

Uniwersytet im Adama Mickiewicza w Poznaniu

Wydział Matematyki i Informatyki

Michał Hańkowiak

Praca doktorska  
napisana pod kierunkiem  
prof. dr hab. Michała Karońskiego

Poznań 2001

# Spis treści

<b>1</b>	<b>Wstęp.</b>	<b>3</b>
<b>2</b>	<b>Algorytm znajdujący maksymalne skojarzenie.</b>	<b>6</b>
2.1	Definicje i ogólny opis algorytmu. . . . .	6
2.2	Znajdowanie skojarzenia koszącego w dowolnym grafie. . . . .	7
2.3	Znajdowanie skojarzenia koszącego w grafie dwudzielnym. . . . .	11
2.4	Znajdowanie spannera w pojedynczym bloku. . . . .	15
<b>3</b>	<b>Algorytm znajdujący maksymalne <math>f</math>-skojarzenie.</b>	<b>27</b>
3.1	Definicje. . . . .	27
3.2	Ogólny opis algorytmu. . . . .	29
3.3	Znajdowanie $f$ -skojarzenia $p$ -koszącego w grafie. . . . .	32
3.4	Znajdowanie $f$ -skojarzenia koszącego lewostronnie w grafie dwudzielnym. . . . .	42
3.5	Znajdowanie PMfM w grafie. . . . .	49

# 1 Wstęp.

W rozproszonym/synchronicznym modelu obliczeń mamy do czynienia z grafem komunikacyjnym, którego wierzchołki są procesorami a krawędzie kanałami komunikacyjnymi. Każdy wierzchołek może wysyłać komunikaty do (bezpośrednich) sąsiadów, poprzez krawędzie grafu komunikacyjnego. Model jest synchroniczny, gdyż obliczenia wykonuje się w **rundach**: w każdej rundzie wszystkie procesory wysyłają komunikaty do sąsiadów, odbierają komunikaty od sąsiadów oraz wykonują lokalne obliczenia. W danej chwili wszystkie procesory wykonują obliczenia w tej samej rundzie.

Nie stawiamy żadnych ograniczeń co do długości i liczby komunikatów wysyłanych w czasie działania algorytmu, a także co do czasu trwania lokalnych obliczeń oraz objętość pamięci pojedynczego procesora. Zakładamy, że wierzchołki mają przydzielone (różne) identyfikatory, które można zapisać na  $O(\log n)$  bitach, gdzie  $n$  to liczba wierzchołków grafu komunikacyjnego. Zazwyczaj zakładamy, że wszystkie wierzchołki znają wartość  $n$ , nie mają natomiast żadnej innej wiedzy na temat grafu komunikacyjnego.

Będziemy się zajmować jedynie algorytmami obliczającymi pewne teorio-grafowe funkcje w grafie komunikacyjnym (takie jak np maksymalne skojarzenie), co różni nasz model od innych modeli obliczeń, w których dane problemu są dostarczane ze "świata zewnętrznego".

Omawiane w tej pracy algorytmy są projektowane pod kątem czasu działania tj liczby rund jakich, w najgorszym przypadku, wymaga dany algorytm. Zauważmy, że jeśli pewien algorytm działa w czasie  $t$ , to możemy to zinterpretować w ten sposób, że każdy wierzchołek "widzi" w grafie komunikacyjnym kulę o promieniu  $t$  i o środku w sobie samym, następnie oblicza wartość pewnej funkcji, zależącą od kuli. Otrzymana wartość staje się wynikiem działania algorytmu na tym wierzchołku (np może to być kolor wierzchołka, jeśli problem to "kolorowanie wierzchołkowe"). Może się wydawać dziwne, że niektóre nietrywialne funkcje grafowe można obliczyć przy bardzo małym  $t$ : np MIS, czyli maksymalny zbiór niezależny wierzchołków, w cyklu o długości  $n$ , można znaleźć w czasie  $O(\log^* n)$ .

Studia nad rozproszonymi algorytmami grafowymi rozpoczęły się prawdopodobnie wraz z pracą [L92], w której przedstawiono dolne oszacowanie  $\Omega(\log^* n)$  na czas potrzebny do obliczenia MIS-a w cyklu o długości  $n$ . Wcześniej, w pracy [CV86], pojawił się algorytm znajdujący MIS w cyklu, w czasie  $O(\log^* n)$ . Jak widać, złożoność problemu MIS dla cyklu można wyznaczyć dokładnie (z dokładnością do stałego czynnika), co zdarza się bardzo rzadko.

Jeśli graf komunikacyjny ma stały maksymalny stopień, to okazuje się, że wiele klasycznych funkcji grafowych, a mianowicie: MIS, maksymalne skojarzenie,  $(\Delta + 1)$ -kolorownie wierzchołkowe, a nawet  $\Delta$ -kolorownie wierzchołkowe, daje się obliczyć w czasie  $O(\log^c n)$ , gdzie  $n$  jest liczbą wierzchołków, a  $c$  stałą niezależną od  $n$ , tj w czasie poli-logarytmicznym.

Sytuacja zmienia się, gdy nie stawiamy żadnych ograniczeń wobec maksymalnego stopnia grafu komunikacyjnego. Dla następujących problemów: MIS,  $(\Delta + 1)$ -kolorownie wierzchołkowe,  $\Delta$ -kolorownie wierzchołkowe, najlepsze znane *deterministyczne* algorytmy działają w raczej długim czasie  $O(n^{\epsilon(n)})$ , gdzie  $\epsilon(n)$  jest funkcją powoli zbieżną do zera ( $\epsilon(n) = 1/\sqrt{\log n}$ ). Rozwiązania te opierają się na redukcji do problemu obliczania tzw  $(n^{\epsilon(n)}, n^{\epsilon(n)})$ -dekompozycji, którą można znaleźć przy pomocy bardzo złożonego,

deterministycznego algorytmu działającego w czasie  $O(n^{O(\epsilon(n))})$  [PS96]. Z drugiej strony  $O(\Delta)$ -kolorowanie wierzchołkowe (z dużym stałym czynnikiem przy  $\Delta$ ) można obliczyć deterministycznie w niezwykle krótkim czasie  $O(\log^* n)$ , co zostało pokazane w pracy [MP00], jednak kosztem bardzo czasochłonnych obliczeń lokalnych.

Dla wspomnianych wyżej funkcji grafowych istnieją proste *losowe* algorytmy, działające w poli-logarytmicznym czasie, które często są kopią algorytmów dla maszyny PRAM. Także  $(\log n, \log n)$ -dekompozycję można obliczyć losowo w czasie  $O(\log^2 n)$  [LS93]. Przy pewnych dodatkowych założeniach, algorytmy losowe potrafią być bardzo szybkie: przykładowo istnieje możliwość obliczania kolorowania krawędziowego  $(1 + \epsilon)\Delta$ -kolorami, dla małego  $\epsilon$ , w czasie  $O(\log \log n)$  [GP97].

Przedstawimy teraz definicje potrzebne, aby bardziej precyzyjnie opisać zawartość niniejszej pracy. Oczywiście będziemy, w miarę możliwości, używać tradycyjnej terminologii zaczerpniętej z teorii grafów. Dwie krawędzie grafu nazywamy incydentnymi, jeśli mają wspólny koniec. Podgraf nazywamy *skojarzeniem*, jeśli żadne dwie krawędzie tego podgrafu nie są incydentne. Mówimy, że wierzchołek *jest skojarzony*, jeśli jest końcem pewnej krawędzi należącej do skojarzenia. Skojarzenie nazywamy *maksymalnym* (skrót MM = Maximal Matching), jeśli każdy wierzchołek grafu jest albo skojarzony, albo wszyscy jego sąsiedzi są skojarzeni (zauważmy, że nie chodzi tu o skojarzenie o największej możliwej liczbie krawędzi).

W rozdziale drugim pracy zostanie przedstawiony rozproszony, synchroniczny i deterministyczny algorytm znajdujący MM w czasie  $O(\log^4 n)$ . Jest to pierwszy rozproszony algorytm znajdujący MM w czasie poli-logarytmicznym. Główna zasada działania tego algorytmu jest częściowo wzorowana na algorytmie znajdującym MM na maszynie PRAM [IS86]. Polega ona na konstruowaniu ciągu coraz mniejszych podgrafów, których stopnie "dzielą się przez dwa", dzięki czemu na końcu otrzymuje się podgraf stałego stopnia, zawierający większość wierzchołków grafu wejściowego, który można zamienić na "duże" skojarzenie. Problem polega jednak na tym, że niektóre funkcje grafowe można szybko obliczyć na maszynie PRAM, natomiast nie da się tego zrobić (w czasie poli-logarytmicznym) w rozproszonym modelu obliczeń. Przykładem takiej funkcji jest cykl Eulera (a właśnie cyklami Eulera posługiwali się Israeli i Siloach w swoim algorytmie).

Pierwsza wersja omawianego algorytmu, która ukazała się w [HKP98], działała w czasie  $O(\log^7 n)$ , następnie rozmaite elementy algorytmu zostały ulepszone, dzięki czemu czas działania spadł do  $O(\log^4 n)$  [HKP99], [HKP01]. Wydaje się, że tego rezultatu nie da się poprawić, o ile główna zasada działania algorytmu nie zostanie zmieniona.

Zauważmy, że skojarzenie w grafie można zdefiniować jako podgraf, którego wierzchołki mają stopnie mniejsze lub równe 1. Pojęcie to można więc w naturalny sposób uogólnić do tzw  $f$ -skojarzeń: wierzchołki grafu mają przyporządkowane *pojemności*  $f(v)$  - są to dowolne liczby naturalne. Podgraf nazywamy  $f$ -skojarzeniem, gdy stopnie jego wierzchołków są mniejsze lub równe od swoich pojemności. Jeśli pewien wierzchołek ma stopień w  $f$ -skojarzeniu równy pojemności, to taki wierzchołek nazywamy *nasyconym*.  $f$ -skojarzenie nazywamy *maksymalnym* (skrót MfM = Maximal  $f$ -Matching), gdy każdy wierzchołek grafu jest albo nasycony, albo wszyscy jego sąsiedzi są nasyчени. Możemy więc postawić problem znajdowania MfM, dla dowolnych pojemności, w rozproszonym modelu obliczeń. Dodatkowo, możemy zadać pytanie czy problem MfM jest obliczeniowo trudniejszy od problemu MM.

W rozdziale trzecim przedstawimy rozproszony, synchroniczny i deterministyczny algorytm znajdujący maksymalne  $f$ -skojarzenie w czasie  $O(\log^4 n \log^* n)$ , a więc tylko o czynnik  $\log^* n$  gorszym niż w przypadku "zwykłych" maksymalnych skojarzeń, gdzie  $\log^* n := \min\{i : \log^{(i)} n < 1\}$  a  $\log^{(i)} n$  oznacza  $i$ -krotne złożenie operacji logarytmowania. Funkcja  $\log^* n$  przyjmuje bardzo małe wartości, tak więc możemy powiedzieć, że problem MfM nie jest obliczeniowo trudniejszy od problemu MM, w rozproszonym modelu obliczeń.

W algorytmie znajdującym MfM pojawiają się pewne całkiem nowe zagadnienia, specyficzne dla  $f$ -skojarzeń, na przykład te z którymi boryka się procedura ZAMIANA. Oprócz tego, problem MfM wymusił zastosowanie pewnych rozwiązań, które okazały się przydatne także w algorytmie znajdującym MM (za przykład może tu posłużyć pojęcie bloku) i które przyczyniły się do zmniejszenia czasu działania tamtego algorytmu, lub pozwoliły na bardziej ogólne sformułowanie niektórych procedur, umożliwiające naturalne przejście od skojarzeń do  $f$ -skojarzeń.

## 2 Algorytm znajdujący maksymalne skojarzenie.

### 2.1 Definicje i ogólny opis algorytmu.

Przedstawimy teraz definicje i oznaczenia, które będą używane w całej pracy.

Grafem  $G$  nazywamy parę  $(V, E)$ , gdzie  $V$  jest zbiorem wierzchołków, a  $E$  zbiorem krawędzi. Będziemy rozpatrywać jedynie tzw grafy proste, czyli nie posiadające wielokrotnych krawędzi między dwoma wierzchołami oraz nie posiadające pętli, czyli krawędzi łączących jeden wierzchołek. Zbiór wierzchołków i krawędzi grafu  $G$  będziemy oznaczać przez  $V(G)$  i  $E(G)$ . Stopień wierzchołka  $v$  w grafie  $G$  będziemy oznaczać przez  $d_G(v)$ . Graf dwudzielny, którego krawędzie łączą zbiory wierzchołków  $L$  i  $R$ , będziemy oznaczać przez  $G_{LR}$ . Stopień wierzchołka  $v$  w grafie  $G_{LR}$  będziemy oznaczać przez  $d_{G_{LR}}(v)$  lub krócej przez  $d_G(v)$ , o ile nie prowadzi to do nieporozumień.

Przypomnijmy, że dwie krawędzie grafu nazywamy **incydentnymi**, jeśli mają wspólny koniec. Krawędź jest incydentna do wierzchołka  $v$ , jeśli jeden z jej końców to  $v$ . Krawędź jest incydentna do podgrafu, jeśli jest incydentna do pewnego wierzchołka tego podgrafu. Podgraf nazywamy **skojarzeniem**, jeśli żadne dwie krawędzie tego podgrafu nie są incydentne. Wierzchołek nazywamy skojarzonym, jeśli jest końcem pewnej krawędzi należącej do skojarzenia. Skojarzenie nazywamy **maksymalnym** (skrót MM), gdy każdy wierzchołek jest albo skojarzony, albo wszyscy jego sąsiedzi są skojarzeni.

Nasze algorytmy będziemy zapisywać w języku, który można by nazwać "językiem procedur synchronicznych". Procedury synchroniczne mają następującą cechę: jeśli w danym momencie, pewien procesor wykonuje procedurę  $X$ , to wszystkie inne procesory także wykonują procedurę  $X$ . Istnieją procedury synchroniczne elementarne i nieelementarne, przy czym procedura nieelementarna składa się z ciągu wywołań innych procedur synchronicznych (elementarnych i nieelementarnych). Wywołania procedur elementarnych są zazwyczaj zastąpione opisem efektu ich działania. Zauważmy, że wywołanie procedury synchronicznej można traktować jako pewną globalną operację na grafie, mimo że w rzeczywistości procedura ta jest wykonywana przez wszystkie procesory grafu komunikacyjnego.

Przejdziemy teraz do ogólnego omówienia algorytmu opisanego w tym rozdziale. Typowy algorytm znajdujący MM składa się z pętli, w której znajduje się "dobre" skojarzenie, a następnie wykonuje się "modyfikację" grafu, polegającą na usunięciu krawędzi tego skojarzenia wraz ze wszystkimi krawędziami incydentnymi do skojarzenia. Maksymalne skojarzenie to suma wszystkich skojarzeń znajdujących w kolejnych iteracjach pętli. W opisanym w tym rozdziale algorytmie "dobre" skojarzenie to skojarzenie koszące, które definiujemy następująco:

**Definicja 2.1** *Skojarzenie (lub dowolny podgraf) nazywamy **koszącym**, jeśli jest do niego incydentny stały procent wszystkich krawędzi grafu, czyli przynajmniej  $u|E(G)|$  krawędzi gdzie  $u \in (0, 1)$ .*

Przedstawimy teraz formalny zapis procedury SKOJARZENIE, znajdującej skojarzenie maksymalne w dowolnym grafie. Zakładamy, że dysponujemy procedurą KOSZĄCE, znajdującą skojarzenie koszące w dowolnym grafie, w czasie  $O(\log^3 n)$ .

## PROCEDURA SKOJARZENIE

1.  $A := \emptyset$
2. Powtórz  $O(\log n)$  razy:
  - (a) Uruchom procedurę KOSZĄCE, która znajdzie skojarzenie koszące  $M$ .
  - (b) Usuń z grafu krawędzie skojarzenia  $M$ , wraz ze wszystkimi krawędziami incydentnymi do  $M$ .
  - (c)  $A := A \cup M$
3. Zbiór krawędzi  $A$  jest rezultatem działania tej procedury.

**Twierdzenie 2.2** *Procedura SKOJARZENIE znajduje MM w dowolnym grafie, w czasie  $O(\log^4 n)$ .*

**Dowód** Ponieważ procedura KOSZĄCE wymaga  $O(\log^3 n)$  rund, to czas działania procedury SKOJARZENIE możemy oszacować przez  $O(\log^4 n)$ .

Zbiór  $A$  jest oczywiście skojarzeniem w każdej iteracji pętli oraz po jej zakończeniu, dzięki punktowi 2(b) procedury. W każdej iteracji pętli, liczba krawędzi grafu zmniejsza się o stały procent. Tak więc po  $O(\log n)$  iteracjach wszystkie krawędzie grafu powinny zniknąć.

Zauważmy, że jeśli usuniemy krawędzie pewnego skojarzenia, wraz z incydentnymi do tego skojarzenia, i okaże się że po takiej operacji w grafie nie pozostała żadna krawędź, to wynika stąd, że skojarzenie było maksymalne. Gdybyśmy, w punkcie 3 procedury, usunęli z grafu wejściowego krawędzie skojarzenia  $A$  wraz z incydentnymi do  $A$ , to pociągnęłoby to za sobą zniknięcie wszystkich krawędzi, zatem  $A$  jest skojarzeniem maksymalnym.

▽

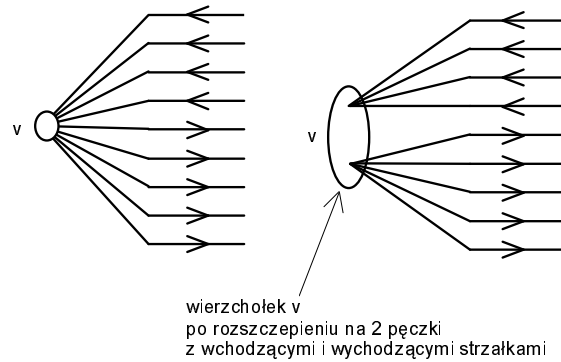
## 2.2 Znajdowanie skojarzenia koszącego w dowolnym grafie.

Pokażemy teraz, jak znaleźć skojarzenie koszące w dowolnym grafie.

Zakładamy, że dysponujemy procedurą KOSZĄCEDWUDZIELNE, znajdującą skojarzenie koszące w grafie dwudzielnym  $G_{LR}$ , w czasie  $O(\log^3 n)$ . Procedura ta wymaga aby wszystkie wierzchołki grafu  $G_{LR}$  "wiedziały" w którym ze zbiorów się znajdują: w zbiorze  $L$ , czy w zbiorze  $R$ .

Będziemy się starali skonstruować w pewien sposób graf dwudzielny i uruchomić w nim procedurę KOSZĄCEDWUDZIELNE. W tym celu wprowadzamy w grafie wejściowym orientację krawędzi, czyli "rysujemy" na krawędziach strzałki, a następnie rozszczepiamy każdy wierzchołek na dwa **pęczki**, z wchodzącymi i wychodzącymi strzałkami. Tę operację, z punktu widzenia pojedynczego wierzchołka, pokazano na rysunku 1 (strona 8).

Graf dwudzielny, jaki otrzymamy w wyniku rozszczepiania, ma po lewej stronie wierzchołki z wychodzącymi strzałkami, a po prawej z wchodzącymi strzałkami. Na rysunku 2 (strona 9) pokazano jak uzyskuje się graf dwudzielny, na przykładzie grafu o pięciu wierzchołkach.



Rysunek 1: Rozszczepianie na 2 pęczki.

W otrzymanym grafie dwudzielnym uruchamiamy procedurę `KOSZĄCEDWUDZIELNE`, która znajduje w nim skojarzenie koszące. Następnie oba pęczki każdego wierzchołka są "sklejane". Niestety, może się zdarzyć, że oba pęczki pewnego wierzchołka były skojarzone, tak więc po sklejeniu nie otrzymamy skojarzenia, a jedynie podgraf z maksymalnym stopniem wynoszącym dwa, czyli cykle i ścieżki. Sklejanie par pęczków i możliwość powstawania wierzchołków ze stopniem dwa, pokazano na rysunku 3 (strona 10).

Dzięki temu, że skojarzenie w grafie dwudzielnym było koszące, cykle i ścieżki także są koszące, czyli jest do nich incydentny stały procent wszystkich krawędzi grafu. W kolejnym kroku musimy zamienić koszące cykle i ścieżki na skojarzenie koszące. Operację tę, pokazaną na rysunku 4 (strona 10), wykonujemy w dwóch etapach:

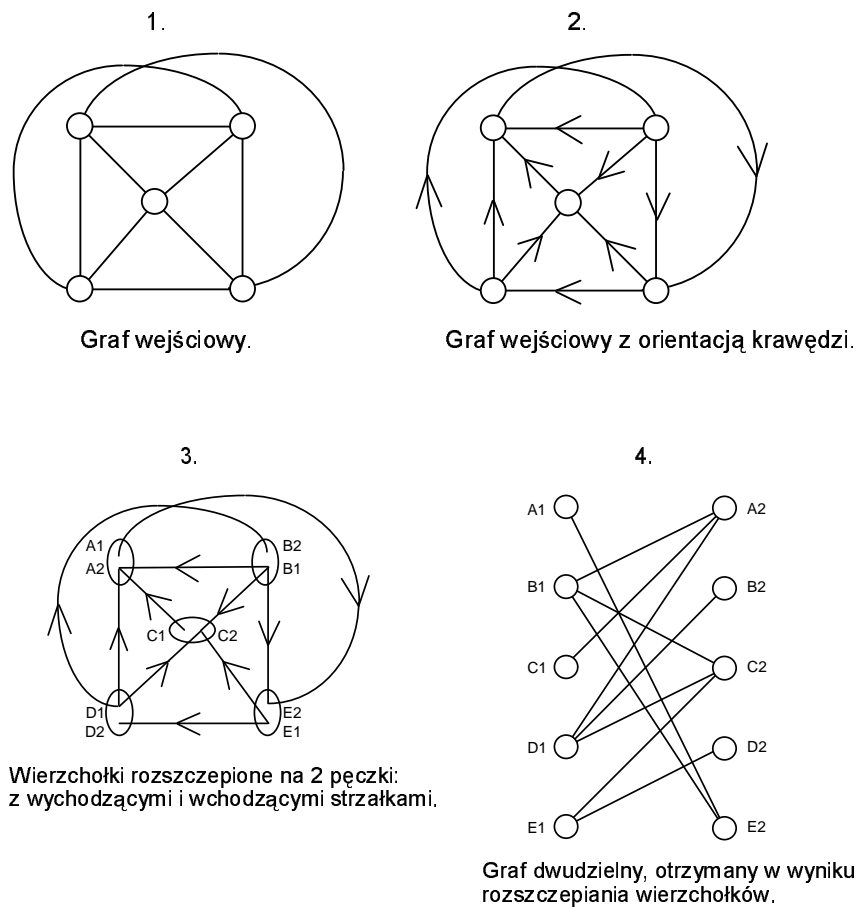
1. W cyklach i ścieżkach (bez końcowych krawędzi) wyznacza się skojarzenie maksymalne, którego krawędzie usuwa się z cykli i ścieżek. Pozostają więc z nich same (izolowane) krawędzie i pary krawędzi.
2. Każda para krawędzi usuwa jedną ze swoich krawędzi, tę z mniejszym stopniem na końcu.

Otrzymujemy w ten sposób skojarzenie, i jak łatwo udowodnić, jest to skojarzenie koszące. Teraz przedstawimy formalny zapis procedury `KOSZĄCE`, znajdującej skojarzenie koszące w dowolnym grafie.

#### PROCEDURA `KOSZĄCE`

1. Zorientuj krawędzie grafu w dowolny sposób.
2. Rozszczep każdy wierzchołek na 2 pęczki: z wchodzącymi i z wychodzącymi strzałkami.  
Otrzymasz graf dwudzielny, w którym po lewej stronie są pęczki z wychodzącymi strzałkami, a po prawej z wchodzącymi.
3. W otrzymanym grafie dwudzielnym uruchom procedurę `KOSZĄCEDWUDZIELNE`, znajdującą skojarzenie koszące.
4. Sklej oba pęczki każdego wierzchołka.





