



WYDZIAŁ ELEKTRONIKI,
TELEKOMUNIKACJI
I INFORMATYKI

Imię i nazwisko studenta: Michał Zalewski
Nr albumu: 171671
Poziom kształcenia: Studia drugiego stopnia
Forma studiów: stacjonarne
Kierunek studiów: Informatyka
Specjalność: Algorytmy i technologie internetowe

PRACA DYPLOMOWA MAGISTERSKA

Tytuł pracy w języku polskim: Opracowanie aplikacji ilustrującej grę w dominowanie rzymskie wraz z rozważaniami teoretycznymi

Tytuł pracy w języku angielskim: Development of an application illustrating the game of Roman domination with theoretical considerations

Opiekun pracy: dr inż. Joanna Raczek

OŚWIADCZENIE dotyczące pracy dyplomowej zatytułowanej: Opracowanie aplikacji ilustrującej grę w dominowanie rzymskie wraz z rozważaniami teoretycznymi

Imię i nazwisko studenta: Michał Zalewski
Data i miejsce urodzenia: 07.04.1998, Gdynia
Nr albumu: 171671

Wydział: Wydział Elektroniki, Telekomunikacji i Informatyki
Kierunek: informatyka
Poziom kształcenia: drugi
Forma studiów: stacjonarne
Typ pracy: praca dyplomowa magisterska

Świadomy(a) odpowiedzialności karnej z tytułu naruszenia przepisów ustawy z dnia 4 lutego 1994 r. o prawie autorskim i prawach pokrewnych (t.j. Dz. U. z 2019 r. poz. 1231, z późn. zm.) i konsekwencji dyscyplinarnych określonych w ustawie z dnia 20 lipca 2018 r. Prawo o szkolnictwie wyższym i nauce (t.j. Dz. U. z 2020 r. poz. 85, z późn. zm.),¹ a także odpowiedzialności cywilnoprawnej oświadczam, że przedkładana praca dyplomowa została opracowana przeze mnie samodzielnie.

Niniejsza praca dyplomowa nie była wcześniej podstawą żadnej innej urzędowej procedury związanej z nadaniem tytułu zawodowego.

Wszystkie informacje umieszczone w ww. pracy dyplomowej, uzyskane ze źródeł pisanych i elektronicznych, zostały udokumentowane w wykazie literatury odpowiednimi odnośnikami zgodnie z art. 34 ustawy o prawie autorskim i prawach pokrewnych.

30.05.2023, Michał Zalewski

Data i podpis lub uwierzytelnienie w portalu uczelnianym Moja PG

**) Dokument został sporządzony w systemie teleinformatycznym, na podstawie §15 ust. 3b Rozporządzenia MNiSW z dnia 12 maja 2020 r. zmieniającego rozporządzenie w sprawie studiów (Dz.U. z 2020 r. poz. 853). Nie wymaga podpisu ani stempla.*

¹ Ustawa z dnia 20 lipca 2018 r. Prawo o szkolnictwie wyższym i nauce:

Art. 312. ust. 3. W przypadku podejrzenia popełnienia przez studenta czynu, o którym mowa w art. 287 ust. 2 pkt 1–5, rektor niezwłocznie poleca przeprowadzenie postępowania wyjaśniającego.

Art. 312. ust. 4. Jeżeli w wyniku postępowania wyjaśniającego zebrany materiał potwierdza popełnienie czynu, o którym mowa w ust. 5, rektor wstrzymuje postępowanie o nadanie tytułu zawodowego do czasu wydania orzeczenia przez komisję dyscyplinarną oraz składa zawiadomienie o podejrzeniu popełnienia przestępstwa.

STRESZCZENIE

Dominowanie jest jednym z podstawowych zagadnień teorii grafów. W ramach badań analizowane są jego nowe warianty i konteksty zastosowania. Praca ma na celu opracowanie wprowadzenia teoretycznego do dwóch z nich: gry w dominowanie rzymskie oraz gry w dominowanie rzymskie z podziałem krawędzi. Drugim celem jest stworzenie aplikacji, umożliwiającej wizualizację tych gier i prezentującej podstawowe strategie automatyczne do grania w nie.

Pierwszy z powyższych celów został osiągnięty poprzez przedstawienie podstawowych pojęć oraz rozszerzonych dowodów zaczerpniętych z literatury twierdzeń na temat własności gier. Wprowadzono również twierdzenia pomocnicze, których dowody nie znajdują się w literaturze. Stworzone zostały także ilustracje, ułatwiające zrozumienie opisywanych koncepcji.

Aplikacja została zrealizowana w sposób, umożliwiający jej rozszerzenie dla kolejnych badań nad grami w dominowanie rzymskie. Została ona napisana w języku Python z wykorzystaniem bibliotek pygame oraz igraph. Umożliwia implementację własnych strategii gry w dominowanie rzymskie i testowanie ich w porównaniu z opracowanymi w ramach pracy strategiami bazowymi.

Całość pracy stanowi wstęp do przedstawianych zagadnień, który nie wymaga wysokiego poziomu wiedzy z zakresu teorii grafów i może zostać wykorzystana do dalszych badań nad grami w dominowanie rzymskie lub jako materiał wspomagający implementację ich praktycznych zastosowań.

Słowa kluczowe: dominowanie w grafach, gra w dominowanie rzymskie, gra w dominowanie rzymskie z podziałem krawędzi

Dziedzina nauki i techniki, zgodnie z wymogami OECD: 1.2.a Nauka o komputerach, informatyka i bioinformatyka, 2.2 Elektrotechnika, elektronika i inżynieria informatyczna (w zakresie informatyki technicznej i telekomunikacji)

ABSTRACT

Domination is one of the fundamental topics of graph theory. Research is developing its new variants and application contexts. The goal of this thesis is to develop a theoretical introduction to two of them: the Roman domination game and the Roman domination subdivision game. Another goal is to create an application that would allow visualization of these games and present basic automatic strategies for playing them.

The first of the above goals is achieved by presenting the basic concepts and extended proofs of theorems on the properties of the games taken from the literature. Supporting theorems whose proofs are not found in the literature are introduced as well. Illustrations are also created to facilitate understanding of the concepts described.

The application was implemented in a way that makes it easy to extend it for further research on Roman domination games. It was written in Python language using pygame and igraph libraries. It allows implementing custom strategies for Roman domination games and testing them against the baseline strategies developed as part of the thesis.

The thesis as a whole provides an entry point into the issues presented, which does not require a high level of prior knowledge of graph theory, and can be used for further research into Roman domination games or as a resource to support the implementation of their practical applications.

Keywords: domination in graphs, Roman domination game, Roman domination subdivision game

SPIS TREŚCI

Wykaz ważniejszych oznaczeń i skrótów	7
1. WSTĘP	8
2. DOMINOWANIE RZYMSKIE	10
2.1. Dominowanie w grafach	10
2.1.1. Dominowanie w grafach skierowanych	10
2.1.2. Zbiór dominujący	11
2.1.3. Liczba dominowania	11
2.2. Dominowanie rzymskie	11
2.2.1. Dominowanie rzymskie w grafach skierowanych	11
2.2.2. Liczba dominowania rzymskiego	12
2.2.3. Naiwny algorytm wyznaczania liczby dominowania rzymskiego	12
3. GRA W DOMINOWANIE RZYMSKIE	13
3.1. Gra w dominowanie	13
3.1.1. Liczba gry w dominowanie	13
3.2. Gra w dominowanie rzymskie	13
3.2.1. Liczba gry w dominowanie rzymskie	13
3.3. Liczba gry w dominowanie rzymskie dla wybranych klas grafów	14
3.3.1. Dowolny spójny graf	14
3.3.2. Klika	16
3.3.3. Gwiazda	20
3.3.4. Ścieżka	20
3.3.5. Podwójna gwiazda	22
3.3.6. Drzewo	25
4. GRA W DOMINOWANIE RZYMSKIE Z PODZIAŁEM KRAWĘDZI	28
4.1. Gra w dominowanie z podziałem krawędzi	28
4.1.1. Liczba gry w dominowanie z podziałem krawędzi	28
4.2. Gra w dominowanie rzymskie z podziałem krawędzi	28
4.3. Liczba gry w dominowanie rzymskie z podziałem krawędzi	29
4.3.1. Gwiazda	29
4.3.2. Ścieżka i cykl	30
4.3.3. Graf r -regularny	32
4.3.4. k -wiatrak	33
4.3.5. Graf dwudzielny	35
4.3.6. Podwójna gwiazda	39
4.3.7. Drzewa	43
5. APLIKACJA DO GRY W DOMINOWANIE RZYMSKIE	55
5.1. Funkcje aplikacji	55
5.1.1. Tryb graficzny	55
5.1.2. Tryb tekstowy	55
5.1.3. Tryb testowy	56

5.2. Interfejs użytkownika	56
5.2.1. Tryb graficzny	56
5.2.2. Tryb tekstowy	57
5.2.3. Tryb testowy	59
5.3. Instalacja zależności i uruchomienie aplikacji	60
5.4. Użyte technologie	60
5.4.1. <code>igraph</code>	60
5.4.2. <code>pygame</code>	61
5.5. Projekt aplikacji	61
5.5.1. Moduł <code>game</code>	61
5.5.2. Moduł <code>display</code>	63
5.5.3. Moduł <code>players</code>	64
5.5.4. Moduł <code>utils</code>	64
5.5.5. Moduł <code>config</code>	64
5.5.6. Skrypt <code>rdg.py</code>	65
5.5.7. Skrypt <code>benchmark.py</code>	65
5.6. Strategie automatyczne	66
5.6.1. Strategie odniesienia	66
5.6.2. Strategie oparte na punktacji wierzchołków	66
5.6.3. Strategie MinMax	67
6. PODSUMOWANIE	70
6.1. Osiągnięcia	70
6.2. Dalszy rozwój aplikacji	70
6.3. Wnioski	71
Wykaz rysunków	72
Wykaz tabel	73

WYKAZ WAŻNIEJSZYCH OZNACZEŃ I SKRÓTÓW

$V(G)$	zbiór wierzchołków grafu G
$E(G)$	zbiór krawędzi grafu G
$n(G)$	liczba wierzchołków grafu G
$d(v)$	stopień wierzchołka v
$d^+(v)$	stopień wychodzący wierzchołka v
$d^-(v)$	stopień wchodzący wierzchołka v
$\Delta(G)$	największy stopień wierzchołka w grafie G
$\Delta^+(G)$	największy stopień wychodzący wierzchołka w grafie G
$\Delta^-(G)$	największy stopień wchodzący wierzchołka w grafie G
$\delta(G)$	najmniejszy stopień wierzchołka w grafie G
$\tau(G)$	moc najmniejszego pokrycia wierzchołkowego grafu G
$\alpha(G)$	moc największego zbioru niezależnego grafu G
$N[v]$	sąsiedztwo domknięte wierzchołka v
\mathcal{D}	relacja dominowania
$f(v)$	wartość funkcji dominowania rzymskiego dla wierzchołka v
$f(V)$	sumaryczna wartość funkcji dominowania rzymskiego dla zbioru wierzchołków V
$f_{\min}(v)$	wartość najmniejszej funkcji dominowania rzymskiego dla wierzchołka v
$\gamma(G)$	liczba dominowania grafu G
$\gamma_2(G)$	liczba 2-dominowania grafu G
$\gamma_g(G)$	liczba gry w dominowanie grafu G
$\gamma_{gs}(G)$	liczba gry w dominowanie z podziałem krawędzi grafu G
$\gamma_R(G)$	liczba dominowania rzymskiego grafu G
$\gamma_{Rg}(G)$	liczba gry w dominowanie rzymskie grafu G
$\gamma_{Rgs}(G)$	liczba gry w dominowanie rzymskie z podziałem krawędzi grafu G

1. WSTĘP

Dominowanie w grafach jest zagadnieniem, które od dawna interesuje badaczy. Jego początków można szukać w zagadkach szachowych, takich jak problem królowych, opisywanych już w XIX wieku. [4] W połowie ubiegłego stulecia, wraz z rozwojem ogólnie pojętej teorii grafów, stworzone zostało pojęcie zbioru dominowania. W niedługim czasie stało się częścią ścisłego kanonu wiedzy w tej dziedzinie matematyki i zaczęto opisywać je w podręcznikach [13], co bez wątpienia wpłynęło na poszerzenie zainteresowania tym zagadnieniem i dynamiczny rozwój badań nad nim.

Od tamtego czasu opracowany został szeroki wachlarz rozważań teoretycznych, wprowadzający wiele wariantów relacji dominowania i opisujących ich własności. Dostrzeżono również wiele analogii pomiędzy tym teoretycznym modelem, a rzeczywistymi problemami związanymi z rozmieszczaniem zasobów w różnych kontekstach. [8]

Niniejsza praca skupia się na dominowaniu rzymskim, będącym jednym z wariantów relacji dominowania, opisanym pierwszy raz przez Cockayne'a i in. [6]. Rozważania dotyczą konkretnie dwóch rodzajów gier rozgrywanych na grafach, w których obliczanie wyniku, więc też cel gry, jest podyktowany relacją dominowania rzymskiego. Są to gra w dominowanie rzymskie [3] oraz gra w dominowanie rzymskie z podziałem krawędzi [2].

Celem pracy jest opisanie tych gier oraz przedstawienie ich podstawowych własności na podstawie publikacji, w których zostały wprowadzone. Przedstawione są rozszerzone definicje pojęć związanych z dominowaniem rzymskim oraz powyższymi grami. Przytoczone są również twierdzenia z literatury, których dowody zostały poszerzone i zilustrowane, aby ułatwić czytelnikowi przyswojenie wiedzy z tego zakresu.

Ponadto opracowana jest aplikacja, pozwalająca na rozgrywanie obu tych gier pomiędzy dwoma graczami lub przeciwko opracowanym strategiom automatycznym. Pozwala ona na wyznaczenie powiązanych z grami w dominowanie rzymskie własności grafów. Jest również zaprojektowana w sposób, umożliwiający jej rozszerzenie o nowe strategie. Tryb testowy aplikacji pozwala na zestawienie ich ze sobą i porównanie ich efektywności w każdej z gier.

W rozdziale 2. przedstawione jest wprowadzenie do zagadnienia dominowania w grafach, wraz z definicjami podstawowych pojęć. Następnie opisana jest relacja dominowania rzymskiego, przedstawiona w kontraście do standardowego dominowania. Zarysowany jest również podstawowy algorytm wyznaczania liczby dominowania rzymskiego grafu oraz wyprowadzona jest jego złożoność obliczeniowa.

Rozdział 3. poświęcony jest grze w dominowanie rzymskie. Na początku wprowadzona jest oryginalna koncepcja gry w dominowanie oparta na standardowej relacji dominowania. Dalej opisana jest gra w dominowanie rzymskie, po czym przedstawiona jest seria twierdzeń o liczbie gry w dominowanie rzymskie. Twierdzenia zostały przytoczone z literatury, natomiast dowody zostały rozszerzone względem oryginalnych. Celem było bardziej kompleksowe opracowanie, pozwalające czytelnikowi nieposiadającemu specjalistycznej wiedzy w tej dziedzinie zrozumieć dowody oraz używane w nich koncepcje, których wytłumaczenia są często pomijane w publikacjach naukowych. Ponadto dowody zostały zilustrowane w sposób wspomagający wizualizację problemu.

W tym rozdziale sformułowane zostało również autorskie twierdzenie 3.4 oraz autorski dowód lematu 3.6, który pojawia się w literaturze bez dołączonego dowodu.

Rozdział 4. w analogiczny sposób wprowadza zagadnienie gry w dominowanie rzymskie z podziałem krawędzi wraz z twierdzeniami dotyczącymi liczby gry w dominowanie rzymskie z podziałem krawędzi. W celu udowodnienia niektórych z zależności wprowadzane są pojęcia pokrycia wierzchołkowego, zbioru niezależnego oraz skojarzenia i wywodzone są twierdzenia pomocnicze na ich temat.

W rozdziale 5. opisano stworzoną w ramach pracy aplikację do gry w dominowanie rzymskie i dominowanie rzymskie z podziałem krawędzi. Przedstawiono jej funkcje oraz interfejs użytkownika. Wyjaśniony został dokładnie proces instalacji zależności koniecznych do uruchomienia aplikacji, aby umożliwić czytelnikowi łatwe skorzystanie z niej. Zaprezentowano również projekt aplikacji z wyróżnieniem poszczególnych modułów i użytych zewnętrznych bibliotek. Na końcu wymieniono i opisano działanie zaimplementowanych strategii automatycznych, w tym jednej strategii opracowanej koncepcyjnie w ramach pracy. Następnie podsumowano wyniki eksperymentu, w wyniku którego wyznaczono efektywne parametry dla tej strategii.

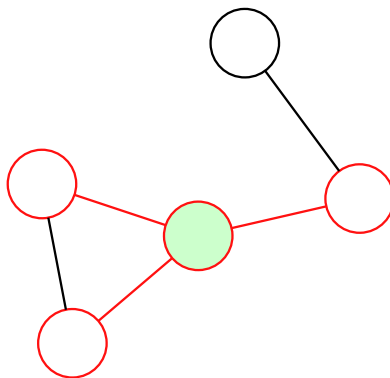
Ostatni rozdział 6. stanowi podsumowanie obu części pracy. Wyszczególnione są cele osiągnięte w ramach pracy. Wskazano również potencjalne obszary dla dalszego rozwoju stworzonej aplikacji. Na końcu przedstawiono wnioski wypływające z ogółu pracy.

2. DOMINOWANIE RZYMSKIE

2.1. Dominowanie w grafach

Relacja dominowania \mathcal{D} w grafie prostym G zachodzi pomiędzy wierzchołkami $v_1, v_2 \in V(G)$, jeżeli $v_1 = v_2$ (v_1 i v_2 to ten sam wierzchołek) lub $\{v_1, v_2\} \in E(G)$ (v_1 i v_2 są wierzchołkami sąsiednimi).

W innym ujęciu relacja dominowania może być rozumiana jako tożsama pojęciu sąsiedztwa domkniętego. Sąsiedztwo domknięte wierzchołka v , oznaczane jako $N[v]$, to zbiór wierzchołków grafu G zawierający wierzchołek v oraz wszystkie sąsiednie do niego wierzchołki, tj. $N[v] = \{v\} \cup \{u \in V(G) : \{v, u\} \in E(G)\}$. Można więc powiedzieć, że wierzchołek $v \in V(G)$ dominuje każdy wierzchołek ze swojego sąsiedztwa $N[v]$ ($v \mathcal{D} N[v]$).



Rys. 2.1. Czerwone wierzchołki stanowią sąsiedztwo domknięte wierzchołka wypełnionego zielonym kolorem, czyli są przez niego dominowane

2.1.1. Dominowanie w grafach skierowanych

Relacja dominowania w grafach skierowanych może być zdefiniowana analogicznie do tej w grafach prostych, lecz brane pod uwagę są tylko wierzchołki sąsiadujące poprzez łuki, wychodzące z wierzchołka v .

W związku z tym relacja ta nadal jest tożsama z relacją sąsiedztwa domkniętego (z ang. *out-neighborhood*), które jednak jest zdefiniowane nieco inaczej ze względu na obecność krawędzi skierowanych. Sąsiedztwo domknięte wierzchołka v składa się z niego samego oraz wszystkich wierzchołków z nim incydentnych, tzn. takich, dla których istnieje łuk, który łączy z nimi wierzchołek v . Formalna definicja sąsiedztwa domkniętego wierzchołka v w grafie skierowanym D będzie więc wyglądać następująco:

$$N[v] = \{v\} \cup \{u \in V(D) : (v, u) \in E(G)\}.$$

Podobnie jak w przypadku dominowania w grafach prostych, relacja dominowania zachodzi pomiędzy wierzchołkiem $v \in V(D)$ a jego sąsiedztwem domkniętym $N[v]$.

2.1.2. Zbiór dominujący

Zbiór dominujący to taki podzbiór wierzchołków grafu, który dominuje wszystkie wierzchołki w tym grafie. Istotnym szczególnym przypadkiem jest **najmniejszy zbiór dominujący**, czyli taki zbiór dominujący, którego moc jest najmniejsza.

Przyjmijmy, że dla zbiorów A, B relacja $A \mathcal{D} B$ oznacza, że elementy zbioru A dominują wszystkie elementy zbioru B . Wtedy najmniejszy zbiór dominujący D grafu G jest definiowany jako:

$$D \subseteq V(G), D \mathcal{D} V(G), \forall A: A \mathcal{D} V(G) |D| \leq |A|.$$

2.1.3. Liczba dominowania

Moc najmniejszego zbioru dominującego w grafie G jest określana mianem liczby dominowania grafu G i oznaczana typowo symbolem $\gamma(G)$:

$$\gamma(G) = |D_{min}|.$$

2.2. Dominowanie rzymskie

Relacja dominowania rzymskiego jest wariantem klasycznej relacji dominowania w grafach. Różnica polega na wprowadzeniu funkcji dominowania rzymskiego, przypisującej każdemu z wierzchołków grafu wartość:

$$f : V(G) \rightarrow \{0, 1, 2\}.$$

Działanie relacji dominowania będzie zależne od wartości nadanej wierzchołkowi:

- Wierzchołki o wartości $f(v) = 0$ nie dominują żadnego z wierzchołków.
- Wierzchołki o wartości $f(v) = 1$ dominują tylko same siebie.
- Wierzchołki o wartości $f(v) = 2$ są w relacji dominowania z innymi wierzchołkami zgodnie ze standardowym modelem dominowania, tj. dominują całe swoje domknięte sąsiedztwo $(N[v])$.

Dodatkowo można zdefiniować funkcję dominowania rzymskiego dla zbioru wierzchołków V odpowiadającą sumie wartości wszystkich zawartych w nim wierzchołków:

$$f(V) = \sum_{v \in V} f(v).$$

2.2.1. Dominowanie rzymskie w grafach skierowanych

Definicja dominowania rzymskiego w grafach skierowanych nie różni się wiele od dominowania rzymskiego na grafach prostych. W tym wypadku również określona jest funkcja f , przypisująca każdemu z wierzchołków grafu wartość ze zbioru $\{0, 1, 2\}$. Wierzchołki, którym przypisano wartości 0, 1, mają identyczne własności do tych w grafach nieskierowanych, tj.:

- Jeżeli $f(v) = 0$, to wierzchołek v nie dominuje żadnego wierzchołka.
- Jeżeli $f(v) = 1$, to wierzchołek v dominuje tylko sam siebie.

Różnica zachodzi tylko w przypadku, gdy wierzchołkowi v przypisana jest wartość 2. Wtedy dominuje on sam siebie oraz wszystkie wierzchołki, do których wychodzi od niego łuk. Jest to

tożsame relacji standardowego dominowania w grafach skierowanych. Wierzchołek v , w którym $f(v) = 2$, dominuje więc całe swoje domknięte sąsiedztwo $N[v]$.

2.2.2. Liczba dominowania rzymskiego

W przeciwieństwie do konwencjonalnej relacji dominowania pojęcie zbioru dominującego nie ma swojego odpowiednika w przypadku dominowania rzymskiego. Jest tak, ponieważ w relacji dominowania wyróżnić można zbiór dominujący D , czyli zbiór wierzchołków dominujących oraz zbiór wierzchołków dominowanych $V(G) \setminus D$. W przypadku dominowania rzymskiego natomiast wierzchołki można podzielić na trzy zbiory wierzchołków o różnych „rolach” według liczby przypisywanej im przez funkcję dominowania rzymskiego f , w związku z czym trudno o naturalną definicję *zbioru dominującego rzymskiego*.

Istnieje jednak odpowiednik pojęcia liczby dominowania, zdefiniowany w nieco inny sposób niż ta dla konwencjonalnej relacji, ale będący jej koncepcyjnym rozszerzeniem. Liczba dominowania rzymskiego grafu G jest najmniejszą możliwą wartością funkcji $f(V(G))$, gdzie $V(G)$ jest zbiorem wierzchołków G .

2.2.3. Naiwny algorytm wyznaczania liczby dominowania rzymskiego

Wnioskiem z rozdziału 2.2.2 jest, że do znalezienia liczby dominowania rzymskiego dla danego grafu G konieczne będzie wyszukanie takiej funkcji f , przypisującej wartości jego wierzchołkom, dla której wszystkie wierzchołki grafu będą zdominowane oraz ich sumaryczna wartość będzie najmniejsza.

Naiwna implementacja algorytmu wyznaczania liczby dominowania rzymskiego będzie więc musiała przejrzeć przestrzeń wszystkich możliwych funkcji, przypisujące wartości wierzchołkom i dla każdej z nich sprawdzić, czy wszystkie wierzchołki są zdominowane, oraz zsumować ich wartości.

Ponieważ każdy z wierzchołków może mieć przypisaną jedną z trzech wartości, moc zbioru wszystkich funkcji dominowania rzymskiego będzie wynosiła $3^{n(G)}$, gdzie $n(G)$ oznacza liczbę wierzchołków grafu G . Sprawdzenie, czy wierzchołek grafu jest zdominowany, będzie wymagało w pesymistycznym przypadku rozważenia wszystkich sąsiednich do niego wierzchołków. W skrajnym przypadku jeden wierzchołek może mieć maksymalnie $n(G) - 1$ sąsiednich wierzchołków.

Liczba operacji w procedurze wyznaczającej liczbę dominowania rzymskiego w grafie G będzie więc ograniczona od góry przez wyrażenie:

$$\mathcal{O}(3^{n(G)} \cdot n(G) \cdot (n(G) - 1)) = \mathcal{O}(3^{n(G)} \cdot n^2(G)).$$

3. GRA W DOMINOWANIE RZYMSKIE

3.1. Gra w dominowanie

Gra w dominowanie jest grą rozgrywaną na grafie prostym G pomiędzy dwoma graczami: Dominatorem i Psujem (z ang. *avoider*).

Gracze wykonują swoje ruchy w naprzemiennych turach. Graczem zaczynającym rozgrywkę jest Dominator. Ruch polega na wybraniu przez gracza jednej z nieskierowanych krawędzi grafu (nazywanych *krawędziami wolnymi*) i zorientowaniu jej, czyli przekształceniu w krawędź skierowaną pomiędzy tymi samymi wierzchołkami w wybranym kierunku. Gra kończy się, gdy wszystkie krawędzie zostaną zorientowane. Jej efektem jest powstanie grafu skierowanego G' .

Celem gry dla Dominatora jest takie prowadzenie rozgrywki, aby liczba dominowania w grafie wynikowym ($\gamma(G')$) była jak najmniejsza, tzn. aby najmniejsza liczba wierzchołków dominowała cały graf.

Rola Psuja jest odwrotna. Ma on za zadanie zorientować krawędzie tak, aby liczba dominowania otrzymanego grafu skierowanego była jak największa.

3.1.1. Liczba gry w dominowanie

Liczba gry w dominowanie to wartość charakterystyczna dla grafu prostego G równa liczbie dominowania w wynikowym grafie skierowanym G' po rozegraniu na grafie G gry w dominowanie przez graczy, przy założeniu, że obaj posługują się optymalną strategią. Jest ona oznaczana symbolem $\gamma_g(G)$.

Łatwo zauważyć, że liczba gry w dominowanie jest zawsze określona. W przypadku dostępu do nieograniczonych zasobów obliczeniowych przez obu graczy możliwe jest dokonanie przez nich przeglądu całości drzewa gry i określenie, np. z użyciem algorytmu *minimax*, strategii dającej optymalny dla nich wynik. Tak więc, zakładając optymalność gry po obu stronach, $\gamma_g(G)$ zawsze będzie konkretną wartością dla zadanego grafu skierowanego.

3.2. Gra w dominowanie rzymskie

Gra w dominowanie rzymskie jest odpowiednikiem gry w dominowanie dla relacji dominowania rzymskiego. Jej zasady są analogiczne do pierwowzoru. Również rozpoczyna się od grafu prostego G i jest rozgrywana pomiędzy Dominatorem i Psujem orientującymi naprzemiennie nieskierowane krawędzie w grafie. Wynikiem jej, tak jak w przypadku zwykłej gry w dominowanie, jest graf skierowany G' .

Różnica pomiędzy grami jest konsekwencją zmiany celu gry. Celem Dominatora nie jest zmaksymalizowanie liczby dominowania w G' , a liczby dominowania rzymskiego. Podobnie Psuj wykonuje podczas gry ruchy mające na celu minimalizację ostatecznej liczby dominowania rzymskiego w wynikowym grafie.

3.2.1. Liczba gry w dominowanie rzymskie

Podobnie jak miało to miejsce w przypadku gry w dominowanie, gra w dominowanie rzymskie również pozwala na wyznaczenie wartości charakterystycznej dla grafu wejściowego G . Jest ona nazywana liczbą gry w dominowanie rzymskie i oznaczana symbolem $\gamma_{Rg}(G)$. Odpowiada

liczbie dominowania rzymskiego wyznaczonej dla wynikowego grafu G' przy założeniu, że zarówno Dominator, jak i Psuj posługują się optymalnymi strategiami.

Uzasadnienie istnienia takiej liczby jest identyczne jak w przytoczonym powyżej rozumowaniu dla liczby gry w dominowanie.

3.3. Liczba gry w dominowanie rzymskie dla wybranych klas grafów

W ogólnym przypadku ustalenie liczby gry w dominowanie rzymskie wymaga przeszukania całego drzewa gry i znalezienia w nim optymalnej strategii obu graczy, na przykład przy użyciu algorytmu minimax. Taka operacja jest kosztowna obliczeniowo ze względu na swoją wykładniczą złożoność.

Jednakże w wielu przypadkach możliwe jest wyznaczenie zakresu albo konkretnej wartości liczby gry w dominowanie rzymskie, jeżeli graf początkowy spełnia określone własności.

Przykładowo, Bahremandpour i in. [3, rozdz. 3.-4.] przedstawiają szereg ograniczeń, którym podlega liczba gry w dominowanie rzymskie rozgrywanej na konkretnych klasach grafów. Poniżej przytoczono część z ich wyników wraz z rozszerzonymi dowodami.

W ogólności większość dowodów jest przeprowadzana poprzez przedstawienie konkretnej strategii dla Dominatora, zapewniającej, że liczba gry w dominowanie rzymskie nie będzie większa od jakiegoś wyrażenia oraz strategii dla Psuja, zapewniającej dolne ograniczenie jej wartości. W przypadku, gdy strategie obu graczy powodują ograniczenie liczby gry w dominowanie rzymskie od góry i od dołu przez to samo wyrażenie, to oznacza, że przy spełnieniu założonych własności może ona zostać wyznaczona dokładnie ze stałą złożonością obliczeniową.

3.3.1. Dowolny spójny graf

Stożek wychodzący $d^+(v)$ to liczba łuków wychodzących z wierzchołka v . Największy stożek wychodzący wierzchołka należącego do grafu D jest oznaczany symbolem $\Delta^+(D)$.

Analogicznie stożek wchodzący $d^-(v)$ to liczba łuków wchodzących do wierzchołka v , a $\Delta^-(D)$ oznacza największy stożek wchodzący wierzchołka grafu D .

Lemat 3.1. [14] Liczba dominowania rzymskiego grafu skierowanego D o liczbie wierzchołków $n \geq 3$ wynosi $\gamma_R(D) = 3$ wtedy, i tylko wtedy, gdy

1. $n = 3$ oraz $\Delta^+(D) \leq 1$ lub
2. $\Delta^+(D) = n - 2$.

Dowód. Przypadek 1.: Jeżeli $n = 3$, to liczba dominowania rzymskiego równa 3 jest możliwa do osiągnięcia tylko w dwóch przypadkach:

- Wszystkie wierzchołki mają przypisaną wartość równą 1: Jeżeli któryś z wierzchołków dominowałby dwa inne wierzchołki, to liczba dominowania rzymskiego wynosiłaby 2, ponieważ temu wierzchołkowi moglibyśmy przypisać wartość funkcji dominowania rzymskiego równą 2, a wierzchołkom sąsiadującym wartość 0. W związku z tym wartość wszystkich wierzchołków równa 1 implikuje, że każdy z wierzchołków ma co najwyżej jednego sąsiada zewnętrznego, więc $\Delta^+(D) \leq 1$.
- Wierzchołki mają odpowiednio wartości $\{2, 1, 0\}$: Jeżeli wierzchołek o wartości funkcji dominowania rzymskiego równej 2 byłby połączony łukiem wychodzącym z wierzchołkiem o wartości funkcji dominowania rzymskiego wynoszącej 1, to ten wierzchołek mógłby mieć wartość 0, więc liczba dominowania rzymskiego wynosiłaby 2. Wobec

tego wierzchołek o przypisanej wartości 2 może mieć co najwyżej jednego sąsiada (wierzchołek o przypisanej wartości 0), co jest równoważne sytuacji opisanej powyżej.

