

Міністерство освіти і науки України
Національний університет «Києво-Могилянська академія»
Факультет інформатики
Кафедра математики

Кваліфікаційна робота
освітній ступінь – бакалавр

на тему: «**Потоки в мережі. Теорема Форда-Фалкерсона**»

Виконала: студентка 4-го року
навчання
освітньої програми «Прикладна
математика»,
спеціальності 113 Прикладна
математика

Вербівська Юлія Віталіївна

Керівник: Тимошкевич Л. М.
кандидат фіз.-мат. наук, ст. викладач

Рецензент:

Кваліфікаційна робота захищена
з оцінкою _____

Секретар ЕК _____
(підпис)

« _____ » _____ 20__ р.

Міністерство освіти і науки України
Національний університет «Києво-Могилянська академія»
Факультет інформатики
Кафедра математики

ЗАТВЕРДЖУЮ
Зав.кафедри математики,
проф., доктор фіз.-мат. наук
_____ Чорней Р.К.
(підпис)
“ _____ ” _____ 2024

ІНДИВІДУАЛЬНЕ ЗАВДАННЯ
для кваліфікаційної роботи
студенту 4-го курсу, факультету інформатики
Вербівська Юлія Віталіївна

Тема: «Потоки в мережі. Теорема Форда-Фалкерсона»

Зміст кваліфікаційної роботи:

Анотація

Вступ

1. Теорія потоків у мережах. Теорема Форда-Фалкерсона

2. Алгоритм Форда-Фалкерсона

3. Застосування теореми Форда-Фалкерсона до доведення мінімаксних теорем

Висновки

Список літератури

Дата видачі “ _____ ” _____ 2024 Керівник _____
(підпис)

Завдання отримав _____
(підпис)

Графік підготовки кваліфікаційної роботи до захисту

Графік узгоджено « _____ » _____ 2024р.

№ з/п	Перелік робіт	Термін виконання етапу	Підпис наукового керівника	Дата ознайомлення наукового керівника	Примітка
1.	Отримання теми кваліфікаційної роботи.	29.10.2023			
2.	Ознайомлення з темою кваліфікаційної роботи.	5.11.2023			
3.	Розробка плану та структури роботи.	30.11.2023			
4.	Робота з науковою літературою, опис основних означень.	22.01.2024			
5.	Дослідження результатів отриманих в літературі.	22.02.2024			
6.	Робота над текстовим оформленням результатів.	22.04.2024			
7.	Попередній аналіз кваліфікаційної роботи. Виправлення помилок.	30.04.2024			
8.	Попередній захист кваліфікаційної роботи.	15.05.2024			
9.	Захист кваліфікаційної роботи.	04.06.2024			

Науковий керівник _____
(ПІБ)

Виконавець кваліфікаційної роботи _____
(ПІБ)

Зміст

Анотація	3
Вступ	4
Актуальність	4
Мета, завдання дослідження	5
1 Теорія потоків у мережах. Теорема Форда-Фалкерсона	6
1.1 Основні означення теорії графів	6
1.2 Основні означення теорії потоків у мережі	8
1.3 Теорема Форда-Фалкерсона	11
2 Алгоритм Форда-Фалкерсона	13
2.1 Опис алгоритму	13
2.2 Приклад використання алгоритму Форда-Фалкерсона для по- будови максимального потоку в мережі	14
2.3 Аналіз алгоритму	17
Часова складність алгоритму	17
Можливе зациклювання алгоритму при ірраціональних пропу- скних спроможностях	18
Неоптимальна робота алгоритму на мережі малої потужності	20
Парні та непарні пропускі спроможності	21
Приклад застосування алгоритму Форда-Фалкерсона	21
3 Застосування теореми Форда-Фалкерсона до доведення міні- максних теорем теорії графів	23
3.1 Доведення леми Холла	23
3.2 Доведення теореми Кьоніга	26
3.3 Доведення теореми Менгера	30
Висновки	35
Література	36

Анотація

Метою даного дослідження є детальне вивчення та аналіз теорії потоків у мережі, зокрема теореми Форда-Фалкерсона та алгоритма Форда-Фалкерсона.

У роботі було розглянуто теорему Форда-Фалкерсона та її застосування. Також було розглянуто алгоритм Форда-Фалкерсона. Крім того було наведено приклад його застосування до деякої мережі та проведено аналіз його часової складності, оптимальності та роботи за різних умов.

Вступ

Актуальність

Теорія потоків у мережі — це розділ теорії графів, що досліджує ефективну передачу та оптимізацію ресурсів або інформації через мережеві структури. Ця галузь займається моделюванням та аналізом різних систем, включаючи транспортні мережі, мережі зв'язку та розподіл енергії, використовуючи такі поняття, як максимальний потік, пропускна спроможність та мінімальний розріз. У сучасному контексті теорія потоків у мережі є надзвичайно актуальною через стрімке зростання обсягів даних та розвиток технологій, що вимагають якісної оптимізації.

Теорема Форда-Фалкерсона про максимальний потік вважається визначною математичною віхою, з якою пов'язують початок теорії потоків. Уперше вона була сформульована L. R. Ford та D. R. Fulkerson у 1956 році у їхній фундаментальній роботі "Maximal Flow Through A Network"[1]. Як зазначають у своїй роботі автори, задача про максимальний потік уперше була сформульована Т. Е. Harris наступним чином:

Розглянемо залізничну мережу, що з'єднує два міста через ряд проміжних міст, де кожна ланка мережі має номер, що відповідає її пропускній спроможності. Припускаючи, що мережа перебуває у стаціонарному стані, знайдіть максимальний потік з одного міста до іншого.

Згідно зі статтю A. Schrijver "On the history of the transportation and maximum flow problems"[2] Т. Е. Harris, спільно з генералом F. S. Ross, у той час працювали над моделлю залізничного транспортного потоку, що виникла в контексті радянської залізничної системи. Запитання було спрямоване на знаходження максимального потоку в залізничній мережі, що пов'язувала Радянський Союз та країни-супутники. Окрім того, Harris та Ross визначили ще одну задачу, а саме пошук мінімізації перекриття радянської залізничної системи з метою ускладнення її функціонування в умовах війни. Це дослідження було виконане для Військово-повітряних сил США та було класифіковане як секретне аж до 1999 року.

Теорема Форда-Фалкерсона знайшла застосування у багатьох різних сферах і залишається актуальною. Принципи теореми застосовуються, як для ефективного керування транспортними системами і оптимального розподілу ресурсів, так і для розподілу та управління електроенергією, забезпечення оптимального потоку даних для забезпечення високоякісного зв'язку та інше.

Мета, завдання дослідження

Метою даного дослідження є детальне вивчення та аналіз теорії потоків у мережі, зокрема теореми Форда-Фалкерсона та алгоритма Форда-Фалкерсона. Головним завданням роботи є дослідження теореми Форда-Фалкерсона та її застосування для розв'язання задач та доведення теорем теорії графів, а також аналіз алгоритму Форда-Фалкерсона та його застосування.

Завдання роботи включають:

- Вивчення основних означень та понять теорії графів, що необхідні для розуміння подальшої теорії.
- Вивчення основних означень теорії потоків у мережі, зокрема, мережа, пропускна спроможність ребра та ін.
- Огляд теореми Форда-Фалкерсона та її застосування для доведення мінімальних теорем теорії графів.
- Огляд та аналіз алгоритму Форда-Фалкерсона для знаходження максимального потоку у мережі.

Матеріали даного дослідження також можна застосовувати для підготовки на математичних семінарах, факультативних заняттях або для підготовки до олімпіад.

1 Теорія потоків у мережах. Теорема Форда-Фалкерсона

1.1 Основні означення теорії графів

Для належного розуміння подальшої теорії введемо деякі основні означення з теорії графів.

Нехай V — непорожня скінченна множина, а $V^{(2)}$ — множина всіх двоелементних підмножин (непорядкованих пар різних елементів) множини V .

Означення 1.1.1. Графом G називають пару множин (V, E) , де E — довільна підмножина множини $V^{(2)}$ ($E \subseteq V^{(2)}$); позначають $G = (V, E)$.

Елементи множини V називають **вершинами** графа G , а елементи множини E — його **ребрами**. Відповідно V називають **множиною вершин**, а E — **множиною ребер** графа G .

Надалі в роботі використовуватимемо також і наступні позначення:

- $V(G)$ — множина вершин V графа G .
- $E(G)$ — множина ребер E графа G .
- (v_1, v_2) — ребро, що з'єднує вершини v_1 та v_2 .

Нехай задано граф $G = (V, E)$. Якщо $(v_1, v_2) \in E$, то кажуть, що вершини v_1 і v_2 **суміжні**, інакше вершини v_1 і v_2 **несуміжні**. Два ребра називають **суміжними**, якщо вони мають спільну вершину.

Якщо $e = (v_1, v_2)$ — ребро графа, то вершини v_1 і v_2 називають **кінцями** ребра e . У цьому разі кажуть, що ребро e з'єднує вершини v_1 і v_2 .

Вершину v і ребро e називають **інцидентними**, якщо v — кінець ребра e .

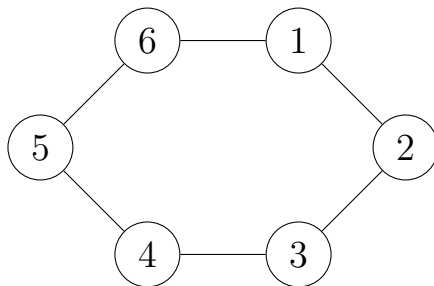


Рис. 1: Граф G

Означення 1.1.2. **Орієнтованим графом (орграфом)** G називають пару множин (V, E) , де $E \subseteq V \times V$. Елементи множини V називають **вершинами**, а елементи множини E — **дугами** орграфа $G = (V, E)$. Таким чином, дуга — впорядкована пара вершин; V — **множина вершин**, а E — **множина дуг** орграфа G .

Якщо $e = (v_1, v_2)$ — дуга, то вершину v_1 називають **початком**, а вершину v_2 — **кінцем** дуги e . Кажуть, що дуга e **веде** із вершини v_1 у вершину v_2 , або виходить із v_1 і заходить у v_2 . Дугу e та вершини v_1 і v_2 називають **інцидентними** між собою, а вершини v_1 і v_2 — **суміжними**.

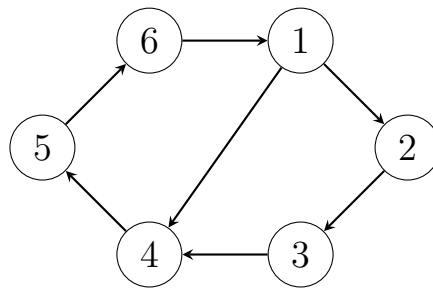


Рис. 2: Орієнтований граф G

Означення 1.1.3. **Мультиграфом** називають граф, у якому допускається наявність **кратних ребер**, тобто дві вершини можуть бути з'єднані більш ніж одним ребром.

