

**НАЦІОНАЛЬНИЙ ТЕХНІЧНИЙ УНІВЕРСИТЕТ УКРАЇНИ
«КИЇВСЬКИЙ ПОЛІТЕХНІЧНИЙ ІНСТИТУТ
імені ІГОРЯ СІКОРСЬКОГО»**

Факультет прикладної математики

Кафедра системного програмування і спеціалізованих комп'ютерних систем

«На правах рукопису»
УДК 004.005

До захисту допущено:
Завідувач кафедри
_____Віталій РОМАНКЕВИЧ
«__»_____2022 р.

**Магістерська дисертація
на здобуття ступеня магістра
за освітньо-професійною програмою
«Системне програмування і спеціалізовані комп'ютерні системи»
зі спеціальності 123 «Комп'ютерна інженерія»**

на тему: «Модифікація способу формування GL-моделей для складних систем»

Виконав:

студент II курсу, групи КВ-11мп
Мережко Іван Петрович _____

Науковий керівник:

Професор кафедри СПСКС, докт. техн. наук.,
професор Романкевич О.М. _____

Рецензент:

Засвідчую, що у цій магістерській дисертації
немає запозичень з праць інших авторів без
відповідних посилань.

Студент _____

Київ – 2022 року

Національний технічний університет України
«Київський політехнічний інститут імені Ігоря Сікорського»
Факультет прикладної математики

Кафедра системного програмування і спеціалізованих комп'ютерних систем
Рівень вищої освіти – другий (магістерський)

Спеціальність – 123 «Комп'ютерна інженерія»

Освітньо-професійна програма «Системне програмування і спеціалізовані
комп'ютерні системи»

ЗАТВЕРДЖУЮ

Завідувач кафедри

_____ Віталій РОМАНКЕВИЧ

«__» _____ 2022 р.

ЗАВДАННЯ
на магістерську дисертацію студента

Мережко Іван Петровича

1. Тема дисертації «Модифікація способу формування GL-моделей для складних систем», науковий керівник дисертації докт. техн. наук., професор Романкевич О.М., затверджені наказом по університету № 4317-С від 25.11.2022

2. Термін подання студентом дисертації 12.12.2022

3. Об'єкт дослідження: багатопроцесорні відмовостійкі системи.

4. Предмет дослідження: методи та засоби розрахунку показників надійності відмовостійких багатопроцесорних систем шляхом виконання експериментів з моделями їх поведінки в потоці відмов.

5. Перелік завдань, які потрібно розробити:

1) Модифікувати алгоритм будування GL-моделей, для систем, де певні процесори можуть брати на себе роботи інших, під час їх відмов, але не навпаки.

6. Орієнтовний перелік графічного (ілюстративного) матеріалу: презентація.

7. Перелік публікацій:

1) Прикладна математика та комп'ютинг. «XV науково-практична конференція магістрантів та аспірантів ПМК-2022 факультету прикладної математики» (Київ, 16-

18 листопада 2022 р.).

2) IX Міжнародна науково-технічна Internet-конференція «Сучасні методи, інформаційне, програмне та технічне забезпечення систем керування організаційно-технічними та технологічними комплексами» (Київ, 25 листопада 2022 р.).

9. Дата видачі завдання

Календарний план

№ з/п	Назва етапів виконання магістерської дисертації	Термін виконання етапів магістерської дисертації	Примітка
1.	Вивчення літератури за тематикою МД	02.10.2020	
2.	Розроблення та узгодження технічного завдання	08.09.2022	
3.	Аналіз існуючих рішень	10.09.2022	
4.	Підготовка матеріалів розділів МД	16.09.2022	
5.	Модифікація існуючого алгоритму	02.10.2022	
6.	Тестування модифікованого алгоритму	15.10.2022	
7.	Оформлення документації дипломного МД	27.11.2021	
8.	Попередній розгляд магістерської дисертації на кафедрі	01.12.2022	

Студент

Іван МЕРЕЖКО

Науковий керівник

Олексій РОМАНКЕВИЧ

РЕФЕРАТ

Актуальність теми обумовлена збільшенням обсягів оброблюваної інформації, вимогам до швидкості її обробки і до надійності апаратних і програмних засобів, що використовуються для цих цілей. У зв'язку з цими обставинами системи обробки інформації і особливо системи управління проектуються відмовостійкими, багатопроцесорними і реконфігурованими, а сфера використання відмовостійких багатопроцесорних обчислювальних систем і систем управління все більше і більше розширюється. Найбільш широке поширення такі системи отримують у тих областях, де відмови і збої апаратури можуть привести до великих фінансових збитків, екологічних катастроф і людських жертв (банківські системи, атомні електростанції, системи управління літаків і ракет, медичні системи та ін).

При розробці відмовостійких багатопроцесорних систем (ВБС) дуже важливо вибрати найбільш оптимальний з точки зору надійності і вартості варіант структури ще на етапах проектування та визначити його так звані «вузькі» місця. Для цього необхідно створювати математичні моделі, що адекватно відбивають поведінку ВБС у потоці відмов і розробляти методи оцінки параметрів надійності ВБС на основі цих моделей, що дозволяють отримати результат обчислень в прийнятний час доступними обчислювальними засобами.

Об'єктом дослідження є багатопроцесорні системи у тому числі відмовостійкі та їх моделі поведінки у потоці відмов.

