

SPIS TREŚCI

1. WSTĘP	2
2. PODSTAWY TEORETYCZNE.....	3
2.1. Teoria grafów	3
2.2. Izomorfizm	5
2.3. Formy kanoniczne grafów	6
2.4. Kolorowanie grafu	7
3. TWIERDZENIE RAMSEYA.....	9
3.1. Historia liczby i twierdzenia Ramseya	9
3.1.1. Twierdzenie Van der Waerden's.....	9
3.1.2. Paul Erdős i teoria Ramseya	10
3.2. Wartości liczb Ramseya	10
4. GENEROWANIE GRAFÓW.....	15
4.1. Rozszerzanie grafu	15
4.2. Generowanie grafów nieizomorficznych	16
4.3. Generowanie grafów Ramesyowskich	17
5. SKLEJANIE GRAFÓW	19
5.1. Dekompozycja problemu	19
5.2. Grafy potrzebne do sklejania	19
5.3. Algorytm sklejania	20
5.4. Zawężanie przedziałów - zasady A-D	21
5.5. Zakończenie algorytmu sklejania	26
6. IMPLEMENTACJA I EKSPERYMENTY	30

1. WSTĘP

Nasza praca opiera się na publikacji $R(4,5) = 25$ wydanej w 1995 przez Brendana D. McKaya oraz Stanisława P. Radziszowskiego [1]. Motywacją do sporządzenia tej pracy było udowodnienie, z wykorzystaniem nowych technologii, że dokładna wartość liczby Ramseya $R(4,5)$ wynosi 25. Motywacją do sporządzenia tej pracy było udowodnienie, że dokładna wartość liczby Ramseya $R(4,5)$ wynosi 25, z wykorzystaniem nowych technologii. Prace nad wyznaczeniem wartości $R(4,5)$ zaczęły się w 1955 wraz z wydaniem przez Greenwooda oraz Gleasona artykułu w którym wyznaczyli oni górną granicę $R(4,5) \leq 31$. W kolejnych latach granica ta była zawężana aż do $25 \leq R(4,5) \leq 27$.

Jak pokazano wcześniej, wygenerowanie wszystkich możliwych dwukolorowych grafów a następnie sprawdzenie ich poprawności byłoby zbyt czasochłonne więc wymagane było inne podejście do problemu. Wykorzystano jedynie wyselekcjonowane grafy (s,t,n) gdzie s oznacza rozmiar maksymalnej klikli która znajduje się w grafie, t oznacza wielkość maksymalnego zbioru niezależnego który należy do grafu, a n oznacza liczbę wierzchołków na których zbudowany jest graf. Celem było skonstruowanie rodziny grafów $R(4,5,24)$ z grafów $R(3,5,d)$ oraz $R(4,4,24-d)$ gdzie $7 \leq d \leq 13$, przy pomocy algorytmu nazwanym "sklejaniem". Ostatnim krokiem przed weryfikacją wyników jest próba rozszerzenia otrzymanych grafów $R(4,5,24)$ o jeden wierzchołek.

W naszej pracy omawiamy techniki i algorytmy służące do przeprowadzenia powyższego dowodu, oraz opisujemy dokonaną implementację.

2. PODSTAWY TEORETYCZNE

Przed omówieniem tematu naszej pracy, należy przedstawić kilka pojęć z teorii grafów, bez których zrozumienia, nie jest możliwe wyznaczanie liczb Ramsey'a.

2.1. Teoria grafów

Definicja 2.1.1 (Graf nieskierowany). Graf nieskierowany to para $G = (V, E)$, gdzie V to zbiór elementów nazywanych wierzchołkami, a E to zbiór par wierzchołków, którego elementy są nazywane krawędziami.

Definicja 2.1.2 (Graf prosty). Graf prosty, to taki graf $G = (V, E)$, gdzie zbiór E nie zawiera duplikatów, ani par (v, v) , dla $v \in V$.

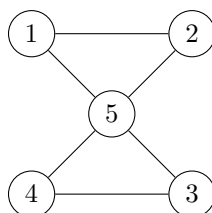
Dla potrzeb naszej pracy, będziemy zakładać, że gdy mówimy o grafie, mamy na myśli graf nieskierowany i prosty.

Definicja 2.1.3 (Rząd grafu). Liczbę elementów zbioru V w grafie $G = (V, E)$ nazywamy rzędem grafu.

Rząd grafu to liczba jego wierzchołków.

Definicja 2.1.4 (Sąsiedztwo). W grafie $G = (V, E)$ wierzchołki $u, v \in V$ sąsiadują wtedy i tylko wtedy gdy $(u, v) \in E$.

W grafach nieskierowanych sąsiedztwo jest relacją symetryczną. Jest nieprzechodnia - z sąsiedztwa u i v oraz v i w nie wynika sąsiedztwo u i w (Patrz rysunek 2.1).

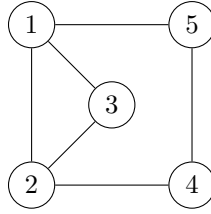


Rysunek 2.1. Graf, w którym wierzchołek 1 sąsiaduje z 2 i 5; 2 z 1 i 5; 3 z 4 i 5; 4 z 3 i 5; 5 z 1, 2, 3 i 4

Definicja 2.1.5 (Klika). Klika K w grafie $G = (V, E)$ jest takim podzbiorem wierzchołków $V(G)$, że dla każdej pary wierzchołków $u, v \in K$ zachodzi

$$(u, v) \in E(G)$$

W uproszczeniu, klika to podzbiór wierzchołków grafu, z których każdy jest połączony z każdym innym wierzchołkiem tego podzbioru. Przykład grafu zawierającego klikę można zobaczyć na rysunku 2.2



Rysunek 2.2. Wierzchołki 1, 2, 3 tworzą klikę stopnia 3 - K_3

Znalezienie stopnia maksymalnej kliky w danym grafie jest trudne obliczeniowo. W ogólności jest to problem rozwiązywany w czasie wielomianowym (choć dla niektórych grup grafów, takich jak grafy planarne, istnieją algorytmy wielomianowe), ale sprawdzenie czy w grafie istnieje klika z góry znanego stopnia, jest łatwe. Przykładowo istnienie K_3 możemy sprawdzić następującym algorytmem:

Algorithm 1 Sprawdzenie czy graf zawiera K_3

```

Require:  $G(V, E)$ 
for all  $v \in V$  do
  for all  $u \in V$  do
    if  $v \neq u \wedge v \in \text{sasiedzi}(u) \wedge \text{sasiedzi}(v) \cap \text{sasiedzi}(u) \neq \emptyset$  then

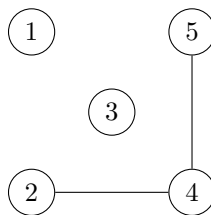
      return jest  $K_3$ 
    end if
  end for
end for

return nie ma  $K_3$ 

```

Definicja 2.1.6 (Zbiór niezależny). Zbiór niezależny N w grafie $G = (V, E)$ to taki podzbiór wierzchołków grafu G , że dla każdej pary wierzchołków $u, v \in N$ zachodzi

$$(u, v) \notin E(V)$$

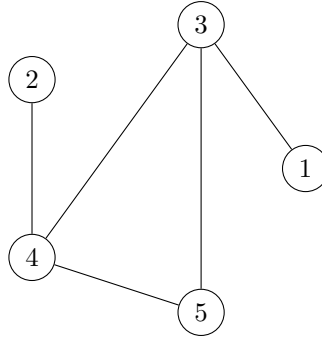


Rysunek 2.3. Wierzchołki 1, 2, 3, 5 tworzą zbiór niezależny 4 - N_4

Zbiór niezależny można również zdefiniować jako przeciwieństwo kliky, lub jako klikę w dopełnieniu grafu. Rysunek 2.3 prezentuje przykład zbioru niezależnego.

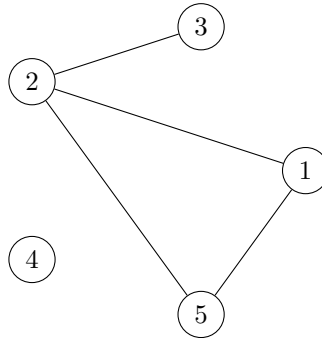
Definicja 2.1.7 (Dopełnienie grafu). Dopełnieniem grafu $G = (V, E)$ nazywamy taki graf $G' = (V, E')$, dla którego zachodzi

$$\forall_{u,v} (u, v) \in E \iff (u, v) \notin E'$$



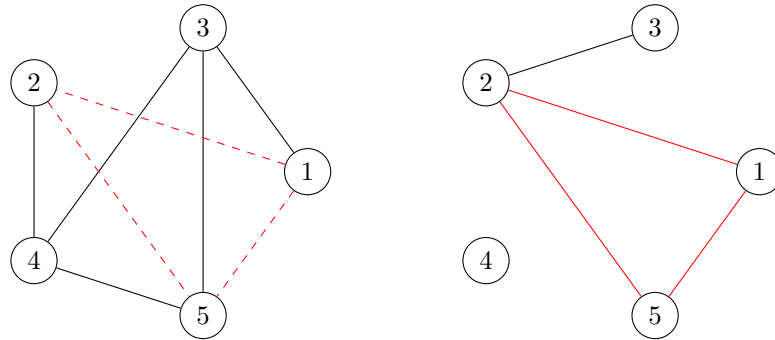
Rysunek 2.4. Graf G

Na rysunku 2.4 trudno zauważyć czy graf G posiada zbiór niezależny N_3 . Spójrzmy na jego dopełnienie (rysunek 2.5):



Rysunek 2.5. Dopełnienie grafu G - graf G'

Natychmiast zauważamy, że graf G' posiada klikę K_3 : Oznacza to, że G ma N_3 na tych samych wierzchołkach (rysunek 2.6).



Rysunek 2.6. K_3 w G' leży między tymi samymi Wierzchołkami co N_3 w G

Definicja 2.1.8 (Podgraf indukowany). Dla grafu $G = (V, E)$ i podzbioru jego wierzchołków $S \subseteq V$, podgrafem indukowanym $G[S]$ nazywamy taki graf, którego wszystkie wierzchołki zawierają się w S i którego zbiór krawędzi zawiera wszystkie krawędzie z E kończące się w S .

$$G[S] = H(S, E_2), \forall (v, u) \in E, (v, u) \in E_2 \iff v \in S \wedge u \in S$$

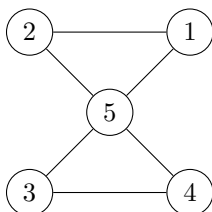
Na potrzeby naszej pracy mówiąc o podgrafie mamy na myśli podgraf indukowany.

2.2. Izomorfizm

Definicja 2.2.1 (Izomorfizm grafów). Izomorfizm grafów G i H to taka bijekcja $f : V(G) \rightarrow V(H)$, że 2 wierzchołki u i v sąsiadują w G tylko jeżeli $f(u)$ i $f(v)$ sąsiadują w H .

Innymi słowy, izomorfizm jest to takie przekształcenie grafu, które zachowuje wszystkie jego właściwości. Grafy, dla których można znaleźć izomorfizm są izomorficzne. Innymi słowy, dwa grafy są izomorficzne, jeżeli ich wierzchołki można nazwać w taki sposób, aby odpowiadające sobie wierzchołki w obu grafach miały dokładnie takie same zbiory sąsiadów.

Przekształćmy graf z rysunku 2.1 zamieniając wierzchołek 1 z 2 oraz 3 z 4:



Rysunek 2.7. Mimo przekształcenia graf zachował swoje własności

Jak widać na rysunku 2.7, wierzchołki o tych samych numerach zachowały swoje zbiory sąsiadów.

Wszystkie własności opisane w tej pracy są przez izomorfizmy niezmienione, a więc na nasz użytek dwa grafy izomorficzne to efektywnie ten sam graf.

