

Міністерство освіти і науки України
НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ “КИЄВО-МОГИЛЯНСЬКА
АКАДЕМІЯ”

Кафедра математики факультету інформатики

**Курсова робота на тему:
Парно-знакові графи**

Керівник курсової роботи:
к. ф.-м. н. *Козеренко С.О.*
(*прізвище та ініціали*)

(*підпис*)
“ _____ ” _____ 2023 р.

Виконала студентка
3-го року навчання спеціальності
113 “Прикладна математика”
Гуназа Анна Олегівна
(*ПІВ*)

Міністерство освіти і науки України
НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ “КИЄВО-МОГИЛЯНСЬКА
АКАДЕМІЯ”

Кафедра математики факультету інформатики

ЗАТВЕРДЖУЮ
Зав. кафедри математики, доц., к.ф-м.н.
_____ Р.К. Чорней
(підпис)
“ _____ ” _____ 2023 р.

ІНДИВІДУАЛЬНЕ ЗАВДАННЯ
на курсову роботу
студентці 3-го курсу факультету інформатики
Гуназі Анні Олегівні

Тема: Парно-зункові графи.

Вихідні дані: Досліджено особливості парно-знакових графів.

Зміст ТЧ до курсової роботи:

Анотація

Вступ

1 Основні означення

2 Основні результати

2.1 Збалансованість і характеристика PSG

2.2 Алгоритм перевірки, чи зв'язний знаковий граф є PSG

2.3 Число 'gna'

2.4 Верхня оцінка на 'gna' для дерев

3. Програмна реалізація алгоритму на Python

3.1 Основні функції

3.2 Застосування алгоритму на прикладах

Висновки

Література

Дата видачі “ _____ ” _____ 2023 р. Керівник _____

(підпис)

Завдання отримав _____

(підпис)

Тема: Парно-знакові графи.

Календарний план виконання роботи:

Номер	Назва етапу курсової	Термін виконання етапу	Примітка
1.	Отримання теми курсової роботи.	01.08.2022	
2.	Ознайомлення з темою курсової.	01.09.2022	
3.	Розробка плану та структури роботи.	15.09.2022	
4.	Робота з науковою літературою, опис основних означень теорії графів.	15.10.2022	
5.	Дослідження властивостей парно-знакових графів.	01.01.2023	
6.	Робота над текстовим оформленням результатів.	01.04.2023	
7.	Попередній аналіз курсової. Виправлення помилок.	01.05.2023	
8.	Захист курсової роботи.	22.05.2023	

Зміст

Анотація	5
Вступ	6
1 Основні означення	7
2 Основні результати	9
2.1 Збалансованість і характеристика PSG	9
2.2 Алгоритм перевірки, чи зв'язний знаковий граф є PSG . . .	13
2.3 Число 'гпа'	14
2.4 Верхня оцінка на 'гпа' для дерев	16
3 Програмна реалізація алгоритму на Python	17
3.1 Основні функції	17
3.2 Застосування алгоритму на прикладах	21
Висновки	22
Література	23

Анотація

Знаковим графом називають пару $S = (G, \sigma)$, де G – граф, $\sigma_f : E(G) \rightarrow \{+, -\}$ – відображення, яке кожному ребру графа ставить у відповідність знак $+$ чи $-$. Знаковий граф $S = (G, \sigma)$ називається *парно-знаковим графом*, якщо існує бієкція $f : V(G) \rightarrow \{1, 2, \dots, n\}$ така, що для кожного ребра uv в G : $\sigma_f(uv) = +$, якщо $f(u)$ та $f(v)$ мають однакову парність, та $\sigma_f(uv) = -$, якщо різну.

Мета роботи полягає в дослідженні парно-знакових графів та їхніх властивостей, побудові верхньої оцінки на ‘гпа’ дерев, та реалізуванні алгоритму перевірки знакового графа на те, чи є він парно-знаковим.

Ключові слова: парно-знаковий граф, збалансованість, число ‘гпа’, знакові дерева, алгоритм.

Вступ

Тема парно-знакових графів є доволі новою та активно досліджується сучасними математиками. Поняття «знакового графа» і «збалансованості» з'явилися вперше в математичній роботі Франка Харарі [3] в 1953 році. Після цього знакові графи повторно були відкриті багато разів, адже вони природно виникають у багатьох областях, серед них топологічна теорія графів, теорія груп та інші.

Знакові графи мають досить широке застосування у соціальній психології, фізиці (зокрема у дослідженні спінового скла), аналізі складних систем, кластеризації даних та нейронауці. До прикладу, у соціальній психології вони використовуються для моделювання ситуацій, де позитивні ребра представляють дружбу, а негативні ребра - ворожнечу. Тоді позитивні цикли означають стабільні соціальні ситуації, а негативні - нестабільні. Якщо розглядати кластеризацію даних, то об'єкти беруться за вершини графа, а ребра отримують свій знак в залежності від того, чи ним з'єднуються подібні елементи.