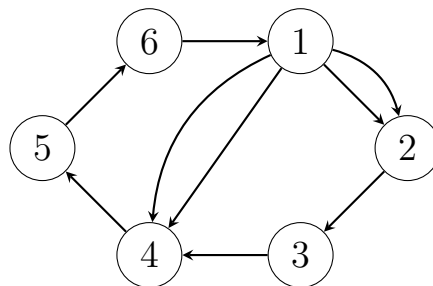


Рис. 3: Мультиграф

Означення 1.1.4. **Шляхом, або маршрутом**, в орграфі $G = (V, E)$ називають таку послідовність його вершин і дуг

$$v_1, e_1, v_2, e_2, \dots, e_k, v_{k+1}, \quad (1.1.1)$$

що $e_i = (v_i, v_{i+1})$, $i = 1, 2, \dots, k$. Кажуть, що цей шлях **веде** з вершини v_1 у вершину v_{k+1} . Кількість k дуг у шляху (1.1.1) називають його довжиною.

Якщо існує шлях, який веде з вершини v_1 у вершину v_2 , то кажуть, що вершина v_1 **досяжна** з вершини v_2 .

Означення 1.1.5. Орграф називають **сильно зв'язним**, якщо будь-які дві його вершини досяжні одна з одною.

Орграф називають **однобічно зв'язним**, якщо для будь-яких двох його вершин принаймні одна досяжна з іншою.

1.2 Основні означення теорії потоків у мережі

Означення 1.2.1. Нехай задано множину вершин V , у якій виділено дві вершини: s (*вхід або джерело*) і t (*вихід або стік*), інші вершини називатимемо *проміжними вузлами*.

Нехай визначена функція $c : V \times V \rightarrow R$, яка задовольняє співвідношенням

$$c(x, y) \geq 0, \quad (1.2.1)$$

$$c(x, s) = 0, \quad (1.2.2)$$

$$c(t, y) = 0 \quad (1.2.3)$$

для будь-яких вершин $x, y \in V$. Тоді $G = (V, s, t, c)$ — **мережа**, функція $c(s, t)$ називається **пропускною спроможністю мережі G** , а функція $c(x, y)$ — **пропускною спроможністю ребра (x, y)** .

Множина $A(G) = \{(x, y) : c(x, y) > 0\}$ називається **множиною стрілок мережі G** .

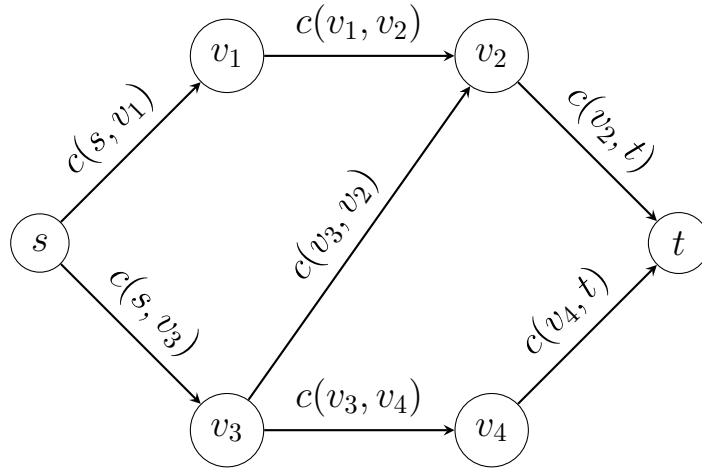


Рис. 4: Мережа G

Означення 1.2.2. Нехай G — мережа, а функція $f : V \times V \rightarrow R$ задовольняє наступним умовам:

(F1) для будь-яких $x, y \in V$ $f(x, y) \leq c(x, y)$;

(F2) для будь-яких $x, y \in V$ $f(x, y) = -f(y, x)$;

(F3) для будь-якої вершини $v \in V$, $v \neq s, t$ виконується умова

$$\sum_{x \in V} f(v, x) = 0 \quad (1.2.4)$$

Тоді f — **потік** в мережі G . Число $|f| = \sum_{x \in V} f(s, x)$ називають **величиною потоку**. Поток мережі G з максимальною величиною називається **максимальним**.

Означення 1.2.3. Нехай G — мережа, а множина її вершин V розбита на дві множини, що не перетинаються, $S \ni s, T \ni t$. Тоді (S, T) — **розріз** мережі G .

Величина

$$c(S, T) = \sum_{x \in S, y \in T} c(x, y) \quad (1.2.5)$$

називається **пропускною спроможністю розрізу**. Іншими словами, пропускна спроможність розрізу — це сума пропускних спроможностей усіх ребер, які ведуть з S в T .

Будь-який розріз мережі G з мінімальною пропускною спроможністю називається **мінімальним**.

Для будь-якого потоку f в мережі G величина

$$f(S, T) = \sum_{x \in S, y \in T} f(x, y) \quad (1.2.6)$$

називається **поток**ом **через розріз** (S, T) . Тобто потік через розріз (S, T) — це сума потоків з S в T .

Зауваження 1.2.1. Очевидно, що мінімальний розріз мережі існує. Можливо, таких розрізів декілька.

Лема 1.2.1. Для будь-якого потоку f і розрізу (S, T) мережі G виконується $|f| = f(S, T)$.

Доведення. Враховуючи умову (F3)

$$|f| = \sum_{x \in V} f(s, x) = \sum_{y \in S} \left(\sum_{x \in V} f(y, x) \right) \quad (1.2.7)$$

У правій частині рівності для будь-яких двох вершин $y, z \in S$ присутні доданки $f(y, z)$ і $f(z, y)$, а враховуючи умову (F2) сума цих доданків буде рівна 0.

Тому,

$$\sum_{y \in S} \left(\sum_{x \in V} f(y, x) \right) = \sum_{x \in S, y \in T} f(x, y) = f(S, T),$$

що і треба було довести. \square

Означення 1.2.4. **Залишкова пропускна спроможність** ребра $c_f(x, y) = c(x, y) - f(x, y)$. Вона завжди невід'ємна.

Означення 1.2.5. Нехай f — потік в мережі G . Розглянемо мережу G_f з однаковими V, s, t і пропускною спроможністю

$$c_f(x, y) = \begin{cases} 0 & \text{при } y = s \text{ або } x = t, \\ c(x, y) - f(x, y) & \text{в інших випадках.} \end{cases} \quad (1.2.8)$$

Називатимемо G_f **залишковою мережею** потоку f .

Доповнюючий шлях потоку f — це будь-який st -шлях (шлях від джерела s до стоку t) в залишковій мережі G_f .

Лема 1.2.2. Нехай f — потік у мережі G , f' — потік в залишковій мережі G_f . Тоді $f + f'$ — потік у мережі G , при чому $|f + f'| = |f| + |f'|$.

Доведення. Перевіримо для потоку $f + f'$ умови (F1), (F2) та (F3). Звідси очевидне твердження про величину потоку.

$$\text{F1. } (f + f')(x, y) = -(f + f')(x, y)$$

$$\text{F2. } (f + f')(x, y) \leq c(x, y) \iff f(x, y) + f'(x, y) \leq c(x, y) \iff f'(x, y) \leq c(x, y) - f(x, y)$$

$$\text{F3. } \sum_{y \in V} (f + f')(x, y) = 0, x \neq s, x \neq t$$

Означення 1.2.6. Нехай P — st -шлях у мережі G , а c — мінімальна пропускна спроможність стрілки шляху P . Визначимо потік f_P вздовж шляху P : $f_P(x, y) = c$ при $xy \in A(P)$, $f_P(x, y) = -c$ при $yx \in A(P)$, $f_P(x, y) = 0$ при $xy, yx \notin A(P)$.

1.3 Теорема Форда-Фалкерсона

Теорема 1.3.1. (L. R. Ford, D. R. Fulkerson, 1956.) У мережі $G = (V, s, t, c)$ заданий потік f . Тоді три наступні твердження є рівносильним:

1. Потік f максимальний.
2. Існує такий розріз (S, T) , що $|f| = c(S, T)$;
3. В залишковій мережі G_f немає доповнюючого шляху.

Іншими словами, величина максимального потоку із джерела s у стік t дорівнює пропускній спроможності мінімального розрізу, що відокремлює s від t .

Доведення.

$1 \Rightarrow 3$. Припустимо протилежне, нехай в залишковій мережі G_f є доповнюючий шлях P , а f_P — потік уздовж шляху P . Тоді за леммою 1.2.2 $f + f_P$ — потік у G , при чому $|f + f_P| = |f| + |f_P| > |f|$, що має протиріччя з максимальністю f .

2 \Rightarrow 1. Розглянемо інший потік f' . За умовою (F1) і леммою 1.2.1 маємо наступне:

$$|f'| = f'(S, T) = \sum_{x \in S, y \in T} f(x, y) \leq \sum_{x \in S, y \in T} c(x, y) = c(S, T) = |f|,$$

звідки й слідує максимальність f .

3 \Rightarrow 2. Нехай S — множина усіх вершин, досяжних з s у залишковій мережі G_f . Оскільки в G_f немає доповнюючого шляху, то $t \notin S$. Тоді $(S, T = V \setminus S)$ — розріз у мережах G та G_f . За побудовою $A_{G_f}(S, T) = \emptyset$, $s \in S$ і $t \in T$, відповідно, $0 = c_f(S, T) - f(S, T)$, звідки $c(S, T) = f(S, T)$. \square

Наслідок 1.3.1. Величина максимального потоку в мережі G рівна пропускній спроможності мінімального розрізу мережі G .

Доведення. Розглянемо максимальний потік f і такий розріз (S, T) , що $|f| = c(S, T)$ (такий розріз існує за теоремою 1.3.1). Тоді для будь-якого іншого розрізу (S', T') маємо $c(S', T') \geq f(S', T') = f(S, T) = c(S, T)$. Тоді маємо, що розріз (S, T) є мінімальним. \square

У наступній теоремі розглядаємо мережу G , що має цілочисельну пропускну спроможність c .

Теорема 1.3.2. У цілочисельній мережі G існує максимальний потік, при чому серед максимальних потоків даної мережі знайдеться цілочисельний.

Доведення. Будемо послідовно будувати потік. Спочатку покладемо $f = 0$. Нехай у деякий момент є цілочисельний потік f в цілочисельній мережі G . Розглянемо залишкову мережу G_f . Якщо у G_f немає доповнюючого шляху P , тоді за теоремою 1.3.1 потік f — максимальний.