Przypadek 2.: Jeżeli $n > 3$, to jedyną funkcją dominowania rzymskiego, dającą liczbę dominowania rzymskiego równą 3, będzie taka, która jednemu z wierzchołków przypisuje wartość 2, jednemu wartość 1, a pozostałym wartość 0. Łatwo zauważyć, że taka sytuacja może mieć miejsce tylko, gdy jeden z wierzchołków dominuje samego siebie i $n - 2$ pozostałe wierzchołki. Wtedy wierzchołek ten przyjmuje wartość 2, a niezdominowany przez niego wierzchołek przyjmuje wartość 1.

Powyższe stwierdzenie jest równoznaczne z $\Delta^+(D) = n - 2$.

□

Lemat 3.2. [14] Liczba dominowania rzymskiego grafu skierowanego D , o liczbie wierzchołków $n \geq 4$ wynosi $\gamma_R(D) = 4$ wtedy, i tylko wtedy, gdy

1. $n = 4$ i $\Delta^+(D) \leq 1$ lub
2. istnieją dwa wierzchołki $v, u \in V(D)$, stanowiące zbiór dominujący grafu D lub
3. $\Delta^+(D) \geq n - 3$.

Dowód. **Przypadek 1.:** W przypadku gdy $n = 4$, liczba dominowania rzymskiego w D może być równa 4 w trzech przypadkach:

- Wszystkie wierzchołki mają przypisaną wartość równą 1: dzieje się tak, kiedy żaden z wierzchołków nie dominuje więcej niż jednego wierzchołka, ponieważ gdyby któryś z wierzchołków dominował przynajmniej dwa wierzchołki, to wierzchołek ten mógłby przyjąć wartość 2, a wierzchołki sąsiadujące z nim zewnętrznie mogłyby przyjąć wartość 0, przez co $\gamma_R(D) \leq 3$.
Innymi słowy jest to sytuacja, w której największy stopień zewnętrzny wierzchołka jest niewiększy od 1 ($\Delta^+(D) \leq 1$).
- Dwa wierzchołki mają wartość równą 2, a pozostałe mają wartość równą 0: wariant odpowiadający przypadkowi 2 (poniżej).
- Jeden z wierzchołków ma przypisaną wartość 2, dwa mają przypisaną wartość równą 1, a pozostały wierzchołek ma przypisaną wartość równą 0: Łatwo zauważyć, że jeżeli wierzchołek o wartości funkcji dominowania rzymskiego równej 2 dominuje tylko samego siebie i wierzchołek o przypisanej wartości 0, to jest to przypadek równoważny z opisanym wyżej przypadkiem, gdzie wszystkie wierzchołki mają przypisaną wartość 1 i również spełniona jest własność $\Delta^+(D) \leq 1$.
Natomiast jeżeli wierzchołek o wartości funkcji dominowania rzymskiego równej 2 dominuje tylko samego siebie lub dominuje jeden z wierzchołków o przypisanej wartości 1, to sumaryczna wartość funkcji dominowania rzymskiego może zostać zmniejszona.

Przypadek 2.: Jasne jest, że jeżeli w D występują dwa wierzchołki u, v , które stanowią zbiór dominujący dla całego grafu, to można przypisać im wartość funkcji dominowania rzymskiego równą 2, a pozostałym wierzchołkom wartość 0, ponieważ każdy z nich jest dominowany przez u lub v .

Przypadek 3.: Jeżeli żaden z powyższych przypadków nie ma zastosowania, to można zauważyć, że aby liczba dominowania rzymskiego w grafie D mogła przyjąć wartość $\gamma_R(D)$, konieczne jest występowanie jednego wierzchołka, który będzie dominował przynajmniej

$n - 3$ inne wierzchołki oraz samego siebie. Wynika to z konieczności występowania jednego wierzchołka o wartości funkcji dominowania rzymskiego równej 2 oraz dwóch wierzchołków o wartości równej 1. Wierzchołek o przypisanej wartości 2 musi dominować przynajmniej wszystkie wierzchołki poza wierzchołkami o wartości równej 1.

Powyższe stwierdzenie oznacza, że jeżeli nie jest spełniony żaden z powyższych przypadków, to graf D musi spełniać własność $\Delta^+(D) \geq n - 3$, żeby jego liczba dominowania rzymskiego była równa 4. Warto również zauważyć, że w przypadku gdy $\Delta^+(D) < n - 3$, to $\gamma_R(D) > 4$.

□

Twierdzenie 3.3. [3] Każdy spójny graf G o liczbie wierzchołków $n \geq 6$ spełnia własność $\gamma_{Rg}(G) \geq 4$.

Dowód. Z lematu 3.2 wiadomo, że dla każdego grafu skierowanego D o $n > 4$ graf ten musi być zdominowany przez co najwyżej dwa wierzchołki lub jego stopień zewnętrzny musi spełniać $\Delta^+(D) \geq n - 3$, aby jego liczba dominowania rzymskiego była mniejsza lub równa 4. □

3.3.2. Klika

Twierdzenie 3.4. Dla cliki K_3 liczba gry w dominowanie rzymskie wynosi $\gamma_{Rg}(K_3) = 2$.

Dowód. W przypadku gry w dominowanie rzymskie na klicie K_3 Dominator zawsze ma do dyspozycji prostą strategię, która pozwala na zdominowanie całego grafu przez jeden wierzchołek, dla którego funkcja dominowania rzymskiego wynosi 2.

Dominator wybiera jeden z wierzchołków $v_1, v_2, v_3 \in K_3$ i orientuje jedną z sąsiadujących z nim krawędzi nieskierowanych tak, aby wychodziła od niego. Przyjmijmy bez straty ogólności, że wybiera wierzchołek v_1 i orientuje krawędź $\overrightarrow{v_1v_2}$.

Psuj ma wtedy możliwość zrobienia ruchu na jednej z dwóch pozostałych krawędzi nieskierowanych i każdą z nich może zorientować w jedną ze stron, więc musi wybrać jeden z czterech możliwych ruchów:

- ruch $\overrightarrow{v_1v_3}$ sprawia, że wierzchołek v_1 dominuje wszystkie wierzchołki w grafie, więc

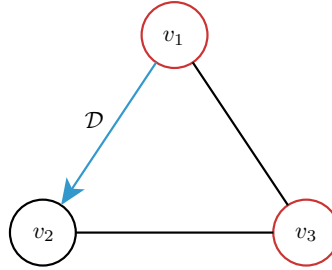
$$\sum_{v \in V(K_3)} f(v) = 2,$$

- ruch $\overrightarrow{v_3v_1}$ pozwala Dominatorowi na wykonanie ruchu $\overrightarrow{v_3v_2}$, który sprawia, że v_3 dominuje wszystkie wierzchołki jak wyżej,
- ruchy $\overrightarrow{v_2v_3}$ i $\overrightarrow{v_3v_2}$ pozostawiają Dominatorowi ruch $\overrightarrow{v_1v_3}$, który sprawia, że wierzchołek v_1 dominuje wszystkie wierzchołki jak wyżej.

Na tej podstawie można wywnioskować, że $\gamma_{Rg}(K_3) \leq 2$.

Jednocześnie łatwo zauważyć, że najmniejsza sumaryczna wartość funkcji dominowania rzymskiego dla wszystkich wierzchołków ($\sum_{v \in V(K_3)} f(v)$) wynosi 2, ponieważ jedyna funkcja dominowania rzymskiego, która nie przypisuje żadnemu z wierzchołków wartości 2 to taka, która przypisuje każdemu z trzech wierzchołków wartość 1, z czego wynika $\sum_{v \in V(K_3)} f(v) = 3$. W związku z tym $\gamma_{Rg}(K_3) \geq 2$, co w połączeniu z powyższym dowodzi równości $\gamma_{Rg}(K_3) = 2$. □

Lemat 3.5. [3] Dla cliki K_4 zachodzi $\gamma_{Rg}(K_4) \leq 3$.



Rys. 3.1. Dominator po zagranii ruchu $\overrightarrow{v_1 v_2}$ ma gwarancję, że wierzchołek v_1 albo v_3 (oznaczone na czerwono) będzie dominował wszystkie wierzchołki w grafie

Dowód. Oznaczmy wierzchołki $v_1, v_2, v_3, v_4 \in V(K_4)$. Dominator może przyjąć następującą strategię: najpierw wybiera jeden z wierzchołków. Można bez straty ogólności rozważyć wierzchołek v_1 . Dominator orientuje dowolną z krawędzi sąsiadujących z v_1 i orientuje ją tak, aby wychodziła z v_1 , np. $\overrightarrow{v_1 v_2}$. Następnie Psuj wykonuje swój ruch.

Łatwo zauważyć, że po ruchu Psuja Dominator ma możliwość wykonania przynajmniej jednego z dwóch ruchów: $\overrightarrow{v_1 v_3}$ lub $\overrightarrow{v_1 v_4}$. W ten sposób wierzchołek v_1 może przyjąć wartość $f(v_1) = 2$, a dwa wierzchołki dominowane przez niego wartości f równe 0. W tej sytuacji pozostałemu wierzchołkowi można przypisać wartość 1.

Przy użyciu tej strategii Dominator może zapewnić, że $\gamma_{Rg}(K_4)$ będzie zawsze nie większa od 3. \square

Lemat 3.6. [3] Dla klikki K_4 zachodzi $\gamma_{Rg}(K_4) \geq 3$.

Dowód. Niech Psuj gra według następującej strategii: dla każdego wierzchołka $v_x \in V$ oblicz $d_x = d^+(v_x) - d^-(v_x)$, gdzie d^+ oznacza stopień wychodzący wierzchołka, a d^- stopień wchodzący.

- Jeżeli istnieje wierzchołek $v_i \in V$, dla którego $d_i = 2$, i możliwe jest wykonanie ruchu $\overrightarrow{uv_i}$, gdzie $u \in V, u \neq v_i$, to wykonaj ten ruch.
- W innym wypadku, jeżeli istnieje v_i , dla którego $d_i = 1$, i możliwe jest wykonanie ruchu $\overrightarrow{uv_i}$, to wykonaj ten ruch.
- Jeżeli nie ma miejsca żadna z powyższych sytuacji, to wykonaj dowolny ruch.

Powyższa strategia zapewnia Dominatorowi możliwość zorientowania przynajmniej jednej krawędzi wchodzącej do każdego wierzchołka, tzn. sprawia, że żaden z wierzchołków nie będzie dominował całego zbioru V . Wobec tego jeden wierzchołek $v_1 \in V$ może dominować co najwyżej dwa inne wierzchołki $v_2, v_3 \in V$, wtedy $f(v_1) = 2, f(v_2) = f(v_3) = 0$ oraz $f(v_4) = 1, v_4 \in V \setminus \{v_1, v_2, v_3\}$, co dowodzi twierdzenia. \square

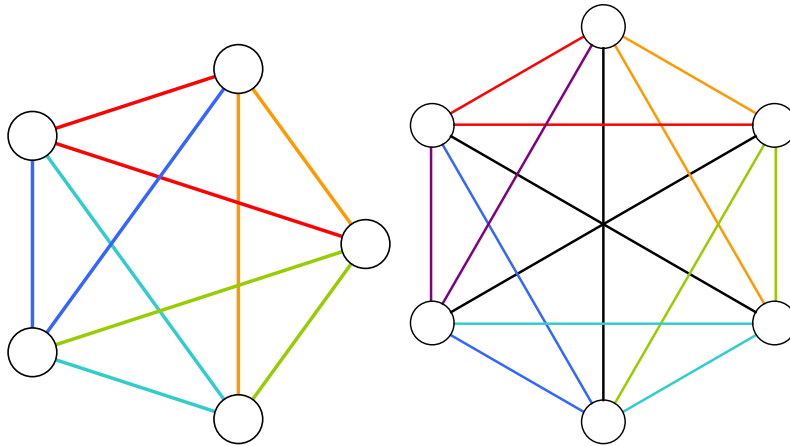
Lemat 3.7. [5] Dla każdej klikki K_n , gdzie $n \geq 5$, można wyznaczyć n parami rozłącznych ścieżek długości 2.

Dowód. Niech wierzchołki będą oznaczone jako $v_0, v_1, \dots, v_{n-1} \in V(K_n)$. Wtedy możliwe jest zdefiniowanie rodziny ścieżek P_K jako $P_{Ki} = \{v_{((i+1) \bmod n)}, v_i, v_{((i+2) \bmod n)}\}$, gdzie $x \bmod y$ oznacza resztę z dzielenia x przez y . \square

Twierdzenie 3.8. [3] Dla klik K_n , gdzie $n \in 4, 5$, liczba gry w dominowanie rzymskie wynosi $\gamma_{Rg}(K_n) = 3$.

Dowód. Równość $\gamma_{Rg}(K_4) = 3$ jest bezpośrednią konsekwencją lematów 3.5 i 3.6.

Dowód równości $\gamma_{Rg}(K_5) = 3$ można podzielić na dwa przypadki.



Rys. 3.2. Ścieżki P_2 w grafach K_5 i K_6 , skonstruowane zgodnie z metodą w lemacie 3.7 zostały oznaczone kolorami

Przypadek 1. $\gamma_{Rg}(K_5) \geq 3$: W przypadku K_5 można wykorzystać zależność udowodnioną w lemacie 3.7 i zilustrowaną na rysunku 3.2. Psuj może przyjąć następującą strategię: jeżeli Dominator wykona ruch na jednej z wyznaczonych ścieżek, który orientuje krawędź jako wychodzącą z centralnego (środkowego) wierzchołka ścieżki, to należy, jeżeli to możliwe, zorientować drugą krawędź ścieżki tak, żeby wchodziła do centralnego wierzchołka. W innym wypadku można wykonać dowolny ruch.

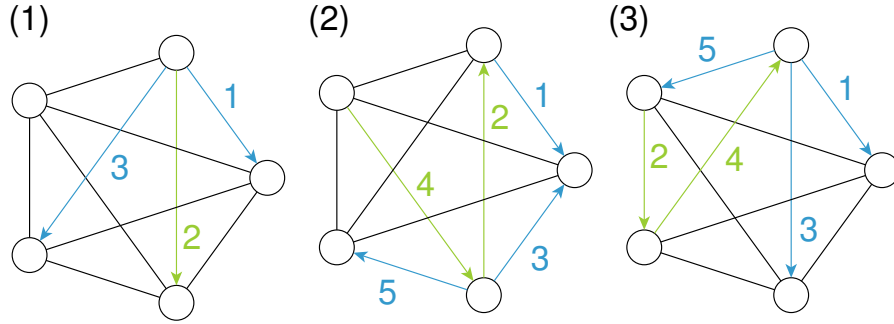
W ten sposób każdy z wierzchołków będzie z końcem gry miał przynajmniej jeden łuk wchodzący, więc nie będzie mógł samodzielnie zdominować pozostałych wierzchołków. Stąd jasno wynika, że $\gamma_{Rg}(K_5) > 2 \implies \gamma_{Rg}(K_5) \geq 3$.

Przypadek 2. $\gamma_{Rg}(K_5) \leq 3$: Rozważania nad tym przypadkiem można rozpocząć od zauważenia następującej zależności: oznaczmy graf powstały w wyniku zorientowania krawędzi w K_5 jako G' . Jeżeli dowolny wierzchołek $v_x \in V(G')$ będzie sąsiedował z trzema wychodzącymi łukami (czyli $d^+(v_x) = 3$), to liczba dominowania rzymskiego tego grafu będzie wynosiła co najwyżej 3.

Jest tak, ponieważ wierzchołek v_x może mieć przypisaną wartość $f(v_x) = 2$. Wierzchołki, do których wychodzą jego łuki, są wówczas przez niego dominowane, więc mogą mieć wartość 0, a pozostały wierzchołek może mieć wartość 1, co daje $\sum_{v \in V} f(v) = 3$.

Dominator może użyć następującej strategii, aby wymusić taką sytuację: w pierwszym ruchu należy zorientować dowolną krawędź $\overrightarrow{v_1 v_2}$. Następnie

- jeżeli Psuj zorientuje krawędź wychodzącą z wierzchołka v_1 , np. $\overrightarrow{v_1 v_3}$, to należy zorientować kolejną krawędź wychodzącą z v_1 , np. $\overrightarrow{v_1 v_4}$, co daje $d^+(v_1) = 3$. (rys. 3.3, (1))
- Jeżeli Psuj zorientuje krawędź wchodzącą do wierzchołka v_1 , np. $\overrightarrow{v_3 v_1}$, to należy zorientować kolejną krawędź wychodzącą z v_3 , np. $\overrightarrow{v_3 v_2}$. W tym momencie $d^+(v_3) = 2$ i pozostają dwie nieskierowane krawędzie sąsiadujące z v_3 , więc w kolejnym ruchu Dominator ma możliwość zorientowania przynajmniej jednej z nich w kierunku od v_3 , co daje $d^+(v_3) = 3$. (rys. 3.3, (2))
- W innym wypadku w kolejnych ruchach Dominator może zorientować przynajmniej dwie kolejne krawędzie wychodzące z v_1 , co daje $d^+(v_1) = 3$. (rys. 3.3, (3))



Rys. 3.3. Na rysunku przedstawiono strategię Dominatora, pozwalającą uzyskać $d^+(v) = 3$. Wybrany wariant jest zależny od ruchów wykonywanych przez Psuja. Ruchy ponumerowane są wg kolejności wykonania. Ruchy Dominatora oznaczono kolorem niebieskim, a ruchy Psuja kolorem zielonym

Niezależnie od ruchów Psuja, Dominator jest w stanie doprowadzić do sytuacji, w której stopień wychodzący jednego z wierzchołków będzie wynosił przynajmniej 3, co implikuje, że liczba gry w dominowanie rzymskie na K_5 zawsze będzie wynosiła co najwyżej 3.

Zbierając powyższe dwa przypadki mamy $3 \leq \gamma_{Rg}(K_5) \leq 3 \implies \gamma_{Rg}(K_5) = 3$.

□

Zbiór 2-dominujący w grafie G to taki zbiór $D \subseteq V(G)$, że każdy wierzchołek G jest dominowany przez przynajmniej dwa z jego elementów. Liczba 2-dominowania grafu oznacza moc najmniejszego możliwego zbioru 2-dominowania w grafie i jest oznaczana jako $\gamma_2(G)$.

Lemat 3.9. [1] Dla każdego grafu prostego G zachodzi $\gamma_{Rg}(G) \leq 2\gamma_2(G)$.

Dowód. Rozważania na temat powiązania liczby gry w dominowanie rzymskie z liczbą 2-dominowania można rozpocząć od przeanalizowania prostszego przypadku standardowej gry w dominowanie.

Niech D_2 będzie zbiorem 2-dominującym w grafie G . Wtedy każdy z wierzchołków w $V(G) \setminus D_2$ ma przynajmniej dwa sąsiadujące wierzchołki w D_2 . Niech $v \in V(G) \setminus D_2$, $d_1, d_2 \in D_2$ i $vd_1, vd_2 \in E(G)$.

Dominator może przyjąć następującą strategię: jeżeli Psuj zagra jeden z ruchów $\overrightarrow{vd_1}, \overrightarrow{vd_2}$, to należy zagrać ruch na drugiej z tej pary krawędzi: $\overrightarrow{d_1v}$ lub $\overrightarrow{d_2v}$. W innym wypadku, jeżeli to możliwe, należy grać ruchy łączące wierzchołek ze zbioru D_2 z wierzchołkiem z $V(G) \setminus D_2$.

W ten sposób Dominator gwarantuje sobie, że zbiór D_2 będzie zbiorem dominującym w wynikowym grafie. Stąd $\gamma_g(G) \leq \gamma_2(G)$ ¹.

Kolejną istotną obserwacją jest to, że w relacji dominowania rzymskiego wierzchołki, dla których $f(v) = 2$ funkcjonują analogicznie do wierzchołków w zwykłej relacji dominowania, tzn. dominują całe swoje domknięte sąsiedztwo. W związku z tym jeżeli D jest zbiorem dominującym w grafie G , to funkcja f , przypisująca wartość 2 wszystkim wierzchołkom w D i wartość 0 wszystkim pozostałym, będzie poprawną funkcją dominowania rzymskiego w grafie G . Stąd $\gamma_{Rg}(G) \leq 2\gamma_g(G)$.

Z powyższych nierówności wynika $\gamma_{Rg}(G) \leq 2\gamma_g(G) \leq 2\gamma_2(G)$.

□

Twierdzenie 3.10. [3] Dla każdej klikki K_n dla przy $n \geq 6$ zachodzi $\gamma_{Rg}(K_n) = 4$.

Dowód. Z lematu 3.9 zachodzi własność $\gamma_{Rg}(G) \leq 2\gamma_2(G)$. W przypadku klikki najmniejszy zbiór 2-dominujący liczy 2 elementy, ponieważ każdy z nich dominuje wszystkie wierzchołki, więc razem dominują każdy z wierzchołków dwukrotnie. Stąd w przypadku klikki $\gamma_{Rg}(K_n) \leq 2 \cdot 2 = 4$.

¹gdzie $\gamma_g(G)$ jest liczbą gry w dominowanie dla G

Jednocześnie z twierdzenia 3.3 dla spójnych grafów o liczbie wierzchołków $n \geq 6$ zachodzi $\gamma_{Rg}(G) \geq 4$.

Z powyższego dla K_n , gdzie $n \geq 6$ mamy $4 \leq \gamma_{Rg}(K_n) \leq 4 \implies \gamma_{Rg}(K_n) = 4$. \square

3.3.3. Gwiazda

Twierdzenie 3.11. [3] W gwiazdzie $K_{1,n-1}$ liczba gry w dominowanie rzymskie jest równa $\lfloor \frac{n+3}{2} \rfloor$.

Dowód. Rozważając zagadnienie dominowania w kontekście gwiazd² można dostrzec, że jest w nich tylko jeden wierzchołek, który dominuje więcej niż dwa wierzchołki w całym grafie. Jest to centrum gwiazdy. Fakt ten jest o tyle istotny w zagadnieniu dominowania rzymskiego, że dla liczby dominowania nie ma znaczenia, czy wierzchołek v dominuje pojedynczy inny wierzchołek u ($f(v) = 2, f(u) = 0$), czy oba wierzchołki dominują tylko same siebie ($f(v) = f(u) = 1$), ponieważ ich sumaryczna wartość funkcji dominowania w obu wypadkach będzie wynosić 2.

W związku z tym w przypadku gwiazdy orientowanie krawędzi w kierunku centrum nie będzie zmniejszało liczby dominowania. Przy założeniu, że centrum będzie miało wartość funkcji dominowania rzymskiego równą 2, każdy łuk zorientowany od centrum oznacza jeden pozostały wierzchołek, dla którego wartość funkcji dominowania będzie wynosić 0, natomiast na każdy łuk skierowaną do centrum będzie przypadał wierzchołek o wartości funkcji dominowania równej 1.

Wobec tego Dominator będzie starał się zorientować jak najwięcej krawędzi w kierunku od centrum, aby zmniejszyć liczbę dominowania, natomiast Psuj będzie próbował zorientować jak najwięcej krawędzi w kierunku centrum gwiazdy.

Jako że Dominator jest graczem rozpoczynającym rozgrywkę, będzie on w stanie zorientować $\lfloor \frac{n-1}{2} \rfloor$ krawędzi na zewnątrz, natomiast Psuj zorientuje $\lfloor \frac{n-1}{2} \rfloor$ krawędzi do wewnątrz. Stąd wartość funkcji dominowania dla $\lfloor \frac{n-1}{2} \rfloor$ wierzchołków będzie wynosiła 0, dla $\lfloor \frac{n-1}{2} \rfloor$ wierzchołków będzie wynosiła 1, a dla pozostałego, centralnego wierzchołka będzie wynosiła 2.

$$\begin{aligned} \gamma_{Rg}(K_{1,n-1}) &= 2 + \left\lfloor \frac{n-1}{2} \right\rfloor \cdot 1 + \left\lceil \frac{n-1}{2} \right\rceil \cdot 0 \\ &= 2 + \left\lfloor \frac{n-1}{2} \right\rfloor = \left\lfloor \frac{n+3}{2} \right\rfloor \end{aligned}$$

\square

3.3.4. Ścieżka

Lemat 3.12. [3] Dla $n \in [4, 14]$ zachodzi $\gamma_{Rg}(P_n) = n - 1 - \lfloor \frac{n-4}{5} \rfloor$.

Powyższy lemat został udowodniony przez Bahremandpourea i in. [3] Przedstawiony tam dowód składa się z wielu poddowodów dla poszczególnych wartości n i nie jest tu przytaczany ze względu na swoją objętość. Jest on natomiast kluczowy dla udowodnienia poniższego twierdzenia.

Twierdzenie 3.13. [3] Dla $n \geq 4$ zachodzi $\gamma_{Rg}(P_n) \leq n - 1 - \lfloor \frac{n-4}{5} \rfloor$.

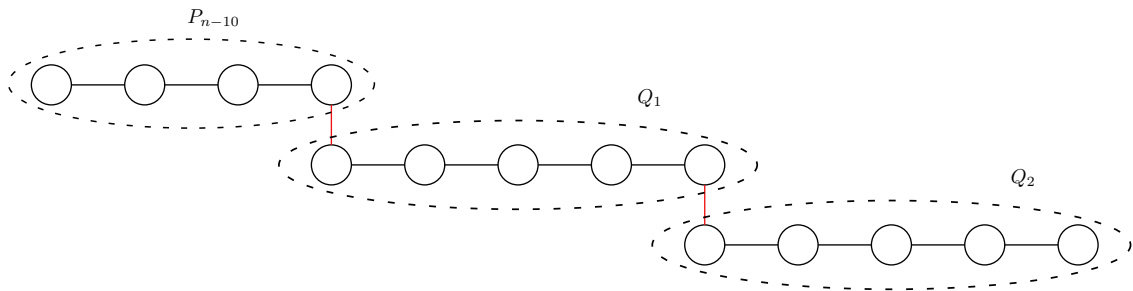
Dowód. Wierzchołki ścieżki P_n będą oznaczone kolejnymi liczbami całkowitymi: v_1, v_2, \dots, v_n . Niech w ścieżce P_n wyróżnione będą trzy części:

- podścieżka P_{n-10} zawierająca wierzchołki v_1, \dots, v_{n-10} ,

²grafów dwudzielnych $K_{1,n-1}$

- podścieżka Q_1 zawierająca wierzchołki v_{n-9}, \dots, v_{n-5} ,
- podścieżka Q_2 zawierająca wierzchołki v_{n-4}, \dots, v_n .

Należy zauważyć, że przy takim podziale krawędzie $v_{n-10}v_{n-9}$ oraz $v_{n-5}v_{n-4}$ nie należą do żadnej z podścieżek.

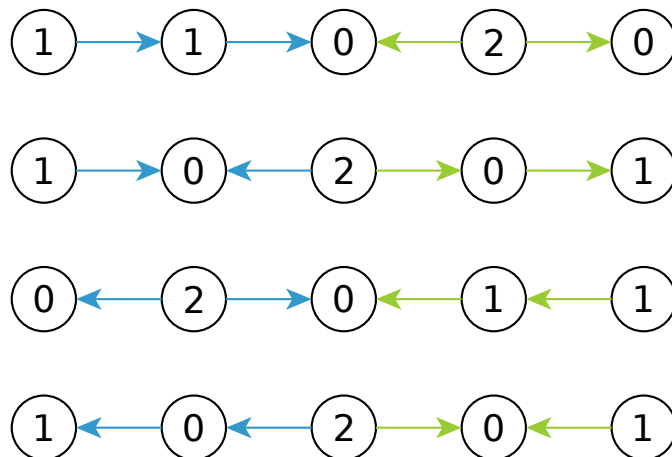


Rys. 3.4. Ścieżka P_n (przykładowo zaprezentowana została P_{14}) dzielona jest na podścieżki P_{n-10} , Q_1 i Q_2 . Łączące je krawędzie, nienależące do żadnej z podścieżek, zostały oznaczone kolorem czerwonym

Na tej podstawie można wywnioskować, że $\gamma_{Rg}(K_3) \leq 2$.

Podścieżki Q_1 i Q_2 są izomorficzne do ścieżki P_5 . Przy założeniu, że Psuj wykonuje pierwszy ruch w grze w dominowanie rzymskie na ścieżce P_5 , Dominator może przyjąć następującą strategię:

- Jeżeli Psuj wykonał ruch na jednej z krawędzi $\{v_1v_2, v_4v_5\}$, to Dominator wykonuje ruch na drugiej z tych krawędzi w tym samym kierunku³.
- Jeżeli Psuj wykonał ruch na jednej z krawędzi $\{v_2v_3, v_3v_4\}$, to Dominator wykonuje ruch na drugiej z tych krawędzi w przeciwnym kierunku.



Rys. 3.5. W grze w dominowanie rzymskie na ścieżce P_5 przy użyciu strategii z dowodu twierdzenia 3.13 zawsze uzyskiwany jest jeden z powyższych rezultatów. W wierzchołkach podano wartości najmniejszej funkcji dominowania rzymskiego. Można zauważyć, że w każdym z tych przypadków liczba dominowania rzymskiego wynosi 4

Łatwo zauważyć, że używając tej strategii Dominator zapewnia, że suma funkcji dominowania rzymskiego będzie równa 4, co zostało zilustrowane na rys. 3.5. Oznacza to, że $\gamma_{Rg}(P_5) \leq 4$.

Dowód twierdzenia jest oparty na indukcji przy założeniu, że zależność jest spełniona dla P_{n-10} , czyli prawdą jest, że $\gamma_{Rg}(P_{n-10}) \leq n - 11 - \lfloor \frac{n-14}{5} \rfloor$. Na podstawie lematu 3.12 wiadomo, że jest tak dla $n \in [14, 24]$.

Niech Dominator przyjmie strategię:

³tj. jeżeli Psuj wykonał ruch $\overrightarrow{v_i v_{i+1}}$, to Dominator powinien wykonać ruch $\overrightarrow{v_j v_{j+1}}$

- Dopóki Psuj wykonuje ruchy na podścieżce P_{n-10} , to Dominator gra na niej optymalną strategią.
- Jeżeli Psuj wykona ruch na podścieżce Q_1 lub Q_2 , to Dominator wykonuje na niej ruch zgodnie z przedstawioną powyżej strategią dla P_5 .
- Jeżeli Psuj wykona ruch na jednej ze ścieżek łączących podścieżki ($\{v_{n-10}v_{n-9}, v_{n-5}v_{n-4}\}$), to Dominator wykonuje na drugiej z nich ruch w dowolnym kierunku.

Liczba dominowania całego grafu jest nie większa od sumy liczb dominowania podścieżek, ponieważ połączenie dwóch podścieżek w wynikowym grafie nie sprawia, że którykolwiek z wierzchołków nie jest dominowany.

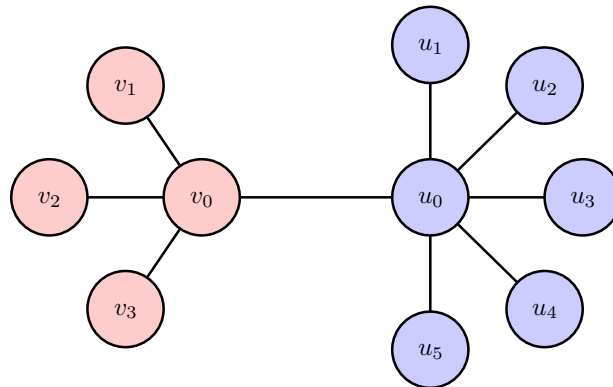
Wobec tego

$$\begin{aligned} \gamma_{Rg}(P_n) &\leq \gamma_R(P'_{n-10}) + \gamma_R(Q'_1) + \gamma_R(Q'_2) \\ &\leq n - 11 - \left\lfloor \frac{n-14}{5} \right\rfloor + 4 + 4 \\ &= n - 1 - \left\lfloor \frac{n-4}{5} \right\rfloor \end{aligned}$$

gdzie P'_{n-10} , Q'_1 , Q'_2 to grafy skierowane powstałe po zakończeniu gry w dominowanie rzymskie na odpowiednio P_{n-10} , Q_1 oraz Q_2 . □

3.3.5. Podwójna gwiazda

Podwójna gwiazda jest grafem powstałym z połączenia centrów dwóch gwiazd. Oznaczana jest jako $DS_{p,q}$ co jest interpretowane jako grafy $K_{1,p}$ i $K_{1,q}$, których środkowe wierzchołki zostały połączone krawędzią. Warto zauważyć, że grafy $DS_{p,q}$ oraz $DS_{q,p}$ są izomorficzne.