Rysunek 2: Jak się uzyskuje graf dwudzielny.

Otrzymasz podgraf ze stopniem maksymalnym 2, czyli cykle i ścieżki, który oznaczamy przez  $C$ . Niech  $\hat{C}$  oznacza podgraf grafu  $C$  indukowany przez wierzchołki stopnia 2.

5. W  $\hat{C}$  znajdź skojarzenie maksymalne  $X$ , za pomocą procedury Cole'a i Vishkin'a (patrz [CV86]).

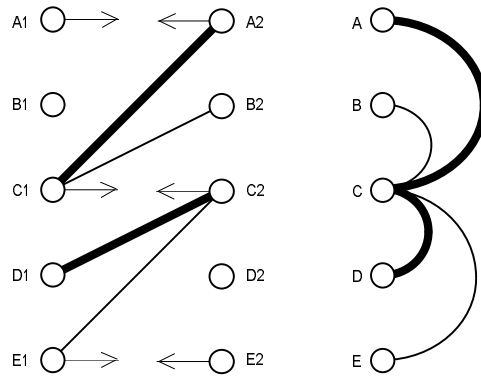
Niech  $C' := C \setminus X$ . Podgraf  $C'$  składa się z (izolowanych) krawędzi i par krawędzi.

6. Dla każdej pary krawędzi  $\{v, u\}, \{u, w\}$  z podgrafu  $C'$ , zaznacz krawędź  $\{v, u\}$  jeśli  $d_G(v) < d_G(w)$ , w przeciwnym wypadku zaznacz  $\{u, w\}$ .

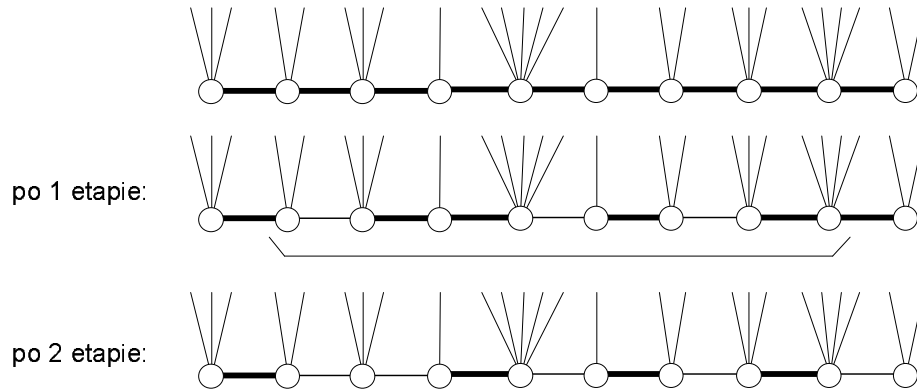
Niech  $C'' := C' \setminus Y$ , gdzie  $Y$  to zbiór zaznaczonych krawędzi.

7. Podgraf  $C''$  jest wynikiem działania tej procedury.

**Twierdzenie 2.3** Procedura KOSZĄCE znajduje skojarzenie koszące w dowolnym grafie, w czasie  $O(\log^3 n)$ .



Rysunek 3: Możliwość powstawania wierzchołków ze stopniem 2.



Rysunek 4: Zamiana koszącej ścieżki na skojarzenie koszące.

**Dowód** W punkcie 5 procedury znajdujemy skojarzenie maksymalne w grafie  $\hat{C}$ . Graf  $\hat{C}$  to zbiór cykli i ścieżek, a więc można znaleźć MM we wszystkich jego składowych, w czasie  $O(\log^* n)$ . Procedura znajdująca MM w cyklu lub w ścieżce, w czasie  $O(\log^* n)$ , została opisana w pracy [CV86]. Czas działania procedury KOSZĄCE jest więc zdeterminowany przez czas działania procedury KOSZĄCEDWUDZIELNE, wywoływanej w punkcie 3, i wynosi  $O(\log^3 n)$ .

Niech  $\text{Incyd}(C)$  oznacza liczbę krawędzi incydentnych do podgrafu  $C$ . Musimy pokazać, że  $C''$  jest skojarzeniem koszącym. Podgraf  $C$  składa się z cykli i ścieżek. Podgraf  $\hat{C}$  zawiera wszystkie składowe grafu  $C$ , z tym że ścieżki są pozbawione końcowych krawędzi. Tak więc, jeśli w składowych  $\hat{C}$  znajdziemy MM i usuniemy ich krawędzie z  $C$ , otrzymując graf  $C'$ , to  $C'$  będzie się składać z krawędzi i par krawędzi, a ponadto  $V(C')$  będzie równe  $V(C)$ . Stąd wynika, że:

$$\text{Incyd}(C') = \text{Incyd}(C)$$

Graf  $C''$  powstaje z grafu  $C'$ , w ten sposób, że każda para krawędzi usuwa jedną ze

swoich krawędzi. Ze sposobu usuwania tych krawędzi wynika, że:

$$\text{Incyd}(C'') \geq \frac{1}{2} \text{Incyd}(C')$$

Tak więc  $C''$  jest skojarzeniem koszącym w grafie wejściowym, gdyż  $C$  było koszące.

▽

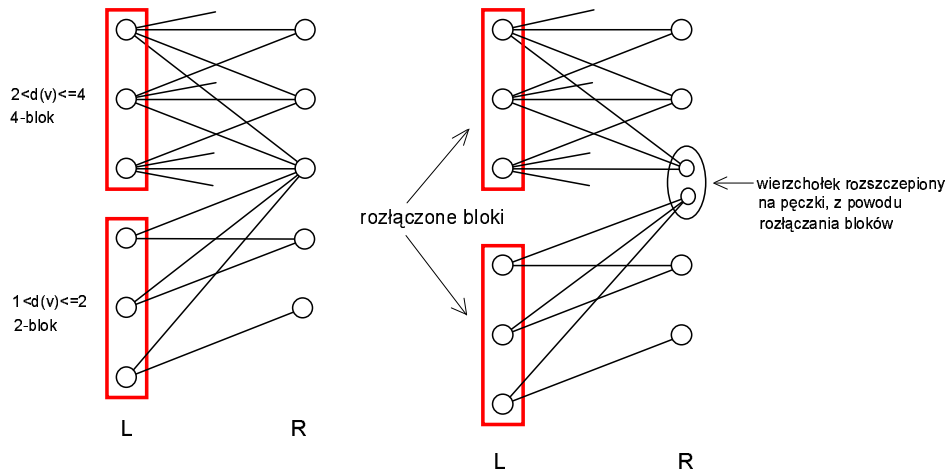
### 2.3 Znajdowanie skojarzenia koszącego w grafie dwudzielnym.

Pokażemy teraz, jak znaleźć skojarzenie koszące w grafie dwudzielnym  $G_{LR}$ , w czasie  $O(\log^3 n)$ . Zbiory  $L$  i  $R$  będziemy niekiedy nazywać lewą i prawą stroną grafu. Zakładamy, że każdy wierzchołek wie, po której stronie grafu się znajduje.

Dzielimy graf dwudzielny na podgrafy, zwane  $D$ -blokami:

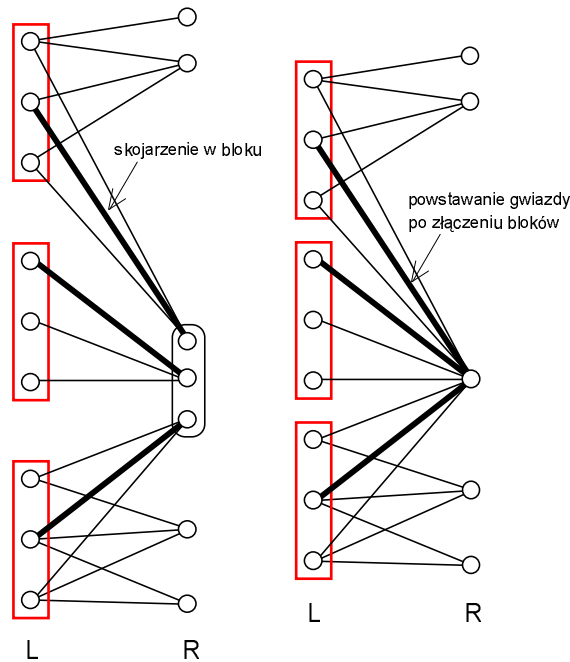
**Definicja 2.4** Niech  $A := \{v \in L : d_G(v) \in (D/2, D]\}$  oraz niech  $B$  będzie zbiorem wszystkich sąsiadów wierzchołków ze zbioru  $A$ . Podgraf  $Q_{AB}$  grafu  $G_{LR}$ , składający się ze wszystkich krawędzi  $G_{LR}$  incydentnych do zbioru  $A$ , nazywamy  $D$ -blokiem.

Zbiór  $D$ -bloków z parametrami:  $D := n, n/2, n/4, \dots, 1$  stanowi podział grafu  $G_{LR}$  na rozłączne krawędziowo podgrafy. Będziemy go nazywać **wektorem bloków**. Dwa bloki mogą mieć wspólne wierzchołki po prawej stronie. Jeśli rozłączymy bloki, tak aby stały się (izolowanymi) składowymi, to wspólne wierzchołki po prawej stronie zostaną rozszczerzone na pęczki, po jednym dla każdego bloku, jak to pokazano na rysunku 5 (strona 11).



Rysunek 5: Rozłączanie bloków.

Zakładamy, że dysponujemy procedurą BLOK, znajdującą skojarzenie koszące w pojedynczym bloku, w czasie  $O(\log^3 n)$ . Aby znaleźć skojarzenie koszące w grafie  $G_{LR}$  rozłączamy bloki, a następnie uruchamiamy w nich procedurę BLOK (we wszystkich blokach równocześnie). Procedura ta znajduje skojarzenia koszące w każdym bloku. Jeśli w każdym bloku mamy skojarzenie, to po złączeniu bloków mogą powstać gwiazdy z wysokim stopniem po prawej stronie, tak jak to pokazano na rysunku 6 (strona 12).



Rysunek 6: Powstawanie gwiazdy po złączeniu bloków.

Otrzymany po złączeniu bloków zbiór gwiazd jest oczywiście koszący. Należy więc zamienić zbiór gwiazd na skojarzenie zachowując własność koszenia. Przypuśćmy że mamy gwiazdę z wierzchołkiem wysokiego stopnia  $v \in R$  i pozostałymi wierzchołkami  $v_1, v_2, \dots, v_k$ ;  $v_i \in L$ . Każdej krawędzi  $vv_i$  przyporządkowujemy wagę  $d_G(v_i)$ . Zauważmy, że waga jednej z krawędzi gwiazdy jest większa od sumy wag pozostałych krawędzi gwiazdy, gdyż należą one do różnych bloków i ich wagi są, w przybliżeniu, różnymi potęgami liczby 2. Oznacza to, że jeśli z każdej gwiazdy pozostawimy na jcięższą krawędź, to otrzymamy skojarzenie koszące, gdyż zbiór gwiazd był koszący.

Przedstawimy teraz formalny zapis procedury KOSZĄCEDWUDZIELNE, znajdującej skojarzenie koszące w grafie dwudzielnym  $G_{LR}$ , w czasie  $O(\log^3 n)$ :

#### PROCEDURA KOSZĄCEDWUDZIELNE

1. Rozłącz bloki wektora bloków.
2. W rozłączonych blokach uruchom (równoległe !) procedurę BLOK, znajdującą skojarzenie koszące w bloku.
3. Złącz bloki wektora bloków. Otrzymasz koszący zbiór gwiazd, utworzony z krawędzi należących do skojarzeń.
4. Dla każdej gwiazdy z wierzchołkami  $v, v_1, \dots, v_k$ , gdzie  $v \in R$ ,  $v_i \in L$ , zaznacz krawędź  $\{v, v_i\}$ , z największym  $d_G(v_i)$ .
5. Zbiór zaznaczonych krawędzi jest wynikiem działania tej procedury.

**Twierdzenie 2.5** *Procedura KOSZĄCEDWUDZIELNE znajduje skojarzenie koszące w grafie dwudzielnym, w czasie  $O(\log^3 n)$ .*

**Dowód** Czas działania procedury KOSZĄCEDWUDZIELNE jest zdominowany przez czas działania procedury BŁOK, wywoływanej w punkcie 2, i wynosi  $O(\log^3 n)$ . Musimy jeszcze pokazać, że krawędzie zaznaczane w punkcie 4 procedury, są skojarzeniem koszącym.

Oznaczmy zbiory  $A$  bloków, wchodzących w skład wektora bloków, przez:  $A_0, A_1, \dots, A_{\log n}$  w taki sposób, że jeśli  $v \in A_i$  to  $d_G(v) \in (n/2^{i+1}, n/2^i]$ . Weźmy pod uwagę pojedynczą gwiazdę ze zbioru gwiazd, jaki powstaje w punkcie trzecim procedury. Gwiazda ta ma wierzchołek z "dużym" stopniem  $v \in R$  oraz wierzchołki:  $v_1, v_2, \dots, v_l$ ;  $v_j \in L$ . Wierzchołki  $v_j$  należą do (różnych) zbiorów  $A_{j_1}, A_{j_2}, \dots, A_{j_l}$ , gdzie  $j_1 < j_2 < \dots < j_l$ . Zauważmy, że:

$$d_G(v_1) \geq \frac{1}{3} \sum_{j=1}^l d_G(v_j)$$

Zaznaczymy krawędź  $vv_1$ . Do zaznaczonej krawędzi będzie incydentna przynajmniej  $\frac{1}{3}$  krawędzi grafu  $G_{LR}$ , które były incydentne do całej gwiazdy. Tak więc, jeśli tę operację wykonamy ze wszystkimi gwiazdami, to zaznaczone krawędzie będą skojarzeniem koszącym, gdyż zbiór gwiazd był koszący.

▽

Pokażemy teraz jak znajduje się skojarzenie koszące w pojedynczym bloku. Okazuje się, że pojedynczy blok stanowi wygodne środowisko do znajdowania skojarzenia koszącego, gdyż jeśli znajdziemy w nim tzw spanner, to bardzo łatwo będzie go można "zamienić" na skojarzenie koszące. Wprowadźmy oznaczenia:  $l(S) := V(S) \cap A$ ,  $r(S) := V(S) \cap B$ , gdzie  $S$  jest podgrafem grafu  $Q_{AB}$ , bez wierzchołków izolowanych.

**Definicja 2.6** *Podgraf  $S$   $D$ -bloku  $Q_{AB}$  nazywamy **spannerem**, jeśli spełnia następujące warunki:*

- $|l(S)| \geq u_0|A|$  gdzie  $u_0$  jest stałą niezależną od  $n$ ,  $u_0 \in (0, 1)$
- $\forall v \in l(S) : 1 \leq d_S(v) \leq c_1$  gdzie  $c_1$  jest stałą
- $\forall v \in r(S) : d_S(v) < c_1 d_Q(v) \frac{1}{D} + 1$

Zakładamy, że dysponujemy procedurą SPANNER, znajdującą spanner w pojedynczym bloku, w czasie  $O(\log^3 n)$ . Zauważmy, że trzeci punkt definicji spannera powoduje, że znajdowanie spannera nie jest zadaniem trywialnym: podgraf wyznaczony w ten sposób, że każdy wierzchołek ze zbioru  $A$  wybiera jedną krawędź prawdopodobnie nie jest spannerem.

Jeśli znaleźliśmy spanner w bloku, to możemy go zamienić na skojarzenie usuwając "nadmiarowe" krawędzie najpierw z lewej, a potem z prawej strony bloku. W Twierdzeniu 2.7 udowodnimy, że skojarzenie to jest koszące. W dowodzie tego twierdzenia pokazuje się, że: "po lewej lub po prawej stronie bloku, suma stopni skojarzonych wierzchołków stanowi stały procent liczby wszystkich krawędzi bloku".

Poniżej przedstawiamy formalny zapis procedury BŁOK, znajdującej skojarzenie koszące w pojedynczym  $D$ -bloku  $Q_{AB}$ :

PROCEDURA BLOK

1. Znajdź w  $D$ -bloku spanner  $S$ , przy pomocy procedury SPANNER.
2. Niech każdy wierzchołek ze zbioru  $A$  zaznaczy jedną (dowolnie wybraną) krawędź, spośród incydentnych do siebie krawędzi podgrafu  $S$ .

Niech  $P$  będzie podgrafem indukowanym przez zaznaczone krawędzie.

3. Niech każdy wierzchołek ze zbioru  $B$  zaznaczy jedną (dowolnie wybraną) krawędź, spośród incydentnych do siebie krawędzi podgrafu  $P$ .

Niech  $M$  będzie podgrafem indukowanym przez zaznaczone krawędzie.

4. Podgraf  $M$  jest wynikiem działania tej procedury.

**Twierdzenie 2.7** *Procedura BLOK znajduje skojarzenie koszące w pojedynczym bloku, w czasie  $O(\log^3 n)$ .*

**Dowód** Czas działania procedury BLOK jest zdominowany przez czas działania procedury SPANNER, wywoływanej w punkcie 1, i wynosi  $O(\log^3 n)$ . Podgraf  $M$ , będący wynikiem działania procedury BLOK, jest oczywiście skojarzeniem. Pozostaje udowodnić, że  $M$  jest skojarzeniem koszącym.

Używamy oznaczeń z procedury BLOK. O podgrafie  $P$  wiemy, że:

- $|l(P)| > u_0|A|$  gdzie  $u_0 \in (0, 1)$
- $\forall v \in l(P) : d_P(v) = 1$
- $\forall v \in r(P) : d_P(v) < c_1 d_Q(v) \frac{1}{D} + 1$

Niech  $m$  oznacza liczbę wszystkich krawędzi bloku, niech  $u_1 \in (0, 1)$ . Rozważmy dwa przypadki:

1.  $\sum_{v \in r(P)} d_Q(v) \geq u_1 m$

Wiemy, że  $r(M) = r(P)$ , tak więc  $M$  jest koszący.

2.  $\sum_{v \in r(P)} d_Q(v) < u_1 m$

Zauważmy, że:

$$|l(M)| = |l(P)| - \sum_{v \in r(P)} (d_P(v) - 1)$$

gdyż każdy wierzchołek z  $r(P)$  usuwa  $d_P(v) - 1$  krawędzi, a ponadto wiemy że  $P$  jest zbiorem gwiazd. Możemy dalej szacować:

$$|l(P)| - \sum_{v \in r(P)} (d_P(v) - 1) \geq u_0|A| - \sum_{v \in r(P)} (c_1 d_Q(v) \frac{1}{D}) \geq$$

$$u_0|A| - \frac{c_1}{D} \sum_{v \in r(P)} d_Q(v) \geq u_0|A| - \frac{c_1}{D} u_1 m \geq$$

$$u_0|A| - \frac{c_1}{D} u_1 D|A| = (u_0 - c_1 u_1)|A| = u_2|A|$$

Stała  $u_1$  musi być dobrana tak, aby  $u_2 := u_0 - c_1 u_1$  było większe od zera.

Tak więc:

$$|l(M)| \geq u_2|A|, \quad u_2 \in (0,1)$$

Wierzchołki po lewej stronie bloku (zbiór  $A$ ) mają stopnie w przedziale  $(D/2, D]$ , zatem  $M$  jest koszący.

▽

## 2.4 Znajdowanie spannera w pojedynczym bloku.

Teraz pokażemy jak znaleźć spanner w pojedynczym  $D$ -bloku  $Q_{AB}$ , w czasie  $O(\log^3 n)$ .

**Definicja 2.8** Niech  $V_{parz}$  i  $V_{nieparz}$  będą zbiorami wierzchołków z parzystymi i nieparzystymi stopniami, w grafie dwudzielnym  $G_{AB}$ . Podgraf  $G'$  grafu  $G_{AB}$  nazywamy **doskonałym splitterem**, jeśli  $\forall v \in V_{nieparz} : d_{G'}(v) = \frac{1}{2}d_G(v) + \frac{1}{2}$  lub  $d_{G'}(v) = \frac{1}{2}d_G(v) - \frac{1}{2}$ , oraz  $\forall v \in V_{parz} : d_{G'}(v) = \frac{1}{2}d_G(v)$ .

Gdybyśmy, w  $(\log D - c)$ -iteracjach, znajdowali doskonały splitter w bloku i usuwali jego krawędzie z bloku, to na końcu otrzymalibyśmy podgraf  $S$ , mający następujące własności:

$$(a) \forall v \in l(S) = A : 1 \leq d_S(v) \leq c_1, \quad (b) \forall v \in r(S) \subseteq B : d_S(v) < c_1 d_Q(v) \frac{1}{D} + 1,$$

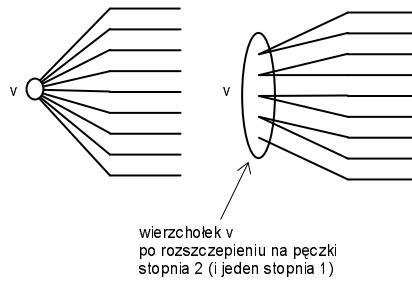
gdzie  $c$  i  $c_1$  są stałymi, który oczywiście jest spannerem.

Autorzy pracy [IS86], w której opisany jest algorytm znajdujący MM na maszynie PRAM, znajdowali doskonały splitter przy pomocy cykli Eulera. Niestety, w modelu rozproszonym znajdowanie cykli Eulera w czasie mniejszym od średnicy grafu jest niemożliwe, jak to pokazano w [Pa]. Zastanowimy się teraz, czy jest możliwe istnienie innej metody znajdowania doskonałego splittera. Doskonały splitter w parzystym cyklu to po prostu kolorowanie krawędzi cyklu dwoma kolorami, na przemian. Można łatwo udowodnić, że znalezienie takiego kolorowania wymaga czasu proporcjonalnego do długości cyklu. Tak więc, w grafie zawierającym składową, będącą długim parzystym cyklem, nie można znaleźć doskonałego splittera, w krótkim czasie. Poniższy fakt dowodzi, że nie jest to możliwe także w grafach bez takiej składowej (dowód tego faktu znajduje się na końcu rozdziału):

**Fakt 2.9** Dla  $r := 4k + 2$  oraz  $n := 4l(r + 1)$ , gdzie  $k$  i  $l$  są dowolnymi liczbami całkowitymi, istnieje graf  $r$ -regularny na  $n$  wierzchołkach, w którym znalezienie doskonałego splittera wymaga czasu  $\Omega(n/r)$ , w rozproszonym, synchronicznym modelu obliczeń.

Tak więc musimy zadowolić się znajdowaniem *przybliżonego* splittera, rozumianego jako graf, w którym "prawie wszystkie" wierzchołki mają stopień "równy w przybliżeniu"  $\frac{1}{2}d_Q(v)$ . Dokładniej, w pierwszej iteracji, przybliżony splitter będzie miał przynajmniej  $(1 - \frac{1}{\log n})|A|$  wierzchołków, ze stopniem w przedziale  $\frac{1}{2}(1 \pm \frac{1}{\log n})d_Q(v)$ . W Twierdzeniu 2.13 udowodnimy, że przybliżony splitter jest wystarczająco silnym narzędziem, aby znaleźć spanner (oczywiście zbiór  $l(S)$  będzie jedynie stałą frakcją  $A$ , podczas gdy używając doskonałego splittera mielibyśmy  $l(S) = A$ ).

Aby znaleźć przybliżony splitter postępujemy następująco: każdy wierzchołek "rozszczepia się" na **pęczki** stopnia 2 oraz, jeśli to konieczne, jeden pęczek stopnia 1. Operację tę pokazano na rysunku 7 (strona 16).



Rysunek 7: Rozszczepianie na pęczki stopnia dwa.

Po rozszczepieniu wszystkich wierzchołków otrzymujemy zbiór cykli i ścieżek, który będziemy nazywać **2-dekompozycją**. Każdy pęczek zachowuje się jakby był osobnym procesorem, podczas gdy w rzeczywistości obliczenia wykonywane przez pęczki pojedynczego wierzchołka są wykonywane przez wierzchołek zawierający te pęczki. Na 2-dekompozycji możemy uruchomić dowolną procedurę wymagającą, aby graf komunikacyjny był cyklem lub ścieżką.

