\mathcal{P})$. Nowe końce ścieżek mogą powstać jedynie podczas operacji usuwania wierzchołka v ze ścieżki (w kroku 1(B)(a)). Jednak ścieżka ta jest natychmiast usuwana ze zbioru $\mathcal{P}$, co kończy dowód podpunktu (a) lematu.

Podpunkt (b) z uwagi, iż do zbioru $\mathcal{Q}$ dodawane są ścieżki jedynie o odpowiedniej długości, o odpowiedniej parzystości, a także cykle spełnia założenia lematu.

W celu udowodnienia podpunktu (c) załóżmy nie wprost, że istnieją dwa wierzchołki $u, v \in ends(\mathcal{P}^{(2i)})$ oraz są one przyległe. Ponieważ podczas działania procedury nie są tworzone żadne nowe wierzchołki w zbiorze $ends(\mathcal{P})$ to $u, v \in ends(\mathcal{P}_*^{(2i)})$. Podczas iteracji nie

zostały pokolorowane na niebiesko żadne do nich incydentne krawędzie, co wynika z podpunktu (a) lematu. Każdy z wierzchołków u, v w kroku 1(A) przypisany został do zbioru U_{Right} lub zbioru U_{Left}. Rozpatrzymy dwa przypadki, odpowiednio, gdy wierzchołki należą i nie należą do tego samego zbioru. W przypadku pierwszym wywołanie algorytmu MVC 3-APROKSYMACJI musi zwrócić krawędź przyległą do wierzchołka u lub v ponieważ w przeciwnym razie nie byłoby to poprawne pokrycie wierzchołkowe. Krawędź ta jest pokolorowana na niebiesko, co prowadzi do sprzeczności. W przypadku drugim zostanie zwrócona krawędź f przyległa do wierzchołka u, v lub sama krawędź e, ze względu na maksymalność skojarzenia. Otrzymaliśmy sprzeczność, ponieważ wierzchołek u lub v po wykonaniu operacji nie będzie już końcem ścieżki.

W celu udowodnienia ostatniej części lematu załóżmy nie wprost, że zbiór D nie jest poprawnym pokryciem wierzchołkowym. Istnieje więc krawędź $e \in G$ taka, że $e \cap D = \emptyset$. Wiemy również, że zbiory $\mathcal{P}$, $\mathcal{Q}$ konstruowane były na bazie minimalnego pokrycia wierzchołkowego, zatem krawędź e ma oba końce w zbiorze $ends(\mathcal{P})$. Otrzymujemy jednak sprzeczność ze względu na niezależność zbioru $ends(\mathcal{P})$, co kończy dowód.

□

Lemat 3.5. *Niech $K \in \mathbb{N}$, natomiast G będzie grafem o maksymalnym stopniu $\Delta(G) \leqslant K$. Istnieje wtedy rozproszony aproksymacyjny algorytm znajdujący w $O(K/\epsilon)$ rundach minimalne pokrycie wierzchołkowe D o współczynniku aproksymacji $2+\epsilon$ w grafie G, czyli*

$$|D| \leqslant (2+\epsilon)|D^*|.$$

Dowód:

Niech $L = 2\lceil 1/\epsilon \rceil$, a zbiór D będzie pokryciem zwróconym w wyniku wywołania FAZY 1 I 2 PROCEDURY MVC-$(2+\epsilon)$-APROKSYMACJI z parametrem L. Z Lematu 3.4 zbiór D jest pokryciem wierzchołkowym w grafie G, a czas działania jest ograniczony przez $O(K/\epsilon)$ rund.

Niech D^* będzie optymalnym wierzchołkowym pokryciem o mocy $|D^*| = \beta(G)$, natomiast $\mathcal{P}, \mathcal{Q}$ będą zbiorami z Lematu 3.4, gdzie $i = L$. D^* zawiera przynajmniej połowę wierzchołków każdego cyklu, każdej nieparzystej ścieżki oraz wewnętrznych wierzchołków każdej ścieżki ze zbioru $\mathcal{P}$.

Jeżeli $P \in \mathcal{Q}$ jest ścieżką o parzystej długości, wtedy $length(P) \geqslant L+2$. Zbiór D^* musi zawierać przynajmniej $(|V(P)|-1)/2$ wierzchołków ze ścieżki P. Stąd

$$|D \cap V(P)| \leqslant 2\,|D^* \cap V(P)| + 1 \leqslant (2 + 2/(L+3))|D^* \cap V(P)| \leqslant (2+\epsilon)|D^* \cap V(P)|.$$

Ponieważ grafy $\mathcal{P}$ i $\mathcal{Q}$ są rozłączne wierzchołkowo otrzymujemy

$$|D| \leqslant (2+\epsilon)|D^*|.$$

□

Własność 3.6. *Nie istnieje algorytm rozproszony, który w czasie stałym ($O(1)$-rund) rozwiązuje problem minimalnego pokrycia wierzchołkowego w grafach o ograniczonym stopniu, ze współczynnikiem aproksymacji $(2-\epsilon)\beta(G)$, gdzie ϵ jest dowolną stałą większą od 0.*

Dowód:

Mamy dany cykl C składający się z n wierzchołków, gdzie n jest parzyste. Wielkość minimalnego pokrycia wierzchołkowego w tym grafie wynosi $n/2$. Załóżmy nie wprost, że D jest znalezionym pokryciem wierzchołkowym o wielkości $|D| \leqslant (2-\epsilon)|D^*|$. Wtedy spełniona jest następująca zależność:

$$|D| \leqslant (2-\epsilon)|D^*| = (2-\epsilon)\frac{n}{2} = (1-\frac{\epsilon}{2})n.$$

Stąd moc zbioru niewybranych wierzchołków do minimalnego pokrycia wierzchołkowego wynosi $|V(C) \setminus D| \geqslant n\epsilon/2$, a wierzchołki te tworzą zbiór niezależny.

Otrzymanie ϵ-aproksymacji MIS jest niemożliwe ze względu na oszacowanie dolne opisane w Rozdziale 5, co prowadzi nas do sprzeczności.

Algorytm $(2+\epsilon)$-MVC dla grafów o ograniczonej lesistości

Podobny rezultat (o tym samym współczynniku aproksymacji) jak w poprzednio opisywanym algorytmie można otrzymać w grafach o ograniczonej lesistości. Jego konstrukcja opiera się na opisanej wcześniej procedurze. Wspomnianą klasę grafów o ograniczonej lesistości zdefiniowaliśmy w rozdziale 1 (definicja 1.7).

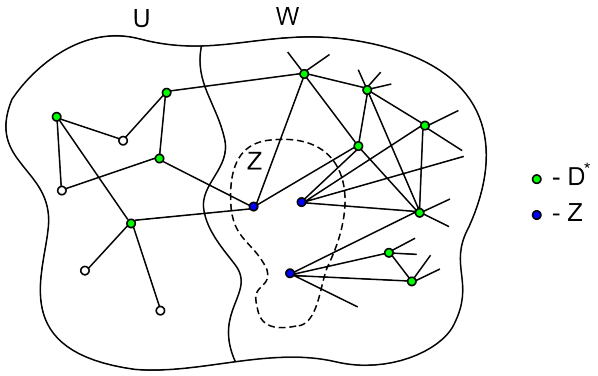
CONST-ARBORICITY-MVC-$(2+\epsilon)$-APROKSYMACJA

Wejście: Dodatnia liczba całkowitą $a \in \mathbb{N}$, $\delta \in (0,1)$, graf $G = (V,E)$ o ograniczonej lesistości $arb(G) \leqslant a$.

Wyjście: Minimalne pokrycie wierzchołkowe D.

1. Niech $C = 3a/\delta$. Utwórz rozłączne podzbiory wierzchołków $U, W \subseteq V$, gdzie $U = \{v \in V(G) | deg(v) \leqslant C\}$, a $W = V(G) \setminus U$.
2. Wywołaj procedurę MVC-$(2+\epsilon)$-APROKSYMACJI dla grafu U z parametrami $K = C$ oraz $\epsilon = \delta/2$. Znalezione pokrycie wierzchołkowe w indukowanym grafie $G[U]$ oznacz jako D.

3. Zwróć zbiór $D \cup W$.



Rysunek 26: Przykład podziału grafu G na zbiory U, W i Z.

Twierdzenie 3.7. *Niech Q będzie zbiorem zwróconym po zakończeniu działania powyższego algorytmu. Wtedy zbiór Q jest pokryciem wierzchołkowym w grafie $G = (V, E)$ o współczynniku aproksymacji $(2 + \delta)$, gdzie δ jest dowolną stałą większą od 0.*

Dowód:

Na wstępie udowodnimy, że zbiór Q jest pokryciem wierzchołkowym w grafie G. W tym celu podzielmy zbiór krawędzi na dwa rozłączne podzbiory E_W oraz E_U zdefiniowane następująco: $E_W = \{uv \in E | u \in W \ \vee v \in W\}, E_U = E \setminus E_W$. Zauważmy, że każda krawędź ze zbioru E_W posiada przynajmniej jeden koniec w zbiorze W, a tym samym w zbiorze Q. Ponadto wszystkie krawędzie ze zbioru E_U posiadają przynajmniej jeden koniec w zbiorze wierzchołków D, co wynika z faktu, iż D jest pokryciem wierzchołkowym w grafie $G[U]$.