Rys. 3.6. Podwójna gwiazda $DS_{3,5} = DS_{5,3}$ ze składowymi klikami oznaczonymi różnymi kolorami

Twierdzenie 3.14. [3] W podwójnej gwieżdzie $DS_{1,q}$, gdzie $q \geq 2$, zachodzi $\gamma_{Rg}(DS_{1,q}) = 2 + \lfloor \frac{n}{2} \rfloor$.

Dowód. Wiadomo że $n = q + 3$, ponieważ liczba wierzchołków jest równa liczbie liści w większej składowej wraz z jej centrum i dwoma wierzchołkami stanowiącymi drugą składową. Stąd $q = n - 3$.

Niech centrum mniejszej składowej będzie oznaczone jako v_0 , a jej liść jako v_1 . Dodatkowo niech centrum większej składowej będzie oznaczone jako u_0 , a jej liście jako u_1, \dots, u_q .

Dominator może przyjąć strategię, polegającą na orientowaniu, dopóki to możliwe, krawędzi pomiędzy $\{u_0\}$ a $\{v_0, u_1, \dots, u_q\}$ w kierunku na zewnątrz od u_0 . Z uwagi na to, że wykonuje

pierwszy ruch w grze, będzie w stanie zorientować w ten sposób przynajmniej $\lceil \frac{q+1}{2} \rceil$ krawędzi. Jeżeli więc $f(u_0) = 2$, to dla wszystkich wierzchołków, do których zostały skierowane krawędzie, najmniejsza funkcja dominowania rzymskiego będzie wynosić 0.

Jednocześnie co najwyżej $\lfloor \frac{q+1}{2} \rfloor$ krawędzi będzie skierowanych z $\{v_0, u_1, \dots, u_q\}$ do $\{u_0\}$ i, co za tym idzie, najmniejsza funkcja dominowania rzymskiego dla ich wierzchołków źródłowych będzie wynosić 1. Dodatkowo $f(v_1) = 1$.

Sumarycznie przy tej strategii najmniejsza funkcja dominowania rzymskiego przyjmuje wartość:

$$\sum f_{\min}(DS_{1,q}) \leq 2 + \left(\left\lfloor \frac{q+1}{2} \right\rfloor + 1 \right) \cdot 1 + \left\lceil \frac{q+1}{2} \right\rceil \cdot 0 = 3 + \left\lfloor \frac{q+1}{2} \right\rfloor,$$

co wyrażone jako funkcja liczby wierzchołków daje:

$$\sum f_{\min}(DS_{1,q}) \leq 3 + \left\lfloor \frac{n-2}{2} \right\rfloor = 2 + \left\lfloor \frac{n}{2} \right\rfloor.$$

Stąd $\gamma_{Rg}(DS_{1,q}) \leq 2 + \lfloor \frac{n}{2} \rfloor$.

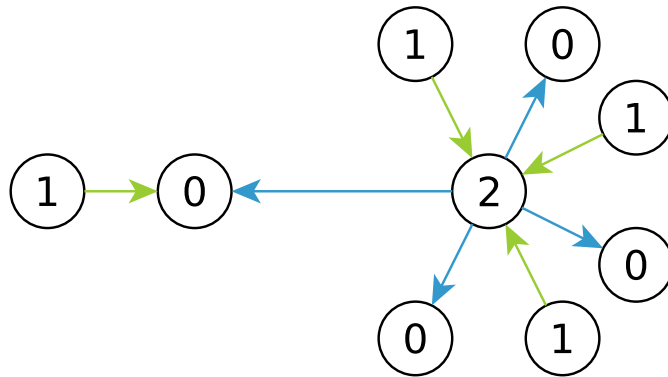
Psuj może zastosować podobną strategię, polegającą na orientowaniu krawędzi z $\{v_0, u_1, \dots, u_q\}$ do $\{u_0\}$, dopóki ma taką możliwość. Wówczas zapewnia, że przynajmniej $\lfloor \frac{q+1}{2} \rfloor$ krawędzi będzie skierowanych do wierzchołka u_0 i, w konsekwencji, co najmniej $\lfloor \frac{q+1}{2} \rfloor$ wierzchołków ze zbioru $\{v_0, u_1, \dots, u_q\}$ będzie miało najmniejszą wartość funkcji dominowania rzymskiego równą 1.

Wówczas najmniejsza funkcja dominowania rzymskiego przybierze wartość:

$$\sum f_{\min}(DS_{1,q}) \geq 2 + \left(\left\lfloor \frac{q+1}{2} \right\rfloor + 1 \right) \cdot 1 + \left\lceil \frac{q+1}{2} \right\rceil \cdot 0 = 3 + \left\lfloor \frac{q+1}{2} \right\rfloor,$$

co, w wyniku rozumowania analogicznego do przypadku strategii Dominatora, implikuje $\gamma_{Rg}(DS_{1,q}) \leq 2 + \lfloor \frac{n}{2} \rfloor$.

Z wywiedzionych powyżej nierówności wynika, że $\gamma_{Rg}(DS_{1,q}) = 2 + \lfloor \frac{n}{2} \rfloor$. □



Rys. 3.7. Przykładowa gra w dominowanie rzymskie rozegrana na $DS_{1,6}$ według strategii z dowodu twierdzenia 3.14. Ruchy Dominatora zostały oznaczone na niebiesko, a Psuja na zielono. W wierzchołkach podane zostały odpowiadające im wartości najmniejszej funkcji dominowania rzymskiego

Twierdzenie 3.15. [3] Dla każdej podwójnej gwiazdy $DS_{p,q}$, gdzie $2 \leq p \leq q$ zachodzi

$$\left\lfloor \frac{n}{2} \right\rfloor + 3 \leq \gamma_{Rg}(DS_{p,q}) \leq \left\lfloor \frac{n+1}{2} \right\rfloor + 3.$$

Dowód. W grafie $DS_{p,q}$ liczba wierzchołków wynosi $n = p + q + 2$.

Górne ograniczenie Niech Dominator przyjmie następującą strategię: w pierwszym ruchu orientuje krawędź w gwieździe $K_{1,q}$ od jej centrum do liścia. Następnie:

- jeżeli Psuj zorientuje jedną z krawędzi $K_{1,q}$, to Dominator orientuje inną krawędź tego grafu od centrum do liścia, jeżeli to możliwe. W przeciwnym wypadku orientuje jedną z krawędzi $K_{1,p}$ w kierunku od centrum do liścia, jeżeli jest taka możliwość;
- jeżeli Psuj zorientuje jedną z krawędzi $K_{1,p}$, to Dominator orientuje inną krawędź tego grafu od centrum do liścia, jeżeli to możliwe. W przeciwnym wypadku orientuje jedną z krawędzi $K_{1,q}$ w kierunku od centrum do liścia, jeżeli jest taka możliwość;
- jeżeli Psuj zorientuje jedną krawędź łączącą centra gwiazd, to Dominator orientuje do zewnątrz krawędź w $K_{1,q}$ lub, jeżeli nie jest to możliwe, krawędź w $K_{1,p}$;
- jeżeli żaden z powyższych ruchów nie jest możliwy, to Dominator orientuje dowolnie krawędź łączącą centra gwiazd.

Grając zgodnie z tą strategią, Dominator zapewnia, że w gwieździe $K_{1,q}$ będzie co najwyżej $\lfloor \frac{q}{2} \rfloor$ łuków zwróconych w kierunku od liści do centrum. Wartość najmniejszej funkcji dominowania rzymskiego dla każdego z tych liści będzie wynosić 1, ponieważ nie ma żadnego innego wierzchołka, który je dominuje. Wówczas wierzchołek centralny może mieć wartość $f = 2$, a pozostałe liście tej gwiazdy wartości równe 0, ponieważ są dominowane przez wierzchołek centralny. Wtedy sumaryczna wartość najmniejszej funkcji dominowania rzymskiego dla $K_{1,q}$ jest nie większa od $2 + \lfloor \frac{q}{2} \rfloor$.

W przypadku $K_{1,p}$ Psuj ma możliwość zagrania pierwszego ruchu, więc zgodnie z powyższą strategią maksymalna liczba łuków skierowanych do centrum będzie wynosiła $\lceil \frac{p}{2} \rceil$. Kierując się analogicznym do przypadku $K_{1,q}$ rozumowaniem, sumaryczna wartość najmniejszej funkcji dominowania rzymskiego dla $K_{1,p}$ jest więc nie większa od $2 + \lceil \frac{p}{2} \rceil$.

Jako że najmniejsza funkcja dominowania rzymskiego grafu złożonego z dwóch podgrafów jest nie większa od sumy najmniejszych funkcji dominowania rzymskiego tych podgrafów, to z powyższego wynika, że

$$\begin{aligned} \gamma_{Rg}(DS_{p,q}) &\leq 2 + \lfloor \frac{q}{2} \rfloor + 2 + \lceil \frac{p}{2} \rceil = 4 + \lfloor \frac{q}{2} \rfloor + \lceil \frac{p+1}{2} \rceil \\ &\leq 4 + \left\lfloor \frac{q+p+1}{2} \right\rfloor = 4 + \left\lfloor \frac{n-1}{2} \right\rfloor \\ &= 3 + \left\lfloor \frac{n+1}{2} \right\rfloor. \end{aligned}$$

Dolne ograniczenie Dowód wyrażenia na dolne ograniczenie przebiega podobnie do powyższego dowodu na górne ograniczenie. Niech Psuj przyjmie strategię, polegającą na odpowiadaniu na ruchy Dominatora:

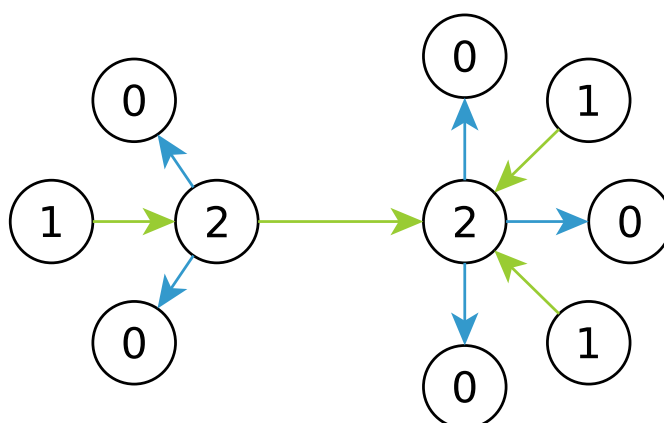
- jeżeli Dominator orientuje krawędź w podgrafie $K_{1,q}$ lub krawędź łączącą centra, to Psuj orientuje krawędź w podgrafie $K_{1,q}$ w kierunku od centrum. Jeżeli ten ruch nie jest możliwy, to orientuje krawędź w $K_{1,p}$ w kierunku od centrum;
- jeżeli Dominator orientuje krawędź w podgrafie $K_{1,p}$, to Psuj orientuje krawędź w podgrafie $K_{1,p}$ w kierunku od centrum. Jeżeli ten ruch nie jest możliwy, to orientuje krawędź w $K_{1,q}$ w kierunku od centrum;

- jeżeli żaden z powyższych ruchów nie jest możliwy, to Psuj orientuje dowolnie krawędź łączącą centra.

W ten sposób w podgrafie $K_{1,q}$ jest przynajmniej $\lfloor \frac{q}{2} \rfloor$ łuków skierowanych do centrum, a w podgrafie $K_{1,p}$ przynajmniej $\lfloor \frac{p}{2} \rfloor$ takich łuków. W najmniejszej funkcji dominowania rzymskiego centra będą miały wartość równą 2, i będzie przynajmniej $\lfloor \frac{q}{2} \rfloor + \lfloor \frac{p}{2} \rfloor$ wierzchołków o wartości 1, a pozostałe wierzchołki będą miały wartość 0. Stąd, podobnie jak dla powyższego wprowadzenia:

$$\begin{aligned} \gamma_{Rg}(DS_{p,q}) &\geq 2 \cdot 2 + \lfloor \frac{q}{2} \rfloor + \lfloor \frac{p}{2} \rfloor \geq 4 + \lfloor \frac{q+p}{2} \rfloor \\ &= 4 + \lfloor \frac{n-2}{2} \rfloor = 3 + \lfloor \frac{n}{2} \rfloor. \end{aligned}$$

□



Rys. 3.8. Przykładowa gra w dominowanie rzymskie rozegrana na $DS_{3,5}$ wg strategii z dowodu tw. 3.15

3.3.6. Drzewo

Twierdzenie 3.16. [3] Dla każdego drzewa T , gdzie $n(T) \geq 2$, liczba gry w dominowanie rzymskie jest ograniczona od dołu

$$\lfloor \frac{n}{2} \rfloor + 1 \leq \gamma_{Rg}(T).$$

Dowód. Przeprowadzimy dowód indukcyjny. Na początku zostaną rozważone przypadki drzew o dwóch i trzech wierzchołkach. Łatwo zauważyć, że w obu tych przypadkach drzewo jest jednocześnie ścieżką.

W przypadku $n(T) = 2$ w trakcie gry w dominowanie rzymskie zostanie wykonany tylko jeden ruch. Wtedy jeden z wierzchołków, który dominuje drugi wierzchołek, może przyjąć wartość funkcji dominowania rzymskiego równą 2, a pozostały wierzchołek wartość 0. Alternatywnie oba wierzchołki mogą przyjąć wartość 1. Obie funkcje dominowania rzymskiego są najmniejsze i ich wartość jest równa 2. Stąd $\lfloor \frac{n}{2} \rfloor + 1 \leq \gamma_{Rg}(T) \iff 2 \leq 2$, więc nierówność postawiona w twierdzeniu jest spełniona.

W przypadku $n(T) = 3$ Psuj może przyjąć strategię, polegającą na zorientowaniu jedynej nieskierowanej krawędzi pozostałej po ruchu Dominatora w tym samym kierunku, w którym Dominator zwrócił pierwszą krawędź. Konkretnie ruchy zostały przedstawione w tabeli 3.1. Łatwo

zauważyć, że grając zgodnie z tą strategią, Psuj zapewnia, że żaden pojedynczy wierzchołek nie będzie w stanie zdominować całego grafu, więc najmniejsza funkcja dominowania rzymskiego przyjmie wartość 3. Stąd $\lceil \frac{n}{2} \rceil + 1 \leq \gamma_{Rg}(T) \iff 3 \leq 3$, więc w przypadku $n(T) = 3$ nierówność z twierdzenia również jest prawdziwa.

Ruch Dominatora	Ruch Psuja
$\overrightarrow{v_1 v_2}$	$\overrightarrow{v_2 v_3}$
$\overrightarrow{v_2 v_1}$	$\overrightarrow{v_3 v_2}$
$\overrightarrow{v_2 v_3}$	$\overrightarrow{v_1 v_2}$
$\overrightarrow{v_3 v_2}$	$\overrightarrow{v_2 v_1}$

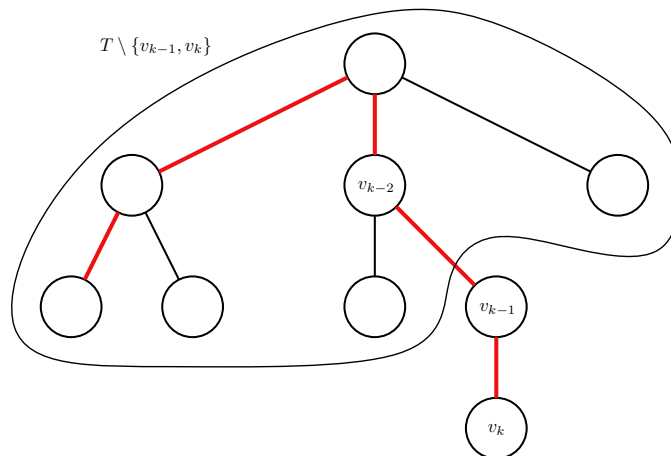
Tabela 3.1. Optymalna strategia Psuja na drzewie o 3 wierzchołkach izomorficznym do P_3

Z powyższego wiadomo, że twierdzenie jest prawdziwe dla $n \leq 3$. Niech w dalszej części dowodu $n(T) \geq 4$. Niech również założenie $\lceil \frac{n-2}{2} \rceil + 1 \leq \gamma_{Rg}(T)$ będzie przyjęte jako prawdziwe.

W każdym drzewie można wyznaczyć najdłuższą ścieżkę o długości równej średnicy tego drzewa. Wierzchołki najdłuższej ścieżki w drzewie T będą oznaczone jako v_1, v_2, \dots, v_k , gdzie k to średnica T . Wiadomo, że v_1 oraz v_k są liśćmi drzewa T , ponieważ w przeciwnym wypadku możliwe byłoby przedłużenie tej ścieżki, więc nie byłaby najdłuższą ścieżką w T . Można zauważyć, że może istnieć wiele najdłuższych ścieżek o równej długości. W szczególności v_{k-1} może sąsiadować z więcej niż jednym liściem.

Rozważane są dwa przypadki:

Przypadek $d(v_{k-1}) = 2$: W tym przypadku v_k jest jedynym liściem sąsiadującym z v_{k-1} , Psuj może przyjąć następującą strategię: dopóki Dominator wykonuje ruchy na grafie indukowanym na T przez wierzchołki $V(T) \setminus \{v_{k-1}, v_k\}$, to Psuj odpowiada według optymalnej strategii dla tego grafu. Jeżeli Dominator wykona ruch na jednej z krawędzi $v_{k-2}v_{k-1}, v_{k-1}v_k$, to Psuj wykonuje ruch na drugiej z tych krawędzi, orientując ją w tym samym kierunku, podobnie do strategii przedstawionej powyżej dla $n(T) = 3$ (konkretne ruchy przedstawiono w tabeli 3.2).



Rys. 3.9. Najdłuższa ścieżka w drzewie T w przypadku $d(v_{k-1}) = 2$

Ruch Dominatora	Ruch Psuja
$\overrightarrow{v_{k-2} v_{k-1}}$	$\overrightarrow{v_{k-1} v_k}$
$\overrightarrow{v_{k-1} v_{k-2}}$	$\overrightarrow{v_k v_{k-1}}$
$\overrightarrow{v_{k-1} v_k}$	$\overrightarrow{v_{k-2} v_{k-1}}$
$\overrightarrow{v_k v_{k-1}}$	$\overrightarrow{v_{k-1} v_{k-2}}$

Tabela 3.2. Optymalna strategia Psuja na ścieżce $v_{k-2}v_{k-1}v_k$ drzewa T izomorficznej do P_3

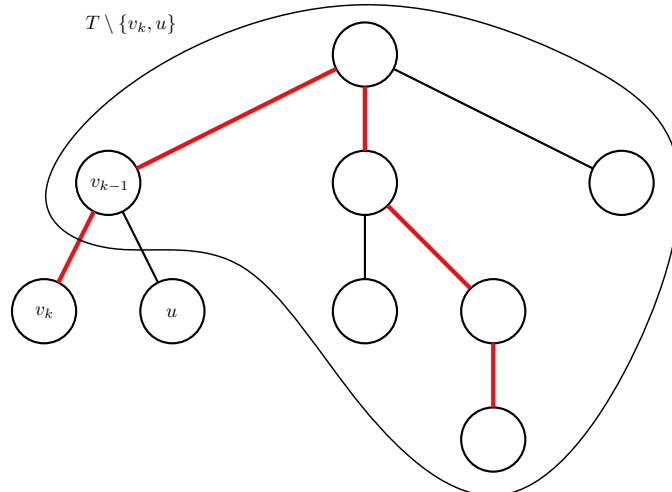
Grając wg powyższej strategii, Psuj zapewnia, że najmniejsza funkcja dominowania rzymskiego dla grafu wynikowego gry na grafie T będzie musiała przypisać przynajmniej jednemu z wierzchołków $\{v_{k-1}, v_k\}$ wartość równą 1.

Z założenia prawdą jest, że $\lceil \frac{n-2}{2} \rceil + 1 \leq \gamma_{Rg}(T \setminus \{v_{k-1}, v_k\})$. Z powyższego wniosku wiadomo, że $\gamma_{Rg}(T)$ jest przynajmniej o 1 większe od $\gamma_{Rg}(T \setminus \{v_{k-1}, v_k\})$:

$$\begin{aligned} \gamma_{Rg}(T) &\geq \gamma_{Rg}(T \setminus \{v_{k-1}, v_k\}) + 1 \\ &\geq \left\lceil \frac{n-2}{2} \right\rceil + 2 \\ &\geq \left\lceil \frac{n}{2} - 1 \right\rceil + 2 \geq \left\lceil \frac{n}{2} \right\rceil + 1. \end{aligned}$$

To wyprowadzenie dowodzi twierdzenia dla tego przypadku.

Przypadek $d(v_{k-1}) \geq 3$: Ten przypadek implikuje, że $d(v_{k-1})$ sąsiaduje z więcej niż jednym liściem. Niech u będzie jednym z liści sąsiadujących z v_{k-1} różnym od v_k .



Rys. 3.10. Najdłuższa ścieżka w drzewie T w przypadku $d(v_{k-1}) \geq 3$

Wówczas Psuj może przyjąć strategię, polegającą na graniu zgodnie ze strategią optymalną na drzewie $T \setminus \{v_k, u\}$, dopóki Dominator wykonuje ruchy na krawędziach innych niż $\{v_{k-1}v_k, v_{k-1}u\}$. Natomiast jeżeli Dominator wykona ruch na jednej z tych krawędzi, to Psuj może wykonać ruch na drugiej z nich, orientując ją w kierunku wierzchołka v_k . W ten sposób zapewnia, że najmniejsza funkcja dominowania rzymskiego na zorientowanym grafie T będzie musiała przypisać przynajmniej jednemu z wierzchołków $\{v_k, u\}$ wartość 1.

Korzystając z założenia indukcyjnego w kontekście grafu $T \setminus \{v_k, u\}$ oraz powyższego wniosku można udowodnić prawdziwość twierdzenia, używając wyprowadzenia identycznego z tym w przypadku $d(v_{k-1}) = 2$.

□

4. GRA W DOMINOWANIE RZYMSKIE Z PODZIAŁEM KRAWĘDZI

4.1. Gra w dominowanie z podziałem krawędzi

Inną grą na grafie opartą na relacji dominowania jest gra w dominowanie z podziałem krawędzi [10]. Podobnie jak standardowy wariant gry w dominowanie jest to gra rozgrywana pomiędzy dwoma graczami: Dominatorem i Psujem. Wykonują oni ruchy w naprzemiennych turach, rozpoczynając od Dominatora. Każdy z nich wykonuje ruch na jednej z nieskierowanych krawędzi (tzn. takich, na których nie zrobiono wcześniej ruchu) grafu prostego G , na którym rozgrywana jest gra.

W przeciwieństwie do podstawowego wariantu ruchy wykonywane przez graczy nie są identyczne. Dominator w ramach swojego ruchu oznacza jedną z krawędzi z $E(G)$. Na oznaczonej krawędzi nie można już do końca gry zrobić innego ruchu. Jest to więc odebranie ruchu na tej krawędzi Psujowi.

Psuj natomiast, w ramach swojego ruchu, dokonuje podziału krawędzi z $E(G)$. Oznacza to, że wybiera jedną z nieskierowanych krawędzi $\{u, v\} \in E(G)$ i usuwa ją z gry. Zamiast niej dodaje do grafu gry nowy wierzchołek x i dwie krawędzie: $\{u, x\}$ i $\{x, v\}$. W ten sposób rozdziela bezpośrednią krawędź pomiędzy u a v i sprawia, że najkrótsza ścieżka pomiędzy nimi ma długość 2.

Rozdzielenie bezpośredniego połączenia pomiędzy wierzchołkami ma istotną implikację dla relacji dominowania. Dopóki istnieje krawędź $\{u, v\}$ najmniejszy zbiór dominujący w grafie G zawierający jeden z wierzchołków u lub v będzie dominował drugi wierzchołek z pary. Po rozdzieleniu tej krawędzi na krawędzie $\{u, x\}$ i $\{x, v\}$ wierzchołek u nie dominuje już wierzchołka v i wierzchołek v nie dominuje wierzchołka u .

Tak jak w przypadku gry w dominowanie, obaj gracze mają swoje własne cele. Dominator stara się przez wykonywane ruchy zminimalizować liczbę dominowania grafu powstałego po rozegraniu gry, natomiast Psuj wykonuje ruchy mające na celu zmaksymalizowanie tej wartości.

4.1.1. Liczba gry w dominowanie z podziałem krawędzi

Tak jak w przypadku gry w dominowanie, także gra w dominowanie z podziałem krawędzi pozwala na wyznaczenie wartości charakterystycznej dla spójnego grafu prostego G [10]. Tak samo jak w grze w dominowanie będzie to liczba dominowania w wynikowym grafie G' , uzyskanym po zakończeniu gry, przy optymalnej strategii używanej przez Dominatora i Psuja. Warto zauważyć, że w tym wypadku G' jest grafem prostym, ponieważ do G dodawane są tylko nowe wierzchołki oraz dodawane i usuwane są krawędzie nieskierowane. Wartość ta jest opisywana jako $\gamma_{gs}(G) = \gamma(G')$.

4.2. Gra w dominowanie rzymskie z podziałem krawędzi

Gra w dominowanie rzymskie jest odpowiednikiem gry w dominowanie używającej liczby dominowania rzymskiego wynikowego grafu jako kryterium oceny rozgrywki [2]. Analogicznie dla gry w dominowanie z podziałem krawędzi również została zdefiniowana podobna gra, używająca dominowania rzymskiego zamiast zwykłego dominowania do oceny wyniku.

Zasady gry w dominowanie rzymskie z podziałem krawędzi są identyczne z tymi w grze używającej klasycznego dominowania. Jediną różnicą jest fakt, że celem Dominatora i Psuja jest

uzyskanie ostatecznie grafu G' , dla którego liczba dominowania rzymskiego będzie odpowiednio najmniejsza i największa. Aby go osiągnąć Dominator w swojej turze oznacza jedną z nieskierowanych krawędzi, a Psuj dzieli jedną z nich.

4.3. Liczba gry w dominowanie rzymskie z podziałem krawędzi

Dla gry w dominowanie rzymskie z podziałem krawędzi podobnie jak dla innych gier w dominowanie można wyznaczyć wartość odpowiadającego wynikowi gry przy optymalnej rozgrywce obu graczy [2]. Odpowiada ona liczbie dominowania rzymskiego grafu wynikowego G' po optymalnej rozgrywce. Opisywana jest więc jako $\gamma_{Rgs}(G) = \gamma_R(G')$.

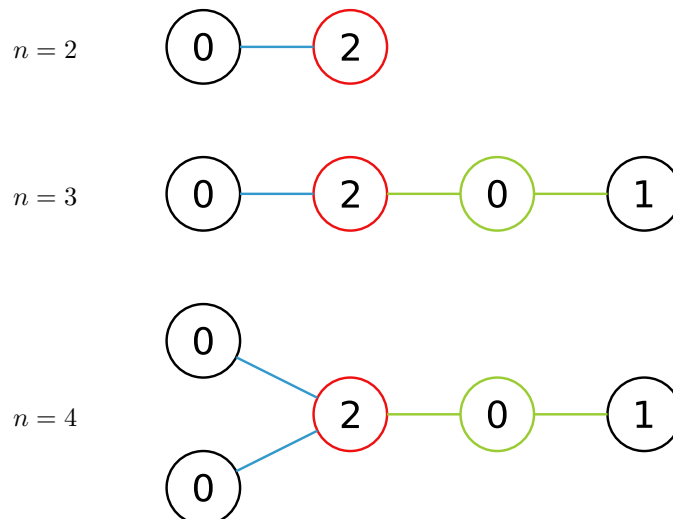
Podobnie jak w przypadku gry w dominowanie rzymskie, możliwe jest wyróżnienie poszczególnych klas grafów, dla których można wyznaczyć liczbę gry w dominowanie rzymskie z podziałem krawędzi lub przedział, w którym się znajduje. Jedną z prac zgłębiającą tę tematykę jest publikacja Amjadięgo i in. [2] Poszczególne przedstawione tam twierdzenia zostały przytoczone poniżej w poszerzonej formie.

4.3.1. Gwiazda

Twierdzenie 4.1. [2] Dla każdej gwiazdy $K_{1,n-1}$ liczba gry w dominowanie rzymskie z podziałem krawędzi wynosi $\gamma_{Rgs}(K_{1,n-1}) = \lceil \frac{n+2}{2} \rceil$.

Dowód. W przypadku gry na gwiazdzie zarówno Dominator, jak i Psuj będą działać według tej samej strategii, to jest wykonywać ruchy na kolejnych krawędziach łączących centrum gwiazdy z liśćmi. Dominator, z uwagi na to, że wykonuje pierwszy ruch, oznaczy $\lceil \frac{n-1}{2} \rceil$ spośród $n-1$ krawędzi, natomiast Psuj podzieli $\lfloor \frac{n-1}{2} \rfloor$ krawędzi.

Niezależnie od n będzie istniała najmniejsza funkcja dominowania rzymskiego, przypisująca wierzchołkowi centralnemu wartość 2. Przykładowe najmniejsze funkcje dla małych wartości n zostały przedstawione na rys. 4.1.



Rys. 4.1. Przykłady gry w dominowanie rzymskie z podziałem krawędzi na gwiazdach $K_{1,n-1}$ dla $n \in \{2, 3, 4\}$. Wierzchołki centralne zostały oznaczone kolorem czerwonym, wierzchołki dodane w wyniku podziału krawędzi przez Psuję, jak i same podzielone krawędzie kolorem zielonym, a kolorem niebieskim krawędzie oznaczone przez Dominatora

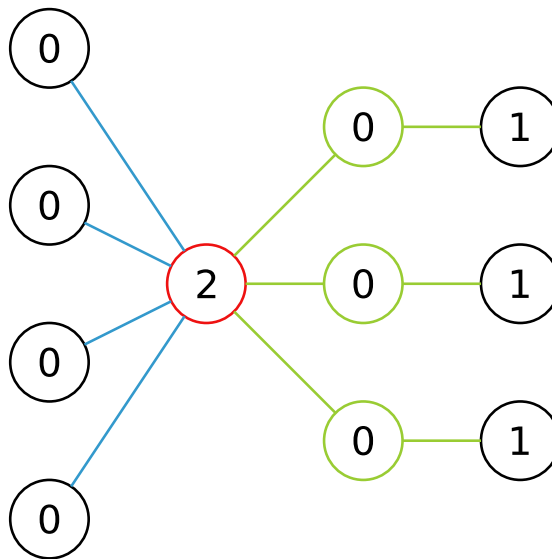
W związku z tym liście połączone z centrum krawędziami oznaczonymi przez Dominatora mogą mieć przypisaną wartość równą 0. Rozpatrzmy podgraf złożony z wierzchołka centralnego,

liścia gwiazdy oraz wierzchołek dodanego w wyniku podziału krawędzi przez Psuja. Najmniejsza funkcja dominowania rzymskiego w tym podgrafie, przy założeniu wartości przypisanej wierzchołkowi centralnemu równej 2, będzie przypisywać dodanemu wierzchołkowi wartość równą 0, a liściowi wartość równą 1.

Stąd w grafie będzie jeden (centralny) wierzchołek o przypisanej wartości równej 2 oraz $\lfloor \frac{n-1}{2} \rfloor$ wierzchołków o przypisanej wartości równej 1. Pozostałym wierzchołkom przypisana będzie wartość 0, więc

$$\gamma_{Rgs}(K_{1,n-1}) = 2 + \left\lfloor \frac{n-1}{2} \right\rfloor = \left\lceil \frac{n+2}{2} \right\rceil.$$

□



Rys. 4.2. Przykłady gry w dominowanie rzymskie z podziałem krawędzi na gwiazdzie $K_{1,7}$. Wierzchołek centralny został oznaczony kolorem czerwonym, a wierzchołki dodane w wyniku podziału krawędzi przez Psuja kolorem zielonym

4.3.2. Ścieżka i cykl

Lemat 4.2. [6] Liczba dominowania rzymskiego dla ścieżek P_n i cykli C_n jest równa $\lceil \frac{2n}{3} \rceil$.

