

Міністерство освіти і науки України  
Харківський національний університет радіоелектроніки

Факультет комп'ютерної інженерії та управління  
(повна назва)

Кафедра електронних обчислювальних машин  
(повна назва)

**КВАЛІФІКАЦІЙНА РОБОТА**  
**Пояснювальна записка**

Рівень вищої освіти другий (магістерський)

Методи управління потоками даних в високошвидкісних  
комп'ютерних мережах

(тема)

Виконав:

студент II курсу, групи СПМ-21-2  
Філіппов В.В.  
(прізвище, ініціали)

Спеціальність 123 – Комп'ютерна інженерія  
(код і повна назва спеціальності)

Тип програми освітньо-наукова  
(освітньо-професійна або освітньо-наукова)

Освітня програма Комп'ютерні системи та мережі  
(повна назва освітньої програми)

Керівник: доц. Янковський О.А.  
(посада, прізвище, ініціали)

Допускається до захисту

Зав. кафедри ЕОМ

Коваленко А.А.  
(прізвище, ініціали)

2023 р.

Харківський національний університет радіоелектроніки

Факультет \_\_\_\_\_ комп'ютерної інженерії та управління \_\_\_\_\_

Кафедра \_\_\_\_\_ електронних обчислювальних машин \_\_\_\_\_

Рівень вищої освіти \_\_\_\_\_ другий (магістерський) \_\_\_\_\_

Спеціальність \_\_\_\_\_ 123 – Комп'ютерна інженерія \_\_\_\_\_  
(код і повна назва)

Тип програми \_\_\_\_\_ освітньо-наукова \_\_\_\_\_  
(освітньо-професійна або освітньо-наукова)

Освітня програма \_\_\_\_\_ Комп'ютерні системи та мережі \_\_\_\_\_  
(повна назва)

ЗАТВЕРДЖУЮ:

Зав. кафедри \_\_\_\_\_  
(підпис)

“ \_\_\_\_\_ ” \_\_\_\_\_ 20\_\_ р.

**ЗАВДАННЯ**

**НА КВАЛІФІКАЦІЙНУ РОБОТУ**

студенту \_\_\_\_\_ Філіппову Владлену Валерійовичу \_\_\_\_\_  
(прізвище, ім'я, по батькові)

1. Тема роботи Методи управління потоками даних в високошвидкісних комп'ютерних мережах

затверджена наказом по університету від “ 3 ” квітня 2023 р. № 318 Ст

2. Термін подання студентом роботи до екзаменаційної комісії 12 травня 2023р.

3. Вхідні дані до роботи 1) моделі та методи для керування мережевими інформаційними потоками; 2) сучасні вимоги до мережних показників; 3) перелік використаних програмних та апаратних засобів: ОС Windows 10, OpNet 14, NS-2.

4. Перелік питань, що потрібно опрацювати в роботі \_\_\_\_\_

1) аналіз сучасного стану проблеми

2) огляд технологій управління живучістю оптичних мереж WDM

3) моделі управління мережним трафіком

4) вибір програмних та апаратних засобів реалізації

5) проведення експериментальних досліджень

б) висновки

5. Перелік графічного матеріалу із зазначенням креслеників, схем, плакатів, комп'ютерних ілюстрацій (слайдів) \_\_\_\_\_  
Слайдів презентації – 19 шт. \_\_\_\_\_  
\_\_\_\_\_  
\_\_\_\_\_  
\_\_\_\_\_  
\_\_\_\_\_  
\_\_\_\_\_

6. Консультанти розділів роботи (заповнюється за наявності консультантів згідно з наказом, зазначеним у п.1 )

Найменування розділу	Консультант (посада, прізвище, ім'я, по батькові)	Позначка консультанта про виконання розділу	
		підпис	дата

### КАЛЕНДАРНИЙ ПЛАН

№	Назва етапів роботи	Термін виконання етапів роботи	Примітка
1	Аналіз стану проблеми та сучасних методів її вирішення	04.04.23–10.04. 23	
2	Огляд технологій підвищення живучості мереж	11.04. 23–14.04. 23	
3	Розробка моделі управління мережним трафіком	15.04. 23 –24.04. 23	
4	Вибір програмних та апаратних засобів реалізації	25.04. 23 –27.04. 23	
5	Тестування запропонованих методів	28.04. 23–03.05. 23	
6	Оформлення пояснювальної записки	04.05. 23–11.05. 23	

Дата видачі завдання 4 квітня 2023 р.

Студент \_\_\_\_\_  
(підпис)

Керівник роботи \_\_\_\_\_  
(підпис)

доц. Янковський О.А. \_\_\_\_\_  
(посада, прізвище, ініціали)

## РЕФЕРАТ

Пояснювальна записка кваліфікаційної роботи: 109 с., 30 рис., 10 табл., 1 дод., 22 джерела.

ВІДНОВЛЕННЯ, ЕВРИСТИКА, ЛОГІЧНА ТОПОЛОГІЯ, МАРШРУТИЗАЦІЯ ТА ПРИЗНАЧЕННЯ ДОВЖИНИ ХВИЛІ, МЕРЕЖА З МАРШРУТИЗАЦІЄЮ ЗА ДОВЖИНОЮ ХВИЛІ, ФІЗИЧНА ТОПОЛОГІЯ, ЗАХИСТ, WDM

Метою кваліфікаційної магістерської роботи є дослідження та розробка алгоритмів живучості на основі концепції р-циклів в оптичних сітчастих мережах WDM з метою збалансування оптимальності рішення та обчислювальної складності. Запропоновано два нових евристичних підходи до обчислення ефективних р-циклів, які можуть призвести до кращої продуктивності з точки зору ефективності пропускну здатності. Спочатку визначається набір р-циклів через ідентифікацію основних циклів у мережі, а потім застосовується формулювання цілочисельної лінійної програми (ILP) для захисту всього даного попиту на трафік. Запропоновано евристичний підхід без використання моделі ILP на основі критерію відповідності між розподілом робочих каналів і розподілом захищених каналів на вибраному наборі р-циклів.

## ABSTRACT

Master's thesis: 109 pages, 30 figures, 10 tables, 1 appendices, 22 sources.

HEURISTICS, LOGICAL TOPOLOGY, PHYSICAL TOPOLOGY, RECOVERY, SECURITY, WAVELENGTH ROUTING AND ASSIGNMENT, WAVELENGTH ROUTING NETWORK, WDM

The purpose of the qualifying master's work is to research and develop survivability algorithms based on the concept of p-cycles in WDM optical mesh networks in order to balance the optimality of the solution and the computational complexity. Two new heuristic approaches are proposed to compute efficient p-cycles that can lead to better performance in terms of bandwidth efficiency. First, a set of p-cycles is determined by identifying the main cycles in the network, and then an integer linear program (ILP) formulation is applied to protect all given traffic demand. A heuristic approach is proposed without using the ILP model based on the correspondence criterion between the distribution of working channels and the distribution of protected channels on a selected set of p-cycles.

## ЗМІСТ

ПЕРЕЛІК УМОВНИХ ПОЗНАЧЕНЬ, СИМВОЛІВ, ОДИНИЦЬ, СКОРОЧЕНЬ І ТЕРМІНІВ .....	8
ВСТУП .....	9
1 МЕТА КВАЛІФІКАЦІЙНОЇ РОБОТИ .....	10
2 ОГЛЯД СУЧАСНОГО СТАНУ ПРОБЛЕМИ .....	11
2.1 Фізична та логічна топології.....	12
2.2 Несправності в мережах WDM.....	15
2.3 Передача даних у мережі WDM за умови відсутності збоїв .....	16
2.4 Передача даних у мережі WDM за допомогою схеми захисту .....	17
3 ОПТИЧНІ МЕРЕЖІ WDM.....	19
3.1 Основні пристрої WDM.....	19
3.2 Маршрутизація за довжиною хвилі .....	21
3.3 Мережі з одним і кількома переходами.....	22
3.4 Логічна топологія в мережах, де не враховується можливість несправностей.....	24
3.4 Підходи MILP .....	28
3.5 Евристичні підходи.....	29
3.6 Живучість і управління несправностями.....	30
3.7 Відновлення .....	32
3.8 Захист .....	33
3.8.1 Захист виділеного шляху.....	34
3.8.2 Захист спільного шляху.....	37
4 ЖИВУЧІСТЬ МЕРЕЖІ.....	40
4.1 Заздалегідь спланований захист .....	40
4.2 Динамічне відновлення .....	44
4.3 Захист спільного резервного шляху.....	46
4.4 Попередньо налаштований цикл захисту .....	47

4.5	Формалізація ІЛР.....	50
4.6	Евристичні підходи.....	52
5	АНАЛІТИЧНА ЧАСТИНА .....	61
5.1	Теорія графів.....	61
5.2	Тестові мережі .....	67
6	ГЕНЕРАЦІЯ ЕФЕКТИВНОЇ МНОЖИНИ КАНДИДАТІВ У Р- ЦИКЛИ.....	70
6.1	Попередня теорія.....	70
6.2	Генерація потенційних р-циклів.....	71
6.2.1	Алгоритм пошуку в ширину-глибину (АПШГ).....	71
6.2.2	Створення набору кандидатів на р -цикли .....	73
6.3	Результати та аналіз .....	76
6.3.1	Створення множини основних циклів .....	76
6.3.2	Оптимальність рішення та обчислювальна складність.....	77
7	ЕВРИСТИЧНИЙ ПІДХІД ДЛЯ ПРОЕКТУВАННЯ ЖИВУЧИХ МЕРЕЖ WDM .....	83
7.1	Евристичний підхід для ефективного вибору циклу (ЕЕВЦ) .....	83
7.2	Результати моделювання та аналіз.....	88
7.2.1	Результати моделювання для топології NSFNet.....	89
7.2.2	Результати моделювання для топології COST239.....	90
7.2.3	Результати моделювання для топології EON.....	91
7.2.4	Результати моделювання для топології USA networks .....	92
	ВИСНОВКИ.....	95
	ПЕРЕЛІК ДЖЕРЕЛ ПОСИЛАННЯ .....	96
	ДОДАТОК А Графічний матеріал кваліфікаційної роботи.....	99

ПЕРЕЛІК УМОВНИХ ПОЗНАЧЕНЬ, СИМВОЛІВ, ОДИНИЦЬ, СКОРОЧЕНЬ  
І ТЕРМІНІВ

APS – автоматичне захисне перемикання (англ., Automatic Protection Switching)

D-MLTDA – алгоритм проектування логічної топології з декількома переходами (англ., Decreasing Multi-hop Logical Topology Design Algorithm)

GLTDA – алгоритм проектування жадібної логічної топології (англ., Greedy logical topology design algorithm)

I-MLTDA – алгоритм проектування логічної топології із збільшенням кількості переходів (англ., Increasing Multi-hop Logical Topology Design Algorithm)

IP – Інтернет-протокол (англ., Internet protocol)

ILP – цілочисельне лінійне програмування (англ., Integer Linear Programming)

ITU-T – міжнародний союз електрозв'язку з телекомунікацій (англ., International Telecommunications Union on Telecommunications)

OADM – оптичний мультиплексор додавання/виведення (англ., Optical Add/Drop Multiplexer)

OXC – оптичний перехресний зв'язок Optical Cross-Connect

SDH – синхронна цифрова ієрархія (англ., Synchronous Digital Hierarchy)

SONET – синхронна оптична мережа (англ., Synchronous Optical Network)

UPSR – односпрямоване кільце з комутацією каналів (англ., Unidirectional Path Switched Ring)

WDM – мультиплексування з розділенням хвиль (англ., Wavelength Division Multiplexing)

## ВСТУП

Мультиплексування за довжиною хвилі (WDM) є важливою технологією, яка дозволяє передавати великі обсяги даних через оптичні мережі. Усі оптичні мережі на основі WDM були використані для покращення загальної пропускної здатності зв'язку та забезпечення можливого вибору для проектування магістральних мереж.

Однак через високе трафікове навантаження, яке може нести кожен канал зв'язку в мережі WDM, стійкість до збоїв такої мережі стає дуже важливою. Живучість у цьому контексті – це здатність мережі підтримувати безперервність обслуговування в разі збоїв, оскільки збій може призвести до величезних втрат даних.

Останніми роками було вивчено багато механізмів живучості, а їх продуктивність оцінювалася через ефективність ємності, час відновлення та здатність до відновлення. Особливу увагу приділено механізмам живучості для кільцевих і сітчастих топологій. Кільцевий механізм має перевагу в тому, що він може запропонувати швидкий час відновлення (50~60 мс), але він вимагає резервування в порядку 100%~200% (співвідношення резервної ємності до робочої ємності). З іншого боку, сітчасті захисні механізми вимагають менше резервування (50%~70%), ніж кільцеві механізми, але мають більший час відновлення.

Концепція попередньо налаштованого циклу захисту (р-цикл) була розроблена як гібрид кільцевих і сітчастих механізмів захисту, їй притаманні швидкий час відновлення кільцевого захисту та ефективність пропускної здатності сітчастих топологій.

Метод р-циклу засновано на замкнутих циклічних маршрутах для скорочення часу відновлення. Однак, на відміну від кільцевого механізму, який захищає лише робочі канали на кільці, метод р-циклу пропонує корисні резервні шляхи для захисту розрізнених проміжків, а також прогонів у циклі.

## 1 МЕТА КВАЛІФІКАЦІЙНОЇ РОБОТИ

Оптичні мережі, засновані на технології WDM, можуть мати дуже високу пропускну здатність по волоконно-оптичним з'єднанням у мережі, що дозволяє значно збільшити використання пропускну здатності. Однак висока пропускну здатність, яку забезпечують такі канали зв'язку, створює потенційну проблему втрати величезної кількості інформації, коли відбувається збій вузла або зв'язку. Таким чином, живучість мережі є дуже важливим питанням в оптичних мережах, щоб запобігти таким величезним втратам.

Метою кваліфікаційної магістерської роботи є дослідження та розробка алгоритмів підвищення живучості мережі на основі концепції р-циклів в оптичних сітчастих мережах WDM з метою балансування оптимальності складності рішення та обчислення.

Однак визначення оптимального набору р-циклів для захисту є NP-складною проблемою. Існуючі підходи до вирішення проблеми р-циклів полягають у використанні моделі цілочисельного лінійного програмування (ILP). Метою моделі ILP є досягнення оптимального рішення з точки зору мінімізації резервної ємності при збереженні 100% захисту. Однак модель ILP стає важкорозв'язною з великомасштабними мережами, де кількість можливих змінних дуже велика.

Евристичні підходи можуть досягти майже оптимальних рішень протягом прийняттого часу відновлення. Дуже бажані прості та ефективні евристичні методи. Таким чином, необхідно розробити евристичні алгоритми для обчислення ефективних р-циклів на графах мережі, які можуть призвести до кращої продуктивності з точки зору ефективності використання пропускну здатності.

## 2 ОГЛЯД СУЧАСНОГО СТАНУ ПРОБЛЕМИ

Швидке зростання кількості додатків, що потребують інтенсивної пропускної здатності, таких як телебачення високої чіткості, відео та цифрове аудіо, створило безпрецедентний попит на пропускну здатність в Інтернеті.

За деякими оцінками, використання пропускної здатності в Інтернеті подвоюється кожні два роки, і очікується, що зростаючий попит на пропускну здатність мереж триватиме в найближчі роки. Завдяки останнім досягненням в оптичних технологіях, особливо розробці методів мультиплексування за довжиною хвилі (WDM), кількість необробленої смуги пропускання, доступної на волоконно-оптичних лініях зв'язку, зросла на кілька порядків. Тепер одне волокно може мати 100 або більше довжин хвиль із можливою швидкістю передачі даних від 2,5 до 10 гігабіт на секунду на кожній довжині хвилі [1]. Але це призводить до серйозної невідповідності поточним технологіям комутації, які ще не здатні перемикати на цих високих сукупних швидкостях.

IP через WDM було запропоновано як можливе вирішення цієї проблеми [2]. IP через WDM – це нова структура рівнів, у якій IP-пакети безпосередньо відображаються в канали з різними довжинами хвилі, не проходячи через проміжні рівні, такі як ATM або SONET/SDH. Такі архітектури оптимізують як мережеве обладнання, так і відповідне програмне забезпечення, і водночас забезпечують гнучку інфраструктуру. Запуск IP-трафіку безпосередньо через мережу WDM вважається оптичною мережею наступного покоління.

Мультиплексування за довжиною хвилі (WDM) ділить величезну смугу пропускання волокна на  $K$  довжин хвиль несучої, що не перекриваються,  $\lambda_1, \lambda_2, \dots, \lambda_K$ , які називаються каналами WDM [3]. Значення  $K$  визначається використовуваною технологією xWDM.

За допомогою WDM кожне фізичне волоконно-волоконне з'єднання може підтримувати багато оптичних сигналів (стільки, скільки є довжин хвиль несучих на волокні). Найбільш частою причиною збою в мережах WDM є обрив волокна через розрив будь-якого волокна в мережі.

## 2.1 Фізична та логічна топології

Фізична топологія оптичної мережі складається з набору кінцевих вузлів, вузлів маршрутизаторів та оптичних волокон, що з'єднують ці вузли. Кожен кінцевий вузол у мережі здатний генерувати дані для передачі, отримувати дані і, як правило, є комп'ютером, який має кілька оптичних передавачів і приймачів. Вузол маршрутизатора – це оптичний пристрій, підключений до кількох вхідних волокон і кількох вихідних волокон. Оптичні сигнали на будь-якому вхідному волокні можуть передаватися на будь-яке вихідне волокно залежно від способу програмування маршрутизатора. Фізична топологія оптичної мережі може бути зручно представлена оргграфом  $G$ , де:

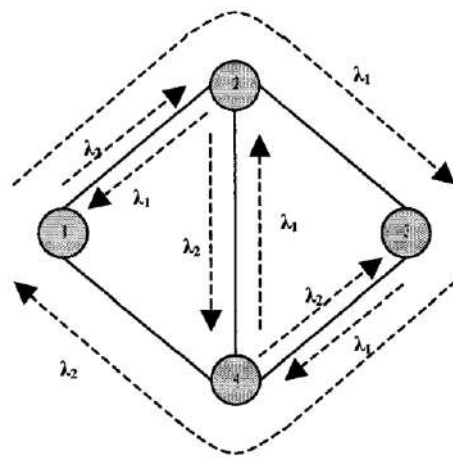
- вузли в графі  $G$  є кінцевими вузлами та вузлами маршрутизаторів у фізичній мережі;
- якщо пара вузлів  $(i, j)$  з'єднана волокном у фізичній мережі, на графі  $G$  існує спрямоване ребро (також зване ланкою) від  $i$  до  $j$ .

Світловий шлях в оптичній мережі – це шлях зв'язку «точка-точка», який з'єднує передавач у вузлі джерела з приймачем у вузлі призначення [3], де оптико-електронне перетворення не потрібне в будь-якому проміжному вузлі.

Такий шлях, де немає оптико-електронного перетворення, називається повністю оптичним.

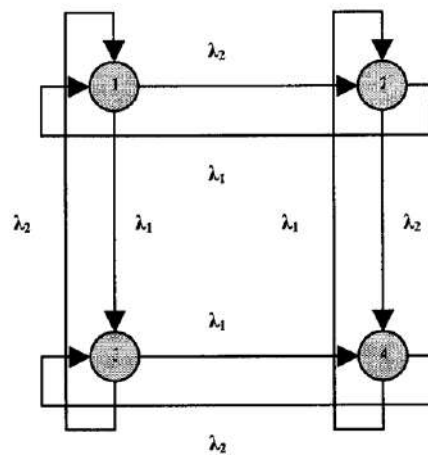
Якщо подивитись на графову модель фізичної топології мережі, наведену на рисунку 2.1, світловий шлях починається від кінцевого вузла (джерела зв'язку), проходить через низку зв'язків (оптичні волокна у

фізичній мережі) і проміжний вузли (оптичні маршрутизатори) і закінчується на кінцевому вузлі (пункт призначення для зв'язку). У найпростішому та найбільш широко використовуваному випадку, шлях світла використовує ту саму довжину хвилі несучої від вузла джерела до вузла призначення на всіх волоконно-волоконних лініях від вузла джерела через проміжні вузли до вузла призначення. що також називається обмеженням безперервності довжини хвилі [4].



(a)

Фізична топологія



(б)

Логічна топологія

Рисунок 2.1 – Фізична та логічна топологія

Після маршрутизації світлових шляхів у фізичній топології, як правило, існує багато світлових шляхів, які спільно використовують одне фізичне з'єднання. Якщо два шляхи світла використовують одне і те ж волокно, то їм потрібно призначити дві різні довжини хвилі несучої на цьому волокні [5].

Якщо існує світловий шлях від вузла Р до вузла Q, ми можемо передати дані від Р до Q за допомогою передавача в Р, який модулює деяку несучу довжину хвилі  $A_i$  для деякого  $i$ ,  $1 < i < K$ , використовуючи дані для надсилання і вводить його в волокно, що з'єднує вузол джерела з першим проміжним вузлом на шляху від Р до Q. Це передасть сигнал на довжині

хвилі  $\lambda_i$  до першого проміжного вузла, який є маршрутизатором. Цей маршрутизатор уже запрограмований на спрямування вхідного сигналу на довжині хвилі  $\lambda_i$  до відповідного волокна, яке з'єднує перший маршрутизатор з наступним маршрутизатором на шляху до вузла призначення. Цей процес триває, поки сигнал не досягне вузла призначення. Такий зв'язок, що включає лише один світловий шлях, також називається однострибковим.

Як показано на рисунку 2.1 (б), зв'язок від вузла 1 до вузла 3 включає один перехід, хоча у фізичній топології (рисунок 2.1 (а)) між вузлом 1 і вузлом 3 немає прямого оптоволоконного зв'язку.

Щоб забезпечити можливе спілкування лише за допомогою одного переходу, мережа повинна мати світлові шляхи між усіма парами джерело-одержувач. У мережі з  $N$  вузлами це означає, що нам потрібно щонайменше  $n(n-1)$  світлових шляхів, і, як правило, це не практично.

У реальних мережах лише деякі пари кінцевих вузлів з'єднані світловим шляхом. Якщо два кінцеві вузли  $X$  і  $Y$  не з'єднані світловим шляхом, потрібно визначити послідовність вузлів  $A_1, A_2, \dots, A_k$  так, щоб існував світловий шлях від вихідного вузла  $X$  до вузла  $A_1$ , шлях світла від вузла  $A_1$  до вузла  $A_2, \dots$ , світловий шлях від вузла  $A_k$  до вузла призначення  $Y$ . Зв'язок від  $X$  до  $Y$  може бути здійснений за допомогою кількох переходів. Перший перехід полягає в тому, щоб надіслати дані з вихідного вузла  $X$  до вузла  $A_1$ .

Наступним переходом є відправлення даних з вузла  $A_1$  до вузла  $A_2$ . Цей процес триває до останнього переходу, де дані надсилаються з вузла  $A_k$  до вузла призначення  $Y$ . Такі мережі часто називають мережами з кількома переходами.

У зв'язку з цим поняття логічної топології, представлені графом  $G$ , є корисним де:

- вузли в графі  $G$  є кінцевими вузлами у фізичній мережі;
- якщо пара вузлів  $(i, j)$  з'єднана світловим шляхом, то в графі  $G$  є орієнтоване ребро від  $i$  до  $j$ .

З точки зору графа  $G$ , щоб знайти шлях для вихідного вузла  $X$  для зв'язку з вузлом призначення  $Y$ , необхідно знайти шлях  $X \rightarrow A_1 \rightarrow A_2 \rightarrow \dots \rightarrow A_k \rightarrow Y$ . Відтепер будемо посилалися на ребро в логічній топології як на логічне ребро. Наприклад, як показано на рисунку 2.1 (б), ребро  $(1 \rightarrow 3)$  є логічним ребром.