W celu udowodnienia drugiej części twierdzenia załóżmy, że zbiór D^* jest optymalnym pokryciem wierzchołkowym w grafie G, a zbiór $Z = W \setminus D^*$ będzie zbiorem wszystkich wierzchołków ze zbioru W nie należących do optymalnego minimalnego pokrycia wierzchołkowego D^*.

Zwróćmy uwagę, iż zbiór wierzchołków Z nie zawiera elementów ze zbioru D^*, a tym samym wszystkie krawędzie o jednym końcu zawartym w zbiorze Z drugi posiadają w zbiorze D^*. Wiemy również, że każdy wierzchołek $z \in Z$ posiada stopień $d(z) > C$, a stąd

$$C|Z| \leqslant |E(Z, D^*)| \leqslant E(D^* \cup Z).$$

Korzystając z definicji lesistości otrzymujemy, że $C|Z| \leqslant a(|Z| + |D^*|)$ oraz

$$|Z| \leqslant \frac{a|D^*|}{C-a} \leqslant \delta|D^*|/2. \tag{2}$$

Oczywiste jest, iż zbiór $D^* \cap U$ jest pokryciem wierzchołkowym w indukowanym grafie $G[U]$, a z Lematu 3.5 uzyskujemy nierówność $|D| \leqslant (2+\delta/2)|D^* \cap U|$. Moc pokrycia wierzchołkowego Q zwróconego przez algorytm oszacować możemy więc w następujący sposób:

$$|Q| = |D \cup W| \overset{(1)}{\leqslant} (2+\delta/2)|D^* \cap U| + |D^* \cap W| + \delta|D^*|/2 \overset{(2)}{\leqslant} (2+\delta)|D^*|.$$

Nierówność (1) jest naturalną konsekwencją dwóch faktów poniżej opisanych. Zbiór D jest $(2+\epsilon)$-aproksymacją minimalnego pokrycia wierzchołkowego w grafie U oraz można go podzielić na dwa zbiory wierzchołków $W \cap D^*$ i Z. Wykorzystując oszacowanie z równania (2) dowodzimy tę nierówność.

W kolejnym kroku stosując proste szacowanie czynników w równaniu otrzymujemy nierówność (2), co kończy dowód.

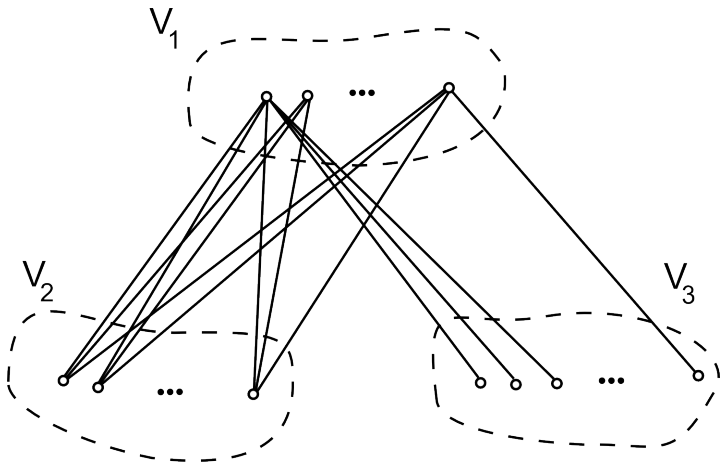
□

Rozszerzenie tego algorytmu na grafy "globalnie rzadkie" (spełniające warunek $|E(G)| \leqslant c|V(G)|$, gdzie $c > 0$) okazuje się niemożliwe. Dowodzi to poniższy przykład.

Przykład 3.8. *Niech $n \in N$ będzie dodatnią liczbą całkowitą. Rozważmy graf $G = (V, E)$ zbudowany z trzech rozłącznych zbiorów wierzchołków oznaczonych symbolami V_1, V_2, V_3 o mocach odpowiednio $n^{1/3}, n^{2/3}, n - n^{1/3} - n^{2/3}$. Ponadto niech $G[V_1, V_2] = K_{n^{1/3}, n^{2/3}}$, a zbiór wierzchołków V_1 przyległy będzie do $n^{2/3} - n^{n/3} - 1$ wierzchołków ze zbioru V_3. Oznacza to, iż $\underset{v \in V_3}{\forall} deg(v) = 1$.*

W przykładzie tym zauważmy, iż liczba wszystkich krawędzi w grafie wynosi $|E| = n - n^{1/3} - n^{2/3} + n < 2n$. Zbiór V_1 tworzy pokrycie wierzchołkowe grafu G, a jego moc wynosi $n^{1/3}$.

W grafie tym niemożliwe jest zastosowanie zaproponowanego algorytmu, ponieważ stopnie wierzchołków ze zbioru $v \in V_1 \cup V_2$ nie są ograniczone żadną stałą. W rezultacie wykorzystanie bezpośrednio procedury w grafach o ograniczonym stopniu jest niewykonalne.



Rysunek 27: Przykład grafu "rzadkiego globalnie" o nieograniczonej lesistości. Zbiory V_1, V_2, V_3 posiadają odpowiednio $n^{1/3}, n^{2/3}, n - n^{1/3} - n^{2/3}$ wierzchołków.

4 Rozproszone algorytmy sublogarytmiczne.

Szerokie zastosowanie praktyczne algorytmów lokalnych w informatyce wynika z ich szybkiego czasu działania. Akceptowalny okazuje się również kompromis pomiędzy złożonością czasową algorytmów, a ich współczynnikiem aproksymacji. W rozdziale tym przedstawione zostaną algorytmy, których złożoność czasowa jest bardzo bliska stałej i wynosi $O(\log^* n)$, gdzie wartość $log^* n$ oznacza liczbę logarytmowań wymaganą by wartość była mniejsza lub równa 1. Nieznaczne wydłużenie czasu działania algorytmu np. w przypadku minimalnego zbioru dominującego pozwala prawie aż 72 krotnie poprawić współczynnik aproksymacji.

4.1 Zastosowane techniki: klastrowanie i ciężkie gwiazdy.

Jedną z technik zastosowaną w algorytmach sublogarytmicznych jest utworzenie podziału ważonego grafu planarnego $G = (V, E, \omega))$ na zbiory wierzchołków $V_1, V_2, \ldots, V_k$, tak aby waga krawędzi pomiędzy dowolnymi różnymi zbiorami V_i, V_j ($i \neq j$ była znacząco mniejsza niż waga wierzchołkowa całego grafu $\omega(G)$.

Definicja 4.1. *Niech $G = (V, E)$ będzie grafem planarnym, a $\bar{\omega} : E \to R^+$ funkcją wagi na krawędziach. Ponadto niech $U_1, U_2, \ldots, U_l \subset V$ będą rozłącznymi podzbiorami wierzchołków V takimi, że $G[U_i]$ jest spójny. Oznaczmy jako $\tilde{G}$ ważony graf, w którym U_i jest ściśnięty do pojedynczego wierzchołka. Dodatkowo dla każdego $u, u' \in V(\tilde{G})$, gdzie $u \neq u'$ definiujemy następująco wagę na krawędziach w tym grafie:*

$$\bar{\omega}_{\tilde{G}}(u, u') = \sum_{x \in U, y \in U'} \bar{\omega}(x, y), \tag{3}$$

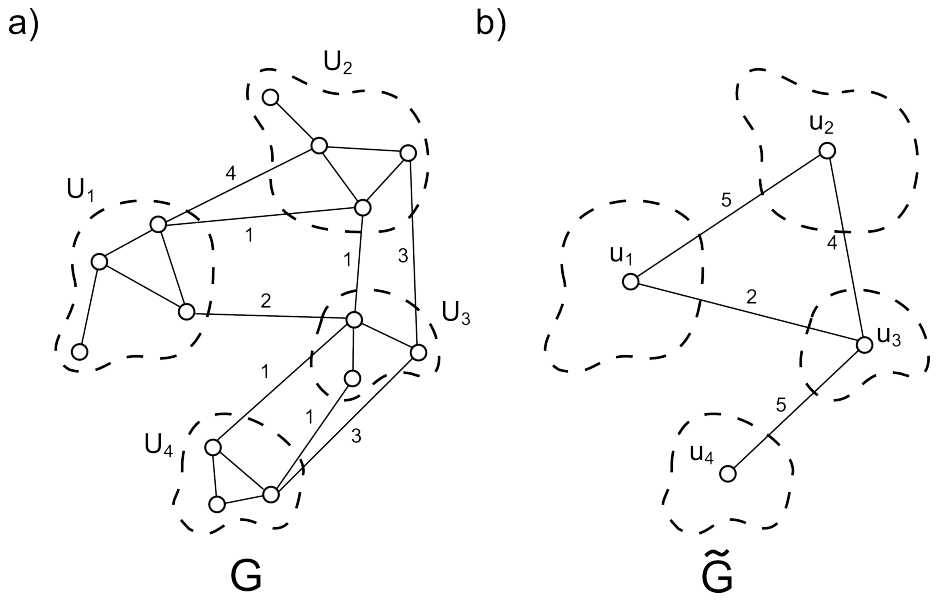
gdzie $U = U_i$, a u otrzymany został przez ściśnięcie zbioru wierzchołków U_i (U' definiujemy analogicznie).