Definicja 2.2.2 (Automorfizm grafu). Automorfizm grafu G to taka permutacja f zbioru wierzchołków G , że para wierzchołków u, v jest sąsiadująca tylko jeżeli $f(u)$ i $f(v)$ sąsiadują.

Automorfizm to izomorfizm z grafu początkowego do tego samego grafu. Złożenie automorfizmów również jest Automorfizmem.

Definicja 2.2.3 (Grupa automorfizmu). Grupa automorfizmów grafu G to zbiór składający się z wszystkich automorfizmów grafu G .

2.3. Formy kanoniczne grafów

Definicja 2.3.1 (Forma kanoniczna). Forma kanoniczna $C(G)$ grafu G to taki graf izomorficzny do G , że każdy graf izomorficzny do G ma taką samą formę kanoniczną[14].

$$Canon: G \rightarrow G / \simeq$$

$$G \simeq H \iff Canon(G) = Canon(H)$$

Forma kanoniczna grafu to takie pokrycie grafu G , które jest identyczne dla wszystkich grafów izomorficznych z G . Przykładowo, takie pokrycie można zaimplementować jako numerowanie wierzchołków poczynawszy od takich o najmniejszej liczbie sąsiadów. Jeżeli kilka wierzchołków ma taką samą liczbę sąsiadów to albo niższy numer otrzymuje wierzchołek, który ma sąsiada (sąsiadów) z najniższym numerem albo losowy.

Algorithm 2 Sprowadzenie grafu do formy kanonicznej

Require: $G(V, E)$

$tab = \text{sort}(V, \text{iloscSasiadowRekurencyjnie})$

for all $v \in V(G)$ **do**

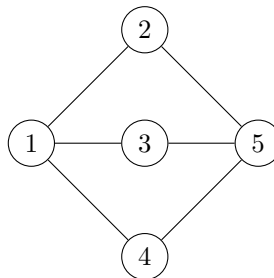
$v.numer = tab.indeks(v)$

end for

Powyższy algorytm poprawnie tworzy formy kanoniczne jedynie dla bardzo prostych grafów. Rzeczywiste algorytmy są dużo bardziej zoptymalizowane i Korzystają z heurystyk pozwalających zoptymalizować proces dla określonych przypadków[14]. Sprowadzanie do formy kanonicznej jest kluczowe dla szybkiego wyszukiwania izomorfizmów.

Definicja 2.3.2 (Orbita). Orbita w grafie G to taki podzbiór jego wierzchołków, których wszystkie permutacje są izomorfizmami.

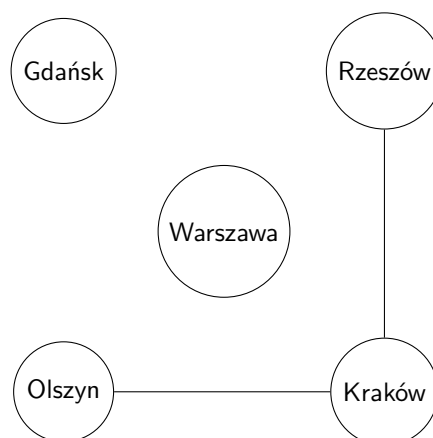
Wszystkie wierzchołki wewnątrz orbity mogą zostać dowolnie zamienione, nie zmieniając właściwości grafu. Ich zamiana jest automorfizmem. Przykład (rysunek 2.8):



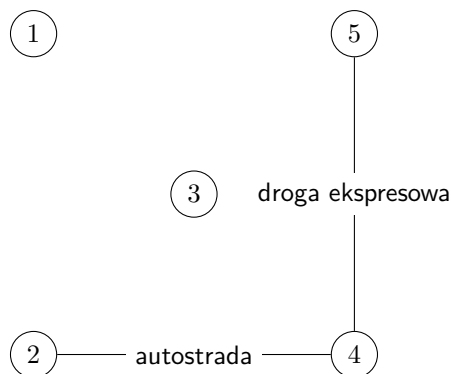
Rysunek 2.8. Orbita pierwsza: 1, 5; Orbita druga: 2, 3, 4

2.4. Kolorowanie grafu

Definicja 2.4.1 (Oznakowanie grafu). Dla grafu $G = (V, E)$, wierzchołkowym oznakowaniem grafu nazywamy funkcję mapującą wierzchołki do oznaczeń (rysunek 2.9). Analogicznie, funkcję mapującą krawędzie do oznaczeń nazywamy krawędziowym oznakowaniem grafu (rysunek 2.10).



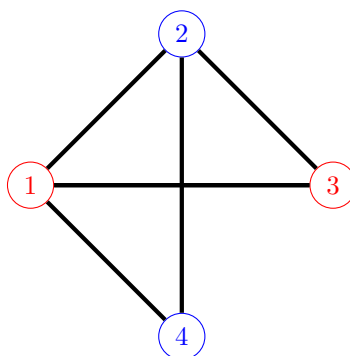
Rysunek 2.9. Przykład wierzchołkowego oznakowania grafu G z rysunku



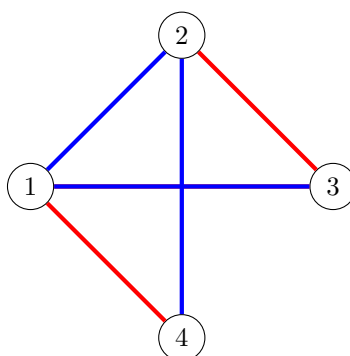
Rysunek 2.10. Przykład krawędziowego oznakowania grafu G

Oznakowywanie elementów grafów ma wiele zastosowań. Często nadaje się krawędziom wartości mające reprezentować np. koszt pokonania drogi między dwoma punktami. Innym często wykorzystywanym oznakowaniem grafów jest kolorowanie.

Definicja 2.4.2 (Kolorowanie grafu). Kolorowanie grafów jest szczególnym przypadkiem oznakowywania grafu w którym oznaczenia są nazywane kolorami i mogą być przypisywane tylko zgodnie z określonymi ograniczeniami (rysunek 2.11 i 2.12).



Rysunek 2.11. Przykładowe kolorowanie wierzchołkowe grafu



Rysunek 2.12. Przykładowe kolorowanie krawędziowe grafu

Kolorowanie najczęściej idzie w parze z ograniczeniami, których należy przestrzegać przypisując elementom wartości. Najczęściej dotyczą one zakazu kolorowania sąsiadujących wierzchołków lub krawędzi tym samym kolorem.

Innym ograniczeniem dla kolorowania może być zakaz pomalowania klikli ustalonego stopnia n - K_n - tym samym kolorem. Dla $n = 2$ otrzymamy zakaz sąsiedztwa wspomniany wcześniej.

3. TWIERDZENIE RAMSEYA

Twierdzenie Ramseya mówi o konieczności pojawienia się pewnych układów w pozornym chaosie, co oznacza że każda większa struktura będzie zawierała jakąś podstrukturę. Zagadnienie można łatwo przedstawić posługując się teorią grafów, dla uproszczenia zostanie użyte kolorowanie dwoma kolorami.

Twierdzenie 1. Niech $r \in \mathbb{N}$. Istnieje takie $n \in \mathbb{N}$ gdzie dla każdego 2-kolorowego K_n grafu znajdzie się jednokolorowy podgraf K_r w K_n . [6]

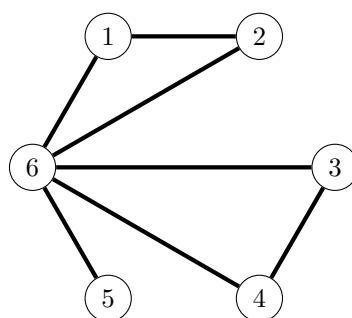
Z powyższego twierdzenia wynika, że będziemy omawiać dwukolorowe struktury.

Definicja 3.0.1 (Liczba Ramseya). Niech $r \in \mathbb{N}$ i $b \in \mathbb{N}$. Liczba Ramseya, wyrażana jako $n = R(r, b)$, jest najmniejszą liczbą całkowitą taką że 2-kolorowy graf K_n , używając koloru czerwonego i niebieskiego do kolorowania krawędzi, implikuje czerwony monochromatyczny podgraf K_r lub niebieski monochromatyczny podgraf K_b [6]

W celu ułatwienia obliczeń i umożliwienia użycia technik generacji grafów, używamy uproszczenia reprezentacji grafowej. Zamiast wyznaczać kolorowania grafu pełnego, używamy wszystkich grafów prostych gdzie kolor krawędzi zamieniony jest na istnienie lub brak istnienia krawędzi pomiędzy parą wierzchołków.

Definicja 3.0.2 (Graf Ramseyowski). Niech $r \in \mathbb{N}$, $b \in \mathbb{N}$ i $n \in \mathbb{N}$. Graf Ramseyowski, zapisany jako $R(r, b, n)$ oznacza graf mający n wierzchołków, nie zawierający klikki o rozmiarze r i nie zawierający zbioru niezależnego rozmiaru b . [6]

W kontekście tej pracy, określamy graf jako ramseyowski jeżeli nie posiada on kolorowania określonego w definicji 3.0.2. Oznacza to, że graf spełniający $R(4, 5, 6)$ to graf zbudowany na 6 wierzchołkach, który nie posiada klikki 4 rzędu ani zbioru niezależnego 5 rzędu. Przykładowy graf $R(4, 5, 6)$ jest przedstawiony na rysunku 3.1



Rysunek 3.1

Celem tej pracy jest wykazanie, że stworzenie grafu $R(4, 5, 25)$ jest niemożliwe, a istnieją grafy $R(4, 5, 24)$, z czego wynika, że liczba Ramseya $R(4, 5) = 25$.

3.1. Historia liczby i twierdzenia Ramseya

W 1930 roku zostało opublikowane dzieło Franka Plumptona Ramseya "On a Problem of Formal Logic"[10], które posłużyło jako podstawę do teorii którą dzisiaj znamy jak Teoria Ramseya.

3.1.1. Twierdzenie Van der Waerden's

Twierdzenie opublikowane przez Van der Waerdena w 1927 roku, przed powstaniem Twierdzenia Ramseya lecz uważana za jedną z jego gałęzi.

Twierdzenie 2. Dla dowolnych liczb naturalnych r oraz k istnieje taka liczba naturalna n która określa zbiór $\{1, 2, 3, \dots, n\}$ który jest pokolorowany na r różnych kolorów, z przynajmniej k liczbami naturalnymi w ciągu arytmetycznym które są tego samego koloru.[6]

Dla przykładu, $W(2, 3) = 9$. Zbiór o ośmiu elementach można podzielić na następujące podzbiory aby w żadnym z podzbiorów nie wystąpił szereg arytmetyczny o trzech elementach, $\{1, 2, 3, 4, 5, 6, 7, 8\} \rightarrow \{1, 2, 5, 6\}, \{3, 4, 7, 8\}$. W przypadku zbioru o dziewięciu elementach taki podział nie jest możliwy. Dodając 9 do dowolnego podzbioru utworzony zostanie ciąg arytmetyczny o trzech elementach np. $\{1, 5, 9\}$ lub $\{7, 8, 9\}$. Podobna sytuacja zajdzie dla podziału $\{1, 4, 5, 8\}, \{2, 3, 6, 7\}$

3.1.2. Paul Erdős i teoria Ramseya

Happy Ending problem, czyli problem zaprezentowany przez Paula Erdősa w 1933 roku brzmi następująco.

Twierdzenie 3. Dowolny zbiór 5 punktów w przestrzeni, w pozycja ogólna, zawiera podzbiór 4 punktów które formują wielokąt wypukły.