Якщо ж у G_f є доповнюючий шлях P , то розглянемо потік f_P уздовж шляху P в залишковій мережі G_f і покладемо $f := f + f_P$. Отримали новий цілочисельний потік, при чому його пропускна спроможність на $|f_P|$ більша, аніж у попереднього. Так як з кожним кроком величина потоку зростає на цілу величину, процес обов'язково закінчиться, а в результаті ми отримаємо цілочисельний максимальний потік в мережі G . \square

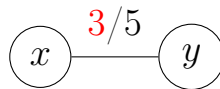
2 Алгоритм Форда-Фалкерсона

2.1 Опис алгоритму

Алгоритм або метод Форда-Фалкерсона, запропонований L. R. Ford та D. R. Fulkerson (1956), - це жадібний алгоритм, що обчислює максимальний потік у мережі.

Класичне формулювання задачі передбачає деяку мережу водопостачання, у якій наявні одне джерело витoku води, до якого приєднано систему труб з різною пропускною здатністю. Уся вода з цієї системи збирається ("стікає") в одному стоку. Завдання полягає у визначенні максимального можливого потоку води через цю мережу труб.

Надалі позначатимемо потік $|f|$ та його пропускну спроможність $c(x, y)$ на графі, як "**потік / пропускна спроможність**". Наприклад, якщо матимемо ребро (x, y) з пропускною спроможністю 5 та потоком 3, зображуватимемо його наступним чином:



Ідея алгоритму Форда-Фалкерсона полягає у наступному. Обирається шлях із джерела до стоку такий, щоб для кожного ребра залишкова пропускна спроможність була строго більшою за нуль. При цьому ребра на даному шляху можуть проходити як у прямому, так і в зворотному напрямках. Серед залишкових пропускних спроможностей ребер даного шляху обирається мінімальне значення. На це значення збільшується потік на кожному з ребер даного шляху. Далі шукається наступний аналогічний шлях. Робота алгоритму продовжується доти, доки такі шляхи можливо знайти.

Важливо зазначити, що даний алгоритм відноситься до класу **недетермінованих**, тобто кожен наступний крок алгоритму визначається неоднозначно. Час роботи алгоритму, тобто кількість кроків, які він виконає, також залежить від того, як саме будуть обиратися кроки.

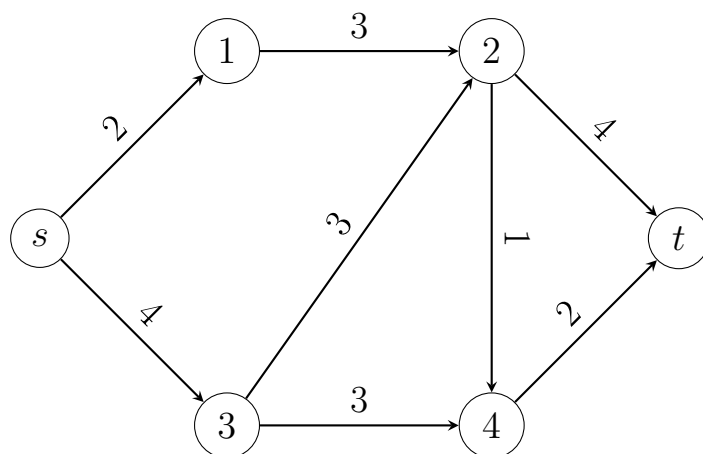
Алгоритм Форда-Фалкерсона:

1. Ініціалізуємо початковий потік мережею, як нульовий:
 $\forall e(x, y) \in E f(x, y) = 0$.
2. Шукаємо у залишковій мережі будь-який шлях з джерела s до стоку t , для якого його дуги задовольняють наступну умову: $f(x, y) \leq c(x, y)$. У разі, якщо такого потоку не існує, то потік, що вже наявний у мережі і є максимальним;
3. Знайдений шлях будемо називати **збільшувальним шляхом**. Через цей шлях пускаємо максимально можливий на ньому потік;
4. Визначаємо на даному шляху ребро з найменшою пропускною здатністю на шляху, позначимо як c_{min} ;
5. Для кожного ребра на даному збільшувальному шляху збільшимо потік на c_{min} , а в протилежному напрямку — зменшимо на c_{min} ;
6. Оновимо залишкову мережу: обчислимо нову пропускну спроможність для усіх ребер на збільшувальному шляху, а також для протилежних їм ребер. Якщо нова пропускна спроможність не рівна нулю, додаємо ребро до залишкової мережі, а якщо дорівнює нулю — видаляємо.
7. Повертаємося на крок 2.

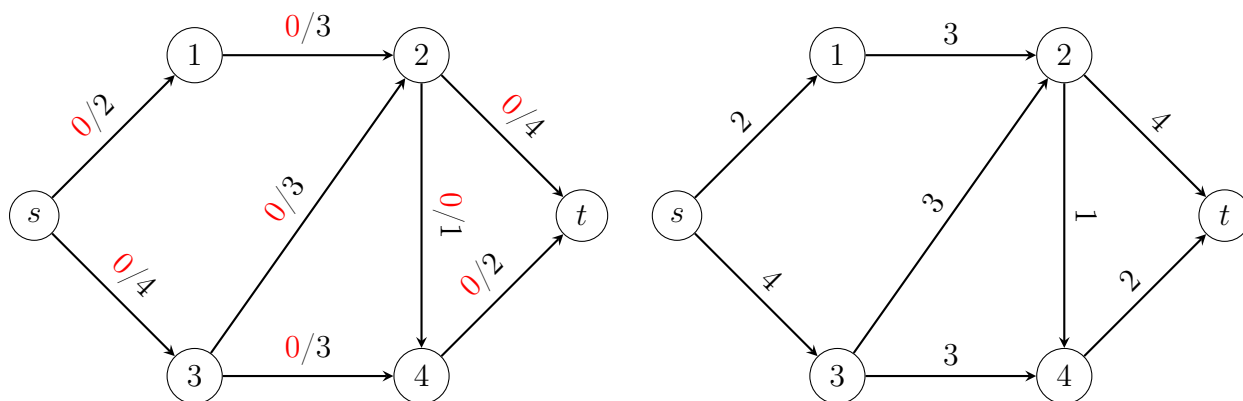
Варто зазначити, що алгоритм Форда-Фалкерсона не вказує точно, яким чином обирати шлях на кроці 2 або ж спосіб, за яким це робити. Через це, алгоритм гарантує збіжність лише для цілих значень пропускну спроможностей. Проте, навіть у випадку цілочисельних пропускну спроможностей, якщо значення пропускну спроможностей є великими, алгоритм може і не досягти оптимального рішення або буде працювати дуже довго.

2.2 Приклад використання алгоритму Форда-Фалкерсона для побудови максимального потоку в мережі

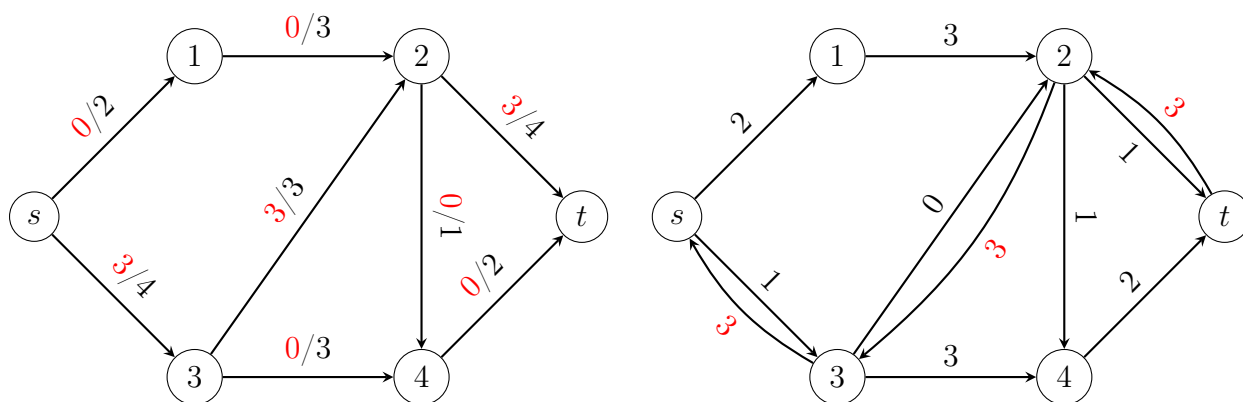
Нехай маємо наступну мережу з цілочисельними пропускну спроможностями вказаними на ребрах:



Спробуємо знайти максимальний потік за допомогою алгоритму Форда-Фалкерсона. Для початку, встановимо потужність потоку на ребрах рівну 0. Ліворуч бачимо потік (зараз $|f| = 0$), праворуч — залишкову мережу.

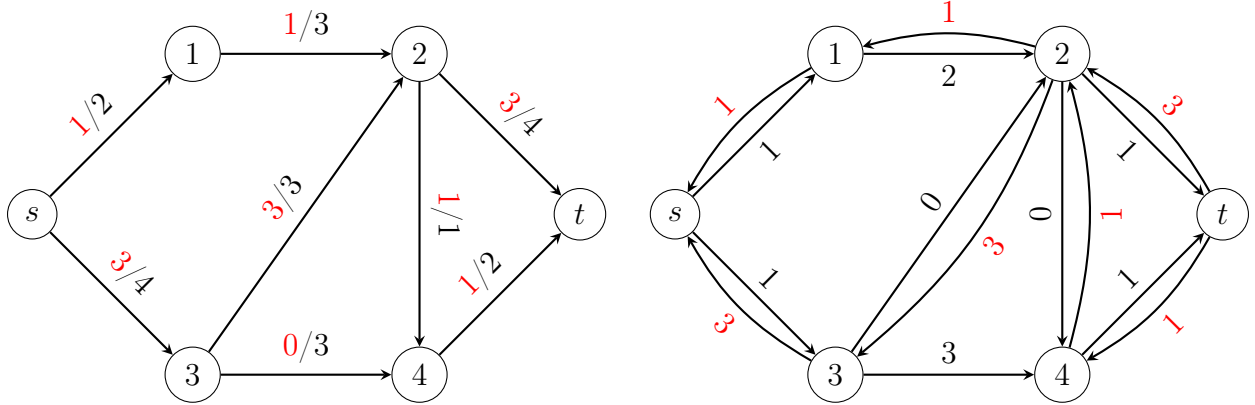


Пустимо 3 одиниці потоку шляхом $s - 3 - 2 - t$. Матимемо наступний потік і залишкову мережу.