Dowód. Zarówno ścieżkę, jak i cykl, w którym $n \bmod 3 = 0$, można rozpatrywać jako ciąg $\frac{n}{3}$ połączonych podgrafów P_3 , gdzie $V(P_3) = \{v_1, v_2, v_3\}$. W takim podgrafie najmniejsza funkcja dominowania rzymskiego przypisze wierzchołkowi v_2 wartość równą 2, a wierzchołkom v_1, v_3 wartość 0. Wówczas sumaryczna wartość przypisana wierzchołkom w każdym podgrafie będzie równa 2. Stąd liczba dominowania rzymskiego będzie wynosić $\gamma_R(P_n) = \gamma_R(C_n) = 2 \cdot \frac{n}{3}$.

Rozważmy przypadki ścieżki i cyklu, gdzie $n \bmod 3 \neq 0$:

$n \bmod 3 = 1$: Graf może zostać podzielony na $\frac{n-1}{3}$ ścieżek P_3 , o sumarycznej wartości najmniejszej funkcji dominowania rzymskiego równej 2, jak wyżej. Pozostały wierzchołek nie jest dominowany przez żaden z wierzchołków o przypisanej wartości 2, więc musi mieć przypisaną wartość równą 1. Sumarycznie daje to liczbę dominowania rzymskiego równą

$$\gamma_R(P_n) = \gamma_R(C_n) = 2 \cdot \frac{n-1}{3} + 1 = \frac{2n}{3} + \frac{1}{3}.$$

$n \bmod 3 = 2$: Podobnie do powyższego przypadku graf można podzielić na $\frac{n-2}{3}$ ścieżek P_3 o sumarycznej przypisanej wartości 2. Pozostałe dwa wierzchołki będą miały przypisane wartości 1, 1 lub 2, 0, co sumarycznie daje wartość 2. W związku z tym liczba dominowania rzymskiego będzie równa

$$\gamma_R(P_n) = \gamma_R(C_n) = 2 \cdot \frac{n-2}{3} + 2 = \frac{2n}{3} + \frac{2}{3}.$$

Zbierając powyższe wyniki mamy

$$\left. \begin{array}{l} n \bmod 3 = 0 \rightarrow \gamma_R(P_n) = \gamma_R(C_n) = \frac{2n}{3} \\ n \bmod 3 = 1 \rightarrow \gamma_R(P_n) = \gamma_R(C_n) = \frac{2n}{3} + \frac{1}{3} \\ n \bmod 3 = 2 \rightarrow \gamma_R(P_n) = \gamma_R(C_n) = \frac{2n}{3} + \frac{2}{3} \end{array} \right\} \Rightarrow \gamma_R(P_n) = \gamma_R(C_n) = \left\lceil \frac{2n}{3} \right\rceil.$$

□

Twierdzenie 4.3. [2] Dla ścieżki P_n i cyklu C_n zachodzi $\gamma_{Rgs}(P_n) = \gamma_{Rgs}(C_n) = n$.

Dowód. Dowód zaczniemy od obserwacji, że podział krawędzi w grafie P_n skutkuje powstaniem ścieżki o jeden wierzchołek dłuższej. Dlatego graf wynikowy gry w dominowanie rzymskie z podziałem krawędzi na ścieżce będzie również ścieżką. Podobnie graf wynikowy gry na cyklu będzie również cyklem.

Ścieżka P_n składa się z $n-1$ krawędzi, natomiast cykl C_n składa się z n krawędzi. Jako że gracze wykonują ruchy naprzemiennie, a Dominator wykonuje pierwszy ruch, to Psuj podzieli $\lfloor \frac{n-1}{2} \rfloor$ krawędzi w przypadku ścieżki P_n lub $\lfloor \frac{n}{2} \rfloor$ krawędzi w przypadku cyklu C_n . Stąd grafem wynikowym gry na P_n będzie $P'_n = P_{n+\lfloor \frac{n-1}{2} \rfloor}$, a gry na C_n będzie $C'_n = C_{n+\lfloor \frac{n}{2} \rfloor}$.

Z definicji $\gamma_{Rgs}(G) = \gamma_R(G')$, więc

$$\begin{aligned} \gamma_{Rgs}(P_n) &= \gamma_R(P'_n) = \gamma_R(P_{n+\lfloor \frac{n-1}{2} \rfloor}) \\ \gamma_{Rgs}(C_n) &= \gamma_R(C'_n) = \gamma_R(C_{n+\lfloor \frac{n}{2} \rfloor}) \end{aligned}$$

Z lematu 4.2 wynika, że:

$$\begin{aligned} \gamma_{Rgs}(P_n) = \gamma_R(P_{n+\lfloor \frac{n-1}{2} \rfloor}) &= \left\lceil \frac{2(n + \lfloor \frac{n-1}{2} \rfloor)}{3} \right\rceil = \left\{ \begin{array}{l} \text{jeżeli } 2 \mid n : \left\lceil \frac{3n-2}{3} \right\rceil = \left\lceil n - \frac{2}{3} \right\rceil \\ \text{jeżeli } 2 \nmid n : \left\lceil \frac{3n-1}{3} \right\rceil = \left\lceil n - \frac{1}{3} \right\rceil \end{array} \right\} = n \\ \gamma_{Rgs}(C_n) = \gamma_R(C_{n+\lfloor \frac{n}{2} \rfloor}) &= \left\lceil \frac{2(n + \lfloor \frac{n}{2} \rfloor)}{3} \right\rceil = \left\{ \begin{array}{l} \text{jeżeli } 2 \mid n : \left\lceil \frac{3n}{3} \right\rceil = \lceil n \rceil \\ \text{jeżeli } 2 \nmid n : \left\lceil \frac{3n-1}{3} \right\rceil = \left\lceil n - \frac{1}{3} \right\rceil \end{array} \right\} = n \end{aligned}$$

□

4.3.3. Graf r -regularny

Lemat 4.4. [7] Dla grafu prostego G zachodzi $\gamma_R(G) \geq \frac{2n}{\Delta+1}$, gdzie Δ oznacza największy stopień wierzchołka w grafie G .

Dowód. Funkcja dominowania rzymskiego f dzieli zbiór wierzchołków na trzy podzbiory (V_0, V_1, V_2) . Zawartym w nich wierzchołkom przypisane są odpowiednio wartości $(0, 1, 2)$. Stąd mamy więc definicję liczby dominowania rzymskiego dla grafu G :

$$\gamma_R(G) = |V_1| + 2|V_2|.$$

Można zauważyć, że każdy z wierzchołków, któremu została przypisana wartość 0, z definicji dominowania rzymskiego musi sąsiadować z wierzchołkiem, któremu została przypisana wartość 2. W związku z tym liczba wierzchołków w zbiorze $|V_0|$ nie może być większa niż sumaryczna liczba wierzchołków sąsiadujących z wierzchołkami o przypisanej wartości 2. Jednocześnie największa liczba wierzchołków sąsiadujących z wierzchołkiem wynosi Δ , więc prawdą jest, że $|V_0| \leq \Delta|V_2|$.

Każdy z n wierzchołków grafu G należy do tylko jednego ze zbiorów (V_0, V_1, V_2) , więc

$$\begin{aligned} n = |V_0| + |V_1| + |V_2| &\leq \Delta|V_2| + |V_2| + |V_1| = (\Delta + 1)|V_2| + |V_1| \\ \frac{n}{\Delta + 1} &\leq |V_2| + \frac{1}{\Delta + 1}|V_1|. \end{aligned}$$

Zakładając, że w grafie G istnieje przynajmniej jedna krawędź, to $\Delta \geq 1$. Stąd:

$$\frac{2n}{\Delta + 1} \leq 2|V_2| + \frac{2}{\Delta + 1}|V_1| \leq 2|V_2| + |V_1| = \gamma_R(G).$$

□

Twierdzenie 4.5. [2] Dla r -regularnego grafu G liczba gry w dominowanie rzymskie z podziałem krawędzi jest ograniczona od dołu:

$$\gamma_{Rgs}(G) \geq \frac{2(n + \lfloor \frac{rn}{4} \rfloor)}{r + 1}$$

Dowód. Twierdzenie to jest bezpośrednią konsekwencją użycia lematu 4.4 w odniesieniu do grafu wynikowego G' gry w dominowanie rzymskie z podziałem krawędzi na grafie G i wykorzystania zależności $\gamma_{Rgs}(G) = \gamma_R(G')$.

Graf G' różni się od G tym, że niektóre z jego krawędzi zostały podzielone. Można zaobserwować, że podział krawędzi nie zmienia stopnia żadnego z istniejących wierzchołków grafu, a stopień nowego wierzchołka jest równy 2. Oznacza to, że jeżeli $\Delta(G) = r \geq 2$, to $\Delta(G') = \Delta(G) = r$.¹

Dominator i Psuj wykonują ruchy naprzemiennie, a Dominator wykonuje pierwszy ruch, więc w trakcie gry podzielonych zostanie $\lfloor \frac{m(G)}{2} \rfloor$ krawędzi, gdzie $m(G)$ jest liczbą krawędzi w grafie G . Z lematu o uściskach dłoni [17, 1.3.3.] wiadomo, że suma stopni wszystkich wierzchołków grafu jest równa $2m$, więc w przypadku grafu r -regularnego mamy $m = \frac{rn}{2}$, czyli $\lfloor \frac{rn}{4} \rfloor$ krawędzi grafu G zostanie podzielonych. Przy podziale każdej z krawędzi do grafu dodawany jest jeden nowy

¹Graf regularny, gdzie $r = 1$, jest izomorficzny do ścieżki P_2 i spełnia zależność przedstawioną w treści twierdzenia.

wierzchołek, więc liczba wierzchołków w grafie G' będzie wynosić

$$n(G') = n(G) + \left\lfloor \frac{rn}{4} \right\rfloor.$$

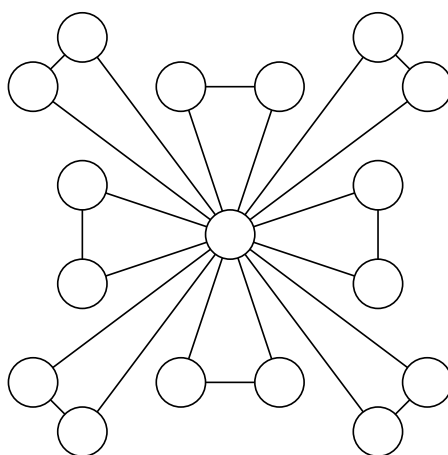
Stąd, używając wzoru z lematu 4.4, mamy

$$\gamma_{Rgs}(G) = \gamma_R(G') \geq \frac{2n(G')}{\Delta(G') + 1} = \frac{2(n(G) + \lfloor \frac{rn}{4} \rfloor)}{r + 1}.$$

□

4.3.4. k -wiatrak

k -wiatrak F_k jest grafem powstałym z k kopii grafu $K_3 = C_3$, gdzie jeden z wierzchołków jest wspólny dla wszystkich kopii.



Rys. 4.3. Graf F_8 (8-wiatrak) składający się z 8 grafów K_3 połączonych wspólnym wierzchołkiem centralnym

Twierdzenie 4.6. [2] Dla każdego k -wiatraka zachodzi $\gamma_{Rgs}(F_k) = 2 + \lfloor \frac{k}{2} \rfloor + 2 \lfloor \frac{k}{2} \rfloor$.

Dowód. Każdy z k cykli C_3 , składających się na graf F_k , może w trakcie gry w dominowanie zostać podzielony od zera do trzech razy. Można zauważyć, że mimo podziałów krawędzi każdy z cykli pozostaje dalej cyklem, a jego długość jest równa sumie liczby początkowych wierzchołków (3) oraz liczby podzielonych krawędzi.

W trakcie gry nie zmieniają się również stopnie wierzchołków. Wierzchołek centralny będzie miał zawsze $2m$ wierzchołków sąsiadujących, natomiast pozostałe wierzchołki będą miały stopień 2. Można więc wywnioskować, że niezależnie od wartości m zawsze będzie istnieć taka najmniejsza funkcja dominowania rzymskiego, która przypisze wierzchołkowi centralnemu wartość 2.

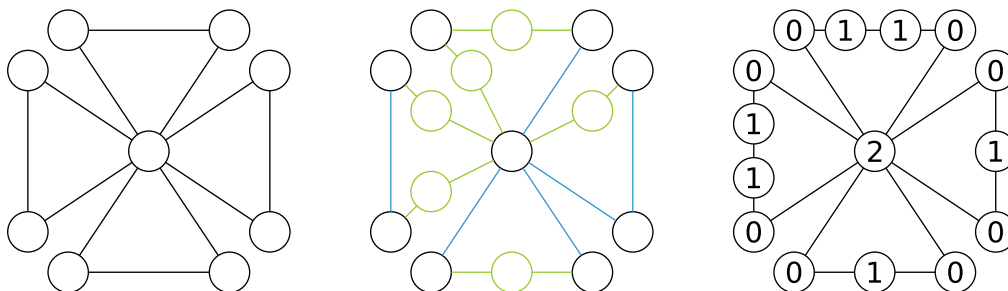
Na tej podstawie możliwe jest sformułowanie zależności pomiędzy liczbą podzielonych krawędzi w danym cyklu grafu początkowego, a sumaryczną wartością przypisaną wierzchołkom w odpowiadającym mu cyklu w grafie wynikowym. Została ona przedstawiona w tabeli 4.1.

Niech Psuj przyjmie następującą strategię:

1. Podziel krawędź w cyklu, w którym jest jedna podzielona i jedna oznaczona krawędź.
2. Jeżeli to niemożliwe, podziel krawędź w cyklu, w którym dwie krawędzie są oznaczone.
3. Jeżeli to niemożliwe, podziel krawędź w cyklu, w którym żadna z krawędzi nie jest podzielona ani oznaczona.
4. Jeżeli to niemożliwe, podziel dowolną krawędź.

Liczba podzielonych krawędzi w cyklu grafu F_k	Sumaryczna wartość najmniejszej funkcji dominowania rzymskiego wierzchołków w tej części
0	2 (wierzchołek centralny)
1	2 (wierzchołek centralny) + 1
2	2 (wierzchołek centralny) + 2
3	2 (wierzchołek centralny) + 2

Tabela 4.1. Wartości najmniejszej funkcji dominowania rzymskiego w pojedynczym cyklu grafu F_k



Rys. 4.4. Ilustracja gry w dominowanie rzymskie z podziałem krawędzi na grafie F_4 . Od lewej strony: graf początkowy F_4 , przykładowe ruchy Dominatora i Psuja, najmniejsza funkcja dominowania rzymskiego na wynikowym grafie

Rozważmy działanie tej strategii w praktyce. Jeżeli Dominator oznaczy pierwszą krawędź danego cyklu, to Psuj, jeśli ma jeszcze taką możliwość, podzieli krawędź innego cyklu, w którym wszystkie krawędzie są jeszcze wolne² (punkt 3. strategii). Dopiero gdy Dominator oznaczy drugą z krawędzi tego cyklu, Psuj podzieli jego ostatnią krawędź (punkt 2. strategii). Natomiast w przypadku cyklu, w którym Psuj wykona pierwszy ruch (na podstawie punktu 3. strategii), kolejny ruch będzie należał do Dominatora, po czym ostatnia krawędź zostanie podzielona przez Psuja (na podstawie punktu 1. strategii).

Można zauważyć, że w ten sposób w każdym z cykli będzie jedna lub dwie podzielone krawędzie. Co więcej, liczba ta będzie zależeć od tego, który z graczy wykonał pierwszy ruch w danym cyklu: gdy będzie to Dominator, to zostanie podzielona jedna krawędź, gdy Psuj – dwie.

Istotnym jest również fakt, że gdy Dominator wykonuje pierwszy ruch na jednym z cykli, Psuj wykonuje pierwszy ruch na innym, o ile jest to jeszcze możliwe. Z założenia Dominator rozpoczyna rozgrywkę, więc będzie w stanie wykonać pierwszy ruch na $\lceil \frac{k}{2} \rceil$ cyklach. W każdym z nich będzie podzielona jedna krawędź. W pozostałych $\lfloor \frac{k}{2} \rfloor$ cyklach zostaną podzielone dwie krawędzie.

Zgodnie z informacjami z tabeli 4.1 cykle z jedną podzieloną krawędzią będą miały sumaryczną wartość najmniejszej funkcji dominowania rzymskiego równą 1, a cykle z dwiema podzielonymi krawędziami wartość równą 2 (nie wliczając wierzchołka centralnego). Stąd sumaryczna wartość najmniejszej funkcji dominowania rzymskiego f_{min} wynosi

$$f_{min}(V(F_k)) = 2 + \left\lceil \frac{k}{2} \right\rceil + 2 \left\lfloor \frac{k}{2} \right\rfloor.$$

To dowodzi, że $\gamma_{Rgs}(F_k) \geq 2 + \lceil \frac{k}{2} \rceil + 2 \lfloor \frac{k}{2} \rfloor$. Zauważmy, że Dominator może użyć analogicznej do przedstawionej wyżej strategii Psuja i uzyskać taką samą sumaryczną wartość najmniej-

²tj. nie wykonano na nich ruchu

szej funkcji dominowania rzymskiego, niezależnie od ruchów Psuja. Łącząc ten fakt i powyższą nierówność mamy

$$\gamma_{Rgs}(F_k) = 2 + \left\lceil \frac{k}{2} \right\rceil + 2 \left\lfloor \frac{k}{2} \right\rfloor.$$

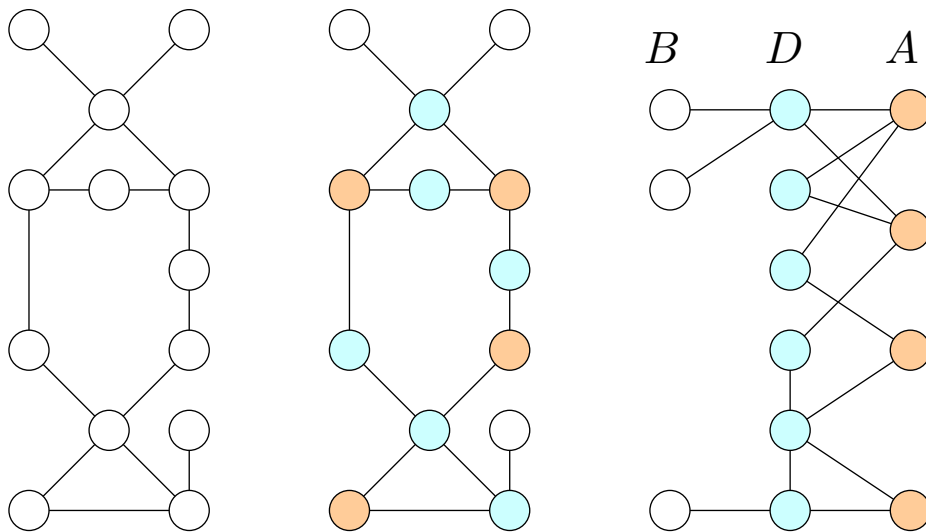
□

4.3.5. Graf dwudzielny

Lemat 4.7. [2] Jeżeli D jest takim zbiorem dominującym w grafie G , że $V(G) \setminus D$ jest zbiorem niezależnym, $A \subset V(G) \setminus D$ jest zbiorem wierzchołków, które mają przynajmniej dwa wierzchołki sąsiadujące należące do D , a $B = V(G) \setminus (D \cup A)$, to

$$\gamma_{Rgs}(G) \leq 2|D| + \left\lfloor \frac{|B|}{2} \right\rfloor.$$

Dowód. Zbiór $V(G) \setminus D = A \cup B$ jest niezależny, co oznacza, że wierzchołki należące do A lub B będą sąsiadować jedynie z wierzchołkami z D . Wierzchołki D natomiast mogą sąsiadować z wierzchołkami ze wszystkich trzech zbiorów (A, B, D). Jeżeli każdemu z wierzchołków zbioru D przypisano by w grafie wynikowym G' wartość funkcji dominowania rzymskiego równą 2, to wierzchołki należące do $A \cup B$, posiadające przynajmniej jedną niepodzieloną krawędź łączącą je z wierzchołkiem ze zbioru D , mogłyby przyjąć wartość równą 0.



Rys. 4.5. Podział na zbiory D (niebieski), B i A (pomarańczowy) przykładowego grafu

Dominator może rozpocząć grę oznaczeniem jednej z krawędzi łączących wierzchołek z B z wierzchołkiem z D . Następnie może przyjąć następującą strategię uzależnioną od ruchów Psuja:

1. Jeżeli Psuj podzieli krawędź łączącą wierzchołek z B z wierzchołkiem z D , to Dominator oznacza inną krawędź łączącą te dwa zbiory.
2. Jeżeli Psuj podzieli krawędź łączącą wierzchołek $v \in A$ z wierzchołkiem z D , to Dominator oznacza krawędź łączącą v z innym wierzchołkiem z D . Z definicji zbioru A pomiędzy tymi zbiorami muszą istnieć przynajmniej dwie krawędzie.

3. Jeżeli Psuj podzieli inną krawędź lub powyższy ruch nie jest możliwy, to Dominator oznacza dowolną krawędź.

Punkt 2. powyższej strategii gwarantuje, że każdy wierzchołek w A będzie połączony bezpośrednio z przynajmniej jednym wierzchołkiem w D w grafie wynikowym G' . Dodatkowo pierwszy ruch Dominatora oraz punkt 1. strategii sprawia, że przynajmniej $\left\lceil \frac{|B|}{2} \right\rceil$ wierzchołków ze zbioru B będzie bezpośrednio połączonych z wierzchołkiem ze zbioru D . Przyjmując opisaną wcześniej funkcję dominowania rzymskiego, wierzchołki te będą miały przypisaną wartość 0.

Zauważmy, że wszystkie wierzchołki dodane w wyniku dzielenia krawędzi mogą również mieć wartość funkcji dominowania rzymskiego równą 2 z uwagi na fakt sąsiedowania z wierzchołkami ze zbioru D , które mają przypisaną wartość 2. Pozostałe $\left\lfloor \frac{|B|}{2} \right\rfloor$ wierzchołków zbioru B będzie połączone ze zbiorem D poprzez podzieloną krawędź. Wierzchołki te nie sąsiadują z wierzchołkami o przypisanej wartości 2, przez co same muszą mieć przypisaną wartość równą 1.

Podsumowując, jeżeli zostanie zastosowana powyższa strategia i opisana funkcja dominowania rzymskiego, to w grafie G' będzie $|D|$ wierzchołków o przypisanej wartości 2 oraz $\left\lfloor \frac{|B|}{2} \right\rfloor$ wierzchołków o przypisanej wartości 1. Pozostałe wierzchołki będą miały wartość funkcji dominowania rzymskiego równą 0. Stąd

$$\gamma_{Rgs}(G) \leq 2|D| + \left\lfloor \frac{|B|}{2} \right\rfloor.$$

□

Pokrycie wierzchołkowe grafu G to taki zbiór wierzchołków, że każda krawędź w $E(G)$ zawiera przynajmniej jeden wierzchołek z tego zbioru. Moc najmniejszego pokrycia wierzchołkowego w grafie G jest nazywana jego liczbą pokrycia wierzchołkowego i jest oznaczana symbolem $\tau(G)$.

Zbiór niezależny w grafie G to taki podzbiór wierzchołków, że żadne dwa wierzchołki z tego zbioru nie sąsiadują ze sobą. Moc najliczniejszego zbioru niezależnego w grafie G jest oznaczana przez $\alpha(G)$.

Lemat 4.8. [11] *Twierdzenie Gallai'ego*

Dla każdego grafu G

$$\alpha(G) + \tau(G) = n(G).$$

Dowód. Niech $U \subset V(G)$ będzie pokryciem wierzchołkowym grafu G . Z definicji pokrycia wierzchołkowego przynajmniej jeden z wierzchołków każdej krawędzi w $E(G)$ należy do U . Oznacza to, że nie istnieje taka krawędź, która łączyłaby dwa wierzchołki nienależące do U , czyli należące do $V(G) \setminus U$. Stąd $V(G) \setminus U$ jest zbiorem niezależnym. Podobnie jeżeli $I \subset V(G)$ jest zbiorem niezależnym grafu G , to $V(G) \setminus I$ jest pokryciem wierzchołkowym.

Niech U_0 będzie najmniej licznym pokryciem wierzchołkowym grafu G ($\tau(G) = |U_0|$). Z powyższego $V(G) \setminus U$ jest zbiorem niezależnym.

Niech $I \subset V$ będzie zbiorem niezależnym liczniejszym od $V(G) \setminus U_0$. Wówczas $V(G) \setminus I$ jest pokryciem wierzchołkowym oraz $|V(G) \setminus I| < |U_0|$, co jest sprzeczne z założeniem, że U_0 jest najmniej licznym pokryciem wierzchołkowym. Stąd nie może istnieć zbiór niezależny liczniejszy od $V(G) \setminus U_0$, czyli

$$\alpha(G) = |V(G) \setminus U_0| = n(G) - |U_0| = n(G) - \tau(G)$$

$$\alpha(G) + \tau(G) = n(G).$$

□

Lemat 4.9. [2] Jeżeli w grafie G zachodzi $\delta(G) \geq 2$, to $\gamma_{Rgs}(G) \leq 2(n - \alpha(G))$.

Dowód. Niech $U \subset V(G)$ będzie najmniejszym pokryciem wierzchołkowym grafu G . Z definicji przynajmniej jeden wierzchołek każdej krawędzi w grafie G należy do U . Każdy z wierzchołków w G należy do przynajmniej dwóch krawędzi. Oznacza to, że każdy z wierzchołków zbioru niezależnego $V(G) \setminus U$ sąsiaduje z przynajmniej dwoma wierzchołkami należącymi do U , czyli U jest zbiorem 2-dominującym w grafie G .

Podzielmy wierzchołki $V(G)$ na zbiory (B, D, A) zgodnie z lematem 4.7:

- $D = U$, ponieważ U jest zbiorem dominującym i zarazem pokryciem wierzchołkowym w G , więc $V(G) \setminus U$ jest zbiorem niezależnym.
- $A = V(G) \setminus U$, ponieważ U jest zbiorem 2-dominującym w G , więc każdy z wierzchołków w $V(G) \setminus U$ sąsiaduje z przynajmniej dwoma wierzchołkami należącymi do U .
- $B = \emptyset$, ponieważ $B = V(G) \setminus (A \cup D) = V(G) \setminus V(G)$.

Zgodnie z wzorem z lematu 4.7

$$\gamma_{Rgs}(G) \leq 2|D| + \left\lfloor \frac{|B|}{2} \right\rfloor = 2|U| + \left\lfloor \frac{|\emptyset|}{2} \right\rfloor = 2|U|.$$

Z lematu 4.8 mamy:

$$\tau(G) = |U|$$

$$\alpha(G) + \tau(G) = n(G)$$

$$|U| = n(G) - \alpha(G),$$

zatem

$$\gamma_{Rgs}(G) \leq 2(n(G) - \alpha(G)).$$

□

Skojarzenie jest zbiorem krawędzi, które ze sobą nie sąsiadują, tj. nie mają wspólnych wierzchołków.

Skojarzenie M w grafie G , w którym krawędzie łączą wszystkie wierzchołki w $V(G)$, jest nazywane skojarzeniem doskonałym. Ponieważ krawędzie w M są parami rozłączne (nie mają wspólnych wierzchołków), spełniona jest własność $2|M| = n(G)$.

Lemat 4.10. [12] *Twierdzenie Königa*

Dla grafu dwudzielnego G zachodzi $\tau(G) = |M|$, gdzie M jest największym skojarzeniem w grafie G .

Lemat 4.11. [16] Dla grafu dwudzielnego G

$$\alpha(G) = \frac{n(G)}{2}$$

wtedy, i tylko wtedy, gdy G ma skojarzenie doskonałe i $n(G)$ jest parzyste.

Dowód. Z lematu 4.8 mamy własność $\alpha(G) + \tau(G) = n(G)$. Podstawiając do powyższego równania zależność z lematu 4.10, otrzymujemy $\alpha(G) + |M| = n(G)$.

Przyjmijmy, że $\alpha(G) = \frac{n(G)}{2}$. Wówczas $|M| = \frac{n(G)}{2}$. Z powyższego równania otrzymujemy $n(G) = 2|M|$, co oznacza, że w grafie G występuje skojarzenie doskonałe. Dodatkowo $\alpha(G)$ oznacza moc zbioru, która jest wartością całkowitą, więc $n(G)$ musi być podzielne przez 2.

Przyjmijmy teraz, że w grafie G występuje skojarzenie doskonałe i $n(G)$ jest parzyste. Wówczas $2|M| = n(G)$. Stąd $\alpha(G) + \frac{n(G)}{2} = n(G) \implies \alpha(G) = \frac{n(G)}{2}$. \square

Twierdzenie 4.12. [2] Dla grafu dwudzielnego G , w którym $\delta(G) \geq 2$, zachodzi $\gamma_{Rgs}(G) \leq n(G)$.

W szczególności, jeżeli G nie ma skojarzenie doskonałego lub G ma nieparzystą liczbę wierzchołków, to $\gamma_{Rgs}(G) < n(G)$.

Dowód. W grafie dwudzielnym G zachodzi $\alpha(G) \geq \frac{n(G)}{2}$, ponieważ oba z tworzących go rozdzielnych zbiorów wierzchołków są zbiorami niezależnymi, więc największy zbiór niezależny w grafie ma przynajmniej taką moc jak większy z tych zbiorów.

Z lematu 4.9 mamy

$$\gamma_{Rgs}(G) \leq 2(n(G) - \alpha(G)) \leq 2\left(n(G) - \frac{n(G)}{2}\right) = n(G).$$

Zgodnie z lematem 4.11 w przypadku, gdy w grafie G nie ma skojarzenia doskonałego lub $n(G)$ jest nieparzyste, to

$$\gamma_{Rgs}(G) \leq 2(n(G) - \alpha(G)) < 2\left(n(G) - \frac{n(G)}{2}\right) = n(G).$$

\square

Twierdzenie 4.13. [2] W pełnym grafie dwudzielnym $K_{p,q}$, gdzie $2 \leq p \leq q$, zachodzi

$$p + 2 \leq \left\lceil \frac{2(p + q + \lfloor \frac{pq}{2} \rfloor)}{q + 1} \right\rceil \leq \gamma_{Rgs}(K_{p,q}) \leq 2p.$$

Dowód. Z lematu 4.9 mamy $\gamma_{Rgs}(K_{p,q}) \leq 2(n(K_{p,q}) - \alpha(K_{p,q}))$. Największym zbiorem niezależnym w pełnym grafie dwudzielnym jest większy z tworzących go rozdzielnych zbiorów wierzchołków, który, zgodnie z przyjętymi założeniami, ma moc równą q , więc $\alpha(K_{p,q}) = q$. Stąd

$$\gamma_{Rgs}(K_{p,q}) \leq 2(n(K_{p,q}) - q) = 2(p + q - q) = 2p.$$

Graf $K'_{p,q}$ powstały w wyniku gry w dominowanie rzymskie z podziałem krawędzi na grafie $K_{p,q}$ różni się od oryginalnego grafu tym, że zostały w nim podzielone niektóre krawędzie. Podział krawędzi nie zmienia stopni żadnego z oryginalnych wierzchołków oraz dodaje jeden wierzchołek o stopniu 2. Zgodnie z założeniem $2 \leq p \leq q$ w $K_{p,q}$ wszystkie wierzchołki mają stopień większy

od lub równy 2. Wobec tego przeprowadzenie gry na grafie nie wpłynie na największy stopień wierzchołka w grafie: $\Delta(K'_{p,q}) = \Delta(K_{p,q}) = q$.

Liczba krawędzi w grafie początkowym wynosi $p \cdot q$. W wyniku gry w dominowanie rzymskie z podziałem krawędzi Psuj podzieli $\lfloor \frac{pq}{2} \rfloor$ krawędzi, więc sumaryczna liczba wierzchołków w grafie $K'_{p,q}$ będzie wynosiła $n(K'_{p,q}) = n(K_{p,q}) + \lfloor \frac{pq}{2} \rfloor = p + q + \lfloor \frac{pq}{2} \rfloor$.