У цій роботі розглядаються парно-знакові графи та досліджуються їх властивості. Основні поняття та результати включають збалансованість, характеристику збалансованих знакових графів, алгоритм перевірки графа на парно-знаковість, дослідження числа $\gamma_{\text{пн}}$, яке дорівнює найменшій кількості негативних ребер у парно-знаковому графі, та верхню оцінку $\gamma_{\text{пн}}$ для дерев. У другій частині роботи реалізовується вище згаданий алгоритм на мові програмування Python та наводиться його застосування на конкретних прикладах.

1 Основні означення

Означення 1.1. *Неорієнтований граф* – це пара $G = (V, E)$, де

$$V = V(G) \text{ – множина вершин,}$$

$$E = E(G) \subset V^{(2)} = \{\{a, b\} \mid a, b \in V\} \text{ – множина ребер.}$$

Означення 1.2. *Простий граф* – це граф без кратних ребер та петель.

Означення 1.3. *Зв'язний граф* – це граф, між будь-якою парою вершин якого існує шлях, який їх сполучає.

Надалі в роботі всі графи вважаються простими та зв'язними, якщо не вказано протилежне.

Означення 1.4. *Компонента зв'язності* графа – це його максимальний (за включенням) зв'язний підграф.

Означення 1.5. *Шлях* між парою вершин $x, y \in V(G)$ – це послідовність вершин v_0, v_1, \dots, v_n , де $x = v_0$ – початок шляху, $y = v_n$ – кінець шляху, $\forall i = \overline{0, n-1} : v_i v_{i+1} \in E(G)$.

Означення 1.6. *Цикл* – це шлях, у якого початок та кінець збігаються.

Означення 1.7. *Двочастковий граф* – граф, множину вершин якого можна розбити на дві підмножини (*частки*) так, щоб жодні дві вершини з однієї підмножини не були суміжними, тобто

$$V = V_1 \sqcup V_2, \quad E \subset \{\{a, b\} \mid a \in V_1, b \in V_2\}.$$

Означення 1.8. *Повний граф* K_n – граф, у якому кожні дві вершини з'єднані ребром, тобто $E = V^{(2)}$.

Означення 1.9. *Зірка* $K_{1,n}$ – це повний двочастковий граф з єдиним внутрішнім вузлом і n листками.

Означення 1.10. *Дерево* – це зв'язний граф, який не містить циклів.

Означення 1.11. *Знаковий граф* – це пара $S = (G, \sigma)$, де

$$G \text{ – граф,}$$

$$\sigma_f : E(G) \rightarrow \{+, -\}$$

Ребра зі знаком $+(-)$ називають *позитивними (негативними)* ребрами графа S .

На схематичних зображеннях графа для позначення позитивних ребер використовуватиметься суцільна лінія, негативних – пунктирна.

Означення 1.12. Знаковий граф $S = (G, \sigma)$ називається *парно-знаковим графом*, якщо існує бієкція $f : V(G) \rightarrow \{1, 2, \dots, n\}$ така, що для кожного ребра uv в G : $\sigma_f(uv) = +$, якщо $f(u)$ та $f(v)$ мають однакову парність, та $\sigma_f(uv) = -$, якщо різну.

Надалі в роботі парно-знаковий граф може позначатися як PSG – з англійської – parity signed graph.

Означення 1.13. Знаковий граф називається *позитивним*, якщо він не містить негативних ребер: $E^- = \emptyset$, та *негативним*, якщо не містить позитивних ребер: $E^+ = \emptyset$.

Означення 1.14. Знаковий граф називається *однорідним*, якщо він є або позитивним, або негативним. Інакше граф називається *неоднорідним*.

Означення 1.15. *Позитивна (негативна) компонента* знакового графа - це максимальний зв'язний знаковий підграф, у якого всі ребра є позитивними (негативними).

2 Основні результати

2.1 Збалансованість і характеристика PSG

Означення 2.1. Якщо добуток знаків будь-якого циклу графа S додатний, то S називають *збалансованим знаковим графом*.

Теорема 2.2. *Будь-який парно-знаковий граф є збалансованим.*

Доведення. Випливає з того, що будь-який парно-знаковий цикл не може мати непарної кількості негативних ребер. Від супротивного. Нехай граф C є парно-знаковим графом і кількість негативних ребер $|E^-(C)|$ є непарною. Цикл $C = C_n$ містить позитивні та негативні підланцюги. Кінці позитивних підланцюгів мають однакову парність, тому їх можна стягнути. Помічаємо, що негативні підланцюги довжини два P_2^- також мають кінці однакової парності, тому їх теж стягуємо. У результаті отримуємо одну вершину v з негативною петлею, тобто v має одночасно різну парність, що неможливо. Отже, парно-знаковий цикл завжди має парну кількість негативних ребер, тому він є збалансованим. \square

У зворотню сторону ця теорема не діє. Наприклад, позитивний знаковий граф, який є збалансованим, не є парно-знаковим. Інший приклад надано на рисунку 1.

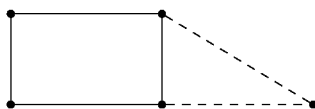


Рис. 1: Збалансований граф, який не є PSG

Теорема 2.3. *Нехай S - зв'язний знаковий граф. Тоді S є збалансованим тоді і тільки тоді, коли множину вершин графа $V(S)$ можна розділити на дві підмножини V_1 і V_2 так, щоб ребра $uv \in E(V_1, V_2)$ були негативними і лише вони.*

Доведення. Необхідність. Нехай V_1, V_2 - розбиття вершин, C - цикл в G . Тоді C містить парну кількість ребер що з'єднує V_1 з V_2 . Отже, C містить парну кількість негативних ребер.