Wprowadzimy teraz kilka pomocniczych definicji oraz faktów.

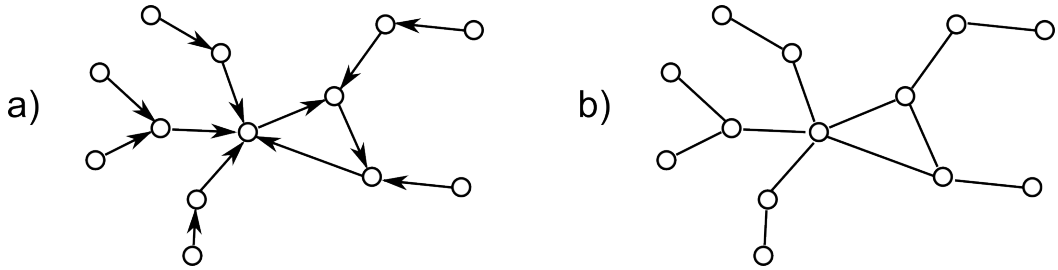
Definicja 4.2. *Skierowany graf $\overrightarrow{F}$ o maksymalnym wyjściowym stopieniu wierzchołków w zbiorze F wynoszącym jeden nazywamy* **pseudo-lasem**.

Jeżeli $\overrightarrow{F}$ jest skierowanym grafem, wtedy przez F oznaczamy graf otrzymany z $\overrightarrow{F}$ poprzez pominięcie orientacji krawędzi.

Fakt 4.3. *Niech $G = (V, E)$ będzie planarnym grafem z funkcją wagi na krawędziach $\bar{\omega} : E \to R^+$. Istnieje rozproszony algorytm znajdujący w dwóch rundach pseudo-las $\overrightarrow{F}$ taki, że F jest podgrafem G oraz $\bar{\omega}(F) \geqslant \bar{\omega}(G)/6$.*



Rysunek 28: Przykład tworzenia grafu $\tilde{G}$ z definicji 4.1 oraz nadawania wagi na krawędziach.



Rysunek 29: Przykład pseudo-lasu oraz utworzonego z niego grafu F.

Dowód:

Na początku zauważmy istnienie pseudo-lasu $\overrightarrow{F}$, gdzie F jest podgrafem grafu G oraz $\bar{\omega}(F) \geqslant \bar{\omega}(G)/3$. Wynika to z faktu, iż G jest grafem planarnym, a z twierdzenia Nasha-Williamsa ([19]) graf jest sumą co najwyżej 3 lasów. Ustalając korzeń drzewa otrzymujemy skierowany graf F o właściwości $\bar{\omega}(F) \geqslant \bar{\omega}(G)/3$.

W celu udowodnienia, że taki pseudo-las może zostać znaleziony rozważmy następującą rozproszoną procedurę: każdy wierzchołek v wybiera najcięższą krawędź $\{u, v\}$ incydentną do wierzchołka v i ustawia orientację z wierzchołka v do u. Jeżeli krawędź została wybrana przez dwa wierzchołki to wybierany jest tylko jeden dowolny kierunek krawędzi. Tak otrzymany graf jest pseudo-lasem. Skoro G jest sumą trzech lasów F_1, F_2, F_3 oraz $2\bar{\omega}(F) \geqslant \max\{\bar{\omega}(F_i)\}$, to otrzymujemy $\bar{\omega}(F) \geqslant \bar{\omega}(G)/6$.

□

Fakt 4.4. *Niech $\overrightarrow{F}$ będzie spójnym pseudo-lasem takim, że średnica F wynosi d wtedy $\overrightarrow{F}$ zawiera skierowaną ścieżkę o długości przynajmniej $d/2$.*

Niech $\overrightarrow{F}$ będzie pseudo-lasem wierzchołków właściwie pokolorowanych kolorami ze zbioru S. Dla wierzchołka v i zbioru kolorów $C \subset S$, niech $in(v, C)$ będzie zbiorem skierowanych krawędzi (u, v) (z u do v) takich, że kolor u zawarty jest w zbiorze C. Analogicznie definiujmy $out(v, C)$. Jeżeli C jest puste wtedy $in(v, C)$ oraz $out(v, C)$ są puste i ich wagi są równe zero.

Ciężkie gwiazdy

Wejście: Planarny graf $G = (V, E)$.

Wyjście: Ciężkie gwiazdy $S_1, S_2, \ldots, S_l$ ze zbioru wierzchołków w grafie G.

1. Znajdź pseudo-las $\overrightarrow{F}$ w grafie G używając procedury z faktu 4.3.
2. Pokoloruj właściwie wierzchołki $\overrightarrow{F}$ używając kolorów $\{1, 2, 3\}$ za pomocą algorytmu Cole-Vishkin ([2]).
3. Dla każdego nieizolowanego wierzchołka v wykonaj równolegle:
   (a) Jeżeli kolor wierzchołka v jest równy 1, to v zaznacza wszystkie krawędzie z $in(v, \{2, 3\})$ jeżeli $\bar{\omega}(in(v, \{2, 3\})) > \bar{\omega}(out(v, \{2, 3\}))$, w przeciwnym przypadku v zaznacza krawędź z $out(v, \{2, 3\})$.
   (b) Jeżeli kolor v równy jest 2 wtedy v zaznacza wszystkie krawędzie ze zbioru $in(v, \{3\})$ jeżeli $\bar{\omega}(in(v, \{3\})) > \bar{\omega}(out(v, \{3\}))$, w przeciwnym przypadku v zaznacza krawędź ze zbioru $out(v, \{3\})$.
4. Niech Q_i oznacza spójną składową grafu indukowanego przez zaznaczone krawędzie. Równolegle znajdź w każdej składowej Q_i wierzchołkowo rozłączne gwiazdy o wadze przynajmniej $\bar{\omega}(Q_i)/2$ i zwróć je (można to w prosty sposób wykonać ukorzeniając Q_i oraz rozważając parzyste i nieparzyste poziomy).

Lemat 4.5.

$$diam(Q_i) < 10.$$

Dowód:

Załóżmy nie wprost, że $diam(Q_i) \geqslant 10$. Wtedy z faktu 4.4 istnieje skierowana ścieżka o długości przynajmniej 5 (o pięciu krawędziach). W przypadku, gdy istnieje wewnętrzny wierzchołek v leżący na tej ścieżce o kolorze równym 1, to otrzymujemy sprzeczność, gdyż albo krawędź wychodząca albo wchodząca do/z wierzchołka v jest zaznaczona w kroku 3(a), ale nie obie.

W przeciwnym przypadku musi istnieć wewnętrzny wierzchołek koloru 2, z dwoma sąsiadami o kolorze równym 3. W punkcie 3(b) zostanie wybrana dokładnie jedna z tych krawędzi, co prowadzi do sprzeczności.

□

Lemat 4.6. Procedura ciężkie gwiazdy *zwraca gwiazdy o wadze co najmniej* $\bar{\omega}(G)/24$ *i działa w czasie* $O(\log^* |G|)$ *rund.*

Dowód:

Z faktu 4.3 otrzymujemy $\bar{\omega}(F) \geqslant \bar{\omega}(G)/6$. Każda krawędź w F posiada jeden koniec w kolorze 1 i drugi w kolorze $\{2, 3\}$ lub jeden koniec o kolorze 2 i drugi o kolorze 3. Podsumowując krawędzie zaznaczane w korku 3(a) oraz 3(b) tworzą podział krawędzi pseudo-lasu $E(F)$, w ten sposób, że waga $\bigcup Q_i \geqslant \frac{\bar{\omega}(F)}{2}$. Z nierówności wynika, że utworzone gwiazdy posiadają co najmniej połowę wagi składowej Q_i, przez co waga ich wynosi co najmniej $\bar{\omega}(G)/24$.

Pierwszy, trzeci i czwarty krok wymagają $O(1)$ rund, natomiast kolorowanie z kroku 2 może zostać znalezione w $O(\log^* |G|)$ rundach, co kończy dowód.