Skojarzenie w ścieżce nazywamy prawie doskonałym (skrót **nPM = near Perfect Matching**), jeśli wszystkie wierzchołki ścieżki, za wyjątkiem co najwyżej końców, są skojarzone. Gdybyśmy potrafili znajdować nPM-y w składowych 2-dekompozycji, to moglibyśmy znaleźć doskonały splitter, gdyż każdy pęczek stopnia 2 miałby dokładnie jedną krawędź należącą do nPM-u. Poszczególne kroki, jakie należałoby wykonać, pokazano na rysunku 8 (strona 17).

Niestety, znajdowanie nPM-ów w składowych 2-dekompozycji wymaga czasu proporcjonalnego do średnicy tych składowych, która może być bardzo duża. Musimy więc postąpić inaczej. Składowe 2-dekompozycji dzielimy na dwie grupy. Składową nazywamy **długą**, jeśli jej długość jest większa lub równa od  $\ell := 100 \log^2 n$ , w przeciwnym wypadku nazywamy ją **krótką**. Spójny podgraf składowej 2-dekompozycji nazywamy **segmentem**.

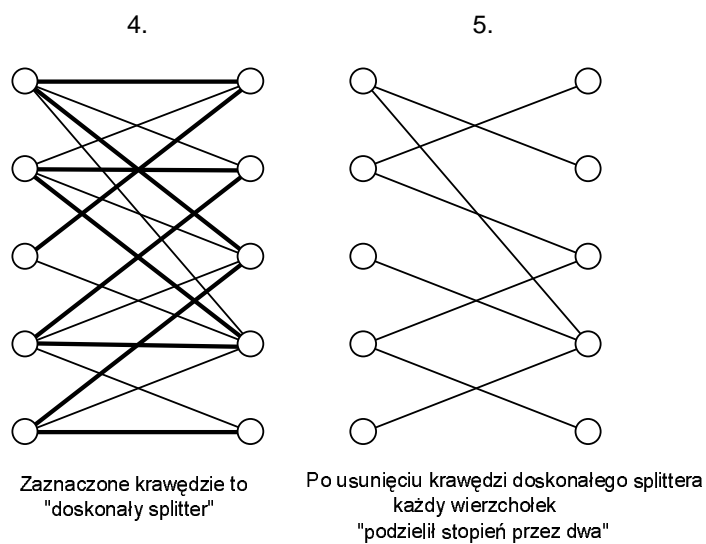
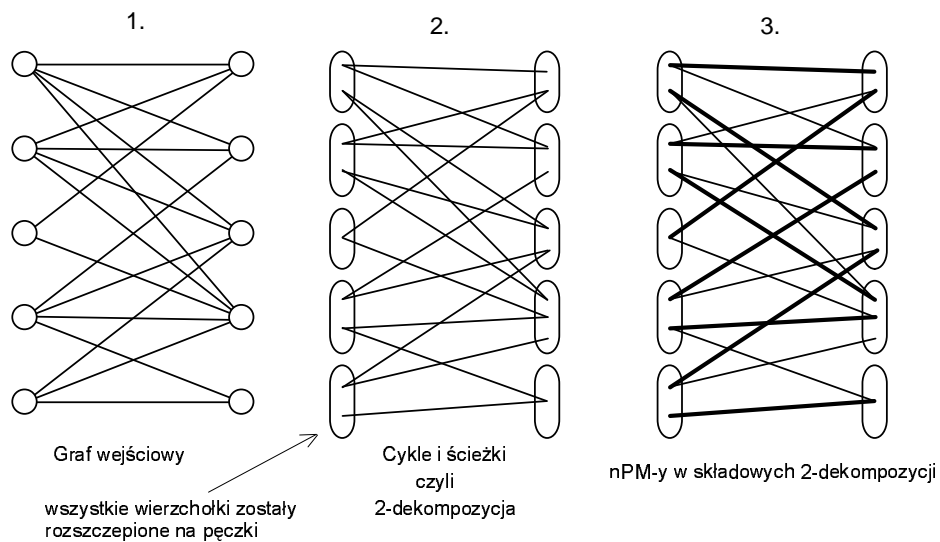
W każdej krótkiej składowej możemy znaleźć nPM, w czasie  $O(\ell)$ . W każdej długiej składowej znajdujemy "nPM-y w segmentach", przy czym każdy segment ma długość większą lub równą od  $\ell$ . Granice segmentów znajdują się na wierzchołkach, co pokazano na rysunku 9 (strona 18).

Okazuje się, że "nPM-y w segmentach, o długości większej lub równej od  $\ell$ " można znaleźć w czasie  $O(\ell)$ , przy pomocy procedury DŁUGIESTRZAŁKI, wykorzystując 2-kolorowanie wierzchołkowe, jakie posiadają wszystkie składowe (to 2-kolorowanie pochodzi od dwudzielnego bloku), co zostanie dokładnie omówione później.

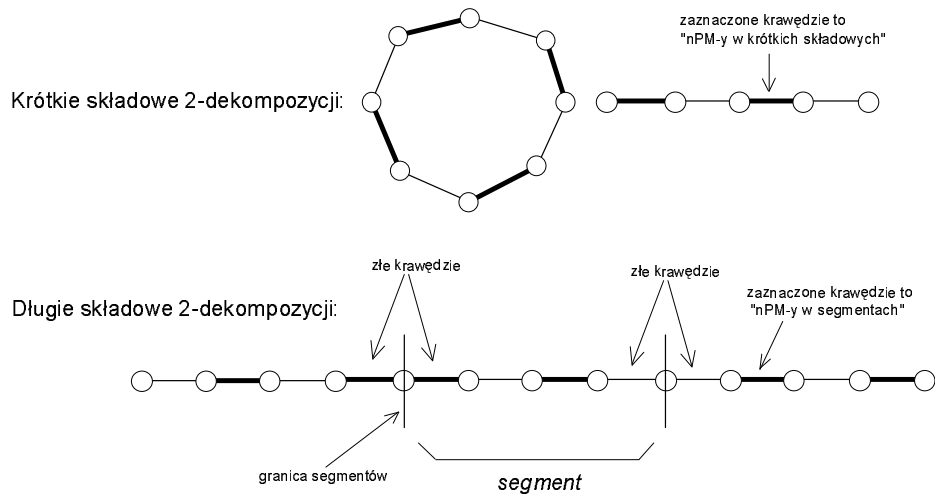
Możemy teraz precyzyjnie zdefiniować **przybliżony splitter** jako podgraf złożony z krawędzi nPM-ów w krótkich składowych oraz z krawędzi "nPM-ów w segmentach" długich składowych 2-dekompozycji.

Oczywiście pęczki graniczne segmentów mogą mieć zero lub dwie krawędzie należące do przybliżonego splittera, dlatego takie pęczki nazywamy **złymi pęczkami**, a ich krawędzie (tj krawędzie incydentne do granic segmentów) nazywamy **złymi krawędziami**. Krawędzie, które nie są złe będziemy nazywać **dobrymi**. Wierzchołki, które posiadają dużo złych pęczków, będą miały zbyt duży lub zbyt mały stopień w przybliżonym split-





Rysunek 8: Powstawanie doskonałego splittera.



Rysunek 9: nPM-y w krótkich i długich składowych.

terze. Takie wierzchołki nazywamy niedopuszczalnymi.

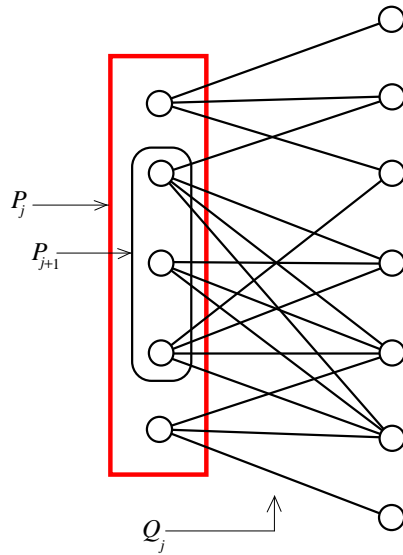
**Definicja 2.10** Wierzchołek  $v \in A$  nazywamy **dopuszczalnym**, jeśli liczba incyden-  
 tnych do niego złych krawędzi jest mniejsza lub równa od  $d_Q(v)/p$ , gdzie  $p := \log n$ . W  
 przeciwnym wypadku wierzchołek nazywamy **niedopuszczalnym**.

Jeśli segmenty są dostatecznie długie, to złych pęczków musi być mało i większość  
 wierzchołków, to wierzchołki dopuszczalne. Jak się okaże, po znalezieniu przybliżonego  
 splittera, prawie wszystkie wierzchołki są dopuszczalne.

Jednym z powodów, dla których blok musi być grafem dwudzielnym jest to, że  
 bez tego założenia podczas rozszczepiania wierzchołków mogłoby powstać bardzo dużo  
 krótkich nieparzystych cykli np trójkątów. Każdy trójkąt musi zawierać jeden zły pęczek.  
 Tak więc mielibyśmy dużo złych pęczków i dużo niedopuszczalnych wierzchołków.

Aby znaleźć spanner będziemy, w  $(\log D - c)$  iteracjach, znajdować przybliżony split-  
 ter i usuwać jego krawędzie oraz złe krawędzie. W kolejnych iteracjach mamy więc do  
 czynienia z coraz mniejszymi podgrafami, z których ostatni jest spannerem. Usuwanie  
 złych krawędzi ma na celu niedopuszczenie do pojawienia się zbyt dużych stopni w tych  
 podgrafach.

Teraz przedstawimy formalny zapis procedury SPANNER, znajdującej spanner w  $D$ -  
 bloku  $Q_{AB}$ , w czasie  $O(\log D \log^2 n)$ . Zakładamy, że dysponujemy procedurą SPLITTER,  
 która zaznacza krawędzie przybliżonego splittera oraz złe krawędzie, w grafie podanym na  
 wejściu, w czasie  $O(\log^2 n)$ . Stosujemy następujące oznaczenia:  $Q_0 := Q_{AB}$ ,  $P_0 := A$ ;  $Q_j$   
 to graf na początku  $j$ -tej iteracji,  $P_j = l(Q_j)$ ;  $P_{j+1}$  to zbiór wierzchołków dopuszczalnych,  
 po uruchomieniu procedury SPLITTER w  $j$ -tej iteracji;  $d_j(v) := d_{Q_j}(v)$ .



Rysunek 10: Pojedynczy blok.

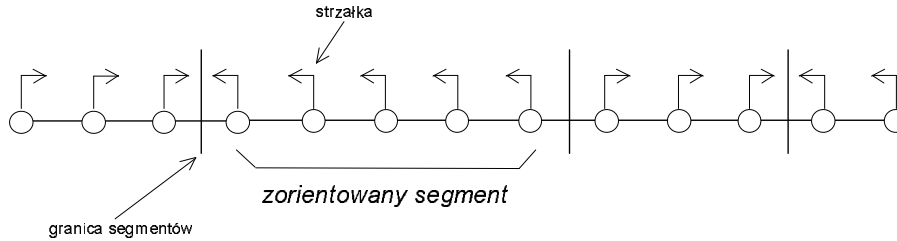
#### PROCEDURA SPANNER

1. Dla  $j := 0, \dots, k - 1$  gdzie  $k = \log D - c$ , wykonaj:
  - (a) Uruchom procedurę SPLITTER, która zaznaczy krawędzie przybliżonego splittera oraz złe krawędzie w grafie  $Q_j$ .
  - (b) Niech do zbioru  $P_{j+1}$  wejdą wierzchołki zbioru  $P_j$ , które są dopuszczalne, tj mają co najwyżej  $d_j(v)/p$  incyduentnych złych krawędzi.
  - (c) Z grafu  $Q_j$  usuń zaznaczone krawędzie (zarówno krawędzie przybliżonego splittera jak i złe krawędzie) oraz usuń wierzchołki ze zbioru  $P_j \setminus P_{j+1}$ . Graf, jaki otrzymamy, oznaczamy przez  $Q_{j+1}$ .
2. Graf  $Q_k$ , traktowany jako podgraf  $Q_0$ , jest wynikiem działania tej procedury.

Ponizej zdefiniowane zostaną pojęcia potrzebne, aby omówić procedurę SPLITTER. Potem zostaną przedstawione Fakty 2.11 i 2.12, analizujące pojedynczą iterację procedury SPANNER (w której jest wywoływana procedura SPLITTER). Następujące potem Twierdzenie 2.13 dowodzi, że procedura SPANNER znajduje podgraf spełniający definicję spannera.

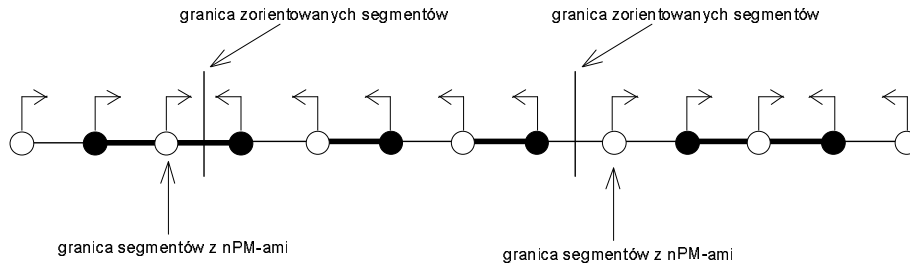
Każdy wierzchołek składowej 2-dekompozycji posiada **strzałkę**, wskazującą na jednego (dowolnego) z dwóch sąsiadów. **Zorientowany segment** to zbiór kolejnych wierzchołków, których strzałki wskazują w tym samym kierunku. W każdej składowej 2-dekompozycji, strzałki wyznaczają podział na zorientowane segmenty, jak to pokazano na rysunku 11 (strona 20).

Zakładamy, że dysponujemy procedurą DŁUGIESTRZAŁKI, znajdującą zorientowane segmenty, z których każdy ma długość większą lub równą od  $\ell$ , w czasie  $O(\ell)$ .



Rysunek 11: Zorientowane segmenty.

Przypomnijmy, że wszystkie składowe 2-dekompozycji bloku posiadają 2-kolorowanie wierzchołkowe, kolorami na przemian czarnym i białym, gdyż blok  $Q_{AB}$  jest grafem dwudzielnym, a kolejne wierzchołki składowej znajdują się raz po lewej, raz po prawej stronie bloku. Tak więc jeśli znajdziemy w składowych zorientowane segmenty o długości większej lub równej od  $\ell$ , to będzie je można łatwo zamienić na "nPM-y w segmentach", o długości większej lub równej od  $\ell$ ". Można to zrobić przyjmując zasadę, że każdy czarny wierzchołek oznacza krawędź, na którą wskazuje jego strzałka, a zaznaczone krawędzie traktujemy jako krawędzie nPM-ów. Granice "segmentów zorientowanych" znajdują się na krawędziach. Dla każdej takiej granicznej krawędzi, jeden z jej końców może się stać granicą "segmentów z nPM-ami", jak to pokazano na rysunku 12 (strona 20).



Rysunek 12: Powstawanie nPM-ów w segmentach.

Obecnie skoncentrujemy się na dokładnym opisie procedury SPLITTER. Procedura ta znajduje (zaznacza) w grafie krawędzie przybliżonego splittera oraz zle krawędzie, w czasie  $O(\log^2 n)$ . Oto formalny zapis procedury SPLITTER:

PROCEDURA SPLITTER

1. Rozszczep każdy wierzchołek grafu na pęczki stopnia 2 (i ewentualnie jeden stopnia 1). Otrzymasz 2-dekompozycję grafu.
2. Uruchom procedurę DŁUGIESTRZAŁKI, która znajdzie "zorientowane segmenty, o długości większej lub równej od  $\ell$ ", gdzie  $\ell := 100 \log^2 n$ , w czasie  $O(\ell)$ , w każdej składowej 2-dekompozycji.

3. W każdej składowej 2-dekompozycji, niech każdy czarny wierzchołek zaznaczy krawędź, na którą wskazuje jego strzałka. Zaznaczone krawędzie tworzą nPM-y lub "segmenty z nPM-ami".
4. Krawędzie nPM-ów zaznacz jako krawędzie "przybliżonego splittera". Krawędzie incydentne do granic "segmentów z nPM-ami" zaznacz jako "złe" krawędzie.
5. Zaznaczone krawędzie to wynik działania tej procedury.

Teraz przechodzimy do analizy działania procedury SPLITTER. W następnych dwóch faktach opiszemy co się dzieje ze stopniami dopuszczalnych wierzchołków, oraz z mocą zbioru takich wierzchołków, po pojedynczym uruchomieniu procedury SPLITTER. W dalszym ciągu będziemy używać oznaczeń z procedury SPANNER, a ponadto niech  $\Delta_j$  i  $\delta_j$  oznaczają maksymalny i minimalny stopień wierzchołków zbioru  $P_j$  w grafie  $Q_j$ . Następny fakt mówi, że stopień dopuszczalnych wierzchołków dzieli się przez 2 prawie dokładnie.

**Fakt 2.11** *Dla każdego  $v \in P_{j+1}$ :*

$$\frac{1}{2} \left( \left( 1 - \frac{2}{\log n} \right) d_j(v) - 1 \right) \leq d_{j+1}(v) \leq \frac{1}{2} (d_j(v) + 1),$$

gdzie  $j$  jest numerem dowolnej iteracji procedury SPANNER. Prawa nierówność jest prawdziwa dla wszystkich  $v \in V(Q_{j+1})$ .

**Dowód** Niech  $e_+$  i  $e_-$  oznaczają liczbę dobrych i złych krawędzi incydentnych do wierzchołka  $v$ . Oczywiście  $d_j(v) = e_+ + e_-$ . Aby ograniczyć  $d_{j+1}(v)$  od góry, zauważmy że najgorszy przypadek ma miejsce gdy  $v$  nie ma incydentnych złych krawędzi. Dlatego gdy  $d_{j+1}(v)$  jest nieparzyste i pęczek stopnia 1 nie ma zaznaczonej krawędzi, to wtedy:

$$d_{j+1}(v) \leq \frac{1}{2}(e_+ + 1) = \frac{1}{2}(d_j(v) + 1)$$

Jest to prawdziwe dla wszystkich  $v \in V(Q_{j+1})$ .

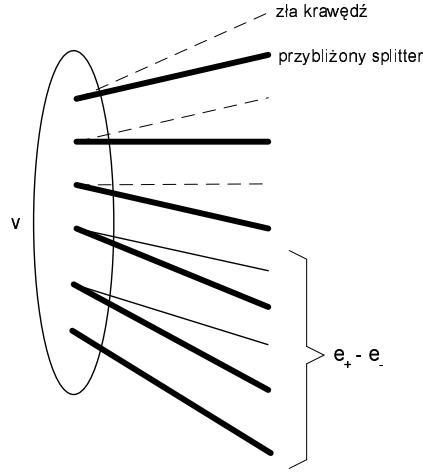
Aby oszacować  $d_{j+1}(v)$  z dołu zauważmy, że dopuszczalny wierzchołek  $v$  traci największą liczbę krawędzi, gdy wszystkie jego pęczki stopnia dwa ze złą krawędzią, mają drugą krawędź należącą do przybliżonego splittera, przez co pęczek traci obie krawędzie. Możemy więc, dla wierzchołka z nieparzystym  $d_j(v)$ , oszacować:

$$d_{j+1}(v) \geq \frac{1}{2}(e_+ - e_- - 1) \geq \frac{1}{2} \left( \left( 1 - \frac{2}{\log n} \right) d_j(v) - 1 \right)$$

ponieważ, z definicji wierzchołka dopuszczalnego,  $e_- \leq d_j(v)/\log n$ . Ideę powyższego szacowania zilustrowano na rysunku 13 (strona 22).

▽

Następny fakt mówi, że po uruchomieniu procedury SPLITTER, w  $j$ -tej iteracji procedury SPANNER, prawie wszystkie wierzchołki ze zbioru  $P_j$  są dopuszczalne, o ile  $\frac{\Delta_j}{\delta_j}$  jest ograniczone przez stałą.



Rysunek 13: Szacowanie  $d_{j+1}(v)$  z dołu;  $e_- = 3$ ,  $e_+ = 8$ ,  $d_{j+1}(v) = 2$ .

**Fakt 2.12** Dla wszystkich iteracji  $j := 0, \dots, k - 1$  procedury SPANNER, spełniony jest następujący warunek:

$$|P_{j+1}| \geq |P_j| \left(1 - \frac{2/100 \Delta_j}{\log n \delta_j}\right)$$

**Dowód** W  $j$ -tej iteracji procedury SPANNER jest uruchamiana procedura SPLITTER, która, między innymi, zaznacza niektóre krawędzie grafu  $Q_j$  jako złe. Przypomnijmy, że  $P_{j+1}$  to zbiór wierzchołków dopuszczalnych. Niech  $N_{j+1}$  będzie zbiorem wierzchołków niedopuszczalnych oraz niech  $be[N_{j+1}]$  będzie liczbą złych krawędzi incydentnych do  $N_{j+1}$ . Zauważmy, że  $|P_{j+1}| = |P_j| - |N_{j+1}|$ .

Dolne ograniczenie dla  $be[N_{j+1}]$  wynika z faktu, że wierzchołki w  $N_{j+1}$  mają, z definicji, przynajmniej  $\delta_j / \log n$  złych krawędzi. Stąd:

$$be[N_{j+1}] \geq |N_{j+1}| \frac{\delta_j}{\log n}$$

Przypomnijmy, że nasz graf jest dwudzielny, tak więc wszystkie cykle (w szczególności krótkie) 2-dekompozycji mają parzystą długość. Dlatego wszystkie złe krawędzie są incydentne do granic segmentów z nPM-ami w długich składowych 2-dekompozycji.

Potrafimy ograniczyć od góry liczbę złych krawędzi incydentnych do  $N_{j+1}$  przez całkowitą liczbę złych krawędzi w  $Q_j$ . Wiemy, że  $|E(Q_j)| \leq \Delta_j |P_j|$  oraz, że każdy segment z nPM-em ma długość przynajmniej  $\ell = 100 \log^2 n$  i zawiera co najwyżej dwie złe krawędzie, zatem możemy oszacować:

$$be[N_{j+1}] \leq \frac{2|P_j|\Delta_j}{100 \log^2 n}$$

z czego, po połączeniu z poprzednim wzorem, wynika teza tego faktu. ▽

Następne twierdzenie stwierdza, że procedura SPANNER znajduje podgraf spełniający definicję spannera:

**Twierdzenie 2.13** *Wywołanie procedury SPANNER, z  $D$ -blokiem  $Q_{AB}$  na wejściu, prowadzi do znalezienia spannera z parametrami  $u_0 = \frac{1}{2}$  i  $c_1 = 16$ , w czasie  $O(\log^3 n)$ .*

**Dowód** Procedura SPANNER składa się z pętli, wykonującej  $O(\log D)$  iteracji, w których najbardziej czasochłonne jest wywołanie procedury SPLITTER, działającej w czasie  $O(\log^2 n)$ . Tak więc czas działania procedury SPANNER można oszacować przez  $O(\log^3 n)$ .

Musimy sprawdzić, czy wynik działania procedury SPANNER, tj podgraf  $Q_k$ , spełnia trzy warunki definicji spannera (używamy wszystkich oznaczeń z procedur SPANNER i SPLITTER):

- $|P_k| > u_0|P_0|$
- $\forall v \in P_k : 1 \leq d_k(v) \leq c_1$
- $\forall v \in r(Q_k) : d_k(v) < c_1 d_0(v) \frac{1}{D} + 1$

Niech  $k := \log D - c$ . Korzystając  $k$ -krotnie z Faktu 2.11 otrzymujemy, dla każdego  $v \in P_k$ :

$$q^k (d_0(v) + 1) - 1 \leq d_k(v) \leq \left(\frac{1}{2}\right)^k (d_0(v) - 1) + 1,$$

gdzie  $q = (1 - 2/\log n)/2$ , przy czym górne ograniczenie jest prawdziwe dla wszystkich  $v \in V(Q_k)$ . Górne ograniczenie w ostatniej nierówności dowodzi spełnienia trzeciego warunku definicji spannera, jeśli przyjmiemy  $c_1 := 2^c$ .