Достатність. Нехай S - збалансований знаковий граф. Зафіксуємо $u \in V(G)$. Покладемо $V_1 := \{x \in V(G) : \exists u \sim x \text{ з парною кількістю негативних ребер}\}$, $V_2 := V(G) \setminus V_1$, $V(G) = V_1 \sqcup V_2$.

Покажемо, що підграф $G[V_1]$ не містить негативних ребер. Від супротивного, припустимо, що існує $ab \in E(V_1)$ - негативне ребро. Тоді існує P_1, P_2 з парною кількістю негативних ребер. Маємо замкнутий маршрут із непарною кількістю негативних ребер. Він містить простий цикл із непарною кількістю негативних ребер, тому в $G[V_1]$ їх немає. Аналогічно, $G[V_2]$ не містить негативних ребер. Отже, в даному розбитті негативні ребра поєднуюватимуть лише вершини з компонент V_1 та V_2 .

□

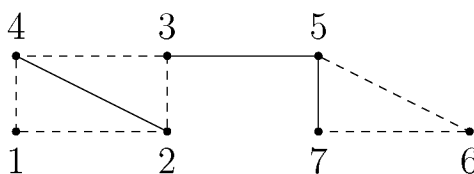
Теорема 2.4. *Нехай S - зв'язний знаковий граф. Тоді S є парно-знаковим графом тоді і тільки тоді, коли $V(S)$ можна розділити на дві підмножини V_1 і V_2 так, щоб ребра $uv \in E(V_1, V_2)$ були негативними і лише вони, а також щоб $||V_1| - |V_2|| \leq 1$.*

Доведення. Необхідність. Нехай існують такі V_1 і V_2 . Нехай $|V_1| \geq |V_2|$. Тоді V_1 - непарні вершини, V_2 - парні. З цього випливає, що S є парно-знаковим графом.

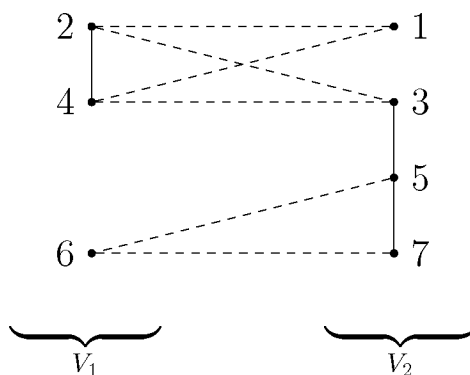
Достатність. Нехай S - парно-знаковий граф. Тоді існує $f : V \rightarrow \{1, \dots, n\}$. Покладемо, що V_1 - множина вершин з непарним номером, V_2 - з парним номером. Дійсно, якщо $|V(S)|$ парна, то $|V_1| = |V_2|$, якщо - непарна, то $|V_1| = |V_2| + 1$. Тобто, $||V_1| - |V_2|| \leq 1$. □

Приклад 2.5. Розбиття Харарі.

Маємо наступний парно-знаковий граф G :



Отримуємо таке розбиття:



Усі вершини зліва належать до компоненти V_1 , справа - до V_2 . Самі компоненти поєднані між собою негативними ребрами. І справді виконується умова $||V_1| - |V_2|| \leq 1$.

Наслідок 2.6. *Зв'язний негативний знаковий граф S є парно-знаковим графом тоді і тільки тоді, коли він є негативним двочастковим знаковим графом, у якого частки відрізняються не більш як на одну вершину.*

Наслідок 2.7. *Знакова зірка $K_{1,n}$ є PSG тоді і тільки тоді, коли вона задовольняє наступні умови:*

1. $|E^-(K_{1,n})| = |E^+(K_{1,n})|$ або
 $|E^-(K_{1,n})| = |E^+(K_{1,n})| + 2$, якщо n - парне.
2. $|E^-(K_{1,n})| = |E^+(K_{1,n})| + 1$, якщо n - непарне.

Доведення. Необхідність. Нехай вершина u - центр зірки $K_{1,n}$. Покладемо $f(u) = 1$. Тоді:

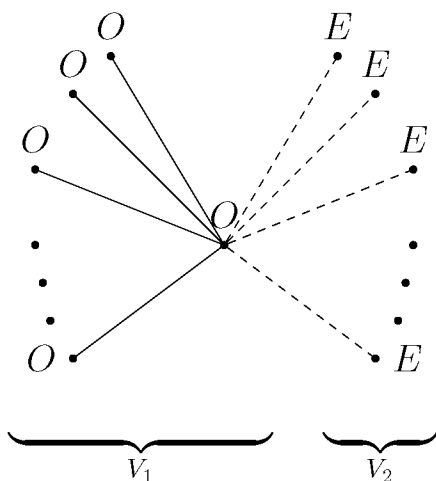
1. Якщо n парне, то серед номерів вершин $\{2, \dots, n+1\}$ однакова кількість парних і непарних чисел, тобто $|E^-(K_{1,n})| = |E^+(K_{1,n})|$. Тоді для вершин $v \in V(K_{1,n})$ покладемо $f(v)$ - парне для $uv \in E^-(K_{1,n})$ і $f(v)$ - непарне для $uv \in E^+(K_{1,n})$. Отримали бієкцію $f : V(K_{1,n}) \rightarrow \{1, 2, \dots, n\}$, тому $K_{1,n} \in PSG$.
2. Якщо n непарне, то серед $\{2, \dots, n+1\}$ парних чисел на одне більше. Тоді для $v \in V(K_{1,n})$ покладемо $f(v)$ - парне для $uv \in E^-(K_{1,n})$ і $f(v)$ - непарне для $uv \in E^+(K_{1,n})$. Отримали бієкцію f , тому $K_{1,n} \in PSG$.