Podstawiając powyższe wartości do wzoru danego lematem 4.4, mamy

$$\gamma_{Rgs}(K_{p,q}) = \gamma_R(K'_{p,q}) \geq \frac{2(p + q + \lfloor \frac{pq}{2} \rfloor)}{q + 1}.$$

Z definicji $\gamma_{Rgs}(K_{p,q})$ jest liczbą całkowitą, więc prawdą jest również, że

$$\gamma_{Rgs}(K_{p,q}) \geq \left\lceil \frac{2(p + q + \lfloor \frac{pq}{2} \rfloor)}{q + 1} \right\rceil.$$

Wtedy

$$\left\lceil \frac{2(p + q + \lfloor \frac{pq}{2} \rfloor)}{q + 1} \right\rceil \geq \left\lceil \frac{2p + 2q + pq - 1}{q + 1} \right\rceil \geq \frac{2p + 2q + pq - 1}{q + 1}.$$

Przeanalizujmy zmienność wartości uzyskanego wyrażenia względem q :

$$\frac{\partial}{\partial q} \left(\frac{2p + 2q + pq - 1}{q + 1} \right) = \frac{(2 + p)(q + 1) - (2p + 2q + pq - 1)}{(q + 1)^2} = \frac{p + 3}{(q + 1)^2}.$$

Z powyższego wynika, że wartość tego wyrażenia jest niemalejąca niezależnie od wartości q . Zatem prawdą będzie, że

$$\frac{2p + 2q + pq - 1}{q + 1} \geq \frac{4p + p^2 - 1}{p + 1}.$$

Wychodząc z początkowego założenia mamy

$$\begin{aligned} p \geq 2 &\implies 5p \geq 10 \\ &\implies p^2 + 4p - 1 \geq p^2 - p + 9 \\ &\implies p^2 + 4p - 1 \geq (p - 2)(p + 1) + 7 \geq (p - 2)(p + 1) \\ &\implies \frac{4p + p^2 - 1}{p + 1} \geq p - 2. \end{aligned}$$

Podsumowując, otrzymujemy

$$\left\lceil \frac{2(p + q + \lfloor \frac{pq}{2} \rfloor)}{q + 1} \right\rceil \geq \left\lceil \frac{2p + 2q + pq - 1}{q + 1} \right\rceil \geq \frac{2p + 2q + pq - 1}{q + 1} \geq \frac{4p + p^2 - 1}{p + 1} \geq p - 2,$$

co kończy dowód. □

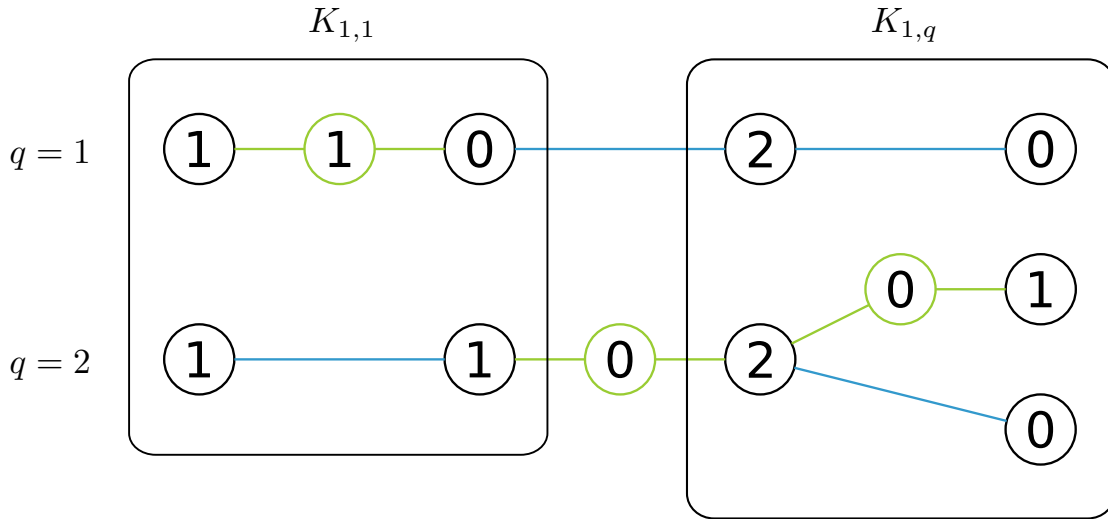
4.3.6. Podwójna gwiazda

Twierdzenie 4.14. [2] W podwójnej gwiazdzie $DS_{1,q}$ liczba gry w dominowanie rzymskie z podziałem krawędzi wynosi $2 + \lfloor \frac{n(DS_{1,q})}{2} \rfloor$.

Dowód. Liczba wierzchołków grafu $K_{1,q}$ wynosi $n(DS_{1,q}) = q + 3$, natomiast liczba krawędzi wynosi $m(DS_{1,q}) = q + 2$.

Jeżeli $q = 1$, to graf $DS_{1,q}$ jest izomorficzny do ścieżki P_4 . Jego liczba gry w dominowanie rzymskie z podziałem krawędzi jest, zgodnie z twierdzeniem 4.3, równa $n(P_4) = 4$, co jest zgodne z twierdzeniem.

Jeżeli $q = 2$, to Psuj ma zawsze możliwość podzielenia przynajmniej jednej z krawędzi podgrafu $K_{1,q}$. Wówczas najmniejsza wartość funkcji dominowania rzymskiego dla grafu wynikowego wynosi 5, co również jest zgodne z twierdzeniem.



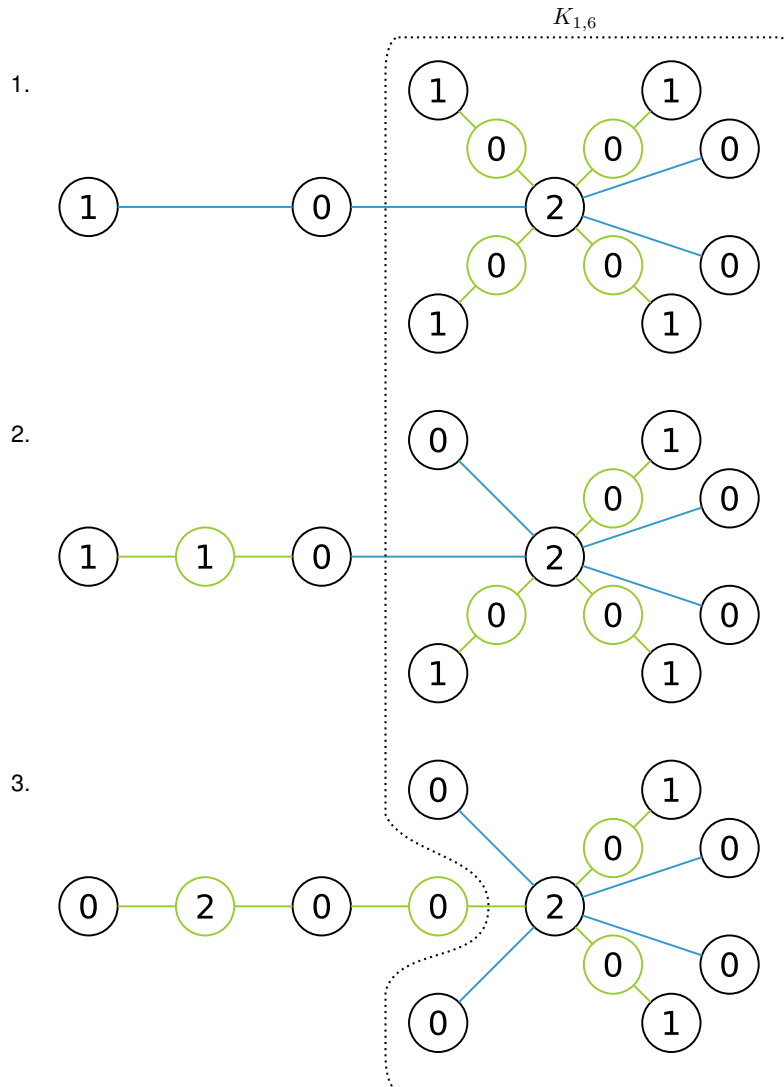
Rys. 4.6. Przykładowe najmniejsze funkcje dominowania rzymskiego w grafach wynikowych gry w dominowanie rzymskie z podziałem krawędzi na grafie $DS_{1,q}$ dla $q \in \{2, 3\}$

Niech u będzie centrum podgrafu $K_{1,1}$, v centrum podgrafu $K_{1,q}$, a w liściem podgrafu $K_{1,1}$. Dla $q \geq 3$ najmniejsza funkcja dominowania rzymskiego zawsze przypisze wierzchołkowi v w grafie wynikowym $DS'_{1,q}$ wartość 2. Wówczas każda podzielona krawędź w podgrafie $K_{1,q}$ zwiększa liczbę dominowania rzymskiego o 1, ponieważ usuwana jest krawędź pomiędzy v i liściem, więc v już go nie dominuje.

Rozważmy podgraf indukowany przez wierzchołki $\{w, u, v\}$. Jest to ścieżka P_3 i wiadomo, że v ma przypisaną wartość równą 2. Rozważmy trzy przypadki:

1. Jeżeli żadna z krawędzi tej ścieżki nie zostanie podzielona, to najmniejsza funkcja dominowania rzymskiego przypisze wierzchołkowi u wartość 0, ponieważ jest on dominowany przez v , natomiast wierzchołkowi w zostanie przypisana wartość 1. Brak podziału krawędzi w tym podgrafie będzie oznaczał, że w podgrafie $K_{1,q}$ zostało podzielonych $\lfloor \frac{m(DS_{1,q})}{2} \rfloor = \lfloor \frac{q+2}{2} \rfloor$ krawędzi. Liczba dominowania rzymskiego wynikowego grafu będzie więc równa $\lfloor \frac{q+2}{2} \rfloor + 3$.
2. Jeżeli zostanie podzielona krawędź wu , to wierzchołek u będzie nadal dominowany przez v , więc zostanie mu przypisana wartość 0, natomiast wierzchołkom w i nowo utworzonemu zostanie przypisana wartość 1. Łatwo zauważyć, że przypadek podziału krawędzi uv będzie miał identyczny efekt, ponieważ będzie skutkowało powstaniem izomorficznego grafu wynikowego. Oznacza to również, że $\lfloor \frac{m(DS_{1,q})}{2} \rfloor - 1 = \lfloor \frac{q+2}{2} \rfloor - 1$ spośród krawędzi podgrafu $K_{1,q}$ zostało podzielonych. W związku z tym liczba dominowania rzymskiego w grafie wynikowym będzie równa $\lfloor \frac{q+2}{2} \rfloor - 1 + 4 = \lfloor \frac{q+2}{2} \rfloor + 3$.
3. Jeżeli zostaną podzielone obie krawędzie wu i wv , to wierzchołkowi utworzonemu w wyniku podziału krawędzi wu najmniejsza funkcja dominowania rzymskiego przypisze wartość 2, wierzchołkowi v również wartość 2, natomiast wierzchołkom w , u i wierzchołkowi powstałemu w wyniku podziału krawędzi wv wartość 0. W tej sytuacji w podgrafie $K_{1,q}$ podzielonych

zostanie $\left\lfloor \frac{m(DS_{1,q})}{2} \right\rfloor - 2 = \left\lfloor \frac{q+2}{2} \right\rfloor - 2$ krawędzi i, co za tym idzie, tyłu liściom tego podgrafu zostanie przypisana wartość minimalnej funkcji dominowania rzymskiego równa 1. Pozostałe wierzchołki będą miały przypisaną sumaryczną wartość równą 4, więc liczba dominowania rzymskiego grafu wynikowego wyniesie $\left\lfloor \frac{q+2}{2} \right\rfloor - 2 + 4 = \left\lfloor \frac{q+2}{2} \right\rfloor + 2$.



Rys. 4.7. Przykład gry w dominowanie rzymskie z podziałem krawędzi na grafie $DS_{1,6}$. Wewnątrz wierzchołków wpisane zostały wartości najmniejszej funkcji dominowania rzymskiego dla grafu wynikowego. Jak widać, niezależnie od tego, które krawędzie zostaną podzielone, nie zmienia się suma wartości. Przykłady odpowiadają przypadkom rozważanym w dowodzie twierdzenia 4.14

Jak widać, niezależnie od tego, które krawędzie zostaną podzielone, będzie prawdziwa równość

$$\begin{aligned} \gamma_R(DS'_{1,q}) &= \left\lfloor \frac{q+2}{2} \right\rfloor + 2 \\ \implies \gamma_{Rgs}(DS_{1,q}) &= \left\lfloor \frac{q+2}{2} \right\rfloor + 2 = \left\lfloor \frac{n(DS_{1,q}) - 1}{2} \right\rfloor + 2 = \left\lfloor \frac{n(DS_{1,q})}{2} \right\rfloor + 2. \end{aligned}$$

□

Twierdzenie 4.15. [2] W podwójnej gwiazdzie $DS_{p,q}$, gdzie $2 \leq p \leq q$, zachodzi

$$\gamma_{Rgs}(DS_{p,q}) = \left\lceil \frac{n(DS_{p,q}) + 1}{2} \right\rceil + 2.$$

Dowód. Graf $DS_{p,q}$ ma $n(DS_{p,q}) = p + q + 2$ wierzchołków. Liczba krawędzi natomiast wynosi $m(DS_{p,q}) = p + q + 1 = n - 1$. W zbiorze krawędzi zawiera się p krawędzi w podgrafie $K_{1,p}$, q krawędzi w podgrafie $K_{1,q}$ oraz jedna krawędź uv , gdzie u jest wierzchołkiem centralnym $K_{1,p}$, a v jest wierzchołkiem centralnym $K_{1,q}$.

Zauważmy, że, podobnie jak w poprzednim dowodzie, liczba dominowania rzymskiego grafu $DS'_{p,q}$, powstałego w wyniku gry w dominowanie rzymskie z podziałem krawędzi, będzie równa sumie

- liczby krawędzi podzielonych w podgrafie $K_{1,p}$ (oznaczanej dalej jako p'),
- liczby krawędzi podzielonych w podgrafie $K_{1,q}$ (oznaczanej dalej jako q') oraz
- wartości najmniejszej funkcji dominowania rzymskiego dla wierzchołków u i v ,

która wyniesie 2 dla każdego z nich: $\gamma_R(DS'_{p,q}) = p' + q' + 4$.

Przyjmijmy symbol η , którego wartość jest równa 1, jeżeli krawędź uv została podzielona lub 0 w przeciwnym wypadku. W wyniku ruchów wykonanych przez Psuja zostanie zatem podzielone $\left\lfloor \frac{n(DS_{p,q})-1}{2} \right\rfloor$ krawędzi. Zauważmy, że $\left\lfloor \frac{n(DS_{p,q})-1}{2} \right\rfloor = p' + q' + \eta$.

Rozważmy dwa przypadki:

Przypadek 1.: $2 \mid n(DS_{p,q})$

Wtedy $p' + q' + \eta = \frac{n(DS_{p,q})-2}{2}$. Stąd mamy

$$\gamma_R(DS'_{p,q}) = \frac{n(DS_{p,q}) - 2}{2} + 4 - \eta = \frac{n(DS_{p,q})}{2} + 3 - \eta.$$

Przypadek 2.: $2 \nmid n(DS_{p,q})$

Wówczas $p' + q' + \eta = \frac{n(DS_{p,q})-1}{2}$. Stąd

$$\gamma_R(DS'_{p,q}) = \frac{n(DS_{p,q}) - 1}{2} + 4 - \eta = \frac{n(DS_{p,q}) + 1}{2} + 3 - \eta.$$

Zauważmy, że w obu przypadkach liczba dominowania rzymskiego jest mniejsza, jeżeli krawędź uv została podzielona. Wobec tego naturalną strategią dla Psuja będzie dzielenie krawędzi innych niż krawędź uv , jeżeli jest to tylko możliwe. Najlepsza strategia Dominatora będzie wtedy próbowała wymusić na przeciwniku podział krawędzi uv poprzez oznaczanie wszystkich innych krawędzi, dopóki to możliwe.

Jeżeli liczba wierzchołków $DS_{p,q}$ jest parzysta, to liczba krawędzi innych od uv również jest parzysta. Oznacza to, że korzystając z powyższej strategii Dominator nie będzie w stanie wymusić na Psuju podziału krawędzi uv , więc

$$2 \mid n(DS_{p,q}) \implies \eta = 0 \implies \gamma_R(DS'_{pq}) = \frac{n(DS_{p,q})}{2} + 3.$$

W przeciwnym wypadku liczba krawędzi innych niż uv będzie nieparzysta i Ψ_{uj} będzie zmuszony do podzielenia tej krawędzi, stąd

$$2 \nmid n(DS_{p,q}) \implies \eta = 1 \implies \gamma_R(DS'_{pq}) = \frac{n(DS_{p,q}) + 1}{2} + 2.$$

Oba powyższe przypadki można opisać z użyciem jednego wzoru:

$$\gamma_{Rgs}(DS_{p,q}) = \gamma_R(DS'_{pq}) = \left\lceil \frac{n(DS_{p,q}) + 1}{2} \right\rceil + 2.$$

□

4.3.7. Drzewa

Twierdzenie 4.16. [2] W każdym drzewie T zachodzi

$$\gamma_{Rgs}(T) \leq 2\gamma_2(T) - 1.$$

Dowód. Dowód jest przeprowadzony za pomocą indukcji względem liczby wierzchołków w drzewie T . Rozważmy przypadki małych wartości $n(T)$:

- $n(T) = 2$

Wtedy w grafie znajduje się jedna krawędź, która zostanie oznaczona przez Dominatora, więc graf wynikowy $T' = P_2$. Stąd, korzystając z lematu 4.2, $\gamma_{Rgs}(T) = \gamma_R(T') = \left\lceil \frac{2n(T')}{3} \right\rceil = 2$. Zbiór 2-dominujący natomiast będzie zawierał oba wierzchołki, więc $\gamma_2(T) = 2$. Stąd

$$\gamma_{Rgs}(T) = 2 \leq 2\gamma_2(T) - 1 = 3.$$

- $n(T) = 3$

Wtedy jedna z krawędzi grafu zostanie podzielona, więc graf wynikowy $T' = P_4$. Analogicznie do powyższego przykładu $\gamma_{Rgs}(T) = \gamma_R(T') = \left\lceil \frac{3n(T')}{4} \right\rceil = 3$. Zbiór 2-dominujący w grafie T będzie również zawierał dwa wierzchołki, stanowiące pierwszy i ostatni wierzchołek jego najdłuższej ścieżki, więc $\gamma_2(T) = 2$. Stąd

$$\gamma_{Rgs}(T) = 3 \leq 2\gamma_2(T) - 1 = 3.$$

Wiemy więc, że twierdzenie jest prawdziwe dla $n(T) \leq 3$. W dalszej części dowodu rozważymy przypadki drzew T , gdzie $n(T) \geq 4$. Przyjmijmy założenie indukcyjne, że twierdzenie jest prawdziwe dla drzewa o liczbie wierzchołków mniejszej od $n(T)$.

Jeżeli średnica T jest równa 1, to $T = P_2$ i, jak pokazano wcześniej, spełnia nierówność postawioną w twierdzeniu.

Jeżeli średnica T jest równa 2, to T jest gwiazdą. Wtedy z twierdzenia 4.1 $\gamma_{Rgs}(T) = \left\lceil \frac{n(T)+2}{2} \right\rceil$. Ponieważ wszystkie z liści mają tylko jeden wierzchołek sąsiadujący, to wszystkie z nich muszą być częścią najmniejszego zbioru 2-dominującego w T . Wtedy, dla $n \geq 3$, wierzchołek centralny będzie 2-dominowany przez zbiór liści, więc $\gamma_2(T) = n(T) - 1$. Ponieważ

$$\gamma_{Rgs}(T) = \left\lceil \frac{n(T) + 2}{2} \right\rceil \leq 2\gamma_2(T) - 1 = 2n(T) - 3,$$

to twierdzenie w tym przypadku jest prawdziwe.

Jeśli natomiast średnica grafu T jest równa 3, to T jest podwójną gwiazdą. Z twierdzenia 4.15 mamy wtedy $\gamma_{Rgs}(T) = \left\lceil \frac{n(T)+1}{2} \right\rceil + 2$. Najmniejszy zbiór 2-dominujący ponownie musi zawierać wszystkie liście, a w przypadku, gdy jeden z podgrafów ma tylko jeden liść, to musi również zawierać jeden wierzchołek centralny, czyli $\gamma_2(T) \leq n(T) - 1$.

$$\gamma_{Rgs}(T) = \left\lceil \frac{n(T)+1}{2} \right\rceil + 2 \leq 2\gamma_2(T) - 1 \leq 2n(T) - 3,$$

więc twierdzenie jest prawdziwe także w tym przypadku.

Założmy więc w dalszej części dowodu, że średnica T jest większa od lub równa 4.

Niech P będzie najdłuższą ścieżką w grafie T i niech $V(P) = \{v_1, \dots, v_{k-1}, v_k\}$.

Niech D będzie najmniejszym zbiorem 2-dominującym w T niezawierającym wierzchołka v_{k-1} , czyli $\gamma_2(T) = |D|$. Istnienie takiego zbioru jest pewne, gdyż v_{k-1} sąsiaduje jedynie z wierzchołkiem v_{k-2} oraz liśćmi. Liście, mając tylko jeden wierzchołek sąsiadujący, muszą należeć do każdego zbioru 2-dominującego. Jeżeli $d(v_{k-1}) \geq 3$, to v_{k-1} jest 2-dominowany przez same liście. W przeciwnym wypadku v_k jest liściem i należy do zbioru 2-dominującego, natomiast zbiór 2-dominujący zawierający v_{k-2} , co zapewnia 2-dominowanie wierzchołka v_{k-1} , jest na pewno nie większy od zbioru zawierającego v_{k-1} .

Rozważmy dwa przypadki:

Przypadek 1. $d(v_{k-1}) = 2$

W tym przypadku v_k jest jedynym liściem sąsiadującym z v_{k-1} . Niech T_1 będzie grafem indukowanym z T niezawierającym wierzchołków $\{v_{k-1}, v_k\}$ ³.

Zauważmy, że $\gamma_2(T_1) \leq |D| - 1$, ponieważ wierzchołek v_k musiał zawierać się w D , żeby v_{k-1} mógł być 2-dominowany. Liczba 2-dominowania T_1 może być mniejsza od $|D| - 1$ dlatego, że wierzchołek v_{k-2} , który musi być zawarty w D , żeby wierzchołek v_{k-1} był 2-dominowany, nie musi być koniecznie zawarty w najmniejszym zbiorze 2-dominującym grafu T_1 .

Niech Dominator rozpocznie grę od optymalnego ruchu w grafie T_1 , a następnie:

- jeżeli Psuj wykona ruch w grafie T_1 , to Dominator wykonuje ruch według optymalnej strategii w grafie T_1 ;
- jeżeli Psuj wykona ruch na jednej z krawędzi $\{v_{k-2}v_{k-1}, v_{k-1}v_k\}$, to Dominator wykonuje ruch na drugiej z tych krawędzi.

W ten sposób w czasie gry zostanie podzielona tylko jedna spośród krawędzi T znajdujących się poza T_1 . W grafie T' znajdzie się więc ścieżka $\{v_{k-2}, s, v_{k-1}, v_k\}$ lub $\{v_{k-2}, v_{k-1}, s, v_k\}$, gdzie s jest wierzchołkiem powstałym w wyniku podziału krawędzi. Najmniejsza funkcja dominowania rzymskiego dla grafu wynikowego T'_1 może wówczas zostać rozszerzona w następujący sposób aby objąć graf wynikowy T' :

- przedostatniemu wierzchołkowi ścieżki (v_{k-1} w przypadku pierwszej ścieżki i s w drugiej) jest przypisana wartość 2;
- drugiemu i ostatniemu wierzchołkowi ścieżki jest przypisana wartość 0.

To rozszerzenie może nie być najmniejszą możliwą funkcją dominowania rzymskiego dla grafu T' , więc

$$\gamma_{Rgs}(T) \leq \gamma_{Rgs}(T_1) + 2.$$

³Sytuacja ta jest przedstawiona na rysunku 3.9.

Z założenia indukcyjnego mamy $\gamma_{Rgs}(T_1) \leq 2\gamma_2(T_1) - 1$, a z wcześniejszej części dowodu wynika $\gamma_2(T_1) \leq |D| - 1 \implies \gamma_2(T_1) \leq \gamma_2(T) - 1$, stąd

$$\begin{aligned}\gamma_{Rgs}(T) &\leq \gamma_{Rgs}(T_1) + 2 \leq 2\gamma_2(T_1) + 1 \\ &\leq 2(\gamma_2(T) - 1) + 1 = 2\gamma_2(T) - 1.\end{aligned}$$

Przypadek 2. $d(v_{k-1}) \geq 3$

Wierzchołek v_{k-1} sąsiaduje z $d(v_{k-1}) - 1$ liśćmi, które muszą należeć do D . Niech T_1 będzie grafem indukowanym z T , niezawierającym wierzchołka v_k oraz sąsiadujących z nim liści⁴.

Jeżeli D jest zbiorem 2-dominującym w grafie T , to D bez liści będzie również zbiorem 2-dominującym w grafie T_1 , natomiast, podobnie jak w poprzednim rozważanym przypadku, nie musi być najmniejszym zbiorem 2-dominującym. Stąd i z założenia $d(v_{k-1}) \geq 3$ mamy

$$\gamma_2(T_1) \leq |D| - (d(v_{k-1}) - 1) = \gamma_2(T) - d(v_{k-1}) + 1.$$

Niech Dominator rozpocznie grę od optymalnego ruchu w grafie T_1 , a następnie:

- jeżeli Psuj wykona ruch w grafie T_1 , to Dominator wykonuje optymalny ruch w grafie T_1 ;
- jeżeli Psuj wykona ruch na jednej z krawędzi sąsiadujących z wierzchołkiem v_{k-1} , to Dominator wykonuje ruch na jednej z krawędzi pomiędzy v_{k-1} a liściem. Jeżeli to niemożliwe, wykonuje dowolny ruch w T_1 .

Strategia ta gwarantuje, że w czasie gry podzielone zostanie co najwyżej $\left\lfloor \frac{d(v_{k-1})-1}{2} \right\rfloor = \left\lfloor \frac{d(v_{k-1})}{2} \right\rfloor$ krawędzi pomiędzy v_{k-1} , a liśćmi. Podobnie jak w poprzednim przypadku możliwe jest rozszerzenie najmniejszej funkcji dominowania rzymskiego dla grafu wynikowego T'_1 na graf wynikowy T' . Wierzchołkowi v_{k-1} może zostać przypisana wartość 2, natomiast liściom sąsiadującym z podzielonymi krawędziami wartości 1. Wtedy suma wartości tej funkcji dominowania rzymskiego wyniesie $\gamma_{Rgs}(T'_1) + 2 + \left\lfloor \frac{d(v_{k-1})}{2} \right\rfloor \cdot 1$. Najmniejsza funkcja dominowania rzymskiego dla T' może mieć co najwyżej taką wartość, więc

$$\gamma_{Rgs}(T) = \gamma_R(T') \leq \gamma_{Rgs}(T_1) + 2 + \left\lfloor \frac{d(v_{k-1})}{2} \right\rfloor.$$

Korzystając z założenia indukcyjnego, mamy $\gamma_{Rgs}(T_1) \leq 2\gamma_2(T_1) - 1$. Weźmy też wcześniej wyprowadzoną własność $\gamma_2(T_1) \leq \gamma_2(T) - d(v_{k-1}) + 1$. Stąd i z powyższego

$$\begin{aligned}\gamma_{Rgs}(T) &\leq \gamma_{Rgs}(T_1) + 2 + \left\lfloor \frac{d(v_{k-1})}{2} \right\rfloor \\ &\leq 2\gamma_2(T_1) - 1 + 2 + \left\lfloor \frac{d(v_{k-1})}{2} \right\rfloor \\ &\leq 2(\gamma_2(T) - d(v_{k-1}) + 1) + 1 + \left\lfloor \frac{d(v_{k-1})}{2} \right\rfloor \\ &= 2\gamma_2(T) - 2d(v_{k-1}) + 3 + \left\lfloor \frac{d(v_{k-1})}{2} \right\rfloor.\end{aligned}$$

⁴Sytuacja ta jest przedstawiona na rysunku 3.10.

Dodatkowo, ponieważ $d(v_{k-1}) \geq 3$, to $-2d(v_{k-1}) + 3 + \left\lfloor \frac{d(v_{k-1})}{2} \right\rfloor \leq -2$, więc

$$\gamma_{Rgs}(T) \leq 2\gamma_2(T) - 2d(v_{k-1}) + 3 + \left\lfloor \frac{d(v_{k-1})}{2} \right\rfloor \leq 2\gamma_2(T) - 2 < 2\gamma_2(T) - 1.$$

□

Twierdzenie 4.17. [2] W każdym drzewie T zachodzi

$$\gamma_{Rgs}(T) \geq \left\lceil \frac{n+2}{2} \right\rceil,$$

gdzie n jest liczbą wierzchołków w T .

Dowód. Jeżeli $n = 2$, to T jest ścieżką P_2 , więc $\gamma_{Rgs}(T) = 2 \geq \left\lceil \frac{n+2}{2} \right\rceil = 2$, czyli własność z twierdzenia zachodzi.

Jeżeli $n = 3$, to T jest ścieżką P_3 , więc graf wynikowy T' jest ścieżką P_4 . $\gamma_{Rgs}(T) = \gamma_R(T') = 3 \geq \left\lceil \frac{n+2}{2} \right\rceil = 3$, co oznacza, że w tym wypadku własność również zachodzi.

W przypadku, gdy graf T ma średnicę mniejszą od lub równą 2, to T jest gwiazdą, więc, z twierdzenia 4.1, $\gamma_{Rgs}(T) = \left\lceil \frac{n+2}{2} \right\rceil$. Natomiast jeżeli graf T ma średnicę równą 3, to jest on podwójną gwiazdą, czyli, z twierdzenia 4.15, mamy $\gamma_{Rgs}(T) = \left\lceil \frac{n+1}{2} \right\rceil + 2 = \left\lceil \frac{n+5}{2} \right\rceil > \left\lceil \frac{n+2}{2} \right\rceil$. W obu tych przypadkach nierówność postawiona w twierdzeniu jest prawdziwa.

Założmy więc dalej, że $n(T) \geq 4$ oraz średnica T jest większa lub równa 4. Dodatkowo, przyjmijmy założenie indukcyjne, że dla każdego drzewa T_s , gdzie $n(T_s) < n(T)$, zachodzi $\gamma_{Rgs}(T_s) \geq \left\lceil \frac{n(T_s)+2}{2} \right\rceil$.

Niech $P = \{v_1, v_2, \dots, v_{k-1}, v_k\}$ będzie najdłuższą ścieżką w T . Niech $\{v_1, u_1, \dots, u_{d(v_2)-2}\}$ będą liśćmi sąsiadującymi z v_2 , a $\{v_k, u'_1, \dots, u'_{d(v_{k-1})-2}\}$ niech będą liśćmi sąsiadującymi z v_{k-1} .

Rozważmy przypadek, gdy średnica T wynosi 4 oraz $d(v_3) = d(v_{k-2}) = 3$. Łatwo zauważyć, że w tym przypadku $v_3 = v_{k-1}$. Niech Psuj przyjmie następującą strategię:

- jeżeli Dominator oznaczy jedną z krawędzi $\{v_2v_3, v_3v_4\}$, to Psuj dzieli drugą z nich;
- jeżeli Dominator oznaczy jedną z pozostałych krawędzi (łączyjąc jeden z liści z v_2 lub v_4), to Psuj dzieli inną z tych krawędzi.