Pamiętając o tym, że  $\forall v \in A : D/2 < d_0(v) \leq D$  otrzymujemy, dla każdego  $v \in P_k$ :

$$q^k \left(\frac{D}{2} + 1\right) - 1 \leq d_k(v) \leq \left(\frac{1}{2}\right)^k (D - 1) + 1,$$

przy czym dla  $c = 4$  i dostatecznie dużych  $n$  mamy:

$$q^k \left(\frac{D}{2} + 1\right) - 1 > 2^{c-1} e^{-2(1+\frac{2}{\log n})} - 1 > 0$$

$$\left(\frac{1}{2}\right)^k (D - 1) + 1 < 2^c + 1 = c_1 + 1 = 17$$

co dowodzi spełnienia drugiego warunku definicji spannera.

Pozostał nam do sprawdzenia pierwszy warunek definicji. Stosując  $k$ -krotnie Fakt 2.12 otrzymujemy:

$$|P_k| \geq |P_0| \prod_{j=0}^{k-1} \left(1 - \frac{2/100 \Delta_j}{\log n \delta_j}\right)$$

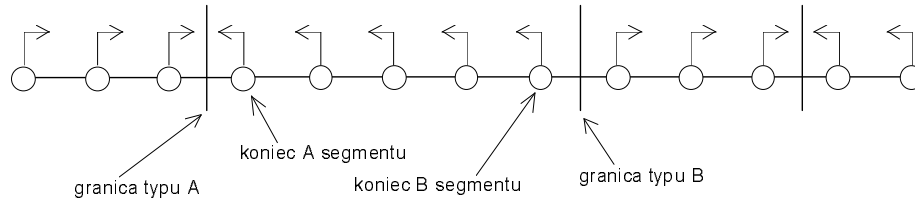
Zauważmy, że:

$$\frac{\Delta_j}{\delta_j} \leq \frac{2^{-j}(D-1)+1}{q^j(D/2+1)-1} \leq 16$$

ponieważ środkowy ułamek jest rosnący jako funkcja zmiennej  $j$ , dla  $j \leq k$ . Dlatego, dla dostatecznie dużych  $n$ , mamy:

$$|P_k| \geq |P_0| \left(1 - \frac{32/100}{\log n}\right)^{\log n} \geq u_0 |P_0|, \quad u_0 = \frac{1}{2}$$

Teraz podamy dokładny opis procedury DŁUGIESTRZAŁKI, znajdującej zorientowane segmenty o długości większej lub równej od  $\ell$ , w czasie  $O(\ell)$ , w dowolnym cyklu lub ścieżce. Zauważmy, że istnieją dwa rodzaje granic między sąsiednimi zorientowanymi segmentami, granice typu A i B, pokazane na rysunku 14 (strona 24).



Rysunek 14: Granice typu A i B między zorientowanymi segmentami.

Oto formalny zapis procedury DŁUGIESTRZAŁKI:

PROCEDURA DŁUGIESTRZAŁKI

1. Na początku wszystkie strzałki wskazują na dowolnego sąsiada.
2. Dla  $i := 1, \dots, \log \ell$  wykonaj:
  - (a) Niech wszystkie granice typu A wykonają:
    - i. Jeśli oba segmenty dotykające granicy mają długość mniejszą od  $2^i$ , wtedy wybierz jeden z nich i odwróć go (tj odwróć wszystkie jego strzałki).
    - ii. Jeśli jeden segment ma długość mniejszą od  $2^i$ , to go odwróć.
  - (b) Jeśli na końcach ścieżki znajdują się segmenty o długości mniejszej od  $2^i$ , to je odwróć.

**Twierdzenie 2.14** *Procedura DŁUGIESTRZAŁKI znajduje zorientowane segmenty o długości większej lub równej od  $\ell$ , w czasie  $O(\ell)$ .*

**Dowód** Można sprawdzić, czy zorientowany segment ma długość mniejszą od  $2^i$  w czasie  $O(2^i)$ , taki segment można odwrócić także w czasie  $O(2^i)$ , tak więc czas działania całej procedury to:

$$\sum_{i=1}^{\log \ell} O(2^i) = O(\ell)$$

Zauważmy, że możemy sprawdzić czy zorientowany segment ma długość mniejszą od  $2^i$  w czasie  $O(2^i)$ , tylko dzięki temu, że pracujemy w modelu synchronicznym. Koniec B segmentu może wysłać komunikat w kierunku końca A, i jeśli komunikat nie dotrze w czasie  $2^i$ , to oznacza to, że segment jest dłuższy niż  $2^i$  (jego dokładnej długości nie znamy, lecz nie jest to nam potrzebne).

Musimy jeszcze udowodnić, że po zakończeniu działania procedury DŁUGIESTRZAŁKI, wszystkie segmenty zorientowane mają długość większą lub równą od  $\ell$ .



Oznaczmy przez  $T(i)$  zdanie: "na końcu  $i$ -tej iteracji wszystkie segmenty mają długość większą lub równą od  $2^i$ ". Zdanie  $T(0)$  jest prawdziwe. Załóżmy, że prawdziwe jest zdanie  $T(i)$ , chcemy udowodnić zdanie  $T(i+1)$ . Na początku iteracji  $(i+1)$  istnieją tylko "długie segmenty", o długości większej lub równej od  $2^{i+1}$  oraz "krótkie segmenty" o długości większej lub równej od  $2^i$  i mniejszej od  $2^{i+1}$ . Musimy udowodnić, że na końcu tej iteracji wszystkie segmenty są długie. Każdy krótki segment może znaleźć się w następujących dwóch sytuacjach:

1. dzieli granicę typu A z długim segmentem lub z krótkim segmentem; w obu przypadkach nasz krótki segment zniknie
2. jego koniec A jest na końcu ścieżki; w takim wypadku krótki segment także zniknie, dzięki specjalnemu traktowaniu końcówek ścieżek

Tak więc prawdziwe jest zdanie  $T(\log \ell)$ , stwierdzające że po zakończeniu działania procedury DŁUGIESTRZAŁKI, wszystkie zorientowane segmenty mają długość większą lub równą od  $\ell$ .

▽

Na zakończenie tego rozdziału udowodnimy fakt, przedstawiony wcześniej, pokazujący istnienie grafów regularnych o dużej średnicy, w których znalezienie doskonałego splittera nie jest możliwe w krótkim czasie:

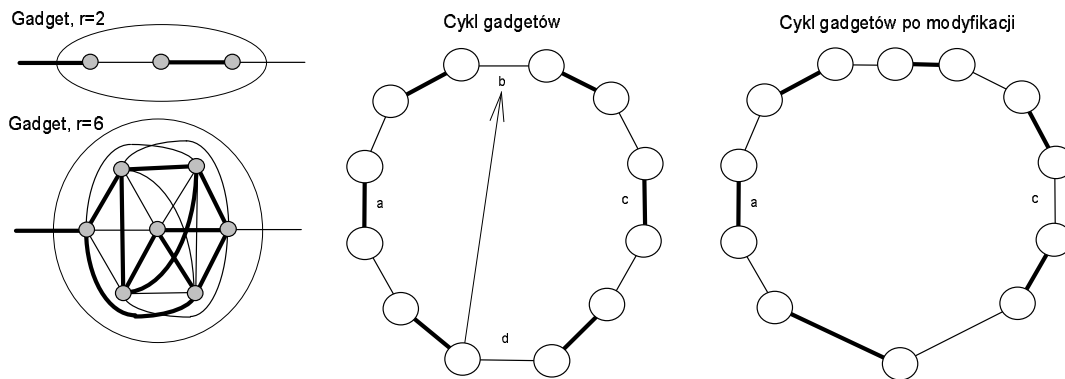
**Fakt 2.9** *Dla  $r := 4k + 2$  oraz  $n := 4l(r + 1)$ , gdzie  $k$  i  $l$  są dowolnymi liczbami całkowitymi, istnieje graf  $r$ -regularny na  $n$  wierzchołkach, w którym znalezienie doskonałego splittera wymaga czasu  $\Omega(n/r)$ , w rozproszonym, synchronicznym modelu obliczeń.*

**Dowód** Niech  $G$  będzie grafem, którego wierzchołki mają parzyste stopnie i który ma parzystą liczbę krawędzi. Przypomnijmy, że doskonałym splitterem w grafie  $G$  nazywamy podgraf  $G'$ , taki że  $\forall v : d_{G'}(v) = \frac{1}{2}d_G(v)$ .

Dla  $r := 4k + 2$ , gdzie  $k$  jest całkowite, zbudujemy  $r$ -regularny graf komunikacyjny, o dużej średnicy, dla którego będzie można udowodnić, że znalezienie doskonałego splittera wymaga czasu rzędu średnicy grafu. Graf ten to cykl zbudowany z "gadżetów", które są małymi grafami  $r$ -regularnymi, z dwoma wychodzącymi krawędziami (zwanymi "końcówkami" gadżetu). Dla wygody, krawędzie doskonałego splittera  $G'$  będziemy nazywać "czarnymi", a pozostałe "białymi". Gadżety muszą być tak skonstruowane, aby po znalezieniu dowolnego doskonałego splittera w cyklu gadżetów, każdy gadżet miał obie końcówki innego koloru. Oznacza to, że krawędzie łączące gadżety muszą być, na przemian, czarne i białe.

Gadżet powstaje przez rozcięcie jednej krawędzi grafu  $K_{r+1}$ , czyli grafu pełnego na  $(r+1)$  wierzchołkach. Obie części rozciętej krawędzi stają się końcówkami gadżetu. Niech  $u$  oznacza liczbę pomalowanych na czarno końcówek gadżetu,  $u$  może przyjmować wartości 0, 1, 2. Oczywiście suma  $\sum_v d_{G'}(v) + u$ , gdzie sumowanie przebiega po wierzchołkach pojedynczego gadżetu, jest liczbą parzystą, gdyż każda czarna krawędź jest liczona dwukrotnie. Tak więc  $(r+1)\frac{r}{2} + u$  jest parzyste, co dla  $r = 4k + 2$  oznacza, że  $8k^2 + 10k + 3 + u$  jest parzyste, czyli  $u = 1$ , a więc końcówki gadżetu muszą mieć różny kolor.

Zauważmy, że średnica "cyklu gadżetów" wynosi  $\Omega(n/r)$ . Przypuśćmy, że pewien algorytm, działający w czasie mniejszym od  $\frac{1}{5}$  średnicy cyklu gadżetów, znalazł doskonały



Rysunek 15: Gadget i "cykl gadgetów".

splitter. Dwa gadgety, mające wspólną końcówkę  $x$ , będziemy nazywać "gadgetami krawędzi  $x$ ". Oznaczmy cztery krawędzie między gadgetami, leżące w równych odległościach, przez  $a, b, c, d$ . Krawędzie  $a$  i  $c$  leżą naprzeciwko siebie i mają równe kolory. Zmodyfikujemy teraz graf komunikacyjny, przenosząc jeden z gadgetów krawędzi  $d$ , na krawędź  $b$ . Po takiej modyfikacji krawędzie  $a$  i  $c$  powinny mieć różne kolory. Jednak ani  $n$  ani kule jakie "widzą" krawędzie  $a$  i  $c$  nie zmieniły się, zatem ich kolory także się nie zmieniły i nadal są równe.

Dowodzi to, że nie jest możliwe znalezienie doskonałego splittera w cyklu gadgetów, w czasie mniejszym od  $\frac{1}{5}$  jego średnicy.

▽

### 3 Algorytm znajdujący maksymalne $f$ -skojarzenie.

#### 3.1 Definicje.

Zakładamy, że wierzchołki grafu mają przyporządkowane całkowite pojemności  $f(v) \geq 1$ . Wtedy podgraf  $M$  nazywamy  $f$ -skojarzeniem, gdy  $\forall v : d_M(v) \leq f(v)$ . Wierzchołek  $v$  nazywamy **nasyconym**, gdy  $d_M(v) = f(v)$ .  $F$ -skojarzenie nazywamy **maksymalnym** (w skrócie **MfM**), gdy każdy wierzchołek grafu jest albo nasycony, albo wszyscy jego sąsiedzi są nasycony. Jeżeli w grafie  $G$  dany jest podgraf  $M$ , to **modyfikację** grafu  $G$  względem podgrafu  $M$  określamy jako ciąg następujących trzech operacji:

1. Usuń z grafu  $G$  krawędzie należące do podgrafu  $M$ .
2. (Aktualizuj pojemności)  
Dla każdego wierzchołka  $v$ : niech  $f(v) := f(v) - d_M(v)$ .
3. (Usuń nasycone wierzchołki)  
Dla każdego wierzchołka  $v$ : gdy  $f(v) \leq 0$  to usuń wierzchołek  $v$ .

Przyjmujemy, że "modyfikacja względem pewnego  $f$ -skojarzenia" jest jedyną operacją jaką wolno wykonywać na grafie wejściowym. W punkcie trzecim mamy warunek  $f(v) \leq 0$  zamiast  $f(v) = 0$ , aby można było mówić o modyfikacji względem dowolnego podgrafu, a nie tylko względem  $f$ -skojarzenia.

Algorytm znajdujący MfM, opisany w tym rozdziale, składa się najogólniej mówiąc z głównej pętli, w której:

1. znajduje się "dobre"  $f$ -skojarzenie
2. wykonuje się względem niego modyfikację
3. dodaje się to  $f$ -skojarzenie do tymczasowego  $f$ -skojarzenia wynikowego

Główna pętla musi wykonać tyle iteracji, aby wszystkie krawędzie grafu zostały usunięte. Sumowanie  $f$ -skojarzeń pojawiających się w iteracjach głównej pętli nazywamy ich **akumulowaniem**. Wiemy, że zakumulowane  $f$ -skojarzenie wynikowe jest maksymalne, gdyż gdybyśmy wykonali modyfikację grafu wejściowego względem tego  $f$ -skojarzenia, to wszystkie krawędzie grafu wejściowego zostałyby usunięte.

$F$ -skojarzenie znajdowane w głównej pętli, powinno być "dobre" w tym sensie, że powinno gwarantować zniknięcie wszystkich krawędzi, po niewielkiej liczbie iteracji. Dobre  $f$ -skojarzenie ma zazwyczaj własność, że po modyfikacji względem niego pewna funkcja zmniejsza wartość o stały procent (tą funkcją mogła by być np liczba krawędzi). Taka funkcja musi mieć następujące cechy:

1. jest wielomianem zmiennej  $n$ , gdzie  $n$  jest liczbą wierzchołków
2. jeśli wartość funkcji jest równa zero, to obliczenia zostały skończone

W opisanym w tym rozdziale algorytmie MfM, "dobre"  $f$ -skojarzenie to  $f$ -skojarzenie kosztujące, zdefiniowane następująco:

**Definicja 3.1** *f-skojarzenie (lub dowolny podgraf) nazywamy **koszącym**, jeśli po modyfikacji grafu względem tego f-skojarzenia (podgrafu), funkcja wagowa*

$$\Phi = \sum_v f(v)d(v)$$

*zmniejsza wartość o stały procent.*

Mówimy, że podgraf "jest koszący na zbiorze  $Z$ ", jeśli po modyfikacji grafu funkcja  $\Phi^Z = \sum_{v \in Z} f(v)d(v)$  zmniejsza wartość o stały procent. Jeśli w grafie dwudzielnym  $G_{LR}$  dysponujemy podgrafem koszącym na zbiorze  $L$ , to będziemy go nazywać podgrafem koszącym lewostronnie. Często znajdujemy pewien podgraf koszący (który nie jest  $f$ -skojarzeniem) a następnie usiłujemy zamienić go na  $f$ -skojarzenie, zachowując własność koszenia. Dlatego pojęcia modyfikacji i koszenia są zdefiniowane dla dowolnych podgrafów, a nie tylko dla  $f$ -skojarzeń.

Jeżeli wierzchołek ma stopień w podgrafie większy od swojej pojemności, to mówimy, że na tym wierzchołku jest **nadmiar**. Podgraf, w którym stopnie wierzchołków co najwyżej dwukrotnie przekraczają pojemności nazywamy  **$f$ -cyklem**. W algorytmie MM, z poprzedniego rozdziału, jako szkodliwy efekt uzyskiwania dwudzielnosci pojawiały się koszące cykle i ścieżki, które należało zamienić na skojarzenie koszące. W algorytmie MfM pojawia się  $f$ -cykl koszący, który należy zamienić na  $f$ -skojarzenie koszące. Niestety, operacja ta jest znacznie bardziej skomplikowana niż w przypadku zwykłych skojarzeń choćby dlatego, że jeśli pojemności są większe od 1 to  $f$ -cykl w niczym nie przypomina "zwykłego" cyklu.

Wprowadzamy następujące oznaczenia: symbolem  $W[G]$  będziemy oznaczać wartość wyrażenia  $W$ , zależącą od grafu  $G$ . Symbolem  $W[G//Q]$  będziemy oznaczać wartość wyrażenia  $W$ , obliczoną po modyfikacji grafu  $G$  względem podgrafu  $Q$ . Przykład:  $f(v)d(v)[G//Q]$  to wartość iloczynu pojemności i stopnia wierzchołka  $v$ , obliczona po modyfikacji grafu  $G$  względem podgrafu  $Q$ . W przypadku prostych wyrażeń stosujemy nieco uproszczony zapis:  $d_G(v) := d(v)[G]$ ,  $d_{G//Q}(v) := d(v)[G//Q]$ . Używając tych oznaczeń możemy teraz zapisać definicję podgrafu koszącego następująco:

$$\text{"Podgraf } Q \text{ jest koszący"} \Leftrightarrow \Phi[G//Q] \leq (1 - u)\Phi[G]$$

gdzie  $u \in (0, 1)$  i nie zależy od  $n$ .

**Uwaga:** dla uproszczenia obliczeń wprowadzamy następującą zasadę: jeśli w wyniku modyfikacji grafu względem podgrafu  $Q$  wierzchołek  $v$  zostaje usunięty, to wierzchołek ten może nadal figurować w obliczeniach, lecz przyjmujemy że:

$$d(v)[G//Q] := 0, \quad f(v)[G//Q] := f(v)[G] - d_Q(v)$$

Zauważmy, że przyjęcie takiej zasady nie ma wpływu na wartość funkcji  $\Phi[G//Q]$ , czy też jakiegokolwiek innej "sumy po wierzchołkach", w której występuje czynnik  $d(v)[G//Q]$ .

Jeśli w opisie modyfikacji warunek  $f(v) \leq 0$  zastąpić warunkiem  $f(v) \leq p - 1$ , to taką operację nazywamy  **$p$ -modyfikacją**. Po  $p$ -modyfikacji wszystkie wierzchołki spełniają warunek  $f(v) \geq p$ .

**Definicja 3.2** *Podgraf nazywamy  **$p$ -koszącym**, jeśli po  $p$ -modyfikacji względem tego podgrafu, wartość funkcji  $\Phi$  zmniejsza się o stały procent.*

Zauważmy, że podgraf  $p$ -koszący, dla  $p > 1$ , nie musi być koszący. Aby znaleźć  $f$ -skojarzenie koszące (lewostronne), w grafie dwudzielnym  $G_{LR}$ , w czasie  $O(\log^3 n)$ , potrzebne jest, jak się wkrótce okaże, założenie  $\forall v \in R : f(v) \geq p$ . Jeśli chcemy znaleźć  $f$ -skojarzenie koszące, wykonać względem niego modyfikację i znów znaleźć  $f$ -skojarzenie koszące, to zamiast modyfikacji konieczne będzie wykonanie  $p$ -modyfikacji, gdyż zapewnia ona zachowanie warunku  $f(v) \geq p$ . Oczywiście zakumulowane  $f$ -skojarzenie będzie wtedy  $p$ -koszące. Symbolem  $W[G/p/Q]$  oznaczamy wartość wyrażenia  $W$ , obliczoną po  $p$ -modyfikacji grafu  $G$  względem podgrafu  $Q$ .

Podczas zamiany  $f$ -cyklu koszącego na  $f$ -skojarzenie  $p$ -koszące (czym zajmuje się procedura ZAMIANA) przydatne jest pojęcie **częściowo maksymalnego  $f$ -skojarzenia** (skrót **PMfM**) o następującej definicji:

**Definicja 3.3**  $f$ -skojarzenie  $Q$  nazywamy częściowo maksymalnym jeśli

$$\forall v : i(v) \geq 2(f(v) - d_Q(v)) - s(v) \quad (1)$$

gdzie  $i(v)$  oznacza liczbę nasyconych sąsiadów wierzchołka  $v$ ,  
 $s(v) := \max\{0, 2f(v) - d(v)\}$ .

Nazwa PMfM ma następującą motywację: w "normalnym" MfM wszyscy sąsiedzi nienasyconego wierzchołka muszą być nasycony, natomiast w PMfM osłabiono to wymaganie do jedynie  $2(f(v) - d_Q(v)) - s(v)$  sąsiadów, a więc jest ono spełnione "częściowo". PMfM jest znacznie łatwiejsze do znalezienia niż MfM. W grafie dwudzielnym może ono być znalezione w stałym czasie, przy pomocy procedury CZĘŚCIOWEDWUDZIELNE. Jeśli warunek (1) jest spełniony tylko na zbiorze  $X$ , to mówimy że  $Q$  jest PMfM-em ograniczonym do zbioru  $X$ .

### 3.2 Ogólny opis algorytmu.

Aby znaleźć MfM w dowolnym grafie, należy uruchomić w nim procedurę FSKOJARZENIE. Składa się ona z pętli, w której znajduje się  $f$ -skojarzenie, takie że po modyfikacji względem niego, górne ograniczenie pojemności wszystkich wierzchołków (zmienna  $q$ ) "logarytmuje się".

PROCEDURA FSKOJARZENIE

1. Niech  $q := n$ ,  $S := \emptyset$ . Powtórz  $O(\log^* n)$  razy:
  - (a) Uruchom procedurę ZLOGARYTMUJF, znajdującą  $f$ -skojarzenie  $X$ .
  - (b) Wykonaj modyfikację grafu względem  $X$ .
  - (c)  $q := \log^2 q$ ,  $S := S \cup X$
2. Powtórz  $O(1)$  razy:
  - (a) Znajdź "zwykłe" maksymalne skojarzenie  $X$
  - (b) Wykonaj modyfikację grafu względem  $X$ .
  - (c)  $S := S \cup X$
3. Podgraf  $S$  jest wynikiem działania tej procedury

**Twierdzenie 3.4** *Procedura FSKOJARZENIE znajduje MfM, w dowolnym grafie, w czasie  $O(\log^4 n \log^* n)$ .*

**Dowód** W każdej iteracji pętli z punktu 1, przed wywołaniem procedury ZLOGARYTMUJF spełniony jest warunek  $\forall v : f(v) < q$ . Procedura ZLOGARYTMUJF znajduje takie  $f$ -skojarzenie, że po modyfikacji względem niego spełniony jest warunek  $\forall v : f(v) < \log^2 q$ . Zauważmy, że  $16 = \log^2 16$ , tak więc po  $O(\log^* n)$  iteracjach pętli pozostaną jedynie wierzchołki z  $f(v) \leq 16$ .

Każde skojarzenie maksymalne, znajdowane w iteracjach pętli z punktu 2, jest równocześnie  $f$ -skojarzeniem. Skoro jest to skojarzenie maksymalne, to każda krawędź ma przynajmniej jeden koniec skojarzony. Tak więc, po modyfikacji, przynajmniej jeden z końców każdej krawędzi zmniejsza pojemność o 1. Po 32 iteracjach, każda krawędź musi mieć przynajmniej jeden koniec z zerową pojemnością, co oznacza że krawędź ta zostanie usunięta w trakcie modyfikacji.

Procedura ZLOGARYTMUJF, podobnie jak i procedura znajdująca MM, opisana w poprzednim rozdziale, działają w czasie  $O(\log^4 n)$ , tak więc procedura FSKOJARZENIE działa w czasie  $O(\log^4 n \log^* n)$ .