При розробці схеми для врахування всіх запитів на зв'язок в оптичній мережі необхідно вирішити, які кінцеві вузли потрібно з'єднати світловим шляхом. Іншими словами, необхідно спроектувати логічну топологію для заданої фізичної топології.

## 2.2 Несправності в мережах WDM

Пошкодження оптоволокна, як згадувалося раніше, є найпоширенішою причиною збою мережі та може призвести до збою всіх світлових шляхів, що використовують несправне волокно.

Кінцевим результатом є значна втрата даних. Для того, щоб мати надійну та живучу мережу WDM, під час проектування логічної топології потрібно не лише пристосуватись до трафіку, указанного користувачем, але й мати можливість справлятися з несправностями, наприклад, мати стратегії для управління несправностями. Оскільки пошкодження одного волокна є переважною формою збою в оптичних мережах, у цій галузі було зосереджено багато досліджень.

Існує два основних підходи до управління несправностями: захист і відновлення. Під час відновлення резервна ємність, доступна після виникнення несправності, використовується для перенаправлення порушеного трафіку. У захисті резервна ємність резервується під час встановлення виклику. Для кожного первинного світлового шляху створюється резервний світловий шлях, який зберігається в резерві під час проектування логічної топології. За відсутності збоїв у зв'язку використовується тільки первинний світловий шлях.

Коли виникає збій через розріз волокна, це впливає на кілька основних шляхів світла, оскільки вони використовують несправне волокно. Зв'язок, який використовує ці пошкоджені світлові шляхи, відновлюється за допомогою відповідних резервних світлових шляхів.

### 2.3 Передача даних у мережі WDM за умови відсутності збоїв

Враховуючи вимоги користувача до передачі даних, мета полягає в тому, щоб визначити стратегію передачі даних таким чином, щоб обсяг трафіку на найбільш завантаженому передавачі або приймачі був якомога меншим.

З точки зору логічної топології, це означає мінімізацію трафіку на логічному каналі, що передає максимальний трафік. Трафік на логічному каналі, що передає максимальний трафік, часто називають перевантаженням мережі.

Враховуючи вимоги користувача до передачі даних і опис фізичної топології, необхідно вирішити наступні підзадачі:

- визначити оптимальну логічну топологію  $G$  для мережі;
- для того, щоб мати дійсний шлях світла, що відповідає кожному логічному каналу в  $G$ , визначається відповідна довжина хвилі та маршрут через фізичну топологію, що називається проблемою призначення маршруту та довжини хвилі (RWA);
- для кожного запиту на передачу даних, скажімо, від вузла джерела  $z$  до вузла призначення  $j$ , знаходиться оптимальний спосіб маршрутизації даних від  $i$  до  $j$ , використовуючи шлях з кількома переходами в логічній топології, щоб мінімізувати перевантаження.

Ці підпроблеми взаємопов'язані, оскільки мета – мінімізувати перевантаження, а останню підзадачу неможливо вирішити, доки не буде вирішено підзадачу 1. Підзадача 1, у свою чергу, вимагає вирішення підзадачі 2, щоб переконатися, що логічна топологія справді реалізована.

RWA є основною проблемою керування передачею даних у мережі WDM. Якщо в усіх волокнах існує достатня кількість каналів довжини хвилі, шляхи встановлюються для з'єднання кожного кінцевого вузла без проблем RWA. Але, на жаль, на практиці волокна мають обмежену кількість довжин хвиль, тому RWA стає складною мережевою проблемою для маршрутизації даних у мережі WDM.

RWA класифікується на два типи проблем залежно від попиту на трафік: статичні та динамічні. Для статичного трафіку запити на з'єднання відомі заздалегідь, і для цих з'єднань налаштовуються шляхи. Мета полягає в тому, щоб мінімізувати кількість довжин хвиль, запитуваних для даної мережі. Проблема RWA для статичного трафіку відома як Static Lightpath Establishment (SLE).

З іншого боку, для динамічного трафіку світловий шлях встановлюється для кожного запиту на з'єднання, коли надходить запит на з'єднання. Мета полягає в тому, щоб встановити світловий шлях і призначити довжини хвиль, щоб максимізувати ймовірність підключення або покращити ефективність мережі. Проблема RWA для динамічного трафіку відома як Dynamic Lightpath Establishment (DLE).

#### 2.4 Передача даних у мережі WDM за допомогою схеми захисту

Мета в цьому випадку така ж, як і в попередньому пункті – визначити стратегію передачі даних, щоб мінімізувати перевантаження мережі. Єдина відмінність полягає в тому, що мається схема керування відмовами, яка подбає про окремі обриви волокна в мережі. Враховуючи вимоги користувача до передачі даних і опис фізичної топології, підпроблеми, які необхідно вирішити, подібні до тих, які були описані в попередньому розділі, і такі:

- визначити оптимальну логічну топологію  $G$  для мережі;
- відповідно до кожного логічного каналу в  $G$  визначається відповідна

довжина хвилі та маршрут через фізичну топологію для основного та резервного світлового шляху (волокна, які використовуються в основному світловому шляху, мають відрізнятися від волокон, що використовуються в резервному світловому шляху);

- для кожного запиту на передачу даних від вузла джерела  $i$  до вузла призначення  $j$ , знаходиться оптимальний спосіб маршрутизації даних від  $i$  до  $j$ , використовуючи шлях з кількома переходами в логічній топології, щоб мінімізувати перевантаження.

Ці підпроблеми знову взаємопов'язані з тих самих причин, які згадані в попередньому пункті.

## 3 ОПТИЧНІ МЕРЕЖІ WDM

Мультиплексування за довжиною хвилі (WDM) розділяє величезну пропускну здатність волокна (потенційно кілька десятків терабіт на секунду) на багато несучих довжин хвиль (канали WDM) [4]. Передачі на різних довжинах хвиль поєднуються в одне волокно за допомогою мультиплексорів довжин хвиль.

Кількість довжин хвиль, які може передати певне волокно, залежить від характеристик волокна та інших технологічних обмежень, таких як чутливість оптичних компонентів, вплив перехресних перешкод на волокна. Завдяки швидкому розвитку технологій WDM потужність передачі по одному волокну дедалі зростає. Крім того, численні вдосконалення оптичних компонентів, таких як оптичні передавачі, приймачі, фільтри, підсилювачі та мультиплексори додавання/відведення, тепер зробили можливим мати до 200 довжин хвиль на одному волокні, кожна з яких підтримує швидкість передачі даних від 2,5 до 10 гігабіт на секунду.

### 3.1 Основні пристрої WDM

Оптичне перехресне з'єднання (Optical cross-connect, ОХС): цей пристрій забезпечує можливість маршрутизації або перемикання окремих оптичних сигналів на будь-якому зі своїх вхідних портів на будь-який вихідний порт, не вимагаючи від сигналу жодного оптико-електронного перетворення.

Якщо ОХС оснащений перетворювачами довжини хвилі, він також може змінювати довжину хвилі вхідного оптичного сигналу, коли він проходить через перемикач [6] (рисунок 3.1). Використання ОХС на проміжних вузлах дозволяє встановити повністю оптичний канал зв'язку (світловий шлях) між двома кінцевими вузлами.

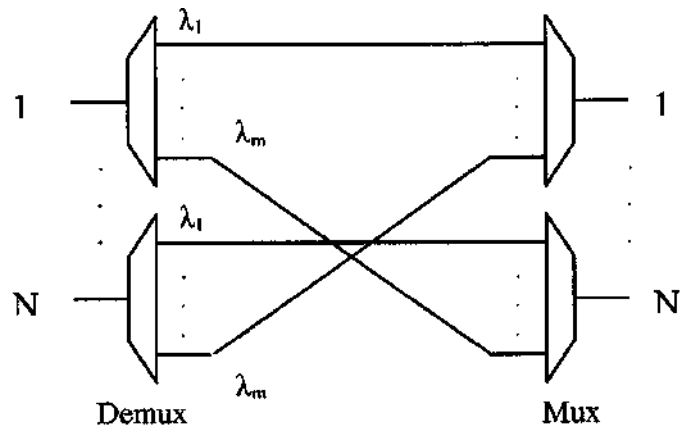


Рисунок 3.1 – Оптичне крос-з'єднання (ОХС)

Мультиплексор: цей пристрій поєднує кілька оптичних сигналів на різних довжинах хвиль несучої, щоб їх можна було передавати по одному волокну.

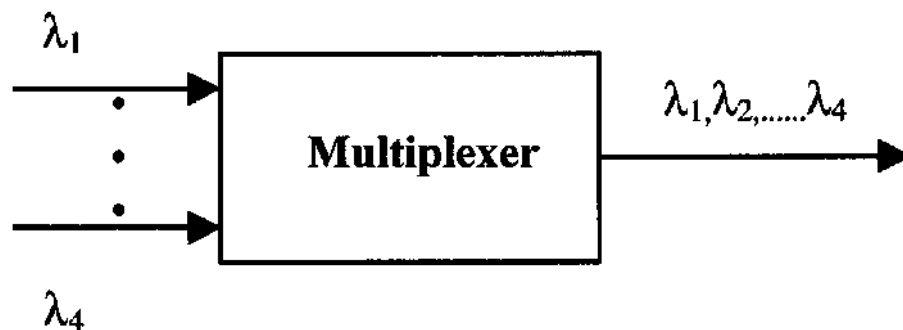


Рисунок 3.2 – Мультиплексор

Демумльтиплексор: цей пристрій розділяє кілька оптичних сигналів на різних довжинах хвиль несучої на одному волокні на сигнали на кількох волокнах, де кожне волокно передає лише один сигнал на різній довжині хвилі.

Оптичний мультиплексор додавання/виведення (OADM): Цей пристрій дозволяє вибірково додавати або відкидати один або більше оптичних сигналів, які мають певну довжину хвилі несучої, у вузлі, і дозволяє іншим

оптичним сигналам проходити через вузол без впливу. Як показано на рисунку 3.4, через OADM  $\lambda_3$  додається, а  $\lambda_4$  відкидається,  $\lambda_1$  і  $\lambda_8$  пропускаються оптично.

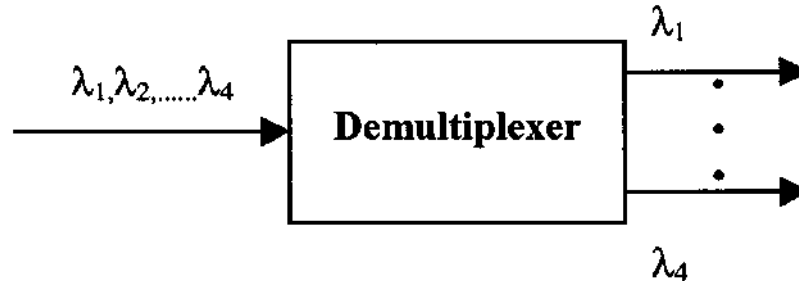


Рисунок 3.3 – Демультимплексор

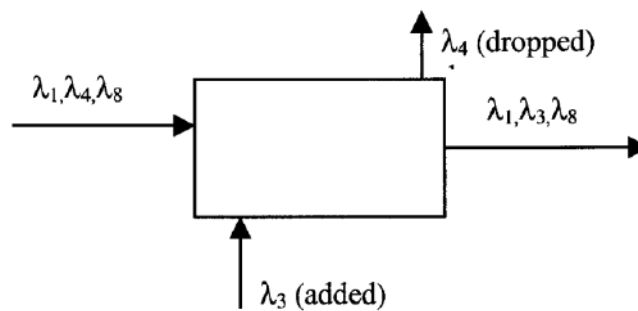


Рисунок 3.4 – Оптичний мультиплексор додавання/виведення (OADM)

### 3.2 Маршрутизація за довжиною хвилі

Мережі з маршрутизацією за довжиною хвилі – це оптичні мережі, у яких маршрутизація та комутація оптичних сигналів здійснюється на основі їхніх довжин хвиль. Відповідно до [6], мережа WDM була започаткована концепцією маршрутизації за довжиною хвилі. Принцип полягає в тому, що кожен високошвидкісний потік даних пов'язаний із певною довжиною хвилі оптичної несучої, яка безпосередньо маршрутизується через мережу на основі її довжини хвилі від кінцевого вузла, який є джерелом потоку даних,

до кінцевого вузла, який є пункт призначення потоку даних без перетворення в електронний сигнал на проміжних вузлах, які є вузлами маршрутизатора. Це стало можливим завдяки розробці повністю оптичних маршрутизаторів, які можуть обробляти багато каналів WDM одночасно.

Важливою особливістю оптичної мережі з маршрутизацією по довжині хвилі є прозорість формату сигналу. Кожен світловий шлях може передавати різні типи потоків даних одночасно з різними бітовими швидкостями. Крім того, довжини хвиль можна просторово повторно використовувати в мережі. Це означає, що однакова довжина хвилі може бути розподілена на два або більше окремих шляхів світла, якщо ці шляхи світла не мають спільного волокна.

Таким чином, хоча кількість доступних довжин хвиль може бути обмеженою, кількість шляхів світла, які може підтримувати мережа, зазвичай набагато більша.

Нарешті, мережа може бути налаштована таким чином, щоб у разі збою світлові шляхи могли бути автоматично перенаправлені на альтернативні шляхи. Це забезпечує високий ступінь надійності в мережі.

### 3.3 Мережі з одним і кількома переходами

Топологія оптичної мережі може бути з одним переходом або з кількома переходами (перехід визначається як логічне з'єднання між двома вузлами без будь-якої оптико-електронної обробки на проміжних пристроях, якщо такі є).

У топології з одним переходом оптичні сигнали переносять інформацію від вузла джерела до вузла призначення без будь-якого оптико-електронного перетворення. Слід зазначити, що інформація від джерела до вузла призначення може направлятися через кілька оптичних пристроїв (наприклад, пасивний зіркоподібний зв'язувач, повністю оптичний мультиплексор, повністю оптичний маршрутизатор), однак у цих пристроях

не відбувається електронної обробки. У топології з кількома переходами трафік від вузла джерела до вузла призначення може проходити через деякі проміжні вузли, де він піддається оптоелектронним перетворенням, обробляється та надсилається через інші з'єднання, можливо, на іншій довжині хвилі.

Наприклад, на рисунку 3.5 вузол 1 може досягти вузла 2 за один перехід (стрибок). Таким чином, незалежно від того, що вузол 1 передає на довжині хвилі  $\lambda_1$  досягає вузла 2 зі швидкістю світла (у волокні). З іншого боку, вузол 1 може досягти вузла 3 за два переходи. Після першого переходу сигнал від вузла 1 піддається оптико-електронному перетворенню у вузлі 2. Потім на основі адреси призначення в повідомленні, наданому вузлом 1, вузол 2 вирішує, що повідомлення має бути перенаправлено до вузла 3. вузол 2 перекодує повідомлення в оптичному домені та пересилає його до вузла 3 на довжині хвилі  $\lambda_1$ . Подібним чином, вузол 2 може досягти вузла 5 через три переходи.

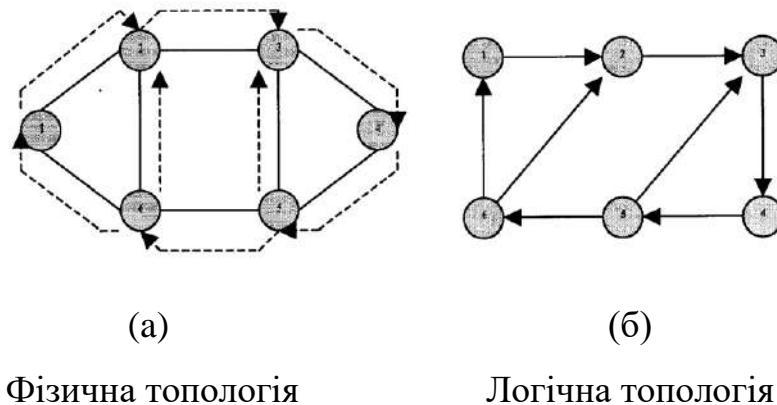


Рисунок 3.5 – Фізична та логічна топологія

Мережа WDM з  $N$  кінцевими вузлами, де весь зв'язок використовує один перехід, потребує  $n(n-1)$  світлових шляхів. Оскільки це неможливо для нетривіальних мереж, більшість мереж WDM використовують багатопрохідний режим.

### 3.4 Логічна топологія в мережах, де не враховується можливість несправностей

У даній фізичній топології можна вбудовувати різні логічні топології, використовуючи кілька паралельних каналів WDM і належним чином налаштовуючи елементи мережі. У мережі з маршрутизацією за довжиною хвилі різні конфігурації комутації маршрутизаторів за довжиною хвилі призведуть до різних логічних топологій. Наприклад, для тієї самої фізичної топології на рисунку 3.5 (а), якщо створити шляхи світла, як показано на рисунку 3.6 (а), можна отримати логічну топологію на рисунку 3.6 (б), у якій кожен вузол має світловий шлях до всіх інших вузлів.

Слід зазначити, що якщо топологія мережі відносно статична, то можна використовувати приймачі з фіксованою довжиною хвилі замість регульованих. Крім того, мережі з маршрутизацією по довжині хвилі можуть використовувати переваги повторного використання довжини хвилі.

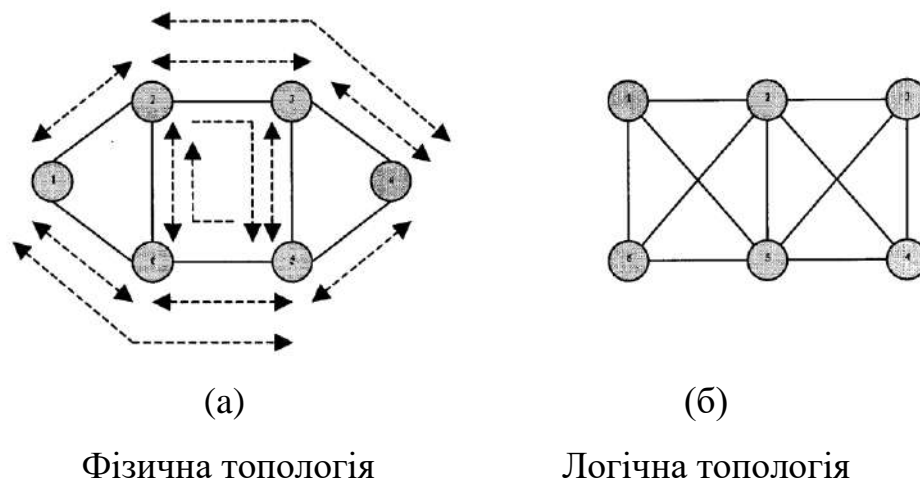


Рисунок 3.6 – Інша можлива логічна топологія

Наведене вище обговорення показує, що велика кількість логічних топологій може бути відображена в одній фізичній топології. Дуже важливим питанням є механізм, за допомогою якого можна вибрати хорошу логічну

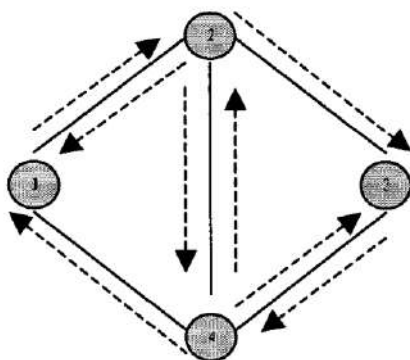
топологію. Також важливо вирішити, який саме параметр необхідно оптимізувати при проектуванні таких мереж. Деякі відповідні параметри – це діаметр логічної топології та кількість електронної обробки, яку потрібно виконати для обробки трафіку на кінцевому вузлі.

Запропоновані логічні топології можна загалом класифікувати на дві категорії: довільні та регулярні. Шаблони вузлових з'єднань у звичайних топологіях дуже систематичні та чітко визначені, що спрощує операції маршрутизації та керування.

Іншою важливою перевагою є те, що діаметр таких логічних топологій, як відомо, дуже малий. Протягом останніх кількох років було запропоновано кілька звичайних логічних топологій для оптичних мереж. Однак важко додати довільну кількість вузлів до звичайної топології і все ще зберігати свою чітко визначену структуру. Іншим істотним недоліком є те, що така структура не враховує відому динамічну схему трафіку.

У мережах з кількома переходами, які не мають попередньо визначеної топології, немає чіткої моделі підключення.

На основі трафіку між усіма парами вузлів мета розробки полягає в тому, щоб розробити логічну топологію, яка обробляє трафік найбільш ефективним способом.



(a)

Фізична топологія

	(1)	(2)	(3)	(4)
(1)	0.00	0.20	0.80	0.40
(2)	0.30	0.00	0.10	0.70
(3)	0.90	0.40	0.00	0.10
(4)	0.60	0.20	0.40	0.00

(б)

Матриця трафіку

Рисунок 3.7 – Фізична топологія та матриця трафіку

Одним із зручних способів представлення трафіку між різними парами вузлів у мережі з  $n$  вузлами є використання матриці ( $n \times n$ ). Така матриця називається матрицею трафіку. Записи в матриці трафіку зазвичай отримують шляхом спостереження за довгостроковими моделями трафіку в мережі.

Мережа з чотирма вузлами показана на рисунку 3.7 (а), а можлива матриця трафіку для цієї мережі показана на рисунку 3.7 (б). Запис  $t_{ij}$  представляє обсяг трафіку від вузла  $i$  до вузла  $j$ .

Зручно нормалізувати трафік між будь-якою парою вузлів і представити його як частку пропускної здатності світлового шляху. Наприклад, кількість трафіку від вузла 1 до вузла 3 задано як 0,80 у матриці трафіку, показаній на рисунку 3.7 (б). Якщо світловий шлях у мережі може передавати 2,5 Гбіт/с, це означає, що фактичний трафік становить 2,0 Гбіт/с. Цей запис  $t_{ij}$  представляє довгостроковий середній (або піковий) трафік між парою вузлів ( $i, j$ ).

З матриці трафіку  $A$  на рисунку 3.7 (б) можна побачити, що існує великий трафік між вузлом 1 і вузлом 3 ( $t_{13} = 0,80$  і  $t_{31} = 0,90$ ), але, як показано на рисунку 3.7 (а), немає світлового шляху від вузла 1 до вузла 3, тому трафік досягає вузла 3 за два переходи. Він має пройти оптико-електронне перетворення у вузлі 2. Крім того, від вузла 1 до вузла 2 він використовує світловий шлях  $p_{12}$ , а від вузла 2 до вузла 3 – світловий шлях  $p_{23}$ , трафік на цих двох шляхах буде  $p_{12}: t_{12} + t_{23} = 1,00$  і  $p_{23}: t_{23} + t_{31} = 0,90$ , що робить ці два шляхи дуже перевантаженими, тоді як інші шляхи, такі як  $p_{41}$ , несуть лише менше половини пропускної здатності шляху. Для трафіку  $t_{31}$ , двома можливими маршрутами є  $(3 \rightarrow 4 \rightarrow 1)$  і  $(3 \rightarrow 4 \rightarrow 2 \rightarrow 1)$ , але жоден із них не може нести  $t_{31}$ , що дорівнює 0,90, тому що хоча шлях  $p_{34}$  має 0,90 резервної ємності, але світловий шлях  $p_{41}$  має лише 0,40 вільної ємності, а для світлових шляхів  $p_{42}$  і  $p_{21}$  ніхто не має достатньої вільної ємності для перенесення  $t_{31}$ . Тому наведена вище логічна схема не дуже хороша для цієї матриці трафіку.

Кінцеві вузли повинні перетворювати оптичний сигнал в електронний і навпаки, а мережа з нижчим значенням перевантаження означає менше перетворення на кінцевих вузлах, що призводить до меншої складності електронних схем для перетворення.

Для даної топології фізичної мережі мінімізація рівня перевантаження на основі матриці трафіку є хорошою метрикою для розробки логічної топології.