Twierdzenie to zostało uogólnione w 1935 roku przez George Szekeresa oraz Paula Erdősa

Twierdzenie 4. Dla dowolnej liczby całkowitej N , każdy dowolnie duży i skończony zbiór punktów w przestrzeni, w pozycji ogólnej, zawiera podzbiór składający się z N punktów który tworzy wielokąt wypukły. [11]

Prace nad Happy Ending problem sprawiły że Paul Erdős natrafił na publikacje Ramseya z 1928 roku. Spowodowało to że Erdős rozpoczął prace nad liczbami Ramseya, co przyczyniło się do rozwoju tej teorii.

Party problem lub inaczej Theorem on Friends and Strangers, jest to problem dzięki któremu można przedstawić przykład zastosowania liczby Ramseya. Brzmi on następująco: jaka jest najmniejsza liczba osób jaką trzeba zaprosić na przyjęcie tak aby trójka z nich były wspólnymi znajomymi lub trójka z nich była dla siebie nieznanymi[15]? Aby odpowiedzieć na to pytanie można zasięgnąć o pomoc do teorii Ramseya gdzie znajdziemy że odpowiedź na postawione pytanie stanowi liczba Ramseya $R(3, 3) = 6$ która jest opisana w dalszej części pracy. Powyższe pytanie można przedstawić w bardziej formalny sposób: Znajdź najmniejszą liczbę gości którzy muszą zostać zaproszeni tak aby przynajmniej m znało się wzajemnie a n była dla siebie obca [16], lub inaczej; Znajdź najmniejszą liczbę wierzchołków dla których graf będzie zawierać klikę stopnia n o kolorze czerwonym lub klikę stopnia m o kolorze niebieskim. Rozwiązaniem dla tego problemu są liczby Ramseya.

3.2. Wartości liczb Ramseya

1. $R(1, k) = R(k, 1) = 1$

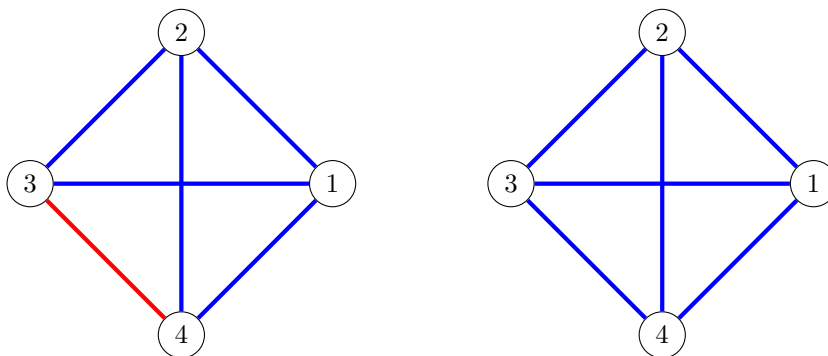
W przypadku gdy jeden z parametrów wynosi 1 aby spełnić warunek wystarczy jeden wierzchołek. Jednokolorowy graf K_1 jest pojedynczym wierzchołkiem i spełnia zarówno warunek dla $R(1, b)$ oraz $R(r, 1)$, patrz rysunek 3.2.

1

Rysunek 3.2. $R(1,k,1)$

2. $R(2,k) = R(k,2) = k$

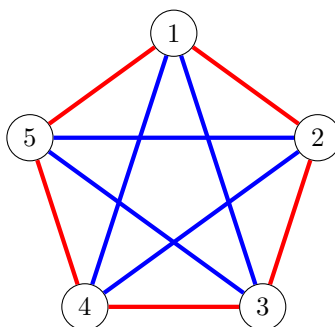
W przypadku gdy jeden z parametrów wynosi 2 nie możemy postąpić analogicznie jak w przypadku gdy $k=1$, gdyż graf K_2 nie spełni warunku gdy $k > 2$ dla kolorowania jednym kolorem. Tak samo każdy graf pełny o rozmiarze mniejszym niż k nie spełnia warunku w sytuacji gdy zostanie użyty kolor ograniczony liczbą k aby pokolorować go w jednolity sposób. Dlatego też liczba wierzchołków w grafie musi wynosić k co zawsze spełni jeden z dwóch warunków, pierwszy w przypadku, gdy wszystkie krawędzie zostaną pokolorowane jednym kolorem lub drugi, gdy chociaż jedna krawędź będzie drugiego koloru. Przykładowo dla $R(2,4)$ gdzie pierwszy kolor (dla $r=2$) będzie oznaczony kolorem czerwonym a drugi ($b=4$) niebieskim, patrz rysunek 3.3.



Rysunek 3.3. Przykłady grafów niespełniających $R(2,4,4)$

3. $R(3,3)=6$

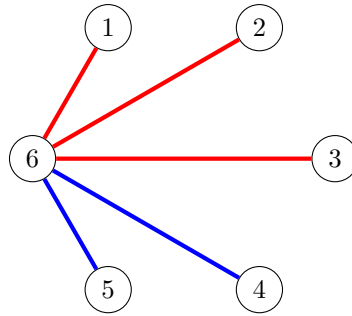
$R(3,3)$ jest pierwszym nietrywialnym przykładem liczby Ramsey'a, lecz nadal na tyle prostym aby łatwo móc ją wyznaczyć. Łatwo można wykluczyć K_3 , K_4 oraz K_5 za pomocą następującego pokolorowania krawędzi (rysunek 3.4).



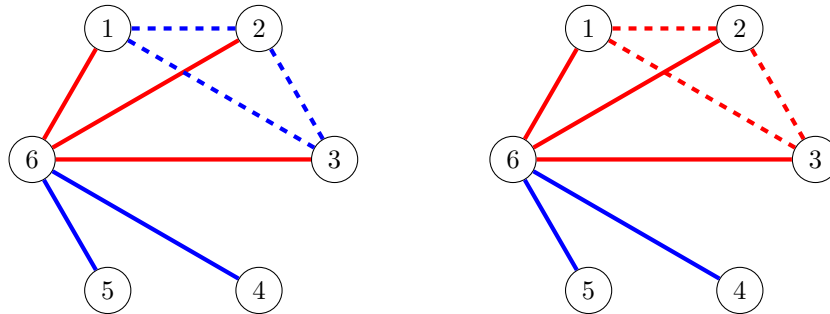
Rysunek 3.4. Graf $R(3,3,5)$

Powyższy rysunek pokazuje sposób kolorowania dla grafu 5-wierzchołkowego, ale wykluczenie dowolnego wierzchołka daje poprawne kolorowanie dla grafu 4-wierzchołkowego, dowolnych dwóch dla grafu 3-wierzchołkowego itd.

Aby udowodnić że $R(3,3) = 6$ przeanalizujmy kolorowanie grafu pełnego o 6 wierzchołkach.



Rysunek 3.5. Pokolorowane krawędzie wychodzące z jednego wierzchołka



Rysunek 3.6. Pokolorowania implikujące powstanie czerwonej lub niebieskiej K_3

Po wybraniu dowolnego wierzchołka i pokolorowaniu wychodzących z niego krawędzi co najmniej trzy z nich będą miały wspólny kolor. Na rysunku 3.5 tym kolorem jest kolor czerwony, krawędzie tego koloru połączone są z trzema innymi wierzchołkami. Patrząc na trzy wierzchołki do których zostały poprowadzone krawędzie czerwone, łatwo zauważyć, że aby uniknąć powstania trójkąta czerwonego należy połączyć te wierzchołki kolorem niebieskim, lecz robiąc to powstanie klika o rozmiarze trzy koloru niebieskiego (rysunku 3.6). Dowodzi to że $R(3, 3) = 6$.

4. Inne liczby Ramseya

Udowodnienie wartości pozostałych liczb Ramseya zostanie pominięte, gdyż: stopień skomplikowania dowodu rośnie wraz z ilością wierzchołków, nie istnieje żaden znany łatwy obliczeniowo sposób na określenie dokładnej wartości tej liczby, oraz wyznaczenie dokładnej wartości często jest na tyle trudne że istnieje jedynie jej bliższe oszacowanie. Tabela 3.7 prezentuje dokładne wartości, lub górne i dolne granice dla dwukolorowych liczb Ramseya $R(k, l)$ $k < 10$, $l < 10$ (wartości dla k i l równego 2 albo 1 zostały opisane wcześniej). Jako że wartości dla liczb Ramseya są symetryczne ($R(r, b) = R(b, r)$) wypełniony został jedynie górny trójkąt w tabeli.

$R(k, l)$	3	4	5	6	7	8	9	10
3	6	9	14	18	23	28	36	40 42
4		18	25	36 41	49 61	59 84	73 115	92 149
5			43 49	58 87	80 143	101 216	133 316	143 442
6				102 165	115 298	134 495	183 780	204 1171
7					205 540	217 1031	252 1713	292 2826
8						282 1870	329 3583	343 6090
9							565 6588	581 12677
10								798 23556

Rysunek 3.7. Wartości liczb ramseya dla $3 \leq k \leq 10$ i $3 \leq l \leq 10$. Górny wiersz odpowiada wartościom k a boczny wartościom l . Dwie liczby zapisane w jednej komórce oznaczają granicę dla danej liczby: najpierw zapisana jest dolna a następnie górna granica

Istnieją dwa główne podejścia na wyznaczanie liczb Ramseya. Pierwszym z nich, gdy wyznaczenie dokładnej liczby nie jest możliwe polega na wyznaczeniu górnej oraz dolnej granicy. Przykładową pracą gdzie udowodniane były granice dla liczb Ramseya ($R(5, 5) \leq 49$ oraz $R(4, 6) \leq 41$) jest praca Brendana D. McKaya oraz Stanisława P. Radziszowskiego "Subgraph Counting Identities and Ramsey Numbers" [17]. Drugim sposobem natomiast jest wyznaczenie dokładnej wartości tej liczby. W tym przypadku bardzo często wykorzystywane są komputery z zaprojektowanymi do tego celu algorytmami. Przykładowymi pracami gdzie ta metoda została wykorzystana jest praca na której bazujemy, autorstwa Brendana D. McKaya oraz Stanisława P. Radziszowskiego $R(4, 5) = 25$ [1] oraz praca Charlesa M. Grinstead i Sama M. Robertsa "On the Ramsey Numbers $R(3, 8)$ and $R(3, 9)$ " gdzie posłużyli się algorytmem do wyznaczenia $R(3, 9) = 36$ oraz ustalenia granic dla liczby $R(3, 8)$ ($28 \leq R(3, 8) \leq 29$)[18].

Powodem dla podawania granicy dolnej oraz górnej jest, jak wspomniano wcześniej, brak uniwersalnej i opłacalnej formuły do określenia dokładnej wartości. Dla grafu pełnego K_n , który ma $\frac{n(n-1)}{2}$ krawędzi, istnieje $c^{\frac{n(n-1)}{2}}$ grafów które trzeba przeszukać (gdzie c oznacza liczbę kolorów). Oznacza to że złożoność przeszukiwania wszystkich możliwych grafów metoda naiwną to $O(c^{n^2})$ przy c kolorach i n wierzchołkach. Przykładowo dla $R(4, 6)$ gdybyśmy chcieli sprawdzić dolną granicę 36 [3], należałoby sprawdzić wszystkie dwukolorowania K_{36} , który ma $\binom{36}{2} = 630$ krawędzi. Istnieje więc $2^{630} \approx 4,4555 \cdot 10^{189}$ różnych sposobów na pokolorowanie tego grafu. Dlatego przy obecnych możliwościach obliczeniowych nie jest możliwe rozwiązanie tego problemu używając podejścia naiwnego.