▽

W procedurze ZLOGARYTMUJF dzielimy zbiór wierzchołków na wierzchołki z małymi pojemnościami (zbiór  $A$ ) i dużymi pojemnościami (zbiór  $B$ ). Najpierw obliczamy  $f$ -skojarzenia i wykonujemy modyfikacje w podgrafie indukowanym przez  $B$ , tak długo aż wszystkie krawędzie tego podgrafu znikną. Następnie robimy to samo w podgrafie dwudzielnym między zbiorami  $A$  i  $B$ , znów tak długo aż wszystkie krawędzie tego podgrafu znikną. Podział na zbiory  $A$  i  $B$  wprowadzamy z tego powodu, że procedury KOSZĄCE i KOSZĄCEDWUDZIELNE, używane do znajdowania  $f$ -skojarzeń, wymagają aby pojemności wierzchołków zbioru  $B$  były większe lub równe od  $p := \log^2 q$ , gdzie  $q$  jest górnym ograniczeniem pojemności.

#### PROCEDURA ZLOGARYTMUJF

1. Niech  $p := \log^2 q$ ,  $A := \{v : f(v) < p\}$ ,  $B := \{v : f(v) \geq p\}$ ,  $S := \emptyset$ .

2. (Na tym etapie likwidujemy krawędzie łączące wierzchołki zbioru  $B$ .)

Powtórz  $O(\log n)$  razy:

(a) Uruchom procedurę KOSZĄCE, w podgrafie  $G[B]$ , tj indukowanym przez wierzchołki zbioru  $B$ , z pojemnościami  $f_{G[B]}(v) := f_G(v)$ .

Procedura KOSZĄCE znajduje  $f$ -skojarzenie  $p$ -koszące, w podgrafie  $G[B]$ , które oznaczamy przez  $M$ .

(b) Wykonaj modyfikację grafu względem  $M$  i uaktualnij zbiory  $A$ ,  $B$ .

(c)  $S := S \cup M$ .

3. (Na tym etapie likwidujemy krawędzie łączące wierzchołki zbiorów  $A$  i  $B$ .)

Powtórz  $O(\log n)$  razy:

(a) Uruchom procedurę KOSZĄCEDWUDZIELNE, w podgrafie dwudzielnym  $G_{AB}$ , tj składającym się z krawędzi łączących zbiory  $A$  i  $B$ , z pojemnościami  $f_{G_{AB}}(v) := f_G(v)$ .

Procedura KOSZĄCEDWUDZIELNE znajduje  $f$ -skojarzenie koszące na zbiorze  $A$ , w podgrafie  $G_{AB}$ , które oznaczamy przez  $M$ .

- (b) Wykonaj modyfikację grafu względem  $M$  i uaktualnij zbiory  $A, B$ .
- (c)  $S := S \cup M$ .

4. Podgraf  $S$  jest wynikiem działania tej procedury.

**Twierdzenie 3.5** *Niech graf wejściowy spełnia warunek:*

$$\forall v : f(v) < q \tag{2}$$

Procedura ZLOGARYTMUJF, w czasie  $O(\log^4 n)$ , znajduje  $f$ -skojarzenie  $S$  takie, że po modyfikacji grafu względem  $S$  będzie spełniony warunek:

$$\forall v : f(v) < p := \log^2 q \tag{3}$$

**Dowód** W wyniku uruchomienia procedury KOSZĄCE, w pętli z punktu 2, otrzymujemy  $f$ -skojarzenie  $p$ -koszące. Tak więc, aby suma (4) zmniejszyła się o stały procent nie wystarczy sama modyfikacja, lecz potrzebna jest  $p$ -modyfikacja. W punkcie 2(b), zaraz po modyfikacji uaktualnia się zbiory  $A$  i  $B$ , a "modyfikacja" plus "uaktualnienie zbiorów" to to samo co " $p$ -modyfikacja". Tak więc po każdej iteracji pętli z punktu 2, następująca funkcja:

$$\sum_{v \in B} f(v) d_B(v) \tag{4}$$

zmniejsza wartość o stały procent. Po  $O(\log n)$  iteracjach, wszystkie krawędzie łączące wierzchołki zbioru  $B$  powinny więc zniknąć. W powyższym wzorze występuje  $d_B(v)$  zamiast  $d_G(v)$ , gdyż dysponujemy  $f$ -skojarzeniem  $p$ -koszącym w  $G[B]$ , a nie w grafie  $G$  na zbiorze  $B$ . Zauważmy, że założenia jakich wymaga procedura KOSZĄCE, czyli  $\forall v : p \leq f(v) < q$ , są zachowane po każdej iteracji pętli, dzięki uaktualnianiu zbiorów  $A$  i  $B$ .

Po każdej iteracji pętli z punktu 3, następująca funkcja:

$$\sum_{v \in A} f(v) d_B(v) \tag{5}$$

zmniejsza wartość o stały procent. Po  $O(\log n)$  iteracjach, wszystkie krawędzie łączące zbiory  $A$  i  $B$  powinny zniknąć. Dwie końcowe uwagi poprzedniego paragrafu stosują się także tutaj.

Po zakończeniu obu pętli pozostały jedynie wierzchołki zbioru  $A$ , a więc spełniające warunek (3). Czas działania procedur KOSZĄCE i KOSZĄCEDWUDZIELNE wynosi  $O(\log^3 n)$ , tak więc czas działania procedury ZLOGARYTMUJF wynosi  $O(\log^4 n)$ .

▽

### 3.3 Znajdowanie $f$ -skojarzenia $p$ -koszącego w grafie.

Omówimy teraz procedurę KOSZĄCE, znajdującą  $f$ -skojarzenie  $p$ -koszące w dowolnym grafie wejściowym (dowolnym w tym sensie, że nie musi być dwudzielny). Procedura ta składa się z 2 etapów: najpierw znajduje się  $f$ -cykl koszący, czyli podgraf koszący, w którym stopnie wierzchołków przekraczają pojemności co najwyżej dwukrotnie, a następnie zamienia się  $f$ -cykl na  $f$ -skojarzenie, zachowując (do pewnego stopnia) własność koszenia.

#### PROCEDURA KOSZĄCE

1. Uruchom procedurę CYKL, która znajduje  $f$ -cykl koszący  $C$ .
2. Uruchom procedurę ZAMIANA, w celu zamiany  $f$ -cyklu  $C$  na  $f$ -skojarzenie  $p$ -koszące, które oznaczamy przez  $M$ .
3.  $M$  jest rezultatem działania tej procedury.

**Twierdzenie 3.6** *Niech graf wejściowy spełnia warunek:*

$$\forall v : p \leq f(v) < q \quad (6)$$

*Procedura KOSZĄCE znajduje w grafie wejściowym  $f$ -skojarzenie  $p$ -koszące, w czasie  $O(\log^3 n)$ .*

**Dowód** Warunek (6) jest wymagany przez procedurę CYKL. Procedury CYKL i ZAMIANA działają w czasie  $O(\log^3 n)$ , tak więc procedura KOSZĄCE działa także w czasie  $O(\log^3 n)$ .

▽

Celem procedury CYKL jest znalezienie  $f$ -cyklu koszącego, w dowolnym grafie wejściowym. Mamy do dyspozycji procedurę KOSZĄCEDWUDZIELNE (opisaną niżej), znajdującą  $f$ -skojarzenie koszące w grafie dwudzielnym. Aby uruchomić tę procedurę musimy w jakiś sposób skonstruować dwudzielność. Robimy to metodą znaną z algorytmu znajdującego zwykle maksymalne skojarzenie: wprowadzamy orientację krawędzi i rozszczepiamy każdy wierzchołek  $v$  grafu wejściowego na dwa wierzchołki  $v_1$  i  $v_2$ , zwane pęczkami, przy czym rozszczepienie polega na tym, że spośród krawędzi incydentnych do  $v$ , krawędzie z wychodzącymi strzałkami są incydentne do  $v_1$ , a krawędzie z wchodzącymi strzałkami do  $v_2$ . W ten sposób otrzymujemy graf dwudzielny, który po lewej stronie ma wierzchołki z wychodzącymi strzałkami, a po prawej z wchodzącymi. Po znalezieniu  $f$ -skojarzenia koszącego w grafie dwudzielnym musimy zlepić pary pęczków. Otrzymamy wtedy pewien podgraf w grafie wejściowym, przy czym stopień każdego wierzchołka podgrafu nie przekracza  $2f(v)$ , gdyż  $\forall v : f(v_1) = f(v_2) = f(v)$ . Tak więc otrzymamy  $f$ -cykl zamiast  $f$ -skojarzenia.

#### PROCEDURA CYKL

1. Zorientuj każdą krawędź (czyli narysuj na niej strzałkę). Niech strzałka krawędzi  $vw$  wskazuje na  $v$  jeśli  $f(v) < f(w)$ , niech wskazuje na  $w$  jeśli  $f(w) < f(v)$ , w przeciwnym wypadku niech wskazuje na wierzchołek z mniejszym identyfikatorem.
2. Niech każdy wierzchołek rozszczepi się na dwa pęczki: z wchodzącymi strzałkami i z wychodzącymi strzałkami.



Otrzymasz w ten sposób "rozszczepiony" graf dwudzielny, który ma po lewej stronie wierzchołki z wychodzącymi strzałkami, natomiast po prawej z wchodzącymi.

Obu pęczkom przyporządkuj pojemność taką, jaką ma ich wierzchołek macierzysty.

3. W otrzymanym grafie dwudzielnym znajdź  $f$ -skojarzenie koszące lewostronnie, uruchamiając w nim procedurę KOSZĄCEDWUDZIELNE.
4. Niech każdy wierzchołek zlepi swoje dwa pęczki. Podgraf składający się z krawędzi  $f$ -skojarzenia (znalezionego w poprzednim punkcie) oznacz przez  $C$ .
5. Podgraf  $C$  jest wynikiem działania tej procedury.

**Twierdzenie 3.7** *Niech graf wejściowy spełnia warunek:*

$$\forall v : p \leq f(v) < q \quad (7)$$

*Procedura CYKL znajduje w grafie wejściowym  $f$ -cykl koszący, w czasie  $O(\log^3 n)$ .*

**Dowód** Najbardziej czasochłonną operacją procedury CYKL jest wywołanie procedury KOSZĄCEDWUDZIELNE, w punkcie 3. Procedura KOSZĄCEDWUDZIELNE działa w czasie  $O(\log^3 n)$ , tak więc procedura CYKL działa także w czasie  $O(\log^3 n)$ . Dzięki warunkowi (7), rozszczepiony graf dwudzielny spełnia założenia wymagane przez procedurę KOSZĄCEDWUDZIELNE.

Podgraf  $C$  jest niewątpliwie  $f$ -cyklem. Musimy pokazać, że jest  $f$ -cyklem koszącym. Rozszczepiony graf dwudzielny powstający w punkcie 2 procedury oznaczmy przez  $G_{LR}$ . W Fakcie 3.8 udowodnimy, że jeśli  $f$ -skojarzenie w  $G_{LR}$  jest koszące na  $L \cup R$ , to  $f$ -cykl  $C$  jest koszący. Tymczasem w punkcie 3 znajdujemy jedynie  $f$ -skojarzenie koszące na  $L$ . Musimy więc udowodnić, że w tym wypadku "koszenie na  $L$ " oznacza automatycznie "koszenie na  $L \cup R$ ".

Przypomnijmy, że każdy wierzchołek  $v$  rozszczepia się na dwa pęczki, z wychodzącymi i wchodzącymi strzałkami, których stopnie oznaczamy przez  $d_l(v)$  i  $d_r(v)$ . Dla każdej krawędzi (ze strzałką)  $e$ , oznaczmy pojemność "beltu" przez  $f_l(e)$ , a pojemność "grotu" przez  $f_r(e)$ . Z założenia  $\forall e : f_l(e) \geq f_r(e)$ .

Oczywiście

$$\sum_v f(v)d_l(v) = \sum_e f_l(e)$$

oraz

$$\sum_v f(v)d_r(v) = \sum_e f_r(e)$$

zatem

$$\sum_v f(v)d_l(v) \geq \sum_v f(v)d_r(v)$$

Wiemy, że  $\Phi^L[G_{LR}] = \sum_v f(v)d_l(v)$  oraz  $\Phi^R[G_{LR}] = \sum_v f(v)d_r(v)$ , zatem  $\Phi^L[G_{LR}] \geq \Phi^R[G_{LR}]$ . Tak więc, jeśli po modyfikacji wartość  $\Phi^L$  zmniejsza się o stały procent, to także wartość  $\Phi^{L \cup R} = \Phi^L + \Phi^R$  zmniejsza się o stały procent, co oznacza, że  $f$ -skojarzenie jest koszące na  $L \cup R$ .

Wydawałoby się, że istnieje prosta metoda znajdowania  $f$ -skojarzenia koszącego na  $L \cup R$ , polegająca na tym, że znajdujemy  $f$ -skojarzenie koszące na  $L$ , wykonujemy względem niego modyfikację, następnie znajdujemy  $f$ -skojarzenie koszące na  $R$

i na końcu akumulujemy oba  $f$ -skojarzenia. Nie możemy jednak zastosować takiej metody, gdyż aby znaleźć  $f$ -skojarzenie koszące na  $R$ , musi być spełniony warunek  $\forall v \in L : f(v) \geq p$  (wymagany przez procedurę KOSZĄCEDWUDZIELNE), tak więc konieczna jest  $p$ -modyfikacja zamiast modyfikacji. Jeśli wykonamy  $p$ -modyfikację, to na końcu otrzymamy  $f$ -skojarzenie  $p$ -koszące na  $L \cup R$  i  $f$ -cykl też będzie jedynie  $p$ -koszący, tymczasem trzeci etap procedury ZAMIANA wymaga  $f$ -cyklu koszącego.

▽

**Fakt 3.8** *Jeśli w "rozszczepionym" grafie dwudzielnym  $G_{LR}$  (powstającym w punkcie 2 procedury CYKL) znaleziono  $f$ -skojarzenie koszące na zbiorze  $L \cup R$ , to po zlepianiu par pęczków otrzymamy  $f$ -cykl koszący w  $G$ .*

**Dowód** Załóżmy, że  $C$  jest  $f$ -skojarzeniem koszącym w  $G_{LR}$ , na zbiorze  $L \cup R$ , czyli:

$$\Phi[G_{LR}/C] \leq (1 - u)\Phi[G_{LR}]$$

dla pewnego  $u \in (0, 1)$ . Musimy udowodnić, że po zlepianiu par pęczków,  $f$ -cykl  $C$  jest koszący w  $G$ , czyli:

$$\Phi[G/C] \leq (1 - u)\Phi[G]$$

Wystarczy więc udowodnić, że:

$$(a) \Phi[G/C] \leq \Phi[G_{LR}/C], \quad (b) \Phi[G_{LR}] = \Phi[G].$$

Dwa pęczki każdego wierzchołka  $v$  grafu  $G$  oznaczmy przez  $v_l, v_r$  gdzie  $v_l \in L, v_r \in R$ . Najpierw udowodnimy warunek (b). Niech:  $f(v) := f(v)[G], d(v) := d(v)[G], f(v_l) := f(v_l)[G_{LR}], d(v_l) := d(v_l)[G_{LR}], f(v_r) := f(v_r)[G_{LR}], d(v_r) := d(v_r)[G_{LR}]$ . Wiemy, że  $f(v) = f(v_l) = f(v_r)$  i  $d(v) = d(v_l) + d(v_r)$ . Z uwagi na to, że:

$$\Phi[G] = \sum_v f(v)d(v), \quad \Phi[G_{LR}] = \sum_v f(v_l)d(v_l) + \sum_v f(v_r)d(v_r)$$

warunek (b) niewątpliwie jest spełniony.

Teraz zajmiemy się dowodzeniem warunku (a). Wprowadźmy dodatkowe oznaczenia dla wierzchołka  $v$ :  $f'(v) := f(v)[G/C], d'(v) := d(v)[G/C]$ .

Wprowadźmy dodatkowe oznaczenia dla pęczków  $v_l, v_r$ :

$$\begin{aligned} f_l(v) &:= f(v_l)[G_{LR}], d_l(v) := d(v_l)[G_{LR}], f_r(v) := f(v_r)[G_{LR}], d_r(v) := d(v_r)[G_{LR}], \\ f'_l(v) &:= f(v_l)[G_{LR}/C], d'_l(v) := d(v_l)[G_{LR}/C], d_{C_l}(v) := d_C(v_l)[G_{LR}], \\ f'_r(v) &:= f(v_r)[G_{LR}/C], d'_r(v) := d(v_r)[G_{LR}/C], d_{C_r}(v) := d_C(v_r)[G_{LR}]. \end{aligned}$$

Wiemy, że:

$$\Phi[G/C] = \sum_v f'(v)d'(v), \quad \Phi[G_{LR}/C] = \sum_v f'_l(v)d'_l(v) + \sum_v f'_r(v)d'_r(v)$$

Aby udowodnić warunek (a) pokażemy, że:

$$\forall v \in V(G) : f'(v)d'(v) \leq f'_l(v)d'_l(v) + f'_r(v)d'_r(v) \quad (8)$$

Jeśli którykolwiek z wierzchołków  $v, v_l, v_r$  jest usuwany, to wtedy nierówność (8) jest spełniona (gdyż usuwanie  $v_l$  lub  $v_r$  implikuje usuwanie  $v$ ). Pozostaje nam więc

udowodnienie (8), gdy żaden z wierzchołków  $v$ ,  $v_l$ ,  $v_r$  nie jest usuwany. W dalszych rozważaniach będziemy opuszczać "(v)". Na początku zauważmy, że:

$$d' \leq d'_l + d'_r$$

gdyż stopień wierzchołka po modyfikacji względem pewnego podgrafu zależy od stopnia tego wierzchołka w podgrafie oraz od liczby usuwanych sąsiadów. W tym wypadku  $d_C = d_{C_l} + d_{C_r}$ , natomiast liczba usuwanych sąsiadów wierzchoła  $v$  przy zlepionych parach pęczków jest większa lub równa od liczby usuwanych sąsiadów pęczków  $v_l$  i  $v_r$  przy rozłączonych parach pęczków.

Wiemy, że:  $f'_l = f_l - d_{C_l}$ ,  $f'_r = f_r - d_{C_r}$ ,  $f_l = f_r = f$  oraz  $f' = f - d_C$ . Tak więc aby udowodnić nierówność (8) wystarczy wykazać, że:

$$(f - d_C)d' \leq (f_l - d_{C_l})d'_l + (f_r - d_{C_r})d'_r$$

co wynika z:

$$(f - (d_{C_l} + d_{C_r}))(d'_l + d'_r) \leq (f - d_{C_l})d'_l + (f - d_{C_r})d'_r$$

co jest równoważne:

$$(d_{C_l} + d_{C_r})(d'_l + d'_r) \geq d_{C_l}d'_l + d_{C_r}d'_r$$

▽

Teraz opiszemy procedurę ZAMIANA, której celem jest zamiana  $f$ -cyklu koszącego na  $f$ -skojarzenie  $p$ -koszące. Oczywiście procedura ta polega głównie na usuwaniu "nadmiarowych" krawędzi  $f$ -cyklu, ale jak się okaże, nie tylko. Procedura składa się z trzech etapów. W każdym etapie operujemy na pewnym podgrafie. W pierwszym i drugim etapie jedynie usuwamy krawędzie  $f$ -cyklu. W trzecim etapie nie tylko usuwamy, ale i dodajemy krawędzie. W każdym etapie stosowana jest inna metoda wyboru krawędzi do usunięcia lub dodania. Wcześniejsze etapy "przygotowują grunt" dla późniejszych, tj zapewniają spełnienie warunków, dzięki którym późniejsze etapy mogą być wykonane.

Wejście do procedury ZAMIANA stanowi graf  $G$ , wraz z  $f$ -cyklem koszącym  $C$ . Zbiór wierzchołków  $V(G)$  dzielimy na trzy rozłączne zbiory:

- $L = \{v : d_C(v) < f(v)\}$ ,  $L$  to zbiór wierzchołków nienasyconych
- $M = \{v : d_C(v) = f(v)\}$ ,  $M$  to zbiór wierzchołków nasyconych
- $R = \{v : f(v) < d_C(v) \leq 2f(v)\}$ ,  $R$  to zbiór wierzchołków z nadmiarem

Rysunek 16: Trzy etapy procedury ZAMIANA.

W każdym etapie procedury ZAMIANA usuwamy z podgrafu  $C$  krawędzie oraz (ewentualnie) dodajemy do podgrafu  $C$  nowe krawędzie, otrzymując podgraf  $C'$ . Dokładniej, jeśli zbiór usuwanych krawędzi oznaczymy przez  $X$ ,  $X \subseteq E(C) \subseteq E(G)$ , a zbiór dodawanych krawędzi oznaczymy przez  $Y$ ,  $Y \subseteq E(G)$ , to wtedy podgraf  $C'$  jest indukowany przez zbiór krawędzi  $E(C) \setminus X \cup Y$ . Zbiory  $L'$ ,  $M'$ ,  $R'$  definiujemy podobnie

jak zbiory  $L, M, R$  jednak za pomocą podgrafu  $C'$  zamiast  $C$ :  $L' = \{v : d_{C'}(v) < f(v)\}$ ,  $M' = \{v : d_{C'}(v) = f(v)\}$ ,  $R' = \{v : f(v) < d_{C'}(v) \leq 2f(v)\}$ . Aby uprościć zapis, na końcu każdego etapu zmienne  $C', L', M', R'$  przechodzą na  $C, L, M, R$ . Symbolem  $N_G(v)$  będziemy oznaczać zbiór sąsiadów wierzchołka  $v$  w grafie  $G$ . Symbol  $d_L(v)$  oznacza liczbę sąsiadów wierzchołka  $v$ , należących do zbioru  $L$ . Symbol  $d_{C_L}(v)$  oznacza liczbę sąsiadów wierzchołka  $v$ , w podgrafie  $C$ , należących do zbioru  $L$ .

Na rysunku 16 (strona 35) pokazano, którymi krawędziami zajmują się poszczególne etapy procedury ZAMIANA. A oto formalny zapis tej procedury:

#### PROCEDURA ZAMIANA

##### • Etap 1

(W tym etapie usuwamy krawędzie podgrafu  $C$  łączące wierzchołki zbioru  $R$ .)

Niech  $\hat{C}$  będzie podgrafem grafu  $C$ , indukowanym przez zbiór wierzchołków  $R$ .

Każdemu wierzchołkowi  $v \in V(\hat{C})$  przyporządkuj pojemność  $\hat{f}(v) := d_C(v) - f(v)$ .

W podgrafie  $\hat{C}$  znajdź PMfM  $X$ , przy pomocy procedury CZĘŚCIOWE.

$$C' := C \setminus X$$

##### • Etap 2

$$C := C', L := L', M := M', R := R'$$

(W tym etapie usuwamy krawędzie podgrafu  $C$  łączące zbiory  $L$  i  $R$ .)

Każdej krawędzi  $uv$  gdzie  $u \in L, v \in R$  przyporządkuj wagę:

$$\text{waga}(uv) := |N_{G \setminus C}(u) \cap L|.$$

Niech każdy wierzchołek zbioru  $R$  zaznaczy  $\min(d_C(v) - f(v), \frac{1}{2}d_{C_L}(v))$  krawędzi podgrafu  $C$ , łączących go ze zbiorem  $L$ , wybierając krawędzie z najmniejszymi wagami. Zbiór zaznaczonych krawędzi oznaczmy przez  $X$ .

$$C' := C \setminus X$$

##### • Etap 3

$$C := C', L := L', M := M', R := R'$$

(W tym etapie usuwamy krawędzie podgrafu  $C$  łączące zbiory  $M$  i  $R$  oraz dodajemy do podgrafu  $C$  krawędzie leżące między  $L$  i  $M$ .)