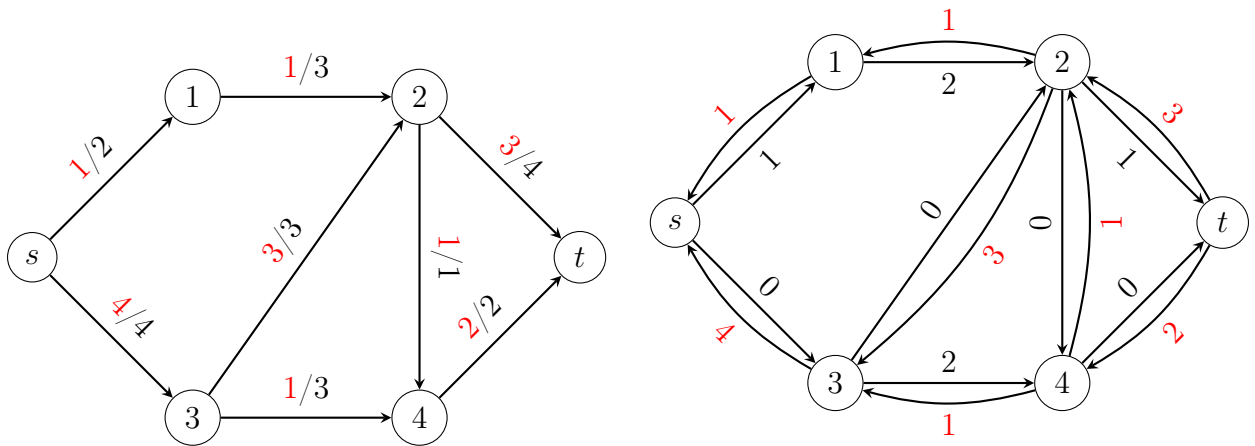


Продовжимо застосовувати алгоритм Форда-Фалкерсона. Бачимо доповнюючий шлях $s - 1 - 2 - 4 - t$. Мінімальна пропускна спроможність цього шляху складає 1 одиницю. Тоді пустимо 1 одиницю потоку цим шляхом.

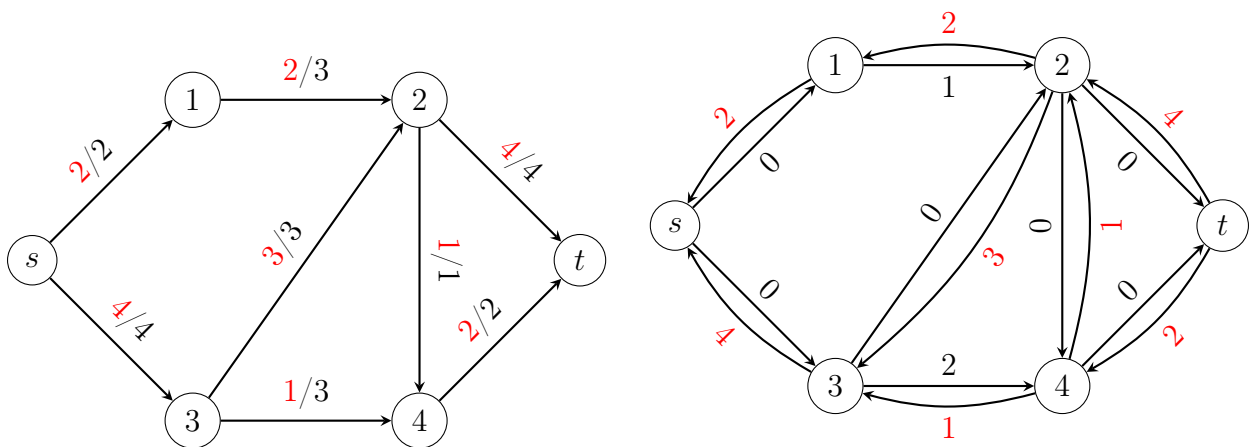
Отримаємо



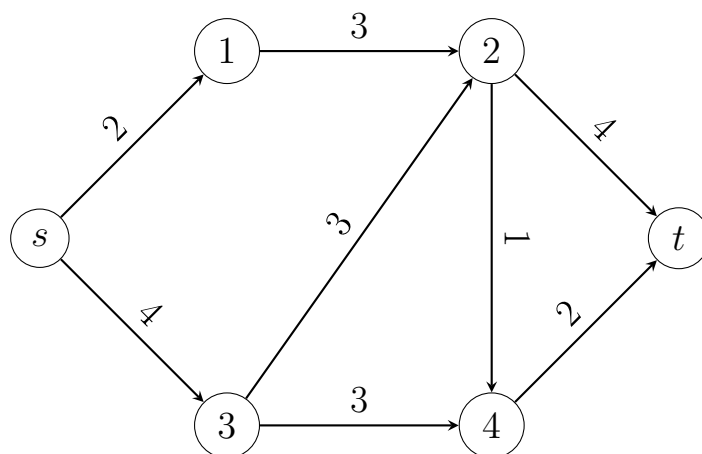
Бачимо, що є ще один можливий збільшувальний шлях, а саме $s-3-4-t$, що має пропускну спроможність 1. Пустимо цим шляхом 1 одиницю потоку.



Продовжуємо шукати збільшувальні шляхи. Бачимо шлях $s-1-2-t$ з пропускну спроможністю 1. Пустимо цим шляхом 1 одиницю потоку.



Як бачимо збільшувальних шляхів не лишилося, а отже ми маємо максимальний потік. Потужність максимального потоку вийшла рівна 6. Як



бачимо, на залишковій мережі не лишилося доступних шляхів з джерела s до стоку t .

Якщо ми поглянемо на початкову мережу.

Легко можемо побачити, що, дійсно, мінімальним розрізом є розріз пропускної спроможності 6. Таких розрізи можна зробити 2: через ребра $(s, 1) - (s, 3)$ та через ребра $(2, t) - (4, t)$.

2.3 Аналіз алгоритму

У цьому підрозділі проведемо аналіз алгоритму Форда-Фалкерсона.

Часова складність алгоритму

Означення 2.3.1. Часова складність алгоритму - це міра обчислювальних ресурсів (зазвичай часу), що потрібні для виконання алгоритму в залежності від розміру вхідних даних. Вона визначає, як швидко зростає час виконання алгоритму зі збільшенням розміру вхідних даних. Часова складність надає оцінку часу виконання у абстрактних одиницях, наприклад, у кількості простих операцій.

Зазвичай часова складність виражається у вигляді функції $T(n)$, де n - розмір вхідних даних. Щоб позначити оцінку часової складності алгоритму використовується символ O . Так, наприклад, якщо часова складність алгоритму складає $O(n^2)$, це означає, що час виконання зростає квадратично відносно розміру вхідних даних.

На часову складність алгоритму Форда-Фалкерсона значно впливає те,

яким чином було визначено збільшувальні шляхи. Зазвичай, для знаходження збільшувальних шляхів використовуються такі методи, як Breadth-first search (BFS) або Depth-first search (DFS), тобто алгоритм пошуку у ширину та пошуку у глибину.

Розглянемо часову складність алгоритму за умови, що всі пропускні спроможності ребер є **цілочисельними**, а також для пошуку збільшувальних шляхів використовується алгоритм Depth-first search. Нехай максимальний потік у мережі має потужність F , тоді алгоритм робить не більше ніж F ітерацій, адже на кожній з ітерацій ми додаємо не менше однієї одиниці потоку з s в t . Часова складність алгоритму DFS складає $O(V + E)$, де V — кількість вершин, а E — кількість ребер. Якщо припускаємо, що граф є простим, тобто між будь-якими двома вершинами не більше за одне ребро, то маємо $E \geq V - 1$, тобто вважаємо, що часова складність пошуку збільшувального шляху рівна $O(E)$.

Тоді, у найгіршому випадку, за умови цілочисельній пропускній спроможності стрілок, тобто у такому, що ми додаємо по 1 одиниці потоку у мережу, *часова складність алгоритму Форда-Фалкерсона рівна $O(E * F)$.*

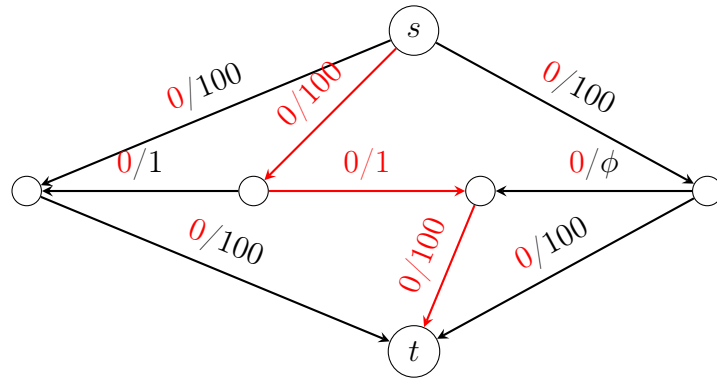
Таким чином, можемо сказати, що алгоритм досить добре працює у випадках, коли значення максимального потоку F є невеликим. Втім, без обмежень, що стосувалися б вибору збільшувальних шляхів, він не є ефективним у загальному випадку, а його часова складність може ставати **експоненційною**. У випадку ірраціональних пропускних спроможностей або у погано структурованих мережах алгоритм може обирати такі шляхи збільшення, що кожного разу до потоку буде додаватися дуже малий приріст, що у свою чергу може призвести до дуже великої кількості ітерацій, тобто до експоненційної складності.

Можливе зациклювання алгоритму при ірраціональних пропускних спроможностях

Якщо у мережі дозволені ірраціональні пропускні спроможності, алгоритм може зациклитися і не досягти максимального потоку, додаючи усе менший приріст до загального потоку на кожній ітерації. Таким чином, алгоритм може взагалі не сходитися до максимально можливого потоку у мережі або й

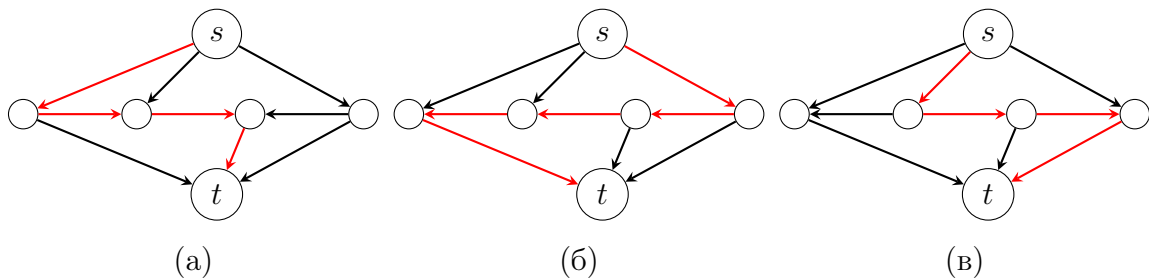
навіть до значної його частити. Приклад такого ефекту наводить у своїй роботі "The smallest networks on which the Ford-Fulkerson maximum flow procedure may fail to terminate" Uri Zwick [13]

Розглянемо наступну мережу, що складається з 6 вузлів. 6 з 9 ребер мають велику цілочисельну пропускну спроможність, 2 - пропускну спроможність рівну 1 і 1 ребро має пропускну спроможність $\phi = (\sqrt{5} - 1)/2 \approx 0.618034$, яку було обрано таким чином, щоб $1 - \phi = \phi^2$.