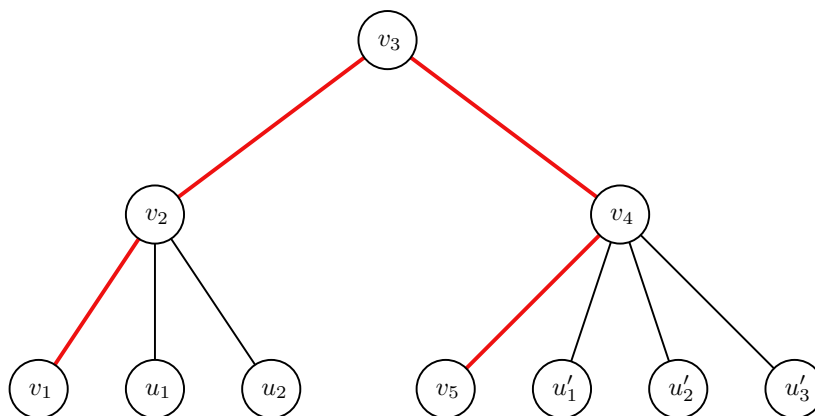
Wówczas zawsze będzie istnieć najmniejsza funkcja dominowania rzymskiego, która w grafie wynikowym przydzieli wartość 2 wierzchołkom v_2 i v_4 , wartość 1 liściom sąsiadującym z podzielonymi krawędziami oraz wartość 0 pozostałym wierzchołkom. Ponieważ strategia Psuja zawsze odpowiada na ruchy Dominatora, zostanie podzielone $\left\lfloor \frac{n-2}{2} \right\rfloor$ spośród $n - 2$ krawędzi sąsiadujących z liśćmi, więc

$$\gamma_{Rgs}(T) \geq 2 \cdot 2 + \left\lfloor \frac{n-2}{2} \right\rfloor \cdot 1 = \left\lfloor \frac{n+6}{2} \right\rfloor = \left\lceil \frac{n+5}{2} \right\rceil > \left\lceil \frac{n+2}{2} \right\rceil,$$

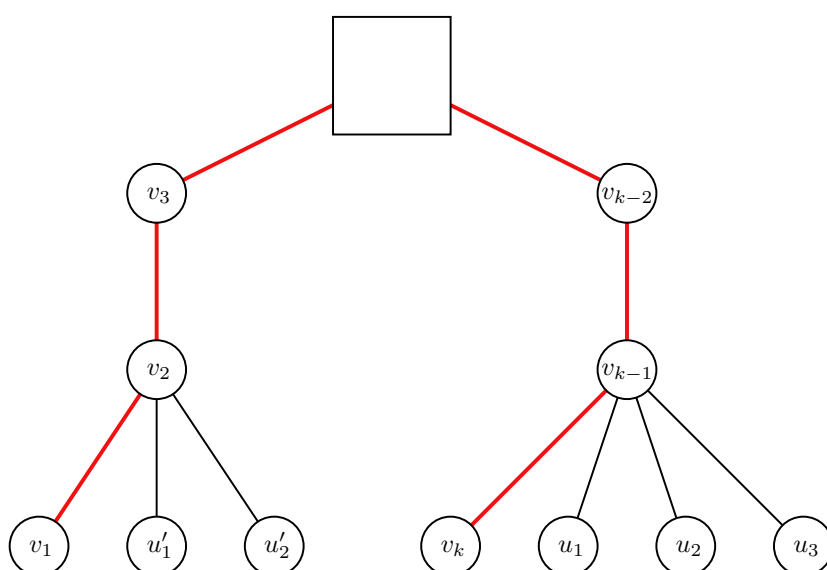
co kończy dowód w tym przypadku.

W dalszej części dowodu założmy, że średnica T jest większa od 4 lub $d(v_3) \geq 3$.

Rozważmy teraz przypadek, w którym $d(v_2)$ jest parzyste. Przypadek ten jest analogiczny do przypadku, gdy $d(v_{k-1})$ jest parzyste, ponieważ wystarczy wówczas odwrócić numerację



Rys. 4.8. Drzewo T o średnicy 4, $v_3 = v_{k-2}$, gdzie $d(v_3) = d(v_{k-2}) = 3$. Najdłuższa ścieżka P została oznaczona kolorem czerwonym



Rys. 4.9. Drzewo T , w którym średnica jest większa od 4. Najdłuższa ścieżka P została oznaczona kolorem czerwonym. Kwadratem została oznaczona część drzewa oddalona od v_1 oraz v_k o dystans większy niż 2 krawędzie

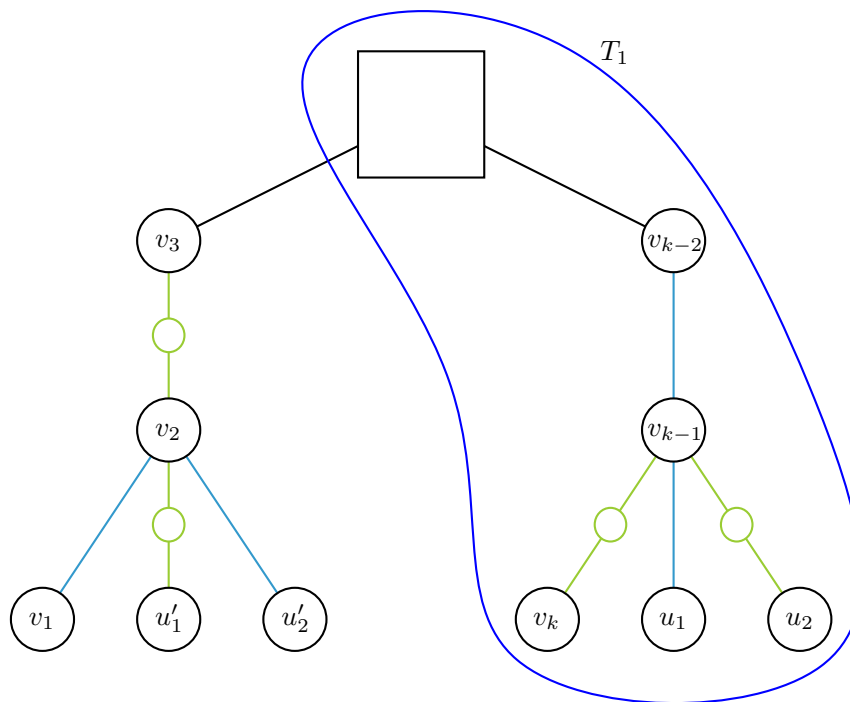
wierzchołków w ścieżce P . Niech T_1 będzie podgrafem indukowanym na wierzchołkach $V(T) \setminus \{v_2, v_1, u_1, \dots\}$, czyli niezawierającym poddrzewa, składającego się z wierzchołka v_2 oraz sąsiadujących z nim liści. Wtedy Psuj może przyjąć strategię:

- jeżeli Dominator wykonał ruch na krawędzi należącej do T_1 , to Psuj wykonuje ruch według optymalnej strategii na T_1 ;
- jeżeli Dominator wykonał ruch na jednej z pozostałych krawędzi $\{v_2v_3, v_2v_1, v_2u_1, \dots\}$, to Psuj wykonuje ruch na dowolnej innej spośród tych krawędzi.

Wówczas $\gamma_R(T'_1) = \gamma_{Rgs}(T_1)$, gdzie T'_1 jest grafem wynikowym gry w dominowanie rzymskie z podziałem krawędzi na podgrafie T_1 . W wyniku użycia powyższej strategii możliwe są dwa przypadki:

Przypadek 1. Krawędź v_2v_3 została podzielona.

Wtedy spośród krawędzi pomiędzy wierzchołkiem v_2 , a sąsiadującymi liśćmi, zostało podzielone $\frac{d(v_2)-2}{2}$ krawędzi. Najmniejsza funkcja dominowania rzymskiego dla podgrafu T'_1 może zostać rozszerzona tak, aby była funkcją dominowania rzymskiego dla grafu wynikowego T' :



Rys. 4.10. Drzewo T' , w którym $d(v_2)$ jest parzyste. Kwadratem została oznaczona część drzewa oddalona od v_1 oraz v_k w grafie początkowym o dystans większy niż 2 krawędzie. Kolorem niebieskim został oznaczony podgraf T_1' , którego dolne ograniczenie liczby dominowania rzymskiego jest założeniem indukcyjnym

- wartość 2 jest przydzielona wierzchołkowi v_2 ;
- wartość 1 jest przydzielona $\frac{d(v_2)-2}{2}$ liściom sąsiadującym z podzielonymi krawędziami;
- wartość 0 jest przydzielona pozostałym wierzchołkom (w tym wszystkim wierzchołkom dodanym w wyniku podziału krawędzi, ponieważ są dominowane przez wierzchołek v_2).

Suma tej funkcji dominowania rzymskiego jest wtedy równa $\gamma_{Rgs}(T_1) + 2 + \frac{d(v_2)-2}{2}$. Zauważmy, że wierzchołek powstały w wyniku podziału krawędzi v_2v_3 jest dominowany przez wierzchołek v_2 i ma przypisaną wartość 0, więc jest to najmniejsza funkcja dominowania rzymskiego dla grafu T' niezależnie od tego, jaka wartość została przypisana wierzchołkowi $v_3 \in T_1$. Stąd

$$\gamma_R(T') = \gamma_{Rgs}(T_1) + 2 + \frac{d(v_2) - 2}{2} = \gamma_{Rgs}(T_1) + \frac{d(v_2)}{2} + 1.$$

Przypadek 2. Krawędź v_2v_3 nie została podzielona.

Wówczas pomiędzy wierzchołkiem v_2 a sąsiadującymi z nim liśćmi zostało podzielone $\frac{d(v_2)}{2}$ krawędzi. Po rozszerzeniu najmniejszej funkcji dominowania rzymskiego grafu T_1' tak, aby włączyć do jej dziedziny cały graf T analogicznie do przypadku 1., ma ona sumaryczną wartość $\gamma_{Rgs}(T_1) + 2 + \frac{d(v_2)}{2}$. Niezależnie od wartości przypisanej wierzchołkowi v_3 , najmniejszą wartością funkcji dominowania rzymskiego dla wierzchołka v_2 w T' będzie 2, więc tak rozszerzona funkcja jest najmniejszą funkcją dominowania rzymskiego dla T' . Stąd

$$\gamma_R(T') = \gamma_{Rgs}(T_1) + \frac{d(v_2)}{2} + 2.$$

Z powyższych przypadków wynika, że

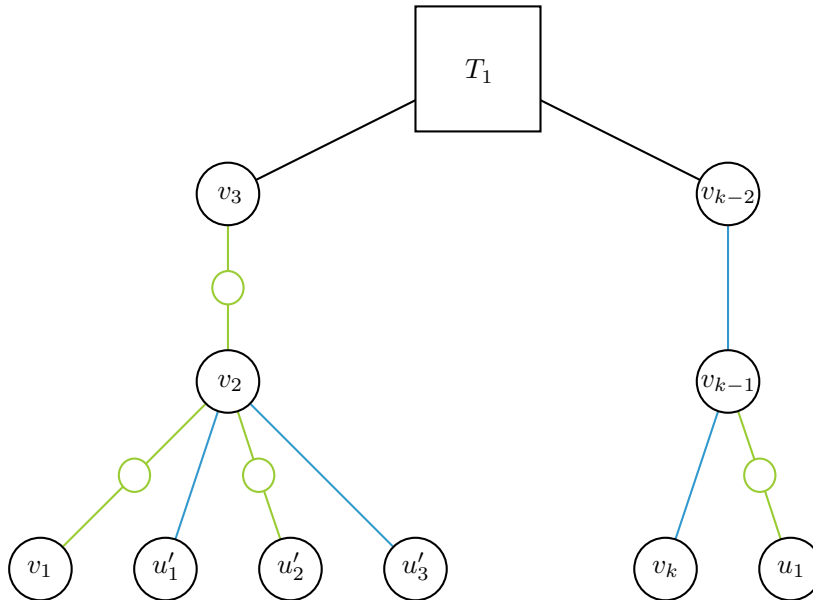
$$\gamma_{Rgs}(T) = \gamma_R(T') \geq \gamma_{Rgs}(T_1) + \frac{d(v_2)}{2} + 1.$$

Korzystając z założenia indukcyjnego, otrzymujemy więc

$$\begin{aligned} \gamma_{Rgs}(T) &\geq \left\lceil \frac{n(T_1) + 2}{2} \right\rceil + \frac{d(v_2)}{2} + 1 \\ &= \left\lceil \frac{n(T) - d(v_2) + 2}{2} \right\rceil + \frac{d(v_2)}{2} + 1 \\ &= \left\lceil \frac{n(T) + 2}{2} \right\rceil + 1 > \left\lceil \frac{n(T) + 2}{2} \right\rceil, \end{aligned}$$

co dowodzi twierdzenia dla przypadku, gdy $d(v_2)$ lub $d(v_{k-1})$ jest parzyste.

Przyjmijmy więc dalej założenie, że zarówno $d(v_2)$, jak i $d(v_{k-1})$ są nieparzyste. Niech T_1 będzie podgrafem indukowanym na wierzchołkach $V(T) \setminus \{v_2, v_1, u_1, \dots, v_{k-1}, v_k, u'_1, \dots\}$.



Rys. 4.11. Drzewo T' , w którym $d(v_2)$ i $d(v_{k-1})$ są nieparzyste. Kwadratem został oznaczony podgraf T_1 , którego dolne ograniczenie liczby gry dominowania rzymskiego jest założeniem indukcyjnym

W takiej sytuacji niech Psuj gra według następującej strategii:

- jeśli Dominator wykonał ruch na krawędzi należącej do T_1 , to Psuj dzieli inną krawędź w T_1 według optymalnej strategii;
- jeśli Dominator oznaczył jedną z krawędzi $\{v_2v_3, v_{k-1}, v_{k-2}\}$, to Psuj dzieli drugą z nich;
- jeśli Dominator oznaczył jedną z pozostałych krawędzi (łączyjących v_2 i v_{k-1} z sąsiadującymi z nimi wierzchołkami), to Psuj dzieli inną krawędź z tego zbioru.

Wtedy po zakończeniu gry dokładnie jedna z krawędzi $\{v_2v_3, v_{k-1}, v_{k-2}\}$ będzie podzielona. Dodatkowo $\frac{d(v_2)+d(v_{k-1})-2}{2}$ krawędzi łączących wierzchołki v_2 i v_{k-1} z sąsiadującymi liśćmi zostanie podzielone. Podobnie do poprzedniego przypadku, również w tej sytuacji można rozszerzyć najmniejszą funkcję dominowania rzymskiego dla grafu wynikowego T'_1 na graf wynikowy T' . Poszczególnym wierzchołkom zostaną przypisane następujące wartości:

- wierzchołkom v_2 oraz v_{k-1} zostanie przypisana wartość 2;

- liściom, których krawędzie łączące je z v_2 lub v_{k-1} zostały podzielone, zostanie przypisana wartość 1;
- pozostałym wierzchołkom zostanie przypisana wartość 0.

Również podobnie do poprzedniego przypadku, niezależnie od wartości przypisanych wierzchołkom v_3 i v_{k-2} , ta rozszerzona funkcja jest najmniejszą funkcją dominowania rzymskiego.

Wtedy $\gamma_{Rgs}(T) = \gamma_R(T') = \gamma_{Rgs}(T_1) + \frac{d(v_2) + d(v_{k-1}) - 2}{2} + 4$. Z założenia indukcyjnego mamy więc

$$\begin{aligned} \gamma_{Rgs}(T) &\geq \left\lceil \frac{n(T_1) + 2}{2} \right\rceil + \frac{d(v_2) + d(v_{k-1}) - 2}{2} + 4 \\ &= \left\lceil \frac{n(T) - d(v_2) - d(v_{k-1}) + 2}{2} \right\rceil + \frac{d(v_2) + d(v_{k-1}) - 2}{2} + 4 \\ &= \left\lceil \frac{n(T) + 2}{2} \right\rceil + 4 > \left\lceil \frac{n(T) + 2}{2} \right\rceil, \end{aligned}$$

co kończy dowód. □

Wierzchołek wspierający jest wierzchołkiem sąsiadującym z przynajmniej jednym liściem. *Wierzchołek wspierający końcowy* jest wierzchołkiem wspierającym, w którego sąsiedztwie znajduje się tylko jeden wierzchołek niebędący liściem.

Twierdzenie 4.18. [2] Dla drzewa T o $n \geq 2$ wierzchołkach zachodzi

$$\gamma_{Rgs}(T) \leq n.$$

Dowód. Jeżeli $n \in \{2, 3\}$, to T jest izomorficzne odpowiednio do P_2 lub P_3 , więc z twierdzenia 4.3 zachodzi $\gamma_{Rgs}(T) = n$.

Jeżeli T ma średnicę równą 2, to T jest gwiazdą i z twierdzenia 4.1 mamy

$$\gamma_{Rgs}(T) = \left\lceil \frac{n+2}{2} \right\rceil = \left\lceil \frac{n}{2} \right\rceil + 1 \leq n \text{ dla } n \geq 4.$$

Jeżeli T ma średnicę równą 3, to T jest podwójną gwiazdą i z twierdzenia 4.15 wynika, że

$$\gamma_{Rgs}(T) = \left\lceil \frac{n+1}{2} \right\rceil + 2 \leq n \text{ dla } n \geq 5.$$

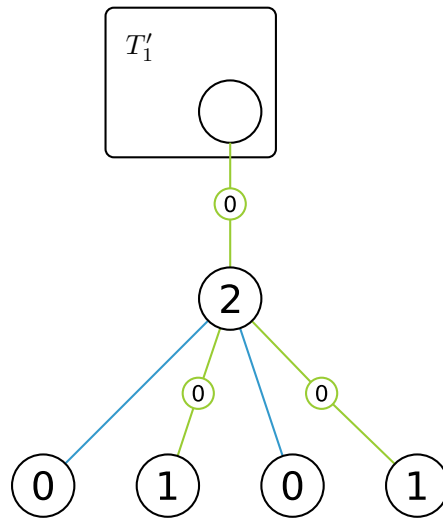
Jedyną podwójną gwiazdą, dla której $n < 5$, jest $DS_{1,1}$, gdzie $n = 4$. Jest ona izomorficzna do P_4 , więc z twierdzenia 4.3 $\gamma_{Rgs}(T) = \gamma_{Rgs}(DS_{1,1}) = 4$.

Założmy więc dalej, że T ma średnicę większą niż 3 oraz $n \geq 4$. Ponieważ wiemy, że twierdzenie jest spełnione dla $n \leq 3$, to przyjmijmy założenie indukcyjne, że twierdzenie jest prawdziwe dla każdego drzewa o mniejszej liczbie wierzchołków niż rozpatrywane.

Rozważmy różne strategie dla Dominatora, gwarantujące górną granicę liczby gry w dominowanie rzymskie w zależności od liczebności wierzchołków wspierających końcowych w T .

Przyjmijmy najpierw, że istnieje przynajmniej jeden wierzchołek wspierający końcowy x , gdzie $d(x) \geq 5$. Niech T_1 będzie podgrafem T niezawierającym wierzchołka x oraz wierzchołków $\{y_1, y_2, \dots, y_{d(x)-1}\}$, będących liśćmi sąsiadującymi z x . Dodatkowo niech z będzie wierzchołkiem sąsiadującym z x , który nie jest liściem. Wtedy Dominator może rozpocząć grę od podziału krawędzi w T_1 zgodnego z optymalną strategią, a następnie grać według następującej strategii:

- dopóki Psuj dzieli krawędzie w T_1 , to Dominator gra w T_1 , używając optymalnej strategii;
- jeżeli Psuj podzieli jedną z pozostałych krawędzi, to Dominator oznacza wolną krawędź w $\{xy_1, xy_2, \dots, xy_{d(x)-1}\}$.



Rys. 4.12. Drzewo T' , w którym przynajmniej jeden wierzchołek wspierający końcowy jest stopnia większego od lub równego 5. W wierzchołkach zostały wpisane wartości rozszerzenia najmniejszej funkcji dominowania w podgrafie T'_1 na pozostałe wierzchołki

Grając zgodnie z tą strategią, Dominator będzie w stanie oznaczyć przynajmniej $\lfloor \frac{d(x)}{2} \rfloor$ krawędzi z $\{xy_1, xy_2, \dots, xy_{d(x)-1}\}$. Można rozszerzyć najmniejszą funkcję dominowania rzymskiego dla grafu wynikowego T'_1 gry w dominowanie rzymskie z podziałem krawędzi na T_1 tak, aby pokrywała cały graf wynikowy T' gry na grafie T . Wierzchołkowi x może zostać przypisana wartość 2, natomiast liściom sąsiadującym z podzielonymi krawędziami, łączącymi je oryginalnie z x , można przypisać wartość 1. Wtedy

$$\begin{aligned} \gamma_R(T') &= \gamma_R(T'_1) + 2 + \left\lfloor \frac{d(x)}{2} \right\rfloor \\ \implies \gamma_{Rgs}(T) &\leq \gamma_{Rgs}(T_1) + 2 + \left\lfloor \frac{d(x)}{2} \right\rfloor. \end{aligned}$$

Stosując założenie indukcyjne oraz $d(x) \geq 5$, mamy więc

$$\begin{aligned} \gamma_{Rgs}(T) &\leq n - d(x) + 2 + \left\lfloor \frac{d(x)}{2} \right\rfloor \\ d(x) &\geq 5 \\ \implies -d(x) + \left\lfloor \frac{d(x)}{2} \right\rfloor &\leq -5 + 2 = -3 \\ \implies -d(x) + 2 + \left\lfloor \frac{d(x)}{2} \right\rfloor &\leq -1 \\ \implies n - d(x) + 2 + \left\lfloor \frac{d(x)}{2} \right\rfloor &\leq n - 1 < n \\ \implies \gamma_{Rgs}(T) &< n. \end{aligned}$$

Stąd twierdzenie jest spełnione dla każdego drzewa, które ma przynajmniej jeden wierzchołek wspierający końcowy stopnia 5.

Dalej rozważmy przypadek, gdy T wszystkie wierzchołki wspierające końcowe mają co najwyżej stopień równy 4 oraz istnieją przynajmniej dwa wierzchołki wspierające końcowe x, x' , które mają stopień równy 4.

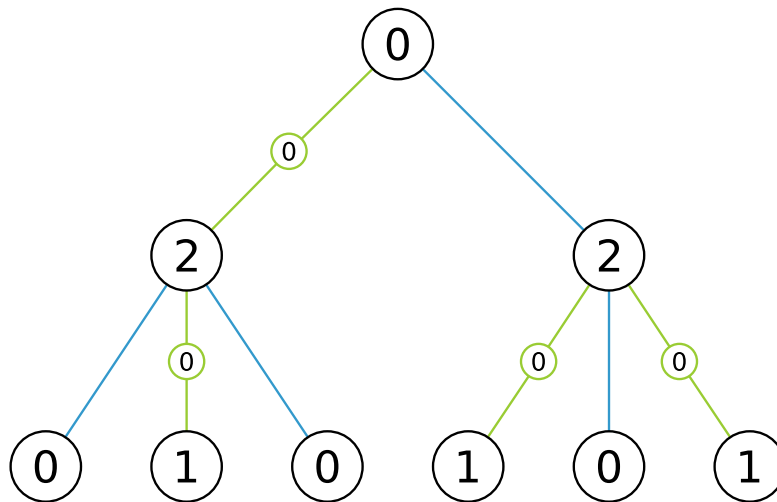
Niech $\{y_1, y_2, y_3\}$ będą liśćmi sąsiadującymi z wierzchołkiem x , a z niech będzie wierzchołkiem sąsiadującym z x , który nie jest liściem. Analogicznie $\{y'_1, y'_2, y'_3\}$ niech będą liśćmi sąsiadującymi z x' , a z' niech będzie wierzchołkiem, który nie jest liściem i sąsiaduje z x . Dodatkowo niech T_2 będzie podgrafem T indukowanym na zbiorze wierzchołków $V(T) \setminus \{x, y_1, y_2, y_3, x', y'_1, y'_2, y'_3\}$.

Rozważmy najpierw szczególny przypadek, w którym $z = z'$ oraz $n = 9$, który spełnia założenie o średnicy T większej lub równej 4. Wtedy T_2 jest grafem składającym się z jednego wierzchołka, więc nie spełnia założenia twierdzenia $n \geq 2$, co oznacza, że nie może zostać zastosowane wobec niego założenie indukcyjne.

W tym przypadku Dominator może przyjąć następującą strategię: w pierwszym ruchu oznacza krawędź zx , a w kolejnych ruchach oznacza dowolną dostępną krawędź ze zbioru $\{xy_1, xy_2, xy_3, x'y'_1, x'y'_2, x'y'_3\}$. W ten sposób zapewnia, że podzielone zostaną trzy krawędzie z tego zbioru oraz krawędź zx' .

Wówczas zostanie podzielona dokładnie jedna z krawędzi $\{zx, zx'\}$ oraz dokładnie trzy z krawędzi łączących x i x' z liśćmi. Najmniejsza funkcja dominowania rzymskiego dla grafu wynikowego T' przypisze wierzchołkom $\{x, x'\}$ wartość 2, liściom, których sąsiadujące krawędzie zostały podzielone wartość 1, a pozostałym wierzchołkom wartość 0. Stąd

$$\gamma_{Rgs}(T) = \gamma_R(T') = 2 \cdot 2 + 3 \cdot 1 = 7 < n.$$



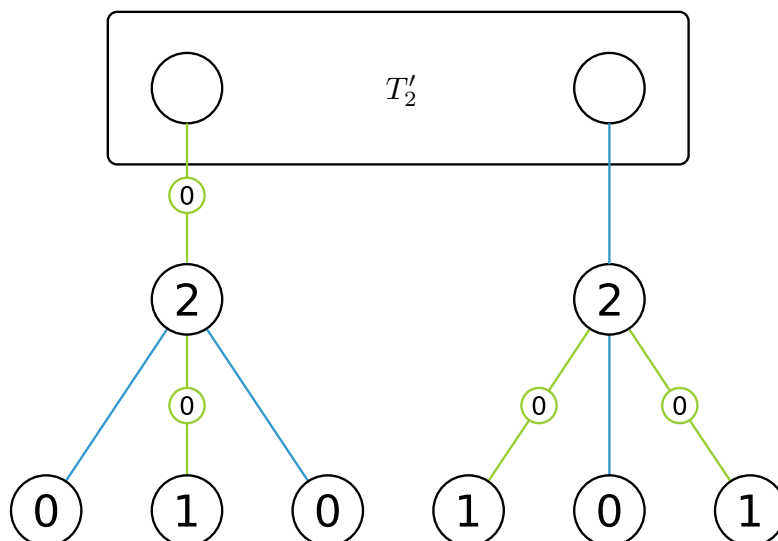
Rys. 4.13. Drzewo T' , dla przypadku $n(T) = 9$ i $z = z'$, w którym nie można użyć założenia indukcyjnego. Gra została rozegrana zgodnie z podaną strategią. Wewnątrz wierzchołków zostały wpisane wartości najmniejszej funkcji dominowania rzymskiego

Wyłączając powyższy przypadek, możemy użyć założenia indukcyjnego $\gamma_{Rgs}(T_2) \leq n - 8$.

Wtedy Dominator może przyjąć podobną do poprzedniej strategię, rozpoczynając od optymalnego ruchu w T_2 :

- jeżeli Psuj podzielił krawędź w T_2 , to Dominator wykonuje optymalny ruch dla T_2 ;
- jeżeli Psuj podzielił krawędź ze zbioru $\{zx, zx'\}$, to Dominator oznacza drugą z tych krawędzi;

- jeżeli Psuj podzielił krawędź ze zbioru $\{xy_1, xy_2, xy_3, x'y'_1, x'y'_2, x'y'_3\}$, to Dominator oznacza inną z tych krawędzi.



Rys. 4.14. Drzewo T' , w którym wszystkie wierzchołki wspierające końcowe mają stopień nie większy od 4 i przynajmniej dwa z nich mają stopień równy 4. W wierzchołkach zostały wpisane wartości rozszerzenia najmniejszej funkcji dominowania rzymskiego podgrafu T'_2 na cały graf wynikowy T'

W ten sposób Psuj podzielił trzy z sześciu krawędzi łączących wierzchołki $\{x, x'\}$ z liśćmi oraz jedną z krawędzi $\{zx, zx'\}$. Wtedy możliwe jest rozszerzenie najmniejszej funkcji dominowania rzymskiego dla grafu wynikowego T'_2 tak, aby obejmowała T' . Rozszerzona funkcja dominowania rzymskiego może przypisać wartość 2 wierzchołkom $\{x, x'\}$, wartość 1 liśćmi, których sąsiadujące krawędzie zostały podzielone oraz 0 pozostałym wierzchołkom w $V(T') \setminus V(T'_2)$. Tak rozszerzona funkcja nie musi być najmniejszą funkcją dominowania rzymskiego na T' , ponieważ wierzchołek z albo z' jest dominowany przez odpowiednio x albo x' , więc jego minimalna możliwa wartość funkcji dominowania rzymskiego może być mniejsza w T' niż w T'_2 . Stąd i z założenia indukcyjnego mamy

$$\gamma_{Rgs}(T) = \gamma_R(T') \leq \gamma_{Rgs}(T_2) + 2 \cdot 2 + 3 \cdot 1 = \gamma_{Rgs}(T_2) + 7 \leq n - 8 + 7 < n.$$

Ostatnim rozważanym przypadkiem, wyczerpującym założenia dowodu, jest drzewo T , w którym jest co najwyżej jeden wierzchołek wspierający końcowy o stopniu 4, a wszystkie pozostałe mają stopień co najwyżej 3. Weźmy wtedy wierzchołek wspierający końcowy x , zbiór sąsiadujących z nim liści $\{y_1, \dots, y_{d(x)-1}\}$, sąsiadujący z nim wierzchołek niebędący liściem z oraz podgraf T_1 indukowany na zbiorze wierzchołków $V(T) \setminus \{z, x, y_1, \dots, y_{d(x)-1}\}$.

Wtedy Dominator może przyjąć strategię, polegającą na rozpoczęciu od optymalnego ruchu na podgrafie T_1 , a następnie:

- jeżeli Psuj podzielił krawędź w T_1 , to Dominator wykonuje optymalny ruch w T_1 ;
- jeżeli Psuj podzielił krawędź ze zbioru $\{zx, xy_1, \dots, xy_{d(x)-1}\}$, to Dominator oznacza krawędź ze zbioru $\{xy_1, \dots, xy_{d(x)-1}\}$, a jeżeli to niemożliwe krawędź xz . Jeżeli ten ruch także nie jest możliwy do wykonania, Dominator oznacza dowolną krawędź w T_1 .

Przypadek 1. $d(x) = 2$

W tym przypadku z, x, y_1 jest ścieżką P_3 w grafie T . Jeżeli Dominator rozgrywa grę według

powyższej strategii, to co najwyżej jedna z jej krawędzi zostanie podzielona. Oznacza to, że w T' będzie znajdować się ścieżka P_4 z, s, x, y_1 lub z, x, s, y_1 , gdzie s jest wierzchołkiem powstałym w wyniku podziału krawędzi. Wówczas można rozszerzyć najmniejszą funkcję dominowania rzymskiego w podgrafie wynikowym T_1 przypisując trzeciemu wierzchołkowi tej ścieżki (x w pierwszym przypadku i s w drugim) wartość 2, a pozostałym wierzchołkom ze zbioru $\{s, x, y_1\}$ wartość 0. Wtedy zachodzi

$$\gamma_{Rgs}(T) = \gamma_R(T') \leq \gamma_{Rgs}(T_1) + 2 \leq n - 2 + 2 = n.$$

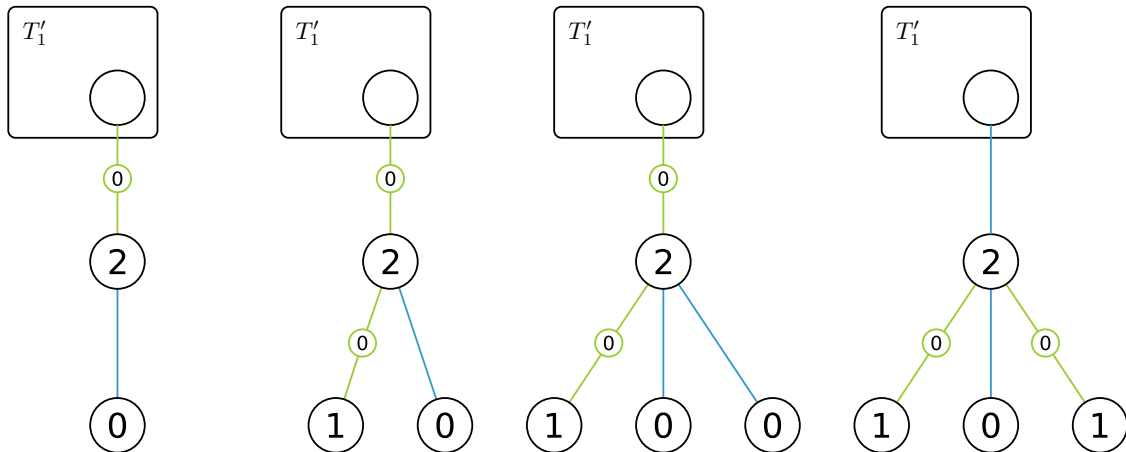
Przypadek 2. $d(x) > 2$

Przy grze Dominatora według przedstawionej strategii. Psuj będzie w stanie podzielić co najwyżej $\left\lceil \frac{d(x)-1}{2} \right\rceil = \left\lfloor \frac{d(x)}{2} \right\rfloor$ krawędzi pomiędzy wierzchołkiem x a liśćmi. Najmniejsza funkcja dominowania rzymskiego dla grafu wynikowego T'_1 może zostać rozszerzona tak, aby pokrywała cały graf wynikowy T' poprzez przypisanie wartości 2 wierzchołkowi x , liściom niepołączonym w T' z x wartość 1, a pozostałym wierzchołkom w $V(T') \setminus V(T'_1)$ wartość 0. Stąd i z założenia indukcyjnego mamy

$$\gamma_{Rgs}(T) = \gamma_R(T') \leq \gamma_{Rgs}(T_1) + 2 + \left\lfloor \frac{d(x)}{2} \right\rfloor \leq n - d(x) + 2 + \left\lfloor \frac{d(x)}{2} \right\rfloor.$$

Dla $d(x) > 2$ zachodzi $-d(x) + \left\lfloor \frac{d(x)}{2} \right\rfloor < -2$, więc

$$\gamma_{Rgs}(T) \leq n + 2 - d(x) + \left\lfloor \frac{d(x)}{2} \right\rfloor < n.$$



Rys. 4.15. Możliwe przypadki drzewa T' , gdy jest co najwyżej jeden wierzchołek wspierający końcowy stopnia 4, a wszystkie pozostałe wierzchołki mają stopień nie większy od 4. Od lewej: wynik gry dla $d(x) = 2$, wynik gry dla $d(x) = 3$ oraz dwa możliwe wyniki gry dla $d(x) = 4$. W wierzchołkach podane zostały wartości przypisane im przez rozszerzenie najmniejszej funkcji dominowania rzymskiego podgrafu T'_1 na graf wynikowy T'

□

5. APLIKACJA DO GRY W DOMINOWANIE RZYMSKIE

Jednym z celów pracy było stworzenie aplikacji, pozwalającej na grę w dominowanie rzymskie. W tym rozdziale opisano jej funkcje, technologie użyte do jej wytworzenia oraz zarys implementacji. Przedstawione zostały również automatyczne strategie do gry w dominowanie rzymskie oraz dominowanie rzymskie z podziałem krawędzi dla Dominatora i Psuja.