Аналогічно для $f(u) \neq 1$.

Достатність. Доведемо через критерій Харарі. $K_{1,n} \in PSG$. Тоді існує розбиття Харарі $V(K_{1,n}) = V_1 \sqcup V_2$, а також $||V_1| - |V_2|| \leq 1$.

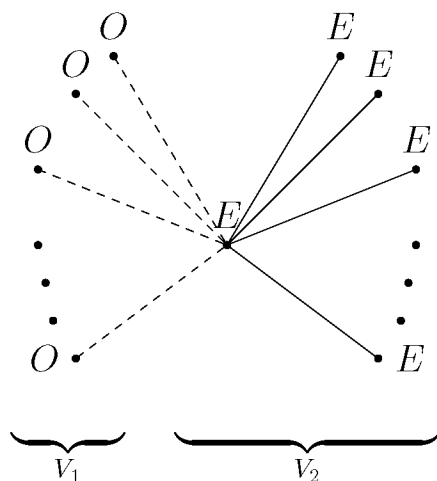
1. Якщо n парне, то $|V(K_{1,n})| = n+1$ - непарне. Нехай $|V_1| = |V_2| + 1$. Тоді V_1 відповідає за непарні числа. Є два варіанти:

а)



Центр u зірки – непарне число, $u \in V_1$. Тоді $|E^+(K_{1,n})| = |E^-(K_{1,n})|$

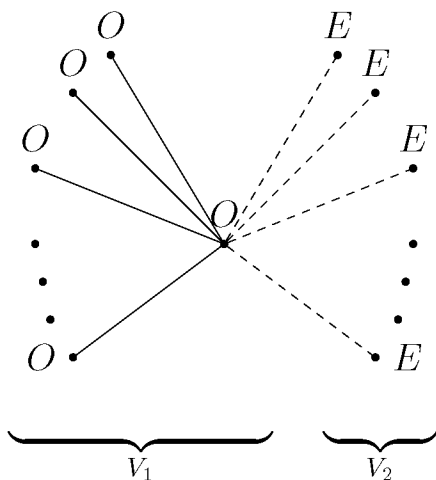
б)



Центр u – парне, $u \in V_2$. Тоді $|E^-(K_{1,n})| = |E^+(K_{1,n})| + 2$

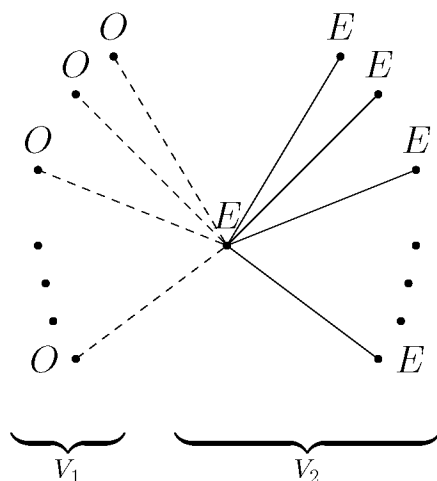
2. Якщо n непарне, то $|V(K_{1,n})| = n + 1$ парне. Тоді $|V_1| = |V_2|$. Нехай множина V_1 – непарні числа. Є два варіанти:

а)



Центр u зірки – непарне число, $u \in V_1$. Тоді $|E^+(K_{1,n})| = |E^-(K_{1,n})| - 1$

б)



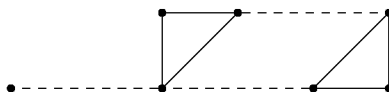
Центр u – парне, $u \in V_2$. Тоді $|E^-(K_{1,n})| = |E^+(K_{1,n})| + 1$

□

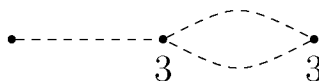
2.2 Алгоритм перевірки, чи зв'язний знаковий граф є PSG

1. Перевірити, чи S є збалансованим. Якщо ні, то S не є PSG.
2. Стягнути кожен позитивну компоненту в точку і помітити її числом вершин, які були в компоненті.
3. Кожна непомічена вершина отримує номер 1.
4. Отриманий граф є негативно-однорідним та збалансованим, а тому і двочастковим. Знайти двочасткове розбиття і просумувати мітки в кожній частці. Якщо суми відрізняються більш, ніж на 1, то S не є PSG. Інакше - PSG.