Припускаємо, що алгоритм спочатку обирає даний збільшувальний шлях (на рисунку його виділено червоними стрілками). У такому разі, 3 горизонтальні ребра, зліва на право, матимуть залишкові пропускні спроможності рівні 1, 0, φ, відповідно. Доведемо за індукцією, що залишкові пропускні спроможності для горизонтальних ребер становлять $\phi^{k-1}, 0, \phi$ для деякого невід'ємного цілого числа k .

Розглядаємо 3 наступні шляхи:



1. Збільшуємо потік шляхом (б), додаємо ϕ^k одиниць потоку. Тоді залишкові пропускні спроможності на горизонтальних шляхах — $\phi^{k+1}, \phi^k, 0$.
2. Збільшуємо потік шляхом (в), додаємо ϕ^k одиниць потоку. Залишкові пропускні спроможності на горизонтальних шляхах — $\phi^{k+1}, 0, \phi^k$.
3. Збільшуємо потік шляхом (б), додаємо ϕ^{k+1} одиниць потоку. Залишкові пропускні спроможності на горизонтальних шляхах — $0, \phi^{k+1}, \phi^{k+2}$.

4. Збільшуємо потік шляхом (а), додаємо ϕ^{k+1} одиниць потоку. Залишкові пропускні спроможності тепер рівні ϕ^{k+1} , 0, ϕ^{k+2} .

Отже, за індукцією випливає, що після $4n + 1$ кроків збільшення, горизонтальні ребра мають залишкові пропускні спроможності ϕ^{2n-2} , 0, ϕ^{2n-1} . Якби кількість ітерацій зростала до нескінченності, значення потоку наближалось б до

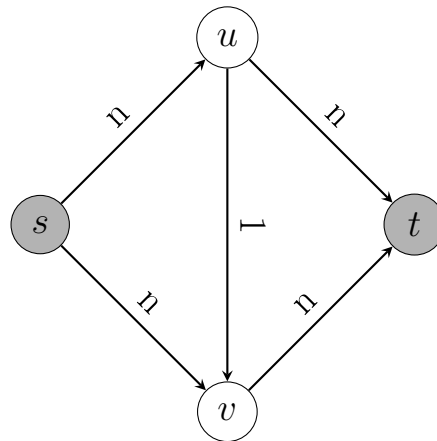
$$1 + 2 \sum_{i=1}^{\infty} \phi^i = 1 + \frac{2}{1 - \phi} = 4 + \sqrt{5} < 7,$$

хоча очевидно, що в нашому випадку максимальний потік рівний $2 * 100 + 1 = 201 > 7$.

Неоптимальна робота алгоритму на мережі малої потужності

Ефективність та часова складність алгоритму Форда-Фалкерсона, як уже було вказано раніше, дуже сильно залежить від вибору збільшувальних шляхів. На прикладі продемонструємо, як замість 2 ітерацій, алгоритм може зробити $2 * n$ ітерацій, де n - довільне натуральне число більше за 0.

Розглянемо наступну мережу



Очевидно, що максимальний потік даною мережею складає $2 * n$. Втім, алгоритм Форда-Фалкерсона може кожен раз обирати шлях, що включатиме в себе ребро $u - v$, обираючи по черзі шляхи $s - u - v - t$ та $s - v - u - t$, таким чином збільшуючи потік усього на 1 одиницю кожен раз, і, відповідно, сягаючи $2 * n$ ітерацій, що явно не є оптимальним.

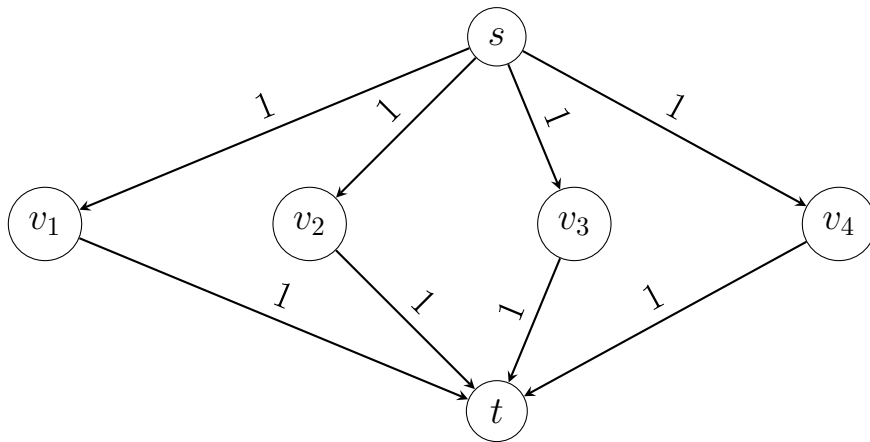
Парні та непарні пропускні спроможності

Дамо відповіді на 2 наступних запитання, стосовно мережі з цілочисельними пропускними спроможностями:

1. Чи існує такий максимальний потік у мережі, де усі ребра мають **парну** пропускну спроможність, що має потужність, що є **парним** числом?
2. Чи існує такий максимальний потік у мережі, де усі ребра мають **непарну** пропускну спроможність, що має потужність, що є **непарним** числом?

У разі, якщо ми маємо мережу, де усі пропускні спроможності є парними числами, то розділивши усі пропускні спроможності на 2, отримаємо цілочисельний максимальний потік, що має парну потужність. Так само, помноживши усі пропускні спроможності цілочисельної мережі на 2, ми отримаємо мережу, де всі пропускні спроможності є парними, як і потужність максимального потоку.

Натомість, для мережі, де усі пропускні спроможності є непарними, максимальний потік не обов'язково матиме непарну пропускну спроможність. Продемонструємо це на конкретному прикладі:



Очевидно, що потужність максимального потоку в даній мережі складає 4 одиниці. Тобто попри те, що усі пропускні спроможності є непарними, потужність максимального потоку є парним числом.

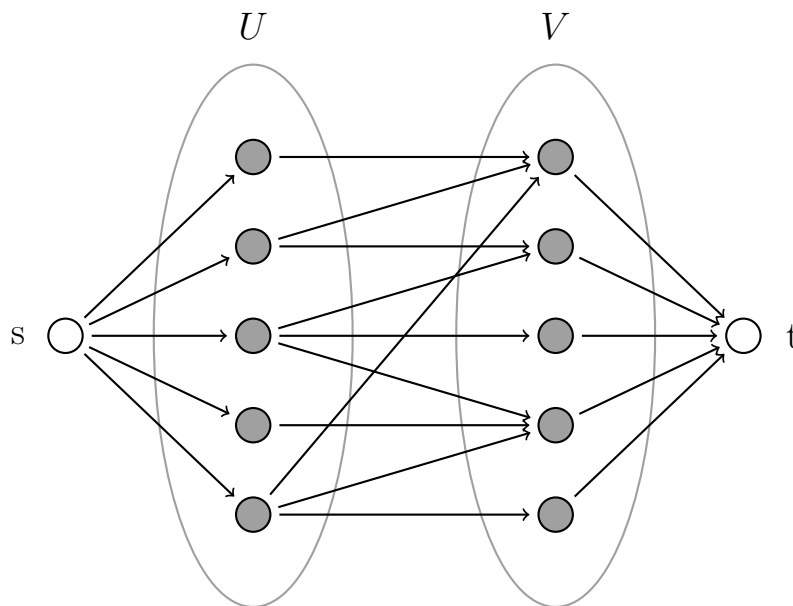
Приклад застосування алгоритму Форда-Фалкерсона

Нехай маємо наступну задачу: існує матриця A , що має розмірність $n \times m$ та складається з невід'ємних дійсних чисел таких, що суми елементів для кожного рядка та стовпчика є цілими числами. Чи існує матриця B розмірності

$n \times m$, що складається з невід'ємних цілих чисел, та має такі ж суми, як і матриця A , для кожного рядка та стовпчика?

Позначимо суми у рядках та стовпцях матриці A , як a_1, \dots, a_n і b_1, \dots, b_m відповідно. Розглянемо повний двочастковий граф $G = (U \cup V, E)$, де $U = \{u_1, \dots, u_n\}$ і $V = \{v_1, \dots, v_m\}$. Перетворимо даний граф на мережу наступним чином:

- направимо усі стрілки з U у V та встановимо їм нескінченну пропускну спроможність;
- додамо джерело s і з'єднаємо його з усіма вершинами U стрілками, що мають пропускну спроможність a_i для вершини u_i ;
- додамо стік t і додамо стрілки з кожної вершини V в t , що матимуть пропускну спроможність b_j для вершини v_j .



Очевидно, що мінімальна пропускну здатність розрізу дорівнює $a_1 + \dots + a_n = b_1 + \dots + b_m$. Оскільки всі пропускні спроможності є цілими числами, теорема Форда-Фалкерсона вказує, що у мережі існує максимальний потік, що є цілим числом. Тоді можемо визначити елемент B_{ij} матриці B як значення цього потоку на ребрі $u_i v_j$. Таким чином, ми довели, що існує матриця B розмірності $n \times m$, що складається з невід'ємних цілих чисел, та має такі ж суми, як і матриця A , для кожного рядка та стовпчика.

3 Застосування теореми Форда-Фалкерсона до доведення мінімаксних теорем теорії графів

3.1 Доведення леми Холла

У цьому підрозділі доведемо лему Холла за допомогою теореми Форда-Фалкерсона. Для початку введемо означення **двочасткового графу**.

Означення 3.1.1. Граф $G = (V, E)$ називають **двочастковим**, якщо існує таке розбиття $\{V_1, V_2\}$ множини його вершин V на дві підмножини (**частки**), що для довільного ребра $(v, w) \in E$ або $v \in V_1$ і $w \in V_2$ або $v \in V_2$ і $w \in V_1$.

Означення 3.1.2. Множина ребер $M \subset E(G)$ називають **паруванням**, якщо жодні два його ребра не мають спільної вершини. **Максимальним паруванням** називається парування з максимальною кількістю ребер.

Будемо казати, що множина вершин $W \subset V(G)$ *покриває* ребро $e \in E(G)$, якщо існує вершина $w \in W$, інцидентна e . Будемо казати, що множина ребер $F \subset E(G)$ *покриває* вершину $v \in V(G)$, якщо існує ребро $f \in F$, інцидентне v .