Для оптичної мережі WDM, якщо ми зможемо встановити світловий шлях між кожною парою джерело – пункт призначення, ми одразу отримаємо мережу з мінімальним значенням перевантаження. У такому рішенні немає оптико-електронного перетворення всередині мережі, за винятком кінцевих вузлів.

Якщо таке рішення можливе для мережі, воно усуває електронні вузькі місця та називається повністю оптичною мережею. Для повністю оптичної мережі, що має  $n$  вузлів, повністю зв'язана логічна топологія потребує  $n(n-1)$  шляхів та кількість довжин хвиль що необхідні для підтримки  $n(n-1)$  шляхів можуть бути надзвичайно великими, якщо  $n$  велике. Тому повністю оптична логічна топологія, як правило, неможлива.

Для того, щоб якнайкраще використати потужність мережевої інфраструктури WDM, найважливішим завданням є ідентифікація найкращої можливої логічної топології.

Для даної фізичної мережі найкраща логічна топологія визначається шаблоном потоків трафіку в мережі. Звичайні методи проектування логічної топології були зосереджені на максимізації загального обсягу трафіку в мережі або мінімізації перевантаження в мережі [7]. Протягом багатьох років цю проблему активно вивчали багато дослідників і було запропоновано багато підходів.

Ці підходи можна класифікувати на два типи: підходи MILP (змішане цілочисельне лінійне програмування) та евристичні підходи.

### 3.4 Підходи MILP

Багато дослідників розглядають задачу проектування логічної топології як задачу оптимізації. Вони сформулювали набір ефективного змішаного цілочисельного лінійного програмування та використовували інструменти цілочисельного лінійного програмування для його вирішення. Для обмеженої кількості передавачів і приймачів у кожному вузлі, з'єднаних волоконно-оптичними лініями, які підтримують обмежену кількість довжин хвиль, і заданої матриці трафіку, елементи якої відповідають (середньому) трафіку, яким обмінюються всі пари вузлів джерело – пункт призначення, бувають такі цілі оптимізації:

- мінімізація середньої кількості переходів;
- мінімізація максимального рівня перевантаження в мережі;
- мінімізація середньої затримки пакетів на вузлах.

Кількість переходів, максимальний рівень перевантаження та середня затримка пакетів взаємопов'язані.

Наприклад, мінімізація середньої кількості переходів також призводить до зменшення середнього потоку трафіку на каналах, і, отже, знижує рівень заторів.

Це опосередковано мінімізує затримку в черзі. Прямо чи опосередковано найбільш логічна конструкція топології спрямована на мінімізацію максимального рівня перевантаження в мережі [8].

Використовуючи підхід MILP, в деяких випадках об'єднують проблему проектування логічної топології та задачу маршрутизації. Оскільки об'єднану задачу складно розв'язати з точки зору обчислень, особливо для великих мереж, прийнято розділяти її на дві підпроблеми: проблему проектування логічної топології та проблему маршрутизації та вирішувати ці дві проблеми незалежно.

Проблема проектування логічної топології формулюється як проблема MILP, а задача маршрутизації – як проблема LP.

### 3.5 Евристичні підходи

Рішення MILP не піддаються обчисленням для практичних мереж, оскільки вони генерують велику кількість обмежень і цілочисельних змінних. Для вирішення цієї проблеми було запропоновано багато евристичних підходів. Загалом існує чотири типи евристичних підходів [8]:

1 Евристичні рішення проблеми MILP – генетичні алгоритми, моделювання відпалу, методи локального пошуку зі змінною глибиною. Недолік, полягає в тому, що "якість рішення не задовольняє" [8].

2 Максимізація потоків трафіку з одним переходом – алгоритм проектування логічної топології максимізації з одним переходом (SMLTDA). Але якщо ступінь входу/виходу вузлів невеликий, а матриця трафіку мала, це може призвести до незв'язаної логічної топології. Жадібний алгоритм проектування логічної топології (GLTDA), він використовує декілька паралельних світлових шляхів і намагається ввести логічні зв'язки між вузлами, які обмінюються великою кількістю трафіку.

3 Евристична максимізація потоків трафіку з одним і кількома переходами – є два типи:

- AL (додавання світлових шляхів) – алгоритм проектування логічної топології із збільшенням кількості переходів (I-MLTDA), він починається з незв'язаної топології та послідовно додає світлові шляхи, намагаючись на кожному кроці мінімізувати рівень перевантаження в мережі;

- RL (Remove lightpaths) – алгоритм проектування логічної топології з декількома переходами (D-MLTDA), він починається з повністю зв'язаної топології та послідовно видаляє світлові шляхи, вибираючи ті, які передають найменші потоки трафіку.

4 Алгоритми, засновані на прийнятті попередньо встановленої регулярної логічної топології – алгоритм проектування попередньо встановленої логічної топології (PLTDA) і алгоритм проектування випадкової логічної топології (RLTDA).

### 3.6 Живучість і управління несправностями

Сучасні мережі WDM здатні забезпечувати дуже високу пропускну здатність із швидкістю передачі даних від 2,5 до 10 гігабіт на секунду на кожній довжині хвилі несучої та 100 або більше довжин хвиль на кожному волокні.

Тому, якщо мережа не працює через збій, навіть на кілька секунд, це зазвичай призводить до величезних втрат даних. Таким чином, високодоступна мережа, яка є дуже стійкою до мережевих збоїв, є дуже важливою вимогою для мереж WDM. Термін живучість означає здатність мережі продовжувати надавати послуги за наявності збоїв. Живучість може розглядатися на багатьох різних рівнях мережевої ієрархії, при цьому кожен рівень реалізує власні механізми захисту.

У цій кваліфікаційній роботі увага зосереджується на роботі протоколу IP через мережі WDM. У мережах, зоснованих на IP через WDM механізми захисту та відновлення можуть бути реалізовані на оптичному (WDM) рівні або на рівні IP. На рівні WDM для кожного основного шляху резервний шлях і відповідна довжина хвилі резервуються заздалегідь під час налаштування маршруту.

На рівні IP може бути забезпечено надлишкове забезпечення пропускну здатності мережі, щоб після збою можна було динамічно виявляти альтернативні маршрути через мережу, використовуючи резервну потужність.

Швидкість відновлення, тобто як швидко мережа може бути відновлена після збою, є ключовим критерієм при розробці надійної та живучої оптичної мережі. ІТУ-Т (Міжнародний союз телекомунікацій із телекомунікацій) визначає, як розраховується час відновлення [9], рисунок 3.8.

У разі збою вузлу поруч із несправним компонентом у мережі потрібен деякий час, щоб виявити цей збій. Цей інтервал часу називається часом виявлення (T<sub>1</sub>).

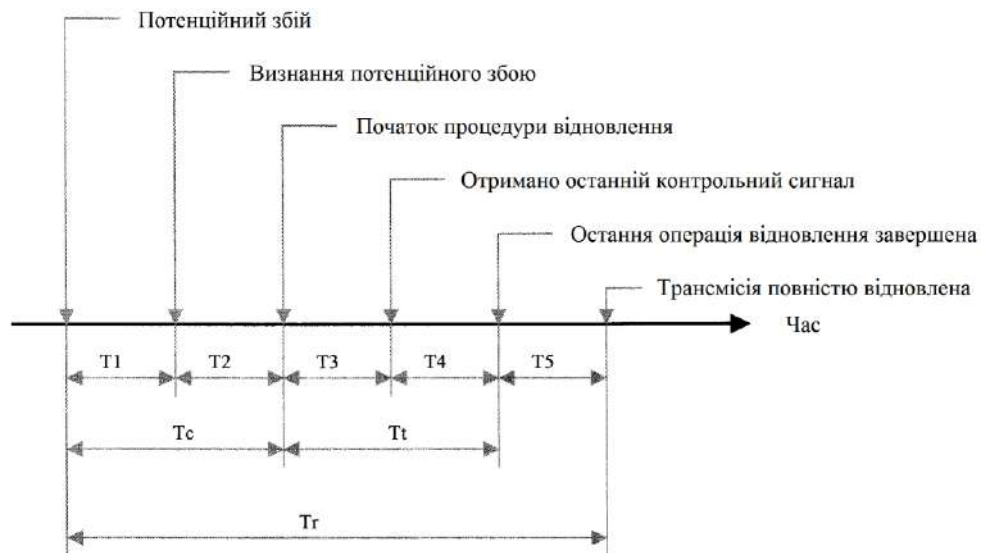


Рисунок 3.8 – Час відновлення відповідно до ITU-T

Через деякий час збій підтверджується, і починається процедура відновлення. Інтервал часу між цим моментом часу і розпізнаванням несправності називається часом очікування ( $T_2$ ).

Під час відновлення передаються сигнали керування та обробляються отримані сигнали. Цей проміжок часу називається часом процедури відновлення ( $T_3$ ).

Час, необхідний для обробки останнього отриманого керуючого сигналу, називається часом передачі відновлення ( $T_4$ ).

На останньому кроці може бути завершена перевірка операції захисного перемикавання або деяка повторна синхронізація. Цей інтервал часу називається часом відновлення ( $T_5$ ).

Для всіх цих п'яти часових інтервалів час між виникненням несправності та підтвердженням несправності також називається часом підтвердження ( $T_c$ ).

Час між підтвердженням несправності та завершенням останньої операції відновлення також називається часом передачі ( $T_t$ ). Загальний проміжок часу від виникнення несправності до повного відновлення називається часом відновлення ( $T_r$ ).

У телекомунікаційній галузі традиційно потрібен час відновлення близько 50 мс.

Методи відновлення на рівні IP зазвичай призводять до часу відновлення, який значно довший (порядку секунд). У той час як час відновлення за допомогою методів захисту оптичного рівня становить близько мілісекунд, що може значною мірою мінімізувати втрати даних. Крім того, оптичний рівень може ефективно мультиплексувати ресурси захисту (такі як резервні довжини хвилі та волокна) між кількома мережевими програмами вищого рівня, а також живучість на оптичному рівні забезпечує захист протоколів вищого рівня, які можуть не мати вбудованого захисту. Як правило, схеми відновлення більш ефективні у використанні потужностей завдяки спільному використанню резервних потужностей, тоді як схеми захисту мають швидший час відновлення та забезпечують гарантії відновлення.

### 3.7 Відновлення

Відновлення може бути використано для забезпечення або більш ефективних маршрутів після завершення захисту, або додаткової стійкості проти подальших несправностей до усунення першої несправності. Зазвичай механізм відновлення досить повільний (від секунд до хвилин) і може бути обчислений централізованою системою керування. Відповідно до поточної технології відновлення кільцевої оптичної мережі займає близько 50 мс, а для наскрізної сітчастої оптичної мережі потрібно кілька сотень мс.

Відновлення забезпечує другий крок захисту від збоїв мережі. Відновлення може обробляти як збої зв'язку, так і збої вузла чи кількох одночасних збоїв. Зазвичай воно використовується в сітчастих топологіях. У динамічному відновленні резервна ємність, доступна в мережі, використовується для відновлення служб, на які впливає збій. Відновлення може бути реалізовано централізованим або розподіленим способом.

### 3.8 Захист

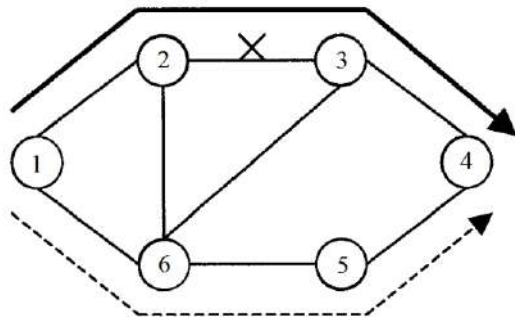
Фізичний рівень близький до більшості поширених збоїв у мережі, таких як обрив оптоволокна. Механізми живучості на оптичному рівні передбачають виявлення цього та виконання простого перемикавання, щоб перенаправити трафік через альтернативний шлях.

Захист як механізм нижнього рівня забезпечує перший крок захисту від типових збоїв в оптичній мережі. Захист залежить від топології та технології і пропонує швидке відновлення, але захист може бути не в змозі захистити від збою вузла або кількох несправностей. Зазвичай для цілей захисту виділяється фіксована кількість пропускної здатності, щоб швидко передавати трафік від несправних об'єктів до робочих. Ці ресурси можуть бути призначені для кожного сценарію збою або можуть бути спільними для різних сценаріїв збою. Іншими словами, залежно від того, як використовується виділена попередньо призначена ємність, існує два механізми захисту, виділений або спільний захист відповідно. Крім того, залежно від того, як забезпечити захист, існує два типи підходів до захисту: захист шляху і захист каналу відповідно.

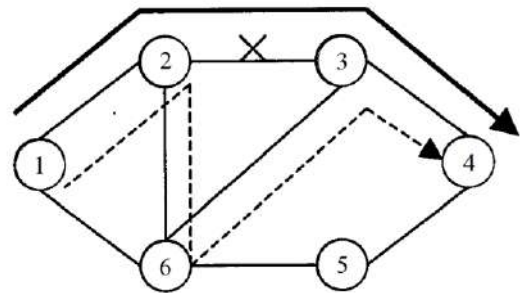
У захисті шляху резервний шлях, який також є шляхом наскрізного відновлення, резервується під час встановлення зв'язку. У разі збою відновлення трафіку виконується вузлами джерела та одержувача вздовж резервного шляху між вузлами.

У захисті каналу (також називається захистом лінії) резервні шляхи резервуються навколо кожного каналу основного шляху. Відновлення трафіку виконується вузлами, суміжними з аварією, напр. кінцеві вузли невіддаленого з'єднання динамічно виявляють резервний шлях навколо з'єднання, усі з'єднання, які перетинають збійне з'єднання, перенаправляються на резервний шлях навколо цього з'єднання. Наприклад, як показано на рисунку 3.9, для основного шляху (1—>2—>3—>4), коли збій виникає на (2—>3), у схемі захисту шляху трафік перемикається на

резервний шлях (1→6→5→4), рисунок 3.9 (а), і в схемі захисту каналу трафік буде спрямовано на (1→2→6→3→4), рисунок 3.9 (б), тут канал (2→6→3) є резервним каналом (2→3).



(а) Захист шляху



(б) Захист каналу

Рисунок 3.9 – Захист шляху та захист каналу

### 3.8.1 Захист виділеного шляху

У захисті виділеного шляху 50% усієї ємності в мережі зарезервовано для цілей захисту. Очевидно, що спеціальний захист забезпечує найвищий рівень захисту, але призводить до неефективного використання мережі. Типовим прикладом є архітектура UPSR (Unidirectional Path Switched Ring), що використовується в кільцевих мережах SONET/SDH.

Спеціальний захист шляху може бути двох типів:

- 1+1 Захист;
- 1:1 Захист.

При використанні типу захисту 1+1 Захист трафік, який потрібно захистити, надсилається одночасно двома паралельними шляхами (зазвичай непересічними маршрутами) від джерела до пункту призначення. Під час нормальної роботи вузол призначення отримує два ідентичних потоки трафіку та просто вибирає один із них, відстежуючи якість сигналу або частоту бітових помилок. Коли на вибраному шляху виникає збій, пункт

призначення просто перемикається на інший шлях. Коли початковий робочий шлях відновлено, нічого робити не потрібно, оскільки відремонтований шлях тепер стає захисним. Цей вид захисту не потребує сигналізації захисту, оскільки вузол призначення може самостійно впоратися зі збоєм, вузлу джерела не потрібно нічого робити, але завжди направляти трафік на альтернативний шлях щоразу, коли він надсилає трафік до пункту призначення. Це робить захист 1+1 захист дуже простим у реалізації, а час його відновлення дуже короткий. Але його недолік очевидний, він витрачає цінну пропускну здатність.

Тип захисту 1:1 захист все ще використовує два паралельні шляхи від джерела до пункту призначення, але різниця полягає в тому, що під час нормальної роботи трафік передається лише по одному шляху (основному шляху) одночасно, він не надсилає трафік на альтернативний шлях (резервний шлях).

Коли виникає збій на основному шляху, вузли джерела та призначення перемикаються на резервний шлях. В односпрямованих системах передачі, оскільки трафік надсилається лише в одному напрямку через волокно, вихідний вузол не може сам по собі усвідомити збій волокна, тому вузол призначення повинен повідомити джерело про перехід на резервний шлях захисту, і, отже, для цього потрібен протокол сигналізації, який називається автоматичним захисним перемиканням (APS). У разі використання систем двонаправленої передачі (де трафік передається в обох напрямках через волокно) обидва кінці можуть виявити збій, тому сигналізація не потрібна. Перевага цього типу захисту полягає в тому, що під час нормальної роботи невикористаний шлях захисту може бути використаний для передачі низькопріоритетного трафіку. Це забезпечує краще використання мережевих ресурсів, ніж захист 1+1 захист.

У разі збою високопріоритетний трафік перемикається на шлях захисту, а низькопріоритетний трафік відкидається. Слід зазначити, що якщо немає додаткового передавача чи приймача для обробки

низькопріоритетного трафіку, ця перевага зникає. Коли початковий робочий шлях (первинний шлях) відновлюється, бажано, щоб трафік знову перемикався на основний шлях, щоб можна було обробляти нові збої, а трафік з низьким пріоритетом міг продовжувати використовувати шлях захисту. Недоліком є те, що для односпрямованої системи зв'язку вона вимагає сигналізації та додає накладні витрати на зв'язок, тому вона має довший час відновлення.

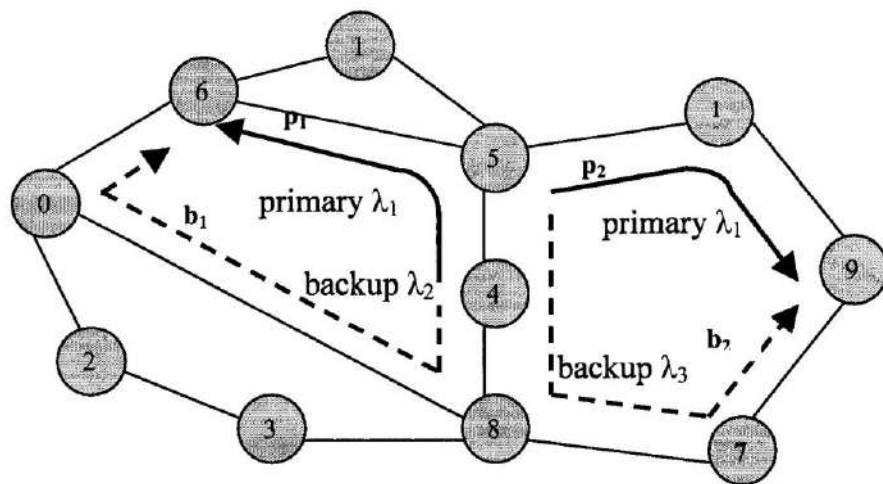


Рисунок 3.10 – Спеціальний захист шляху (1:1 Захист)

На рисунку 3.10 основний шлях  $p_1$  (4→5→6) використовує довжину хвилі  $\lambda_1$ , а його відповідний резервний шлях  $b_1$  (4→8→0→6) використовує довжину хвилі  $\lambda_2$ ; інший первинний світловий шлях  $p_2$  (5→10→9) також використовує довжину хвилі  $\lambda_1$ . Це можливо, оскільки  $p_1$  і  $p_2$  не мають спільного волокна, а відповідний резервний шлях  $b_2$  (5→4→8→7→9) використовує довжину хвилі  $\lambda_3$ . Резервний шлях  $b_2$  не може використовувати довжину хвилі  $\lambda_2$ , оскільки він спільний проміжок (4→8) із  $b_1$ , а довжина хвилі  $\lambda_2$  краю (4→8) використовується  $b_1$ .

Як показано на рисунку 3.10, у захисті виділеного шляху ресурси, які використовуються одним шляхом резервного копіювання, не можуть спільно використовуватися іншими шляхами резервного копіювання.



Захист спільного шляху забезпечує більш ефективне використання мережевих ресурсів (пропускної здатності), ніж захист виділеного шляху.

Як згадувалося вище, WDM передає оптичний сигнал з використанням різних довжин хвиль через одне оптичне волокно, щоб технологія була більш ефективною з точки зору використання ємності. Однак WDM створює проблеми з маршрутизацією та розподілом довжини хвилі (RWA), коли кілька оптичних сигналів використовують одне і те ж волокно в мережах. Світловий шлях, який є оптичним каналом, повинен використовувати ту саму довжину хвилі для передачі оптичних сигналів від вузла джерела до вузла призначення. Це міркування відоме як обмеження безперервності довжини хвилі. Це може призвести до високої ймовірності блокування та неефективного використання каналів довжини хвилі. Щоб подолати цю проблему, використовуються перетворювачі довжини хвилі.

Для вирішення питань часу відновлення механізму захисту сітки та проблем ємності кільцевого захисту було запропоновано попередньо налаштований цикл захисту (р-цикл) [10]. Основний ідеал захисту р-циклу був розроблений для використання переваг як кільцевих, так і сітчастих механізмів захисту. Завдяки швидкому відновленню кільцевих механізмів і продуктивності сітчастого механізму. Захист р-циклу дозволяє досягти низької резервної потужності шляхом визначення відповідного набору р-циклів. Однак визначення оптимального набору р-циклів для захисту є NP-складною проблемою. Існуючі підходи до вирішення проблеми р-циклу полягають у використанні моделі цілочисельного лінійного програмування (ILP) або використовуючи евристичні методи. Метою моделі ILP є досягнення оптимального рішення з точки зору мінімізації резервної ємності при збереженні 100% захисту. Однак модель ILP стає важкорозв'язною з великомасштабними мережами, де кількість можливих змінних дуже велика.

Загалом обчислювальна складність і оптимальність рішень є двома показниками продуктивності, які висувають суперечливі вимоги в задачах оптимізації.

Одним із найважливіших питань у проблемі оптимізації мережі є використання пропускної здатності. Зазвичай це оцінюється шляхом вимірювання надмірності. Низька надмірність є більш ефективною, ніж висока надмірність через те, що висока надмірність вимагає великої резервної потужності для захисту від збоїв.

Обчислювальна складність є важливим фактором при оцінці алгоритму. Він визначається як час, потрібний алгоритму для пошуку рішення. Як обговорювалося, модель ІЛР страждає від високої обчислювальної складності в щільних мережах, навіть якщо вона обчислює оптимальне рішення. Якщо алгоритм може досягти майже оптимального рішення в межах прийнятної обчислювальної складності, він, швидше за все, буде більш бажаним рішенням проблеми.

Обчислювальна складність і оптимальність рішень зазвичай порівнюються одна з одною. Незважаючи на те, що можна досягти швидкого часу обчислення, низька надмірність не може переглядати попередню роботу, виконану з точки зору моделей ІЛР та евристичних підходів для проектування мережі р-циклу.

## 4 ЖИВУЧИСТЬ МЕРЕЖІ

В даний час оптичні мережі WDM потрібні для задоволення зростаючих потреб у мережевому трафіку. Таким чином, надійний дизайн мережі проти збою в мережі WDM стає критичною проблемою, оскільки збій каналу може призвести до величезних втрат даних. У цьому контексті кожен первинний шлях, який постраждав від збою, має бути переключений на альтернативний шлях. Основні шляхи називаються робочими, а альтернативні – резервними. Багато дослідників вивчали традиційні мережі, а також оптичні мережі, і наразі було запропоновано декілька схем живучості оптичних мереж. Техніку забезпечення живучості в оптичних мережах можна класифікувати на дві категорії залежно від того, як розподіляється резервна ємність: попередньо спланований захист і динамічне відновлення.