Предметом дослідження є методи і алгоритми будування GL-моделей для складних систем, що використовуються для розрахунку параметрів надійності.

Мета роботи. Модифікація алгоритму будування GL-моделей для складних відмовостійких реконфігурованих систем, де лише частина процесорів може брати на себе роботу інших під час їх відмови.

Методи дослідження. В роботі використовуються теорія графів та алгоритм будування GL-моделей, математична логіка, теорія автоматів.

Наукова новизна полягає в модифікації існуючого способу будування GL-моделей, яка дозволить створювати більш прості GL-моделі, зокрема з меншою

кількістю ребер.

Практична цінність модифікованого алгоритму полягає в тому, що вихідний граф має меншу кількість ребер, що дозволяє розраховувати надійність системи з більшою точністю.

Апробація роботи. Матеріали роботи були представлені на щорічній науково-технічній конференції магістрантів кафедри СКС ФПМ НТУУ "КПІ" ПМК-2022 і на IX Міжнародна науково-технічна Internet-конференція «Сучасні методи, інформаційне, програмне та технічне забезпечення систем керування організаційно-технічними та технологічними комплексами».

Структура і обсяг роботи.

У вступі обґрунтовується актуальність теми роботи, формулюється мета і завдання дослідження.

У розділі 1 наведені відомості про ВБС та переведенні основні поняття та характеристики теорії надійності .

У розділі 2 описано алгоритми будування графо-логічних моделей (GL-моделей) для базових ВБС 2-відмовостійких систем. Показано загальний підхід будування моделей для таких систем.

У розділі 3 розглянуті відомі методи розрахунку надійності базових ВБС (так званих k-out-of-n) систем.

У розділі 4 показана розроблена модифікація алгоритму будування GL-моделей. Наведені порівняльні приклади будування GL-моделей простим та модифікованим алгоритмами.

У висновках зроблено загальні висновки по роботі, проаналізовано отримані результати.

У додатку 1 наведено копії графічного матеріалу. Атестаційна магістерська робота представлена на 90 сторінках і складається з введення, 4 розділів, висновку і містить 6 рисунків, 15 таблиць, список використаних джерел з 24 найменувань.

Ключові слова: БАГАТОПРОЦЕСОРНІ ВІДМОВОСТІЙКІ СИСТЕМИ, НАДІЙНІСТЬ, GL-МОДЕЛЬ

ABSTRACT

The relevance of the topic is due to the increase in the volume of processed information, the requirements for the speed of its processing and the reliability of the hardware and software tools used for these purposes. In connection with these main information processing systems and especially control systems are designed to be fault-tolerant, multi-processor and reconfigurable, and the scope of use of fault-tolerant multi-processor computing systems and control systems is expanding more and more. The widest distribution of such systems is supported in those cases where failures and malfunctions of equipment can lead to large financial losses, environmental disasters and human casualties (banking systems, nuclear power plants, aircraft control and medical systems, etc.).

When developing fault-tolerant multiprocessor systems (FMS), it is very important to choose the most optimal structure option from the point of view of reliability and cost even at the design and application stages of the so-called "bottlenecks". For this, it is necessary to create mathematical models that adequately reflect the behavior of the VBS in failure flows and to develop methods for estimating the reliability parameters of the VBS based on these models, which can obtain the result obtained at the accepted time by available computing means.

The object of research is fault-tolerant systems that have many elements, or FBS (fault-tolerant multiprocessor systems).

The subject of research are methods and algorithms for building GL-models for complex systems that have certain criteria.

The goal of the work. Create a modification of the algorithm for building GL-models for complex systems, where some processors can take over the work of others during their failure.

Research methods. The paper deals with the theory of graphs and the algorithm for building GL-models.

Scientific news will not be able to modify the existing algorithm, which made it possible to more effectively create GL-models for a system with a certain criterion.

The practical value of the modified algorithm is that it allows more efficient construction of GL models, which in turn allows faster and more accurate calculation of system reliability.

Approbation of work. The materials of the work are presented at the annual scientific and technical conference of master's students of the department of SCS FPM NTUU "KPI" PMK-2022 and at the IX International scientific and technical Internet conference "Modern methods, information, software and technical support of management systems of organizational, technical and technological complexes".

Structure and scope of work.

The introduction substantiates the relevance of the topic of the work, formulates the purpose and task of the research.

Chapter 1 provides information about VBS and translated the main concepts and characteristics of reliability theory.

Chapter 2 describes algorithms for building graph-logic models (GL-models) for basic VBS, for the case of $m \geq 2$. The general approach of building models for such systems is shown.

Section 3 presents well-known methods for calculating the reliability of basic VBS (so-called k-out-of-n) systems.

Section 4 shows a modified algorithm for a system with a certain criterion, as well as examples.

In the conclusions, general conclusions on the work are made, the obtained results are analyzed.

Appendix 1 contains copies of graphic material. The attestation master's work is presented on 90 pages and consists of an introduction, 4 chapters, a conclusion and contains 6 figures, 15 tables, a list of used sources from 24 names.