Aby podsumować problem znajdowania dokładnych wartości liczb Ramseya można posłużyć się słowami Paula Erdősa: Jeżeli kosmici najechaliby ziemię i postawili ultimatum, że jeżeli ludzkość nie znajdzie $R(5, 5)$ w ciągu roku, to zniszczą ziemię, najlepszym wyborem byłoby zebranie całej mocy obliczeniowej jaką aktualnie dysponuje ludzkość w celu pozyskania tej liczby. Jednak w przypadku gdy kosmici zażądali by $R(6, 6)$ najlepszym wyborem było by wypowiedzenie im wojny.[12]

5. Granice liczby Ramseya

Granica górna może być łatwo wyliczona stosując nierówność $R(r, b) \leq R(r-1, b) + R(r, b-1)$ [9]. Nie jest to jednak zadowalający wynik, ani zadowalający sposób na wyznaczanie górnego limitu. Poprzednie wartości liczb Ramseya mogą nie być znane oraz sama granica przy znanych wcześniejszych wartościach nie jest najbardziej optymalną. Wzór jawny który opisuje wcześniej podany przypadek to: $R(r, b) \leq \binom{r+b-2}{r-1}$. Przytoczona górna granica jest granicą naiwną. Granica dolna jest wyznaczana z użyciem metod probabilistycznych. Paul Erdős jako pierwszy w 1947 roku zaprezentował dowód z użyciem metod probabilistycznych na granicę dolną dla liczb $R(k, k)$ [8, 9]. Metoda ta opierała się na wykazaniu że w losowo pokolorowanym grafie K_n prawdopodobieństwo znalezienia jednokolorowego grafu K_k jest mniejsza od 1 dla pewnej wartości.

4. GENEROWANIE GRAFÓW

4.1. Rozszerzanie grafu

Do przeprowadzenia poprawnego rozumowania na grafach, wymagane jest wpięrow pozyskanie zbioru grafów. Operację tę nazywamy generowaniem grafów, i rozpoczyna się ją od rozszerzania grafów.

Definicja 4.1.1. Rozszerzeniem grafu nazywamy przekształcenie grafu G w graf G' poprzez dodanie do G wierzchołka v i dowolnego podzbioru możliwych krawędzi pomiędzy v a $V(G)$.

Możliwych rozszerzeń grafu o n wierzchołkach jest 2^n . Kompletnym rozszerzeniem nazywamy zbiór grafów uzyskany przez wykonanie każdego możliwego rozszerzenia.

Najprostszym przykładem rozszerzenia grafu jest rozszerzanie grafu jednowierzchołkowego, pokazanego na rys. 4.1.



Rysunek 4.1. Graf jednowierzchołkowy

Podczas rozszerzania powyższego grafu istnieją dwie możliwości na sposób dodania krawędzi pomiędzy Grafem G a nowym wierzchołkiem. W celu wygenerowania wszystkich grafów dwuwierzchołkowych należy przeprowadzić rozszerzenia na wszystkie (dwa) możliwe sposoby jak na rysunku 4.2.

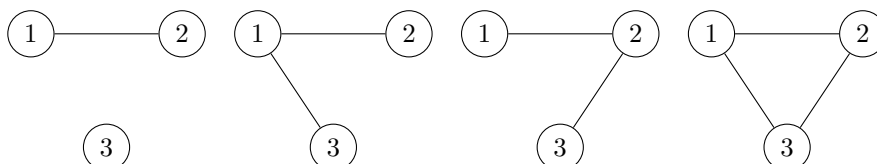


Rysunek 4.2. Oba możliwe rozszerzenia grafu jednowierzchołkowego

W podobny sposób można uzyskać kompletny zbiór grafów o dowolnej liczbie wierzchołków. Wystarczy jedynie rekurencyjnie rozszerzać zbiór grafów, zaczynając od jednego wierzchołka i kończąc, gdy nasz zbiór zawiera grafy o wymaganym rzędzie. Przez rozszerzenie zbioru grafów mamy na myśli rozszerzenie każdego z grafów ze zbioru na wszystkie możliwe sposoby.

Takie podejście prowadzi co prawda do uzyskania każdego możliwego grafu, jest jednak zatrażająco nieefektywne. Jest tak ze względu na obecność grup grafów izomorficznych w zbiorach generowanych w ten sposób.

W celu zobrazowania problemu przeprowadźmy dalsze rozszerzenie dwuwierzchołkowego grafu pełnego z rysunku 4.2. Określmy ten graf jako graf G . Rozszerzenie grafu dwuwierzchołkowego w sposób kompletny prowadzi do uzyskania 4 grafów, pokazanych na rysunku 4.3.



Rysunek 4.3. Możliwe rozszerzenia grafu G opisane dalej jako G'_1 do G'_4

Warto zauważyć, że powstały zbiór 4 grafów zawiera dwa grafy izomorficzne, G'_2 i G'_3 . Można więc powiedzieć, że zbiór G'_n zawiera trzy zbiory grafów izomorficznych. W przypadku grafów trójwierzchołkowych wygenerowanych przez rozszerzenie grafów na rysunku 4.3, uzyskamy 8 grafów, ale jedynie

4 grupy izomorficzne. Trywialnie więc zauważyć, że jeżeli do G'_n dodamy również rozszerzenia drugiego grafu z 4.3 przybędzie tylko jeden unikalny graf.

Ze względu na naturę problemu poruszanego w tej pracy, zbiór grafów izomorficznych nie wnosi więcej informacji, niż jeden jej przedstawiciel. Dzieje się tak głównie ze względu na fakt, że izomorfizm nie zmienia największego stopnia klikli ani zbioru niezależnego w grafie. Wyeliminowanie nadmiarowych grafów izomorficznych ze zbioru wynikowego wydaje się więc wysoko pożądane w celu zwiększenia wydajności generacji zarówno pod względem czasowym jak i pamięciowym. Żeby zobrazować wagę takiej optymalizacji wystarczy podać kilka pierwszych wartości dla ilości grafów wygenerowanych naiwną metodą rozszerzania w porównaniu do ilości grafów nieizomorficznych.

Rząd grafów	Ilość rozszerzeń[19]	Ilość grafów nieizomorficznych[13]
1	1	1
2	2	2
3	8	4
4	64	11
5	1024	34
6	32768	156
7	2097152	1044
8	268435456	12346

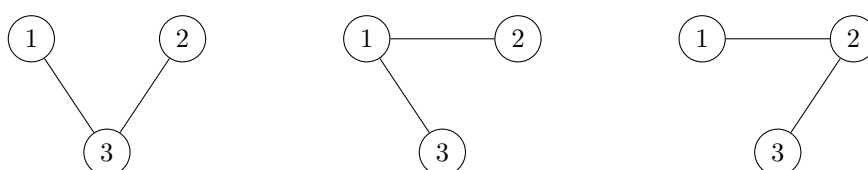
Rysunek 4.4. Tabela porównująca ilość możliwych rozszerzeń i ilość grafów nieizomorficznych w przypadku tworzenia kompletnego zbioru grafów określonego rzędu

4.2. Generowanie grafów nieizomorficznych

Użytecznym jest fakt, że na dowolnym etapie generacji zbiór grafów izomorficznych w zbiorze do rozszerzenia można bezproblemowo zredukować do tylko jednego z jego przedstawicieli. Jest tak ze względu na poniższe twierdzenie.

Twierdzenie 5. Jeżeli rozszerzamy izomorficzne grafy G oraz H do zbiorów G'_n oraz H'_m , to dla dowolnego elementu zbioru G'_n istnieje izomorfizm do elementu zbioru H'_m .

Dzięki powyższemu twierdzeniu, jeżeli na każdej iteracji generacji grafów będziemy usuwać nadmiarowe grafy izomorficzne to wynikowy zbiór grafów nie będzie zawierał grafów rozszerzonych z grafów już izomorficznych, ale nie straci żadnych istotnych dla nas grafów. Dzięki temu, pozostają dwie możliwości powstania zbiorów grafów izomorficznych do zredukowania: grafy izomorficzne powstałe z rozszerzania różnych grafów, oraz grafy izomorficzne powstałe z rozszerzania tego samego grafu. Obie te możliwości można eliminować, nie porównując grafów w stylu "każdy z każdym", co jest wysoko pożądane, jako że znajdowanie izomorfizmów w zbiorze grafów to problem nietrywialny, który mógłby znacznie spowolnić działanie generacji. Aby zobrazować metody eliminowania, na rysunku 4.5 jest pokazana grupa izomorfizmów wynikająca z rozszerzenia grafów trójkątnych.



Rysunek 4.5. Grupa grafów izomorficznych powstająca z rozszerzania grafów trójkątnych

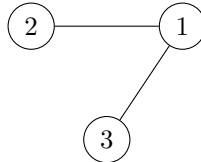
Powyższe grafy będą opisywane jako G_1 , G_2 i G_3 .

Każdy z tych grafów należy sprowadzić do formy kanonicznej a następnie wyznaczyć istniejące w nim orbity.

Przykładowym przekształceniem grafu G_2 do formy kanonicznej jest

$$\{1, 2, 3\} \rightarrow \{2, 1, 3\}$$

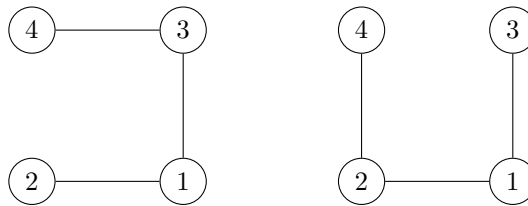
Efektom takiego przekształcenia jest graf przedstawiony na rysunku 4.6



Rysunek 4.6. Graf G_3 w formie kanonicznej

Orbitami tego grafu będą zbiory $\{1\}$ oraz $\{2,3\}$.

Dzięki temu wiemy, że w kolejnym kroku nie trzeba podłączać nowego wierzchołka do obu wierzchołków z orbity, wystarczy zrobić to tylko dla jednego wierzchołka z orbity $\{2,3\}$, co widać na rysunku 4.7.



Rysunek 4.7. Przykładowe izomorficzne rozszerzenia grafu G_3

Dlatego podczas rozszerzania grafu rozważamy nie pojedyncze wierzchołki, do których będziemy łączyć nowy wierzchołek, a całe orbity. Zmniejsza to ilość połączeń do rozważenia z 2^n do $n + 1$, gdzie n jest liczebnością orbity.

Drugą techniką użyteczną przy usuwaniu izomorfizmów jest sprowadzenie wszystkich wygenerowanych grafów danego stopnia do formy kanonicznej i sprawdzanie czy dodany wierzchołek znajduje się w ostatniej orbicie. Jeżeli tak nie jest, to graf można odrzucić, ponieważ graf do niego izomorficzny powstanie w innej gałęzi.

4.3. Generowanie grafów Ramesyowskich

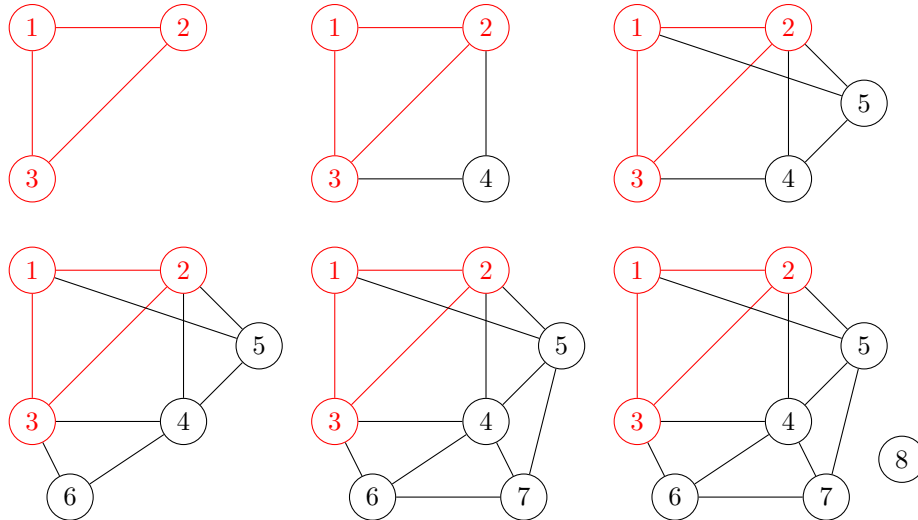
Generowanie grafów nieizomorficznych jest wciąż zbyt wymagające dla naszych potrzeb. Ilość grafów nieizomorficznych poszczególnych rzędów rośnie wykładniczo, i dla grafów rzędu 17, największego generowanego przez rozszerzanie w tej pracy, byłoby ich ponad $2 * 10^{26}$ [13]. Jasnym więc jest, że ilość grafów generowanych na poszczególnych etapach rozszerzania należy bardziej ograniczyć. Z pomocą przychodzi następujące twierdzenie:

Twierdzenie 6. Dla dowolnego grafu G posiadającego klikę K_i każdy graf H , który powstaje przez dowolną sekwencję rozszerzeń grafu G , zawiera klikę K_j gdzie $j \geq i$.