Класифікація методів живучості показана на рисунку 4.1. Крім того, далі розглядається захист спільного резервного шляху (Shared Backup Path Protection, SBPP) і попередньо налаштований цикл захисту (р-цикл), у якому шляхи захисту відомі заздалегідь. Зокрема, захист р-циклу стає популярним, оскільки цей метод може балансувати між використанням ємності та часом обчислення.

### 4.1 Заздалегідь спланований захист

Попередньо спланований захист резервує резервні шляхи від збою одночасно з виділенням робочих шляхів. Усі робочі канали динамічно перемикаються на зарезервовані шляхи у разі збою. Автоматичний захисний комутатор (APS), мережевий механізм Self Healing Ring (SHR) і механізм mesh-мережі належать до попередньо спланованого захисту, оскільки ці схеми заздалегідь виділяють резервні шляхи. Ця техніка захисту може запропонувати швидкий час обчислень, оскільки резервні шляхи попередньо

обчислені, а вузли на робочому шляху потрібно лише змінити на вузли на резервного шляху. Однак використання пропускну здатності є основною проблемою в цій схемі захисту, за винятком механізмів сітчастої мережі. На відміну від двох інших схем, сітчастий механізм забезпечує ефективне використання, але має більший час обчислення.

Автоматичне захисне перемикання (Automatic Protection Switch, APS). Автоматичне захисне перемикання є однією з привабливих схем забезпечення задовільної якості обслуговування в мережах. APS можна розділити на три різні архітектури залежно від призначення ресурсів захисту: 1+1 APS, 1:1 APS і 1:N APS. Ці три моделі вибирають резервний шлях після збою з'єднання. 1+1 APS — це вид спеціального захисту, 1:1 APS і 1:N APS — спільний захист.

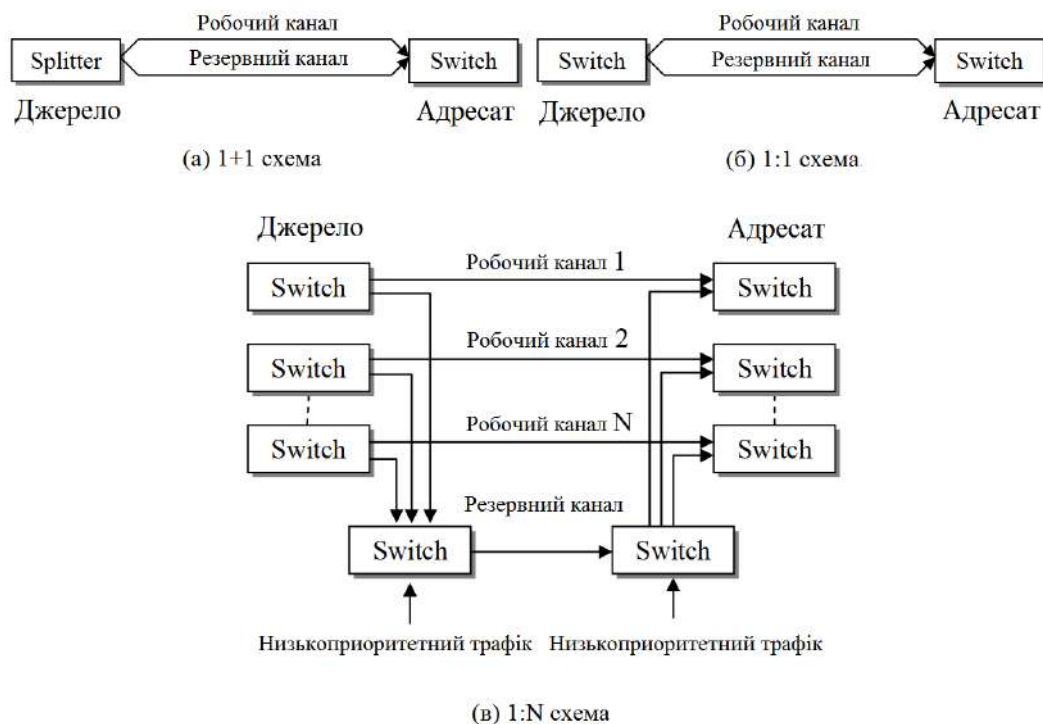


Рисунок 4.1 – Автоматичне захисне перемикання

1 + 1 APS передає сигнал інформації по робочому каналу і резервному зарезервованому каналу одночасно. Вузол призначення контролює два отримані сигнали та вибирає кращий. Після єдиного збою зв'язку приймач у

пункті призначення динамічно перемикається виключно на резервний канал для підтримки служби. Перевага цього методу полягає в тому, що він здатний запропонувати швидкий час відновлення через найшвидшу можливу швидкість перемикання, яка вимагає 60 мс або менше. Однак 1+1 APS вимагає великої резервної ємності та щонайменше 100% резервування.

Конфігурація 1:1 пропонує зарезервований канал для кожного робочого каналу. На відміну від 1+1 APS, 1:1 APS передає лише оптичний сигнал по робочому каналу, а зарезервований канал не передає цей сигнал. Крім того, зарезервований канал можна використовувати для передачі низькопріоритетного трафіку. Трафік з низьким пріоритетом перенаправляється, а робочий канал перемикається на зарезервований канал, коли виникає збій одного з'єднання. З цієї причини цей метод кращий, ніж 1+1 APS, з точки зору використання ємності мережі. З іншого боку, час його відновлення вище, ніж 1+1 APS, оскільки робочий канал повинен переключитися на зарезервований канал після виявлення збою зв'язку.

На рисунку 4.1 (в) показана схема захисту спільного каналу, конфігурація 1:N. У цій конфігурації N робочих каналів спільно використовують один зарезервований канал для відновлення після збоїв одного з N робочих каналів. Захист спільного каналу поширюється лише на збій будь-якого з N робочих каналів, і, таким чином, трафік потрібно переключити назад на основний робочий канал після усунення збою. Зарезервований канал здатний захистити лише один збій робочих каналів. Отже, конфігурація 1:N APS пропонує ефективне використання пропускної здатності, але має високу ймовірність блокування, коли захищений канал виходить з ладу. Більш популярним є розширення 1:N APS, конфігурація k:N, у якій доступні k зарезервованих каналів.

В оптичних мережах кільцева топологія, відома як кільце самовідновлення (Self Healing Ring, SHR), є найпростішою захисною мережею. SHR може легко забезпечити 100% відновлюваність і запропонувати швидший час обчислень завдяки простому управлінню і

простій політиці маршрутизації. Отже, насправді це перспективна архітектура. З іншого боку, основна проблема кільцевих мереж полягає в тому, що вони вимагають принаймні 100% резервування. Основна ідея SHR полягає в тому, що половина загальної пропускної здатності резервується для роботи, а інша половина використовується як резервна пропускна здатність. При виході з ладу однієї ланки робочий канал перенаправляється по резервному в зворотному напрямку. SHR поділяється на дві категорії залежно від напрямку руху: однонаправлені SHR (unidirectional SHR, USHR) і двонаправлені SHR (bidirectional SHR, BSHR).

Незважаючи на те, що найпопулярнішою фізичною топологією є мережа SONET Self-Healing Ring (SHR), mesh-мережі стають дедалі важливішими. Це пояснюється тим, що сітчаста топологія є кращою для великих мереж, тоді як кільцева топологія – для простих і невеликих мереж. У живучості сітчастої мережі складніше вирішити проблему оптимізації, оскільки необхідно враховувати більшу кількість каналів маршрутизації та довжин хвиль. Але живучість сітчастої мережі забезпечує високе використання пропускної здатності, оскільки резервну пропускну здатність можна використовувати для захисту від збоїв мережі, тому цей мережевий механізм є кращим з точки зору ефективності. Живучість на основі сітчастої мережі класифікується на дві категорії залежно від процесу повторного з'єднання: захист шляху та захист каналу.

У живучості на основі шляху повідомлення сповіщення про несправність інформує вихідний вузол і вузол призначення про кожен шлях, який проходить невдалу лінію. Коли відбувається збій мережі, робочий шлях перемикається на зарезервований шлях, який має бути неперетинним з відповідним робочим шляхом.

Зарезервований шлях розподіляється між парами кінцевих вузлів шляху. Наприклад, несправний канал (2-3) відновлюється зарезервованим шляхом (1-6-5-4), як показано на рисунку 4.2 (а). Живучість на основі шляху є більш ефективною у використанні пропускної здатності порівняно з

живучістю на основі каналу, оскільки вона потребує лише резервної потужності для всього зарезервованого шляху замість кожної ланки вздовж шляху.

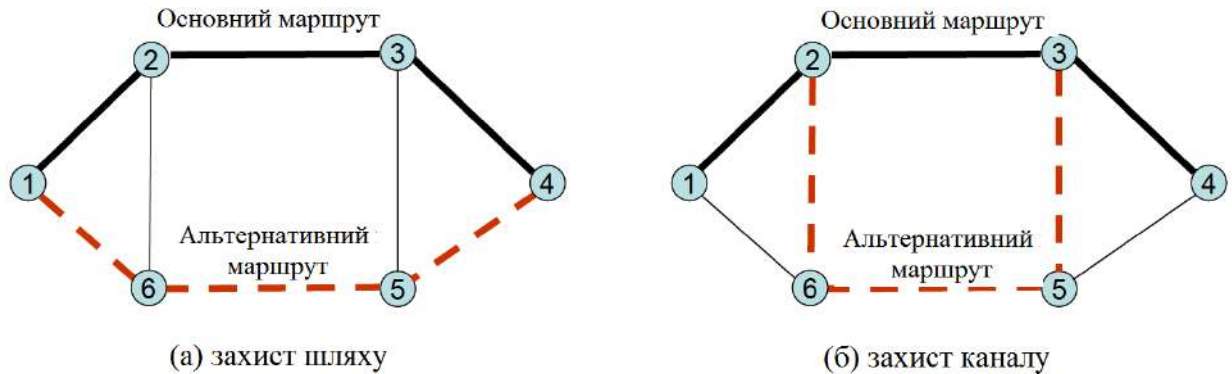


Рисунок 4.2 – Два основні механізми підвищення живучості в сітчастих мережах

У методі захисту каналу всі альтернативні шляхи вже зарезервовані під час обчислення робочого шляху. Після збою зв'язку пари кінцевих вузлів пошкодженого каналу негайно перемикаються на зарезервований шлях. У живучості на основі каналу час відновлення є швидшим, ніж живучість на основі шляху, оскільки механізму шляху потрібен довший час для створення повідомлення про помилку. Рисунок 4.2 (б) ілюструє приклад живучості на основі каналу. Пари вузлів пошкодженого каналу (2-3) автоматично перемикаються на зарезервований шлях (2-6-5-3).

#### 4.2 Динамічне відновлення

Динамічне відновлення виявляє шляхи резервної передачі в мережі після збою каналу. Як правило, це ефективніше з точки зору використання пропускної здатності, ніж попередньо запланований захист, оскільки немає необхідності резервувати резервну пропускну здатність. З іншого боку, час обчислення є складною проблемою для вирішення динамічного відновлення.

Цей механізм можна розділити на відновлення шляху та відновлення каналу. Відновлення шляху є більш ефективним, ніж відновлення каналу з точки зору використання резервної пропускної здатності. Відновлення каналу відбувається швидше, ніж відновлення шляху.

У методі відновлення шляху відразу визначається альтернативний шлях від вузла джерела до вузла призначення, коли виникає збій. Цей метод не може гарантувати 100% відновлення, оскільки шлях відновлення може бути легко заблокований, коли пропускна здатність мережі вичерпується під час збою. Крім того, час обчислення ( $T$ ) є важливим питанням, якому слід приділити належну увагу. Розраховується він як [11]:

$$T = F + (m + 1)C + 2(m + 1)D + 2mP, \quad (4.1)$$

де  $m$  – кількість переходів у шляху відновлення,  $F$  – час для виявлення збою каналу,  $C$  – час для налаштування, тестування та налаштування ОХС,  $D$  – час для обробки повідомлення на вузлі, а  $P$  – затримка розповсюдження на кожному каналі.

Під час відновлення каналу резервна пропускна здатність резервується на момент збою. Динамічно виявляється резервний канал навколо суміжних вузлів пошкодженого каналу.

Таким чином, це забезпечує ефективне використання пропускної здатності. Збій мережі можна усунути через будь-яку кількість зарезервованих каналів. Необхідно визначити час обчислення, щоб забезпечити якість обслуговування при відновленні каналу, а формулювання часу наведено як [11]:

$$T = F + nP + (n + 1)D + (m + 1)C + 2mP + 2(m + 1)D, \quad (4.2)$$

де  $n$  – о кількість переходів від вихідного вузла пошкодженого каналу до вихідного вузла джерела.

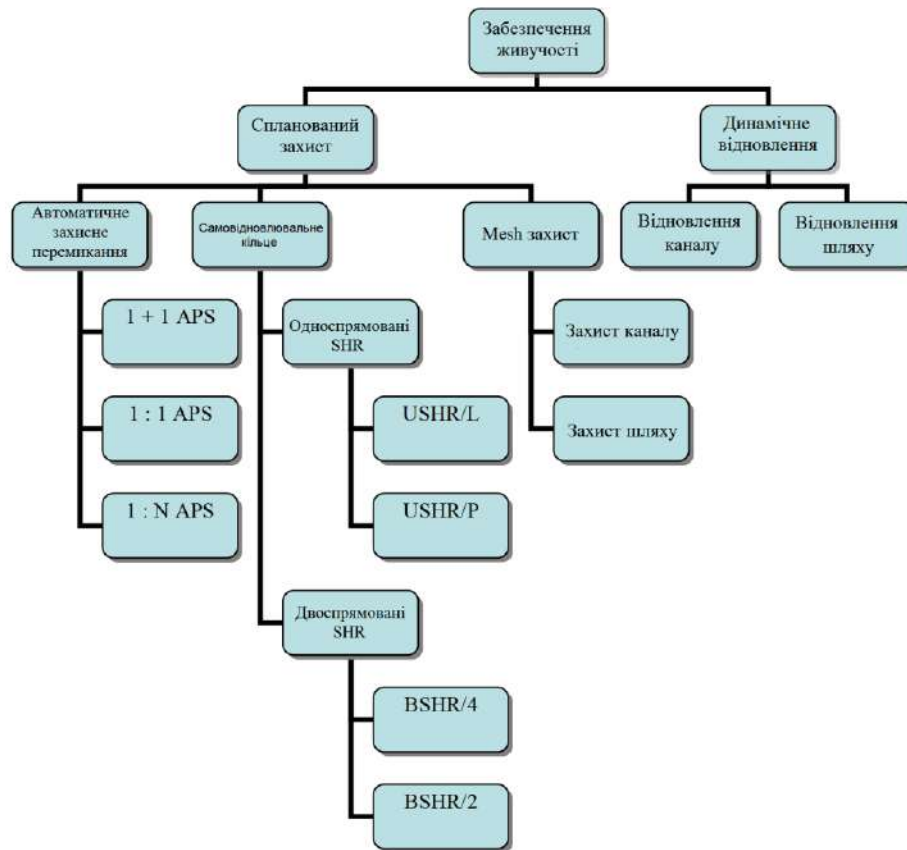


Рисунок 4.3 – Класифікація методів забезпечення живучості в оптичних мережах

#### 4.3 Захист спільного резервного шляху

Захист спільного резервного шляху (hard Backup Path Protection, SBPP) подібний до схеми 1+1 APS, оскільки два непересічні шляхи, робочий і резервний, направляються для передачі оптичного сигналу. Однак ця схема може розподіляти резервну пропускну здатність через резервні канали з кількох служб. З цієї причини SBPP може запропонувати ефективне використання пропускну здатності. У SBPP між вихідним і кінцевим вузлом основного робочого шляху знаходяться один або кілька резервних шляхів, але для захисту від збоїв мережі обирається лише один резервний шлях. Резервний шлях має бути каналом і вузлом, який не перетинається з основним шляхом. Це пов'язано з тим, що канал або вузол на робочому

шляху можуть постраждати, коли цей робочий шлях виходить з ладу. На рисунку 4.4 показано набір із трьох робочих шляхів, які можуть спільно використовувати резервні потужності в SBPP. Кожен робочий шлях має резервний шлях, який попередньо обчислюється, і кілька резервних шляхів можуть спільно використовувати резервну пропускну здатність на деяких ділянках. Наприклад, кожен робочий шлях спільно використовує канал E-F для резервування на рисунку 4.4, це можливо, оскільки вони не перетинаються з відповідними основними шляхами. Крім того, канал E-G належить до двох резервних шляхів. Резервна пропускну здатність попередньо розподіляється для захисту від збою мережі.

У разі виходу з ладу одного каналу вузли на робочому шляху повинні переналаштуватися для переходу на резервний шлях, що впливає на час відновлення працездатності. Як правило, час відновлення становить близько 200 мс [12]. SBPP є одним із методів живучості, який забезпечує високу ефективність пропускну здатності, але з повільним часом обчислення.

#### 4.4 Попередньо налаштований цикл захисту

Попередньо налаштований цикл захисту (р-цикл) класифікується як захист каналу в сітчастих мережах. Основна ідея р-циклів полягає у відновленні збоїв за допомогою механізму кільцевої мережі в сітчастій мережі. Цей механізм є перспективним підходом для вирішення проблеми проектування мережі в контексті живучості.

Основна концепція р-циклу була запропонована Гровером і Стамателакісом у 1998 році [10]. Він був розроблений як гібрид кільцевих і сітчастих механізмів захисту. Він має переваги щодо швидкого відновлення захисту кільця та ефективності пропускну здатності сітчастих топологій. Метод р-циклу заснований на замкнених циклічних маршрутах. Однак, на відміну від кільцевого механізму, який захищає лише робочі канали на кільці, метод р-циклу пропонує корисні резервні шляхи для захисту передачі,

а також каналів у циклі. Розмежована ланка р-циклу – це ланка, яка не належить цьому циклу, але кінцеві вузли якої лежать на р-циклі. Наприклад, на рисунку 4.5 (а) проміжки С-Е і С-Ф є розмежованими ланками р-циклу (А-С-В-Ф-Е).

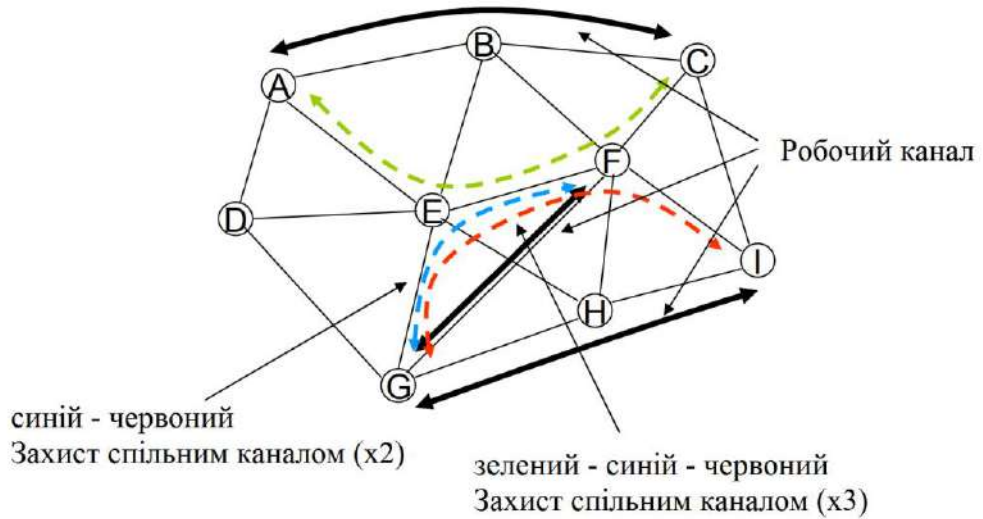


Рисунок 4.4 – Захист спільного резервного шляху

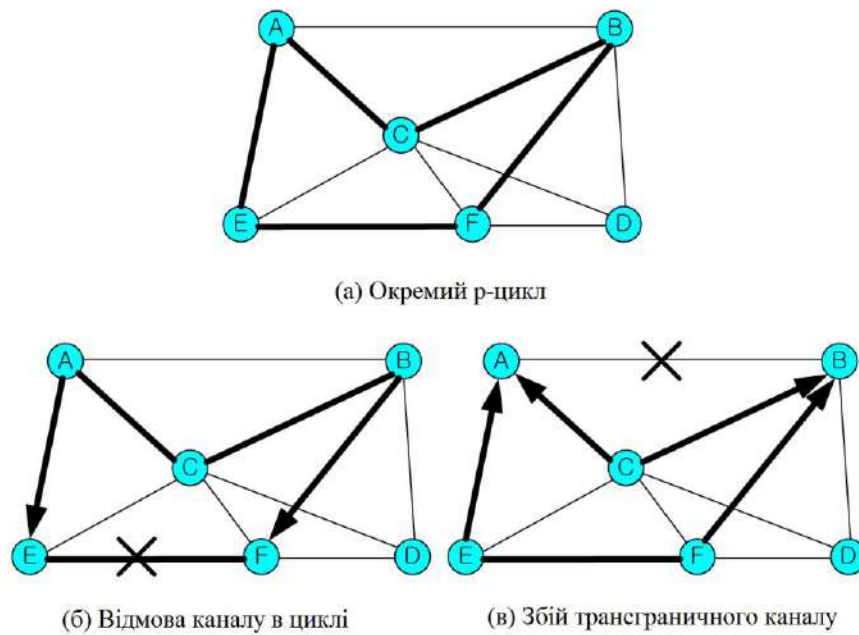


Рисунок 4.5 – Приклад внеску р-циклу на альтернативному шляху

Метод р-циклу може запропонувати два шляхи відновлення для трансплантованих зв'язків, не вимагаючи жодної резервної пропускної здатності. Як наслідок, це є надійний і ефективний метод використання пропускної здатності оптичних сітчастих мереж.

На рисунку 4.5 показано приклад внеску р-циклу. На рисунку 4.5 (а) зображено р-цикл (A-B-C-D-E-F). Альтернативний шлях для відмови каналу в циклі забезпечує єдиний шлях відновлення, як показано на рисунку 4.5 (б). Але коли транскордонне з'єднання (E-F) виходить з ладу, два вузли, що охоплюють збійне з'єднання, перемикаються на альтернативний шлях, і пропускна здатність резервується в обох напрямках уздовж циклу за допомогою р-циклу. Крім того, метод р-циклу забезпечує два альтернативні шляхи для цієї відмови, як показано на рисунку 4.5 (в).

Захист р-Cycle пропонує корисні шляхи відновлення залежно від каналу з несправним з'єднанням. Пошук відповідних шляхів є важливою проблемою захисту р-циклу, оскільки ефективні шляхи можуть запропонувати краще використання пропускної здатності. Позначення  $u_{i,j}$  представляє кількість корисних шляхів, де  $j$ -й р-цикл пропонує шляхи для працездатності, коли відбувається відмова. Нижче наведено кількість корисних р-циклів шляху:

- $u_{i,j} = 0$ , якщо один або обидва кінцеві вузли проміжку  $i$  не є вузлами циклу  $j$ ;
- $u_{i,j} = 1$ , якщо обидва кінцеві вузли проміжку  $i$  є вузлами циклу  $j$ , а проміжок  $i$  не входить до циклу;
- $u_{i,j} = 2$ , якщо обидва кінцеві вузли проміжку  $i$  є вузлами циклу  $j$ , а проміжок  $i$  не міститься в циклі.

Метод р-циклу дає змогу мінімізувати резервну пропускну здатність шляхом визначення відповідного набору р-циклів. Однак визначити ефективний і достатній набір р-циклів важко. Багато дослідників досліджували, як побудувати відповідні кандидати на р-цикли для вирішення цієї оптимізаційної задачі.