Keywords: MULTIPROCESSOR FAILURE-TOLERANT SYSTEMS, RELIABILITY, GRAPH-LOG

Зміст

Перелік скорочень та термінів	3
ВСТУП.....	4
1 ХАРАКТЕРИСТИКИ НАДІЙНОСТІ БАГАТОПРОЦЕСОРНИХ СИСТЕМ.....	5
1.1 Відмовостійкість в багатопроцесорних системах	5
1.2 Поняття надійності системи.....	8
1.3 Показники надійності	12
Висновки до розділу 1	18
2. ПОВЕДІНКА ВІДМОВОСТІКИХ БАГАТОПРОЦЕСОРНИХ СИСТЕМ В ПОТОЦІ ВІДМОВ. ЇХ МОДЕЛІ.....	19
2.1 Традиційні ВБС моделі для підрахунку надійності	19
2.2 Графи під час розрахунку надійності.....	33
2.3 Графи на основі GL-моделей для підрахунку відмовостійкості ВБС	34
2.4 Базові $K(2, n)$ системи та GL-моделі для них.....	38
2.5 GL-моделі для небазових ВБС	42
2.6 Приклади	47
Висновки до розділу 2	57
3 ІСНУЮЧІ МЕТОДИ ПІДРАХУНКУ НАДІЙНОСТІ ДЛЯ БАЗОВИХ ВБС	58
3.1 Структурна функція k -out-of- n :G и k -out-of- n :F системи	58
3.2 Мінімальний набір шляхів і розділів.....	59
3.3 Барлоу і Хейдтманна	61
3.4 Метод Рушді.....	62
Висновки до розділу 3	63
4 МОДИФІКОВАНИЙ АЛГОРИТМ БУДУВАННЯ GL-МОДЕЛЕЙ ДЛЯ СКЛАДНИХ СИСТЕМ, ДЕ ПЕВНІ ПРОЦЕСОРИ МОЖУТЬ БРАТИ НА СЕБЕ РОБОТУ ІНШИХ.....	64
4.1 Алгоритм будування GL-моделі простої системи.....	64
4.2 Модифікація алгоритму для складних систем	66
Висновки до розділу 4	77

Висновки	78
-----------------------	-----------

СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ	79
---	-----------

ДОДАТКИ

Додаток А. Презентація магістерської дисертації

Додаток Б. Наукові публікації

Перелік скорочень та термінів

ВБС – відмовостійка багатопроцесорна система

ОС – обчислювальна система

ІОС – інформаційно-обчислювальна система

ТЗ – технічне завдання

GL-модель – графо-логічна модель

ВСТУП

В сучасному житті можна зустріти багато різних приладів і об'єктів які включають в себе складні багатопроцесорні системи. Наприклад супутники, літаки, банки, медичні системи та інше. Вихід з ладу кожної такої системи може призвести до важких наслідків, тому питання відмовостійкості є одним з головних під час їх проєктування. Через високу складність систем, моделювання роботи відмовостійких багатопроцесорних систем (ВБС) набуло великого значення. Можливість швидко та просто змоделювати та проаналізувати ці моделі є однією з головних задач нашого часу, бо від цього залежить можливість працездатності системи у потоці відмов.

Сучасні методи аналізу ВБС використовують доволі складні формули підрахунків, або ж алгоритми перебору станів системи, що займає багато часу та ресурсів. Є деякі алгоритми аналізу простих систем, такі як двополюсні граfi і моделі на базі марковських ланцюгів. Всі вище перелічені алгоритми не підходять для складних систем, так як зі збільшенням кількості процесорів їх аналіз занадто ускладнюється.

Одним з найбільш ефективних методів дослідження ВБС у потоці відмов на даний час є GL-моделі. Вони корисні тим, що дозволяють доволі швидко і ефективно проводити аналіз великих і складних систем. По своїй структурі GL-моделі це неорієнтовні граfi з булевими функціями на ребрах.

Основною метою даної роботи є покращення алгоритму будування GL-моделі для систем, де певні, процесори можуть брати на себе роботу інших, під час їх відмов, але не навпаки.

1 ХАРАКТЕРИСТИКИ НАДІЙНОСТІ БАГАТОПРОЦЕСОРНИХ СИСТЕМ.

1.1 Відмовостійкість в багатопроцесорних системах

Сучасні багатопроцесорні системи (як системи космічних і літальних апаратів, системи забезпечення життєдіяльності, системи транзакцій та ін.) повинні мати підвищенні характеристики до відмовостійкості через високу вартість їх помилки. Зазвичай, високі вомига до надійності реалізуються завдяки використанні різних видів елементів. Надлишковість в системі можна забезпечити таким резервуванням - інформаційним, структурним, функціональним, тимчасовим.

В теорії будування багатопроцесорних систем, застосовуються такі поняття: «відмовостійкість», «відмовостійка система» . Надлишковість у відмовостійких системах зазвичай використовується під час процесу визначення відмов та можливого усунення їх впливу на функціонування системи.

Зазвичай, під час моделювання відмовостійких цифрових приладів використовують різні методи дублювання. Ця задача, в останні роки, вирішується шляхом зміни структур і алгоритмів системи, мета яких - збереження її працездатності при виникненні відмов. В першу чергу це стосується, багатопроцесорних систем, це забезпечується введенням розвинутих засобів діагностики та реконфігурації.

Реконфігурація – це зміна складу та структури засобів та/або способу їх взаємодії між собою. Вона буває двох видів – статична та динамічна. Статична реконфігурація - це відключення із строю модулів системи. Динамічна реконфігурація - пряме розміщення, непряме розміщення , поступова деградація системи.

Після реконфігурації відбувається відновлення ВБС - спочатку на програмному рівні, потім на автоматному рівні.