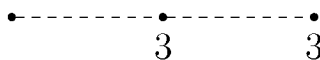
Приклад 2.8. Покажемо як діє алгоритм на даному графі G :



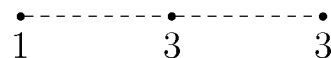
1. Як бачимо, усі цикли графа є додатними, тому G є збалансованим. Перша умова виконується, отже переходимо до наступного кроку.
2. Після стягнення позитивних компонент та проставлення міток отримуємо такий граф:



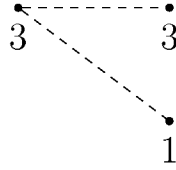
Оскільки наявність подвійних ребер не впливає на двочасткове розбиття, то приберемо їх та отримаємо:



3. Маркуємо одиницею непомічену вершину:



4. Знаходимо двочасткове розбиття і сумуємо мітки в кожній частці:



У результаті маємо, що у першій частці три вершини, у другій – чотири, тобто $|3 - 4| \leq 1$, а отже граф $S \in PSG$.

2.3 Число ‘гна’

Означення 2.9. Для графа G число ‘гна’ – це найменша кількість негативних ребер $E^-(S_f)$ серед всіх можливих знакових графів S_f над G , утворених бієкцією $f : V(G) \rightarrow \{1, \dots, n\}$. Позначається $\sigma^-(G)$.

Зауваження 2.10. $\sigma^-(G) = \min \left\{ |E(A, V \setminus A)| : |A| = \left\lfloor \frac{n}{2} \right\rfloor \right\}$.

Твердження 2.11. Для знакової зірки $K_{1,n}$, яка містить $n + 1$ вершину, виконується рівність: $\sigma^-(K_{1,n}) = \left\lfloor \frac{n}{2} \right\rfloor$.

Доведення. Кількість негативних ребер залежить від парності центральної вершини u . Розглянемо два випадки:

1. $n \in$ непарним. Тоді:

Нехай u має номер $n + 1$. Листки занумеруємо від 1 до n . Серед них рівно $\frac{n-1}{2}$ парних вершин і $\frac{n+1}{2}$ непарних. Оскільки $n + 1 \in$ парним, то кількість негативних ребер $|E^-(K_{1,n})| = \frac{n+1}{2}$.

Нехай u має номер n . Серед листків рівно $\frac{n+1}{2}$ парних вершин і $\frac{n-1}{2}$ непарних. Знову маємо $|E^-(K_{1,n})| = \frac{n+1}{2}$.

2. $n \in$ парним. Тоді:

Нехай u має номер $n + 1$. Листки занумеруємо від 1 до n . Серед них рівно $\frac{n}{2}$ парних вершин і $\frac{n}{2}$ непарних. Оскільки $n + 1 \in$ непарним, то $|E^-(K_{1,n})| = \frac{n}{2}$.

Нехай u має номер n . Серед листків рівно $(\frac{n}{2} - 1)$ парних вершин і $(\frac{n}{2} + 1)$ непарних. Маємо $|E^-(K_{1,n})| = \frac{n}{2} + 1$.

В обох випадках мінімальна кількість негативних ребер $\sigma^-(K_{1,n}) = \left\lfloor \frac{n}{2} \right\rfloor$. \square

Наслідок 2.12. Для будь-якого натурального числа k існує парно-знаковий граф S , такий що $\sigma^-(S) = k$.

Доведення. $\forall k \in \mathbb{N}$ існує знакова зірка S із $\sigma^-(S) = k$. \square

Теорема 2.13. Число ‘гпа’ для шляху P_n , де $n \geq 2$, дорівнює 1. Число ‘гпа’ для циклу C_n дорівнює 2.

Теорема 2.14. Нехай G – зв’язний граф. Тоді $\sigma^-(G) = 1$ тоді і тільки тоді, коли G має міст, який з’єднує два підграфи, потужність яких відрізняється не більше, ніж на 1.

Доведення. Необхідність. Якщо $\sigma^-(G) = 1$, то існує парно-знаковий граф $S = (G, \sigma)$ із рівно одним негативним ребром. Усі вершини позитивної компоненти мають однакову парність, а оскільки серед номерів вершин $\{1, \dots, n\}$ є парні та непарні числа, то позитивних компонент дві або більше. Він видалення одного ребра граф може розпастися на дві або менше компоненти. Оскільки негативне ребро у нас одне, то воно і буде мостом.

Достатність. За допомогою функції f розставимо мітки на вершинах графа G так, щоб один підграф складався виключно з парних вершин, інший підграф – непарних. Тоді ребра всередині цих підграфів є позитивними, а ребро, що їх з’єднує – негативним. Таким чином, найменша кількість негативних ребер $\sigma^-(G) = 1$. \square

Твердження 2.15. Для будь-якого знакового графа G виконується нерівність: $\sigma^-(G) + \sigma^-(\overline{G}) \leq \left\lceil \frac{n}{2} \right\rceil \cdot \left\lfloor \frac{n}{2} \right\rfloor$, $n = |V(G)|$.

Для знакової зірки $K_{1,n}$, яка містить $n + 1$ вершину, виконується рівність: $\sigma^-(K_{1,n}) = \left\lceil \frac{n}{2} \right\rceil$.