Лема 3.1.1. Лема Холла (Р.Холл, 1935), також відома як теорема про одруження, є однією з мінімаксних лем у теорії графів. Вона надає необхідні та достатні умови для існування вибору унікальних елементів із певного набору скінченних множин..

Формулювання леми.

Є n хлопців і декілька дівчат. Відомо, що для кожного $1 \leq k \leq n$ довільні k юнаків у сукупності знайомі принаймні з k дівчатами. Тоді всі юнаки можуть обрати по нареченій зі своїх знайомих дівчат.

Твердження у теорії графів

Нехай $G = (V, E)$ — деякий граф, і $A \subseteq V, B \subseteq V$ підмножини його вершин, такі що $A \cup B = V$, і для довільного ребра e , такого що $e = \{v, u\}$, справедливе твердження

$$(v \in A \wedge u \in B) \vee (v \in B \wedge u \in A),$$

тобто граф G є двочастковим. Тоді для даного графа існує набір ребер, що

з'єднують вершини з множини A з *різними* вершинами множини B тоді й лише тоді коли для кожної підмножини елементів $K \subseteq A$ виконується

$$|W| \geq |K|,$$

де:

$$W = \{w \in B : (\exists k \in K)k, w \in E\}$$

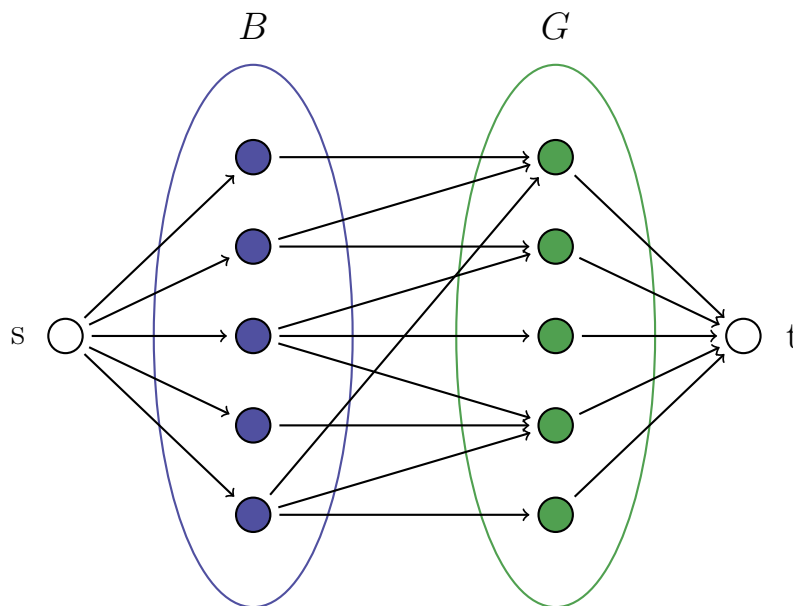
множина елементів з множини B , що з'єднані ребрами хоча б з одним елементом K .

Доведення.

Нехай маємо множину вершин B (boys), що відповідає множині юнаків, юнаки представлені вершинами цієї множини, а також множину вершин G (girls), що відповідає множині дівчат. Множини вершин G і B є частками двочасткового графу.

Введемо також вершини витік (s) і стік (t). Стрілки з витіку (s) входять у кожну вершину множини B , а з усіх вершин множини G стрілки виходять в стік (t). Потік у даній мережі є цілочисельним, а кожна стрілка має максимальну пропускну спроможність 1 (кожна стрілка має пропускну спроможність 1 або 0).

Розглядатимемо стрілки з множини B , як факт того, що юнак b і дівчина g , що поєднані стрілкою, знайомі між собою.



Теорема стверджує, що ми зможемо одружити усіх юнаків, якщо довільні k юнаків знайомі з принаймні k дівчатами. З леми випливає, що дівчат також

має бути не менше, аніж n . Отже,

$$\begin{cases} |B| = n, \\ |G| = m, m \geq n \end{cases}$$

Маємо цілочисельну мережу. За теоремою 1.3.2 у цілочисельній мережі G існує максимальний потік, при чому серед максимальних потоків даної мережі знайдеться цілочисельний.

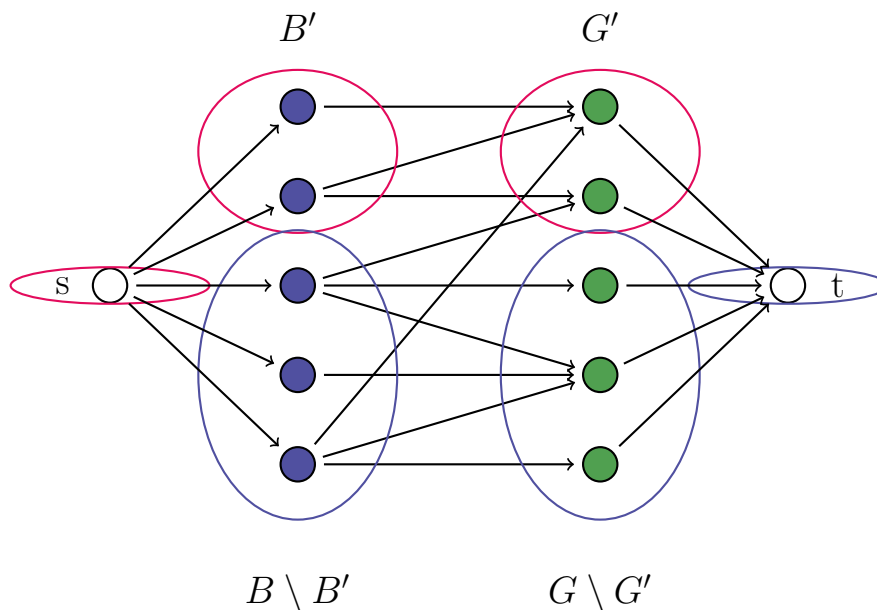
Розглянемо отриману вище мережу та доведемо, що для будь-якого довільного розрізу $\forall(S, T)$, $c(S, T) \geq n$.

Позначимо рожевим певну множину вершин S , що містить витік (s), множину вершин $B' \subseteq B$, $|B'| = k$, $k \leq n$ та множину вершин $G' \subseteq G$, $|G'| = l$, $l \leq m$.

$$S = \{s\} \cup B' \cup G'$$

Синім позначимо множину вершин T , що містить стік (t) та вершини з B та G , що не входять в B' та G' відповідно. Матимемо $|B \setminus B'| = n - k$ та $|G \setminus G'| = m - l$.

$$T = \{t\} \cup (B \setminus B') \cup (G \setminus G')$$



З вершини s у вершини множини $(B \setminus B')$ входить $n - k$ стрілок ($|B'| = k$, $|B| = n$). k юнаків (розглядаємо вершини B , як юнаків) знайомі з принаймні k дівчатами, а отже, з B' у $(G \setminus G')$ виходить $k - l$ стрілок. Відповідно, з G' ($|G'| = l$) у t виходить l стрілок.

Використовуючи наступне рівняння

$$\sum_{u \in S, v \in T} c(u, v) = c(S, T)$$

маємо

$$c(S, T) = \sum_{b \in B \setminus B'} c(s, b) + \sum_{g \in G'} c(g, t) + \sum_{b \in B', g \in G \setminus G'} c(b, g) = n - k + k - l + l = n$$

Оскільки з s виходить усього n стрілок з максимальною пропускнуою спроможністю 1, то відповідно, максимальний потік мережі не може мати потужність більшу за n .

Отже, якщо виконуються умови леми Холла, то пропускна спроможність мережі дорівнює n . А отже, оскільки ми розглядаємо стрілки, як факт знайомства юнака і дівчини, а потік у них — факт шлюбу, то відповідно усі n юнаків одружаться.

3.2 Доведення теореми Кьоніга

У даному підрозділі доведемо теорему Кьоніга за допомогою теореми Форда-Фалкерсона.

Для початку сформулюємо теорему Кьоніга.

Теорема 3.2.1. Нехай є клітчатa таблицка і на ній стоять скінченна кількість тур. Тоді максимильна кількість тур, що не б'ють одна одну рівна мінімальній кількості полос або ж рядів (рядків або стовпчиків), якими можна покрити ці тури.

Розглянемо також графське формулювання теореми. Для цього спочатку введемо означення **вершинного покриття**.

Означення 3.2.1. Підмножина вершин графа називається **вершинним покриттям**, якщо кожне ребро графа інцидентне деякій вершині з цієї підмножини.

Графське формулювання теореми Кьоніга. Максимальна потужність парування у дводольному графі дорівнює мінімальній потужності вершинного покриття.

Доведення. Для початку нагадаємо, що Тура в шахах рухається на будь-яку відстань по горизонталі або вертикалі, захоплюючи фігури, що знаходяться на її шляху. А отже, щоб дві тури не били одна одну, вони мають знаходитися в різних рядках та стовпцях. Позначимо максимальну кількість тур, що не б'ють одна одну M .

Нехай маємо m рядів. Оскільки, щоб тури не били одна одну, вони мають знаходитися у попарно різних рядках і стовпцях, очевидно, що з одного ряду ми можемо взяти максимум 1 туру, щоб вона не била інші. Таким чином на m рядках можна розмістити максимум m тур.

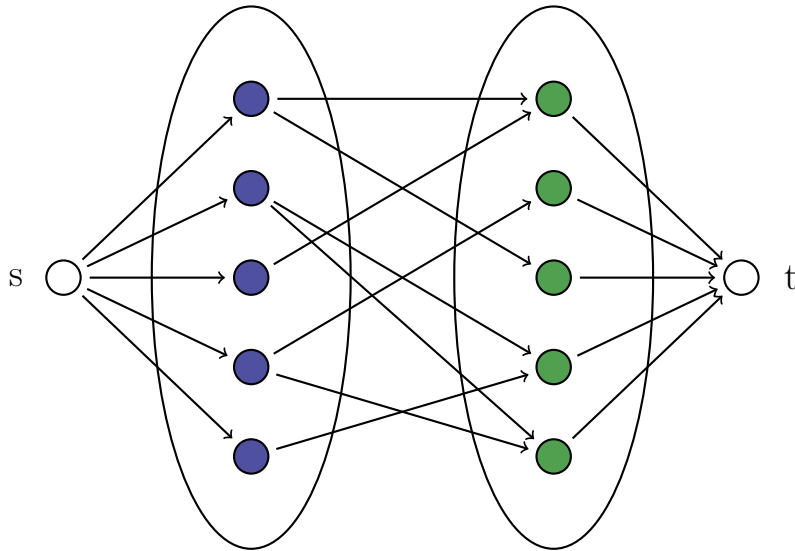
Доведемо це від супротивного. Якщо припустити, що у клітчатій таблиці, що має m рядів і безліч тур, ми хочемо обрати $m + 1$ тур, що не б'ють одна одну. За принципом Діріхле, ми оберемо 2 тури з одного ряду, а вони битимуть одна одну. Відповідно, на m рядках ми можемо обрати не більше m тур.

$$M \leq m$$

Аналогічно з доведенням леми Холла, розглянемо мережу, що складається з витоку (s), стоку (t) та двочасткового графу між ними, де одна частка відповідатиме стовпцям, а інша — рядкам. З витоку (s) виходитимуть стрілки в кожен з вершин-стовпців, а з кожної вершини-рядка виходитимуть у стік (t). Стрілки між вершинами-стовпцями та вершинами-рядками відповідатимуть позиції тури на клітчатій таблиці і матимуть максимальну пропускну спроможність 1, як і всі інші стрілки.