Powyższe twierdzenie wynika z następującego rozumowania:

Gdy graf G rozszerzymy do grafu G' , to graf G jest podgrafem grafu G' . Dzieje się tak, ponieważ rozszerzanie nie modyfikuje krawędzi w oryginalnym grafie, a jedynie dodaje krawędzie do nowego

wierzchołka. Przy pomocy indukcji matematycznej można więc powiedzieć, że dowolny graf H , który powstaje poprzez wielokrotne rozszerzanie grafu G musi zawierać graf G jako swój podgraf. Z tego powodu, graf H musi również zawierać klikę z oryginalnego grafu G , oraz może posiadać klikę większą powstałą w wyniku rozszerzania. Przykład zachowania oryginalnego grafu podczas rozszerzania jest pokazany na rysunku 4.8



Rysunek 4.8. Kolejne rozszerzenia grafu, które nie zaburzają oryginalnego grafu początkowego zaznaczonego kolorem czerwonym.

Identyczne rozumowanie można przeprowadzić dla zbiorów niezależnych w oryginalnym grafie, prowadząc do analogicznego twierdzenia dotyczącego zbiorów niezależnych.

Twierdzenie 7. Dla dowolnego grafu G posiadającego zbiór niezależny N_i każdy graf H , który powstaje przez dowolną sekwencję rozszerzeń grafu G , zawiera zbiór niezależny N_j gdzie $j \geq i$.

Twierdzenia te prowadzą do wniosku podobnego do tego o eliminacji grafów izomorficznych, gdzie jeżeli na każdym etapie algorytmu wyeliminujemy grafy łamiące ograniczenia dotyczące klik i zbioru niezależnego, to na dalszych etapach algorytmu nie powstaną takie i tylko takie grafy, które również łamałyby te ograniczenia. Dzięki temu można znacznie ograniczyć ilość grafów tworzonych na późniejszych etapach generacji. Omównione ograniczenia porwadzą do powstania algorytmu generacji grafów w następującej formie:

Algorithm 3 Generowanie grafów ramseyowskich o podanym stopniu

Require: $n > 0$

$i \leftarrow 1$

grafy[] \leftarrow graf jednowierzchołkowy

while $i < n$ **do**

 rozszerz(grafy)

 wyluczNieramseyowskie(grafy)

 wyluczIzomorfizmy(grafy)

$i \leftarrow i + 1$

end while

Kolejność operacji wyluczIzomorfizmy i wyluczNieramseyowskie nie ma znaczenia pod kątem poprawności - jednak dla grafów większych stopni operacja znajdowania orbit staje się znacznie bardziej

skomplikowana w porównaniu do relatywnie prostego poszukiwania klik i zbiorów niezależnych, przez co taka kolejność poprawia ogólną efektywność.

5. SKLEJANIE GRAFÓW

Generowanie grafów wysokiego rzędu jest bardzo wymagające zarówno czasowo jak i pamięciowo. Dlatego stosujemy technikę nazywaną sklejaniem grafów.

Definicja 5.0.1. Sklejanie grafów opiera się na połączeniu dwóch zbiorów grafów wejściowych w celu uzyskania zbioru grafów wyjściowych o pożądanych charakterystykach.

5.1. Dekompozycja problemu

Definicja 5.1.1. Niech F jest grafem, $v \in V(F)$, $W \subseteq V(F)$. Funkcja $N_F(v, W)$ będzie zwracać podzbiór wierzchołków W , które sąsiadują z v :

$$N_F(v, W) = \{w \in W \mid (v, w) \in E(F)\}$$

Definicja 5.1.2. $F[V(F) - v]$ będziemy zapisywać jako $F - v$.

Twierdzenie 8. Weźmy $x \in V(F)$. Niech podgraf indukowany $G_x = F[N_F(x, V(F))]$ oraz $H_x = F[V(F) - N_F(x, V(F) - x)]$. Jeśli F jest $R(4, 5, 25)$ to G_x jest $R(3, 5, d)$ a F_x jest $R(4, 4, 24 - d)$.

Powyższe twierdzenie jest nieoczywiste, dlatego przeprowadzimy krótki dowód.

Niech graf F zawiera przynajmniej jedną klikę K_3 oraz przynajmniej jeden zbiór niezależny N_4 . Zauważmy że dla każdej kilki K_3 zachodzi jedna z sytuacji:

1. $x \in K_3$
2. $\exists v \in K_3, (v, x) \in E(F) \wedge \nexists u \in K_3, (u, x) \in E(F)$
3. $\nexists v \in K_3, (v, x) \in E(F)$

W przypadku pierwszym klika zostanie rozbita, dwa jej wierzchołki będą w G_x , a x nie będzie w żadnym.

W przypadku drugim wierzchołki tworzące K_3 , które są połączone z x trafią do G_x , niepołączone trafią do H_x .

W przypadku trzecim całość K_3 trafi do grafu H_x , gdzie klika K_3 może się pojawić, nie łamiąc $R(4, 4, n)$.

Z kolei dla każdego zbioru N_4 musi zachodzić jedna z sytuacji:

1. $x \in N_4$
2. $\exists v \in N_4, (v, x) \in E(F) \wedge \nexists u \in N_4, (u, x) \in E(F)$
3. $\nexists v \in N_4, (v, x) \in E(F)$

W przypadku pierwszym zbiór niezależny zostanie rozbit, trzy jego wierzchołki będą w H_x , a x nie będzie w żadnym.

W przypadku drugim wierzchołki tworzące N_4 , które są połączone z x trafią do G_x , niepołączone trafią do H_x .

W przypadku trzecim całość N_4 trafi do grafu G_x , gdzie zbiór niezależny może się pojawić, nie łamiąc $R(3, 5, n)$.

5.2. Grafy potrzebne do sklejania

Korzystając z wiedzy z poprzedniego podrozdziału możemy stwierdzić, że do utworzenia wszystkich grafów $R(4, 5, 24)$ wystarczy połączenie na wszystkie możliwe sposoby wszystkich nieizomorficznych grafów $R(3, 5, d)$ z wszystkimi nieizomorficznymi grafami $R(4, 4, 24 - d)$. Warto zauważyć, że liczby $R(3, 5)$ oraz $R(4, 4)$ są znane i wynoszą odpowiednio 14 i 18. Oznacza to, że nasze grafy G będą maksymalnie rzędu 13 a grafy H rzędu 17. Co za tym idzie, możemy też wskazać ograniczenia dolne: minimalny rząd G to $24 - 17 = 7$ a minimalny rząd H to $24 - 13 = 11$.

Zbiór danych wymagany do poprawnego sklejania jest uzyskany poprzez wygenerowanie wszystkich nieizomorficznych grafów G rzędów 7-13 spełniających $R(3, 5, n)$, oraz wszystkich nieizomorficznych grafów H rzędów 11-17 spełniających $R(4, 4, n)$.

Grafy dzielone są na grupy zgodnie z ich rzędem. Dalsze sklejanie przeprowadzone jest na parze grup grafów G i H . Celem sklejania jest uzyskanie grafów rzędu 24 poprzez połączenie grafów G oraz H , a więc suma rzędów grafów G oraz H w grupach branych pod uwagę musi być równa 24. Oznacza to, że istnieje 6 par grup spełniających wymagania.

5.3. Algorytm sklejania

Na dalszych etapach sklejania pojawia się pojęcie stożka. Szczególnym przypadkiem stożka używanym na etapie poszukiwania połączeń grafów G oraz H jest stożek prawdopodobny.

Definicja 5.3.1. Stożek to podzbiór wierzchołków dowolnego grafu.

Definicja 5.3.2. Prawdopodobny stożek (ang. feasible cone) to podzbiór wierzchołków grafu H , który nie tworzy klik K_3 [1]. Symbolizuje on połączenia pomiędzy wierzchołkiem grafu G a wszystkimi wierzchołkami $V(H)$, które należą do stożka.

Podzbiór wierzchołków grafu H o którym jeszcze nie wiemy, czy jest prawdopodobnym stożkiem będziemy określać mianem stożka.

Dla przeciętnego grafu $R(4, 4, 14)$ znajdziemy około 4000 prawdopodobnych stożków. Jeżeli do każdego z wierzchołków grafu G przypiszemy stożek, uzyskamy takie połączenie grafów G oraz H gdzie każdy wierzchołek grafu G jest sąsiadem wszystkich wierzchołków w przypisanym mu stożku. Główna część algorytmu sklejania polega na eliminacji wszystkich kombinacji stożków, które prowadzą do powstania kilku rzędu 4 lub zbioru niezależnego rzędu 5. Każdy taki zbiór prawdopodobnych stożków zawiera poprawne połączenia grafów G oraz H . Następnie zbiory prawdopodobnych stożków są rozbijane, aby wyizolować jedynie akceptowalne grafy. Taki algorytm sklejania pozwala uzyskać grafy $R(4, 5, n)$ o różnych rzędach (poprzez inne parowanie grup grafów G i H), ale w tej pracy zajmujemy się jedynie grafami 24-wierzchołkowymi.

Pierwszym etapem sklejania grafów jest stworzenie dla danego grafu H zbioru prawdopodobnych stożków przed wybraniem grafu G do sklejania. Należy odrzucić wszystkie stożki które obejmują klikę trzeciego rzędu. Takie stożki eliminują wszystkie grafy wynikowe ponieważ połączenie klik trzeciego rzędu z dowolnym wierzchołkiem powoduje wystąpienie klik czwartego rzędu. Podejście naiwne, czyli pojedyncze sprawdzenie wszystkich potencjalnie prawdopodobnych stożków nie jest jednak opłacalne ze względu na że ilość stożków dla grafu n wierzchołkowego, która jest równa 2^n . W celu przyspieszenia obliczeń stożki grupowane są w przedziały.

Definicja 5.3.3. Przedział $P = [B, T]$ to zbiór stożków, ograniczony przez stożek górny T i stożek dolny B , który zawiera wszystkie stożki spełniające zależność:

$$X : B \subseteq X \subseteq T$$

Znajdowanie prawdopodobnych stożków rozpoczynamy od pojedynczego przedziału gdzie $B = \emptyset$, a T zawiera wszystkie wierzchołki H . Taki przedział zawiera wszystkie możliwe stożki. W celu wyodrębnienia prawdopodobnych stożków korzystamy z poniższych własności:

1. Dla wierzchołka w , który spełnia $w \notin B$ oraz $w \in T$ prawdą jest, że $[B, T] = [B + \{w\}, T] \cup [B, T - \{w\}]$ oraz $[B + \{w\}, T] \cap [B, T - \{w\}] = \emptyset$
2. Jeżeli ograniczenie przedziału B zawiera klikę stopnia 3, to wszystkie stożki w przedziale również ją zawierają.
3. Jeżeli ograniczenie przedziału T nie zawiera klikę stopnia 3, to wszystkie stożki w przedziale jej nie zawierają.