5.1. Funkcje aplikacji

5.1.1. Tryb graficzny

Podstawowym trybem działania aplikacji jest rozgrywka pomiędzy dwoma graczami w trybie graficznym. Gracze przyjmują role Dominatora i Psuja i, wykonując naprzemiennie ruchy na krawędziach, starają się uzyskać odpowiednio najmniejszą lub największą liczbę dominowania rzymskiego w grafie wynikowym. Możliwy jest wybór trybu gry:

- gra w dominowanie rzymskie – ruchy graczy powodują zorientowanie krawędzi w jedną ze stron;
- gra w dominowanie rzymskie z podziałem krawędzi – ruch Dominatora powoduje oznaczenie krawędzi, zaś ruch Psuja powoduje podział krawędzi przez dodanie wierzchołka.

Aplikacja umożliwia również zastąpienie jednego lub obu graczy strategią automatyczną. Wówczas ruchy zastąpionego gracza są wybierane na podstawie opracowanego algorytmu. Opis dostępnych strategii automatycznych znajduje się w rozdziale 5.6.

Rozgrywka kończy się, gdy na każdej z dostępnych w początkowym grafie krawędzi został wykonany ruch. Po wykonaniu ostatniego ruchu w grze aplikacja znajduje najmniejszą funkcję dominowania rzymskiego dla grafu wynikowego i wyświetla jej wartości dla poszczególnych wierzchołków oraz sumę jej wartości, będącą jednocześnie wynikiem gry. Algorytm wyznaczania najmniejszej funkcji dominowania rzymskiego grafu wynikowego¹ jest opisany w rozdziale 5.5.1.

Ze względu na wykładniczą złożoność obliczeniową, dla grafów powyżej pewnej liczby wierzchołków znajdowana najmniejsza funkcja dominowania rzymskiego. Liczba ta jest częścią konfiguracji aplikacji opisanej w rozdziale 5.5.5.

Domyślnie graf początkowy jest generowany losowo. Liczba jego wierzchołków oraz krawędzi może być konfigurowana przez parametry podawane przy uruchamianiu aplikacji. Alternatywnie możliwe jest wczytanie grafu z pliku w formacie *GraphML* przygotowanego z użyciem innego narzędzia.

5.1.2. Tryb tekstowy

Z użyciem odpowiedniego parametru aplikacja może zostać uruchomiona w trybie bez graficznej reprezentacji gry, co może być przydatne do porównywania efektywności strategii automatycznych. Po zakończeniu gry wyświetlana jest liczba dominowania rzymskiego grafu wynikowego. Możliwe jest również rozegranie określonej parametrem liczby gier, z których wyników na końcu wyciągana jest średnia. W przypadku losowo generowanego grafu początkowego każda z gier zostanie przeprowadzona na innym grafie, co pozwala na lepszą ocenę efektywności strategii.

¹skierowanego w przypadku gry w dominowanie rzymskie oraz nieskierowanego w przypadku gry w dominowanie rzymskie z podziałem krawędzi

Ten tryb może również zostać zastosowany w celu ustalenia liczby gry w dominowanie rzymskie lub liczby gry w dominowanie rzymskie z podziałem krawędzi dla określonego, wczytanego z pliku grafu przy zastosowaniu dla obu graczy optymalnej strategii MinMax.

5.1.3. Tryb testowy

Alternatywnie aplikacja może zostać uruchomiona w trybie testowym, pozwalającym na skonfigurowanie testów strategii automatycznych. Możliwe jest określenie wielu parametrów takiego testu, takich jak:

- liczba wierzchołków i krawędzi w grafach testowych,
- strategię automatyczne dla Dominatora wraz z ich parametrami,
- strategię automatyczne dla Psuja wraz z ich parametrami,
- tryb gry (standardowy albo z podziałem krawędzi),
- liczba iteracji testu,
- liczba równoległych procesów.

Testy przeprowadzane są na zasadzie *każdy z każdym*, tzn. każda strategia automatyczna dla Dominatora rozgrywa partię z każdą strategią automatyczną dla Psuja. W jednej iteracji testu każda z par gra na tym samym grafie początkowym. Wszystkie grafy wykorzystywane w trakcie testu są zapisywane to plików w formacie *GraphML* z numerami odpowiadającymi numeracji iteracji, co pozwala na późniejsze odtworzenie dowolnej z gier.

Wyniki testu zapisywane są w pliku *CSV*. Każdy wiersz odpowiada jednej rozgrywce i zawiera informacje o:

- numerze iteracji,
- nazwie strategii automatycznej dla Dominatora,
- nazwie strategii automatycznej dla Psuja,
- funkcji dominowania rzymskiego grafu wynikowego,
- sumie tej funkcji,
- czasie trwania gry.

Konfiguracja testu jest zapisywana w formacie słownikowym języka Python razem z wynikami, co pozwala na późniejsze łatwe odtworzenie testu w tej samej konfiguracji. Dodatkowo w formie tekstowej przedstawiane jest podsumowanie wyników testu, zawierające średni wynik gry dla każdej z par, każdej strategii Dominatora oraz każdej strategii Psuja.

5.2. Interfejs użytkownika

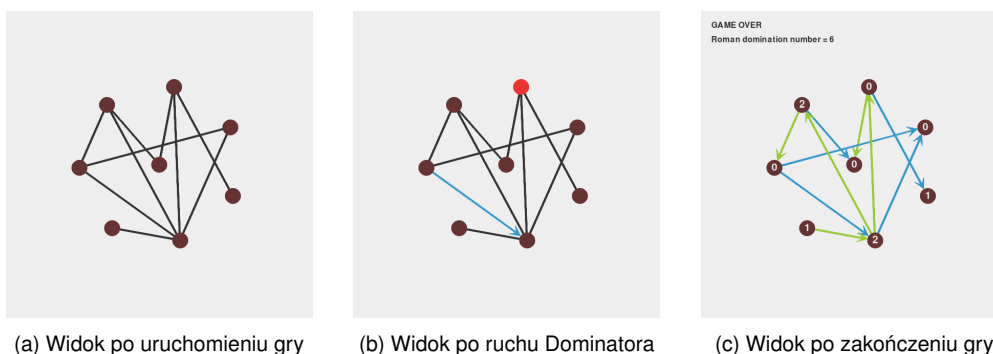
5.2.1. Tryb graficzny

Tryb graficzny gry jest uruchamiany z użyciem skryptu `rdg.py`. Opcje gry są określane przy uruchamianiu gry jako parametry skryptu. Domyślnie uruchamiana jest gra w dominowanie rzymskie na losowo wygenerowanym grafie o 8 wierzchołkach i 10 krawędziach pomiędzy dwoma graczami. Wszystkie dostępne parametry zostały przedstawione w tabeli 5.1.

Po uruchomieniu aplikacji w trybie graficznym użytkownikowi prezentowane jest okno gry, wyglądające podobnie do przedstawionego na rysunku 5.1a. Jeżeli przynajmniej jeden z graczy jest kontrolowany przez użytkownika, może on wykonać ruch poprzez wybranie kolejno dwóch połączonych wierzchołków. Pierwszy z wybranych wierzchołków jest podświetlany tak, jak zostało to przedstawione na rysunku 5.1b.

Parametr	Opis
-h, --help	Wyświetlenie pomocy dotyczącej parametrów
-s	Gra z podziałem krawędzi
--headless	Uruchomienie trybu tekstowego
-d STRATEGIA	Strategia Dominatora
--dominator STRATEGIA	
-a STRATEGIA	Strategia Psuja
--avoider STRATEGIA	
-i LICZBA_ITERACJI	Liczba gier do rozegrania
--iterations LICZBA_ITERACJI	
-v LICZBA_WIERZCHOŁKOW	Liczba wierzchołków generowanego grafu początkowego
--vertices LICZBA_WIERZCHOŁKOW	
-e LICZBA_KRAWEDZI	Liczba krawędzi generowanego grafu początkowego
--edges LICZBA_KRAWEDZI	
--graphml PLIK	Plik zawierający graf początkowy
--log {DEBUG, INFO, WARNING, ERROR, CRITICAL}	Określenie poziomu wyświetlanych komunikatów

Tabela 5.1. Parametry skryptu `rdg.py` pozwalające na konfigurację gry



Rys. 5.1. Okno aplikacji uruchomionej w trybie graficznym z domyślnymi parametrami. Na rys. 5.1b jeden z wierzchołków został zaznaczony przez użytkownika grającego jako Psuj. Wybranie kolejnego, połączony z nim wierzchołka spowoduje wykonanie ruchu. Rys. 5.1c przedstawia widok po zakończeniu gry

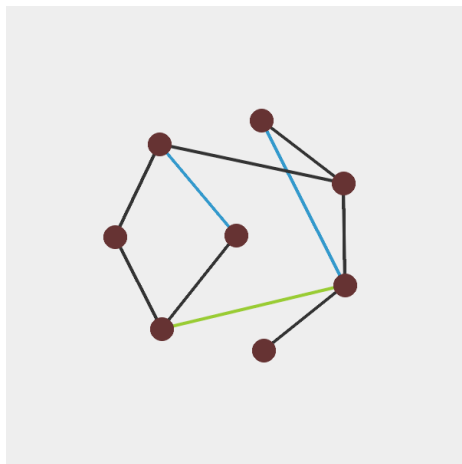
W przypadku standardowej gry w dominowanie rzymskie kolejność wybieranych wierzchołków określa kierunek orientacji krawędzi. Ruchy Dominatora oznaczane są kolorem **niebieskim**, natomiast ruchy Psuja kolorem **zielonym**.

Po zakończeniu gry wyświetlony zostaje stosowny komunikat oraz liczba dominowania rzymskiego wynikowego grafu skierowanego. Wierzchołki są oznaczone odpowiadającymi im wartościami najmniejszej funkcji dominowania rzymskiego dla grafu wynikowego (rys. 5.1c).

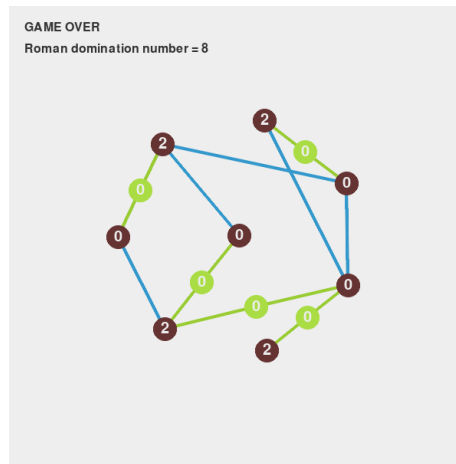
W przypadku gry z podziałem krawędzi ruchy są przedstawiane przez kolorowanie krawędzi, bez oznaczania ich orientacji. Wierzchołki dodawane w wyniku ruchów Psuja są widoczne dopiero w momencie zakończenia gry. Są one oznaczone kolorem oznaczającym również ruchy Psuja w celu odróżnienia ich od wierzchołków grafu początkowego (rys. 5.2b). Wtedy również w wierzchołkach przedstawione są wartości najmniejszej funkcji dominowania rzymskiego w grafie wynikowym i komunikaty identyczne do tych w podstawowym wariantcie gry.

5.2.2. Tryb tekstowy

Tryb tekstowy aplikacji jest również uruchamiany z użyciem skryptu `rdg.py` z parametrem `-headless`. Wymagane jest również określenie strategii automatycznych dla obu graczy, ponieważ nie ma w tym trybie możliwości wprowadzania ruchów przez użytkownika. Rozgrywana jest wówczas jedna lub więcej gier (w zależności od wartości parametru `-i`), a następnie wypisywana jest średnia liczba dominowania rzymskiego dla grafu wynikowego.



(a) Widok w trakcie gry



(b) Widok po zakończeniu gry

Rys. 5.2. Okno aplikacji uruchomionej w trybie graficznym w grze z podziałem krawędzi. Rys. 5.2b przedstawia widok po zakończeniu gry wraz z wierzchołkami dodanymi w wyniku ruchów Psuja

Przykładowe wywołanie trybu tekstowego może wyglądać następująco:

```
./rdg.py --headless -i 2 -d MinMaxOptD -a MinMaxOptA -v 6 -e 8.
```

Przy takim wywołaniu aplikacja rozegra dwie gry, w których Dominator będzie grał strategią MinMaxOptD, a Psuj strategią MinMaxOptA. Graf początkowy o 6 wierzchołkach i 8 krawędziach zostanie wygenerowany losowo. Dla tak wywołanej aplikacji wyjście będzie wyglądać podobnie do przedstawionego poniżej.

```
1 == Iteration 0 ==
2 MinMaxOptD(D) move 1 -> 3 (worst possible score=4, eval depth=-1)
3 MinMaxOptA(A) move 0 -> 3 (worst possible score=4, eval depth=-1)
4 MinMaxOptD(D) move 1 -> 4 (worst possible score=4, eval depth=-1)
5 MinMaxOptA(A) move 2 -> 3 (worst possible score=4, eval depth=-1)
6 MinMaxOptD(D) move 0 -> 4 (worst possible score=4, eval depth=-1)
7 MinMaxOptA(A) move 1 -> 5 (worst possible score=4, eval depth=-1)
8 MinMaxOptD(D) move 3 -> 4 (worst possible score=4, eval depth=-1)
9 MinMaxOptA(A) move 2 -> 5 (worst possible score=4, eval depth=-1)
10 GAME OVER
11 Roman domination number = 4
12 Dominator's moves: [(1, 3), (1, 4), (0, 4), (3, 4)]
13 Avoider's moves: [(0, 3), (2, 3), (1, 5), (2, 5)]
14 == Iteration 1 ==
15 MinMaxOptD(D) move 3 -> 4 (worst possible score=5, eval depth=-1)
16 MinMaxOptA(A) move 0 -> 3 (worst possible score=5, eval depth=-1)
17 MinMaxOptD(D) move 1 -> 3 (worst possible score=5, eval depth=-1)
18 MinMaxOptA(A) move 2 -> 1 (worst possible score=5, eval depth=-1)
19 MinMaxOptD(D) move 2 -> 5 (worst possible score=5, eval depth=-1)
20 MinMaxOptA(A) move 0 -> 2 (worst possible score=5, eval depth=-1)
21 MinMaxOptD(D) move 1 -> 5 (worst possible score=5, eval depth=-1)
22 MinMaxOptA(A) move 4 -> 1 (worst possible score=5, eval depth=-1)
23 GAME OVER
24 Roman domination number = 5
25 Dominator's moves: [(3, 4), (1, 3), (2, 5), (1, 5)]
26 Avoider's moves: [(0, 3), (2, 1), (0, 2), (4, 1)]
27 Average Roman Domination Number: 4.5
28
```


5.2.3. Tryb testowy

Testy strategii automatycznych są uruchamiane z użyciem skryptu `benchmark.py`. Konfiguracja testów jest specyfikowana za pomocą słownika określającego poszczególne parametry. Przykładowa konfiguracja została przedstawiona poniżej.

```
1 CONFIG = {
2     "vertices": 8,
3     "edges": 13,
4     "iterations": 10,
5     "num_processes": 7,
6     "subdivision": True,
7     "out_file": "test_1",
8     "p1_strategies": {
9         "Reference": ("Reference", ()),
10        "MinMaxOptD": ("MinMaxOptD", ()),
11        "VW(-1,2,2)": ("VWScoreD", ((-1, 2, 2),)),
12    },
13    "p2_strategies": {
14        "Reference": ("Reference", ()),
15        "MinMaxOptA": ("MinMaxOptA", ()),
16        "VW(0,-1,0)": ("VWScoreA", ((0, -1, 0),)),
17    },
18 }
19
```

Znaczenie poszczególnych parametrów zostało opisane w tabeli 5.2. Każda ze strategii stanowi wpis w słowniku odpowiadającym Dominatorowi lub Psujowi. Jego kluczem jest nazwa strategii używana podczas testu, natomiast jego wartość jest parą złożoną z nazwy strategii oraz argumentów podawanych przy jej inicjalizacji. W ten sposób w ramach testu można testować wielokrotnie tę samą strategię z różnymi parametrami.

Parametr	Opis
"vertices"	Liczba wierzchołków w grafach testowych
"edges"	Liczba krawędzi w grafach testowych
"iterations"	Liczba iteracji zestawienia „każdy z każdym”
"num_processes"	Liczba równoległych procesów
"subdivision"	Flaga decydująca, czy rozgrywana jest gra standardowa, czy z podziałem krawędzi
"out_file"	Nazwa dla plików wyjściowych: pliku z konfiguracją oraz wynikowego pliku <i>CSV</i>
"p1_strategies"	Słownik strategii Dominatora
"p2_strategies"	Słownik strategii Psuja

Tabela 5.2. Parametry konfiguracji testów strategii automatycznych

Dodatkowo przy uruchamianiu aplikacji istnieje możliwość nadpisania wszystkich parametrów poza listami strategii z poziomu wiersza poleceń. Nadpisywanie liczby wierzchołków i krawędzi, liczby iteracji oraz rodzaju gry jest możliwe przez podanie argumentów identycznych z tymi dla skryptu `rdg.py` (tab. 5.1), natomiast liczba procesów konfigurowana jest z użyciem parametru `-p`, `--parallel LICZBA_PROCESOW`, a nazwa pliku wyjściowego z użyciem `-o NAZWA_PLIKU`.

Skrypt `benchmark.py` umożliwia również uruchomienie z argumentem `--dry-run`, który powoduje wyświetlenie konfiguracji testu i liczby gier do rozegrania bez jego uruchamiania. Jest to przydatne w przypadku, gdy chcemy sprawdzić, czy użyta została poprawna konfiguracja oraz upewnić się, że nie ma w niej błędów przed uruchomieniem testów.

5.3. Instalacja zależności i uruchomienie aplikacji

Aplikacja wykorzystuje biblioteki spoza biblioteki standardowej Pythona, w związku z czym przed jej uruchomieniem konieczna jest instalacja zależności. Aby uniknąć potrzeby ich globalnej instalacji, do kodu programu została załączona konfiguracja narzędzia *pipenv*, pozwalającego na łatwe stworzenie wirtualnego środowiska, zawierającego wymagane biblioteki.

Aby zainstalować narzędzie *pipenv*, należy wykonać polecenie:

```
python -m pip install pipenv
```

Następnie w celu utworzenia wirtualnego środowiska oraz zainstalowania wymaganych bibliotek wystarczy wykonać polecenie

```
pipenv install
```

w katalogu zawierającym pliki aplikacji.

Po wykonaniu powyższych kroków aplikacja jest gotowa do użycia. Aby uruchomić skrypt `rdg.py` lub `benchmark.py` wewnątrz wirtualnego środowiska, należy poprzedzić jego wywołanie poleceniem `pipenv run`.²

5.4. Użyte technologie

Aplikacja jest napisana w języku Python w wersji 3 i wykorzystuje funkcjonalność biblioteki standardowej dodaną w wersji 3.10, więc przynajmniej ta wersja jest wymagana do jej uruchomienia. Dodatkowo jak wspomniano w rozdziale 5.3, wykorzystywane są dodatkowe biblioteki spoza biblioteki standardowej języka, których użycie opisano poniżej.

5.4.1. *igraph*

igraph jest otwartoźródłową biblioteką napisaną w języku C, która implementuje funkcjonalność związaną z analizą grafów. [9] Jej autorzy tworzą również interfejs, umożliwiający łatwe użycie jej w programach napisanych w języku Python, o nazwie *python-igraph*.

Aplikacja wykorzystuje klasę `igraph.Graph` z biblioteki jako podstawę do reprezentacji grafów gry w dominowanie rzymskie z i bez podziału krawędzi. Umożliwia ona wykonywanie podstawowych działań na grafie, takich jak dodawanie i usuwanie wierzchołków i krawędzi oraz przypisywanie im atrybutów.

Wykorzystywany jest również dostarczony przez bibliotekę *igraph* algorytm, rozmieszczający wierzchołki na płaszczyźnie. Jego wyjście jest skalowane tak, aby dopasować współrzędne wierzchołków do rozmiarów ekranu gry, co pozwala na wyświetlenie ich na ekranie w położeniu zgodnym z wyjściem algorytmu.

Dodatkowo użyta została metoda pozwalająca na wczytywanie grafu z pliku w formacie *GraphML*, dzięki czemu można prowadzić rozgrywkę na przygotowanym wcześniej grafie.

W aplikacji została zaimplementowana możliwość wizualizacji drzewa gry rozpatrywanego przez strategie automatyczne, bazujące na jego przeglądaniu. Poza wspomnianą wyżej klasą bazową grafu wykorzystana została funkcja `igraph.drawing.plot`, która pozwala na zapisanie drzewa gry w formie obrazu w formacie SVG.

²Na przykład `pipenv run ./rdg.py --headless -i 2 -d MinMaxOptD -a MinMaxOptA -v 6 -e 8`.

5.4.2. *pygame*

Biblioteka *pygame* [15] jest łatwym w użyciu, a jednocześnie elastycznym narzędziem do implementacji interfejsów użytkownika, m. in. gier. W ramach tej pracy stanowi podstawowy element, z użyciem którego stworzony został tryb graficzny gry. Jest używana do tworzenia okna gry, rysowania jej stanu w postaci grafu oraz wypisywania komunikatów. Dodatkowo odpowiada za obsługę zdarzeń generowanych przez użytkownika, więc umożliwia graczowi wybieranie ruchów w trakcie gry oraz zakończenie jej w dowolnym momencie.

Do rysowania reprezentacji gry wykorzystano eksperymentalny moduł `pygame.gfxdraw`. W odróżnieniu od standardowego modułu do rysowania kształtów `pygame.draw` udostępnia on funkcje rysowania dowolnych wielokątów z wykorzystaniem anty-aliasingu krawędzi, co znacząco poprawia wygląd narysowanych elementów.

5.5. **Projekt aplikacji**

Aplikacja podzielona jest na moduły, odpowiadające różnym aspektom jej działania, które wykorzystują zewnętrzne zależności do realizacji swoich funkcji. Dwa uruchamialne skrypty, `rdg.py` oraz `benchmark.py`, używają zaimplementowanych modułów, umożliwiając użytkownikowi zagranie w grę w dominowanie rzymskie, standardową lub z podziałem krawędzi, oraz testowanie strategii automatycznych grających w te gry. Poniżej opisane zostało działanie i współdziałanie komponentów aplikacji. Zależności pomiędzy nimi zostały również przedstawione w formie graficznej na rysunku 5.3.

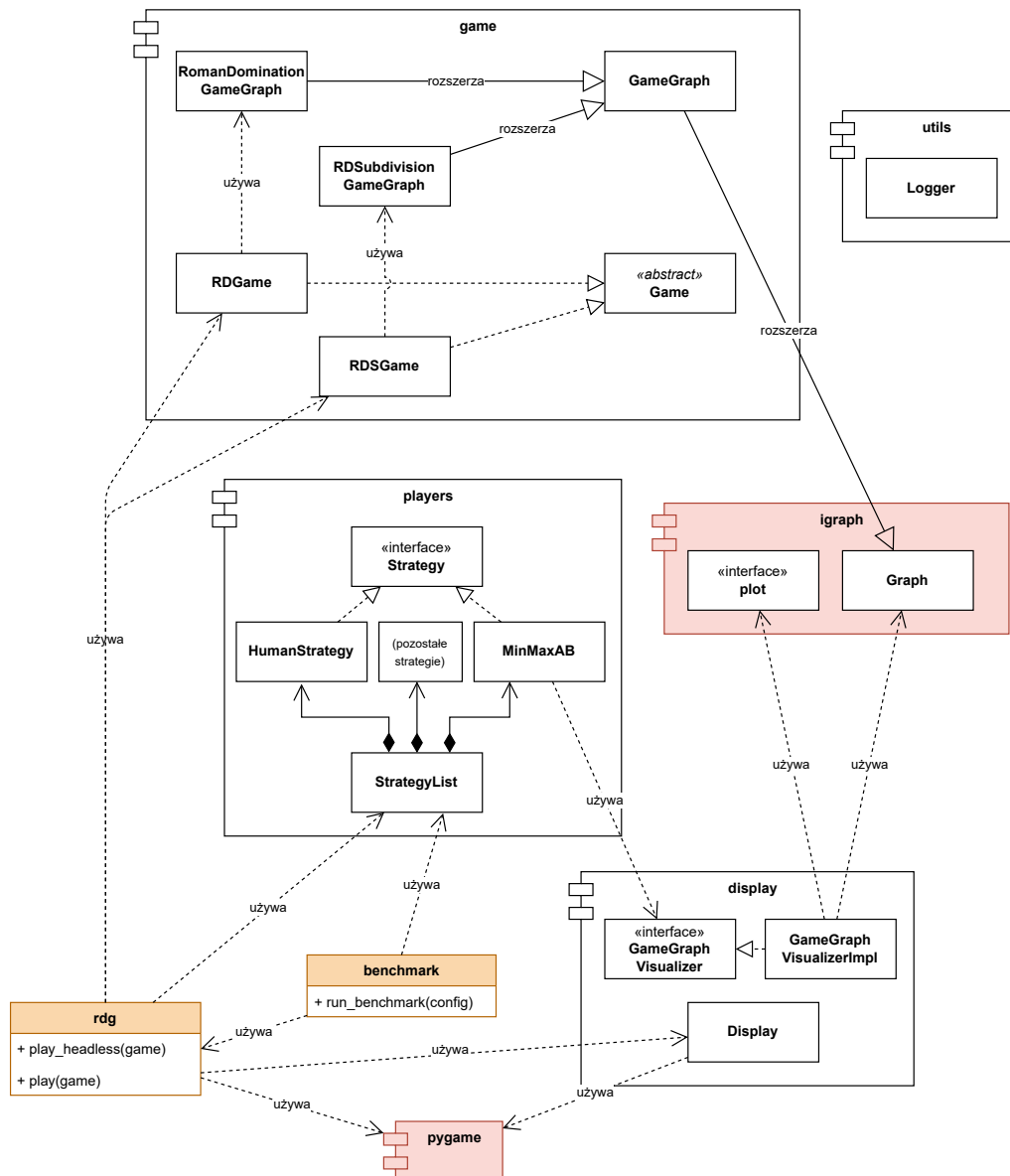
5.5.1. *Moduł game*

Moduł `game` implementuje mechanikę gry w dominowanie rzymskie oraz jej wariant z podziałem krawędzi. Składa się on z dwóch klas bazowych: `GameGraph` oraz `Game`. Dla każdej z nich istnieją po dwie klasy pochodne, odpowiadające każdemu z wariantów gry.

`GameGraph` i jej implementacje są odpowiedzialne za reprezentację grafu gry oraz wykonywanie na nim ruchów. Klasa bazowa dziedziczy po klasie `igraph.Graph`, która jest reprezentacją grafu dostarczaną przez zewnętrzną bibliotekę. Zawiera również metody pomocnicze, pozwalające na uzyskanie wszystkich ruchów wykonanych przez obu graczy, listy krawędzi, na których jeszcze nie został wykonany żaden ruch, grafu początkowego, oraz informacji, czy gra została ukończona. Dodatkowo klasa bazowa implementuje algorytm wyznaczania najmniejszej funkcji dominowania rzymskiego dla grafu wynikowego oraz, co z tego bezpośrednio wynika, jego liczby dominowania rzymskiego.

Wspomniany algorytm polega na przeszukaniu wszystkich funkcji $V(G') \rightarrow \{0, 1, 2\}$ dla grafu wynikowego G' . Dla każdej z nich sprawdzane jest, czy jej sumaryczna wartość dla wszystkich wierzchołków jest najmniejszą dotychczas oraz czy jest poprawną funkcją dominowania rzymskiego w grafie G' . Jeżeli pierwszy z tych warunków, którego sprawdzenie jest szybsze, nie jest spełniony, to nie jest sprawdzany drugi z nich, co jest drobną optymalizacją względem bardziej naiwnego podejścia.

Zmienna reprezentująca najmniejszą dotychczas sumę wartości funkcji dominowania rzymskiego jest inicjalizowana wartością $|V(G')|$, ponieważ funkcja, przypisująca wszystkim wierzchołkom wartość 1, jest poprawną funkcją dominowania rzymskiego. Ogranicza to ilość funkcji, dla których sprawdzana jest poprawność. Funkcja jest traktowana jako liczba w systemie trójkowym



Rys. 5.3. Podział aplikacji na moduły i klasy. Na pomarańczowo zostały oznaczone uruchamialne skrypty, natomiast na czerwono zewnętrzne zależności wykorzystywane w aplikacji

przedstawiona jako sekwencja cyfr o długości równej $|V(G')|$. Kolejna funkcja jest znajdowana przez dodanie wartości 1 do tej liczby.

Przyjmijmy, że $P(v)$ w przypadku standardowej gry w dominowanie rzymskie oznacza zbiór poprzedników³ wierzchołka v , a w przypadku gry w dominowanie rzymskie zbiór wierzchołków sąsiadujących z v . Sprawdzenie poprawności funkcji dominowania rzymskiego dla grafu polega na sprawdzeniu dla każdego z wierzchołków v grafu G' , czy istnieje wierzchołek $u \in P(v)$, któremu została przypisana wartość $f(u) = 2$. Jeżeli jest chociaż jeden wierzchołek v , dla którego ten warunek nie jest spełniony, to oznacza, że funkcja jest niepoprawna, natomiast spełnienie warunku przez wszystkie wierzchołki o przypisanej wartości 0 implikuje poprawność funkcji dominowania rzymskiego.

Klasa `Game` i jej implementacje dla poszczególnych wariantów gry udostępniają metody do wykorzystania przez skrypty w celu zmiany stanu gry oraz pozyskanie informacji na jego temat.

³Wierzchołek $u \in V(G)$ jest poprzednikiem wierzchołka $v \in V(G)$, gdy w $A(G)$ (zbiorze łuków grafu G) istnieje łuk uv .

Stanowią warstwę pośrednią pomiędzy główną pętlą gry a klasą pochodną `GameGraph`, stanowiącą reprezentację stanu gry.