Відмовостійкість системи – це властивість, яка дає можливість продовжувати виконання своїх функцій після виникнення відмови. Слід зауважити, що продуктивність системи може змінюватися, через необхідність виконання додаткових процедур.

Пошук відмов у відмовостійких система реалізується у таких напрямках:

1) вихідний і вхідний контроль (метод тестування) – здійснюється після

виготовлення системи і при її прийомі;

2) оперативний (функціональний) – дозволяє негайно відновити помилку в роботі.;

3) регламентний контроль – проводиться в певний час та по певному завданню;

4) контроль діагностичних засобів – здійснюється або оперативно, або з визначеними нормативами проміжками часу.

Є декілька основних принципів процедури відновлення системи після вдівови:

- розширювальне ядро;
- поєднання централізованого і децентралізованого управління процесами в системі;

- поєднання самовстановлення на системному рівні і на рівні модулів системи.

Одним з головних факторів, який треба враховувати при виборі алгоритму відновлення - це моделювання поведінки системи. За для цього використовуються численні моделі, можна також виділити структурні моделі, імітаційні моделі.

При проектування зазвичай використовують структурні моделі, через те, що такі моделі дозволяють оцінити такі параметри як продуктивності, вартості, надійності та пр. Одна з головних задач моделювання - вибір елементної бази, оптимізація структури, архітектури та процедури розробки системи.

Використання спеціальних моделей дозволяє виявити найменш надійні підсистеми та блоки елементів і прийняти рішення про те, які методи та способи рішення та організації технічного обслуговування потребується. Також система може аналізуватися з точки зору правильного використання різних видів захисту для вирішення завдань підвищення надійності. Під час моделювання системи аналізуються такі параметри як надійність, живучість, відмовостійкість та відновлюваності. Мета цього проектування буває різною –виявлення логічних помилок при проектуванні, або розробленого для системи програмного забезпечення, та аналіз процедур відновлення в процесі експлуатації.

Добре відомі та широко використовуються моделі, описуючі діагностику та контроль в ВБС. Найбільш формовані моделі, умовно звані Р-, В-, R- і К-моделями, що відрізняються між собою за допомогою результатів тестування модулями системи

іншого модуля та урахування структури об'єкта діагностики.

Можна навести декілька методів визначення станів елементів, які використовуються в ВБС: пошук одного непрацездатного елемента з наслідком його заміни або в результаті функціонування системи та продовження пошуку; пошук всіх непрацездатних елементів системи паралельно.

Є дві методи – послідовне діагностування з відновленням, та – одношагове без відновлення.

Більш ефективним є третій метод, початок якого це пошук, в крайній мірі одного працюючого процесора. Далі цей процесор вважається ядром, а інші справні процесори розширюють це ядро. Результати тестування процесорів ядра інших процесорів однозначно вважаються правильними.

Проектування ВБС приходить в декілька етапів: перший це розробка архітектура майбутньої системи до завдань, які майбутня система повинна вирішити.

Що до системи управління складними об'єктами, то щоб забезпечити необхідний рівень надійності, як і оцінити характеристик надійності оброблюваної системи представляє серйозні труднощі, оскільки система може складатися з декількох різних підсистем, будь-яка з них може мати свою кількість процесорів з різними ймовірностями виходу з ладу, свого ступінь відмовостійкості і інше. Якщо розрахована характеристика надійності виявляється нижче тієї, яка визначена в технічному завданні, належить змінити архітектуру системи, вводячи її в невиконавчі процесори, знову і знову перерахувати надійність до тих пор, поки це не буде досягнуто.

Ціль введення зайвих процесорів є забезпеченням певного рівня відмовостійкості, тож є відмова від певних підмножин процесорів, що має призвести до підвищення надійності системи. Зазвичай, це вимагає виконання певних умов, а саме:

- У створюваній системі повинна бути можливість введення додаткових модулів;
- Повинна бути вирішена задача самотестування ВБС використовуючи один із вище наведених методів і з урахуванням особливостей архітектури системи;
- Повинно бути спеціальне програмне забезпечення, яке буде тестувати систему та програмне забезпечення реконфігурації системи, щоб забезпечити відключення

несправних елементів і передачу завдань, які вони вирішували, іншим процесорам, які в момент самотестування були в справному стані;

- Повинне бути спеціальне програмне забезпечення, яке забезпечить відновлення втраченої інформації через відмови і повернення до якого-то рівня нормальної роботи системи або її безпечного залишку.

Таким чином, оптимізація архітектури системи і оптимальний вибір параметрів її ресурсів знаходиться в прямій залежності від оцінки надійності ВБС на ранніх етапах її проектування.

Це ще один раз підтверджує важливість попередньої оцінки і в загальному випадку розрахунку надійних параметрів ВБС. Одним із шляхів вирішення цієї задачі, є створення спеціальних математичних моделей, що адекватно відображають поведінку системи в потоці відмов.

Створення подібних математичних моделей для ВБС –задача, без рішення якої неможливо виконати експерименти. З іншої сторони необхідна певна кількісна оцінка різних надійних параметрів елементів і підсистем. Такий кількісний аналіз треба, щоб забезпечити певний комплекс заходів по забезпеченню високої надійності ВБС.

1.2 Поняття надійності системи

На даний час існує багато складних процесів і модулів, які складаються з багатьох компонентів. Надійність це одна з головних характеристик таких систем. Дослідження поведінки ВБС в потоці відмов дозволяю дослідити та оцінити наскільки системи може працювати загалом.