1. Każdej krawędzi  $uv$  gdzie  $u \in M, v \in R$  przyporządkuj wagę:

$$\text{waga}(uv) := |N_{G \setminus C}(u) \cap (L \cup M)|.$$

Niech każdy wierzchołek zbioru  $R$  zaznaczy  $d_C(v) - f(v)$  krawędzi podgrafu  $C$ , łączących go ze zbiorem  $M$ , wybierając krawędzie z najmniejszymi wagami. Zbiór zaznaczonych krawędzi oznaczmy przez  $X$ .

$$\tilde{C} := C \setminus X$$

2. Wykonaj  $p$ -modyfikację grafu względem  $\tilde{C}$ .

3. W podgrafie dwudzielnym między zbiorami wierzchołków  $L$  i  $M$ , znajdź  $f$ -skojarzenie  $Q$ , koszące na zbiorze  $L$ , przy pomocy procedury KOSZĄCEDWUDZIELNE.

4.  $C' := Q \cup \tilde{C}$ .

• Podgraf  $C'$  jest wynikiem działania tej procedury.

**Twierdzenie 3.9** *Mając na wejściu  $f$ -cykl koszący  $C$ , procedura ZAMIANA konstruuje  $f$ -skojarzenie  $p$ -koszące, w czasie  $O(\log^3 n)$ .*

**Dowód** Procedura CZĘŚCIOWE, uruchomiona w pierwszym etapie, działa w czasie  $O(\log n)$ . Procedura KOSZĄCEDWUDZIELNE, uruchomiona w trzecim etapie, działa w czasie  $O(\log^3 n)$ . Tak więc procedura ZAMIANA działa w czasie  $O(\log^3 n)$ .

Musimy udowodnić, że procedura ZAMIANA rzeczywiście zamienia  $f$ -cykl koszący  $C$  na  $f$ -skojarzenie  $p$ -koszące. Udowodnimy więc, że mimo usuwania krawędzi, podgraf  $C$  - z etapu na etap - zachowuje własność koszenia, oraz że po zakończeniu każdego etapu spełnione są warunki pozwalające usunąć nadmiar w następnych etapach.

Zacznijmy od udowodnienia, że po zakończeniu pierwszego etapu są spełnione następujące warunki:

$$(a) C' \text{ jest } f\text{-cyklem koszącym, (b) } \forall v \in R' : d_{C'_{L' \cup M'}}(v) \geq 2(d_{C'}(v) - f(v))$$

Punkt (b) oznacza, że w drugim i trzecim etapie będzie można usunąć nadmiar, mimo że każdemu wierzchołkowi z  $R$  wolno w tych etapach usunąć co najwyżej połowę incydentnych do siebie krawędzi podgrafu  $C$ .

Przypomnijmy, że w pierwszym etapie  $C' = C \setminus X$ , gdzie  $X$  jest  $f$ -skojarzeniem w podgrafie  $\hat{C} = C[R]$ , z pojemnościami  $\hat{f}(v) = d_C(v) - f(v)$ . Tak więc z podgrafu  $C$  są usuwane krawędzie łączące wierzchołki z nadmiarem, jednak żaden z końców usuwanej krawędzi nie może się stać wierzchołkiem nienasyconym. Zatem  $\Phi[G//C'] = \Phi[G//C]$ , z czego wynika warunek (a), gdyż  $C$  jest podgrafem koszącym.

Aby spełniony był warunek (b), podgraf  $X$  powinien mieć następującą własność:

$$\forall v \in R : i(v) \geq 2(\hat{f}(v) - d_X(v)) - d_{C_{L \cup M}}(v) \quad (9)$$

gdzie  $i(v)$  oznacza liczbę nasyconych sąsiadów wierzchołka  $v$  (nasyconych względem podgrafu  $X$ , w grafie  $\hat{C}$ , z pojemnościami  $\hat{f}(v)$ ). Wtedy, pamiętając że  $d_{C'}(v) = d_C(v) - d_X(v)$ , będzie można oszacować, dla  $v \in R'$ :

$$d_{C'_{L' \cup M'}}(v) \geq d_{C_{L \cup M}}(v) + i(v) \geq 2(\hat{f}(v) - d_X(v)) = 2(d_{C'}(v) - f(v))$$

Procedura CZĘŚCIOWE, uruchomiona w podgrafie  $\hat{C}$ , z pojemnościami  $\hat{f}(v)$ , znajduje podgraf  $X$ , będący PMfM-em, a więc spełniający (z definicji) warunek:

$$\forall v \in R : i(v) \geq 2(\hat{f}(v) - d_X(v)) - s(v)$$

gdzie  $s(v) := \max\{0, 2\hat{f}(v) - d_{\hat{C}}(v)\}$ . Wiemy, że:  $-s(v) \geq -d_{C_{L \cup M}}(v)$ , co można wywnioskować z faktu, że podgraf  $C$  jest  $f$ -cyklem, tak więc  $X$  spełnia warunek (9).

Teraz pokażemy, że po zakończeniu drugiego etapu są spełnione następujące warunki:

$$(a) C' \text{ jest } f\text{-cyklem koszącym, (b) } \forall v \in R' : d_{C'_{M'}}(v) \geq 2(d_{C'}(v) - f(v))$$

Ponownie, punkt (b) gwarantuje, że w trzecim etapie będzie można usunąć nadmiar, mimo że wierzchołkowi ze zbioru  $R$  wolno w trzecim etapie usunąć co najwyżej połowę incydentnych do siebie krawędzi podgrafu  $C$ .

Aby wykazać, że  $C'$  jest koszący zastosujemy poniższy fakt, którego dowód zostanie podany później. Potrzebujemy następującej definicji: dany jest podgraf  $Q$  i zbiór krawędzi  $U$ ,  $U \subseteq E(Q)$ . Zbiór  $U$  nazywamy zbiorem **krawędzi nadmiarowych**, gdy:

$$\{v : d_Q(v) > f(v)\} \subseteq \{v : d_{Q \setminus U}(v) \geq f(v)\}$$

**Fakt 3.10** (*Usuwanie nadmiaru przy pomocy wag.*)

Mamy dwa rozłączne zbiory wierzchołków  $A$  i  $B$  oraz podgraf koszący  $C$ . Zakładamy, że  $A \subseteq \{v : d_C(v) \leq f(v)\}$ ,  $B \subseteq \{v : d_C(v) \geq f(v)\}$  oraz, że dla dowolnego zbioru  $U$ ,  $U \subseteq E(C)$  krawędzi nadmiarowych, leżących między  $A$  i  $B$ , spełniony jest warunek:

$$\forall w \in A : d(w)[G/(C \setminus U)] = d(w)[G/C] \quad (10)$$

Wtedy istnieje procedura usuwania krawędzi podgrafu  $C$ , po której uruchomieniu otrzymujemy podgraf  $C'$  o następujących własnościach:

- (1)  $C'$  jest podgrafem koszącym
- (2) każdy wierzchołek  $v \in B$  traci  $\min(d_C(v) - f(v), \frac{1}{2}d_{C_A}(v))$  krawędzi podgrafu  $C$ , łączących go ze zbiorem  $A$ .

Aby wykazać, że  $C'$  jest koszący wystarczy zastosować powyższy fakt, podstawiając  $A := L$ ,  $B := R$ . Czynności wykonywane w drugim etapie to właśnie procedura, której istnienia dowodzi powyższy fakt. Aby zastosować fakt musimy wyjaśnić, dlaczego spełniony jest warunek (10). Pamiętajmy, że  $C' = C \setminus U$ , gdzie  $U$  jest zbiorem krawędzi nadmiarowych usuwanych w drugim etapie. Podczas modyfikacji względem  $C'$ , dla dowolnego wierzchołka  $u \in L$ , sąsiedzi w  $M$  i  $R$  są usuwani, natomiast sąsiedzi w  $L$  nie są usuwani, tak więc stopień wierzchołka  $u$  po modyfikacji wynosi:

$$d(u)[G/C'] = d(u)[G/C] = |N_{G \setminus C}(u) \cap L|$$

i nie zależy od zbioru krawędzi  $U$ . Skoro spełniony jest warunek (10) to Fakt 3.10 stwierdza, że  $C'$  jest koszący.

Teraz uzasadnimy punkt (b). Na początku drugiego etapu jest spełniony warunek  $\forall v \in R : d_{C_{L \cup M}}(v) \geq 2(d_C(v) - f(v))$ , czyli  $\frac{1}{2}d_{C_L}(v) + \frac{1}{2}d_{C_M}(v) + d_{C_R}(v) \leq f(v)$ . Dla każdego  $v \in R'$  mamy  $d_{C'_{L'}}(v) = \frac{1}{2}d_{C_L}(v)$ , gdyż zgodnie z Faktem 3.10, wierzchołek  $v \in R$  albo zlikwiduje nadmiar i znajdzie się w zbiorze  $M'$ , albo straci  $\frac{1}{2}d_{C_L}(v)$  krawędzi łączących go z  $L$  i znajdzie się w zbiorze  $R'$ . Biorąc pod uwagę, że  $d_{C'_{M'}}(v) = d_{C_M}(v)$  i  $d_{C'_{R'}}(v) = d_{C_R}(v)$ , otrzymamy  $\forall v \in R' : d_{C'_{L'}}(v) + \frac{1}{2}d_{C'_{M'}}(v) + d_{C'_{R'}}(v) \leq f(v)$ , co jest równoważne z warunkiem (b).

Na końcu trzeciego etapu podgraf  $C'$  jest oczywiście  $f$ -skojarzeniem. Aby zakończyć dowód twierdzenia, musimy pokazać, że jest  $f$ -skojarzeniem  $p$ -koszącym.

Napierw udowodnimy, że  $f$ -skojarzenie  $\tilde{C}$ , znajdujące w punkcie 1 trzeciego etapu, jest koszące na zbiorze  $M \cup R$ . Wystarczy zastosować Fakt 3.10, podstawiając  $A := M$ ,  $B := R$ . Znow czynności wykonywane w celu znalezienia  $\tilde{C}$ , to procedura której istnienia dowodzi Fakt 3.10. Aby zastosować ten fakt musimy wprowadzić "słabą" modyfikację, która tym się różni od zwykłej modyfikacji, że wierzchołki ze zbioru  $M$  nie są usuwane (przy zwykłej modyfikacji powinny być usuwane, gdyż są nasycone). Fakt 3.10 można udowodnić także przy słabej modyfikacji. Przy słabej modyfikacji podgraf  $C$  jest koszący jedynie na zbiorze  $M \cup R$ , gdyż po takiej modyfikacji wierzchołki ze zbioru  $R$  znikną, wierzchołki z  $M$  będą miały zerową pojemność, natomiast nic nie można powiedzieć o wierzchołkach z  $L$ . (Zauważmy w tym miejscu, że gdyby  $f$ -cykl  $C$  był  $p$ -koszący, to nie można by było zastosować tego argumentu - dlatego właśnie procedura CYKL musi znajdować  $f$ -cykl koszący, mimo że łatwiej jest znaleźć  $f$ -cykl  $p$ -koszący).

Wyjaśnimy teraz, dlaczego spełniony jest warunek (10) powyższego faktu. Pamiętajmy, że  $\tilde{C} = C \setminus U$ . Podczas słabej modyfikacji względem  $\tilde{C}$ , dla dowolnego wierzchołka

$u \in M$ , sąsiedzi w  $R$  są usuwani, natomiast sąsiedzi w  $L$  i  $M$  nie są usuwani, niezależnie od zbioru usuwanych krawędzi nadmiarowych  $U$ . Zatem stopień wierzchołka  $u$  po słabej modyfikacji wynosi:

$$d(u)[G//\tilde{C}] = d(u)[G//C] = |N_{G \setminus C}(u) \cap (L \cup M)|$$

Tak więc na podstawie Faktu 3.10 wiemy, że  $\tilde{C}$  jest koszące na zbiorze  $M \cup R$ , przy słabej modyfikacji. Skoro jednak jest koszące przy słabej modyfikacji, to jest koszące także przy "normalnej" modyfikacji.

Wiemy, że  $\tilde{C}$  jest koszące na zbiorze  $M \cup R$ . Teraz wyjaśnimy, dlaczego  $f$ -skojarzenie  $C'$  jest  $p$ -koszące na wszystkich wierzchołkach. W tym celu definiujemy dwie funkcje:

$$\Phi_1^Z := \sum_{v \in L \cap Z} f(v)d_{M \cup R}(v) + \sum_{v \in (M \cup R) \cap Z} f(v)d(v)$$

$$\Phi_2^Z := \sum_{v \in L \cap Z} f(v)d_L(v)$$

Gdy  $Z = L \cup M \cup R$ , to symbol  $Z$  będziemy opuszczać. Oczywiście  $\Phi = \Phi_1 + \Phi_2$ .

Gdybyśmy wykonali modyfikację względem  $\tilde{C}$ , to wartość funkcji  $\Phi_1^{M \cup R}$  zmniejszyła by się o stały procent, gdyż  $\Phi_1^{M \cup R} = \Phi^{M \cup R}$ . W punkcie 2 trzeciego etapu procedury wykonuje się  $p$ -modyfikację względem  $\tilde{C}$ , co pociąga za sobą m.in. zniknięcie wierzchołków zbioru  $R$ . Następnie, w punkcie 3 trzeciego etapu, znajdujemy  $f$ -skojarzenie koszące  $Q$ , w podgrafie dwudzielnym między zbiorami  $L$  i  $M$ . Gdybyśmy wykonali modyfikację względem  $Q$ , to zmniejszyła by się o stały procent wartość funkcji:

$$\Phi_1^L = \sum_{v \in L} f(v)d_M(v)$$

Tak więc, po  $p$ -modyfikacji względem  $C' := Q \cup \tilde{C}$ , wartość funkcji  $\Phi_1$  zmniejsza się o stały procent, co wynika z Faktu 3.11 i następującej po nim uwagi (patrz strona 40). Pozostaje udowodnić, że także wartość funkcji  $\Phi$  zmniejsza się o stały procent, czyli że  $C'$  jest  $f$ -skojarzeniem  $p$ -koszącym.

Rozważmy dwa przypadki:

1.  $\Phi_2[G] < c\Phi_1[G]$ , gdzie  $c$  jest stałą całkowitą.

Skoro  $\Phi_1$  zmniejsza wartość o stały procent, a  $\Phi_2$  jest mniejsze od  $\Phi_1$  (z dokładnością do stałego czynnika), to  $\Phi = \Phi_1 + \Phi_2$  także zmniejsza wartość o stały procent.

2.  $\Phi_2[G] \geq c\Phi_1[G]$ .

W tym wypadku udowodnimy, że  $\Phi_2$  zmniejsza wartość o stały procent. Możemy oszacować:

$$\Phi_2[G/p/C'] \leq \Phi_2[G/p/C] \leq \Phi[G/p/C] \leq (1 - u_0)\Phi[G] \leq (1 - u_1)\Phi_2[G]$$

gdzie  $u_0, u_1 \in (0, 1)$ . W powyższym wzorze pierwszą nierówność można wywnioskować ze sposobu w jaki  $C'$  powstaje z  $C$ , druga nierówność jest oczywista, trzecia nierówność wynika z założenia trzeciego etapu, natomiast czwarta nierówność wynika z założenia  $\Phi_2[G] \geq c\Phi_1[G]$ , gdyż:

$$(1 - u_0)\Phi[G] = (1 - u_0)(\Phi_1[G] + \Phi_2[G]) \leq (1 - u_0)\left(\frac{1}{c} + 1\right)\Phi_2[G]$$

przy czym

$$(1 - u_0)\left(\frac{1}{c} + 1\right) < 1 \Leftrightarrow \frac{1}{1+c} < u_0$$

a ostatnia nierówność jest spełniona dla odpowiednio dużej stałej  $c$ .

Skoro  $\Phi_1$  i  $\Phi_2$  zmniejszają wartości o stały procent, to także  $\Phi = \Phi_1 + \Phi_2$  zmniejsza wartość o stały procent.

▽

**Uwaga:** Dodawanie krawędzi do  $f$ -cyklu  $C$  w trzecim etapie procedury ZAMIANA jest odpowiedzialne za to, że ostatecznie otrzymujemy  $f$ -skojarzenie  $p$ -koszące - słabsze od koszącego. Warto wyjaśnić dlaczego zrezygnowano z silniejszej procedury, zamieniającej  $f$ -cykl koszący na  $f$ -skojarzenie koszące. Na rysunku 17 (strona 40) widać sytuację, w której bardzo trudno zamienić  $f$ -cykl  $C$  na  $f$ -skojarzenie koszące  $C'$ , przy pomocy samego usuwania krawędzi  $f$ -cyklu (aby nie zaciemniać obrazu nie wszystkie krawędzie są narysowane; podgraf  $C$  składa się ze wszystkich krawędzi między zbiorami  $M$  i  $R$ ; podgraf  $C'$  składa się z tych krawędzi podgrafu  $C$ , których nie usunięto).

Rysunek 17: Dlaczego trudno jest zamienić  $f$ -cykl na  $f$ -skojarzenie koszące.

Aby  $C'$  był koszący wartość funkcji  $\Phi$  musi się zmniejszać o stały procent, po modyfikacji względem  $C'$ . Oczywiście  $\Phi = \Phi^L + \Phi^M + \Phi^R$ , przy czym składnik  $\Phi^L$  dominuje nad pozostałymi składnikami, tak więc wartość  $\Phi^L$  musi się zmniejszać o stały procent. Niech  $M' := \{v \in M : d_{C'}(v) = f(v)\}$ . Aby wartość  $\Phi^L$  zmniejszała się o stały procent, musi być spełniony następujący warunek:

$$|M'| > u|M|$$

gdzie  $u \in (0, 1)$ . Trudno sobie wyobrazić metodę usuwania nadmiaru z  $f$ -cyklu  $C$ , która by zapewniała spełnienie powyższego warunku.

Teraz przypomnimy i udowodnimy fakty wykorzystywane w dowodzie Twierdzenia 3.9.

**Fakt 3.11** *Mamy 2 rozłączne zbiory wierzchołków  $A$  i  $B$ . Znajdujemy  $f$ -skojarzenie  $M_A$  koszące na  $A$ , następnie wykonujemy modyfikację względem  $M_A$ , potem znajdujemy  $f$ -skojarzenie  $M_B$  koszące na  $B$ . Podgraf  $M := M_A \cup M_B$  jest  $f$ -skojarzeniem koszącym na  $A \cup B$ .*

**Dowód** Oczywiście dowód pomijamy.

▽

**Uwaga:** Powyższy fakt jest prawdziwy także jeśli wykonujemy  $p$ -modyfikację zamiast modyfikacji. Otrzymamy wtedy  $f$ -skojarzenie  $p$ -koszące na  $A \cup B$ . Fakt ten jest prawdziwy także jeśli "koszenie" zdefiniujemy przy pomocy funkcji  $\Phi_1$  z Twierdzenia 3.9 zamiast funkcji  $\Phi$ .

Przypomnijmy, że zbiór **krawędzi nadmiarowych** to taki zbiór krawędzi podgrafu, że po ich usunięciu z podgrafu, wierzchołki które wcześniej miały nadmiar zachowują go lub stają się nasycone.



Rysunek 18: Usuwanie nadmiaru przy pomocy wag.

**Fakt 3.10** (*Usuwanie nadmiaru przy pomocy wag.*)

Mamy 2 rozłączne zbiory wierzchołków  $A$  i  $B$  oraz podgraf koszący  $C$ . Zakładamy, że  $A \subseteq \{v : d_C(v) \leq f(v)\}$ ,  $B \subseteq \{v : d_C(v) \geq f(v)\}$  oraz, że dla dowolnego zbioru  $U$ ,  $U \subseteq E(C)$  krawędzi nadmiarowych, leżących między  $A$  i  $B$ , spełniony jest warunek:

$$\forall w \in A : d(w)[G/(C \setminus U)] = d(w)[G/C]$$

Wtedy istnieje procedura usuwania krawędzi podgrafu  $C$ , po której uruchomieniu otrzymujemy podgraf  $C'$  o następujących własnościach:

- (a)  $C'$  jest podgrafem koszącym,
- (b) każdy wierzchołek  $v \in B$  traci  $\min(d_C(v) - f(v), \frac{1}{2}d_{C_A}(v))$  krawędzi podgrafu  $C$ , łączących go ze zbiorem  $A$ .

**Dowód** Każdej krawędzi  $wv$ , gdzie  $w \in A$ ,  $v \in B$ ,  $wv \in C$ , przyporządkowujemy wagę:

$$\text{waga}(wv) := d(w)[G/C]$$

Wprowadźmy oznaczenia:  $f'(v) := f(v)[G/C]$ ,  $d'(v) := d(v)[G/C]$ ,  $f''(v) := f(v)[G/(C \setminus U)]$ ,  $d''(v) := d(v)[G/(C \setminus U)]$ , gdzie  $U$  jest dowolnym zbiorem krawędzi nadmiarowych, leżących między zbiorami  $A$  i  $B$ .

Z założeń faktu można wywnioskować, że:

$$\forall w \notin A : f''(w)d''(w) = f'(w)d'(w)$$

oraz, że:

$$\forall w \in A : f''(w)d''(w) = f'(w)d'(w) + d_U(w)d'(w)$$

Tak więc wagi mają następującą, zasadniczą, własność:

$$\Phi[G/(C \setminus U)] = \Phi[G/C] + \sum_{e \in U} \text{waga}(e)$$

Procedura usuwania krawędzi polega na tym, że każdy wierzchołek  $v \in B$  wybiera do usunięcia krawędzie z najmniejszymi wagami, w liczbie:

$$\min(d_C(v) - f(v), \frac{1}{2}d_{C_A}(v))$$

spośród krawędzi podgrafu  $C$ , łączących go ze zbiorem  $A$ . Jak widać, warunek (b) jest spełniony. Musimy jeszcze udowodnić, że jest spełniony warunek (a), czyli że  $C \setminus U$  jest podgrafem koszącym, gdzie  $U$  oznacza zbiór krawędzi wybranych do usunięcia.

Wprowadźmy dodatkowo oznaczenia:

$$\Phi := \Phi[G], \Phi' := \Phi[G/C], \Phi'' := \Phi[G/(C \setminus U)],$$

$$W_U := \sum_{wu \in U, w \in A, v \in B} \text{waga}(wu), W_C := \sum_{wu \in C, w \in A, v \in B} \text{waga}(wu).$$

Mamy udowodnić, że  $C \setminus U$  jest podgrafem koszącym, czyli:

$$\Phi'' < (1 - u_1)\Phi, \quad u_1 \in (0, 1) \tag{11}$$

Zauważmy, że  $\Phi'' - \Phi' = W_U$ , oraz że  $W_U \leq \frac{1}{2}W_C$ . Możemy ponadto oszacować  $W_C \leq \sum_{w \in A} d'(w)d_C(w)$ . Tak więc:

$$\Phi'' - \Phi' \leq \frac{1}{2} \sum_{w \in A} d'(w)d_C(w) \quad (12)$$

Jeśli teraz pokażemy, że:

$$\Phi - \Phi' \geq \sum_{w \in A} d'(w)d_C(w) \quad (13)$$

to wtedy ze wzorów (12) i (13) oraz z tego, że  $C$  jest podgrafem koszącym, czyli:

$$\Phi' < (1 - u_0)\Phi, \quad u_0 \in (0, 1)$$

będzie wynikało (11). Aby udowodnić wzór (13) wystarczy zauważyć, że:

$$\Phi - \Phi' \geq \sum_{w \in A} (d(w)f(w) - d'(w)f'(w)) \geq \sum_{w \in A} d'(w)d_C(w)$$

gdyż dla  $w \in A$  mamy  $f'(w) = f(w) - d_C(w)$ .

▽

### 3.4 Znajdowanie $f$ -skojarzenia koszącego lewostronnie w grafie dwudzielnym.

Na początku musimy wprowadzić pojęcie  $(D, F)$ -bloku oraz macierzy  $(D, F)$ -bloków, w grafie dwudzielnym  $G_{LR}$ .

**Definicja 3.12** Niech  $A := \{v \in L : d(v) \in (D/2, D], f(v) \in (F/2, F]\}$  oraz niech  $B$  będzie zbiorem wszystkich sąsiadów wierzchołków ze zbioru  $A$ . Podgraf  $Q_{AB}$  grafu dwudzielnego  $G_{LR}$ , składający się ze wszystkich krawędzi grafu  $G_{LR}$  incydentnych do zbioru  $A$ , nazywamy  **$(D, F)$ -blokiem**.