Стовпці (C)

Рядки (R)



Маємо цілочисельну мережу. За теоремою 1.3.2 у цілочисельній мережі G існує максимальний потік, при чому серед максимальних потоків даної мережі знайдеться цілочисельний.

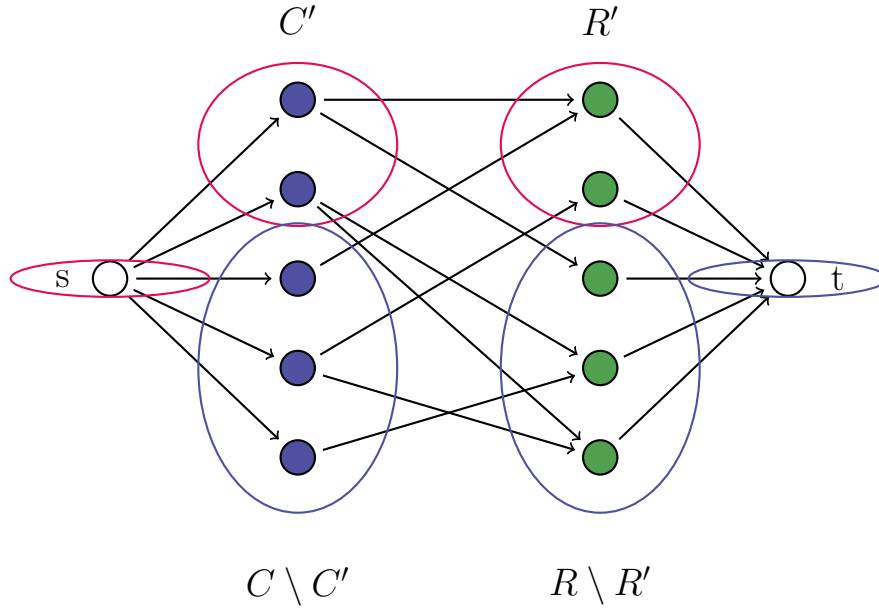
У нашому випадку $\min c(S, T)$ — це мінімальна кількість рядів, що покривають усі тури на таблиці, а $\max |f|$ — максимальна кількість тур на таблиці, що не б'ють одна одну. Таким чином, спробуємо довести, що $\forall c(S, T) \geq m$. Іншими словами, довести, що максимальна кількість тур, що не б'ють одна одну (M) не може бути більше за мінімальну кількість рядів, що їх покриває (m).

Розглянемо наступний розріз. Позначимо рожевим певну множину вершин S , що містить витік (s), множину вершин $C' \subseteq C$ та множину вершин $R' \subseteq R$.

$$S = \{s\} \cup C' \cup R'$$

Синім позначимо множину вершин T , що містить стік (t) та вершини з C та R , що не увійшли в C' та R' відповідно, тобто $C \setminus C'$ та $R \setminus R'$.

$$T = \{t\} \cup (C \setminus C') \cup (R \setminus R')$$



У такому разі, маємо

$$c(S, T) = |C \setminus C'| + |R'| + \sum_{u \in C', v \in R \setminus R'} c(u, v)$$

Обґрунтуємо наступну нерівність

$$c(S, T) = |C \setminus C'| + |R'| + \sum_{u \in C', v \in R \setminus R'} c(u, v) \geq m$$

Очевидним здається, що вершини-стовпці з множини $C \setminus C'$ покриватимуть усі тури, що в них знаходяться (усі ребра, що з них виходять), тобто до $|C \setminus C'|$ тур. Далі, розглянувши множину вершин-рядків R' розуміємо, що вона покриває усі тури, що стоять в цих рядках. У свою чергу $\sum_{u \in C', v \in R \setminus R'} c(u, v)$ ми можемо розглядати, як тури, що стоять на перетинах стовпців множини C' та рядків множини $R \setminus R'$, тобто ця сума покриває усі тури, що не входять в $C \setminus C'$ та R' . Можемо поставити у відповідність цим стрілкам-турам вершини-стовпчики з множини C' , з яких виходять стрілки в $R \setminus R'$, матимемо певну множину стовпців-вершин V , $|V| \leq \sum_{u \in C', v \in R \setminus R'} c(u, v)$ (адже, якщо з однієї вершини виходитимуть 2 стрілки, то ми все одно ставитимемо у відповідність їм одну вершину). Таким чином, $|V|$ стовпців-вершин вистачить, щоб покрити усі стрілки між C' та $R \setminus R'$, тобто усі тури, що стоять на перетині стовпців з C' та рядків з $R \setminus R'$.

Таким чином, маємо, що $M \leq m$, тобто максимальна кількість тур, що не б'ють одна одну, не більше мінімальної кількості рядів, якими їх можна покрити. А також, $\forall c(S, T) \geq m$, тобто будь-який розріз мережі, що відповідає клітчатій таблиці, де розставлені тури, має бути не менше за m , тобто мінімальні кількість рядів, що покривають тури. Маємо

$$m \leq M \leq m \longrightarrow M = m,$$

що і треба було довести.

3.3 Доведення теореми Менгера

У даному підрозділі за допомогою теореми Форда-Фалкерсона доведемо теорему Менгера.

Для початку введемо декілька означень.

Означення 3.3.1. Маршрутом (або **шляхом**) у графі $G = (V, E)$ називають послідовність

$$v_1, e_1, v_2, e_2, \dots, e_k, v_{k+1}$$

вершин v_i і ребер e_i таку, що кожен два сусідні ребра в ній мають спільну вершину, отже, $e_i = (v_i, v_{i+1}), i = 1, 2, \dots, k$.

Означення 3.3.2. Маршрут, у якому всі ребра попарно різні, називають **ланцюгом**, а той, у якому всі вершини попарно різні - **простим ланцюгом**.

Означення 3.3.3. Граф називають **зв'язним**, якщо будь-яку пару його вершин можна з'єднати деяким маршрутом.

Означення 3.3.4. Числом реберної зв'язності графа називається найменше число ребер, вилучення яких приводить до незв'язного графа. Граф, у якого число реберної зв'язності дорівнює k , називають **k -реберно-зв'язним**.

Нехай G - неорієнтований граф без петель, де кратні ребра дозволяються. Для $x, y \in V(G)$ позначимо як $\lambda_G(x, y)$ мінімальне число ребер, що відділяють x та y . Назвемо $\lambda_G(x, y)$ *реберною зв'язністю вершин x та y* . Очевидно, що реберна зв'язність графа може бути задана, як $\lambda(G) = \min_{x, y \in V(G)} \lambda_G(x, y)$.

Означення 3.3.5. Компонентою зв'язності (або **зв'язною компонентою**) графа G називають його зв'язний підграф такий, що не є власним підграфом жодного іншого зв'язного підграфа графа G .

Тепер сформулюємо теорему Менгера у її реберному та вершинному формулюванні. Розглянемо її на прикладі графу A :

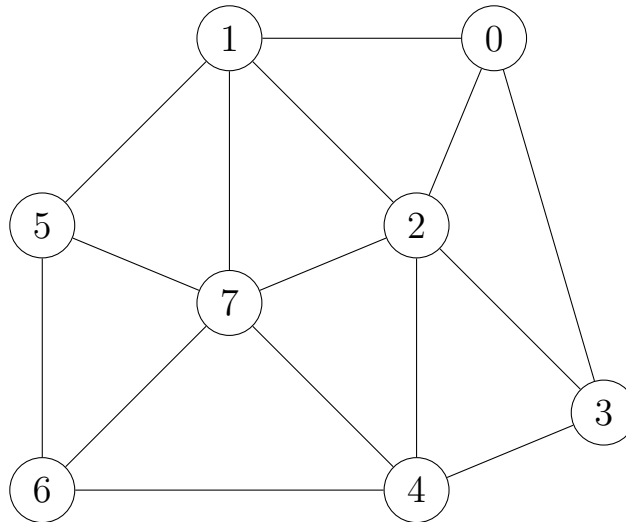


Рис. 11: Граф A

Теорема 3.3.1. Менгера (вершинна форма). Для будь-яких двох несуміжних вершин s і t , найменша кількість вершин, що розділяють ці вершини (після видалення цих вершин s і t опиняються в різних компонентах зв'язності), дорівнює найбільшій кількості простих $(s - t)$ -ланцюгів, що попарно не перетинаються, тобто таких, що за початкову вершину мають s , а за кінцеву — t , усі інші вершини різні.

Приклад. Візьмемо за вершину s вершину 6, а за t — 0 на графі A , тоді матимемо 3 наступні прості $(s - t)$ -ланцюги: 6-5-1-0, 6-7-2-0, 6-4-3-6.

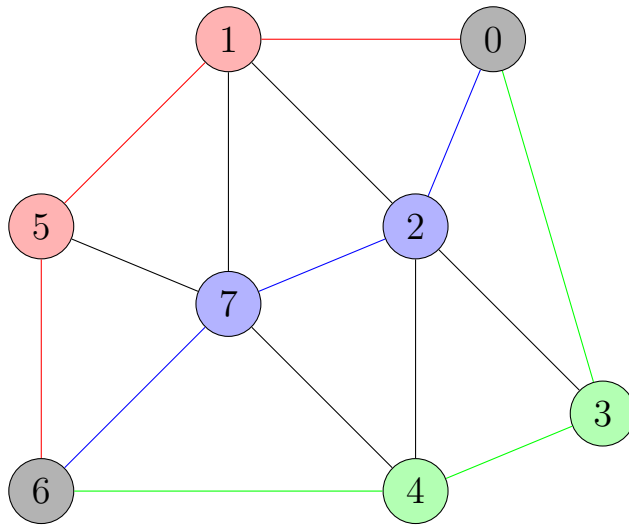


Рис. 12: Граф А

Бачимо, що, дійсно, потрібно прибрати мінімум 3 вершини, аби вершини 0 та 6 опинилися в різних компонентах зв'язності. Приберемо вершини 5, 7, 4 і отримаємо наступний рисунок:

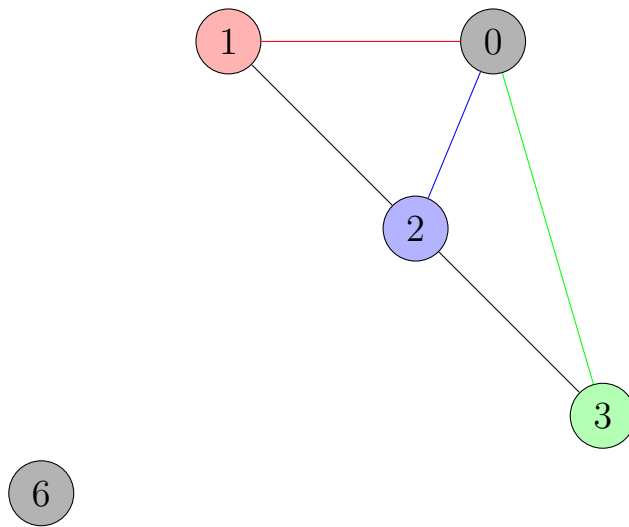


Рис. 13: Граф А

Теорема 3.3.2. Менгера (реберна форма). Для будь-яких двох несуміжних вершин графа найбільша кількість реберно-неперетинних ланцюгів, що з'єднують їх, дорівнює найменшій кількості ребер, які розділяють ці вершини.