Надійність є основною проблемою інженерних виробів. Проблема підрахунку надійності з'явилась з самими першими приладами. Ненадійні вироби ніколи нікому не були потрібні. Зрозуміло, що ненадійними приладами ніхто не буде користуватися. Доволі часто до різних приладів та систем можна знайти інструкцію з різними коефіцієнтами запасу.

Надійність, як властивість, залежить від багатьох факторів, таких як - безвідмовності, ремонтпридатності, довговічності і зберігання. Кожна система також

має відповідати певному ГОСТ-у.

За останні декілька десятиків років поняття надійної системи зазнало багато змін. До неї додалося багато різних факторів, це зумовлено складністю сучасних систем та наступними причинами:

1. Сучасні прилади включають в себе $\sim 10^5$ елементів, що набагато ускладнює їх;
2. Інтенсивність роботи елементів системи при різних умовах: різні температури, тиск, різні швидкості;
3. Підвищенні вимоги якості роботи системи, такі як точність, швидкість, ефективність та інші;
4. Максимальна автоматизація. В сучасних системах майже все автоматизовано, що максимально знижує участь людини у будь яких взаємодіях з нею.

В теорії надійності є певні поняття, без яких не можливо зрозуміти та порахувати її:

Працездатність – можливість системи працювати протягом певного часу.

Напрацювання – обсяг роботи об'єкта.

Відмова – порушення працездатності об'єкта. Може бути викликана різними факторами, як фізичне руйнування, старість, дефект у проектуванні.

Збій – короткочасна відмова.

Сучасні умови експлуатації та інтенсивність роботи систем можуть характеризуватися високими температурами (камери двигунів, робота процесора при великій загрузці), високими швидкостями та інше. Зовнішні фактор також мають великий вплив на працездатність системи. Сучасні прилади можуть використовуватися в вакуумі, з високою вологістю, під час вібрацій та з наявністю радіації. Це все може впливати на виникнення відмови в системі.

Підрахунок надійності в нас час є одним з найвідповідальніших етапів розробки системи, так як її відмови може призвести до великих втрат, як технічних так і економічних: наприклад відмови всього лише одного елемент вартість менше 10

доларів викликала невдачу запуску супутника, вартістю більше $5 * 10^6$ доларів.

Розрахунок надійності зазвичай проводиться після поділу системи на елементи. Кожний елемент система має виконувати свою задачу і мати свій показник надійності.

Кожну систему можна поділити на відновлювану та невідновлювану. Відновлювані системи можуть відновити свою працездатність після відмови певних елементів. Невідновлювані системи – це ті системи, які після відмови відновленню не підлягають.

Для обох видів можна ввести певні поняття:

- напрацювання – тривалість роботи системи в певних умовах;
- технічний ресурс – сумарне напрацювання системи за певний період;
- термін служби – календарна тривалість експлуатації системи до відмови;
- гарантований термін служби – термін служби, протягом якого виробник гарантує справність системи.

Рішення для забезпечення надійності на кожному етапі розробки дає значну економію порівняно з тим, коли технічні рішення треба переглядати на етапах технічного і робочого проектування чи навіть під час експлуатації. Винятковим, для забезпечення надійності, є правильне застосування методів її оцінки та обґрунтування вибору найкращих рішень з точки зору надійності.

На етапі ескізів надійності вимірюється для створення таблиці прогнозованих показників надійності. Це необхідно, щоб показати, що в проект і пропонованої системи показники надійності будуть вище необхідних.

Під час технічного проектування розрахунок надійності використовуються для вибору технічних засобів, що входять в систему, а також для вибору способів контролю, діагностики, створення структури системи, вимог комплектуючих елементів і програмного забезпечення. Надійності на цьому етапі має бути розрахована

багато разів, бо зміна в архітектурі проекти може мінятися доволі часто.

На етапі випробувань проводиться тестування та перевіряється, чи надійність системи відповідає необхідним вимогам. Розрахунки надійності на цьому етапі ускладняється тим, що з для великих системи не може бути великої кількості екземплярів. Доволі часто такі системи пред'являється в дуже скороченому варіанті і треба визначити надійність на підставі обмеженої інформації. Тільки за допомогою поєднання випробувань системи з розрахунками, заснованими на всій наявній інформації про її надійність, можна вирішити задачу оцінки надійності великої системи на етапі випробувань.

На етапі експлуатації надійність використовуються для вибору складу і об'єму запасних виробів, які треба для заміни відмовивших, а також для вимірювання та планування необхідного профілактичного обслуговування.

На етапі ескізного проектування треба вибрати елементну базу, визначати особливості структури та архітектури системи. З цими даними треба провести попередній розрахунок надійності, виявити найменш надійні елементи чи підсистеми і прийняти про резервування відповідних елементів. Досліджується питання про способи автоматичного відновлення системи.

Дуже важливим для забезпечення надійності є виявлення та виправлення помилок у технічній документації. Контроль та забезпечення умов навколишнього середовища є одними з найважливіших факторів на етапі експлуатації, що забезпечують надійну роботу системи. Також достатня кваліфікація та склад персоналу, який обслуговує системи, є важливими факторами її працездатності.

Для прогнозування надійності в ході експлуатації існують спеціальні математичні моделі, які засновані на різних припущеннях про прояву помилок програми.