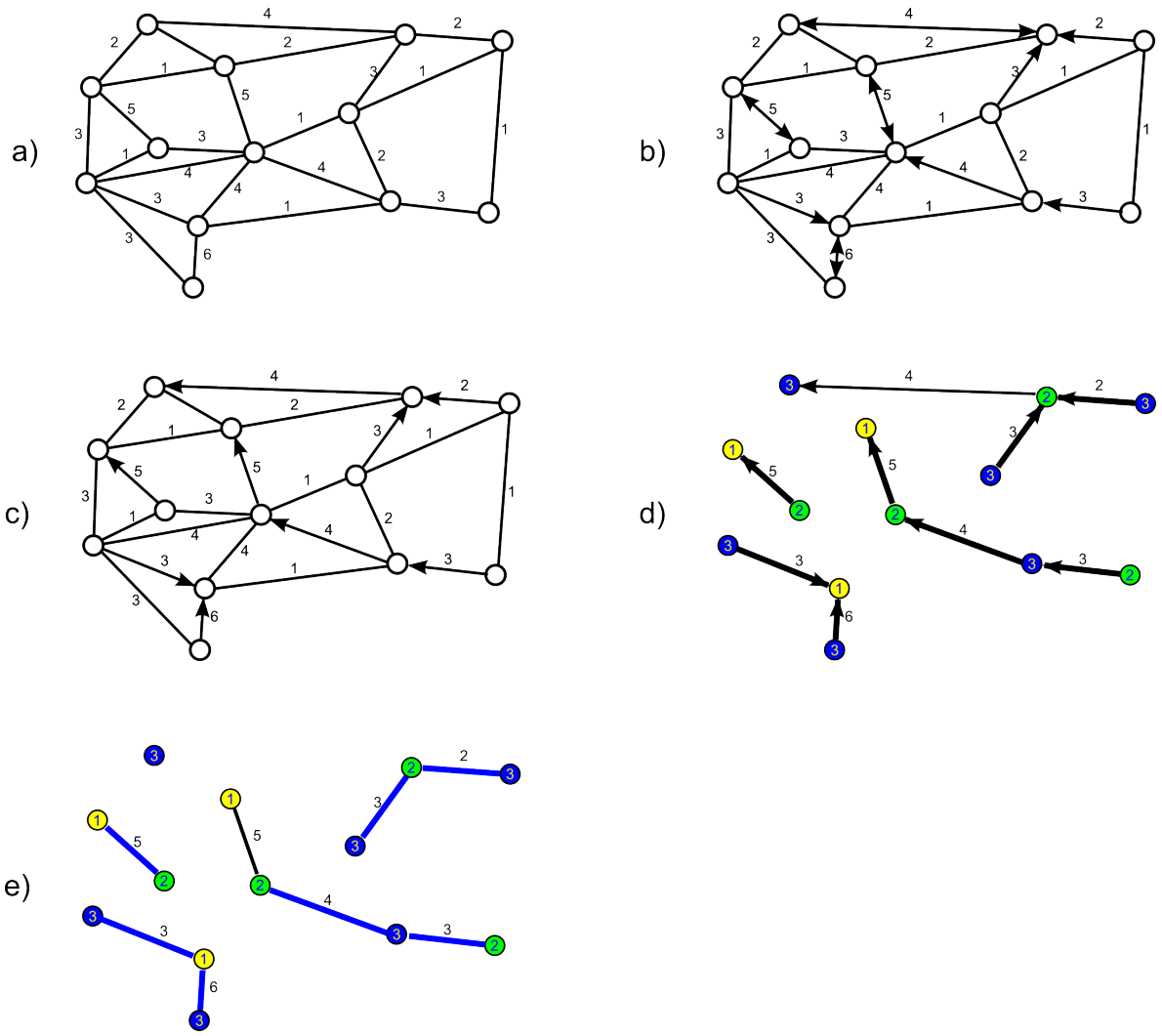
□

Procedura Klastrowania

Wejście: Planarny graf $G = (V, E)$, $\epsilon \in (0, 1)$.

Wyjście: Spójne podzbiory wierzchołków (klastry).

1. $H = G$.
2. Iteruj $\lceil \left(\log \left(\frac{1}{\epsilon} \right) / \log \left(\frac{24}{23} \right) \right) \rceil$ razy:
   (a) Wywołaj procedurę ciężkie gwiazdy w grafie H. Niech $S_1, S_2, \ldots S_k$ będą gwiazdami zwróconymi przez tę procedurę.



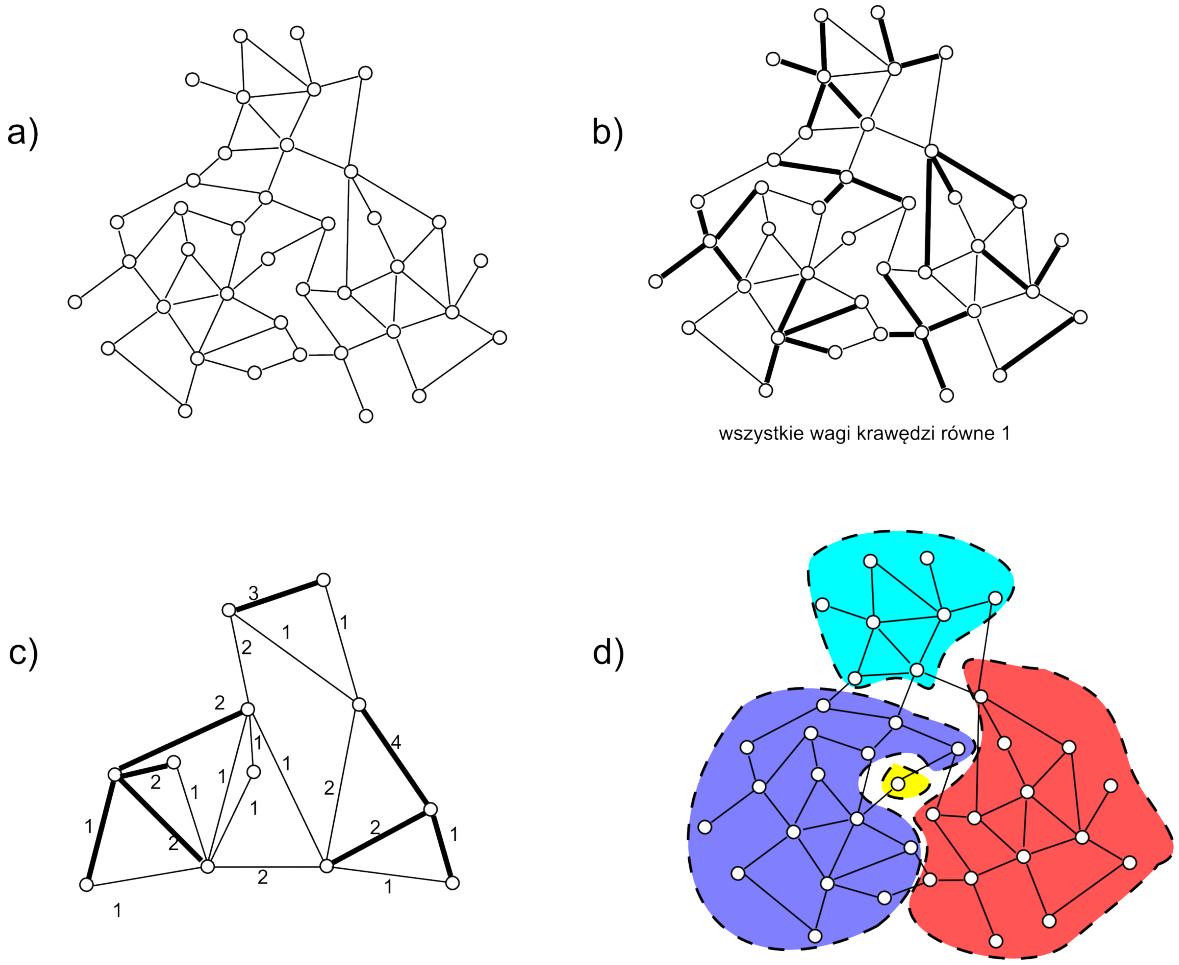
Rysunek 30: Przykład działania procedury znajdującej CIĘŻKIE GWIAZDY: a) wejściowy graf, b-c) znajdowanie pseudo-lasu, d) 3-kolorowanie w pseudo lesie oraz wykonanie kroku 3, e)znalezienie ciężkich gwiazd.

(b) Ściśnij gwiazdy $S_1, S_2, \ldots S_k$ do pojedynczych wierzchołków, a wagi w grafie oblicz zgodnie z (3).

3. Niech W oznacza zbiór wierzchołków ściśniętych do wierzchołka w z grafu $V(G)$. Zwróć zbiór $\{W|w \ \in V(H)\}$.

Lemat 4.7. PROCEDURA KLASTROWANIE $\forall \epsilon \in (0,1)$ *znajduje podział wierzchołków grafu* $V(G)$ *na rozłączne wierzchołkowo zbiory* $(V_1, \ldots, V_l)$ *takie, że waga grafu* $\tilde{G}$ *otrzymanego przez ściśnięcie każdego podzbioru* V_i *spełnia następującą nierówność*

$$\bar{\omega}(\tilde{G}) \leqslant \epsilon \bar{\omega}(G).$$



Rysunek 31: Przykład działania procedury klastrowania: a)graf wejściowy, b)zaznaczone ciężkie gwiazdy, c) operacja ściskania oraz zaznaczone nowe ciężkie gwiazdy, d) skonstruowane klastry w grafie wejściowym.

Algorytm działa w czasie $O(\log^* |G|)$ *rund.*

Dowód:

Z Lematu 4.6 w każdej iteracji algorytm ściska gwiazdy o wadze co najmniej $1/24$ wagi całego grafu. Po l iteracjach waga grafu H wynosi co najwyżej $\left(\frac{23}{24}\right)^l \bar{\omega}(G) \leqslant \epsilon\bar{\omega}(G)$, gdy $l = \lceil\left(\log\left(\frac{1}{\epsilon}\right)/\log\left(\frac{24}{23}\right)\right)\rceil$. Czas działania procedury wynosi $O(\log^* |G|)$ z Lematu 4.6.