Korzystając z własności 2 i 3 jako warunku odpowiednio odrzucenia lub zaakceptowania przedziału można skorzystać z algorytmu rekurencyjnego dzielącego interwały wzdłuż wierzchołków należących do klikę 3 oraz stożka B , dzięki czemu uzyskujemy zbiór przedziałów dla grafu H zawierający jedynie stożki prawdopodobne. Z tego zbioru korzystamy w następnym etapie algorytmu. Dalsze odrzucanie stożków musi odbyć się już w kontekście grafu G .

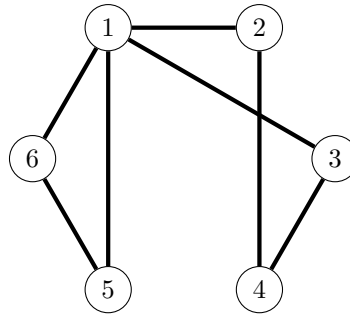
5.4. Zawężanie przedziałów - zasady A-D

Zdefiniujemy 3 funkcje pomocnicze działające na podzbiorze wierzchołków grafu H oznaczonym jako X , generujące podzbiór wierzchołków grafu H . Funkcja H_1 będzie wybierała wszystkich sąsiadów wierzchołków ze zbioru X . H_2 wybiera wszystkie wierzchołki, które nie sąsiadują z jednym lub większą ilością wierzchołków spoza X . H_3 wybiera wszystkie wierzchołki, które tworzą zbiór niezależny o stopniu 3 z dwoma wierzchołkami spoza zbioru X .

$$H_i : X \subseteq V(G) \rightarrow Y \subseteq V(G)$$

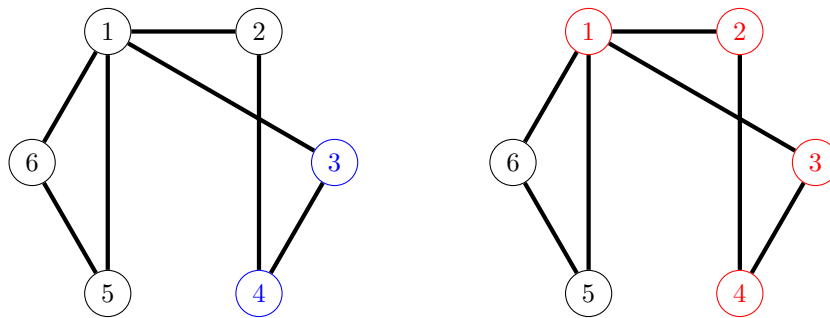
- $H_1(X) \rightarrow X = w \in V(H) | vw \in E(X)$ dla jakiegoś $v \in X$
- $H_2(X) \rightarrow X = w \in V(H) | vw \notin E(X)$ dla jakiegoś $v \notin X$
- $H_3(X) \rightarrow X = w \in V(H) | u, v, w$ jest zbiorem niezależnym dla jakichś $v, u \notin X$

Rozpatrzmy podany graf, rysunek 5.1. Dla tego grafu zostanie zaprezentowane działanie zasad H_1 , H_2 oraz H_3 . Na rysunkach 5.2, 5.3, 5.4, przedstawiony jest sposób działania tych funkcji. Pierwszy graf jest grafem, który jest przekazywany do funkcji, a wierzchołki zaznaczone na niebiesko oznaczają wierzchołki ze zbioru X . Drugi graf zawiera wierzchołki zaznaczone na czerwono, jest to podzbiór wierzchołków które zwraca funkcja.



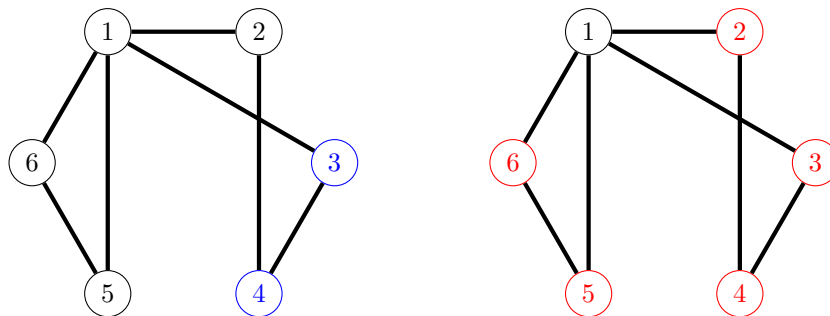
Rysunek 5.1. Graf służący do zaprezentowania poszczególnych funkcji H

Funkcja H_1 znajduje wszystkich sąsiadów wierzchołków ze zbioru wejściowego. W podanym przykładzie na rysunku 5.2 oznacza to, że musimy znaleźć sąsiadów wierzchołków ze zbioru $\{3, 4\}$. Łatwo zauważyć że wierzchołki z nimi sąsiadujące to zbiór wierzchołków $\{1, 2, 3, 4\}$.



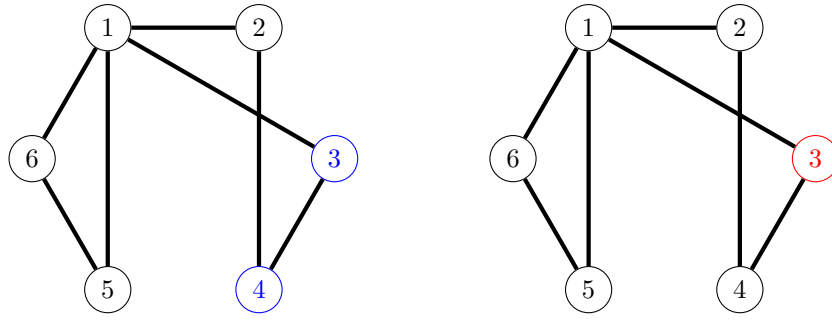
Rysunek 5.2. Wynik działania funkcji H_1

Funkcja H_2 znajduje wszystkie wierzchołki, które nie sąsiadują z co najmniej jednym wierzchołkiem spoza zbioru wejściowego. W podanym przykładzie na rysunku 5.3 oznacza to że musimy znaleźć wierzchołki które nie sąsiadują ze wszystkimi wierzchołkami ze zbioru (wykluczając wierzchołki który aktualnie rozpatrujemy, jeżeli jest w zbiorze) $\{1, 2, 5, 6\}$. Łatwo zauważyć że jedynym wierzchołkiem który sąsiaduje ze wszystkimi wierzchołkami z tego zbioru jest wierzchołek o numerze 1, więc zbiór wyjściowy będzie wyglądać następująco $\{2, 3, 4, 5, 6\}$.



Rysunek 5.3. Wynik działania funkcji H_2

Funkcja H_3 znajduje wszystkie wierzchołki, które tworzą zbiór niezależny o stopniu 3 z dwoma wierzchołkami spoza zbioru X . W podanym przykładzie na rysunku 5.4 oznacza to, że musimy znaleźć wierzchołki które tworzą zbiór niezależny wraz z dwoma wierzchołkami ze zbioru $\{1, 2, 5, 6\}$. Łatwo zauważyć że jedynym wierzchołkiem który spełnia to założenie jest wierzchołek oznaczony numerem 3, gdyż tworzy on zbiór niezależny stopnia 3 wraz z wierzchołkami o numerach 2 oraz 5.



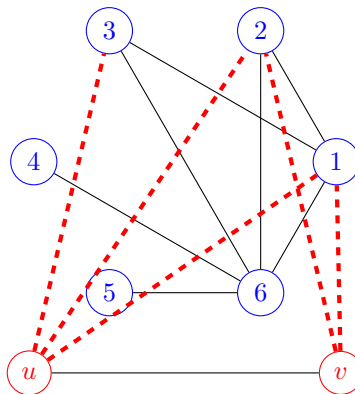
Rysunek 5.4. Wynik działania funkcji H_3

Kolejny etap zawężania zbioru potencjalnych krawędzi pomiędzy grafami G oraz H odbywa się już w kontekście konkretnego grafu G . Każda możliwa kombinacja wygenerowanych przedziałów jest przydzielana do wierzchołków należących do G , dzięki czemu rozważone są wszystkie możliwe sposoby połączenia tych grafów. Dla każdego zbioru przedziałów łączących grafy G oraz H wykonywany jest poniższy zbiór reguł działających na wierzchołki i przedziały stożków im przypisane:

- A - stosowana do 2 wierzchołków $u, v \in G$ sąsiadujących ze sobą
 Jeśli $B_u \cap B_v \cap H_1(B_u \cap B_v) \neq \emptyset$ to nie da się poprawnie skleić tej pary grafów.
 W innym wypadku z T_u usuwamy $H_1(B_u \cap B_v) \cap B_v$
- B - stosowana do 2 wierzchołków $u, v \in G$ nie sąsiadujących ze sobą
 Jeśli $H_3(T_u \cup T_v) \not\subseteq (T_u \cup T_v)$ to nie da się poprawnie skleić tej pary grafów.
 W innym wypadku B_u rozszerzamy do $B_u \cup (H_3(T_u \cup T_v) - T_v)$
- C - stosowana do 3 wierzchołków $u, v, w \in G$ tworzących zbiór niezależny N_3
 Jeśli $H_2(T_u \cup T_v \cup T_w) \not\subseteq (T_u \cup T_v \cup T_w)$ to nie da się poprawnie skleić tej pary grafów.
 W innym wypadku B_u rozszerzamy do $B_u \cup (H_2(T_u \cup T_v \cup T_w) - (T_v \cup T_w))$
- D - stosowana do 4 wierzchołków $u, v, w, z \in G$ tworzących zbiór niezależny N_4
 Jeśli $T_u \cup T_v \cup T_w \cup T_z \neq VH$ to nie da się poprawnie skleić tej pary grafów.
 W innym wypadku B_u rozszerzamy do $B_u \cup (VH - (T_v \cup T_w \cup T_z))$

Zasada A sprawdza czy sąsiednie wierzchołki u, v mają jakąkolwiek parę wspólnych potencjalnych sąsiadów, którzy są sąsiedni względem siebie. W takiej sytuacji powstałaby klika K_4 , więc nie uda się utworzyć grafu.

Przyjrzyjmy się rysunkowi 5.5. Czerwone wierzchołki to u, v z grafu G , a czerwone, przerywane krawędzie reprezentują dolne ograniczenia przypisanych im stożków. Jak widać, uiv muszą zostać połączone z parą sąsiadujących wierzchołków 1 i 2, więc nie da się poprawnie połączyć grafów użyciem takiego przypisania stożków.

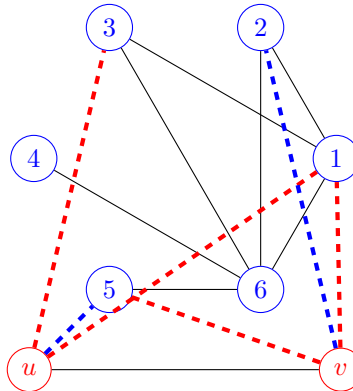


Rysunek 5.5. Przykład sytuacji odrzucanej przez zasadę A

Jeżeli powyższa sytuacja nie zajdzie, musimy usunąć z T_u niezbędnych sąsiadów v , którzy sąsiadują z przynajmniej jednym z niezbędnych sąsiadów u .

Na rysunku 5.6 wprowadzono ograniczenie górne, reprezentowane niebieskimi, przerywanymi krawędziami. Jak widać, to połączenie nie jest odrzucane (istnieje możliwość połączenia bez utworzenia K_4).