Існує декілька способів підвищення та забезпечення надійності програм, які удосконалюються програмним шляхом; вибір алгоритмів, які не чутливі до різних порушень обчислювання; контроль і тестування програм з їх корекцією.

1.3 Показники надійності

Є декілька імовірних показників надійності, які базуються на таких величинах як час до відмови, час між відмовами, а також час відновлення. Зазвичай використовуються різні показники надійності, щоб характеризувати таку систему з різних сторін.

Однією з головних якісних характеристик надійності є функція надійності. За її визначенням це ймовірність безвідмовної роботи системи протягом певного часу або ймовірності того, що при певних умовах, у межах певного часу системи відмови не виникне:

$$P(t) = P(T > t) \quad (1.3.1)$$

де T – час безвідмовної роботи системи,

t – заданий час

$P(A)$ – ймовірність події A .

Доповненням до цієї функції є функція ненадійності системи, яка має вигляд:

$$Q(t) = 1 - P(t) \quad (1.3.2)$$

Її ще скорочено називають ненадійністю – ймовірність того, що за певний проміжок часу система вийде з ладу. Тому функція $Q(t)$ збігається з функцією розподілу часу до відмови.

Показники мають наступні властивості, які очевидні з їх визначень:

1. $P(0) = 1, Q(0) = 0$ (на початку роботи система є повністю працездатною);
2. (час роботи системи скінченний);
3. $\frac{dP(t)}{dt} \leq 0, \lim_{t \rightarrow \infty} Q(t) = 1$ (ості є спадною функцією);
4. $0 \leq P(t) \leq 1, 0 \leq Q(t) \leq 1$;

5. $P(t_2) \leq P(t_1), Q(t_2) \geq Q(t_1)$, якщо $t_2 \geq t_1$.

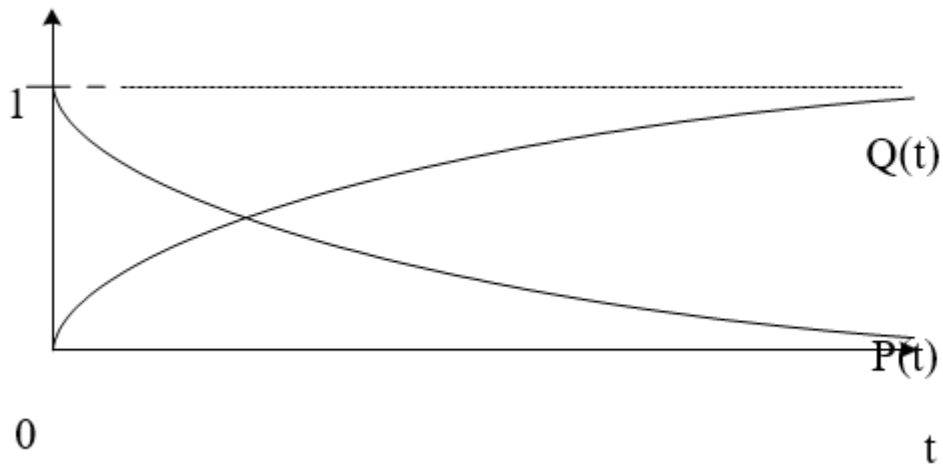


Рисунок 1.3.1 – Ймовірність безвідмовної роботи

6. $P(t) + Q(t) = 1$ (слідкує безпосередньо з визначень надійності та ненадійності).

Щільністю розподілу часу безвідмовної роботи називають похідну від функції ненадійності:

$$f(t) = \frac{dQ}{dt} = -\frac{dP}{dt}. \quad (1.3.3)$$

Функцію $P(t)$ незручно використовувати в якості показника надійності через її функціональну залежність, Тому в технічних умовах зазвичай пишуть окремі значення функції $P(t)$ при значеннях t , обраних з нормованого ряду $t = 100; 500; 1000; 2000; 5000$;

В статистиці значення $P(t)$ можна порахувати лише наближено, у вигляді статистичної оцінки:

$$\hat{P}(t) = \frac{N(t)}{N_0}, \quad (1.3.4)$$

де $N(t)$ – кількість працюючих до моменту часу t об'єктів при початковій кількості N_0 .

Вигляд цієї функції змінюється, через те, що точки розриву, які відповідають відмовам, змінюються з експериментами. Середньоквадратична похибка оцінки (1.3.4) визначається виразом теореми Муавра – Лапласа

$$\sigma_{\hat{P}} \approx \sqrt{\frac{P(1-P)}{N_0}},$$

Функція $\lambda(t)$ виражає інтенсивність виникнення відмов за одиницю часу, і є однією з зручних характеристик надійності через її розрахунок.

Мірою інтенсивності відмов слугує відношення 1.3.5, через те, що чим більше випробувань було зроблено, тим менше ймовірність відмови

$$\lambda(t) = \frac{f(t)}{P(t)} = \frac{\left(\frac{-dP(t)}{dt} \right)}{P(t)} \quad (1.3.5)$$

Зобразивши відношення (1.3.5) як диференціальне рівняння і вирішивши, можна отримати наступну формулу:

$$P(t) = e^{-\int_0^t \lambda(t) dt}. \quad (1.3.6)$$

Статистично інтенсивність потоку відмов оцінюється за відношенням

$$\lambda_c = \frac{N(t, t+\Delta t)}{N(t)\Delta t}, \quad (1.3.7)$$

як число відмов $N(t, t+\Delta t)$ в достатньо малому інтервалі часу, віднесена до числа

тих, що залишилися до моменту часу t зразків $N(t)$.