Zbiór następujących  $(D, F)$ -bloków:

$$\begin{array}{cccccc} (n, n) & (n, \frac{n}{2}) & (n, \frac{n}{4}) & \dots & (n, 4) & (n, 2) & (n, 1) \\ & (\frac{n}{2}, \frac{n}{2}) & (\frac{n}{2}, \frac{n}{4}) & \dots & (\frac{n}{2}, 4) & (\frac{n}{2}, 2) & (\frac{n}{2}, 1) \\ & & (\frac{n}{4}, \frac{n}{4}) & \dots & (\frac{n}{4}, 4) & (\frac{n}{4}, 2) & (\frac{n}{4}, 1) \\ & & & \dots & \dots & \dots & \dots \\ & & & & (4, 4) & (4, 2) & (4, 1) \\ & & & & & (2, 2) & (2, 1) \\ & & & & & & (1, 1) \end{array}$$

nazywamy **macierzą bloków** w grafie dwudzielnym  $G_{LR}$ . Macierz bloków wyznacza podział grafu dwudzielnego na rozłączne krawędziowo podgrafy. Równocześnie zbiory  $A$  bloków wyznaczają podział zbioru  $L$ .

Bloki macierzy można **rozłączyć**, przy czym każdy wierzchołek  $v \in R$  rozszczepia się na pęczki, po jednym dla każdego bloku incydentnego do  $v$ . W każdym z rozłączonych

bloków będziemy znajdować  $f$ -spanner, o następującej definicji:

**Definicja 3.13** Niech  $S$  będzie podgrafem  $(D,F)$ -bloku  $Q_{AB}$ . Oznaczmy  $l(S) := V(S) \cap A$ ,  $r(S) := V(S) \cap B$ . Podgraf  $S$  nazywamy  $f$ -spannerem, jeśli spełnia następujące warunki:

- $|l(S)| \geq u_0|A|$ , gdzie  $u_0$  jest stałą niezależną od  $n$ ,  $u_0 \in (0, 1)$
- $\forall v \in l(S) : u_1 f(v) \leq d_S(v) \leq c_1 f(v)$ , gdzie  $u_1$  i  $c_1$  to stałe,  $u_1 \in (0, 1)$
- $\forall v \in r(S) : d_S(v) < c_1 d_Q(v) \frac{F}{D} + 1$

Jeśli w  $(D,F)$ -bloku  $Q_{AB}$  znajdziemy  $f$ -spanner, to będzie go można zamienić na  $f$ -skojarzenie  $M$  o następującej własności: ” $\sum_{v \in A} f(v)$  lub  $\sum_{v \in A} d(v)$  zmniejsza wartość o stały procent, po modyfikacji względem  $M$ ”, co zostanie udowodnione w Twierdzeniu 3.14.

Wyjaśnimy teraz skąd wzięła się taka a nie inna postać funkcji  $\Phi$ , w definicji koszenia. Okazuje się że, potrafimy znaleźć takie  $f$ -skojarzenie w grafie wejściowym, że po modyfikacji względem niego, w jednych obszarach grafu zmniejsza się o stały procent suma stopni wierzchołków, a w innych obszarach zmniejsza się suma pojemności. (Przez ”obszar” rozumiem zbiór wierzchołków; aby dalsze rozumowanie było poprawne, obszary muszą się składać z wierzchołków o zbliżonych stopniach i o zbliżonych pojemnościach). Nic jednak nie można powiedzieć o sumie stopni, czy też o sumie pojemności wszystkich wierzchołków. Zauważmy, że jeśli w pewnym obszarze zmniejsza się suma stopni ”lub” suma pojemności, to niezależnie od tego który składnik alternatywy jest prawdziwy, zmniejsza się o stały procent wartość sumy  $\sum_v f(v)d(v)$ , gdzie sumowanie przebiega po wierzchołkach obszaru. Tak więc także wartość funkcji  $\Phi := \sum_v f(v)d(v)$ , w której sumowanie przebiega po wszystkich wierzchołkach, zmniejsza się o stały procent.

Najpierw zostanie omówiona procedura BŁOK, znajdująca  $f$ -skojarzenie koszące w pojedynczym bloku, a następnie zajmiemy się procedurą KOSZĄCEDWUDZIELNE, która potrafi zamienić  $f$ -skojarzenia koszące w blokach na jedno duże  $f$ -skojarzenie koszące w grafie dwudzielnym.

Przypuśćmy, że graf dwudzielny  $Q_{AB}$  zawiera podgraf  $S$ . Zdanie ”z podgrafu  $S$  usunięto nadmiar na zbiorze  $A$ , otrzymując podgraf  $S'$ ” oznacza, że dla każdego wierzchołka  $v \in A$  wykonano następujące czynności:

1. jeśli  $d_S(v) \leq f(v)$ , to zaznaczono wszystkie krawędzie podgrafu  $S$  incydentne do  $v$
2. jeśli  $d_S(v) > f(v)$ , to zaznaczono  $f(v)$  spośród krawędzi podgrafu  $S$  incydentnych do  $v$ .

Następnie wszystkie zaznaczone krawędzie utworzyły podgraf  $S'$ .

Procedura BŁOK uruchomiona w bloku  $Q_{AB}$  najpierw znajduje  $f$ -spanner, a następnie usuwa nadmiar, najpierw na zbiorze  $A$ , potem na zbiorze  $B$ , dzięki czemu otrzymujemy  $f$ -skojarzenie. W Twierdzeniu 3.14 udowodnimy, że jest ono koszące na zbiorze  $A$ .

## PROCEDURA BLOK

1. Znajdź  $f$ -spanner w  $(D, F)$ -bloku  $Q_{AB}$ .
2. Usuń nadmiar na zbiorze  $A$ .
3. Usuń nadmiar na zbiorze  $B$ .
4. Otrzymany podgraf jest wynikiem działania tej procedury.

Rysunek 19: Zamiana  $f$ -spannera na  $f$ -skojarzenie.

**Twierdzenie 3.14** *Procedura BLOK znajduje  $f$ -skojarzenie koszące, na zbiorze  $A$ , w pojedynczym  $(D, F)$ -bloku  $Q_{AB}$ , w czasie  $O(\log^3 n)$ .*

**Dowód** W  $(D, F)$ -bloku można znaleźć  $f$ -spanner w czasie  $O(\log^3 n)$ . Znajdujemy go w podobny sposób, w jaki znajdowano "zwykły" spanner, w algorytmie znajdującym "zwykle" maksymalne skojarzenie, przedstawionym w rozdziale drugim. Jak łatwo się przekonać, jeśli w procedurze SPANNER z drugiego rozdziału, weźmiemy:

$$k := \log \frac{D}{F} - c$$

to po jej uruchomieniu otrzymamy podgraf spełniający definicję  $f$ -spannera. Procedura SPANNER działa w czasie  $O(\log^3 n)$ . Punkty 2 i 3 procedury BLOK wykonują się w stałym czasie, tak więc czas działania całej procedury można oszacować przez  $O(\log^3 n)$ .

Wprowadźmy następujące oznaczenia: niech  $S$  będzie  $f$ -spannerem w  $(D, F)$ -bloku  $Q_{AB}$ , znajdowanym w punkcie 1 procedury. Przez  $P$  oznaczmy to, co zostanie z  $S$  po usunięciu nadmiaru na zbiorze  $A$ . Przez  $M$  oznaczmy to, co zostanie z  $P$  po usunięciu nadmiaru na zbiorze  $B$ . Oczywiście  $M$ , czyli wynik działania procedury BLOK, jest  $f$ -skojarzeniem. Wszystkie oznaczenia pokazano na rysunku 19 (strona 44).

Udowodnimy że podgraf  $M$  jest  $f$ -skojarzeniem koszącym na zbiorze  $A$ . Dokładniej pokażemy, że po modyfikacji względem  $M$  jedna z sum,  $\sum_{v \in A} d(v)$  lub  $\sum_{v \in A} f(v)$ , zmniejsza wartość o stały procent. Wynika stąd, że następująca funkcja:

$$\Phi^A = \sum_{v \in A} f(v)d(v)$$

zmniejsza wartość o stały procent po modyfikacji, czyli że  $M$  jest  $f$ -skojarzeniem koszącym na zbiorze  $A$ .

Niech  $m := |E(Q_{AB})|$  oraz niech  $B' := \{v \in B : d_P(v) \geq f(v)\}$ . Zauważmy także, że podgraf  $P$  spełnia następujące warunki:

- $l(P) = l(S) \geq u_0|A|$
- $\forall v \in r(P) : d_P(v) \leq d_S(v) < c_1 d(v) \frac{F}{D} + 1$

Jak już wspomnieliśmy, pokażemy że jedna z sum,  $\sum_{v \in A} d(v)$  lub  $\sum_{v \in A} f(v)$ , zmniejsza wartość o stały procent, po modyfikacji względem  $M$ . W tym celu należy rozważyć następujące dwa przypadki:

$$1. \sum_{v \in B'} d(v) > u_2 m, \quad 2. \sum_{v \in B'} d(v) \leq u_2 m$$

gdzie  $u_2 \in (0, 1)$ . W pierwszym przypadku jest oczywiste, że  $\sum_{v \in A} d(v)$  zmniejsza wartość o stały procent, ponieważ wierzchołki ze zbioru  $B'$  są nasycone, a zatem są usuwane podczas modyfikacji. W drugim przypadku pokażemy, że zmniejsza się o stały procent wartość  $\sum_{v \in A} f(v)$ . Musimy więc udowodnić, że:

$$\sum_{v \in A} d_M(v) \geq u_3 \sum_{v \in A} f(v), \quad u_3 \in (0, 1)$$

Pamiętając, że podczas usuwania nadmiaru na  $P$  jedynie wierzchołki ze zbioru  $B'$  usuwają krawędzie, oraz że każdy taki wierzchołek może usunąć najwyżej  $d_P(v) - f(v) \leq d_P(v) - 1$  krawędzi, możemy oszacować:

$$\begin{aligned} \sum_{v \in A} d_M(v) &\geq \sum_{v \in A} d_P(v) - \sum_{v \in B'} (d_P(v) - 1) \geq \\ &\sum_{v \in A} d_P(v) - \sum_{v \in B'} c_1 d(v) \frac{F}{D} \geq \sum_{v \in A} d_P(v) - c_1 \frac{F}{D} \sum_{v \in B'} d(v) \geq \\ &\sum_{v \in A} d_P(v) - c_1 \frac{F}{D} u_2 m \end{aligned}$$

Pamiętając, że  $m \leq D|A|$  oraz że:

$$\sum_{v \in A} d_P(v) \geq \sum_{v \in l(P)} u_1 f(v) \geq u_1 |l(P)| \frac{F}{2} \geq u_0 u_1 \frac{1}{2} |A| F$$

możemy dalej szacować:

$$\begin{aligned} \sum_{v \in A} d_P(v) - c_1 \frac{F}{D} u_2 m &\geq u_0 u_1 \frac{1}{2} |A| F - c_1 \frac{F}{D} u_2 D |A| = \\ u_0 u_1 \frac{1}{2} |A| F - c_1 u_2 |A| F &= (u_0 u_1 \frac{1}{2} - c_1 u_2) |A| F = u_3 |A| F \geq u_3 \sum_{v \in A} f(v) \end{aligned}$$

czyli

$$\sum_{v \in A} d_M(v) \geq u_3 \sum_{v \in A} f(v)$$

Aby uzasadnić, że  $\Phi^A = \sum_{v \in A} f(v) d(v)$  zmniejsza wartość o stały procent po modyfikacji, oznaczmy przez  $d'(v)$ ,  $f'(v)$ ,  $\Phi^{A'}$  wartości  $d(v)$ ,  $f(v)$ ,  $\Phi^A$  po modyfikacji względem  $M$ . Zakładamy, że  $\sum_{v \in A} f'(v) < (1 - u) \sum_{v \in A} f(v)$  lub  $\sum_{v \in A} d'(v) < (1 - u) \sum_{v \in A} d(v)$ . Trywialne szacowanie daje:

$$\Phi^{A'} \leq 2(1 - u) \Phi^A$$

Oznacza to spadek o stały procent pod warunkiem, że  $u > \frac{1}{2}$ . W przeciwnym wypadku możemy używać przedziałów  $(D/x, D]$  i  $(F/x, F]$  gdzie  $x \in (1, 2)$ , zamiast  $(D/2, D]$  i

$(F/2, F]$ . Wtedy wystarczy, że  $u > 1 - \frac{1}{x}$ , a zmieniona definicja przedziałów nie ma większego wpływu na przedstawione tu rozważania.

**Uwaga:** Między punktami 2 i 3 procedury BŁOK możemy dowolnie obniżyć pojemności wierzchołków ze zbioru  $B$ , nawet do 1, a podgraf który otrzymamy w wyniku działania tej procedury będzie nadal  $f$ -skojarzeniem koszącym na zbiorze  $A$  (oczywiście, w bloku z obniżonymi pojemnościami).

Powyższa uwaga nie wymaga dowodu, ponieważ  $f$ -skojarzenie, z jakim mamy tu do czynienia, powstaje z  $f$ -spannera, który nie zależy od pojemności po prawej stronie bloku. Zauważmy jednak, że nie możemy obniżyć pojemności wierzchołka do 0 i usunąć wszystkich incydentnych do niego krawędzi podgrafu  $P$ , gdyż wtedy nie można by przeprowadzić szacowania z drugiego przypadku, otrzymalibyśmy bowiem:

$$\sum_{v \in A} d_M(v) \geq \sum_{v \in A} d_P(v) - \sum_{v \in B'} d_P(v) \geq \sum_{v \in A} d_P(v) - c_1 \frac{F}{D} u_2 m - |B'|$$

▽

Obecnie omówimy procedurę KOSZĄCEDWUDZIELNE, znajdującą  $f$ -skojarzenie koszące na zbiorze  $L$ , w grafie dwudzielnym  $G_{LR}$ . Procedura ta traktuje graf dwudzielnym jako "macierz bloków", rozłącza bloki, potem we wszystkich blokach (równocześnie) znajduje  $f$ -skojarzenia koszące, przy pomocy procedury BŁOK, następnie łączy bloki oraz zamienia  $f$ -skojarzenia w blokach w jedno duże  $f$ -skojarzenie, w całym grafie dwudzielnym. Procedura KOSZĄCEDWUDZIELNE może być uruchamiana jedynie na grafie dwudzielnym spełniającym pewien warunek: pojemności wierzchołków zbioru  $R$  muszą być dostatecznie duże (dokładne sformułowanie warunku znajduje się w Twierdzeniu 3.15). Ma to związek z usuwaniem nadmiaru, który może się pojawić na zbiorze  $R$ , po złączeniu bloków. Rysunek 20 (strona 46) ilustruje rozłączanie bloków, z punktu widzenia wierzchołka ze zbioru  $R$ .

Rysunek 20: Rozłączanie bloków.

#### PROCEDURA KOSZĄCEDWUDZIELNE

1. Rozłącz bloki macierzy bloków. Wierzchołki zbioru  $R$  zostaną rozszczepione na pęczki. Pęczki mają taką samą pojemność jak ich wierzchołek macierzysty w grafie  $G_{LR}$ .
2. W rozłączonych blokach znajdź  $f$ -skojarzenia koszące, przy pomocy procedury BŁOK.
3. Złącz bloki macierzy bloków. Otrzymasz podgraf, składający się z krawędzi  $f$ -skojarzeń w blokach. Na wierzchołkach zbioru  $R$  może pojawić się nadmiar.
4. Niech każdy wierzchołek zbioru  $R$  (z nadmiarem) usunie nadmiar w taki sposób, że w każdym bloku incydentnym do tego wierzchołka pozostanie przynajmniej jedna krawędź.
5. Podgraf, jaki otrzymasz, jest wynikiem działania tej procedury.

**Twierdzenie 3.15** *Procedura KOSZĄCEDWUDZIELNE znajduje  $f$ -skojarzenie koszące na zbiorze  $L$ , w grafie dwudzielnym  $G_{LR}$ , spełniającym założenia:*

$$\forall v \in L : f(v) < q \quad (14)$$

$$\forall v \in R : f(v) \geq \log^2 q \quad (15)$$

*Procedura działa w czasie  $O(\log^3 n)$ .*

**Dowód** Najbardziej czasochłonną operacją w procedurze KOSZĄCEDWUDZIELNE jest wywołanie procedury BLOK, w punkcie 2, działającej w czasie  $O(\log^3 n)$ . Dlatego czas działania procedury KOSZĄCEDWUDZIELNE to  $O(\log^3 n)$ .

W punkcie 2 procedury, w rozłączonych blokach znajdujemy  $f$ -skojarzenia koszące. Po złączeniu bloków, w punkcie 3, krawędzie  $f$ -skojarzeń tworzą podgraf, który nie jest  $f$ -skojarzeniem, gdyż na zbiorze  $R$  może wystąpić nadmiar (stopień podgrafu przekracza pojemność), co pokazano na rysunku 21 (strona 47). Fakt 3.16, który poniżej przy-

Rysunek 21: Powstawanie nadmiaru po prawej stronie grafu.

taczamy, zapewnia, że podgraf ten jest koszący, przy czym założenie wymagane przez ten fakt wynika ze sposobu przyporządkowania pęczkom pojemności.

**Fakt 3.16** *W grafie  $G_{LR}$  mamy macierz bloków zawierającą bloki  $Q_1, Q_2, \dots, Q_k$ . Zbiory lewo i prawo-stronne bloku  $Q_i$  oznaczamy przez  $A_i, B_i$ . Rozłączamy bloki macierzy. Przyjmujemy, że dla wszystkich  $v \in A_i : f_{Q_i}(v) = f_G(v)$ , nie czynimy założeń co do pojemności wierzchołków z  $B_i$ . W blokach  $Q_i$  znajdujemy  $f$ -skojarzenia  $M_i$ , koszące na  $A_i$ . Oznaczmy  $M := \sum_i M_i$ . Jeśli  $v \in B_i$ , to niech  $r(v)$  oznacza jego wierzchołek "macierzysty", czyli wierzchołek grafu  $G_{LR}$  zawierający pęczek  $v$ .*

*Zakładamy, że jeśli dowolny  $v \in B_i$  jest nasycony względem  $M_i$  w  $Q_i$ , to  $r(v)$  jest nasycony (lub ma nadmiar) względem  $M$  w  $G_{LR}$ . Przy tym założeniu  $M$  jest podgrafem koszącym na  $L$  w  $G_{LR}$ .*

W punkcie 4 procedury usuwamy nadmiar na zbiorze  $R$  w taki sposób, że każdy wierzchołek  $R$  pozostawia przynajmniej jedną krawędź we wszystkich incydentnych do siebie blokach. Ta operacja jest wykonalna dzięki założeniu (15), gdyż  $\log^2 q$  to właśnie liczba wszystkich bloków macierzy bloków. Musimy udowodnić, że  $f$ -skojarzenie, jakie otrzymujemy po usunięciu nadmiaru, jest koszące na zbiorze  $L$ .

Oznaczmy zbiór krawędzi, które zniknęły z podgrafu w punkcie 4 procedury, przez  $U$ . Powróćmy do momentu, w którym procedura BLOK wykonuje obliczenia w rozłączonych blokach. Dokładniej, jesteśmy między punktami 2 i 3 procedury BLOK, gdy nadmiar po lewej stronie każdego bloku został już usunięty, ale nie zrobiono tego po prawej stronie. Możemy wymusić zniknięcie krawędzi ze zbioru  $U$  obniżając odpowiednio pojemności wierzchołków po prawej stronie każdego bloku a następnie, gdy procedura BLOK usuwa nadmiar po prawej stronie, usuwając właśnie krawędzie zbioru  $U$ , tak jak to pokazano na rysunku 22 (strona 48).

Na podstawie uwagi do Twierdzenia 3.14, w każdym bloku (z obniżonymi pojemnościami) mamy  $f$ -skojarzenie koszące. Przypomnijmy, że pojemność można obniżać

Rysunek 22: Usuwanie nadmiaru po prawej stronie grafu.

tylko do 1, zatem w każdym pęczku musimy pozostawić przynajmniej jedną krawędź, co jest przyczyną założenia (15). W wyniku "obniżania pojemności" pojawi się więcej nasyconych pęczków, jednak ich "macierzyste" wierzchołki w grafie dwudzielnym będą nasycone, tak więc założenie Faktu 3.16 jest spełnione. Tak więc, na podstawie tegoż faktu wiemy, że po złączeniu bloków dostaniemy  $f$ -skojarzenie koszące na  $L$ .

**Uwaga :** Procedura KOSZĄCEDWUDZIELNE wymaga założenia:

$$\forall v \in L : f(v) \leq d(v)$$

gdyż wymaga tego sposób znajdowania  $f$ -spannera w bloku. Tymczasem z definicji modyfikacji wynika, że stopień wierzchołka może spaść poniżej pojemności. Aby uporać się z tym problemem należy procedurę KOSZĄCEDWUDZIELNE zaopatrzyć w "prolog" i "epilog".

Niech  $X$ ,  $X \subseteq L$  oznacza zbiór wierzchołków dla których  $f(v) > d(v)$ . Prolog polega na tym, że wierzchołkom ze zbioru  $X$  obniża się pojemności do poziomu stopnia. Zauważmy jednak, że  $f$ -skojarzenie koszące "z obniżonymi pojemnościami" może nie być koszące po przywróceniu pierwotnych pojemności, dlatego w epilogu dokonuje się czegoś w rodzaju "uszlachetniania  $f$ -skojarzenia": sąsiedzi zbioru  $X$  (czyli wierzchołki w odległości 1 od tego zbioru) dodają do  $f$ -skojarzenia krawędzie wchodzące do  $X$ , w jak największej liczbie. Wszyscy sąsiedzi mogą to zrobić równocześnie, gdyż na zbiorze  $X$  nie ma żadnych ograniczeń co do stopnia  $f$ -skojarzenia. W efekcie, po modyfikacji, stopnie wierzchołków z  $X$  będą równe 0, tak więc "uszlachetnione"  $f$ -skojarzenie na pewno będzie koszące.

▽

A oto fakt, którego dwukrotnie używaliśmy w powyższym twierdzeniu:

**Fakt 3.16** *W grafie  $G_{LR}$  mamy macierz bloków zawierającą bloki  $Q_1, Q_2, \dots, Q_k$ . Zbiory lewo i prawo-stronne bloku  $Q_i$  oznaczamy przez  $A_i, B_i$ . Rozłączamy bloki macierzy. Przyjmujemy, że dla wszystkich  $v \in A_i : f_{Q_i}(v) = f_G(v)$ , nie czynimy założeń co do pojemności wierzchołków z  $B_i$ . W blokach  $Q_i$  znajdujemy  $f$ -skojarzenia  $M_i$ , koszące na  $A_i$ . Oznaczmy  $M := \sum_i M_i$ . Jeśli  $v \in B_i$ , to niech  $r(v)$  oznacza jego wierzchołek "macierzysty", czyli wierzchołek grafu  $G_{LR}$  zawierający pęczek  $v$ .*

*Zakładamy, że jeśli dowolny  $v \in B_i$  jest nasycony względem  $M_i$  w  $Q_i$ , to  $r(v)$  jest nasycony (lub ma nadmiar) względem  $M$  w  $G_{LR}$ . Przy tym założeniu  $M$  jest podgrafem koszącym na  $L$  w  $G_{LR}$ .*

**Dowód** Jeśli spełnione jest założenie faktu, to z punktu widzenia zbioru  $L$ , "modyfikacja na złączonych blokach" i "modyfikacja na rozłączonych blokach" (prawie) niczym się nie różnią, a to z kolei oznacza, że wartość funkcji  $\Phi^L$  zmniejszają się (prawie) tak samo, w obu przypadkach. Jest jasne, że  $M$  jest koszący na  $L$ , gdy bloki są rozłączone, jest więc koszący na  $L$  także przy złączonych blokach.