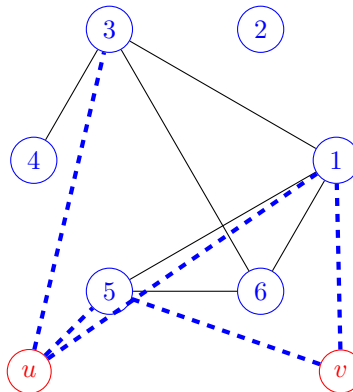
Trzeba więc odrzucić potencjalne połączenia pomiędzy v a niezbędnymi sąsiadami u . Na przykładowym rysunku będzie to oznaczało usunięcie krawędzi $(u, 5)$. Zgodnie z zasadą A nie usuwamy jeszcze krawędzi $(v, 2)$ - zostanie ona usunięta kiedy to v będzie pełniła rolę u w funkcji.



Rysunek 5.6. Przykład przedziału zawężanego przez zasadę A

Zasada B sprawdza czy istnieją w grafie H zbiory niezależne N_3 , z których wierzchołkami żaden z wierzchołków u, v nie może zostać połączony. W takim wypadku nie da się skleić grafów bez utworzenia zbioru niezależnego N_5 .

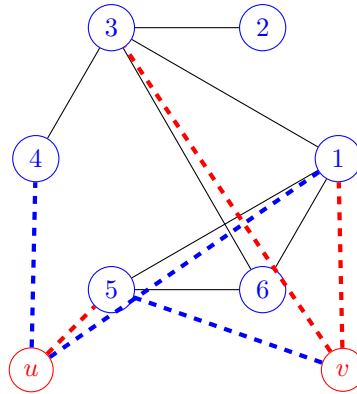
Dla przykładu, przeanalizujmy rysunek 5.7. Wierzchołki 2, 4, 6 tworzą zbiór niezależny i żaden z nich nie mieści się w ograniczeniach głównych wierzchołków u, v . W związku z tym, dana próba połączenia nie zakończy się sukcesem.



Rysunek 5.7. Przykład sytuacji odrzucanej przez zasadę B

Jeżeli powyższa sytuacja nie zachodzi, musimy do zbioru B_u dodać wszystkie wierzchołki, z którymi przynajmniej 2 wierzchołki spoza zbioru $T_u \cup T_v$ nie sąsiadują. Wyjątkiem są te, z którymi można połączyć v .

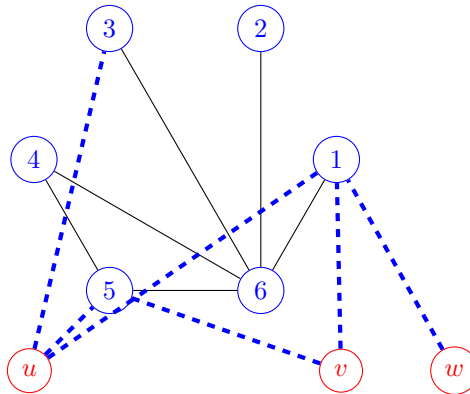
W przykładzie z rysunku 5.8 trzeba dodać wierzchołek 4 do dolnego ograniczenia u .



Rysunek 5.8. Przykład przedziału zawężanego przez zasadę B

Zasada C sprawdza czy istnieją w grafie H dwa niesąsiadujące wierzchołki, z którymi żaden z wierzchołków zbioru niezależnego u, v, w nie może zostać połączony. Oznaczałoby to, że musi powstać zbiór niezależny N_5 , a grafów nie można skleić.

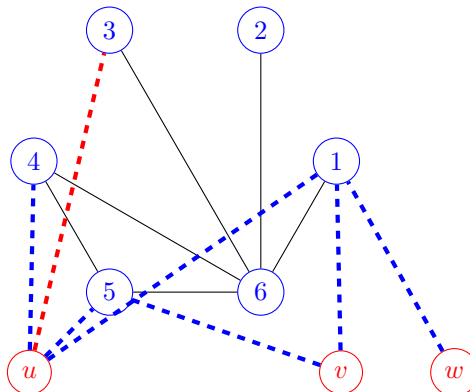
W przykładzie z rysunku 5.9 musi powstać zbiór niezależny N_5 , ponieważ żaden z wierzchołków u, v, w nie może zostać połączony z którymś z niesąsiadujących wierzchołków 2, 4. Ten sposób połączenia zostanie więc odrzucony.



Rysunek 5.9. Przykład sytuacji odrzucanej przez zasadę C

W innym wypadku do B_u trzeba dodać wierzchołki które nie sąsiadują z wierzchołkiem, z którym żaden z wierzchołków u, v, w nie może zostać połączony, z wyjątkiem tych z którymi może sąsiadować v lub w .

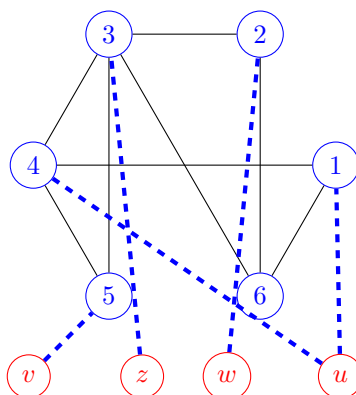
Dla sytuacji z rysunku 5.10 konieczne jest dodanie do dolnego ograniczenia dla u wierzchołka 4.



Rysunek 5.10. Przykład przedziału zawężanego przez zasadę C

Zasada D sprawdza czy graf H zawiera wierzchołek, z którym żaden z niesąsiadujących wierzchołków u, v, w, z nie może sąsiadować. Ponownie, w takim przypadku powstałby zbiór niezależny N_5 .

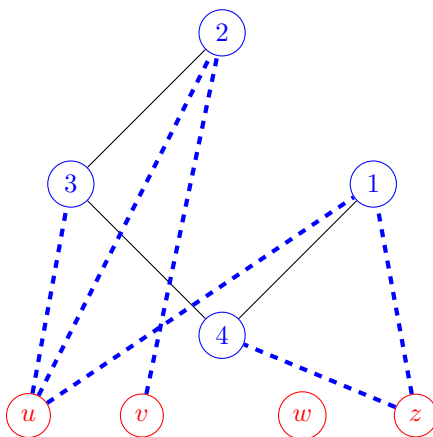
Na rysunku 5.11 żaden z wierzchołków v, z, w, u nie może zostać połączony z wierzchołkiem 6. Ten sposób połączenia należy odrzucić.



Rysunek 5.11. Przykład sytuacji odrzucanej przez zasadę D

Jeżeli powyższe nie zajdzie, musimy dodać do B_u wszystkie wierzchołki, z którymi v, w, z nie mogą sąsiadować.

Przykładowo, na rysunku 5.12 u jest połączone ze wszystkimi wierzchołkami 1-5, ale 1 i 2 mogą też zostać połączone z innymi wierzchołkami, więc do B_u zostanie dodany tylko wierzchołek 3.



Rysunek 5.12. Przykład przedziału zawężanego przez zasadę D

Warto zauważyć, że wynikiem zastosowania dowolnej reguły jest zawsze odrzucenie zestawu przedziałów, zawężenie przedziału lub brak zmian. Reguły są więc aplikowane do momentu, w którym ponowne zaaplikowanie dowolnej reguły dla dowolnych wierzchołków grafu G i odpowiadających im przedziałom nie powoduje już dalszych zmian. Wynikowy zbiór przedziałów daje sposób lub sposoby połączenia grafów G oraz H w graf 24 wierzchołkowy spełniający wymagania pod względem ramseyowskości. Nie wszystkie sposoby połączenia pozostałe przy zatrzymaniu reguł muszą być jednak poprawne. Istnieje możliwość, że wynikowe przedziały posiadają również połączenia niepoprawne. Z tego powodu, po zakończeniu stosowania reguł należy rozbijać wynikowe przedziały, i ponownie stosować reguły, aż wynikiem ich stosowania będzie tylko jeden poprawny graf.

Przez rozbicie zestawu przedziałów mamy na myśli operację, gdzie przedział jest zamieniany na dwa przedziały przy użyciu własności przedziału numer 1. Należy rozbijać tylko te przedziały, dla których $B \neq T$, ponieważ takie przedziały obejmują więcej niż jeden stożek prawdopodobny, a więc i więcej niż jeden sposób połączenia G i H .

5.5. Zakończenie algorytmu sklejania

Po zakończeniu głównej części procesu sklejania uzyskujemy zbiór grafów $R(4, 5, 24)$. W dalszej części pracy graf z tego zbioru będzie oznaczany jako graf \mathcal{F} . Jeżeli zbiór $R(4, 5, 25) \neq \emptyset$ to w wyniku rozszerzenia grafów \mathcal{F} powinniśmy uzyskać przynajmniej jeden graf 25 wierzchołkowy spełniający wymagania. Rozszerzanie zostaje w tym wypadku wykonane metodą inną od wcześniejszego generowania grafów.

Ta metoda bliższa jest wcześniej opisanemu algorytmowi tworzenia przedziałów stożków, choć w tym wypadku mamy tylko jeden wierzchołek, do którego przypisywane są stożki rozszerzające.

Definicja 5.5.1. Stożek rozszerzający to podzbiór wierzchołków grafu \mathcal{F} symbolizujący możliwe połączenia pomiędzy grafem \mathcal{F} a wierzchołkiem do niego dodawanym.

Po raz kolejny, wszystkie możliwe stożki rozszerzające są grupowane w przedział, który w trakcie działania algorytmu jest zawężany. Przedziały stożków rozszerzających mają te same własności, co przedziały stożków prawdopodobnych, w tym wypadku mają jednak znaczenie dwie dodatkowe własności:

1. Dla wierzchołka w , który spełnia $w \notin B$ oraz $w \in T$ prawdą jest, że $[B, T] = [B + \{w\}, T] \cup [B, T - \{w\}]$ oraz $[B + \{w\}, T] \cap [B, T - \{w\}] = \emptyset$
2. Jeżeli ograniczenie przedziału B zawiera klikę stopnia 3, to wszystkie stożki w przedziale również ją zawierają.
3. Jeżeli ograniczenie przedziału T nie zawiera klikę stopnia 3, wszystkie stożki w przedziale jej nie zawierają.
4. Jeżeli ograniczenie przedziału B zawiera przynajmniej jeden wierzchołek ze zbioru niezależnego stopnia 4, to wszystkie stożki w przedziale również go zawierają.
5. Jeżeli ograniczenie przedziału T nie zawiera żadnego wierzchołka ze zbioru niezależnego stopnia 4, wszystkie stożki w przedziale ich nie zawierają.

Etapem przygotowawczym do rozszerzania jest stworzenie listy podzbiorów wierzchołków \mathcal{F} które tworzą klikę lub zbiory niezależne. Takie podzbiory nazywamy dalej strukturami.

Definicja 5.5.2. Struktura to podzbiór wierzchołków grafu \mathcal{F} który nakłada ograniczenia na sposoby rozszerzenia go o wierzchołek.

Celem powstania listy struktur jest przyspieszenie znajdowania stożków rozszerzających, które prowadzą do zaburzenia ramseyowości grafu wynikowego. Jeżeli nowy wierzchołek zostałby połączony ze wszystkimi wierzchołkami K_3 , to powstające K_4 dyskwalifikuje graf. Podobna sytuacja występuje w przypadku braku połączenia do któregośkolwiek z wierzchołków N_4 gdzie w wyniku otrzymujemy niedozwolone N_5 . Mając listę wszystkich struktur, można przystąpić do zawężania przedziału stożków rozszerzających. Rozpoczynamy ponownie od przedziału $[B = \emptyset, T = \{V(\mathcal{F})\}]$ który obejmuje wszystkie możliwe stożki dla grafu \mathcal{F} , a więc również wszystkie możliwe rozszerzenia. Dla każdej struktury S , których listę utworzyliśmy wcześniej dokonujemy porównania przedziałów. Porównywanie odbywa się w następujące sposoby, zależne od tego, czy obecnie rozważana struktura opisuje klikę, czy zbiór niezależny.