5.5.2. Moduł `display`

Graficzny interfejs użytkownika został zaimplementowany w module `display`. Jego podstawowym komponentem jest klasa `Display` odpowiedzialna za tworzenie okna gry i rysowanie jego zawartości, w tym reprezentacji grafu gry oraz komunikatów dla użytkownika. Na publiczny interfejs obiektu klasy `Display` składają się:

- konstruktor inicjalizujący bibliotekę `pygame`, tworzący okno gry oraz generujący współrzędne dla wierzchołków grafu;
- metoda `draw`, która aktualizuje zawartość okna gry, w tym
 - rysuje krawędzie lub łuki (zależnie od wariantu gry), odpowiadające wykonanym dotychczas ruchom,
 - rysuje krawędzie grafu początkowego, na których nie został jeszcze wykonany ruch,
 - rysuje wierzchołki grafu, wyróżniając zaznaczony przez gracza wierzchołek,
 - opcjonalnie wypisuje komunikaty dodane za pomocą metody `add_text`;
- metoda `add_text`, która dodaje wiadomość do wypisania w oknie gry przy następnej jego aktualizacji;
- metoda `find_clicked_vertex`, przyjmująca współrzędne kliknięcia myszą i zwracająca wierzchołek, na którym zostało ono wykonane, lub `None`, jeżeli kliknięcie nie nastąpiło na żadnym wierzchołku;
- metoda `extend_layout`, rozszerzająca listę współrzędnych wierzchołków o nowe wierzchołki dodane w wyniku podziału krawędzi w grze w dominowanie rzymskie z podziałem krawędzi. Wierzchołki dodawane są pośrodku podzielonej krawędzi i oznaczone kolorem odpowiadającym ruchowi Psuja, aby wyróżnić je od wierzchołków grafu początkowego.

Wszystkie rysowane na ekranie gry obiekty są stworzone z podstawowych kształtów dostarczanych przez bibliotekę `pygame` oraz wielokątów. Współrzędne wierzchołków są generowane z użyciem metody `layout()` grafu gry dziedziczonej z klasy bazowej `igraph.Graph`. Następnie współrzędne te są skalowane do wymiaru okna gry.

Klasa `Display` jest wykorzystywana jedynie przez skrypt `rdg.py`, ponieważ tylko z jego użyciem możliwe jest uruchomienie gry w trybie graficznym.

Dodatkowo w module `display` znajduje się interfejs `GameGraphVisualizer` oraz jego implementacja `GameGraphVisualizerImpl`. Pozwala ona na stworzenie wizualizacji drzewa przeszukiwania stanów gry dla strategii, które korzystają z tej techniki (strategie z rodziny `MinMax`). W ramach jednego ruchu strategii jest tworzony graf, w którym wierzchołki odpowiadają stanom gry, a krawędzie ruchom graczy. Graf ten jest reprezentowany jako obiekt klasy `igraph.Graph`, natomiast po zakończeniu przeglądu drzewa stanów jest zapisywany jako grafika w formacie `SVG` z użyciem metody `igraph.drawing.plot`.

Z uwagi na potencjalny duży rozmiar drzewa i, co za tym idzie, złożoną obliczeniowo konwersję do formatu graficznego i duży rozmiar wynikowego pliku, możliwe jest wyłączenie wizualizacji poprzez zmianę flagi w module `config`. Wówczas zamiast pełnej implementacji używana jest klasa `GameGraphVisualizer` zawierająca puste, нефункционалне metody.

5.5.3. Moduł *players*

Moduł *players* zawiera klasy implementujące strategie automatyczne oraz implementację „strategii” gracza sterowanego przez użytkownika. Wszystkie ze strategii dziedziczą po klasie bazowej *Strategy*.

```
1 class Strategy:
2     def __init__(self):
3         pass
4
5     def __call__(self, graph: GameGraph):
6         # Zwraca ruch wybrany przez strategię
7         raise NotImplementedError
8
9     def __repr__(self):
10        return self.__class__.__name__
11
12    def add_selected_vertex(self, vertex: int):
13        # Implementowana tylko przez HumanStrategy
14        pass
15
```

Metoda `add_selected_vertex` jest implementowana tylko przez strategię gracza sterowanego przez użytkownika *HumanStrategy*, ponieważ wybiera on wierzchołki pojedynczo i strategia jest w stanie wykonać ruch dopiero, gdy zaznaczone zostaną dwa wierzchołki.

Warto zauważyć, że w tym module nie ma żadnego mechanizmu powstrzymującego strategię przed zwróceniem nieprawidłowego ruchu. Weryfikacja poprawności ruchu leży po stronie wywołującego strategię. Wywołanie strategii może również zwrócić wartość `None`, co oznacza brak ruchu. Jest to istotne w przypadku klasy *HumanStrategy*, która musi przed wykonaniem ruchu zostać poinformowana o wierzchołkach zaznaczonych przez użytkownika.

5.5.4. Moduł *utils*

utils jest modulem pomocniczym współdzielonym przez wszystkie pozostałe moduły. Zawiera jeden obiekt singleton, odpowiadający za logowanie, który, wraz z zaimportowaniem modułu, inicjalizuje wewnętrzny obiekt klasy `logging.Logger`, pochodzący z biblioteki standardowej języka Python i pozwalający na logowanie komunikatów na różnych poziomach. Poziom komunikatów wypisywanych na standardowym wyjściu może zostać określony przez parametr `--log` skryptów `rdg.py` oraz `benchmark.py`.

5.5.5. Moduł *config*

Moduł *config* pozwala na konfigurację działania oraz wyglądu programu przez użytkownika. Składa się z jednej statycznej klasy *Config*, której pola pozwalają zmieniać takie aspekty aplikacji jak:

- tytuł i rozmiar okna gry,
- rozmiar tekstu komunikatów w oknie gry,
- kolory tła, wierzchołków, krawędzi oraz komunikatów,
- katalogi wyjściowe dla testów oraz wizualizacji drzew stanów gry,
- włączenie/wyłączenie wizualizacji drzew stanów gry,
- maksymalna liczba wierzchołków, dla której szukana będzie najmniejsza funkcja dominowania rzymskiego.

5.5.6. Skrypt *rdg.py*

Skrypt *rdg.py* pozwala na uruchomienie pojedynczej gry lub serii gier pomiędzy dwoma wybranymi strategiami lub graczami. Składa się z dwóch funkcji, implementujących podstawową pętlę gry: *play* oraz *play_headless*.

Pierwsza z nich jest wykorzystywana w przypadku uruchomienia gry w trybie graficznym. Na bazie podanego grafu początkowego tworzona jest nowa gra (obiekt klasy dziedziczącej po *game.Game*) oraz obiekt kontrolujący wyświetlanie interfejsu gry (obiekt klasy *display.Display*).

W ramach pętli gry wykonywane są następujące czynności:

1. ustalana jest strategia gracza, który ma wykonać ruch;
2. sprawdzane są zdarzenia takie jak zamknięcie okna lub kliknięcie myszą. W przypadku kliknięcia w oknie gry sprawdzane jest, czy został zaznaczony wierzchołek. Jeżeli tak, to numer wierzchołka jest przekazywany aktualnej strategii;
3. sprawdzane jest, czy gra została zakończona. Jeżeli tak, to wyświetlane są wartości najmniejszej funkcji dominowania rzymskiego oraz jej suma;
4. w przeciwnym wypadku aktualna strategia jest wywoływana, a zwrócony ruch jest wykonywany;
5. następuje aktualizacja wyświetlanego stanu gry.

Funkcja *play_headless* jest wykorzystywana w przypadku uruchomienia gry w trybie tekstowym oraz do rozgrywania gier w ramach testów uruchamianych przez skrypt *benchmark.py*. Jej działanie jest analogiczne do funkcji *play* za wyjątkiem wyświetlania stanu gry i wczytywania zdarzeń. Jeżeli używana jest ta funkcja, to aplikacja nie wykorzystuje biblioteki *pygame*.

5.5.7. Skrypt *benchmark.py*

Tak, jak opisano w rozdziale 5.2.3, skrypt *benchmark.py* pozwala na uruchomienie skonfigurowanej uprzednio serii testów. *run_benchmark* jest funkcją odpowiadającą za uruchomienie testów na podstawie konfiguracji. W tym celu tworzona jest lista grafów, na których będą rozgrywane gry w kolejnych iteracjach, oraz lista krotek opisujących parametry każdej z gier, która ma zostać rozegrana.

Następnie z takim opisem gry wywoływana jest funkcja *run_instantiate*, która tworzy instancje odpowiednich strategii automatycznych i wywołuje funkcję *play_headless* ze skryptu *rdg.py*, mierząc jednocześnie czas jej wykonania. Po zakończeniu rozgrywki tworzony jest słownik, opisujący grę, zawierający: numer iteracji, który odpowiada indeksowi grafu początkowego w liście grafów, nazwy strategii, najmniejszą funkcję dominowania rzymskiego grafu wynikowego, jej sumę oraz czas rozgrywki.

Sposób wywołania *run_instantiate* zależy od liczby równoległych procesów określonej w konfiguracji testów. Jeżeli jest ona równa 1, to gry rozgrywane są sekwencyjnie, natomiast jeśli jest większa, wykorzystywana jest pula procesów *multiprocessing.Pool* z biblioteki standardowej, na której wykonywana jest metoda *map(run_instantiate, args)*, gdzie *args* to lista krotek z konfiguracjami poszczególnych gier. To powoduje uruchomienie zadanej liczby procesów, które równolegle rozgrywają gry z listy, co przyspiesza przeprowadzanie testów, jeżeli liczba procesów nie przekracza liczby rdzeni procesora.

Po rozegraniu wszystkich gier wynikowe słowniki są zbierane i wyniki zapisywane są w pliku CSV, a na standardowym wyjściu wyświetlane jest ich podsumowanie.

5.6. Strategie automatyczne

W trakcie gry jednego lub obu graczy kontrolowanych przez użytkownika mogą zastąpić strategie automatyczne zaimplementowane w module `players`. Można podzielić je na trzy kategorie: strategie referencyjne, strategie oparte na punktacji wierzchołków oraz strategie MinMax.

5.6.1. Strategie odniesienia

Strategie należące do tej kategorii są bardzo proste i mogą służyć do sprawdzania poprawności działania innych elementów aplikacji oraz jako baza do oceny efektywności innych strategii. Dodatkową ich zaletą jest stały czas działania, niezależny od wielkości grafu gry.

Pierwszą z nich jest strategia losowa, implementowana przez klasę `RandomStrategy`, która wybiera spośród krawędzi grafu, na których nie został jeszcze wykonany ruch. W tym celu użyta jest funkcja `random.choice` z biblioteki standardowej. Następnie z użyciem funkcji `random.shuffle` ustalany jest kierunek ruchu. Ma on znaczenie jedynie w standardowej grze w dominowanie rzymskie.

Druga strategia, implementowana w klasie `ReferenceStrategy`, polega na wybraniu pierwszej z listy krawędzi, na których nie wykonano jeszcze ruchu, która zwracana jest przez obiekt reprezentujący graf. Jest ona równie efektywna co strategia losowa, natomiast działa w sposób deterministyczny, co umożliwia odtworzenie konkretnej gry bez konieczności zmiany ustawień generatora liczb losowych.

5.6.2. Strategie oparte na punktacji wierzchołków

Strategie z tej rodziny dokonują oceny aktualnego stanu gry poprzez nadanie każdemu z wierzchołków grafu pewnej liczby punktów. Wartość ta jest obliczana na podstawie rodzajów krawędzi sąsiadujących z wierzchołkiem. Następnie wybierany jest wierzchołek u o najwyższej punktacji, dla którego $|E_u| > 0$, gdzie E_u jest zbiorem krawędzi sąsiadujących z u , na których nie został jeszcze zagrany ruch. Wtedy znajdowana jest taka krawędź $uv \in E_u$, w której wierzchołek v ma najniższą punktację. Wówczas strategia zwraca ruch \overrightarrow{uv} .

Sposób obliczania punktacji wierzchołków jest zależny od tego, czy rozgrywana jest standardowa gra w dominowanie rzymskie, czy gra w dominowanie rzymskie z podziałem krawędzi. W pierwszym przypadku istnieją trzy typy krawędzi sąsiadujących z wierzchołkiem: łuk wychodzący, łuk wchodzący oraz krawędź nieskierowana (na której nie został jeszcze ruch). W przypadku gry z podziałem krawędzi wyróżnione są inne trzy typy: krawędź oznaczona, krawędź podzielona oraz krawędź wolna (na której nie wykonano jeszcze ruchu). W obu wariantach każdemu typowi krawędzi przypisana jest waga, a punktacja wierzchołka jest sumą wag sąsiadujących z nim krawędzi.

Powyższy heurystyczny algorytm ma w założeniu wyszukiwać wierzchołki, będące dobrymi kandydatami na dominowanie swojego sąsiedztwa (wierzchołki o wartości funkcji dominowania rzymskiego o wartości 2) w grze w dominowanie rzymskie i orientować krawędzie w kierunku sąsiadujących wierzchołków, będącymi najsłabszymi kandydatami do dominowania sąsiedztwa.

Parametry strategii opartych na punktacji wierzchołków (wagi typów krawędzi) zostały dobrane na podstawie przeszukania przestrzeni parametrów z użyciem trybu testowego. Porównane ze sobą zostały wszystkie wariacje parametrów całkowitych w zakresie $[-1, 2]$. Pominięte zostały jedynie strategie, w których wszystkie z parametrów były równe, gdyż takie wagi krawędzi sprawiają, że tracona jest informacja o wykonanych przez graczy ruchach. Przeszukanie wykonano

zarówno dla gry w dominowanie rzymskie, jak i dla gry w dominowanie rzymskie z podziałem krawędzi.

W porównaniu strategii dla standardowej gry w dominowanie rzymskie przeprowadzono 8 iteracji testów na losowych grafach o 12 wierzchołkach i 20 krawędziach. Wyniki najlepszych strategii przedstawiono w tabeli 5.3.

Gracz	Strategia (w. łuku wchodzącego, w. łuku wychodzącego, w. krawędzi nieskierowanej)	Średnia liczba dominowania rzymskiego grafu wynikowego
Dominator	Losowa	7,4693
Dominator	(-1, 2, 2)	6,1660
Dominator	(0, -1, 0)	8,0410
Psuj	Losowa	7,3873
Psuj	(-1, 2, 2)	6,3156
Psuj	(0, -1, 0)	8,1844

Tabela 5.3. Wyniki strategii opartych na punktacji wierzchołków dla standardowej gry w dominowanie rzymskie. Przedstawione zostały strategie z najlepszym wynikiem w roli Dominatora (kolor niebieski) i najlepszym wynikiem w roli Psuja (kolor zielony) oraz strategia losowa. Dla wszystkich zostały podane wyniki podczas gry w roli Dominatora oraz Psuja

Natomiast przy testowaniu strategii dla gry w dominowanie rzymskie z podziałem krawędzi przeprowadzone zostało 5 iteracji testów, w których gry odbywały się na losowych grafach o 8 wierzchołkach oraz 12 krawędziach. Zmniejszenie rozmiaru grafów i liczby iteracji było związane z dłuższym czasem obliczania najmniejszej funkcji dominowania rzymskiego w tym wariancie gry ze względu na dodawane w wyniku podziału krawędzi wierzchołki.

W przeciwieństwie do standardowego wariantu, w grze z podziałem krawędzi podobne ruchy są wartościowe dla obu graczy. Potwierdzają to wykonane testy, w których ta sama strategia (o tych samych parametrach) okazała się być najlepszą zarówno dla Dominatora, jak i dla Psuja. W tabeli 5.4 przedstawiono jej wyniki w porównaniu ze strategią losową.

Gracz	Strategia (w. krawędzi oznaczonej, w. krawędzi podzielonej, w. krawędzi wolnej)	Średnia liczba dominowania rzymskiego grafu wynikowego
Dominator	Losowa	7,6164
Dominator	(-1, 0, -1)	7,1902
Psuj	Losowa	7,6000
Psuj	(-1, 0, -1)	7,7607

Tabela 5.4. Wyniki strategii opartych na punktacji wierzchołków dla gry w dominowanie rzymskie z podziałem krawędzi. Przedstawiona została strategia z najlepszym wynikiem w roli Dominatora i Psuja oraz strategia losowa. Dla wszystkich zostały podane wyniki podczas gry w roli Dominatora oraz Psuja

5.6.3. Strategie MinMax

Trzecią kategorią zaimplementowanych strategii są te, wykorzystujące algorytm MinMax do znalezienia najlepszego ruchu. W ich implementacji zastosowana została technika *alpha-beta pruning*, polegająca na nieprzeoglądaniu poddrzew stanu gry, o których wiadomo, że nie zostaną wybrane przez grającego optymalnie gracza, ponieważ może zostać wykonany inny, korzystniejszy ruch.

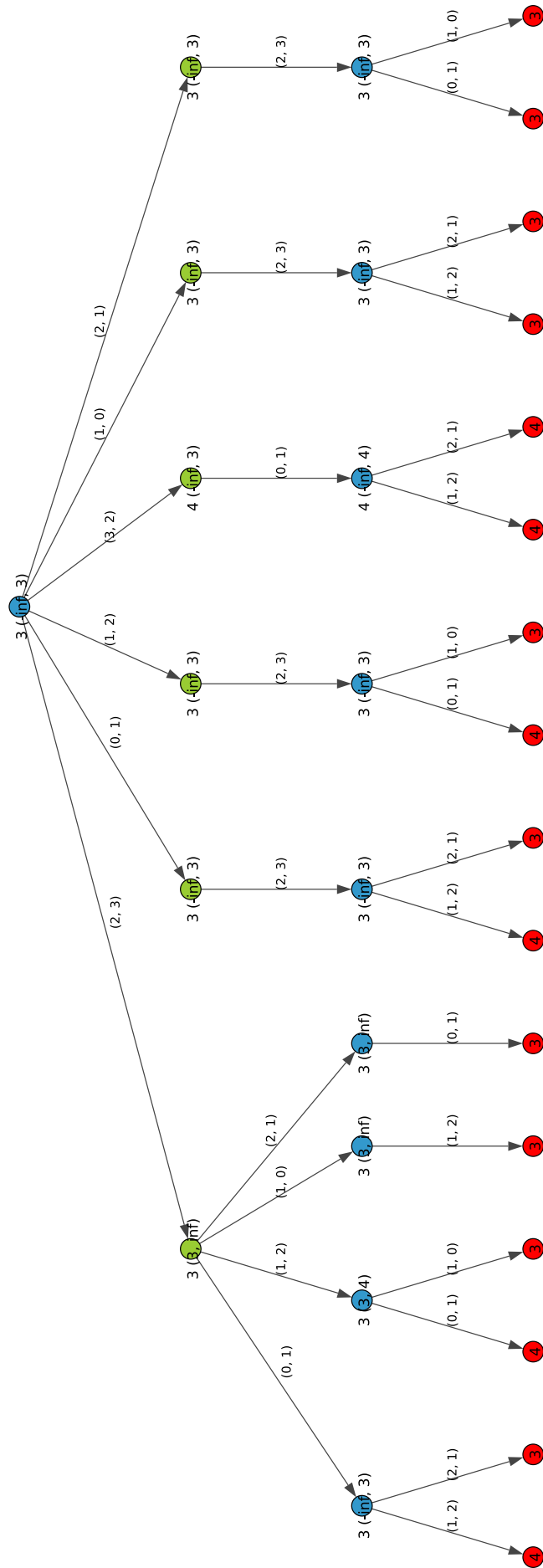
Parametrem strategii MinMax jest głębokość przeszukiwania drzewa gry. Domyślnie zmienia się ona wraz z liczbą pozostałych w grze ruchów w celu zmniejszenia całkowitego czasu obliczeń,

natomiast można ustalić też stałą głębokość lub wymusić przeszukanie całego drzewa stanów gry, co daje optymalną strategię.

W przypadku, gdy przeszukiwane jest całe drzewo stanów gry, wartości wierzchołków terminalnych (liści) odpowiadają liczbom dominowania rzymskiego grafów końcowych. Natomiast gdy przeszukiwanie kończy się przed sprawdzeniem wszystkich ruchów, wartości wierzchołków obliczane są z użyciem następującej heurystyki:

- w przypadku gry w dominowanie rzymskie: liczba dominowania rzymskiego aktualnego grafu gry, przyjmując, że nieskierowana krawędź pomiędzy wierzchołkami odpowiada cyklowi dwum łukom pomiędzy nimi skierowanym w przeciwnie strony;
- w przypadku gry w dominowanie rzymskie z podziałem krawędzi: liczba dominowania rzymskiego aktualnego grafu gry.

Możliwa jest wizualizacja drzewa stanów gry dla strategii MinMax, jeżeli ustawiony jest odpowiedni parametr w module `config`. Implementacja wizualizacji została opisana w rozdziale 5.5.2. Wynikowa grafika przedstawia poszczególne rozważane stany gry wraz z ich wartościami i wartościami parametrów α i β algorytmu *alpha-beta pruning*.



Rys. 5.4. Wizualizacja drzewa stanów gry w strategii MinMax. Wierzchołki niebieskie są minimalizujące, a zielone maksymalizujące. Wierzchołki czerwone symbolizują stany końcowe gry. Pary liczb przy krawędziach oznaczają ruchy graczy, a opisy w wierzchołkach zawierają wartość wierzchołka i przedział $[\alpha, \beta]$

6. PODSUMOWANIE

6.1. *Osiągnięcia*

W ramach niniejszej pracy opracowano teoretyczne wprowadzenie do tematyki gier w dominowanie rzymskie w wariantcie standardowym oraz z podziałem krawędzi poprzez analizę informacji dostępnych w literaturze, uzupełnionych autorskim opisem i dowodami. Wymagane pojęcia przedstawiane są od podstaw, natomiast ilustracje i szczegółowo opracowane dowody umożliwiają łatwiejsze zrozumienie treści, niż ma to miejsce w przypadku zwiezłych publikacji naukowych.

Drugim etapem pracy było stworzenie aplikacji do gry w dominowanie rzymskie oraz gry w dominowanie rzymskie z podziałem krawędzi. Może ona być używana jako samodzielne narzędzie, ale również jako dodatkowy zasób, wspomagający czytanie opracowania teoretycznego. Umożliwia analizowanie strategii przedstawianych w dowodach twierdzeń, jak i definiowanie nowych w ramach badań.

Został również zaimplementowany zestaw podstawowych strategii automatycznych dla obu wariantów gry. Są one przeznaczone do grania na dowolnym grafie, nie są zoptymalizowane dla żadnej konkretnej klasy, w przeciwieństwie do większości strategii przedstawianych w dowodach. Dzięki modularnej architekturze aplikacji możliwe jest rozszerzenie jej o nowe strategie automatyczne. Przygotowany został skrypt, pozwalający na sprawdzenie efektywności poszczególnych strategii w porównaniu z innymi. Zaimplementowane strategie podstawowe mogą więc służyć jako baza porównawcza dla dodatkowych strategii, opracowywanych przez użytkowników.

Jedna z opracowanych strategii, znajdująca ruch do wykonania na podstawie wag przypisanych wierzchołkom, jest autorskim rozwiązaniem, niewystępującym w literaturze. Przeprowadzono eksperyment, w wyniku którego ustalono dla niej efektywne parametry w obu wariantach gier oraz do gry w roli obu graczy.

6.2. *Dalszy rozwój aplikacji*

Aplikacja wytworzona w ramach pracy spełnia postawione jej założenia. W aktualnym stanie może służyć jako narzędzie do lepszego zrozumienia problematyki gier w dominowanie rzymskie oraz badanie efektywności strategii.

W ramach dalszego rozwoju aplikacji można dodatkowo poszerzyć jej użyteczność w tym zakresie poprzez umożliwienie wygenerowania określonych klas grafów, dzięki czemu użytkownik nie musiałby używać w tym celu osobnego narzędzia. Używana biblioteka `pygraph` implementuje generowanie niektórych prostych klas, więc jest to narzędzie, które mogłoby zostać stosunkowo łatwo wykorzystane.

Kolejnym obiecującym kierunkiem rozwoju aplikacji jest dodanie nowych strategii automatycznych. Użyte rozwiązania projektowe pozwalają na ich nieskomplikowane rozwijanie. Może to być szczególnie przydatne w badaniach nad grafami o konkretnych własnościach, gdzie strategie automatyczne mogą wykorzystywać ich charakterystyczne cechy.

Inną możliwością jest uczynienie aplikacji bardziej przyjazną dla użytkownika poprzez dodanie graficznego interfejsu, pozwalającego na ustawienie parametrów gry. Mogłoby to poszerzyć grono odbiorców aplikacji o osoby nieposługujące się komfortowo interfejsem tekstowym.

Istnieje również z pewnością wiele sposobów optymalizacji użytych algorytmów, na przykład w zakresie wyliczania najmniejszej funkcji dominowania rzymskiego, czy realizacji algorytmu MinMax w strategiach automatycznych. Jednym z możliwych rozwiązań byłoby zaimplementowanie sekcji o wysokiej złożoności obliczeniowej w osobnych kompilowanych modułach, napisanych w języku C. Takie rozwiązanie pozwala na lepszą kontrolę nad używanymi strukturami danych i umożliwia potencjalne przyspieszenie wykonania tych części programu. W wersji aplikacji stworzonej w ramach pracy przejrzystość i łatwość analizy kodu źródłowego stanowiła wyższy priorytet niż szybkość działania.

6.3. Wnioski

Obszar teorii grafów, stanowiący temat tej pracy, wydaje się mieć duży, niezrealizowany jeszcze w pełni potencjał praktycznego zastosowania w rzeczywistych problemach optymalizacyjnych. W związku z tym uzasadnione jest prowadzenie dalszych badań teoretycznych, ale równie istotne jest zapewnienie łatwego dostępu dla środowiska pozaakademickiego do zdobytej wiedzy.

Prezentowana praca stanowi wkład w upowszechnieniu tematyki z zakresu gier w dominowanie rzymskie zarówno w środowisku teoretyków z innych dziedzin, jak i praktyków nieposiadających obszernej wiedzy na temat teorii grafów.

Jako uzupełnienie powyższego celu zostało wytworzone narzędzie, które pozwala na wizualizację omawianych koncepcji. Dzięki temu również może ułatwić zrozumienie tematyki, która w komunikacji naukowej często jest przedstawiana w abstrakcyjny sposób, i której adresatami są tylko inni badacze.

WYKAZ LITERATURY

- [1] N. Alon, J. Balogh, B. Bollobás, and T. Szabó. Game domination number. *Discrete Mathematics*, 256(1-2):23–33, 2002.
- [2] J. Amjadi, H. Karami, S. M. Sheikholeslami, and L. Volkmann. Roman game domination subdivision number of a graph. *Transactions on Combinatorics*, 2(4), 2013.
- [3] A. Bahremandpour, S. M. Sheikholeslami, and L. Volkmann. Roman game domination number of a graph. *Journal of Combinatorial Optimization*, 33(2):713–725, 2017.
- [4] Walter William Rouse Ball. *Mathematical Recreations and Problems of Past and Present Times*. Macmillan and Company, 1892. Google-Books-ID: uGgxQAAMAAJ.
- [5] B. Bollobás and T. Szabó. The oriented cycle game. *Discrete Mathematics*, 186(1-3):55–67, 1998.
- [6] E. J. Cockayne, P. A. Dreyer, S. M. Hedetniemi, and S. T. Hedetniemi. Roman domination in graphs. *Discrete Mathematics*, 278(1-3):11–22, 2004.
- [7] E. J. Cockayne, P. J. P. Grobler, W. R. Grundlingh, J. Munganga, and J. H. Van Vuuren. Protection of a graph. *Utilitas Mathematica*, 67, 2005.
- [8] E. J. Cockayne and S. T. Hedetniemi. Towards a theory of domination in graphs. *Networks*, 7(3):247–261, 1977.
- [9] G. Csárdi, T. Nepusz, S. Horvát, V. Traag, F. Zanini, and D. Noom. igraph, January 2023. <http://igraph.org>.
- [10] O. Favaron, H. Karami, and S. M. Sheikholeslami. Game domination subdivision number of a graph. *Journal of Combinatorial Optimization*, 30(1):109–119, 2015.
- [11] T. Gallai. Über extreme punkt-und kantenmengen, annales universitatis scientiarum budapestinensis de rolando eotvos nominatae. *Sectio Mathematica*, 2:133–138, 1959.
- [12] D. König. Gráfok és mátrixok. *Matematikai és Fizikai Lapok*, 38:116–119, 1931.
- [13] Øystein Ore. *Theory of graphs*. Colloquium publications (American Mathematical Society). American Mathematical Society, Providence, R.I., 3d print. with corrections edition, 1967. OCLC: 1222377.
- [14] S. M. Sheikholeslami and L. Volkmann. The Roman domination number of a digraph. *Acta Universitatis Apulensis. Mathematics. Informatics*, 27:77–86, 2011.
- [15] P. Shinnars. PyGame, 2011. <http://pygame.org>.
- [16] L. Volkmann. A characterization of bipartite graphs with independence number half their order. *Australasian Journal of Combinatorics*, 41:219, 2008.
- [17] D. B. West. *Introduction to graph theory*. Prentice Hall, Upper Saddle River, NJ, 1996.

WYKAZ RYSUNKÓW

2.1	Sąsiedztwo - relacja dominowania	10
3.1	Gra w dominowanie rzymskie w klice K_3	17
3.2	Ścieżki P_2 w klikach K_5 i K_6	18
3.3	Strategia Dominatora na grafie K_5	19
3.4	Gra w dominowanie rzymskie na ścieżce P_n	21
3.5	Gra w dominowanie rzymskie na ścieżce P_5	21
3.6	Podwójna gwiazda $DS_{3,5}$	22
3.7	Przykładowa gra w dominowanie rzymskie na $DS_{1,6}$	23
3.8	Przykładowa gra w dominowanie rzymskie na $DS_{3,5}$	25
3.9	Najdłuższa ścieżka w drzewie T w przypadku $d(v_{k-1}) = 2$	26
3.10	Najdłuższa ścieżka w drzewie T w przypadku $d(v_{k-1}) \geq 3$	27
4.1	Przykłady gry w dominowanie rzymskie z podziałem krawędzi na gwiazdach o małej liczbie wierzchołków	29
4.2	Przykłady gry w dominowanie rzymskie z podziałem krawędzi na gwieżdzie $K_{1,7}$	30
4.3	Graf F_8	33
4.4	Ilustracja gry w dominowanie rzymskie z podziałem krawędzi na grafie F_4	34
4.5	Podział na zbiory D , B i A przykładowego grafu	35
4.6	Najmniejsze funkcje dominowania rzymskiego dla gry w dominowanie rzymskie z podziałem krawędzi na $DS_{1,q}$ dla $q \in \{2, 3\}$	40
4.7	Przykład gry w dominowanie rzymskie z podziałem krawędzi na grafie $DS_{1,6}$	41
4.8	Najdłuższa ścieżka w drzewie T ze wspólnym wierzchołkiem $v_3 = v_{k-2}$	47
4.9	Drzewo T , w którym średnica jest większa od 4	47
4.10	Drzewo T' , w którym $d(v_2)$ jest parzyste	48
4.11	Drzewo T' , w którym $d(v_2)$ oraz $d(v_{k-1})$ są nieparzyste	49
4.12	Drzewo T' , w którym jest przynajmniej jeden wierzchołek wspierający końcowy stopnia ≥ 5	51
4.13	Drzewo T' , w którym $n(T) = 9$ oraz $z = z'$	52
4.14	Drzewo T' , w którym są przynajmniej dwa wierzchołki wspierające końcowe stopnia 4	53
4.15	Możliwe przypadki drzewa T' , gdy jest co najwyżej jeden wierzchołek wspierający końcowy stopnia 4	54
5.1	Okno gry w trybie graficznym z domyślnymi parametrami	57
5.2	Okno gry w trybie graficznym w grze z podziałem krawędzi	58
5.3	Podział aplikacji na moduły i klasy	62
5.4	Wizualizacja drzewa stanów gry w strategii MinMax	69

WYKAZ TABEL

3.1	Strategia Psuja na drzewie $T, n(T) = 3$	26
3.2	Strategia Psuja na drzewie $T, n(T) \geq 4$	26
4.1	Wartości najmniejszej funkcji dominowania rzymskiego w pojedynczym cyklu grafu F_k	34
5.1	Parametry skryptu <code>rdg.py</code>	57
5.2	Konfiguracja testów strategii automatycznych	59
5.3	Wyniki strategii opartych na punktacji wierzchołków dla standardowej gry w dominowanie rzymskie	67
5.4	Wyniki strategii opartych na punktacji wierzchołków dla gry w dominowanie rzymskie z podziałem krawędzi	67