□

4.2 Algorytm największego ważonego zbioru niezależnego.

Niech $G = (V, E, \omega)$ będzie ważonym grafem planarnym, gdzie $\omega : V \to R^+$. Dla każdej krawędzi $\{u, v\} \in E$ zdefiniujmy w następujący sposób jej wagę:

$$\bar{\omega}(\{u, v\}) = \min\{\omega(u), \omega(v)\}. \tag{4}$$

Fakt 4.8.

$$\bar{\omega}(G) \leqslant 3\ \omega(G).$$

Dowód:

Z twierdzenia Nasha-Williamsa graf G posiada orientacje wszystkich krawędzi o stopniu wyjściowym mniejszym bądź równym trzy. Dla każdej zorientowanej krawędzi (u, v) (z u do v) otrzymujemy, iż waga krawędzi wynosi co najwyżej $\omega(u)$. Każdy wierzchołek może być punktem startowym co najwyżej trzech krawędzi, co kończy dowód.

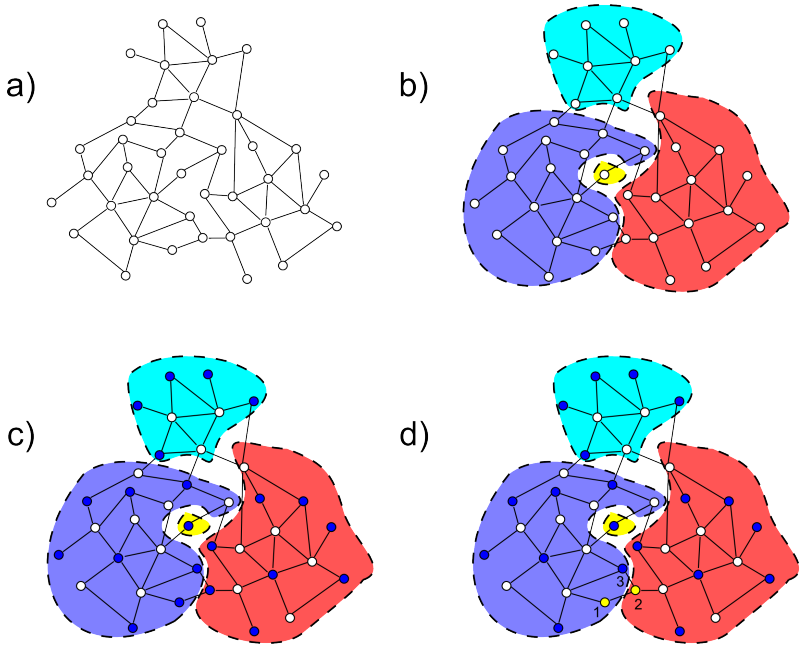
□

W celu aproksymowania największego ważonego zbioru niezależnego wywołamy zmodyfikowaną procedurę z pracy [6]. Działanie algorytmu przedstawię poniżej.

Algorytm MWIS
Wejście: Planarny graf $G = (V, E)$
Wyjście: Zbiór niezależny.

1. Wywołaj procedurę klastrowania. Niech $(V_1, \ldots, V_l)$ będą podzbiorami zwróconymi przez tę procedurę.
2. Równolegle dla każdej indukowanej składowej $G[V_j]$ znajdź optymalny zbiór niezależny i oznacz go jako I_j.
3. Dla każdego wierzchołka $u \in I_i$ usuń go ze zbioru I_i, jeżeli $\exists\{u, v\}$ takie, że $v \in I_j$ oraz $(\omega(u) < \omega(v))$ lub $(\omega(u) = \omega(v)$ i $id(u) < id(v))$.
4. Zwróć $\bigcup_I I_i$.

Zauważmy, iż krok 2 algorytmu możemy szybko wykonać ze względu na stałą $O(1)$ średnicę każdego indukowanego podgrafu $G[V_i]$. Używając analogicznego argumentu do dowodu z pracy [6] otrzymujemy:



Rysunek 32: Przykład działania algorytmu MWIS: a) graf wejściowy, b) klastry (zaznaczone różnymi kolorami), c) na niebiesko zaznaczony zbiór niezależny w każdym klastrze, d) na żółto zaznaczone zostały wierzchołki usuwane ze zbioru I w kroku 3 algorytmu. Większość wag na wierzchołkach zostało pominiętych ze względu na czytelność rysunku.

Twierdzenie 4.9. ALGORYTM MWIS *znajduje w planarnym grafie* $G = (V, E, \omega)$, *gdzie* $\omega : V \to R^+$ *zbiór niezależny* I *o wadze co najmniej* $(1 - \delta)OPT$, *gdzie* OPT *oznacza wagę największego ważonego zbioru niezależnego. Czas działania algorytmu wynosi* $O(\log^* |G|)$ *rund.*

Dowód:

Skoro G jest grafem planarnym, to $OPT \geqslant \omega(G)/4$. Niech I będzie sumą zborów niezależnych I_i po kroku 2 algorytmu. Ponieważ procedura znajduje optymalne rozwiązanie w każdej składowej V_i osobno, to $\omega(I) \geqslant OPT$. W kroku 3 dla każdej krawędzi pomiędzy dwoma klastrami usuwa koniec o mniejszej wadze, a więc waga usuniętych wierzchołków jest równa wadze krawędzi pomiędzy tymi wierzchołkami zgodnie z równaniem(4). Waga usuniętych wierzchołków jest mniejsza lub równa wadze krawędzi pomiędzy klastrami. Z Lematu 4.7 waga ta jest mniejsza lub równa

$$\epsilon\bar{\omega}(G) \leqslant 3\epsilon\omega(G) \leqslant 12\epsilon OPT = \delta OPT,$$

gdzie $\epsilon = \delta/12$.

□

4.3 Algorytm największego skojarzenia.

W przeciwieństwie do problemu zbioru niezależnego problemy: największego skojarzenia, czy zbioru dominującego potrafimy rozwiązać jedynie w przypadku nieważonych wersji tych problemów. Powodem jest fakt, iż w grafach planarnych waga optymalnego rozwiązania MWIS jest proporcjonalna do $\omega(G)$ (np. wynosi co najmniej $\omega(G)/4$). Podobne oszacowanie nie jest prawdziwe dla pozostałych rozważanych problemów.

Jednak w przypadku wersji nieważonych można zastosować procedurę redukującą graf G w ten sposób, by optymalne rozwiązanie było proporcjonalne do wielkości grafu G.

Stosując ideę algorytmu z pracy [8] możemy pokazać, że:

Twierdzenie 4.10. *Istnieje deterministyczny rozproszony algorytm, który dla pewnej ustalonej stałej $0 < \delta < 1$ znajduje w planarnym grafie $G = (V, E)$ w czasie $O(\log^* |G|)$ skojarzenie M takie, że*

$$|M| \geqslant (1 - \delta) OPT_{MM}(G),$$

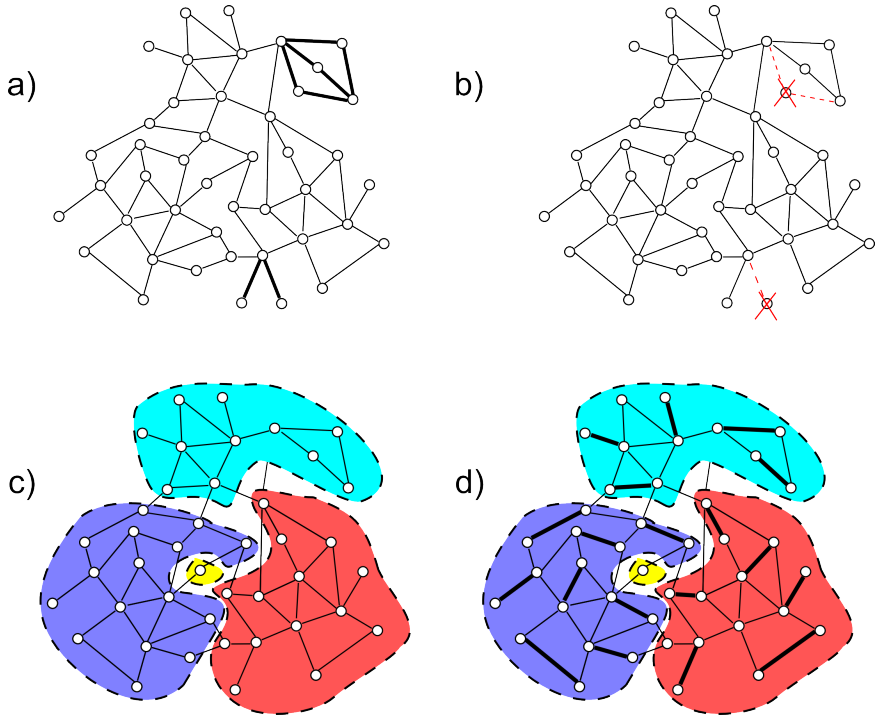
gdzie $OPT_{MM}(G)$ jest wielkością maksymalnego skojarzenia w grafie G.

Algorytm MM
Wejście: Planarny graf $G = (V, E)$
Wyjście: Skojarzenie w grafie G.