Доведення. Зафіксуємо функцію f , яка дає $\sigma^-(G)$. Розглянемо її і для доповнення графа \overline{G} . Нехай V_{odd} – множина вершин, які отримали непарні числа як мітку, а $V_{even} = V \setminus V_{odd}$ – парні. Тоді:

$$\begin{aligned} \sigma^-(G) + \sigma^-(\overline{G}) &\leq |E_f^-(G)| + |E_f^-(\overline{G})| = \\ &= |E_f^-(G)| + |f^{-1}(V_{odd})| \cdot |f^{-1}(V_{even})| - |E_f^-(G)| = \\ &= |f^{-1}(V_{odd})| \cdot |f^{-1}(V_{even})| = \left\lceil \frac{n}{2} \right\rceil \cdot \left\lfloor \frac{n}{2} \right\rfloor. \end{aligned}$$

\square

2.4 Верхня оцінка на ‘гпа’ для дерев

Нехай T – дерево. Для будь-якої вершини x дерева T розглянемо компоненти зв'язності $T \setminus \{x\}$.

Означення 2.16. Вагою $w(x)$ вершини x називають потужність максимальної такої компоненти. Множина вершин x із найменшою вагою називається *центроїдом* дерева.

Наслідок 2.17. Центроїд дерева є вершиною або ребром.



(а) Дерево T_1 з центроїдом-вершиною

(б) Дерево T_2 з центроїдом-ребром

Рис. 2: Приклади дерев з позначеною вагою вершин

Зауваження 2.18. Якщо в дереві T центр є ребром, то $\sigma(T)^- = 1$.

Наслідок 2.19. Нехай T – дерево із 1-точковим центроїдом $\{u\}$. Тоді

$$\sigma^-(T) \leq \left\lfloor \frac{n}{2} \right\rfloor - w(u) + 1.$$

Доведення. Очевидно, що $w(x) \leq \left\lceil \frac{n-1}{2} \right\rceil = \left\lfloor \frac{n}{2} \right\rfloor$. Нехай T_0 – компонента в $T \setminus \{x\}$ така, що $|V(T_0)| = w(u)$. Тоді її доповнення $T \setminus T_0$ має $n - w(u) \geq n - \left\lfloor \frac{n}{2} \right\rfloor = \left\lceil \frac{n}{2} \right\rceil$ вершин. Зафіксуємо піддерево $T' \subset T \setminus T_0$ із рівно $\left\lceil \frac{n}{2} \right\rceil$ вершинами – вони будуть занумеровані непарними числами. Кожна вершина $x \in V(T \setminus T_0) \setminus V(T')$ суміжна із не більше, ніж з однією вершиною з T' . Залежно від того, чи $u \in V(T')$, матимемо оцінку:

$$\sigma^-(T) \leq n - w(u) - \left\lceil \frac{n}{2} \right\rceil + 1 = \left\lfloor \frac{n}{2} \right\rfloor - w(u) + 1.$$

□

Зауваження 2.20. Ця оцінка точна для знакової зірки $K_{1,n}$ і для шляху P_n . Тоді вага буде $w(u) = 1$ та $w(u) = \left\lfloor \frac{n-1}{2} \right\rfloor = \left\lfloor \frac{n}{2} \right\rfloor$ відповідно.

3 Програмна реалізація алгоритму на Python

3.1 Основні функції

- Імпортуємо необхідні модулі Python.

```

1 import networkx as nx
2 from networkx import NetworkXError
3 from collections import deque
4 import matplotlib.pyplot as plt
5

```

- Визначаємо функцію **draw_signed_graph(G)**, яка зображує даний парно-знаковий граф G , використовуючи бібліотеку networkx. Знаки усіх ребер надані, їх правильність на цьому етапі ще не перевіряється.

```

1 def draw_signed_graph(G):
2     H = nx.Graph()
3     H.add_nodes_from(G.nodes())
4
5     for u, v, label in G.edges(data='label'):
6         edge_style = 'solid' if label=="+" else 'dashed'
7         H.add_edge(u, v, style=edge_style)
8
9     pos = nx.spring_layout(H)
10    edge_styles = [H[u][v]['style'] for u, v in H.edges()]
11    edge_colors = ["r" for u, v in H.edges()]
12    nx.draw(H, pos, with_labels=True, node_color='w',
13            edge_color=edge_colors, style=edge_styles, width=2, font_size=16,
14            font_weight='bold')
15
16    plt.axis('off')
17    plt.show()
18

```

- Визначаємо функцію **get_positive_edges(G)**, яка формує список позитивних ребер з наданого знакового графа G .

```

1 def get_positive_edges(G):
2     positive_edges = []
3     for u, v, label in G.edges(data='label'):
4         if label == '+':
5             positive_edges.append((u, v))
6
7     return positive_edges
8

```

- Визначаємо функцію **contract_positive_edges(G)**, яка стягує позитивні ребра наданого графа.

```

1 def contract_positive_edges(G):
2     positive_edges = get_positive_edges(G)
3
4     try:
5         G = contract_edges(G, positive_edges)
6     except (KeyError, ValueError, NetworkXError):
7         return
8
9     return G
10

```

Функція `contract_edges(G, edges)` стягує надані ребра графа, використовуючи вбудовану функцію `contracted_nodes()`.

```

1 def contract_edges(G, edges):
2     H = G.copy()
3     for edge in edges:
4         u, v = edge
5         H = nx.contracted_nodes(H, u, v, self_loops=False)
6
7     return H
8