▽



### 3.5 Znajdowanie PMfM w grafie.

Teraz omówimy zestaw procedur służących do znajdowania PMfM, tj  $f$ -skojarzenia  $Q$  spełniającego warunek:

$$\forall v : i(v) \geq 2(f(v) - d_Q(v)) - s(v)$$

gdzie  $i(v)$  oznacza liczbę nasyconych sąsiadów wierzchołka  $v$ ,  
 $s(v) := \max\{0, 2f(v) - d(v)\}$ .

Okazuje się, że bardzo łatwo znaleźć PMfM w grafie dwudzielnym: potrafimy tego dokonać w stałym czasie, przy pomocy opisanej niżej procedury CZĘŚCIOWEDWUDZIELNE. Aby znaleźć PMfM w dowolnym grafie  $G$ , dzielimy go na rozłączne krawędziowo, dwudzielne podgrafy  $G_1, G_2, \dots, G_{\log n}$  w następujący sposób:

$$\begin{aligned} G_1 &= G_{V_0, V_1} \\ G_2 &= G_{V_{00}, V_{01}} \cup G_{V_{10}, V_{11}} \\ G_3 &= G_{V_{000}, V_{001}} \cup G_{V_{010}, V_{011}} \cup G_{V_{100}, V_{101}} \cup G_{V_{110}, V_{111}} \\ \dots &= \dots \end{aligned}$$

gdzie  $V_b$  oznacza zbiór wierzchołków, których identyfikatory mają prefiks  $b$  ( $b$  jest ciągiem bitów), natomiast  $G_{A,B}$  oznacza podgraf dwudzielny składający się ze wszystkich krawędzi między zbiorami  $A$  i  $B$ . Jak widać,  $G_{k+1}$  powstaje z  $G_k$  poprzez zastąpienie każdego wyrażenia postaci  $G_{V_a, V_b}$  przez  $G_{V_{a0}, V_{a1}} \cup G_{V_{b0}, V_{b1}}$ .

Przypuśćmy, że mamy PMfM  $X$  w  $G_1 \cup \dots \cup G_{k-1}$ . Okazuje się, że jak to zostanie udowodnione w Twierdzeniu 3.17, jeśli wykonamy modyfikację względem  $X$ , następnie znajdziemy PMfM  $Y$  w  $G_k$ , to wtedy  $X \cup Y$  będzie PMfM-em w  $G_1 \cup \dots \cup G_k$ . Na tym spostrzeżeniu opiera się procedura CZĘŚCIOWE. Poniżej przedstawiono formalny zapis procedury CZĘŚCIOWE, znajdującej PMfM w dowolnym grafie. Gdy mówimy, że pewna procedura jest uruchamiana na  $G_k$ , to w rzeczywistości obliczenia prowadzimy nie na  $G_k$ , lecz na tym co zostało z  $G_k$ , po uprzednio przeprowadzonych modyfikacjach.

#### PROCEDURA CZĘŚCIOWE

1. Niech  $X := \emptyset$ . Dla  $k := 1, \dots, \log n$  wykonaj:
  - (a) Znajdź PMfM  $Y$  na  $G_k$ , przy pomocy procedury CZĘŚCIOWEDWUDZIELNE.
  - (b) Wykonaj modyfikację grafu względem podgrafu  $Y$ .
  - (c) Niech  $X := X \cup Y$ .
2.  $X$  jest wynikiem działania tej procedury.

**Twierdzenie 3.17** *Procedura CZĘŚCIOWE znajduje PMfM, w dowolnym grafie, w czasie  $O(\log n)$ .*

**Dowód** Pętla wykonuje  $O(\log n)$  iteracji, w których wywoływana jest działająca w stałym czasie procedura CZĘŚCIOWEDWUDZIELNE, tak więc procedura CZĘŚCIOWE działa w czasie  $O(\log n)$ .

Udowodnimy teraz, że w punkcie 2 procedury, podgraf  $X$  jest PMfM-em w grafie  $G$ . Wystarczy udowodnić, że na końcu  $k$ -tej iteracji pętli podgraf  $X$  jest PMfM-em w grafie

$G_1 \cup \dots \cup G_k$ . Zakładamy, że na początku  $k$ -tej iteracji  $X$  jest PMfM-em w  $G_1 \cup \dots \cup G_{k-1}$ . Musimy więc udowodnić, że  $X \cup Y$  jest PMfM-em w  $(G_1 \cup \dots \cup G_{k-1}) \cup G_k$ .

Udowodnimy to w nieco ogólniejszym sformułowaniu: zakładamy, że mamy dwa rozłączne krawędziowo podgrafy  $G_1$  i  $G_2$ . Znajdujemy PMfM  $M_1$  w  $G_1$ , wykonujemy modyfikację grafu względem  $M_1$ , a następnie znajdujemy PMfM  $M_2$  w  $G_2$ . Udowodnimy, że  $M_1 \cup M_2$  jest PMfM-em w  $G_1 \cup G_2$ . Koncentrujemy się na jednym wierzchołku  $v$ , należącym do obu podgrafów, dlatego w dalszej części będziemy opuszczać " $(v)$ ". Wprowadzimy teraz oznaczenia, które pokazano także na rysunku 23 (strona 50): niech

Rysunek 23: Akumulowanie PMfM-ów w rozłącznych krawędziowo podgrafach.

$M := M_1 \cup M_2$ , niech  $f$  i  $d$  oznaczają początkową pojemność i stopień wierzchołka  $v$  w podgrafie  $G_1 \cup G_2$ , niech  $f_1$  i  $d_1$  oznaczają początkową pojemność i stopień  $v$  w podgrafie  $G_1$ , oraz niech  $f_2$  i  $d_2$  oznaczają pojemność i stopień  $v$  w podgrafie  $G_2$ , po modyfikacji względem  $M_1$  (przy czym modyfikację wykonuje się w całym grafie wejściowym). Zmienna  $x$  oznacza liczbę nasyconych względem  $M_1$  sąsiadów wierzchołka  $v$  w podgrafie  $G_2$ . Wiemy, że:  $d_1 + x + d_2 = d$ ,  $f_1 = f$ ,  $f_2 = f_1 - d_{M_1}$ .

Dzięki temu, że  $M_1$  i  $M_2$  są PMfM-ami, wiemy że:

$$i_k \geq j_k := 2(f_k - d_{M_k}) - s_k, \quad k = 1, 2 \quad (16)$$

gdzie  $s_k = \max\{0, 2f_k - d_k\}$ , a  $i_k$  oznacza liczbę nasyconych (względem  $M_k$ ) sąsiadów wierzchołka  $v$  w podgrafie  $G_k$ .

Aby udowodnić, że  $M$  jest PMfM-em w  $G_1 \cup G_2$ , musimy pokazać że:

$$i \geq 2(f - d_M) - s \quad (17)$$

gdzie  $s = \max\{0, 2f - d\}$ , a  $i$  oznacza liczbę nasyconych (względem  $M$ ) sąsiadów wierzchołka  $v$  w podgrafie  $G_1 \cup G_2$ .

Jest oczywiste, że:

$$i \geq \max\{0, i_1\} + x + \max\{0, i_2\}$$

tak więc, aby udowodnić (17), wystarczy pokazać, że:

$$\max\{0, j_1\} + x + \max\{0, j_2\} \geq 2(f - d_M) - s \quad (18)$$

Zauważmy na początku, że:

$$j_k \geq 0 \iff 2d_{M_k} \leq d_k, \quad k = 1, 2$$

Rozważmy następujące przypadki:

1.  $j_1 \geq 0, j_2 \geq 0$

Musimy udowodnić, że  $j_1 + x + j_2 \geq 2(f - d_M) - s$ , co jest równoważne:

$$2f_2 + x + s \geq s_1 + s_2$$

Zauważmy, że  $s \neq 0$  implikuje  $s_1 \neq 0$ , musimy więc rozważyć następujące podprzypadki:

- (a)  $s \neq 0, s_1 \neq 0, s_2 \neq 0$   
 $2f_2 + x + s \geq s_1 + s_2$  jest równoważne  $x + d_1 + d_2 \geq d$ , co jest prawdą.
- (b)  $s \neq 0, s_1 \neq 0, s_2 = 0$   
 $s \neq 0$  implikuje  $2f > d$ ,  $s_2 = 0$  implikuje  $2f_2 \leq d_2$ , ponadto wiemy że  $d_1 + d_2 \leq d$ , stąd:

$$d_1 + 2f_2 \leq d_1 + d_2 \leq d < 2f = 2f_1$$

$$d_1 < 2(f_1 - f_2) = 2d_{M_1}$$

Jednak  $j_1 \geq 0$  implikuje  $d_1 \geq 2d_{M_1}$ , czyli mamy sprzeczność.

- (c)  $s = 0, s_1 \neq 0, s_2 \neq 0$   
 $2f_2 + x + s \geq s_1 + s_2$  jest równoważne  $x + d_1 + d_2 \geq 2f_1$ , co jest prawdą, gdyż  $s = 0$  implikuje  $d \geq 2f_1$ , oraz  $d = x + d_1 + d_2$ .
- (d)  $s = 0, s_1 = 0, s_2 \neq 0$   
 $2f_2 + x + s \geq s_1 + s_2$  jest równoważne  $x + d_2 \geq 0$ , co oczywiście jest prawdą.
- (e)  $s = 0, s_1 \neq 0, s_2 = 0$   
 $2f_2 + x + s \geq s_1 + s_2$  jest równoważne  $x + d_1 \geq 2d_{M_1}$ , co jest prawdą, gdyż  $j_1 \geq 0$  implikuje  $d_1 \geq 2d_{M_1}$ .
- (f) Przypadek  $s = 0, s_1 = 0, s_2 = 0$  jest trywialny.

## 2. $j_1 < 0, j_2 \geq 0$

Należy udowodnić, że  $x + j_2 \geq 2(f - d_M) - s$  co jest równoważne:

$$x + s \geq s_2$$

Musimy rozważyć następujące podprzypadki:

- (a)  $s \neq 0, s_2 = 0$  lub  $s = 0, s_2 = 0$   
Trywialne.
- (b)  $s = 0, s_2 \neq 0$   
Musimy udowodnić, że  $x \geq 2f_2 - d_2$ .  
 $s = 0$  implikuje  $2f_1 \leq d$ ,  $j_1 < 0$  implikuje  $d_1 - 2d_{M_1} < 0$ , zatem:

$$2f_2 - d_2 = 2f_1 - 2d_{M_1} - d_2 \leq d - 2d_{M_1} - d_2 = x + (d_1 - 2d_{M_1}) \leq x$$

- (c)  $s \neq 0, s_2 \neq 0$   
 $j_1 < 0$  implikuje  $2d_{M_1} > d_1$ . Musimy udowodnić, że  $x + (2f - d) \geq 2f_2 - d_2$ , co jest równoważne  $2d_{M_1} \geq d_1$ , a to wynika z założenia  $j_1 < 0$ .

## 3. $j_1 \geq 0, j_2 < 0$

Należy udowodnić, że  $j_1 + x \geq 2(f - d_M) - s$ , co jest równoważne:

$$x + 2d_{M_2} + s \geq s_1$$

Musimy rozważyć podprzypadki:

(a)  $s_1 = 0$

Trywialne.

(b)  $s_1 \neq 0, s \neq 0$

Musimy udowodnić, że  $x + 2d_{M_2} + (2f - d) \geq (2f_1 - d_1)$ , czyli  $2d_{M_2} \geq d - d_1 - x = d_2$ , co jest prawdą, gdyż  $j_2 < 0$  implikuje  $2d_{M_2} > d_2$ .

(c)  $s_1 \neq 0, s = 0$

Musimy udowodnić, że  $x + 2d_{M_2} \geq 2f_1 - d_1$ . Wiemy, że  $j_2 < 0$  implikuje  $2d_{M_2} > d_2$ , tak więc wystarczy udowodnić, że  $x + d_2 \geq 2f_1 - d_1$ , czyli  $d \geq 2f$ , co wynika z założenia  $s = 0$ .

4.  $j_1 < 0, j_2 < 0$

Należy udowodnić, że:

$$x \geq 2(f - d_M) - s$$

Rozważmy następujące podprzypadki:

(a)  $s = 0$

Musimy udowodnić, że  $x \geq 2(f - d_M)$ . Z założenia  $j_1 < 0$  i  $j_2 < 0$  wynika  $2d_M > d_1 + d_2$ ,  $s = 0$  implikuje  $2f_1 \leq d$ , tak więc:

$$2(f - d_M) = 2f_1 - 2d_M < d - d_1 - d_2 = x$$

(b)  $s \neq 0$

Musimy udowodnić, że  $x \geq 2(f - d_M) - (2f - d)$  czyli  $x \geq d - 2d_M$ .

Wiemy, że  $d - d_1 - d_2 = x$ , zatem  $x \geq d - 2d_M$  jest równoważne  $2d_M \geq d_1 + d_2$ , co wynika z założenia  $j_1 < 0$  i  $j_2 < 0$ .

▽

Teraz przedstawimy formalny zapis procedury CZĘŚCIOWEDWUDZIELNE, znajdującej PMfM w grafie dwudzielnym  $G_{LR}$ :

#### PROCEDURA CZĘŚCIOWEDWUDZIELNE

1. Znajdź PMfM na zbiorze  $R$ , przy pomocy procedury CZĘŚCIOWEJEDNOSTRONNE. Oznacz je przez  $M_1$ .
2. Wykonaj modyfikację grafu względem  $M_1$ .
3. Znajdź PMfM na zbiorze  $L$ , przy pomocy procedury CZĘŚCIOWEJEDNOSTRONNE. Oznacz je przez  $M_2$ .
4.  $M_1 \cup M_2$  jest wynikiem działania tej procedury.

**Twierdzenie 3.18** *Procedura CZĘŚCIOWEDWUDZIELNE znajduje PMfM w  $G_{LR}$ , w stałym czasie.*

**Dowód** Procedura CZĘŚCIOWEDWUDZIELNE znajduje PMfM na zbiorze  $R$  (które oznaczamy przez  $M_1$ ), wykonuje modyfikację względem  $M_1$ , a następnie znajduje PMfM na zbiorze  $L$  (które oznaczamy przez  $M_2$ ). Jest jasne, że zakumulowane  $f$ -skojarzenie (czyli  $M_1 \cup M_2$ ) jest PMfM-em na zbiorze  $R$ , gdyż dodawanie krawędzi do podgrafu nie zmniejsza liczby nasyconych sąsiadów dowolnego wierzchołka z  $R$ . Wątpliwość budzi jedynie, czy jest ono także PMfM-em na zbiorze  $L$ .

Koncentrujemy się na dowolnym wierzchołku  $v \in L$ , dlatego w dalszej części będziemy opuszczać ”( $v$ )”. Wprowadzamy oznaczenia, które pokazano także na rysunku 24 (strona 53): niech  $f$  i  $d$  oznaczają pojemność i stopień wierzchołka  $v$  w punkcie 1 procedury. Po

Rysunek 24: Akumulowanie PMfM-ów ograniczonych do rozłącznych zbiorów wierzchołków.

modyfikacji względem  $M_1$ , pojemność i stopień wierzchołka  $v$  będziemy oznaczać przez  $f_2$  i  $d_2$ . Niech  $M := M_1 \cup M_2$ . Dla wierzchołka  $v$  zachodzi nierówność:

$$i_2 \geq j_2 := 2(f_2 - d_{M_2}) - s_2$$

gdzie  $s_2 = \max\{0, 2f_2 - d_2\}$ , a  $i_2$  oznacza liczbę nasyconych (względem  $M_2$ ) sąsiadów wierzchołka  $v$ . Chcemy udowodnić, że:

$$i \geq j := 2(f - d_M) - s$$

gdzie  $s = \max\{0, 2f - d\}$ , a  $i$  oznacza liczbę nasyconych (względem  $M$ ) sąsiadów naszego wierzchołka. Wprowadźmy dodatkowe oznaczenia: niech  $a$  oznacza liczbę sąsiadów nasyconych względem  $M_1$ , ale bez krawędzi  $M_1$  łączącej z  $v$ ; niech  $b := d_{M_1}$ . Jest oczywiste, że  $f_2 = f - b$ ,  $d_{M_2} = d_M - b$ ,  $d_2 = d - (a + b)$ ,  $i \geq a + i_2 \geq a + j_2$ .

Aby udowodnić, że  $i \geq j$  wystarczy pokazać, że  $a + j_2 \geq j$ , co jest równoważne:

$$a \geq s_2 - s$$

Rozważmy następujące przypadki:

1.  $s_2 \neq 0$ ,  $s = 0$

$$s_2 - s = 2f_2 - d_2 = (2f - d) + (a - b).$$

Założenie  $s = 0$  implikuje  $2f - d \leq 0$ , stąd  $s_2 - s \leq a - b \leq a$ .

2.  $s_2 \neq 0$ ,  $s \neq 0$

$$s_2 - s = (2f_2 - d_2) - (2f - d) = a - b \leq a.$$

3. Pozostałe przypadki są oczywiste.

▽

Pozostała do omówienia najbardziej elementarna spośród procedur służących do znajdowania PMfM, procedura CZĘŚCIOWEJEDNOSTRONNE, znajdująca PMfM ograniczone do zbioru  $B$ , w grafie dwudzielnym  $G_{AB}$ . W procedurze tej, dwukrotnie, wierzchołki zbioru  $B$  proponują krawędzie (wierzchołek  $v$  proponuje  $f(v)$  krawędzi), a następnie

wierzchołki zbioru  $A$  usuwają nadmiarowe krawędzie spośród zaproponowanych. Zauważmy, że procedura ta przypomina naiwną, i oczywiście nieskuteczną, metodę znajdowania MfM.

#### PROCEDURA CZĘŚCIOWEJEDNOSTRONNE

1. Powtórz 2 razy:
  - (a) Niech każdy wierzchołek ze zbioru  $B$  zaznaczy  $f(v)$  incydentnych do siebie krawędzi grafu. Zaznaczone krawędzie tworzą pewien podgraf.
  - (b) Niech wierzchołki ze zbioru  $A$  usuną nadmiarowe krawędzie podgrafu. Otrzymasz wtedy  $f$ -skojarzenie.
  - (c) Wykonaj modyfikację względem znalezionej  $f$ -skojarzenia.
2. Zakumulowane  $f$ -skojarzenie, czyli suma  $f$ -skojarzeń z obu iteracji, jest wynikiem działania tej procedury.

**Twierdzenie 3.19** *Procedura CZĘŚCIOWEJEDNOSTRONNE znajduje PMfM, ograniczone do zbioru  $B$ , w stałym czasie.*

**Dowód** Chcemy udowodnić, że dla dowolnego wierzchołka  $v \in B$  spełnione jest:

$$i(v) \geq 2(f(v) - d_M(v)) - s(v) \quad (19)$$

gdzie  $M$  oznacza  $f$ -skojarzenie będące wynikiem działania CZĘŚCIOWEJEDNOSTRONNE,  $f(v)$  i  $d(v)$  oznaczają początkową pojemność i stopień,  $s(v) := \max\{0, 2f(v) - d(v)\}$ ,  $i(v)$  oznacza liczbę nasyconych względem  $M$  sąsiadów wierzchołka  $v$ . Zauważmy, że w przypadku gdy  $d_M(v) = f(v)$ , nierówność (19) na pewno jest spełniona.

Procedura wykonuje dwie iteracje. Na rysunku 25 (strona 54) pokazano tylko jeden wierzchołek  $v \in B$ , w obu iteracjach. W przypadku drugiej iteracji pokazano dwa przypadki, jakie mogą się pojawić (i są niżej rozważane). Grubą linią narysowano krawędzie

Rysunek 25: Znajdowanie PMfM-u ograniczonego do prawej strony grafu.

zaznaczone przez wierzchołek  $v$ , zwane dalej "grubymi krawędziami". Gruba linia ze skreśleniem to krawędzie odrzucone przez wierzchołki ze zbioru  $A$ . Literą  $u$  oznaczono wierzchołki, które będą usunięte na końcu iteracji, gdyż są nasycone. Zmienna  $b$  oznacza liczbę nasyconych sąsiadów wierzchołka  $v$ , ale bez incydentnej grubej krawędzi, na końcu pierwszej iteracji. Zmienna  $x$  to początkowa pojemność wierzchołka  $v$ . Zmienna  $y$  oznacza liczbę sąsiadów, którzy skreślili grubą krawędź w pierwszej iteracji. Zmienna  $z$  to liczba sąsiadów, którzy skreślili grubą krawędź w drugiej iteracji. Zauważmy, że na początku pierwszej iteracji mamy  $f(v) = x$  i  $d(v) = a + b + x$ , natomiast na początku drugiej iteracji mamy  $f(v) = y$  i  $d(v) = a$ . Od tego momentu opuszczamy w zapisie symbol wierzchołka " $v$ ".

Rozważmy dwa przypadki:

1.  $a \geq y$

Wiemy, że  $i \geq y + z$ ,  $d_M = (x - y) + (y - z) = x - z$ . Aby udowodnić (19), wystarczy pokazać że  $y + z \geq 2(x - (x - z))$ , co jest równoważne  $y \geq z$ .

2.  $a < y$

Wiemy, że  $i \geq y + b + z$ ,  $d_M = (x - y) + (a - z)$ . Aby udowodnić (19), wystarczy pokazać, że  $y + b + z \geq 2(x - (x - y + a - z)) - s$ , czyli:

$$2a + b \geq y + z - s$$

Rozważmy następujące dwa podprzypadki:

(a)  $s = 0$

Wiemy, że  $a + x \geq y + z$ , gdyż  $a \geq z$ ,  $x \geq y$ , a ponadto  $s = 0$  implikuje  $a + b \geq x$ , tak więc  $2a + b \geq y + z$ .

(b)  $s \neq 0$

Jeśli  $s \neq 0$  to  $s = x - a - b$ . Chcemy udowodnić, że  $2a + b \geq y + z - (x - a - b)$ , co jest równoważne  $a + x \geq y + z$ , co jest prawdą.

▽

## Bibliografia

- [CV86] R. Cole, U. Vishkin.  
*Deterministic coin tossing with applications to optimal parallel list ranking.*  
Information and Control, 1986, 70, pp. 32-53.
- [IS86] A. Israeli, Y. Shiloach.  
*An improved algorithm for maximal matching.*  
Information Processing Letters, 1986, 22(2), pp. 57-60.
- [Pa] A. Panconesi.  
*Lecture Notes in Distributed Algorithms, Solution Sheet # 4.*  
<http://www.nada.kth.se/kurser/kth/2D5340>.
- [L92] N. Linial.  
*Locality in distributed graph algorithms.*  
SIAM Journal on Computing, 1992, 21(1), pp. 193-201.
- [PS96] A. Panconesi, A. Srinivasan.  
*On the Complexity of Distributed Network Decomposition.*  
Journal of Algorithms, 1996, 20(2), pp. 356-374.
- [MP00] G. De Marco, A. Pelc.  
*Fast distributed graph coloring with  $O(\Delta)$  colors.*  
2000, preprint.
- [LS93] N. Linial, M. Saks.  
*Decomposing Graphs into Regions of Small Diameter.*  
Proceedings of SODA 91, the Second Annual ACM-SIAM Symposium on Discrete Algorithms, San Francisco, pp. 320-330.
- [GP97] D.A. Grable, A. Panconesi.  
*Nearly optimal distributed edge colouring in  $O(\log \log n)$  rounds.*  
Random Structures and Algorithms, 1997, 10(3), pp. 385-405.
- [HKP98] M. Hańćkowiak, M. Karoński, A. Panconesi.  
*On the distributed complexity of computing maximal matchings.*  
Proceedings of SODA 98, the Ninth Annual ACM-SIAM Symposium on Discrete Algorithms, San Francisco, pp. 219-225.
- [HKP99] M. Hańćkowiak, M. Karoński, A. Panconesi.  
*A faster distributed algorithm for computing maximal matching deterministically.*  
Proceedings of PODC 99, the Eighteen Annual ACM SIGACT-SIGOPS Symposium on Principles of Distributed Computing, pp. 219-228.
- [HKP01] M. Hańćkowiak, M. Karoński, A. Panconesi.  
*On the distributed complexity of computing maximal matchings.*  
SIAM Journal of Discrete Mathematics, 2001, przyjęte do druku.