#### 4.5 Формалізація ІЛР

Формізація ІЛР може бути двох типів: без спільної оптимізації та спільної оптимізації. Для неспільної оптимізації модель ІЛР має два основні формулювання для проектування мереж р-циклу. Перше формулювання полягає в максимізації відновлюваності в межах заданої пропускної здатності та розміщення резервної пропускної здатності. Параметри в моделях такі:

- $E$  – набір мережевих прольотів;
- $P$  – набір усіх циклів-кандидатів у мережі;
- $z_j$  – кількість невідновлюваних робочих каналів на проміжку  $j$ ;
- $w_j$  – кількість робочих каналів довжини хвилі на проміжку  $j$ ;
- $s_j$  – кількість резервних каналів довжини хвилі на проміжку  $j$ ;
- $x_{j,k}$  – кількість корисних циклів шляху, які  $k$  може подолати після відмови проміжку  $j$ ;
- $R_j$  – кількість незахищених робочих каналів на проміжку  $j$ ;
- $n_j$  – кількість копій одиничної ємності циклу  $j$ ;
- $u_{j,k} = 1$ , якщо проміжок  $k$  знаходиться в циклі  $j$ ,  $0$  в іншому випадку;
- $a_j$  – кількість доступних шляхів захисту на проміжку  $j$ ;
- $q_j$  – вартість проміжку  $j$ .

Максимальна відновлюваність за допомогою р-циклів виражається як:

$$\min \sum_{j=1}^{|E|} z_j, \quad (4.3)$$

з урахуванням:

$$s_j \leq \sum_{k=1}^{|P|} x_{j,k} \cdot n_i, \quad \forall k = 1, 2, \dots, E. \quad (4.4)$$

$$R_j + \sum_{k=1}^{|P|} u_{j,k} \cdot n_k = w_j + a_j, \quad \forall j = 1, 2, \dots, E. \quad (4.5)$$

$$0 \leq R_j \leq w_j, \quad \forall j = 1, 2, \dots, E. \quad (4.6)$$

$$n_k \geq 0, \quad \forall k = 1, 2, \dots, E. \quad (4.7)$$

$$R_j \geq 0, \quad \forall j = 1, 2, \dots, E. \quad (4.8)$$

Друге формулювання визначає набір оптимальних кандидатів на р-цикли з мінімальною резервною ємністю. Метою моделі є мінімізація резервної пропускної здатності для 100% відновлюваності р-циклу. Це формулювання може запропонувати оптимальне рішення. Цільова функція тут:

$$\min \sum_{j=1}^{|E|} q_j s_j. \quad (4.9)$$

$$s_j = \sum_{k=1}^{|P|} x_{k,j} n_k, \quad \forall k = 1, 2, \dots, E. \quad (4.10)$$

$$w_j \leq \sum_{k=1}^{|P|} x_{j,k} n_k, \quad \forall j = 1, 2, \dots, E. \quad (4.11)$$

$$n_k \geq 0, \quad \forall k = 1, 2, \dots, P. \quad (4.12)$$

$$s_k \geq 0, \quad \forall k = 1, 2, \dots, E. \quad (4.13)$$

Однак основна проблема з ІЛР полягає в тому, що воно вимагає всіх можливих циклів, яких може бути дуже велика кількість у щільних і великомасштабних мережах. Обчислювальна складність у формулюванні ІЛР є складним питанням. Щоб подолати цю проблему, використовують нове формулювання ІЛР для неспільної оптимізації без перерахування кандидатів.

#### 4.6 Евристичні підходи

Евристичні підходи можна класифікувати на два типи: евристичні підходи, засновані на моделі ІЛР, і чисті евристичні підходи. Евристичний підхід, заснований на ІЛР, спочатку визначає набір обмежених кандидатів на р-цикли, а потім застосовується формулювання ІЛР для забезпечення 100% захисту. Цей підхід може досягти майже оптимальних рішень, але це все ще проблема оптимізації з пов'язаною обчислювальною складністю ІЛР. Чистий евристичний підхід розроблено для досягнення найкращої продуктивності без використання моделі ІЛР. Цей підхід також може досягти майже оптимальних рішень у межах прийнятної обчислювальної складності.

Щоб зменшити кількість р-циклів-кандидатів шляхом обчислення циклів високої ефективності, у [13] було досліджено евристичний підхід «попереднього вибору», а саме апріорну ефективність (АЕ). У цьому контексті цикл високої ефективності визначається як цикл із високою потенційною можливістю захисту від збою мережі. АЕ виражається як:

$$AE(p) = \frac{\sum_{\forall p \in S} X_{p,i}}{\sum_{(\forall p \in S | X_{p,i}=1)} c_i}, \quad (4.14)$$

де  $S$  – набір проміжків у мережі,  $i$  – проміжок у  $S$ .  $X_{p,i}$  – потенціал захисту, пов'язаний із проміжком  $p$ ,  $X_{p,i} = 1$ , коли проміжок  $i$  – це проміжок циклу, і  $X_{p,i} = 2$ , якщо проміжок  $i$  – розрізнений проміжок,  $c_i$  – вартість

пропускної здатності на проміжку  $i$ .  $AE$ , яка визначається як відношення загальної працездатності до загальної вартості прольоту для циклу, є мірою ефективності  $p$ -циклу. Теоретично  $p$ -цикл з високим значенням  $AE$  має високу ефективність для захисту працездатності мережі. Однак підхід «попереднього вибору» може не враховувати мережеві обставини, такі як навантаження трафіку на мережу та ступінь вузла. У результаті ця схема іноді залишає багато втраченої пропускної здатності на деяких прольотах, що не є хорошим рішенням.

В [14] Zhang запропонував алгоритм перемещування зв'язків (Straddling Link Algorithm, SLA) для захисту каналів. Ключовим моментом є генерація початкових  $p$ -циклів-кандидатів мережевого графа. Цей підхід спочатку знаходить найкоротший шлях між двома кінцевими вузлами проміжку, а потім шукає інший найкоротший шлях між тими самими парами вузлів, які є вузлами, не перетинаючими попередній шлях.

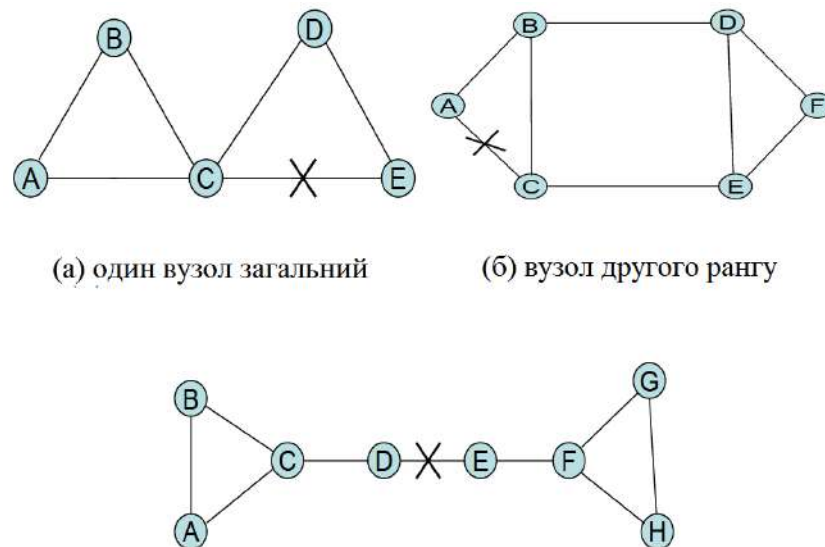


Рисунок 4.6 – Міркування щодо пошуку  $p$ -циклів за SLA

Створюється цикл, поєднуючи два найкоротші шляхи, не пересічні між вузлами (кожен шлях не містить спільних проміжних вузлів). SLA може побудувати лише набір  $p$ -циклів з одним розрізним проміжком. Щоб

створити набір кандидатів на р-цикли в SLA, він відрізає вузли вузлового рангу 1, оскільки жоден цикл не може містити такі вузли. Крім того, він не може побудувати р-цикл у наступних умовах мережі:

- випадок 1 – якщо зв'язок (C-E) виходить з ладу, не існує двох непересічних шляхів, як показано на рисунку 4.6 (а), оскільки існує лише один спільний вузол для кожного шляху;

- випадок 2 – якщо зв'язок (A-C) виходить з ладу, між одним і тим же вузлом є один або більше шляхів, а Але він не в змозі генерувати р-цикли, оскільки існує не більше одного шляху, що не перетинається, як показано на рисунку 4.6 (б);

- випадок 3 – якщо зв'язок (D-E) виходить з ладу, він не може обчислити р-цикл, оскільки між двома вузлами взагалі немає шляху, як показано на рисунку 4.6 (в).

SLA є простим і швидким алгоритмом для створення кандидатів р-циклів, однак кількість р-циклів недостатня, оскільки генерується лише  $O(m)$  р-циклів.

Набір кандидатів на р-цикли, згенерований SLA, є неефективним, оскільки цикли з вищою ефективністю не містяться в наборі, і, таким чином, цей підхід не може задовольнити мінімальну резервну ємність.

#### Лістинг 4.1 – Псевдокод алгоритму SLA

Input: топологія мережі

Output: набір кандидатів на р-цикли

Перевірка стану мережі

$S \leftarrow$  кількість переходів у мережі

for  $i = 1$  to  $|S|$  do

Пошук двох найкоротших непересічних шляхів між двома кінцевими вузлами переходу  $i$

Визначення р-циклу за двома знайденими шляхами.

end for

Інша схема, яка називається зважений пошук циклу на основі DFS (Weighted DFS-based cycle Search, WDCS), була запропонована Лю та Руаном [15]. Метою цього алгоритму є створення хороших циклів-кандидатів. WDCS містить два типи циклів, один – високоефективний набір циклів, а інший – два набори коротких циклів. Високоефективні цикли генеруються за допомогою пошуку спочатку в глибину (Depth First Search, DFS). DFS у цьому алгоритмі вибирає сусідній вузол ( $n$ ) із найвищою вагою серед усіх вихідних ребер із  $n$ , замість вибору будь-якого доступного сусіднього вузла для розширення шляху. Набір ваг спрямованих ребер базується на трьох міркуваннях:

- великі цикли мають вищу ефективність, оскільки ці цикли включають більше проміжків і щоб уникнути повернення до кореневої вершини, цей підхід призначає невелику вагу спрямованим ребрам, які закінчуються в корені (він встановлює  $\text{weight}(s,d) = \gamma$  для всіх  $d \in N(s)$ , де  $0 < \gamma < 1$ , а  $s$  — корінь.  $N(s)$  — набір сусідніх вузлів для  $s$ );

- вузол з високим ступенем спочатку вибирається DFS, і цикл, який включає цей вузол, має високу ймовірність включення більшої кількості розмежованих ребер.

Однак інколи бувають вузли 2 вузлового ступеня. Наприклад, цикл ( $r - v_1 - v_2 - r$ ) з АЕ рівним 1 знайдено WDCS, як показано на рисунку 4.6 (а). По суті, WDCS знаходить високоефективні цикли, тому він пропонує спеціальне рішення для вузла 2 вузлового ступеня. Цикл будується шляхом продовження шляху.

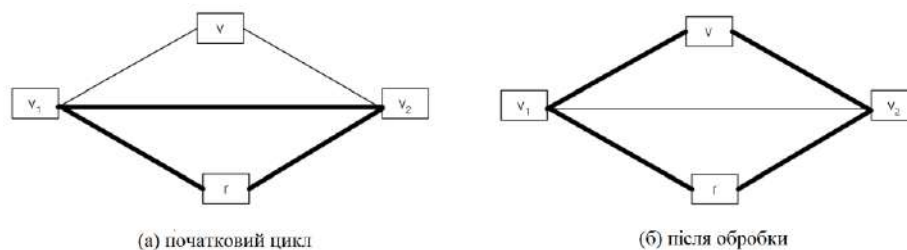


Рисунок 4.7 – Спеціальна обробка для вузлового ступеня 2

Як показано на рисунку 4.6 (б), якщо  $v$  включено в шлях пошуку від  $v_1$ , а вузол  $v$  здатний лише продовжити шлях до  $v_2$ , вузли з 2 вузловими ступенями можуть бути включені в шлях пошуку. Таким чином, вони можуть давати цикли з вищою ефективністю.

Крім того, кожна ланка має два коротких цикли. Для одного з цих циклів зв'язок буде ланкою в циклі, а для іншого циклу – транскордонною ланкою.

Щоб створити ці цикли, він знаходить два найкоротші непересічні шляхи між кінцевими вузлами кожного проміжку. Якщо він знайде два шляхи, він може побудувати два коротких цикли. Якщо знайдено лише один шлях, він створює лише зв'язок у циклі. Кількість циклів у WDCS визначається вхідним параметром  $k$ . Набір кандидатів  $p$ -циклів оцінюється на основі ІЛР з рівномірною та випадковою пропускну здатністю трафіку. Результати показують, що рішення, близького до оптимального, можна досягти, якщо вхідний параметр  $k$ , який є кількістю найкоротших шляхів, збільшити до 43.

Крім того, у цій схемі можуть не враховуватися деякі важливі обставини мережі, наприклад розподіл заданої пропускну здатності.

Були запропоновані інші евристичні підходи, які не залежать від моделі ІЛР [16, 17, 18]. Різні якості продуктивності досягаються залежно від гнучкості вибору  $p$ -циклів кандидатів після перерахування набору  $p$ -циклів. Популярним використовуваним евристичним алгоритмом є Capacitated Iterative Design Algorithm (CIDA), запропонований Doucette [16]. CIDA оцінює ефективність циклу за допомогою фактичної ефективності  $E_w(p)$ , яка визначається як відношення загальної захищеної потужності до загальної вартості діапазону для кожного циклу.  $E_w(p)$  виражається як:

$$E_w(p) = \frac{\sum_{\forall i \in S} w_i X_{p,i}}{\sum_{(\forall i \in S | X_{p,i}=1)} \text{cost}_i}, \quad (4.15)$$

де  $w_i$  – кількість незахищеної пропускну́ї здатності. CIDA створює первинний набір  $p$ -циклів-кандидатів, використовуючи підхід SLA. Потім більш складні  $p$ -цикли генеруються трьома алгоритмами, а саме «SP-add», «Expand» і «Grow».

Алгоритми Grow і Expand засновані на алгоритмі SP-add. «SP-add» створює  $p$ -цикли, використовуючи найкоротший шлях між парними вузлами вибраного інтервалу на  $p$ -циклі. Наприклад, якщо вибрано перехід 7-8, як показано на рисунку 4.8 (а), «SP-add» шукає непересічні найкоротші шляхи між парою вузлів інтервалу.

Найкоротший шлях замінюється на вихідний цикл, щоб створити більш складний  $p$ -цикл, як показано на рисунку 4.8 (б). Потім інші цикли первинного набору обробляються подібним чином. Але алгоритм «Expand» продовжує пошук найкоротших шляхів інших інтервалів у тому самому циклі.

Тобто він продовжує пошук, доки не буде відвідано кожен проміжок у початковому циклі. Найкоротші шляхи кожного проміжку повинні бути непересічним шляхом від кожного попереднього маршруту. Алгоритм «Grow» подібний до «Expand», за винятком того, що операція переходить до першого інтервалу, щоб знайти новий найкоротший шлях після створення  $p$ -циклу.

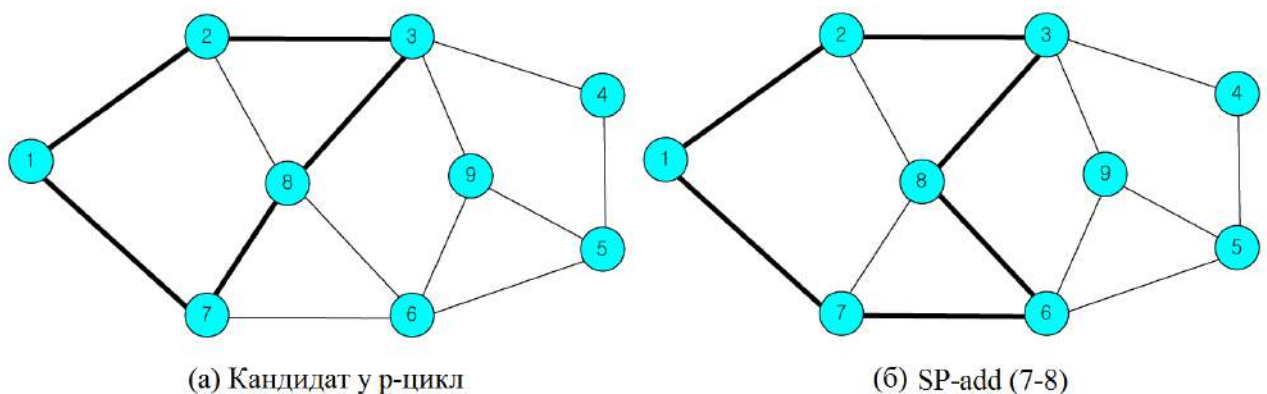


Рисунок 4.8 – Робота SP-add

Після створення набору  $p$ -циклів CIDA вибирає серед набору цикл із високим  $E_w(p)$ , а потім видаляє пропускну здатність, захищену цим вибраним циклом. Залишок пропускну здатності оновлюється, і процес повторюється, доки не буде захищено всю пропускну здатність. На відміну від АЕ, ця формула розраховується відповідно до розподілу пропускну здатності кожного прольоту на цикл. Отже, цей підхід може досягти кращої продуктивності, ніж АЕ. Нарешті, у статті повідомляється, що алгоритм «Grow» на основі CIDA є найефективнішим. Псевдокод CIDA показано в лістингу 4.2.

#### Лістинг 4.2 – Псевдокод алгоритму CIDA

```

Input: набір кандидатів на  $p$ -цикли, незахищена пропускну
здатність
Output: набір  $p$ -циклів
W ← незахищена пропускну здатність
while Sum of W is not zero do
    обчислити  $E_w(p)$  кандидатів у  $p$ -цикли за допомогою
    вибрати максимальний  $E_w(p)$  із набору
    видалити пропускну здатність виділеного  $p$ -циклу з W
end while

```

Алгоритм unity- $p$ -cycle на основі ER був запропонований Zhang [17]. Цей алгоритм враховує односпрямовані  $p$ -цикли та трафік у мережі. Спочатку він створює усі можливі цикли в мережі, а потім обчислює коефіцієнт ефективності (ER), який є відношенням кількості фактично захищених працюючих одиниць до кількості запасних одиниць циклу unity- $p$ . Одиночний  $p$ -цикл із високим ER є більш ефективним циклом. Таким чином, він обирає максимальні Unity- $p$ -цикли на основі ER, щоб зменшити незахищену ємність, доки не буде захищена вся робоча ємність. Виявлено, що він здатен досягти майже оптимального рішення без використання моделі ІЛР.

Чжан та інші [18] запропонували евристичний алгоритм для мінімізації загальної резервної пропускної здатності. Запропонований алгоритм вибирає цикли відповідно до надмірності р-циклу, яка є відношенням резервної ємності циклу до робочої ємності, захищеної циклом. Надмірність ( $R$ ) виражається як:

$$R = \frac{\sum_{i \in S} p_i}{\sum_{i \in S} w_i}, \quad (4.16)$$

де  $w_i$  – кількість одиниць робочої ємності на прольоті  $i$ , а  $p_i$  – відповідна кількість одиниць резервної ємності на прольоті  $i$ . Цикл з невеликою надлишковістю більш ефективний для захисту пропускної здатності. Однак, коли пропускна здатність, захищена циклом, дорівнює нулю, алгоритм має проблеми, оскільки в цій ситуації надлишковість нескінченна. Нижче наведено евристичний алгоритм:

Крок 1. Сформувані всі цикли в мережі.

Крок 2. Для кожного кандидата-циклу обчислити  $R$  за допомогою формули (4.16).

Крок 3. Вибір максимального  $R$  із набору.

Крок 4. Видалити пропускну здатність, захищену вибраним циклом.

Крок 5. Поверніться до кроку 3, доки пропускна здатність на кожному інтервалі не стане нульовою.

Непрості р-цикли були запропоновані Груберном [19]. Вони потрібні для досягнення високої продуктивності, коли недостатньо доступної пропускної здатності для повного захисту. На відміну від звичайних р-циклів, непрості р-цикли дозволяють двічі відвідувати вузол. На рисунку 4.9 (б) показано приклад захисту непростого р-циклу.

Коли зв'язок А-В виходить з ладу, звичайний р-цикл не є найкращим вибором для захисту. Це пояснюється тим, що доступний лише звичайний р-цикл (А-Н-В-С-Д-Е-Ф-Г-А), який повинен проходити кожен вузол. Однак

непростий р-цикл здатний знайти малий цикл (A-H-B-C-H-G-A), щоб покрити збій так, щоб він зменшив резервну пропускну здатність. Результати моделювання показують, що непрості р-цикли зменшують надмірність порівняно зі звичайними р-циклами. Однак кількість р-циклів-кандидатів, необхідних для захисту, збільшується, тому необхідний час обчислення довший.

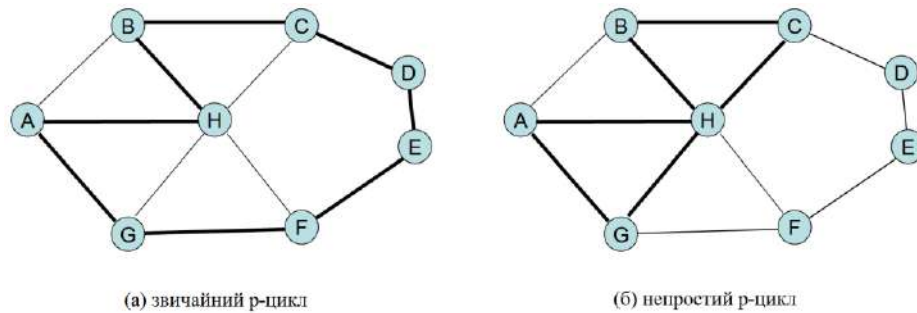


Рисунок 4.9 – Приклад непростих циклів

Таким чином, розглянуті різні підходи, які можна використовувати для захисту живучості мережі. Зокрема, р-цикл захист може запропонувати чудову продуктивність з точки зору оптимальності рішення та складності обчислень. При проектуванні мережі р-циклу оптимізаційна задача була сформульована як цілочисельна лінійна програма (ILP). Були запропоновані різні традиційні евристичні підходи. Однак ці евристичні підходи все ще мають проблему оптимізації з точки зору балансу між оптимальністю рішення та обчислювальною складністю. Підходи SLA, WDCCS і CIDA пропонують швидший час обчислення, але мають проблеми з використанням пропускну здатності в щільних і великих мережах. Підхід «попереднього відбору» має проблему обчислювальної складності, але демонструє хороше використання потужностей мережі.

## 5 АНАЛІТИЧНА ЧАСТИНА

### 5.1 Теорія графів

Загальне рішення для проектування мережі може бути реалізоване за допомогою алгоритмів теорії графів.

Ці алгоритми засновані на посиланні на реальні мережі. Загалом графи поділяються на два типи в залежності від напрямку пропускної здатності на прольотах: орієнтовані та неорієнтовані графи.

Граф представляється як  $G(V, E)$ , де  $V = \{v_1, v_2, \dots, v_N\}$  — набір із  $N$  вершин, а  $E = \{e_1, e_2, \dots, e_M\}$  — це набір ребер, які створюють зв'язки між вершинами  $v_i$  та  $v_j$ . Різниця між двома типами графів полягає лише в напрямку ребер. Наприклад, якщо існує шлях від  $v_i$  до  $v_j$ , напрямком шляху вказується стрілкою в орієнтованих графах, тоді як існує автоматично шлях від  $v_j$  до  $v_i$  в неорієнтованому графі.