Rozumowanie dla ograniczenia wynikającego z klikę przebiega następująco:

Algorithm 4 Porównanie przedziału P do struktury S opisującej klikę

```
if  $S \subseteq T_P$  then
  if  $S \subseteq B_P$  then
    Usuń  $P$  z listy przedziałów
  else
    Podziel przedział  $P$  wzdłuż  $S \setminus B_P$ 
  end if
end if
```

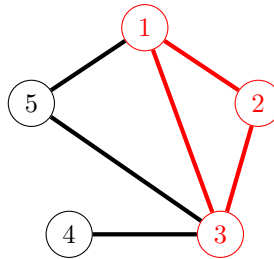
Rozumowanie dla ograniczenia wynikającego ze zbioru niezależnego przebiega z kluczowymi różnicami w warunkach oraz sposobie podziału:

Algorithm 5 Porównanie przedziału P do struktury S opisującej zbiór niezależny

```
if  $S \not\subseteq B_P$  then
  if  $S \not\subseteq T_P$  then
    Usuń  $P$  z listy przedziałów
  else
    Podziel przedział  $P$  wzdłuż  $S \cap T_P$ 
  end if
end if
```

W powyższych algorytmach zostaje użyte pojęcie podziału przedziałów. Podział przedziału polega na użyciu własności 1 dla podanego zbioru wierzchołków w celu wielokrotnego rozdzielania przedziału na dwie części a następnie odrzucenia tego z nich, który zawiera wszystkie stożki łączące sprawdzaną strukturę w niepoprawny sposób. Pozostałe przedziały wynikowe są dodawane do listy przedziałów do sprawdzenia, ponieważ nie łamią obecnie analizowanego ograniczenia.

Dla przykładu załóżmy, że dla grafu G (rysunek 5.13), który próbujemy rozszerzyć o wierzchołek v na rysunku poniżej przekształcamy przedział $P = [\{5\}, \{1, 2, 3, 4, 5\}]$ przy pomocy klik $S = \{1, 2, 3\}$.



Rysunek 5.13. Graf G wraz z zaznaczoną strukturą S

Zaaplikowanie własności 1 dla przedziału P i pierwszego wierzchołka ze struktury czyli $\{1\}$ daje nam

$$[\{5\}, \{1, 2, 3, 4, 5\}] = [\{1, 5\}, \{1, 2, 3, 4, 5\}] \cup [\{5\}, \{2, 3, 4, 5\}]$$

Warto zauważyć, że drugi z wynikowych przedziałów nie łamie już ograniczenia narzuconego przez strukturę P , a więc można dodać go do listy przedziałów i kontynuować podział jedynie na pierwszym z przedziałów wynikowych. Analogiczna sytuacja zachodzi na każdym etapie podziału, ze względu na naturę ograniczenia wypływającego z klik, gdzie usunięcie jednego z wierzchołków klik powoduje akceptację połączenia do reszty z nich.

Kontynuujemy podział pozostałego przedziału dla kolejnego wierzchołka ze zbioru $\{2\}$

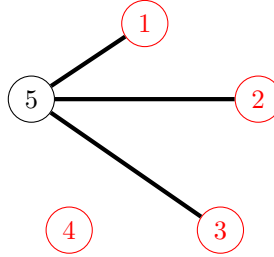
$$[\{1, 5\}, \{1, 2, 3, 4, 5\}] = [\{1, 2, 5\}, \{1, 2, 3, 4, 5\}] \cup [\{1, 5\}, \{1, 3, 4, 5\}]$$

Po raz kolejny można zaakceptować drugi z przedziałów i kontynuować podział pierwszego przez kolejny i ostatni wierzchołek $\{3\}$:

$$[\{1, 2, 5\}, \{1, 2, 3, 4, 5\}] = [\{1, 2, 3, 5\}, \{1, 2, 3, 4, 5\}] \cup [\{1, 2, 5\}, \{1, 2, 4, 5\}]$$

Po podziale przez ostatni z wierzchołków można, jak po każdym kroku, zaakceptować drugi z wynikowych przedziałów. Pierwszy z nich natomiast zawiera strukturę w stożku ograniczającym od dołu. Można go odrzucić ze względu na to, że wszystkie stożki takiego przedziału opisują niedozwolone rozszerzenia grafu \mathcal{F} .

Podział przedziału dla zbioru niezależnego przeprowadzany jest w bardzo podobny sposób. Na każdym etapie podziału akceptowany jest pierwszy przedział wynikowy zamiast drugiego. W ramach przykładu podziału dla zbioru niezależnego założymy, że $P = [\emptyset, \{1, 2\}]$ oraz $S = \{1, 2, 3, 4\}$ i S jest zbiorem niezależnym. Mamy również inny graf G , rysunek 5.14.



Rysunek 5.14. Graf G wraz z zaznaczonymi wierzchołkami należącymi do struktury S

W takim wypadku zbiór wierzchołków, wzdłuż którego dokonujemy podziału, nie obejmuje wszystkich wierzchołków należących do S . Jest tak dlatego, że część z nich należy już do T_P i nie ma potrzeby ani możliwości dzielić przedziału przy ich pomocy. Dla pomniejszego zbioru wierzchołków $\{1, 2\}$ pierwszym etapem podziału będzie

$$[\emptyset, \{1, 2\}] = [\{1\}, \{1, 2\}] \cup [\emptyset, \{2\}]$$

Akceptujemy pierwszy przedział i przechodzimy do kolejnego kroku

$$[\emptyset, \{2\}] = [\{2\}, \{2\}] \cup [\emptyset, \emptyset]$$

Po przejściu wszystkich wierzchołków i zaakceptowaniu pierwszego przedziału można odrzucić drugi, ponieważ wszystkie stożki przez niego objęte łamią ograniczenie narzucone przez S .

Warto zauważyć, że w przypadku podziału przedziałów w taki sposób mamy gwarancje, że jeżeli przedział P spełnia ograniczenie, to wszystkie jego podprzedziały również je spełniają. Ze względu na to, po zaaplikowaniu wszystkich ograniczeń, pozostałe przedziały posiadają poprawne rozszerzenia grafu \mathcal{F} o wierzchołek tak, że nie jest zaburzona jego ramseyowskość.

6. IMPLEMENTACJA I EKSPERYMENTY

Nasz program został zaimplementowany w języku C z wykorzystaniem biblioteki Nauty and Traces[2], która umożliwia łatwe obliczeniowo wykrywanie orbit, co znacznie przyspiesza proces generowania grafów. Autorem biblioteki jest profesor McKay.

Jednym z aspektów powyższego kodu, który wykorzystujemy jest sposób przechowywania grafów w pamięci komputerowej, który bazuje na macierzy sąsiedztwa. Macierz sąsiedztwa to sposób reprezentacji grafu o N wierzchołkach przy użyciu macierzy kwadratowej o wymiarach $N \times N$. Wartość na pozycji (m, n) odpowiada krawędzi pomiędzy wierzchołkami m oraz n (rysunek 6.1).



Rysunek 6.1. Graf wraz z odpowiadającą mu macierzą sąsiedztwa. W tej macierzy 0 odpowiada brakowi krawędzi, a 1 odpowiada jej istnieniu.

Warto zauważyć nadmiarowość macierzy, gdzie każdej krawędzi w grafie odpowiadają dwie wartości. Ta nadmiarowość okazjonalnie pozwala na przyspieszenie obliczeń w zmodyfikowanej wersji macierzy sąsiedztwa używanej w naszym kodzie. Modyfikacja metody macierzy sąsiedztwa polega na odejściu od zapisywania każdej liczby w macierzy jako osobnej wartości. Jako że zajmujemy się jedynie grafami prostymi i niekolorowanymi, to wartości w poszczególnych komórkach mogą wynosić jedynie 0 lub 1. W związku z tym wiersz macierzy można zapisać nie jako n wartości, a jako jedną wartość o odpowiedniej liczbie bitów. Ze względu na to, że największym grafem występującym w naszej pracy jest potencjalny graf 25 wierzchołkowy, 32 bitowa wartość jest wystarczająca żeby pomieścić wiersz macierzy reprezentujący wierzchołek dowolnego grafu który może zostać wygenerowany przez nasz program.

BIBLIOGRAFIA

- [1] Brendan D. McKay, Stanisław P. Radziszowski. $R(4,5) = 25$.
- [2] Brendan D. McKay. *nauty user's guide (version 2.4)*. Department of Computer Science Australian National University Canberra ACT 0200, Australia November 4, 2009
- [3] Stanisław P. Radziszowski. *Small Ramsey Numbers*. Department of Computer Science Rochester Institute of Technology Rochester, NY 14623 June 11, 1994
- [4] Janusz Dybizbański. *Liczby Ramseya z cyklem C_4* . Uniwersytet Warszawski Wydział Matematyki, Informatyki i Mechaniki Listopad 2013
- [5] Christos Nestor Chachamis. *Ramsey Numbers*. May 13, 2018
- [6] Lane Barton IV *Ramsey Theory*. May 13, 2016
- [7] Brendan D. McKay *ISOMORPH-FREE EXHAUSTIVE GENERATION*. May 13, 2016
- [8] P. Erdős, Some Remarks on the Theory of Graphs, *Bulletin of the American Mathematical Society*, 53(4):292–294, 1947
- [9] Christos Nestor Chachamis *Ramsey Numbers*. May 13, 2018
- [10] Ramsey, F.P. (1930), *On a Problem of Formal Logic*. Proceedings of the London Mathematical Society, s2-30: 264-286. <https://doi.org/10.1112/plms/s2-30.1.264>
- [11] Erdős, P.; Szekeres, G. A combinatorial problem in geometry. *Compositio Mathematica*, Volume 2 (1935) , pp. 463-470.
- [12] *Ramsey Theory* by Ronald L. Graham and Joel H. Spencer, in *Scientific American* (July 1990), p. 112-117
- [13] OEIS Foundation Inc. (2020), The On-Line Encyclopedia of Integer Sequences, <http://oeis.org/A000088>.
- [14] Nicolas Bloyet, Pierre-François Marteau, Emmanuel Frenod. Scott : A method for representing graphs as rooted trees for graph canonization. *COMPLEX NETWORKS 2019*, Springer, pp.578-590, 2019, *Studies in Computational Intelligence Series*, $ff10.1007/978 - 3 - 030 - 36687 - 2_48ff.f fhal - 02314658f$
- [15] Rich Wolski CS290I Lecture notes – Ramsey Numbers: Top Hits Played at Parties Thrown by Paul Erdos"<https://sites.cs.ucsb.edu/~rich/class/cs293-cloud/notes/Ramsey/index.html>
- [16] Weisstein, Eric W. "Party Problem."From MathWorld–A Wolfram Web Resource. <https://mathworld.wolfram.com/PartyProblem.html>
- [17] Brendan D McKay, Stanisław P Radziszowski, Subgraph Counting Identities and Ramsey Numbers, *Journal of Combinatorial Theory, Series B*, Volume 69, Issue 2, 1997, Pages 193-209, ISSN 0095-8956, <https://doi.org/10.1006/jctb.1996.1741>. (<http://www.sciencedirect.com/science/article/pii/S0095895696917414>)

- [18] Charles M Grinstead, Sam M Roberts, On the Ramsey numbers $R(3, 8)$ and $R(3, 9)$, Journal of Combinatorial Theory, Series B, Volume 33, Issue 1, 1982, Pages 27-51, ISSN 0095-8956, [https://doi.org/10.1016/0095-8956\(82\)90055-7](https://doi.org/10.1016/0095-8956(82)90055-7). (<http://www.sciencedirect.com/science/article/pii/0095895682900557>)
- [19] OEIS Foundation Inc. (2020), The On-Line Encyclopedia of Integer Sequences, <http://oeis.org/A006125>.