```

- Визначаємо функцію `get_vertices_labeling(G)`, яка згідно з алгоритмом стягує кожен позитивний компоненту в точку та присвоює їй мітку, яка є потужністю цієї компоненти.

```

1 def get_vertices_labeling(G):
2     """
3     This function contracts each positive section to a vertex and
4     labels the new vertex with the order of that section
5     """
6     positive_edges = get_positive_edges(G)
7
8     label_dict = {}
9     for node in G.nodes():
10        label_dict[node] = 1 # assign label 1 to every vertex initially
11
12    vertices_to_delete = []
13
14    H = G.copy()
15    for edge in positive_edges:
16        u, v = edge
17        H = nx.contracted_nodes(H, u, v, self_loops=False)
18        label_dict[u] += 1 # add one to the vertex u if a u-v edge was
19        contracted
20        vertices_to_delete.append(v) # after u-v edge contraction, the
21        vertex v is being removed
22
23    # remove vertices that were contracted
24    label_dict = {key: value for key, value in label_dict.items() if
25        key not in vertices_to_delete}

```

```

23     return label_dict
24

```

- Визначаємо функцію `is_bipartite(G)`, яка перевіряє наданий граф на двочастковість та в разі ствердної відповіді повертає список вершин в кожній частці.

```

1  def is_bipartite(G):
2      """
3      This function returns two lists of partition nodes if G is
4      bipartite
5      """
6      color = {}
7      for v in G:
8          color[v] = None
9
10     partition_1 = []
11     partition_2 = []
12
13     for v in G:
14         if color[v] is None:
15             color[v] = 'white'
16             q = deque([v])
17             partition_1.append(v)
18             while q:
19                 u = q.popleft()
20                 for neighbor in G[u]:
21                     if color[neighbor] is None:
22                         if color[u] == 'white':
23                             color[neighbor] = 'black'
24                             partition_2.append(neighbor)
25                         else:
26                             color[neighbor] = 'white'
27                             partition_1.append(neighbor)
28                             q.append(neighbor)
29                     # if there is an odd cycle, graph is not bipartite
30                     elif color[neighbor] == color[u]:
31                         return
32     return partition_1, partition_2

```

- Визначаємо основну функцію `is_parity_signed_graph(graph)`, яка представляє сам алгоритм та використовує всі функції перелічені вище. На вхід подається знаковий граф, на вихід отримуємо зображення цього графа та значення True або False.

```

1  def is_parity_signed_graph(graph):
2      """
3      This function checks if a given signed graph is a PSG.
4      1) contracts all "+" edges;
5      2) checks if graph is bipartite;

```

```
6 3) finds the two vertex classes and sums up the vertex labels in
7 each class.
8 If the sums differ by more than 1, return False. Otherwise, return
9 True.
10 """
11 G = graph.copy()
12 draw_signed_graph(G)
13
14 G = contract_positive_edges(G)
15 if not G: return False
16
17 partitions = is_bipartite(G)
18
19 if partitions:
20     partition_1, partition_2 = partitions
21     label_dict = get_vertices_labeling(graph)
22
23     total_sum_1 = 0
24     for key in partition_1:
25         total_sum_1 += label_dict[key]
26
27     total_sum_2 = 0
28     for key in partition_2:
29         total_sum_2 += label_dict[key]
30
31     return abs(total_sum_1 - total_sum_2) <= 1
32
33 return False
```

3.2 Застосування алгоритму на прикладах

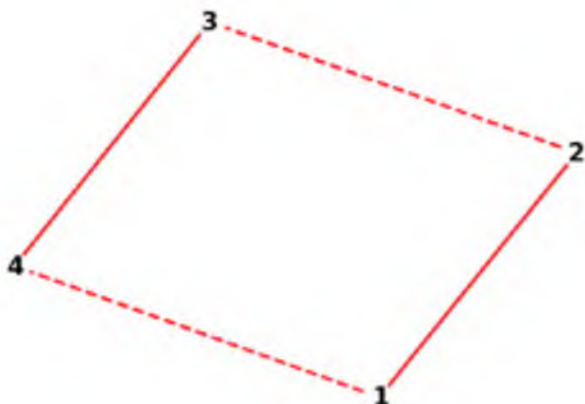
Приклад 3.1. Перевірити, чи знаковий граф G_1 є PSG.

```

1 G1 = nx.Graph()
2 edges1 = [(1, 2, {"label": "+"}),
3           (2, 3, {"label": "-"}),
4           (3, 4, {"label": "+"}),
5           (4, 1, {"label": "-"})]
6 G1.add_edges_from(edges1)
7
8 if is_parity_signed_graph(G1):
9     print("G1 is a PSG")
10 else: print("G1 is not a PSG")
11

```

Отримуємо такий результат: G_1 є PSG.