Ребро позначається множиною  $(v_i, v_j)$ , де  $v_i, v_j \in V$  і  $v_i \neq v_j$ . Два ребра називаються суміжними або сусідніми, якщо вони сполучаються з однією вершиною. Подібним чином дві вершини називаються суміжними або сусідніми, якщо вони знаходяться на одному й тому самому ребрі. Крім того, ступінь вершини визначається кількістю різних ребер, які їй інцидентні. Ступінь зовнішнього та внутрішнього ступеня вершини — це кількість ребер, які виходять або входять із вершини відповідно. Відповідне мережеве середнє по всіх вершинах називається вузловим ступенем (d).

З метою опису визначимо наступні позначення:

- $C$  — набір каналів (пропускна здатність) на інтервалі;
- $\omega$  — набір незахищеної пропускної здатності в мережі;
- $A$  — набір кандидатів у  $p$ -цикли в мережі;
- $\Phi$  — потенційна захисна пропускна здатність;
- $\Phi_i$  — потенційна захисна пропускна здатність циклу  $i$ ;

- $\Delta_i$  – розподіл незахищеної пропускної здатності після вилучення пропускної здатності, захищеної циклом  $i$ ;
- $\Lambda$  – середній внесок незахищеної пропускної здатності;
- $\Lambda_i$  – Середній внесок незахищеної пропускної здатності циклу  $i$ ;
- $\Xi$  – стандартне відхилення середньої незахищеної пропускної здатності;
- $\Psi$  – витрачений резервний потенціал;
- $\Psi_i$  – витрачений резервний потенціал циклу  $i$ ;
- $M$  – максимальна місткість  $\omega$ ;
- $L$  – перехід із максимальною незахищеною пропускною здатністю;
- $\text{cycle}^s$  –  $s$ -й кандидат у  $p$ -цикл у наборі  $(A)$ ;
- $S_{p,i}$  – розподіл циклу  $p$  на проміжку  $i$ ;
- $W_i$  – кількість робочих каналів на прольоті  $i$ ;
- $C_i$  – кількість резервних каналів на проміжку  $i$ .

В алгоритмах теорії графів було проведено багато досліджень для вирішення проблеми єдиного найкоротшого шляху для пошуку шляху з мінімальними витратами від джерела до пункту призначення через підключену мережу. Це питання важливе через широкий спектр його застосування на транспорті.

Крім того, найкоротші шляхи є можливим способом надання альтернативних рішень для проектування мережі.

Модель ILP зазвичай обчислює заданий попит на трафік для кожного переходу на основі алгоритму найкоротшого шляху для отримання оптимального рішення.

Існує кілька алгоритмів для визначення найкоротшого шляху між вузлом джерела та вузлом призначення: наприклад, алгоритм Дейкстри, алгоритм пошуку в ширину (BFS) і Беллмана-Форда. У деяких випадках також необхідно знати  $k$  найкоротших шляхів між парою вузлів. Це пояснюється тим, що  $k$  найкоротших шляхів потрібні для конкретних цілей з точки зору маршрутизації та покращення продуктивності роботи мереж.

Алгоритм Дейкстри знаходить найкоротші шляхи від джерела до пункту призначення за допомогою зваженої відстані. Усі ваги мають бути позитивними. Вихідний вузол має бути єдиним, але одержувачем можуть бути всі інші вузли. Якщо потрібен найкоротший шлях до вузла  $A$ , цей алгоритм завершується, коли найкоротший шлях до вузла  $A$  знайдено. Час, необхідний для алгоритму Дейкстри, дорівнює  $O(N^2)$ , де  $O$  – порядок складності алгоритму. Основна ідея цього алгоритму полягає в тому, щоб спочатку знайти найближчий вузол із початкового вузла. Перший найближчий вузол повинен бути сусідом початкового вузла. Наступний вузол повинен бути сусідом першого найближчого вузла. Алгоритм продовжує пошук, доки не буде знайдено вузол призначення. Псевдокод алгоритму Дейкстри представлений нижче.

#### Лістинг 5.1 – Псевдокод алгоритму Дейкстри

```

Input:  $G(V, E)$ 
Output: найкоротший шлях
 $S \leftarrow$  кореневий вузол
 $D \leftarrow$  вузол призначення
path  $\leftarrow S$ 
while останній вузол шляху  $\in D$  do
    сканувати всіх сусідів з останнього вузла шляху
    визначити з непомічених вузлів вузол з найменшою вагою
    розмістити вибраний вузол в останню позицію шляху
end while

```

Алгоритм пошуку пошуку в ширину (BFS) є одним із найпростіших алгоритмів для пошуку найкоротшого шляху на заданому графу. BFS досліджує всі суміжні вузли від вихідного вузла та зберігає вузли, коли вони знаходяться в черзі. Потім він шукає глибше в одному з сусідніх вузлів. BFS продовжує рух через наступні сусідні вузли, доки не буде знайдено найкоротший шлях. Загалом, BFS обчислює найкоротший шлях за

допомогою охоплюючого дерева, і різні форми охоплюючого дерева генеруються залежно від порядку, у якому сусідні вузли розміщені в черзі. На рисунку 5.1 представлено остовне дерево BFS у заданому графі.

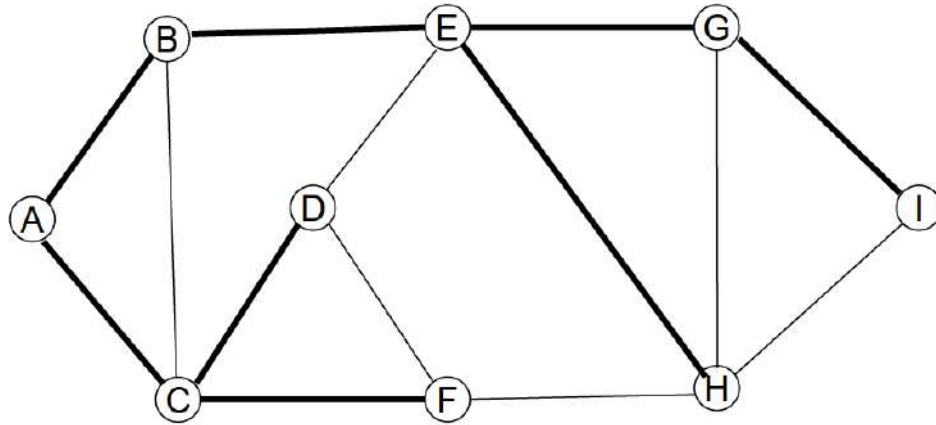


Рисунок 5.1 – Результат пошуку остовного дерева алгоритмом пошуку в ширину

BFS можна використовувати для досягнення чотирьох різних цілей:

- тестування підключення,
- пошук остовного дерева,
- пошук найкоротшого шляху,
- пошук усіх окремих циклів.

У кваліфікаційній роботі використовується алгоритм пошуку в ширину (BFS) для оцінки одного із запропонованих алгоритмів. Мета полягає в тому, щоб визначити всі різні цикли. Для цієї мети BFS представлений як алгоритм в псевдокоді в листингу 5.2.

Алгоритм Беллмана Форда подібний до алгоритму Дейкстри, який призначений для вирішення проблеми найкоротшого шляху, але відмінність полягає в тому, що цей алгоритм може мати як позитивні, так і негативні ваги.

Цей алгоритм шукає найкоротший шлях через процес розширення шляху з вартістю або вагою, як алгоритм Дейкстри.

## Лістинг 5.2 – Псевдокод алгоритму BFS

```

Input:  $G(V, E)$ ,  $s$  кореневий вузол
Output: усі окремі цикли ( $P$ )
 $e \leftarrow$  набір сусідніх вузлів з  $s$ 
queue  $\leftarrow [s, node.e]$ 
while queue  $\neq \emptyset$  do
    path  $\leftarrow$  head(queue)
    queue  $\leftarrow$  queue - path
     $d \leftarrow$  останній вузол з path
    if  $d == s$  then
         $P \leftarrow$  path
    else
         $nei \leftarrow$  set of neighbors of
        queue  $\leftarrow [queue, node.nei]$ 
    end if
end while

```

Обчислювальна складність Беллмана становить  $O(N)$ , коли ступінь вузлів або мережа невеликі, але в найгіршому випадку це  $O(N^3)$ . Якщо всі ребра мають однакове значення одиниці, алгоритм BFS є більш ефективною альтернативою.

Однак, якщо виконується пошук найкоротших шляхів для позитивних ваг, алгоритм Дейкстри є більш ефективною альтернативою для пошуку найкоротших шляхів, ніж алгоритм Беллмана Форда.

Алгоритм  $k$ -найкоротшого шляху створює  $k$  найкоротших шляхів від заданого джерела до вузла призначення. Цей алгоритм знаходить наступний найкоротший шлях таким чином, що він не використовує жодне ребро або проміжок попереднього шляху після того, як знайдено перший найкоротший шлях. Кількість найкоротших шляхів контролюється вхідним параметром  $k$ . Дія цього алгоритму припиняється, коли вартість другого найкоротшого шляху перевищує вартість попередніх шляхів. Алгоритм представлено в лістингу 5.3.

### Лістинг 5.3 – Псевдокод алгоритму $k$ -найкоротшого шляху

Input:  $G(V, E)$ , кількість найкоротших шляхів ( $k$ )

Output:  $k$  найкоротших шляхів

1.  $s$  and  $d$  є вихідним і кінцевим вузлами відповідно
2. Знайти найкоротше дерево від вихідного вузла  $s$  до інших вузлів у мережі
3. Визначити найкоротший шлях  $P_1 = (s, s_1, s_2, \dots, s_n, d)$  як перший найкоротший шлях
4. Видалити зв'язки  $(s_1, s_2, \dots, s_n)$  на найкоротшому шляху  $P_1$  із заданої мережі
5. Знайти другий найкоротший шлях від вихідного вузла до вузла призначення, якщо існує інший найкоротший шлях
6. Продовжувати шукати, доки номер найкоротшого шляху не дорівнюватиме  $k$  або доки не буде знайдено подальшого найкоротшого шляху

Алгоритм  $k$ -найкоротшого шляху можна класифікувати на три окремі класи:

- $k$ -найкоротший зв'язок непересічних шляхів: набір найкоротших шляхів, у якому кожен шлях може мати спільні проміжки, але зв'язок не пересічний;

- $k$ -найкоротші окремі маршрути: множина маршрутів по прольотах, у яких кожен із маршрутів не перетинається з попередніми маршрутами;

- $k$ -найкоротший пролітно-непересічний маршрут: множина маршрутів над прольотами, в яких кожен із маршрутів не перетинається з прольотами на попередніх маршрутах.

Однак ці алгоритми пошуку найкоротшого шляху іноді вимагають великої резервної ємності для захисту всього наявного трафіку в мережі. Така поведінка пояснюється тим, що може знадобитися використовувати дуже багато найкоротших шляхів, щоб забезпечити 100% захист у разі збою мережі.

## 5.2 Тестові мережі

Запропонований у подальших розділах алгоритм був реалізований і оцінений на чотирьох різних тестових мережах. Ці тестові мережі, а саме мережа Національного наукового фонду США (NSFNet), COST239, Європейська оптична мережа (EON) і мережі США, використовуються, як показано на рисунку 5.2. Багато досліджень у цій галузі використовують ці мережеві топології для реалізації та тестування алгоритмів. Це пояснюється тим, що ці мережеві конфігурації засновані на посиленні на реальні національні магістральні мережі. Тому у кваліфікаційній роботі також використовуються ці топології мережі для перевірки теоритичних викладок дослідження.

Першою досліджуваною мережею є (NSFNet), яка відтворена на рисунок 5.2 (а), з 14 вузлами та 21 переходом. Мережа має вузловий ступінь  $d$  3. Початковий NSFNet складався з магістральної мережі США.

Мережа COST239 (рисунок 5.2 (б)) містить 11 вузлів і 26 проміжків і має високий вузловий ступінь  $d \sim 4,7$ . Крім того, вузловий ступінь кожного вузла дорівнює або перевищує 4, щоб досягти рівня резервування лише на 30%. Ця мережа є мережею середнього розміру, як уже згадувалося, але також є дуже щільною мережею.

Топологія Європейської оптичної мережі (EON) має 19 вузлів і 38 смуг, як показано на рисунку 5.2 (в). Вузловий ступінь дорівнює 4.

Топологія США (рисунок 5.2 (г)) містить 28 вузлів і 45 проміжків і має вузловий ступінь  $d$  3.21. Ця мережа заснована на посиленні на магістральну мережу IP США. Нарешті, в кожній мережі існує 139, 3531, 8857 і 7321 циклів відповідно. У таблиці 5.1 узагальнено інформацію про ці тестові мережі.

Складність необ'єднаних моделей ILP у щільних мережах набагато вища, ніж складність у малих мережах, як обговорювалося раніше. Це пояснюється тим, що в щільних мережах кількість змінних велика.

Складність моделі ІЛР становить  $2^{139}$ ,  $2^{3531}$ ,  $2^{8857}$ ,  $2^{7321}$  у кожній мережі відповідно [20]. Коротше кажучи, щільні мережі, які мають високу складність, більш складні для отримання оптимального рішення.

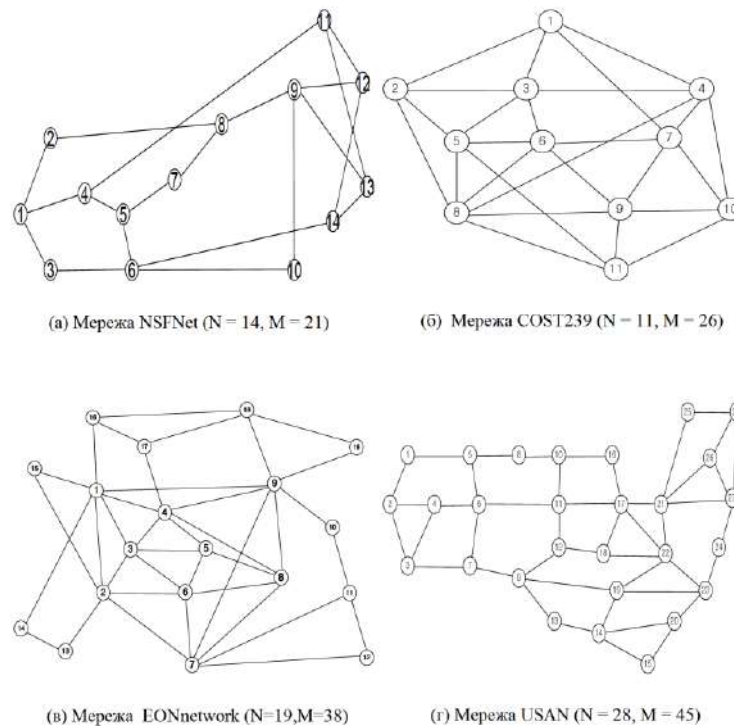


Рисунок 5.2 – Чотири тестові топології мереж

Таблиця 5.1 – Інформація про чотири тестові мережі

Мережа	NSFNet	COST239	EON	USA
Кількість вузлів	14	11	19	28
Кількість прольотів	21	26	38	45
Середній вузловий ступінь	3	4.72	4	3.21
Всі можливі цикли	139	3531	8857	7321
Середнє АЕ	1.42	2.81	2.57	1.70
Циклів за SLA	17	26	27	31

Загалом підключення трафіку пропонується відповідно до двох різних типів потреб трафіку в мережах: статичний трафік і динамічний трафік. При статичному трафіку мережевий трафік призначається заздалегідь, і мережа

повинна задовольняти всі запити трафіку, а також мінімізувати загальну вартість мережі. Модель ILP використовується для вирішення задачі оптимізації.

З динамічним трафіком кожне підключення в мережі запитує трафік випадковим чином, і це розглядається як можливість підвищити ймовірність блокування шляхом динамічної реакції на динамічно змінний трафік. Необхідно враховувати динамічний трафік, оскільки це важливо насправді. Однак у кваліфікаційній роботі розглянуто лише статичну ситуацію та залишено динамічну ситуацію для майбутньої роботи, оскільки статична ситуація в основному є першою областю дослідження, яка переходить у наступну область дослідження з точки зору динамічної ситуації.

Кожному вузлу дозволено містити однакову кількість ресурсів мережі передачі. Крім того, усі смуги WDM мають однакову вагу, яка дорівнює одиниці, і кожна смуга волокна може запропонувати до  $C=32$  каналів довжини хвилі.

Маршрутизація та призначення довжини хвилі (RWA) шляхів світла є важливим питанням, як згадувалося раніше. Канали, що утворюють шлях світла, класифікуються на два випадки залежно від доступності перетворення довжини хвилі: віртуальний шлях довжини хвилі (VWP) і шлях довжини хвилі (WP). VWP може перетворювати вхідну довжину хвилі на будь-яку вихідну довжину хвилі, оскільки перетворювачі довжини хвилі доступні на кожному вузлі, тоді як WP не може перетворювати будь-яку довжину хвилі.

Як обговорювалося, не розглядаємо проблему RWA, тому припускаємо, що повне перетворення довжини хвилі доступне в усіх мережах WDM. Ми визначили використання діапазону (%), яке є відношенням робочої ємності в кожному прольоті до максимальної місткості, забезпеченої в цьому діапазоні, генерується випадковим чином між 25%~40%. Через низьку ймовірність множинних збоїв у мережах усі наші моделювання базуються на збоях одного каналу мережі. Нарешті, програмна платформа – Matlab. Тривалість усього моделювання обмежена 6 годинами.

## 6 ГЕНЕРАЦІЯ ЕФЕКТИВНОЇ МНОЖИНИ КАНДИДАТІВ У Р-ЦИКЛИ

Пропонується нова структура для обчислення ефективного набору р-циклів, що може призвести до підвищення продуктивності з точки зору використання пропускної здатності для захисту мережі від збоїв.

Пропонується алгоритм для знаходження всіх основних циклів. Потім набір кандидатів на р-цикли будується шляхом злиття основних циклів. Кількість запропонованих р-циклів-кандидатів є невеликою, але вони можуть побудувати більше розмежованих зв'язків і мають високу апіорну ефективність. Нарешті, модель ІЛР використовується для вибору адекватного набору р-циклів для забезпечення 100% захисту при мінімізації загальної резервної пропускної здатності.

### 6.1 Попередня теорія

Множина основних циклів визначається як множина простих циклів, отриманих із остовного дерева графа. Ці цикли можуть містити міжциклові зв'язки. Крім того, набір основних циклів не може сконструювати всі можливі цикли, які можуть бути рішеннями в побудові р-циклу. Як зазначалося вище, рішення ІЛР вимагає перерахування всіх можливих циклів. Розмежована ланка циклу – це ланка, яка не належить цьому циклу, але кінцеві вузли якої лежать на циклі. Ланка циклу циклу є ланкою, що належить цьому циклу. Основний цикл – це цикл, який не містить жодних зв'язків з іншими циклами.

Приклад показано на рисунку 6.1, де набір основних циклів  $FC = \{c_1, \dots, c_5\}$ , де  $c_1 = (A-B-F-A)$ ,  $c_2 = (A-F-E-A)$ ,  $c_3 = (C-F-D-C)$ ,  $c_4 = (A-B-C-F-A)$  і  $c_5 = (A-F-D-E-A)$ . Легко побачити, що цикли  $c_4$  і  $c_5$  містять ланки B-F і E-F як розмежовані ланки відповідно. Крім того, FC не може побудувати цикл  $c_6 = (A-B-C-D-E-A)$  і два малих цикли ( $c_7 = (B-C-F-B)$ ,  $c_8 = (D-E-F-D)$ ), тому цей

набір може бути не в змозі побудувати всі можливі цикли в мережі. Однак набір основних циклів містить  $c_6$ ,  $c_7$ ,  $c_8$ , оскільки вони не включають жодних граничних зв'язків. Насправді, множина містить усі можливі основні цикли, які не містять міжміжних зв'язків. Таким чином, набір основних циклів може побудувати всі можливі цикли мережі, об'єднавши ці основні цикли.

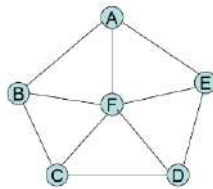


Рисунок 6.1 – Топологія мережі, 6 вузлів, 10 ребер

## 6.2 Генерація потенційних $r$ -циклів

Запропонований метод генерації кандидатів на  $r$ -цикли є двоетапним процесом. Перший етап породжує набір основних циклів. На другому етапі визначається набір ефективних кандидатів на  $r$ -цикли з них. Після того, як сформовано набір кандидатів на  $r$ -цикли, застосовується формула ILP для мінімізації резервної пропускної здатності для вибору оптимального набору  $r$ -циклів, який забезпечить 100% захист працездатності. Для створення основних циклів пропонується новий метод під назвою «Пошук в ширину-глибину» (АПШГ).

### 6.2.1 Алгоритм пошуку в ширину-глибину (АПШГ)

Загалом існує два можливих підходи для створення циклів кандидатів: пошук спочатку в ширину (BFS) і пошук спочатку в глибину (DFS). По суті, ці підходи генерують усі цикли на основі остовних дерев. BFS перевіряє кожен можливий шлях від кожного вузла, щоб визначити, чи він утворює цикл, досліджуючи спочатку найкоротші цикли. Навпаки, DFS досліджує

кожен можливий шлях від першого вузла. Якщо шлях не містить кінцевого вузла, інший шлях досліджується шляхом повернення назад, доки не буде знайдено цикл. На рисунку 6.2 представлено остовне дерево цих підходів.

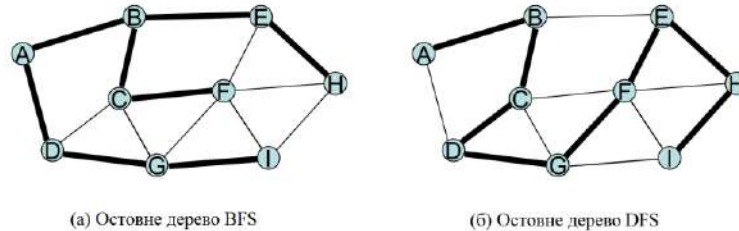


Рисунок 6.2 – Остовне дерево BFS і DFS

У запропонованому методі спочатку досліджуються всі можливі основні цикли, що проходять через початковий вузол A. Потім видаляється вузол A із заданої мережі, щоб уникнути перерахування повторюваних циклів. Потім кожен вузол досліджується поступово, доки в мережі не залишиться інших вузлів.

Псевдокод для генерації основних циклів за допомогою методу показано в лістингу 6.1.

#### Лістинг 6.1 – Генерація основних циклів

```

Input: неорієнтований граф  $G(V, E)$ 
Output: набір усіх основних циклів
 $i \leftarrow 1$ 
 $NoN = |V|$ 
while  $NoN \geq 3$  do
    Знайти всі основні цикли, що проходять через вузол  $i$ ,
    використовуючи АПШГ
    Видалити вузол  $i$  з  $G(V, E)$ 
     $i = i + 1$ 
     $NoN = NoN - 1$ 
end while

```

В алгоритмі генерації основних циклів усі основні цикли, що проходять по кожному вузлу, досліджуються АПШГ. Наступні дві умови використовуються для формування основних циклів.

По-перше, для формування циклу АПШГ потрібні принаймні два вузли, розташовані поруч із кореневим вузлом. Це тому, що цикл не може бути сформований, якщо є лише один сусідній вузол.

По-друге, алгоритм перевіряє всі розширені вузли, щоб знайти вузол, підключений до попереднього вузла. Це пов'язано з тим, що цей вузол має знаходитися на розмежувальній ланці циклу. Алгоритм АПШГ видаляє ці вузли з розширених вузлів. Псевдокод алгоритму АПШГ показаний як лістинг 6.2.