1. Ustal wagę 1 dla każdej krawędzi w grafie G.
2. Usuń indukowane gwiazdy oraz podwójne gwiazdy z grafu G tworząc nowy graf G'.
3. Wywołaj procedurę klastrowania w grafie G'. Niech $V_1, V_2, \ldots, V_k$ będą podzbiorami zwróconymi przez tę procedurę.
4. Znajdź optymalne rozwiązanie w każdym indukowanym podgrafie $G'[V_i]$ i oznacz je jako M_i.
5. Zwróć zbiór $\bigcup_i M_i$.

Dowód:

Powyższy algorytm jest analogiczny do procedury z pracy [8]. Z pracy [8] (Lematu 6) wiadomo, że po usunięciu indukowanych gwiazd oraz podwójnych indukowanych gwiazd z grafu G (co można wykonać w 2 rundach) moc optymalnego skojarzenia w nowym grafie G'



Rysunek 33: Przykład działania algorytmu MM: a) graf wejściowy z pogrubionymi krawędziami oznaczającymi 2-gwiazdy oraz 3-podwójne gwiazdy, b) usuwanie gwiazd oraz podwójnych gwiazd, c) klastry utworzone w punkcie 3 algorytmu, d) znalezione optymalne rozwiązanie w każdym z klastrów z osobna.

wynosi $\Omega(|V(G')|)$. W skojarzeniu może znaleźć się co najwyżej jedna krawędź z pojedynczej gwiazdy oraz 2 krawędzie z podwójnej gwiazdy, wówczas otrzymujemy, że $OPT_{MM}(G) = OPT_{MM}(G')$.

W kolejnym kroku w grafie G' wywołujemy PROCEDURĘ KLASTROWANIA i znajdujemy podział wierzchołków grafu G' (wagi na krawędziach są ustalane na początku na 1). W ostatnim kroku, w każdym indukowanym przez podział podgrafie znajdujemy optymalne rozwiązanie, co możemy wykonać w czasie $O(1)$ rund. Błąd aproksymacji może zostać oszacowany identycznie, jak w Twierdzeniu 4.9.

□

4.4 Algorytm minimalnego zbioru dominującego.

Możliwe jest skonstruowanie minimalnego zbioru dominującego w grafach planarnych w czasie sublogarytmicznym. Algorytm wykonujący tę operację podzielić można na dwie fazy. W pierwszej z nich znajdujemy przybliżone rozwiązanie problemu o stałym współczynniku

aproksymacji, natomiast w drugiej (zarazem ostatniej) w czasie $O(log^* n)$ poprawiamy zbiór dominujący tak, by był on aproksymacją minimalnego zbioru dominującego o współczynniku aproksymacji dowolnie bliskim 1.

Pogarszając minimalnie złożoność czasową algorytmu [17] uzyskujemy znacząco lepszy współczynnik aproksymacji.

Twierdzenie 4.11. *Niech $G = (V, E)$ będzie grafem planarnym, a δ będzie dowolnie ustaloną liczbą taką, że $\delta \in (0, 1)$. Istnieje wówczas deterministyczny rozproszony algorytm znajdujący w grafie G minimalny zbiór dominujący w czasie $O(\log^* |V(G)|)$ rund oraz o współczynniku aproksymacji wynoszącym*

$$|D| \leqslant (1 + \delta)|OPT_{MDS}|,$$

gdzie OPT_{MDS} jest minimalnym zbiorem dominującym w grafie G.

Algorytm MDS

Wejście: Planarny graf $G = (V, E)$

Wyjście: Zbiór dominujący w grafie G.

1. Wywołaj procedurę znajdującą 72-aproksymację minimalnego zbioru dominującego w grafie G. Niech $D = \{w_1, w_2, \ldots, w_k\}$ będzie zwróconym zbiorem dominującym.

2. Każdy wierzchołek $v \in (V \setminus D)$ wybiera dowolny wierzchołek go dominujący tworząc gwiazdy $(W_1, W_2, \ldots, W_k)$.

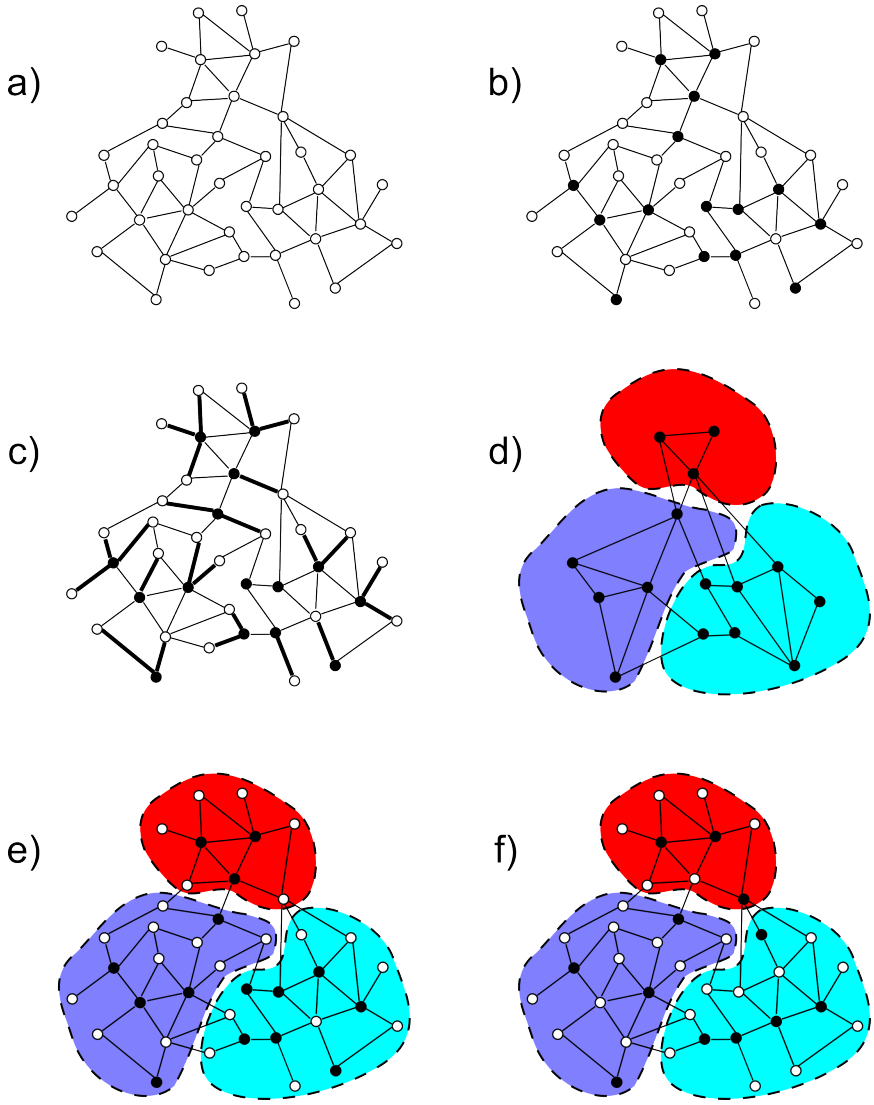
3. Ściśnij każdą gwiazdę $W_1, W_2, \ldots, W_k$ do wierzchołków $w_1, w_2, \ldots w_k$ tworząc planarny graf H. Ustal wagę każdej krawędzi w grafie równą 1.

4. Wywołaj procedurę klastrowania w grafie H znajdując podział $(V_1, V_2, \ldots, V_l)$ zbioru wierzchołków $V(H)$.

5. Na wierzchołkach ze zbioru V_i wykonaj operację odwrotną do "ściskania" tworząc podział $(U_1, U_2, \ldots, U_l)$ grafu $V(G)$.

6. W każdym indukowanym wierzchołkowo podgrafie $G[U_i]$ znajdź optymalny minimalny zbiór dominujący i oznacz go jako D_i.

7. Zwróć zbiór $Q = \bigcup_i D_i$.



Rysunek 34: Przykład działania algorytmu MDS: a) graf wejściowy, b) przykładowa 72-aproksymacja z kroku 1 algorytmu, c) gwiazdy utworzone w punkcie 2 algorytmu, d) klastry utworzone w grafie H z punktu 3 algorytmu, e) podział $(U_1, U_2, \ldots, U_l)$ grafu $V(G)$, f) znalezione optymalne rozwiązanie problemu w każdym klastrze z osobna.

Dowód:

W pierwszej kolejności udowodnię, że zbiór zwracany w kroku 7 algorytmu jest zbiorem dominującym. Rozważmy utworzony podział grafu na rozłączne zbiory wierzchołków $(U_1, U_2, \ldots, U_l)$ w kroku 5 algorytmu. W każdym z nich z osobna algorytm znajduje optymalne rozwiązanie problemu D_i. Oczywistym jest, że $\underset{v \in V}{\forall} \underset{i}{\exists} V \in U_i$. Oznacza to, że dowolny wierzchołek $v \in V$ dominowany jest przez pewien zbiór D_i.

Każdy z kroków algorytmu wymaga co najwyżej $O(log^* n)$ rund, co dowodzi ograniczenia złożoności czasowej algorytmu do $O(\log^* |G|)$.