Крім розглянутих вище показників надійності, які є функціональними показниками, використовують ще числові показники. Одним з найважливішим серед них є час безвідмовної роботи T_0 . Його називають середнім напрацюванням до відмови і просто напрацюванням до відмови. Воно визначається як математичне сподівання величини T - час безвідмовної роботи.

$$T_0 = M(T) = \int_0^{\infty} t f(t) dt = - \int_0^{\infty} t \frac{dP(t)}{dt} dt = - \int_0^{\infty} t dP(t) = - (tP(t)) \Big|_0^{\infty} + \int_0^{\infty} P(t) dt = \int_0^{\infty} P(t) dt$$

Цей природний показник нічого не говорить про характер розподілу часу до відмови. Як приклад, дві, не пов'язані функції $P_1(t)$ і $P_2(t)$, що виражають дві різні ймовірності при малому напрацювання ($t \ll T_0$), можуть мати однакове значення T_0 .

Гамма t_γ – відсоткове напрацювання, яке є гарантованим технічним ресурсом і яким володіють не менше ніж γ % систем, де γ – гарантована ймовірність безвідмовної роботи. Вона виражає значення функції, оберненої до функції надійності – $P^{-1}(\gamma)$ при заданому значенні аргументу:

$$t_\gamma = P^{-1}(\gamma), \quad (1.3.8)$$

де γ – деяка гарантована ймовірність.

Можна сказати гамма - процентне напрацювання t_γ – час безвідмовної роботи системи із заданою ймовірністю γ .

Усі показники надійності придатні для обох видів систем – невідновлюваних та відновлюваних. Однак останні вони характеризують тільки до появи першої відмови.

Але цих характеристик недостатньо опису роботи такої відновлювальної системи, так як вона після відмови ремонтується і її працездатність відновлюється.

Для цього додаються нові показники які додатково описують процес відновлювання системи.

Функція ремонту – ймовірність, що час відновлення системи T_g

буде меншим за заданий час t , тобто

$$R(t) = P\{T_g \leq t\}, \quad (1.3.9)$$

де $R(t)$ – випадкова величина.

$R(t)$ є функцією розподілу часу відновлення, вона – аналог функції ненадійності.

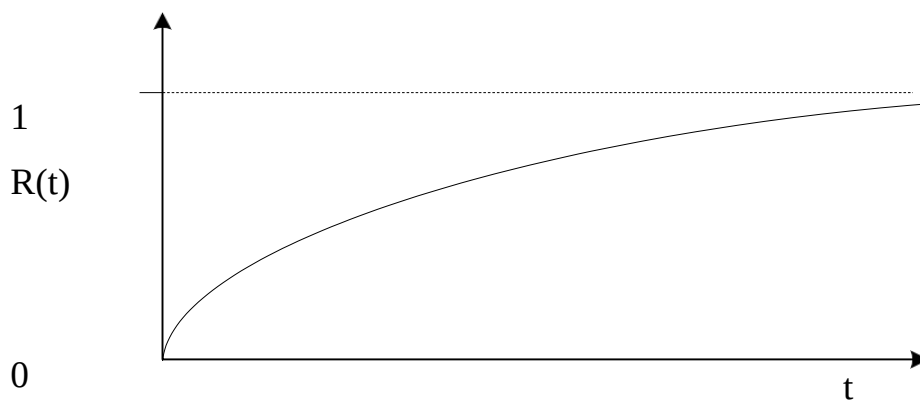


Рисунок 1.3.2 – Функція ремонту

Для відновлювання системи вводяться поняття щільності часу відновлення та інтенсивність ремонту - $r(t)$ та $\mu(t)$.

$$r(t) = \frac{dR(t)}{dt} \quad (1.3.10)$$

$$\mu(t) = \frac{r(t)}{1 - R(t)} \quad (1.3.11)$$

Статистично інтенсивність ремонту визначається як

$$\mu_c = \frac{N'_1(t + \Delta t) - N'_1(t)}{N'_1(t) \cdot \Delta t}, \quad (1.3.12)$$

де $N'_1(t)$ – число відновлених до моменту часу t об'єктів.

Використовуючи (1.3.10) і (1.3.11) за аналогією можна записати наступну формулу:

$$R(t) = 1 - e^{-\int_0^t \mu(t) dt} \quad (1.3.13)$$

Функцію готовності $G(t)$ використовують для характеристики працездатності відновлюваної системи в певний момент часу. За визначенням це ймовірність того, що в момент часу t система справна.

Вона може бути отримана з інших показників надійності, наприклад, з функції ремонту $R(t)$ і ймовірності безвідмовної роботи $P(t)$.

$$T_u = \int_0^T G(t) dt \quad (1.3.14)$$

За наступною формулою можна отримати параметр потоку відмов $\omega(t)$, який показують питому кількість відмов за одиницю часу:

$$\hat{\omega}(t) = \frac{N(t, t + \Delta t)}{N_0 \Delta t}.$$

Коефіцієнт готовності K – це ймовірність того, що в певний момент часу t система може працювати (крім періодів, протягом яких застосування системи за призначенням не передбачається). Цей показник використовується, якщо треба враховувати час відновлення.

$$K_z = \frac{N_{\varepsilon}(t)}{N_0}, \quad (1.3.15)$$

де $N_{\varepsilon}(t)$ – число об'єктів, що перебувають у робочому стані в момент часу t .