### 6.2.2 Створення набору кандидатів на р -цикли

У запропонованому методі ефективний набір р-циклів обчислюється шляхом злиття основних циклів. Існуючі підходи для об'єднання циклів можна класифікувати на дві групи: алгоритм «Додавання» та алгоритм «Об'єднання» [21]. Обидва ці алгоритми створюють р-цикли з більшою кількістю розмежованих зв'язків, щоб р-цикли були ефективнішими. Алгоритм «Додавання» можливий лише тоді, коли розмежована ланка в циклі А є ланкою циклу в циклі В, а ланка в циклі В є ланкою циклу в циклі А. Однак у запропонованому методі основні цикли не мають між собою зв'язків, як згадувалося.

Таким чином, алгоритм «Додавання» не підходить для генерації р-циклів у запропонованому методі. З іншого боку, алгоритм «Об'єднання» використовується, коли пари фундаментальних циклів мають одну спільну ланку, і загальна ланка стає трансмісійною ланкою для р-циклу. Однак два суміжні цикли з більш ніж однією спільною ланкою не можуть утворити новий р-цикл. Це пояснюється тим, що два цикли не можуть побудувати взаємодіючий зв'язок, коли спільні зв'язки цих циклів є суміжними.

## Лістинг 6.2 – Псевдокод алгоритму АПШГ

Input: неорієнтований граф  $G(V, E)$ , кореневий вузол

Output: основні цикли-кандидати

$Nei \leftarrow$  множина сусідів кореневого вузла

if  $|Nei| \leq 2$  then

    return

end if

for  $i = 1$  to  $|Nei| - 1$  do

    for  $j = i + 1$  to  $|Nei|$  do

$s \leftarrow Nei(i)$

$d \leftarrow Nei(j)$

        if  $d \in$  сусідом  $s$  then

            Створіти основний цикл

        else

            pathlist = [s]

            while Pathlist is not empty do

                currentpath  $\leftarrow$  pathlist(1)

$s \leftarrow$  останній вузол із currentpath

                Убрати  $s$  із pathlist

                if  $d \in$  сусіднім вузлом  $s$  then Створіти основний цикл

                else

                    for кожного  $k$  сусіднього вузла  $s$  do

                        if  $d \in$  сусіднім вузлом  $k$  then

                            Створіти основний цикл

                        else

                            Newpath = [currentpath, k]

                            Додайте newpath у список pathlist

                        end if

                    end for

                end if

            end while

        end if

    end for

end for

У запропонованому методі для генерації набору р-циклів використовується алгоритм «Додавання». Спочатку метод вибирає один фундаментальний цикл  $A$  з набору основних циклів. Потім створюється список усіх інших циклів, які мають один спільний зв'язок із циклом  $A$ . Найменший такий цикл обирається для приєднання до циклу  $A$  та формування нового р-циклу.

Якщо такого циклу не існує, виконується перехід до пошуку іншого основного циклу в нашому наборі циклів. Ця процедура повторюється до тих пір, поки не буде знайдено інших р-циклів. Псевдокод алгоритму генерації набору р-циклів наведено в лістингу 6.3.

### Лістинг 6.3 – Псевдокод алгоритму генерації набору р-циклів

```

Input: набір основних циклів
Output: кандидати на р-цикли
listpcycle  $\leftarrow \emptyset$ 
Fc  $\leftarrow$  сукупність основних циклів
for i = 1 to |Fc| do
    pcycle  $\leftarrow$  Fc(i)
    while pcycle is not empty do
        Знайти основні цикли, які мають одну спільну ланку з pcycle
        з Fc
        if Exist then
            Щоб створити новий pcycle, об'єднати pcycle із найменшим
            циклом
            Додати новий pcycle до listpcycle
            pcycle  $\leftarrow$  new pcycle
        else
            pcycle  $\leftarrow \phi$ 
        end if
    end while
end for
Candidate p-cycles  $\leftarrow$  listpcycle  $\cup$  Fc

```

### 6.3 Результати та аналіз

Основна мета – оцінити ефективність запропонованого алгоритму з точки зору оптимальності рішення та часу обчислення. Проводиться також порівняння, наскільки результати роботи запропонованого алгоритму відрізняється від оптимального рішення. Перш за все, оцінюється набір визначених основних циклів. Кількість основних циклів і час обчислення запропонованого алгоритму перевіряються в різних мережевих топологіях. Також перевіряється оптимальність рішення та складність запропонованого підходу. Для середньої та великої мережі використовуються дві тестові мережі, а саме мережі COST239 та мережі USA network. Мережа COST239 також є щільною мережею, як обговорювалося раніше. Оскільки метою підходу є лише мінімізація загальної використаної резервної пропускної здатності, робоча пропускна здатність у кожній ланці генерується випадковим чином із приблизно 24%~36% максимальної.

#### 6.3.1 Створення множини основних циклів

Перше завдання полягає в тому, щоб створити всі можливі основні цикли для двох тестових мереж. Щоб перевірити надійність набору основних циклів, виконується порівняння їх із набором основних циклів, створених BFS. BFS спочатку створює набір усіх можливих циклів у мережі, а потім обираються нові основні цикли з набору.

У мережі COST239 загалом 3531 цикл, і всі 42 основні цикли генеруються BFS. Мережа USA network має 47 основних циклів із загальної кількості 7321 циклів. Запропонований алгоритм знаходить таку саму кількість фундаментальних циклів, як і BFS, в обох цих мережах.

У таблиці 6.1 представлені отримані результати та мережева інформація для тестових мереж. Виявлено, що отримані результати моделювання чудово узгоджуються з результатами, отриманими BFS.

Оскільки проводилася перевірка надійності алгоритму АПШГ, можна створити будь-який можливий набір циклів для тестових мереж на основі набору основних циклів. Крім того, було встановлено, що час обчислення для алгоритму АПШГ на багато менший, ніж у BFS.

Таблиця 6.1 – Усі можливі основні цикли в COST239 і USA network

Мережа	COST239	USAnet
Усі можливі цикли	3531	7321
Усі основні цикли за АПШГ	42	46
Основні цикли BFS	42	46

### 6.3.2 Оптимальність рішення та обчислювальна складність

У наступному моделюванні перевіряється оптимальність рішень і обчислювальну складність запропонованого підходу. Проводиться порівняння результатів запропонованих рішень з оптимальною моделлю ІЛР, використовуючи всі цикли як набір кандидатів. Складність порівнюється через кількість кандидатів р-циклів, введених у модель ІЛР.

Спочатку проводиться обчислення набору ефективних р-циклів кандидатів за допомогою алгоритму генерації набору р-циклів. Потім для досягнення найкращої продуктивності використовується модель ІЛР. У таблиці 6.2 показано продуктивність запропонованого алгоритму для мережі COST239. Діапазон загального попиту на трафік встановлено від 200 до 300. Щоб оцінити валідність алгоритму, виконується ІЛР із перерахуванням повного циклу з тим самим попитом на трафік у тестовій мережі.

З проведеного моделювання виявлено, що досягнуті результати є близькими до оптимальних, як показано в таблиці 6.2. ІЛР використовує 3531 р-цикл для зближення до оптимального рішення. У найкращому випадку (попит на трафік 220) резервування від запропонованого алгоритму становить 33,2%, що є таким самим результатом, як і рішення ІЛР. У гіршому випадку (попит на трафік 300) оптимальний результат – 29,5, тоді як у

запропонованому алгоритмі – 33,7. Таким чином, враховуючи однакові вимоги до трафіку, можна досягти рішень у межах 4% від оптимальних рішень із набору лише 253 р-циклів. Ця кількість кандидатів на р-цикли становить лише 7% від усіх можливих циклів. Крім того, необхідний час обчислення становить менше 2% часу, необхідного для визначення повністю оптимальних рішень для кожного попиту на трафік.

Таблиця 6.2 – Результати моделювання для топології COST239

	COST239 (C = 32, р-цикли:3531)						
	Пропускна здатність	200	220	240	260	280	300
Модель ILP	Резерв %	36.5	33.3	32.1	30.7	30	30.3
	Час (с)	912	19379	17135	16121	21600	7341
Запропонований алгоритм	Резерв %	37	33.2	32.2	32.3	32.6	33.7
	Час (с)	2	369	21	41	105	71
	Кількість циклів	253	253	253	253	253	253

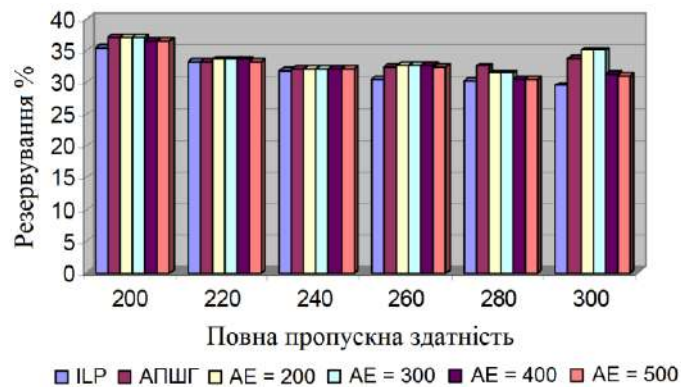
Таблиця 6.3 – Результати моделювання для топології USAN

	USAN (C = 32, р-цикли:7321)						
	Пропускна здатність	410	430	450	470	490	510
Модель ILP	Резерв %	69.5	64.2	68.7	69.7	68.8	68.7
	Час (с)	1699	7753	8385	1068	869	4686
Запропонований алгоритм	Резерв %	71	64.9	70.5	71.7	70.5	71
	Час (с)	27	3	4	31	47	79
	Кількість циклів	302	302	302	302	302	302

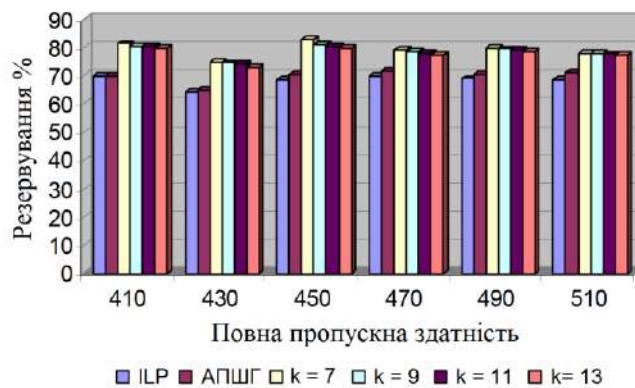
У таблиці 6.3 представлені відповідні результати для топології мережі USA network. Загальна потреба в трафіку становить від 410 до 510. Ці результати також вказують на майже оптимальні рішення. Надлишковість у результатах лише трохи вище оптимальних рішень. Моделювання показує, що результати запропонованого алгоритму лише на 0,5% нижче оптимального рішення в найкращому випадку. Загальна кількість р-циклів, знайдених алгоритмом, становить менше 4% від усіх можливих циклів.

Щоб оцінити результати моделювання, ми порівнюємо їх із підходами АЕ та WDCS розглянутими в розділі 4.6. За допомогою АЕ кількість кандидатів на р-цикли визначається з високих значень АЕ. Кількість р-циклів встановлено від 200 до 500.

На рисунку 6.3 (а) представлено результат моделювання для COST239. Помічено, що продуктивність, досягнута АЕ, близька до повного рішення ІЛР і запропонованого підходу. Цікаво, що кожен результат подібний, коли збільшується кількість циклів. Це тому, що кандидати на р-цикли не містять малих циклів. Однак іноді необхідні невеликі цикли. Наприклад, якщо проміжок із незахищеною ємністю залишився після того, як захищено працездатність за допомогою циклу високої ефективності, цикл високої ефективності витрачає пропускну здатність мережі, і цей випадок може погіршити результати.



(а) Мережа COST239



(б) Мережа USAN

Рисунок 6.3 – Результати порівняння АЕ і запропонованого алгоритму

Для мережі USA network продуктивність AE близька до рішення ILP, як показано на рисунку 6.3 (б). Однак можна помітити, що набір р-циклів запропонованого алгоритму може покращити ефективність більше, ніж набір р-циклів, отриманий за допомогою AE.

За допомогою підходу WDCS кількість р-циклів збільшується в залежності від параметра  $k$ , як згадувалося раніше. На рисунку 6.4 показано зміну кількості кандидатів на р-цикли зі збільшенням  $k$ .

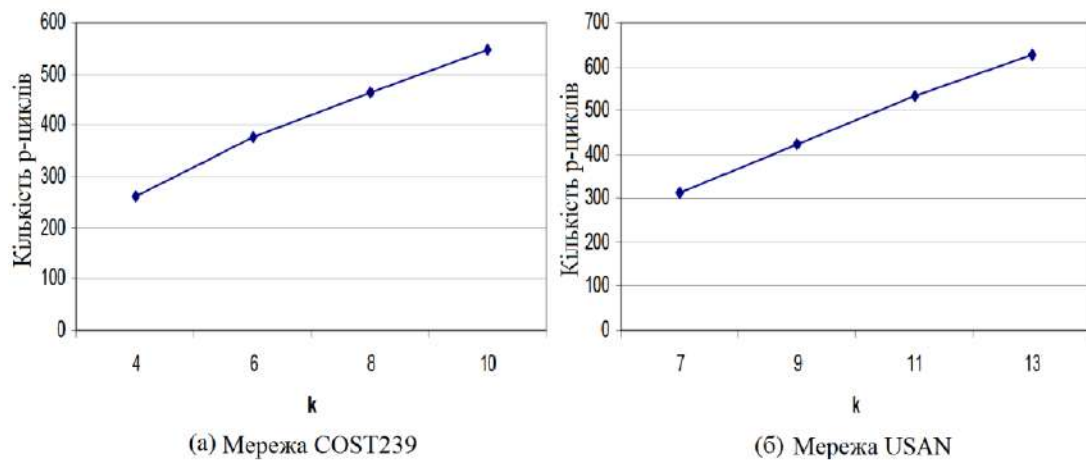


Рисунок 6.4 – Кількість потенційних р-циклів, досягнутих WDCS

Результати моделювання, що показують кількість циклів як функцію  $k$ , представлені на рисунку 6.5. Для топології COST239 результати моделювання запропонованого алгоритму кращі, ніж WDCS з такою ж кількістю кандидатів на р-цикли. У найкращому випадку результат становить 1,9% від оптимального рішення, тоді як WDCS становить 5%. Однак WDCS досягає трохи кращих результатів при збільшенні  $k = 10$  при попиті на трафік до 280. Це очікувано, оскільки збільшення  $k$  створює більше кандидатів р-циклів.

Для топології USA network потенційні р-цикли запропонованого алгоритму можуть досягти продуктивності в межах 3% від рішення ILP, тоді як WDCS досягає результатів лише в межах 12% від оптимального. У

гіршому випадку результати алгоритму АПШГ знаходяться в межах 2,4% від оптимальних, тоді як WDCS знаходиться в межах 14%, як показано на рисунку 6.5 (б).

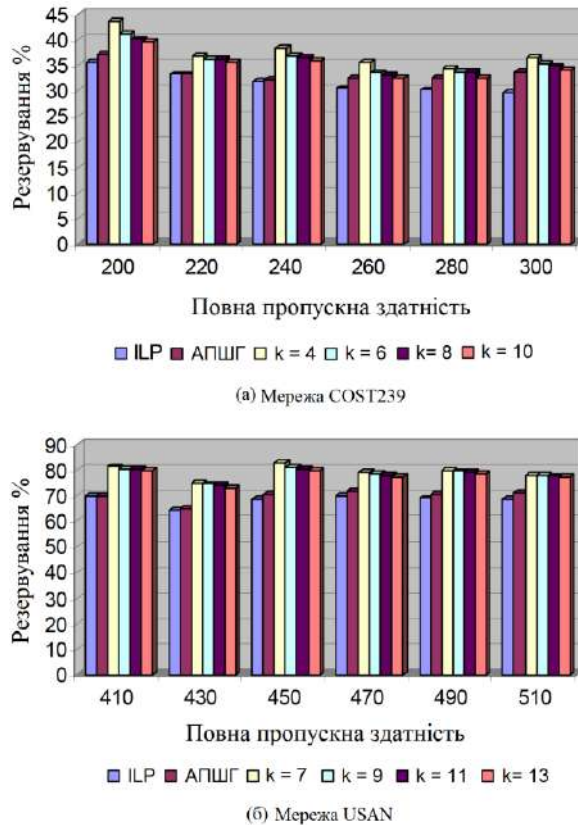


Рисунок 6.5 – Порівняння WDCS та запропонованого алгоритму

На рисунку 6.6 показано відхилення результатів роботи алгоритмів від оптимального значення ( $\Delta_{\text{опт}}(\%)$ ), коли АЕ і  $k$  становлять 300 циклів і 8 відповідно. Ці результати чітко показують, що запропонований нами підхід здатний досягти кращого рішення для мережі США і, отже, ефективний для щільних мереж.

У потенційних  $p$ -циклах, згенерованих запропонованим алгоритмом АПШГ, максимальна АЕ становить 3,73, а середня АЕ становить 2,39 для COST239. Для мережі США максимальні/середні значення АЕ становлять 2,21/1,59 відповідно. Крім того, алгоритм створює цикли з мінімальною АЕ 1, оскільки основні цикли також належать до кандидатського набору  $p$ -циклів.

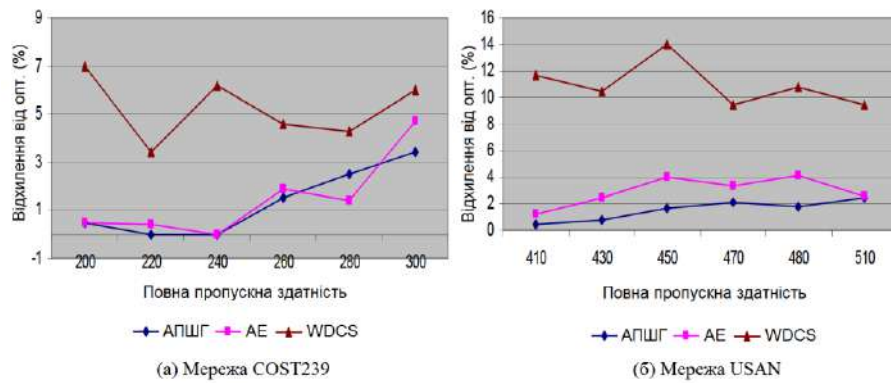


Рисунок 6.6 – Відхилення результатів роботи алгоритмів від оптимального значення

Таким чином, запропоновано ефективний алгоритм для обчислення достатнього набору р-циклів для забезпечення 100% захисту працездатності з мінімальними вимогами до резервної ємності за допомогою формулювання ІЛР. Алгоритм складається з двох етапів. Перший етап генерує всі можливі основні цикли, а другий етап обчислює достатні кандидатські р-цикли з них. Згенеровані основні цикли можна використовувати для побудови всіх можливих циклів мереж.

Для двох змодельованих тестових мереж було виявлено, що запропонований метод може досягти майже оптимальних рішень за допомогою ІЛР. Кількість згенерованих кандидатів р-циклів становить менше 7% від усіх можливих циклів для тестових мереж, а рішення близькі до оптимальних. Тобто запропонований метод здатний досягти задовільних результатів лише з невеликою кількістю кандидатів на р-цикли. Однак метод все ще базується на формулюванні ІЛР із пов'язаною з ним обчислювальною складністю. У наступному розділі буде розглянуто чистий евристичний підхід до проектування мережі р-циклу, щоб уникнути обчислювальної складності моделей ІЛР.

## 7 ЕВРИСТИЧНИЙ ПІДХІД ДЛЯ ПРОЕКТУВАННЯ ЖИВУЧИХ МЕРЕЖ WDM

Метою пропонованого алгоритму є забезпечення 100% захисту при мінімізації загальної резервної пропускної здатності та скороченні часу обчислення.

Щоб досягти такої мети, необхідно спочатку побудувати всі можливі цикли, використовуючи запропонований алгоритм попереднього обчислення всіх циклів (АПОЦ).

Адекватний набір кандидатів на р-цикли потім вибирається за допомогою критеріїв відповідності між розподілом робочих каналів і розподілом захищених каналів у вибраному наборі циклів.

### 7.1 Евристичний підхід для ефективного вибору циклу (ЕЕВЦ)

У запропонованому алгоритмі всі цикли попередньо обчислені. Це тому, що якщо ведеться пошук всіх циклів для великих щільних мереж, час обчислення буде дуже довгим.

Це означатиме, що евристичний підхід матиме ту саму проблему обчислювальної складності, що й модель ІЛР. Таким чином, попередньо обчислюються всі можливі цикли відповідно до РСА перед тим, як запустити запропонований алгоритм.

Потім, щоб визначити відповідний цикл, необхідно обчислити потенціальну захисну здатність ( $\Phi$ ) кожного циклу. Кожен цикл має різну  $\Phi$ , оскільки кожен цикл має різні зв'язки. Потенційна захисна здатність визначається як:

$$\Phi_p = \sum_{i=1}^{M_i} \text{cycle}_{p,i}, \quad (7.1)$$

$$\text{cycle}_{p,i} = \begin{cases} 2 & \text{якщо } i \text{ є розрізненим проміжком } \text{cycle}_p \\ 1 & \text{якщо } i \text{ є проміжком циклу } \text{cycle}_p \\ 0 & \text{інакше} \end{cases} .$$

У запропонованому алгоритмі  $\Phi$  також попередньо обчислюється. Причина полягає в тому, щоб уникнути проблеми складності обчислень. Відповідні цикли визначаються за допомогою вибору циклу на основі стандартного відхилення (АВЦСВ) з трьома факторами, середнім внеском незахищеної пропускної здатності ( $\Lambda$ ), стандартним відхиленням незахищеної пропускної здатності ( $\Xi$ ) і втраченої резервної пропускної здатності ( $\Phi$ ).

Алгоритм продовжує пошук циклів до досягнення 100% захисту всієї працездатності. Конструкція цього евристичного алгоритму підсумовується таким чином.

Крок 1. Попередньо обчислити всі цикли в мережі за допомогою РСА.

Крок 2. Для кожного циклу-кандидата попередньо обчислити  $\Phi$  циклу за допомогою формули (7.1).

Крок 3. Обрати найкращий цикл за допомогою АВЦСВ.

Крок 4. Видалити пропускну здатність, захищену вибраним циклом на кроці 3, і оновити пропускну здатність.

Крок 5. Перейти до кроку 3, доки на кожному прольоті не закінчиться робоча пропускну здатність.

DFS і BFS є основними алгоритмами для визначення всіх циклів. Як згадувалося раніше, ці алгоритми зберігають вузли під час їх пошуку в черзі та досліджують їх у такому порядку. Однак у запропонованому алгоритмі АПОЦ спочатку досліджуються всі можливі цикли, що проходять через початковий вузол.

Потім кожен вузол поступово досліджується, доки не буде виявлено кожен вузол. Метою АПОЦ є побудова всіх можливих циклів у даній мережі. Псевдокод АПОЦ показано у лістингу 7.1.

## Лістинг 7.1 – Попереднє обчислення всіх циклів (АПОЦ)

```

Input: Неорієнтований граф  $G(V, E)$ 
Output: набір усіх циклів
 $i \leftarrow 1$   $N_0 = |V|$ 
while  $N_0 \geq 3$  do
    Знайти всі цикли, що проходять через вузол  $i$ , використовуючи
    алгоритм АПВ (Лістинг 7.2)
    Видалити вузол  $i$  з  $G(V, E)$ 
     $i = i + 1$ 
     $N_0 = N_0 - 1$ 
end while

```

Усі цикли, що проходять через вузол, досліджуються запропонованим алгоритмом проходження кожного вузла (АПВ). Алгоритм АПВ схожий на алгоритм АПШГ, очікуючи, що коли він стикається з першим основним циклом, де вузол призначення підключається лише до вузла джерела, він не зупиняється після побудови одного циклу. Натомість АПВ продовжує пошуки циклів, що проходять через вузол.