W celu udowodnienia współczynnika aproksymacji algorytmu liczbę krawędzi w grafie H można oszacować w następujący sposób $|E(H)| < 3k$, co wynika bezpośrednio z twierdzenia Eulera dla grafów planarnych. Jako stałą ϵ przyjmijmy wartość $\epsilon = \delta c/6$. Po wywołaniu procedury KLASTROWANIA waga konwektorów (krawędzi pomiędzy klastrami) wnosi co najwyżej $\epsilon|E(H)|$. Ponieważ rozważamy każdy z podgrafów $G[U_i]$ z osobna, to zbiór $Q = \bigcup_i D_i$ zwracany w kroku 7 algorytmu może nie być optymalny. Dowolny minimalny zbiór dominujący oznaczmy jako D^*. Niech $\bar{D}$ będzie zbiorem otrzymanym ze zbioru D^* przez dodanie wszystkich wierzchołków w_i takich, że wierzchołek ze zbioru $W_i \subset U_j$ posiada sąsiada w zbiorze $V \setminus U_j$.

Stąd

$$|\bar{D}| \leqslant |D^*| + 2\ \epsilon|E(H)| < |OPT_{MDS}(G)| + 6\epsilon|D| \leqslant |OPT_{MDS}(G)|(1+\delta).$$

Skoro $\bar{D} \cap U_i$ jest zbiorem dominującym w podgrafie indukowanym $G[U_i]$, a D_i jest optymalnym rozwiązaniem w podgrafie indukowanym $G[U_i]$, to

$$|\bigcup D_i| \leqslant |\bar{D}|,$$

co dowodzi

$$|\bigcup D_i| \leqslant (1+\delta)OPT_{MDS}(G).$$

□

5 Oszacowanie dolne.

Oszacowanie dolne złożoności czasowej algorytmów korzysta z jednego z podstawowego twierdzenia z teorii Ramsey-a ([26]). Na podstawie pracy [20] wiadomo, że twierdzenie to może posłużyć do pokazania nie istnienia deterministycznego rozproszonego algorytmu kolorującego właściwie cykl C w czasie $o(\log^* |C|)$ rund używając $O(1)$ kolorów. Twierdzenie to zachodzi również w przypadku algorytmów aproksymujących, co zostanie pokazane poniżej.

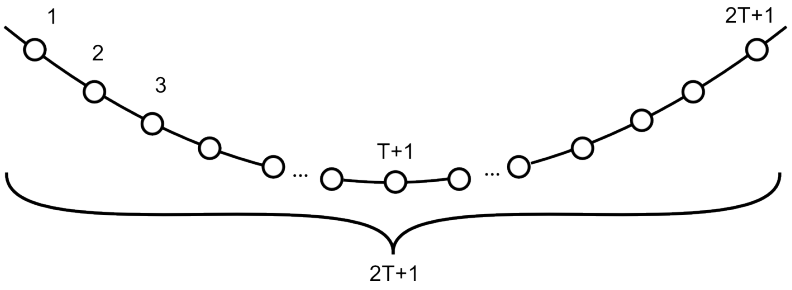
Niech $R(2, m; l)$ oznacza najmniejszą liczbę wierzchołków n takich, że w dowolnym 2-kolorowaniu krawędzi pełnego l-jednorodnego hipergrafu na n wierzchołkach istnieje monochromatyczny pełny podgraf o m wierzchołkach. Liczba ta istnieje i możliwe jest jej oszacowanie za pomocą funkcji wieży.

Twierdzenie 5.1. *Dla dowolnej dodatniej liczby całkowitej nie istnieje deterministyczny algorytm, który w cyklu na n wierzchołkach znajduje zbiór niezależny o wielkości $\Theta(n/\log_{(2T)} n)$ w T rundach. Nie istnieje również deterministyczny rozproszony algorytm, który znajduje w cyklu na n wierzchołkach zbiór niezależny o wielkości $\Theta(n/\log^* n)$ w $\log^* n/2$ rundach.*

Dowód:

Dla uproszczenia załóżmy, że jeżeli $S = \{i_1, \ldots, i_l\}$ jest zbiorem o l elementach, wówczas są one ponumerowane w następującej kolejności $i_k < i_l$, gdy $k < l$. Niech C będzie cyklem na wierzchołkach $[n] = \{1, \ldots, n\}$.

Załóżmy, że mamy dany rozproszony algorytm A znajdujący w T rundach zbiór niezależny w cyklu C na zbiorze wierzchołków $[n]$. Niech $F_A : \binom{[n]}{2T+1} \to \{0, 1\}$ będzie następującą funkcją: $F_A(\{i_1, \ldots, i_{2T+1}\}) = 1$ wtedy i tylko wtedy, gdy wierzchołek i_T wybrany zostanie do zbioru niezależnego przez algorytm A w cyklu zawierającym ścieżkę $i_1, i_2, \ldots i_{2T+1}$.



Rysunek 35: Przykład części grafu $[n]$.

Funkcja F_A jest 2-kolorowaniem właściwym krawędzi pełnego $(2T + 1)$-jednorodnego podgrafu H na zbiorze wierzchołków $[n]$. Z twierdzenia Ramsey-a hipergraf H zawiera monochromatyczny pełny podgraf K na m wierzchołkach. Zauważmy, że jeżeli $m > 2T + 1$ to

kolor K nie może być równy 1, ponieważ jeżeli ścieżka $\{i_1, \ldots, i_{2T+2}\}$ byłaby podgrafem K, to kolor ten implikowałoby, że wierzchołki i_T oraz i_{T+1} wybrane zostałyby do zbioru niezależnego. Otrzymujemy jednak sprzeczność z uwagi, ponieważ wierzchołki te są połączone krawędzią w ścieżce $P = i_1, \ldots, i_{2T+2}$ w cyklu C.

W rezultacie każda krawędź w grafie K jest koloru 0 i jeżeli $K = \{v_1, \ldots, v_m\}$ to żaden z wierzchołków $v_{T+1}, \ldots, v_{m-T}$ nie zostanie zwrócony przez algorytm A. Stąd aż $m - 2T$ z m wierzchołków nie należy do zbioru niezależnego.

Jeżeli $n - m \geqslant R(2, m; 2T + 1)$ to argument ten możemy powtarzać wielokrotnie i niech l oznacza liczbę tych iteracji. Wielkość zbioru niezależnego zwracanego przez algorytm A wynosi więc, co najwyżej $2Tl + R(2, m; 2T + 1)$, a tym samym co najwyżej

$$2nT/m + R(2, m; 2T + 1),$$

gdy $lm \leqslant n$.

Z teorii Ramsey'a wiadomo, że dla pewnej ustalonej stałej c,

$$R(2, m; 2T + 1) \leqslant 2^{2^{\cdots^{2^{cm}}}},$$

gdzie ilość "dwójek" w funkcji wieży wynosi $2T$. Stąd dla dowolnej stałej T, gdy $m = \Theta(\log_{(2T)} n)$ otrzymujemy, że wielkość zbioru niezależnego wynosi co najwyżej $O(n/\log_{(2T)} n)$. Ponadto po przekształceniach uzyskujemy, że otrzymanie zbioru niezależnego o wielkości $\Omega(n/\log^* n)$, T wymaga $\Omega(\log^* n)$ rund, co kończy dowód.

□

Podobne oszacowanie zachodzi w przypadku problemu największego skojarzenia w grafie planarnym. Twierdzenia te dowodzą, że algorytmy sublogarytmiczne zaproponowane w poprzednim rozdziale są optymalne pod względem złożoności czasowej z dokładnością do stałego czynnika C.

Fakt 5.2. *Nie istnieje deterministyczny rozproszony algorytm rozwiązujący problem największego skojarzenia w grafach planarnych o stałym współczynniku aproksymacji w czasie* $o(\log^* |V|)$.

Fakt ten wynika bezpośrednio z Twierdzenia 5. Załóżmy bowiem nie wprost, iż istnieje algorytm znajdujący c-aproksymację największego skojarzenia w cyklu. Niech M będzie skojarzeniem zwróconym przez tę procedurę.

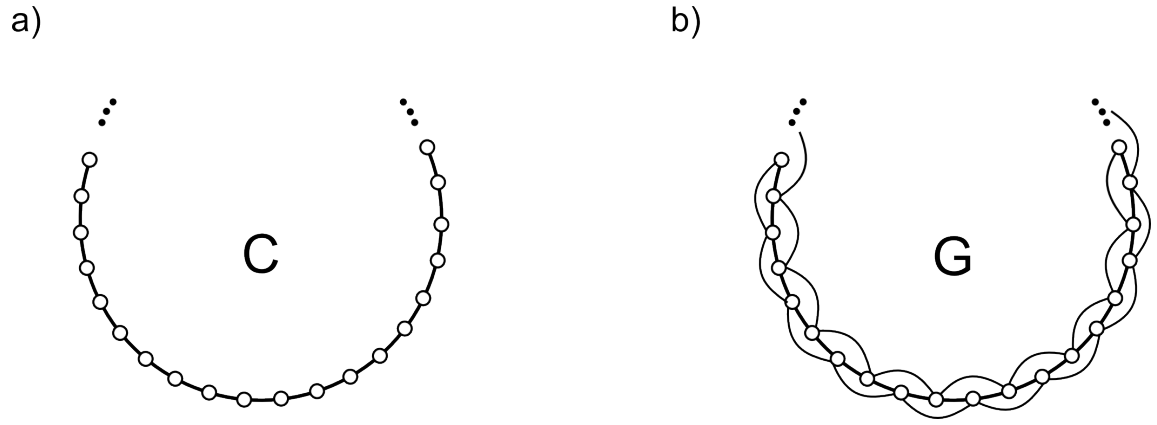
Rozważmy procedurę dodającą do zbioru niezależnego I wierzchołki o mniejszym id z każdej pary wierzchołków połączonych krawędzią ze zbioru M Następnie z każdej pary

wierzchołków przyległych w zbiorze I niech algorytm usuwa wierzchołek o większym id. W ten sposób otrzymujemy sprzeczność, gdyż moc znalezionego zbioru niezależnego $|I|$ wynosi co najmniej $|M|/2$.