G1 is a PSG

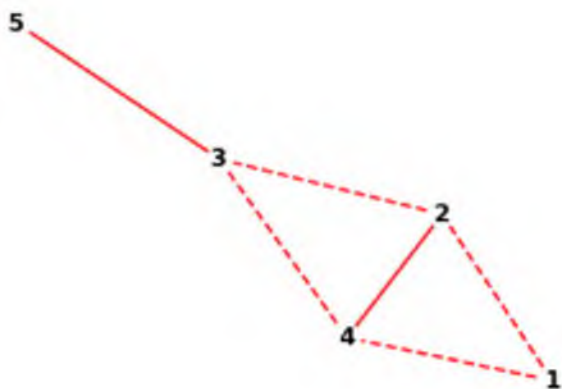
Приклад 3.2. Перевірити, чи знаковий граф G_2 є PSG.

```

1 G2 = nx.Graph()
2 edges2 = [(1, 2, {"label": "-"}),
3           (2, 3, {"label": "-"}),
4           (3, 4, {"label": "-"}),
5           (4, 1, {"label": "-"}),
6           (2, 4, {"label": "+"}),
7           (3, 5, {"label": "+"})]
8 G2.add_edges_from(edges2)
9
10 if is_parity_signed_graph(G2):
11     print("G2 is a PSG")
12 else: print("G2 is not a PSG")
13

```

Отримуємо такий результат: G_2 є PSG.



G2 is a PSG

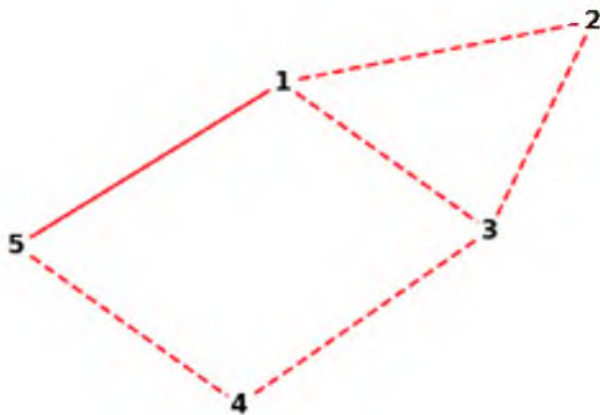
Приклад 3.3. Перевірити, чи знаковий граф G_3 є PSG.

```

1 G3 = nx.Graph()
2 edges3 = [(1, 2, {"label": "-"}),
3           (2, 3, {"label": "-"}),
4           (3, 1, {"label": "-"}),
5           (1, 5, {"label": "+"}),
6           (3, 4, {"label": "-"}),
7           (4, 5, {"label": "-"})]
8 G3.add_edges_from(edges3)
9
10 if is_parity_signed_graph(G3):
11     print("G2 is a PSG")
12 else: print("G3 is not a PSG")
13

```

Отримуємо такий результат: G_3 не є PSG.



G3 is not a PSG

Висновки

У роботі проведено дослідження основних властивостей парно-знакових графів, побудована верхня оцінка на ‘гпа’ для дерев, а також реалізований алгоритм перевірки знакового графа на те, чи є він парно-знаковим. Мета роботи досягнута.

У **Розділі 1** розглянуто базові означення, які використовуються надалі в роботі. Зокрема вводиться поняття парно-знакового графа, позитивних та негативних ребер та компонент.

У **Розділі 2** наведено властивості парно-знакових графів, а також основні результати.

У **Підрозділі 2.1** розглянуто таку властивість знакових графів як збалансованість. Також наведена теорема Харарі про характеризування збалансованих знакових графів.

У **Підрозділі 2.2** наводиться опис алгоритму для визначення, чи є заданий парно-знаковий граф PSG. Також для наочності представлений приклад з ілюстрованим описом кожного кроку алгоритму.

У **Підрозділі 2.3** розглянуто число ‘гпа’ графа G . Доведене твердження про число ‘гпа’ для знакової зірки $K_{1,n}$, а також теорема про необхідні та достатні умови, щоб $\sigma^-(G) = 1$.

У **Підрозділі 2.4** представлений основний результат – верхня оцінка на ‘гпа’ для дерев. Використовуючи поняття ваги вершини $w(u)$ та центроїда дерева $\{u\}$, визначено, що число ‘гпа’ дерева не перевищує наступного виразу: $\left\lfloor \frac{n}{2} \right\rfloor - w(u) + 1$.

У **Розділі 3** представлена програмна реалізація алгоритму перевірки графа на парно-знаковість на мові Python. Також надано приклади застосування отриманої функції.

Література

- [1] Mukti Acharya, Joseph Varghese Kureethara, *Parity labeling in signed graphs*, <https://arxiv.org/pdf/2012.07737>, (2020), 10.
- [2] Mukti Acharya, Joseph Varghese Kureethara, Thomas Zaslavsky, *Characterizations of Some Parity Signed Graphs*, Australasian journal of combinatorics Volume 81(1), (2021), 89–100.
- [3] F. Harary: On the Notion of Balance of a Signed Graph, The Michigan Math. J., 2(1953), 143–146.
- [4] F. Harary, J.A. Kabell, *A simple algorithm to detect balance in signed graphs*, Mathematical Societal Sciences, (1980), 1-6.
- [5] Ligang Jin, Xiaoyue Chen, Yingli Kang, *A study on parity signed graphs: the rna number*, <https://arxiv.org/abs/2111.04956>, (2021), 1-5.