Приклад. Розглянемо на прикладі графу А. За s візьмемо 6, а за t — 0. Розглянемо наступні реберно-неперетинні шляхи:

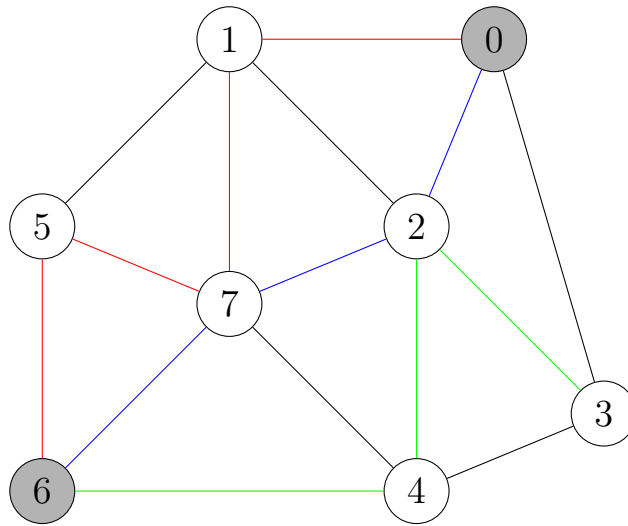


Рис. 14: Граф А

Звісно, також є й інші реберно-неперетинні шляхи, але їх у будь-якому разі буде лише 3, тому розглядаємо саме ці. Бачимо, що, дійсно, аби вершини 6 та 0 опинилися у різних компонентах зв'язності, маємо прибрати мінімум 3 ребра. Це можуть бути ребра $(6-5)-(6-7)-(6-4)$ або ж $(1-0)-(2-0)-(0-3)$. Приберемо ребра $(6-5)-(6-7)-(6-4)$:

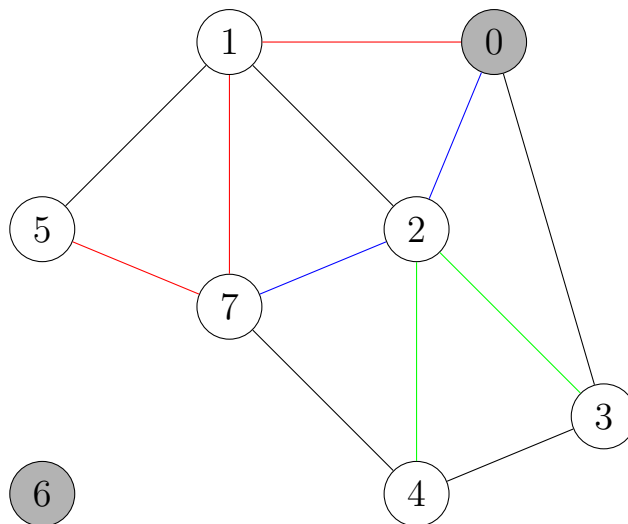


Рис. 15: Граф А

Перейдемо до доведення. Спочатку доведемо теорему Менгера у реберно-формулюванні.

Переформулюємо теорему наступним чином:

1. Нехай $s, t \in V(G)$. Тоді існує $\lambda_G(s, t)$ шляхів з s у t , що не мають спільних ребер.

2. У k -реберно-зв'язному графі G для будь-яких двох вершин s, t існує k шляхів з s у t , що не мають спільних ребер.

Доведення.

1. Побудуємо мережу \vec{G} на множині вершин $V(G)$. За витік вважатимемо вершину s , а за стік — вершину t . Визначимо пропускні спроможності: $c(v_1, v_2)$ рівна кратності ребер v_1v_2 у графі G . Таким чином, пропускна спроможність $c(v_1, v_2) = 0$, якщо ребра v_1v_2 немає, і n , якщо в графі G є n кратних ребер v_1v_2 . Крім того, $c(v, s) = c(t, v) = 0$ для усіх $v \in V(G)$, $v_1, v_2 \notin \{s, t\}$ маємо $c(v_1, v_2) = c(v_2, v_1)$.

Мережа \vec{G} є цілочисельною. Відповідно до теореми 1.3.2, у ній існує максимальний цілочисельний потік f . Нехай $|f| = k$. Спробуємо довести, що потік f розпадається на k $(s - t)$ -шляхів, що не перетинаються.

Побудуємо новий орієнтований граф G' на тих самих вершинах. Якщо $f(x, y) = l > 0$ для $x, y \in V(G)$, проведемо у G' рівно l стрілок xy . Очевидно, що $l \in \mathbb{Z}$ і у графі G є не менше l ребер, що з'єднують x та y . Тоді з вершини s виходить рівно k стрілок, а в кожній відмінній від s і t вершині v , за властивістю потоку кількість стрілок, що виходить, та стрілок, що входить, рівна. Тепер стало очевидно, що потік розпадається на k реберно-неперетинних $(s - t)$ -шляхів у графі G і, можливо, декілька циклів, що не проходять через s та t . Ці цикли нас не цікавлять.

Використаємо теорему Форда-Фалкерсона. Оскільки максимальний потік f в нашій мережі має потужність $|f| = k$, то існує розріз (S, T) , що має пропускну спроможність $c(S, T) = k$. Тоді з S у T виходить k стрілок нашої мережі, у якій пропускна спроможність кожної стрілки рівна 1. Для кожної стрілки xy з S у T помістимо в деяку множину R ребро xy графа G . Тоді $|R| = k$ і не складно зрозуміти, що R відділяє S від T , а відповідно, і s від t у графі G . Відповідно, $k \geq \lambda_G(s, t)$ і твердження 1 доведене.

2. Твердження 2 випливає з твердження 1.

Висновки

У роботі було проведено всебічний аналіз теореми Форда-Фалкерсона. Було розглянуто основні аспекти теореми, а також її було застосовано для доведення деяких мінімаксних теорем теорій графів, а саме теорема Кьоніга, лема Холла та теорема Менгера.

Також, було розглянуто алгоритм Форда-Фалкерсона та проведено аналіз його ефективності. Зокрема, було розглянуто, яку часову складність має даний алгоритм, у яких випадках алгоритм може зациклюватися та як працює у мережі з ірраціональними пропускними спроможностями.

Робота містить 3 розділи.

У першому розділі розглядаються основні означення теорії графів, потоків у мережі та сформульовано теорему Форда-Фалкерсона. У даному розділі розглядаються наступні теми:

1. Основні означення теорії графів
2. Основні означення теорії потоків у мережі
3. Теорема Форда-Фалкерсона

У другому розділі розглядається алгоритм Форда-Фалкерсона, приклад його використання та аналіз даного алгоритму. У ньому розглядаються наступні теми:

1. Опис алгоритму Форда-Фалкерсона
2. Приклад використання алгоритму
3. Аналіз алгоритму

У третьому розділі наведені сформульовані мною доведення мінімаксних теорем теорії графів за допомогою теореми Форда-Фалкерсона, а саме доведення леми Холла, теореми Кьоніга та теореми Менгера.

Література

- [1] *L. R. Ford Jr. and D. R. Fulkerson* (1956) “Maximal Flow Through A Network” , Canadian Journal of Mathematics 8: 399-404
- [2] *Schrijver, Alexander.* (2004). On the History of the Transportation and Maximum Flow Problems. Mathematical Programming. 91. 10.1007/s101070100259.
- [3] *Карпов Д. В.* Теорія графів. 2017
- [4] *Reinhard Diestel.* Graph Theory. GTM 173, 5th edition 2016/17. Springer-Verlag, Heidelberg Graduate Texts in Mathematics, Volume 173
- [5] *Трохимчук Р. М., Нікітченко , М. С.* (2017) Дискретна математика у прикладах і задачах: навч. посібник. - Київ : Київський університет, - 248 с.
- [6] *А. В. Стьопкін, Д. А. Пластун* (2016). Алгоритм Форда-Фалкерсона. Збірник наукових праць фізико-математичного факультету ДДПУ, 0(6), 84-91.
- [7] Flow Network. Brilliant.org. Retrieved from <https://brilliant.org/wiki/flow-network/>
- [8] *Daniel García Solla* (2023) Discovering the Maxflow Mincut Theorem: A Comprehensive and Formal Approach. Retrieved from <https://towardsdatascience.com/discovering-the-maxflow-mincut-theorem-a-comprehensive-and-formal-approach-e2402ce935c7>
- [9] *Luca Trevisan* (2011). Lecture 9. Stanford University — CS261: Optimization
- [10] *Daniel Anderson* (2023). Lecture 11. Carnegie Mellon University - 15-451/651: Algorithm Design and Analysis.
- [11] *Phillip. Hall* (1935). On representatives of subsets. J. London Math. Soc. 10, 26–30

- [12] Big O Notation. Brilliant.org. Retrieved 18:56, May 19, 2024, from <https://brilliant.org/wiki/big-o-notation/>
- [13] *Uri Zwick* (1995). The smallest networks on which the Ford-Fulkerson maximum flow procedure may fail to terminate. Theoretical Computer Science. Volume 148, Issue 1, 165-170
- [14] *Kevin Wayne* (2004). Princeton University. COS 226: Algorithms and Data Structures
- [15] Graph Theory. Brilliant.org. Retrieved from <https://brilliant.org/wiki/graph-theory/>
- [16] *Robin J. Wilson* (1996) Introduction to Graph Theory (4th edition). Longman publishing
- [17] Allan Borodin (2013). CSC 373: Algorithm Design and Analysis. Lecture 12