Fakt 5.3. *Nie istnieje deterministyczny rozproszony algorytm rozwiązujący dla dowolnej stałej $\delta > 0$ problem minimalnego zbioru dominującego w grafach planarnych $G = (V, E)$ o współczynniku aproksymacji $(5 - \delta)$ w czasie $o(\log^* |V|)$.*

Dowód:

Niech δ będzie dowolnie ustaloną stałą. Załóżmy nie wprost, iż istnieje deterministyczny rozproszony algorytm A znajdujący minimalny zbiór dominujący o mocy co najwyżej $(5 - \delta)|OPT_{MDS}|$, gdzie OPT_{MDS} jest minimalnym zbiorem dominującym.



Rysunek 36: Przykład grafu G powstałego z cyklu C.

Załóżmy, że mamy dany cykl C na n wierzchołkach (gdzie n jest podzielne przez 10). Zdefiniujmy 4-regularny graf $G = (V, E)$, gdzie $V = V(C), E = E(C) \cup \{v, u \in V, d_C(v, U) = 2\}$. Możemy łatwo zauważyć, iż moc optymalnego zbioru dominującego wynosi $|OPT_{MDS}(G)| = n/5$ oraz graf G jest planarny. Niech D będzie zbiorem dominującym zwróconym przez algorytm A uruchomionym w grafie G. Oznaczmy jako D_i zbiór wierzchołków w grafie indukowanym wierzchołkowo $C[D]$ o stopniu i.

Niech zbiór I będzie zbiorem niezależnym skonstruowanym następująco:

- Dodaj wierzchołki izolowane w grafie indukowanym $C[D]$ do zbioru niezależnego ($I = D_0$).
- $\underset{v \in D_1}{\forall}$ jeżeli $N_{C[D]}(v) \cap D_1 = \emptyset$ lub ($u \in N_{C[D]}(v)$ oraz $id(v) < id(u)$) dodaj wierzchołek do zbioru I.

Zauważyć możemy, że $|I| \geqslant |D_0|+|D_1|/2$. Ponadto prawdziwa jest nierówność $4|D_0|+3|D_1| \geqslant n-|D|$ wynikająca z oszacowania liczby wierzchołków w cyklu C leżących poza zbiorem D. Ponieważ algorytm A zwraca zbiór dominujący to każdy wierzchołek z tego zbioru musi być zdominowany przez wierzchołki ze zbiorów D_0, D_1, D_2. Maksymalna łączna liczba dominowanych wierzchołków przez te zbiory w grafie G wynosi $4|D_0|+3|D_1|$. Zaznaczmy, że wszystkie wierzchołki dominowane przez zbiór D_2 dominowane są również przez wierzchołki ze zbioru D_1.

Z założenia wiemy, iż moc zbioru D możemy oszacować następująco $|D| \leqslant (5-\delta)|OPT_{MDS}|$, a zatem

$$|D| \leqslant (1-\frac{\delta}{5})n.$$

Podstawiając to oszacowanie do wcześniejszej nierówności otrzymujemy:

$$4|D_0|+3|D_1| \geqslant \frac{\delta n}{5},$$

oraz

$$|I| \geqslant |D_0| + \frac{|D_1|}{2} \geqslant \frac{4|D_0|}{6} + \frac{|D_1|}{2} \geqslant \frac{\delta n}{30}.$$

Sprzeczność z Twierdzeniem 5, co kończy dowód.

□

Literatura

[1] B. Awerbuch, A. V. Goldberg, M. Luby, S. A. Plotkin, Network Decomposition and Locality in Distributed Computation, Proc. 30th IEEE Symp. on Foundations of Computer Science , 1989, pp. 364-369.

[2] R. Cole, U.Vishkin "Deterministic coin tossing with applications to optimal parallel list ranking", Information and Control 70 (1): 3253, (1986);

[3] T. H. Cormen, C. E. Leiserson , R. L. Rivest : "Wprowadzenie do algorytmów", Wydawnictwo Naukowo Techniczne (2005), wydanie 6.

[4] A. Czygrinow, M. Hańćkowiak, "Distributed almost exact approximations for minor-closed families", LNCS 4168, (2006), 244–255.

[5] A. Czygrinow, M. Hańćkowiak, "Distributed Approximations for Packing in Unit-Disk Graphs", LNCS 4731, (2007), 152–164.

[6] A. Czygrinow, M. Hańćkowiak, Distributed algorithms for weighted problems in sparse graphs, Journal of Discrete Algorithms, Vol 4, (4), (2006), 588–607. 14th Annual European Symposium on Algorithms (ESA), (2006), 244–255.

[7] A. Czygrinow, M. Hańćkowiak, Distributed approximation algorithms for weighted problems in minor-closed families, The 13th Annual International Computing and Combinatorics Conference, COCOON 2007, LNCS 4598, (2007), 515–525.

[8] A. Czygrinow, M. Hańćkowiak, E. Szymanska, Distributed approximation algorithms for planar graphs, 6th Conference on Algorithms and Complexity (CIAC 20006), LNCS 3998, (2006), 296–307.

[9] A. Czygrinow , M. Hańćkowiak, W. Wawrzyniak, "Distributed packing in planar graphs", Proceedings of the 20th Annual ACM Symposium on Parallel Algorithms and Architectures, Munich, Germany, June 14-16, (2008), 55–61.

[10] A. Czygrinow , M. Hańćkowiak, W. Wawrzyniak, "Fast Distributed Approximations in Planar Graphs", LNCS 5218, (2008), 78–92.

[11] A. Czygrinow, K. Krzywdziński, E. Szymańska, W.Wawrzyniak "Fast distributed approximation algorithms for the minimum vertex cover problem", manuskrypt (2009)

[12] J.Edmonds "Paths, trees, and flowers". Canad. J. Math. 17: 449467 (1965)

[13] M. Hańćkowiak, M. Karoński, A. Panconesi, " On the distributed complexity of computing maximal matchings", SIAM J. Discrete Math. Vol.15, No.1, (2001), 41–57.

[14] F. Kuhn, T. Moscibroda, R. Wattenhofer, "Lower and Upper Bounds for Distributed Packing and Covering", Technical Report, ETH Zurich, Dept. of Computer Science (2004).

[15] F. Kuhn, T. Moscibroda, T. Nieberg, and R. Wattenhofer, "Fast Deterministic Distributed Maximal Independent Set Computation on Growth-Bounded Graphs", 19th International Symposium on Distributed Computing (DISC), Cracow, Poland, September (2005), 273–287.

[16] F. Kuhn, T. Moscibroda, T. Nieberg, and R. Wattenhofer, "Local Approximation Schemes for Ad Hoc and Sensor Networks", 3rd ACM Joint Workshop on Foundations of Mobile Computing (DIALM-POMC), Cologne, Germany, (2005), 97–103.

[17] C. Lenzen, Y. A. Oswald, R. Wattenhofer, Roger "—What can be approximated locally?", Proceedings of the twentieth annual symposium on Parallelism in algorithms and architecturs (2008), 46–54.

[18] N.Linial "Locality in distributed graph algorithms", SIAM Journal on Computing 21 (1): (1992), 193-201.

[19] C. St J. A. Nash-Williams. Decomposition of finite graphs into forests. Journal of the London Mathematical Society 39:12, (1964),

[20] A. Panconesi, "Distributed Algorithms Notes", (manuskrypt).

[21] A. Panconesi, R. Rizzi "Some simple distributed algorithms for sparse networks", Distributed Computing, vol 14 n.2, (2001) 97–100 .

[22] C.H. Papadimitriou, K. Steiglitz, "Combinatorial optimization: algorithms and complexity" (1998), 432

[23] V. Polishchuk and J. Suomela, A simple local 3-approximation algorithm for vertex cover, Information Processing Letters 109 (2009), 642–645.

[24] V. Polishchuk and J. Suomela, "A local 2-approximation algorithm for the vertex cover problem" LNCS 5805, (2009), 191–205.

[25] J. Schneider, R. Wattenhofer "A log-star distributed maximal independent set algorithm for growth-bounded graphs", proceedings of the Twenty-Seventh ACM Symposium on Principles of Distributed Computing (Toronto, Canada, August 18 - 21, 2008). PODC '08. ACM, New York (2008), 35-44

[26] D. West, Introduction to graph theory, 2nd Ed., Penitence-Hall Inc., (2001).