Існує дві умови, які повинні бути виконані для впровадження РСА. По-перше, видаляються вузли з вузловим ступенем 1 із топології мережі, оскільки з цих вузлів не може бути сформований цикл. По-друге, від'єднується кожне ребро, яке з'єднується з кореневим вузлом після того, як знайдено всі цикли через вузол. Це робиться для того, щоб уникнути перерахування повторних циклів. Вибір циклу на основі стандартного відхилення (АВЦСВ). Основне завдання АВЦСВ – вибрати відповідний набір циклів для захисту.

Серед існуючих евристичних підходів одна з можливих схем для визначення набору  $p$ -циклів полягає в обчисленні захищеного використання пропускнуої здатності для кожного циклу-кандидату. Алгоритм також вимірює використання захищеної ємності шляхом обчислення середньої незахищеної ємності після видалення ємності, захищеної вибраним циклом.

## Лістинг 7.2 – Псевдокод алгоритму проходження кожного вузла (АПВ)

Input: Неорієнтований граф  $G(V, E)$ , кореневий вузол

Output: Кандидати цикли

$Nei \leftarrow$  множина сусідів кореневого вузла

if  $|Nei| < 2$  then

    return

end if

for  $i = 1$  to  $|Nei| - 1$  do

$s \leftarrow Nei(i)$

    pathlist = [root s]

    while Pathlist is not empty do

        currentpath  $\leftarrow$  pathlist

        parentnodes  $\leftarrow$  currentpath

$N \leftarrow$  останній вузол поточного шляху

        remove currentpath from pathlist

        list  $\leftarrow$  neighbor node of  $N$

        Видалити поточний шлях зі списку шляхів

        for each  $k$  node of list do

            if  $k$  is root then

                Створити цикл

            else

                Newpath = [currentpath,  $k$ ]

                Додати newpath у список шляхів pathlist

            end if

        end for

    end while

    disconnect  $s$  from root

end for

Це середнє значення показує, наскільки розподіл незахищеної ємності зменшується під час вибраного циклу. Середнє значення ( $\Lambda$ ) виражається як:

$$\Lambda_j = \text{mean}(\Delta_j), \quad (7.2)$$

$$\Lambda_j = \text{mean}(\Delta_j), \quad (7.2)$$

$$\Delta_j = \acute{\omega} - \Phi_j. \quad (7.3)$$

Якщо кількість циклів, вибраних  $\Lambda$ , дорівнює одному, обирається цикл. Якщо ні, ми застосовуємо стандартне відхилення  $\Lambda(\Xi)$ , щоб вибрати наступний найкращий цикл.  $\Xi$  дає більше інформації для вибору циклу. Наприклад, якщо цикл, вибраний  $\Lambda$ , захищає лише деяку незахищену пропускну здатність, він іноді залишатиме значну втрату робочої пропускну здатності на деяких проміжках. Тобто  $\Xi$  показує, наскільки розповсюджується кожен цикл для покриття незахищеної ємності. Таким чином, використовується  $\Xi$  і вибирається його мінімальне значення з набору  $\Xi$ .  $\Xi$  визначається як:

$$\Phi_p = \sum_{s=1}^{M|} (\Delta_s - \Lambda_s)^2. \quad (7.4)$$

Після того як  $\Xi$  і  $\Lambda_s$  відомі, останній критерій рішення застосовується, якщо кількість циклів все ще не один. Цей критерій використовується для зменшення обсягу резервних потужностей. Втрачена резервна ємність – це робоча ємність для проміжків циклу, тих, які не захищають жодної робочої ємності, але повинні займати резервну ємність. Якщо визначити цикл із найменшим ступенем резервної ємності, але захищеним таким самим використанням ємності, можна досягти кращої продуктивності.

Втрачена резервна пропускну здатність  $\Phi_r$  обчислюється як:

$$\Phi_p = \frac{\sum_{i=1}^{L|} W_{p,i}}{\sum_{i=1}^{L|} C_{p,i}}, \quad (7.5)$$

$$W_{p,i} = \begin{cases} 1 & \text{якщо } S_{p,i} \text{ менше нуля} \\ 0 & \text{інакше} \end{cases},$$

$$C_{p,i} = \begin{cases} 1 & \text{якщо } i \text{ є проміжком циклу } \text{cycle}_p \\ 0 & \text{інакше} \end{cases}.$$

Підхід АВЦСВ визначає лише один найкращий цикл з усіх кандидатів на р-цикли.

Лістинг 7.3 – Псевдокод алгоритму вибора циклу на основі стандартного відхилення (АВЦСВ)

Input: робоча пропускна здатність,  $\Phi$

Output: *цикл*

Unprotected capacity( $\$$ )  $\leftarrow$  робоча пропускна здатність

Обчислити  $(\Delta, \Lambda)$  of  $|\Phi|$  відповідно до формул (7.2) та (7.3)

Визначити набір циклів ( $\eta$ ) з мінімальним значенням  $\Lambda$

Обчислити  $\Xi$  від  $\eta$  за рівнянням (7.4)

Перевірити набір циклів ( $\Theta$ ), який є мінімальним значенням у  $\Xi$

Обчислити резервну пропускну здатність ( $\Psi$ ) набору  $\Theta$  за допомогою рівнянням (7.5)

Вибрати цикл із мінімальним  $\Psi$  із множини  $\Theta$

## 7.2 Результати моделювання та аналіз

Було розглянуто чотири добре відомі мережі: NSFNet, COST239, EON і USA networks. Порівнюється продуктивність запропонованого евристичного алгоритму з оптимальними рішеннями та звичайними евристичними підходами, щоб оцінити ефективність запропонованого підходу.

Перш за все, реалізується алгоритм АПОЦ для обчислення всіх циклів у заданих мережах. Таблиця 7.1 показує ці результати. Вони представлені у вигляді кількості циклів і часу обчислення. З результатів видно, що запропонований алгоритм надійний протягом розумного часу роботи.

Таблиця 7.1 – Усі цикли, згенеровані АПОЦ для тестових мереж

Мережа	NSFNet	COST239	EON	USA
Всі можливі цикли	139	3531	8858	7321
Всі цикли за АПОЦ	139	3531	8858	7321
Час роботи АПОЦ	0.68	12.42	63.82	2384

Після визначення циклів-кандидатів порівнюються характеристики з формулюванням ІЛР, евристичним алгоритмом CIDA і алгоритмом Жанга (Zhang). Кожен алгоритм під час моделювання виконується з усіма циклами. Для кожної мережі показано час обчислення та відсоток відмінності від оптимального рішення (Відхилення (%)).

Для кожної мережі результати моделювання наведені в таблицях 7.2, 7.3, 7.4 і 7.5. Середні результати описані в таблиці 7.6, а рисунок 7.1 показує відмінності від рішень ІЛР. Порівняння показує, що результати CIDA та Zhang абсолютно однакові. Єдина різниця між ними – час обчислення. Час роботи Zhang у середньому трохи довший, ніж CIDA, як показано в таблиці 7.6.

### 7.2.1 Результати моделювання для топології NSFNet

Загальна пропускна здатність варіюється від 193 до 287. Можна спостерігати з чисельних результатів, що продуктивність алгоритму ЕЕВЦ дуже близька до CIDA в цій тестовій мережі. У найкращому випадку CIDA на 1,1% гірше оптимального рішення, тоді як ЕЕВЦ становить 2,4%. Час обчислення вимагає лише 0,03 (с) для обох цих підходів. Тобто можна спостерігати значне покращення часу обчислення запропонованого підходу

порівняно з рішенням ILP. Крім того, надмірність середніх результатів показує, що ЕЕВЦ і CIDA становлять 2,87% і 3,68% від оптимального рішення протягом 0,1(с) часу обчислення.

Таблиця 7.2 – Результати моделювання для топології NSFNet

	NSFNet (C = 32)										
	Пропускна здатність	193	204	212	225	237	243	257	264	271	287
ILP	Резерв %	68.4	64.2	69.7	64	67	60.7	68.1	66.3	65.4	65.5
	Час (с)	2	8	17485	12	11.8	2.2	92.9	12	2	19.3
CIDA	Резерв %	73	68.1	73.1	68.4	70.3	65	73	70.1	69.8	66.6
	Час (с)	0.03	0.03	0.02	0.02	0.03	0.02	0.03	0.03	0.03	0.02
Zhang	Резерв %	72	68.2	73.1	68.4	70.2	65	72	70.3	69.7	66.6
	Час (с)	0.17	0.07	0.12	0.15	0.07	0.11	0.08	0.12	0.08	0.07
ЕЕВЦ	Резерв %	69.4	69.1	72.3	70.2	70.3	63.3	70.8	70.1	68.6	67.9
	Час (с)	0.03	0.03	0.02	0.02	0.03	0.04	0.02	0.03	0.03	0.03

### 7.2.2 Результати моделювання для топології COST239

Кількість попиту на трафік коливається від 240 до 340. Результати моделювання показали, що в середньому рішення, досягнуті CIDA, близькі до оптимального рішення з різницею лише на 3,99% від рішення ILP. У найкращому випадку CIDA на 0,3% відрізняється від оптимальних результатів і з часом обчислення 0,2(с). У гіршому випадку (попит на трафік 295), CIDA знижується на 7,1%. Цей результат показує, що впровадження CIDA може призвести до найгіршого випадку захисту. найгірший випадок відбувається з впровадженням CIDA.

З іншого боку, показано, що продуктивність ЕЕВЦ набагато краща, ніж у CIDA. ЕЕВЦ може досягти відхилення -2,8% від оптимального рішення протягом 0,3 (с), коли використовується попит на трафік 295. Слід зазначити, що в експерименті не обмежується час обчислення ILP до максимуму 21000 (с), тому результат кращий за оптимальний. У гіршому випадку ЕЕВЦ становить лише 1,1% відхилення від оптимального рішення.

Продуктивність ЕЕВЦ у середньому в 184 рази краща, ніж у CIDA, а час роботи становить приблизно  $2 \cdot 10^{-5}$  часу, необхідного для оптимального рішення. Коротше кажучи, запропонований підхід може досягти майже оптимального рішення за прийнятний час обчислення.

Таблиця 7.3 – Результати моделювання для топології COST239

	COST239 (C = 32)										
	Проп. здатніс.	244	254	265	273	287	295	301	318	322	332
ILP	Резерв %	31.6	29.1	30.2	30	30.3	29.2	29.2	30.5	32.9	29.1
	Час (с)	21601	17194	21600	21605	2137	21600	3872	21600	21600	4873
CIDA	Резерв %	35.7	32.3	36	33.3	33.8	36.4	33.1	34.6	33.1	34.7
	Час (с)	0.24	0.2	0.22	0.22	0.2	0.2	0.22	0.3	0.2	0.2
Zhang	Резерв %	35.7	32.3	34	33.3	33.8	36.5	33.1	34.6	33.2	34.8
	Час (с)	0.32	0.2	0.27	0.22	0.21	0.19	0.24	0.32	0.24	0.24
ЕЕВЦ	Резерв %	32.4	29.6	30.1	30.4	31.3	29.2	29.1	30.3	30.1	29.8
	Час (с)	0.3	0.25	0.3	0.25	0.31	0.29	0.34	0.34	0.31	0.32

### 7.2.3 Результати моделювання для топології EON

Тепер проведемо моделювання топології EON, а потім порівнюємо продуктивність цієї топології з іншими підходами, такими як CIDA та ILP. Результати моделювання показують, що CIDA пропонує значно швидший час обчислення порівняно з моделлю ILP. Однак слід зауважити, що рішення, створені за допомогою CIDA, мають велику розбіжність порівняно з моделлю ILP.

Наприклад, у найгіршому випадку CIDA знаходиться на 11,5% від оптимального рішення, а найкращі знайдені рішення CIDA на 8,3%.

Однак якщо зосередитися на часі роботи, CIDA може зробити модель ILP розв'язною з набагато меншою обчислювальною складністю (у 4595 разів швидше, ніж ILP). За допомогою підходу ЕЕВЦ результати показують, що можна знайти баланс між оптимальністю рішення та складністю обчислень.

Таблиця 7.4 – Результати моделювання для топології EON

	EON (C = 32)										
	Пропускн. здатність	355	367	377	384	392	405	411	427	436	445
ILP	Резерв %	52.9	58.6	53.5	51	53.6	54.8	53	55	53.2	49.9
	Час (с)	3915	21600	3828	197	1644	8631	21600	214	8686	21600
CIDA	Резерв %	64.4	66.9	63.3	60.2	64	64.4	65	65.8	63.5	60
	Час (с)	1.8	1.9	1.8	1.7	2	2	2.2	2.1	2.1	2
Zhang	Резерв %	64.4	66.9	63.3	60.2	64	64.4	65	65.8	63.5	60
	Час (с)	2.2	2.34	2.14	2.21	2.73	2.67	2.8	2.91	2.89	2.6
ЕЕВЦ	Резерв %	56.3	59.2	55.1	56.3	56.1	56.3	54.7	57.4	56.4	52.8
	Час (с)	1.9	1.9	1.8	2.3	2.2	2	2.3	2.5	2.6	2.2

## 7.2.4 Результати моделювання для топології USA networks

Ці результати показують, що за допомогою алгоритму ЕЕВЦ можна досягти майже оптимального споживання резервної ємності за набагато менший час роботи, ніж вимагає ILP. Виявлено, скорочення часу обчислень ставить приблизно  $2 \cdot 10^{-4}$  часу, необхідного для пошуку оптимального рішення.

У розподілі попиту на трафік між 520 і 530, ЕЕВЦ відхилення становить 1,1% від рішення ILP і приблизно у 8 разів краще, ніж рішення CIDA.

Таблиця 7.5 – Результати моделювання для топології EON

	USAN (C = 32)										
	Пропускн. здатність	458	467	476	485	494	507	515	526	537	544
ILP	Резерв %	68.6	67.2	71	65.2	67.8	67.9	65.8	67.5	69.5	68.6
	Час (с)	1098	21600	944	399	5109	872	593	2294	16775	21600
CIDA	Резерв %	77.1	73.9	76.9	72.4	75.5	74	73.4	75.9	75.8	77.6
	Час (с)	1.8	1.8	1.5	1.3	1.5	1.5	1.5	1.6	1.7	1.9
Zhang	Резерв %	64.4	66.9	63.3	60.2	64	64.4	65	65.8	63.5	60
	Час (с)	2.2	2.34	2.14	2.21	2.73	2.67	2.8	2.91	2.89	2.6
ЕЕВЦ	Резерв %	72.7	71.3	75	68	70.5	70	67.6	68.6	72.4	72.2
	Час (с)	1.4	1.3	1.6	1.4	1.4	1.6	1.4	1.6	1.5	1.7

З таблиці 7.6 можна побачити, що отримані результати в середньому становлять 2,92% від оптимальних, тоді як CIDA становить 7,34%. Крім того, результати показують, що середній час роботи для ЕЕВЦ трохи швидший, ніж вимагається CIDA.

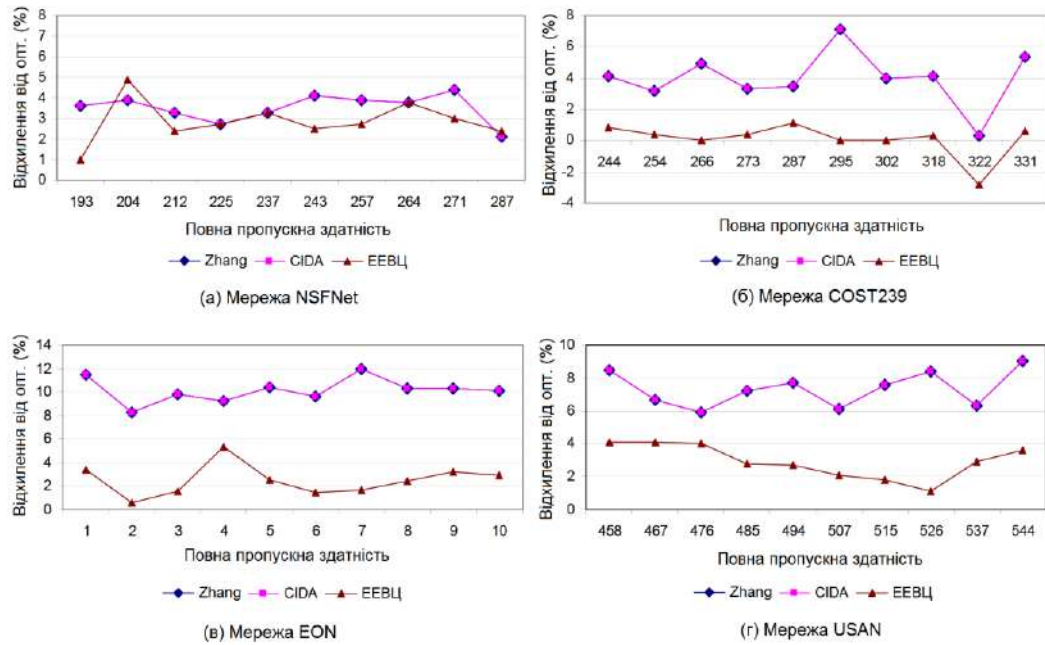


Рисунок 7.1 – Час обчислення та відсоток відмінності від оптимального рішення

Таблиця 7.6 – Середні результати в кожній тестовій мережі

		ILP	CIDA	Zhang	ЕЕВЦ
NSFNet	Резерв %	65.95	69.63	69.63	68.82
	Час (с)	1764	0.03	0.11	0.05
	Відхилення	0	3.68	3.68	2.87
COST239	Резерв %	30.21	34.2	34.2	30.23
	Час (с)	15767	0.22	0.22	0.3
	Відхилення	0	3.99	3.99	0.02
EON	Резерв %	53.55	63.75	63.75	56.06
	Час (с)	9191	1.96	2.55	2.17
	Відхилення	0	10.2	10.2	2.51
USAN	Резерв %	67.91	75.25	75.25	70.83
	Час (с)	7128	1.61	2.96	1.49
	Відхилення	0	7.34	7.34	2.92

З отриманих результатів видно, що запропонований алгоритм ЕЕВЦ має баланс між продуктивністю резервування пропускної здатності та часом обчислення. Тому ЕЕВЦ є більш привабливим для щільних і великих мереж. Що стосується COST239, ЕЕВЦ може досягти майже оптимального рішення протягом 1(s). Для EON і USAN результати перевершують існуючі підходи та мають розумний час обчислення.

Таким чином, представлено та обговорено евристичний алгоритм до ефективного вибору р-циклів ЕЕВЦ, який уникає використання моделі ІЛР.

## ВИСНОВКИ

Метою кваліфікаційної магістерської роботи є дослідження та розробка алгоритмів підвищення живучості мережі на основі концепції р-циклів в оптичних сітчастих мережах WDM з метою балансування оптимальності складності рішення та часу відновлення працездатності мережі.

Запропоновано два нових евристичних підходи до обчислення ефективних р-циклів, які можуть призвести до кращої продуктивності з точки зору ефективності використання пропускну здатності, евристичний підхід, заснований на формулюванні ILP, і чистий евристичний підхід.

З метою підтверження теоретичних викладок було проведено імітаційне моделювання. Результати моделювання показують, що запропоновані алгоритми перевершують існуючі евристичні підходи за показниками. Таким чином, можна зробити висновок, що обидва підходи дуже корисні для досягнення кращої мережевої продуктивності та надійності функціонування у випадках щільних і великих оптичних мереж WDM, що відповідає основній меті кваліфікаційної магістерської роботи.

## ПЕРЕЛІК ДЖЕРЕЛ ПОСИЛАННЯ

1. A. Agrawal, "Algorithms for the Logical Topology Design in WDM All-Optical Networks", aagrwal@usc.edu, March 2002.
2. C. Qiao, "Labeled Optical Burst Switching for IP-over-WDM Integration", IEEE Communications Magazine, vol. 38, no. 9 pp. 104-114, September 2000.
3. L. Sahasrabudhe, S. Ramamurthy, B. Mukherjee, "Fault management in IP-over-WDM networks: WDM protection versus IP restoration" IEEE Journal on Selected Areas in Communications, Vol. 20, No. 1, pp. 21 - 33. January 2002.
4. S. Ramaswami, B. Mukherjee, "Survivable WDM Mesh Networks, Part I- Protection", Proceeding of IEEE INFOCOM'99, pp. 744-751, March 1999.
5. Y. P. Aneja, S. Bandyopadhyay and A. Jaekel, "An efficient protection scheme for WDM Networks using dynamic lightpath allocation", HPC-Asia, December 2002.
6. M. Listanti, V. Eramo, R. Sabella, "Architectural and Technological Issues for Future Optical Internet Networks". IEEE Communication Magazine, vol. 38, No. 9, pp. 82- 92, September, 2000.
7. R. Dutta, G. N. Rouskas, "A survey of Virtual Topology Design Algorithms For Wavelength Routed Optical Networks", Optical Networks Magazine, vol. 1, no. 1, pp. 73-89, January 2000.
8. A. Agrawal, "Algorithms for the Logical Topology Design in WDM All-Optical Networks", aagrwal@usc.edu, March 2002.
9. P. Tomsu, C. Schmutzer, "Next Generation Optical Networks", Prentice-Hall, Inc. 2002.
10. D. S. W. D. Grover, "Bridging the ring-mesh dichotomy with p-cycles," in Design of Reliable Communication Networks (DRCN 2000), (Munich, Germany), pp. 92 – 104, 2000.
11. Ramamurthy, L. Sahasrabudhe, and B. Mukherjee, "Survivable wdm

mesh networks,” *Lightwave Technology, Journal of*, vol. 21, no. 4, pp. 870–883, 2003.

12. G. W. D., *Mesh-Based Survivable Networks*. Prentice Hall, 2003

13. W. Grover and J. Doucette, “Advances in optical network design with p-cycles: Joint optimization and pre-selection of candidate p-cycles,” tech. rep., 2002.

14. H. Zhang and O. Yang, “Finding protection cycles in dwdm networks,” in *IEEE International Conference on Communication*, vol. 5, pp. 2756–2760, 2002.

15. C. Liu and L. Ruan, “Finding good candidate cycles for efficient p-cycle network design,” in *13th international Conference on Computer Communication and Networks (ICCCN 2004)*, pp. 321 – 326, 2004.

16. J. Doucette, D. He, W. D. Grover, and O. Yang, “Algorithmic approaches for efficient enumeration of candidate p-cycles and capacitated p-cycle network design,” *Design of Reliable Communication Networks (DRCN) 2003*, pp. 212 – 220, 2003.

17. Z. Zhang, W.-D. Zhong, and B. Mukherjee, “A heuristic method for design of survivable wdm networks with p-cycles,” *IEEE Communications Letters*, vol. 8, no. 7, pp. 467 – 469, 2004.

18. Z. Zhang, W. Zhong, and B. Mukherjee, “A heuristic algorithm for p-cycles configuration in wdm optical networks,” in *Opto-Electronics and Communications Conference (OECC)*, vol. 23, pp. 568 – 569, 2003.

19. C. G. Gruber, *Deployment of p-cycles in WDM Networks*. PhD thesis, Technische University Munchen, August, 2001.

20. H. N. Nguyen, D. Habibi, V. Q. Phung, S. Lachowicz, K. Lo, and B. Kang, “Joint optimization in capacity design of networks with p-cycle using the fundamental cycle set,” in *IEEE GLOBECOM, (USA)*, 2006.

21. I. Chlamtac, A. Ganz, and G. Karmi, “Purely optical networks for terabit communication,” in *IEEE INFOCOM*, vol. 3, (Washington DC), pp. 887 – 896, 1989.

22. Filippov V. VNF Placement and Connectivity // Vladlen Filippov, Ilya Voloshin, Stanislav Partyka, Oksana Baranova // Fifth International Scientific and Technical Conference «COMPUTER AND INFORMATION SYSTEMS AND TECHNOLOGIES». – April 22-23, 2021, Kharkiv – Riga – Kyiv – Lviv – Baku. – C 89-90.