Різниця $N_{\varepsilon} - N$ це кількість об'єктів, що в момент часу t ремонтуються.

Висновки до розділу 1

У цьому розділі було показано базові характеристики надійності система. Досліджено поняття відмовостійкої багатопроцесорної системи та показані основні терміни з теорії надійності.

2. ПОВЕДІНКА ВІДМОВОСТІКИХ БАГАТОПРОЦЕССОРНИХ СИСТЕМ В ПОТОЦІ ВІДМОВ. ЇХ МОДЕЛІ

2.1 Традиційні ВБС моделі для підрахунку надійності

Доволі відома моделей, яка використовується від час розрахунку показників надійності системи, це двополюсна модель. Її суть полягає в тому, що розрахунок параметрів надійності замінюється на задачу визначення зв'язності двополюсних графів, у яких їх дві вершини виділені особливим чином і називаються вхідним і вихідним полюсами. Відмовою системи вважається повна втрата здатності виконувати свої функції, зазначені в документації, або велике відхилення вихідних значень від зазначених у паспорті параметрів. Вище сказане інтерпретується як можливість проходження по графу від вхідної вершини-полюса до вихідної. При створенні математичної моделі технічної системи критерій відмови визначається як стан елементів, складових дану систему. Вважається, що кожен із елементів системи протікає з точки зору надійності - він може знаходитися лише в двох станах: працездатності та відмови.

Стан схеми визначається завдяки сукупності станів її елементів.

В якості ребер двополюсного графа виступають модулі системи. Очевидно, що структура системи не складається з топологією двохполюсного графа.

Поточний стан i -го елемента описується завдяки спеціальній функції $x_i(t)$:

$$x_i(t) = \begin{cases} 1, \\ 0, \end{cases}$$

Якщо розглядати стан системи в якийсь певний момент часу, то в цьому випадку замість індикаторної функції використовують звичайні булеві змінні x_i .

Стан системи з n елементів можна описано вектором, так званим вектором стану системи

$$X=(x_1, x_2, \dots, x_n)$$

Усього можливо 2^n різних складових систем. В залежності від структури системи це все розділяють на два підмножини, які відповідають стану коли система працездатна і стану відмови. Можна сказати, що на множині векторів, з яких складається система $\{X\}$, з'являється нова функція, що отримала назву логічної структурної функції:

$$F(X)=\begin{cases} 1, & \text{якщо система працездатна;} \\ 0, & \text{в іншому випадку.} \end{cases}$$

Очевидними є дві властивості структурної функції: якщо всі елементи працездатні, то $F(X)=1$, а якщо всі елементи в стані відмови, то $F(X)=0$. Для всіх інших систем, які можуть бути представлені цими векторами, необхідно рахувати значення функції $F(X)$ використовуючи сформований двополюсний граф.

Можна відзначити, що після отримання результатів для структурної функції, розрахунок надійності системи можна змінити з логічних функцій, які залежать від змінних, до ймовірностей цих логічних функцій (для випадку двополюсних графів – до надійності системи), що завищують ймовірність від їх змінних:

Таблиця 2.1.1

№	Логічний вираз	Математичний вираз
1	$f = x_1 \wedge x_2$	$p(f) = p_1 \cdot p_2$
2	$f = x_1 \vee x_2$	а) Якщо події сумісні, то $p(f) = p_1 + p_2 - p_1 \cdot p_2$ б) Якщо події не сумісні, то: $p(f) = p_1 + p_2$
3	$f = \bar{x}$	$p(f) = 1 - p$

Через ці три вирази можна виразити всі інші:

$$f = x_1 \cdot x_2 \cdot x_3 \cdot \dots \cdot x_n = \bigwedge_{i=1}^n x_i \Rightarrow p(f) = \prod_{i=1}^n p_i$$

$$f = x_1 \vee x_2 \vee x_3 \vee \dots \vee x_n = \bigvee_{i=1}^n x_i = \overline{\bigwedge_{i=1}^n \bar{x}_i} \Rightarrow p(f) = 1 - \prod_{i=1}^n (1 - p_i)$$

Є декілька типів базових двополюсних графів:

1. Послідовні з'єднання.

Таким графом інтерпретується будь-яка система, яка не має резервних елементів і відмов, хоча один її елемент припинився у всій системі (рис. 2.1.1).



Рисунок 2.1.1 - Послідовне з'єднання

На кожному ребрі графа задана індикаторна булева функція, що описує стан певного елемента, тому на дугах намальовані прямокутники.

Логічна структурна функція виглядає так:

$$F(X) = \bigwedge_{1 \leq i \leq n} x_i$$

Надійність такої системи дорівнює:

$$P(F(X)) = \prod_{1 \leq i \leq n} p_i$$

Використовуючи $p = e^{-\lambda_i t}$, отримаємо:

$$P_{\Sigma} = \prod_{i=1}^n e^{-\lambda_i t} = e^{-\lambda_1 t} \cdot e^{-\lambda_2 t} \cdot \dots \cdot e^{-\lambda_n t} = e^{-(\lambda_1 + \lambda_2 + \dots + \lambda_n)t} = e^{-\lambda_{\Sigma} t},$$

$$\text{где } \lambda_{\Sigma} = \lambda_1 + \lambda_2 + \dots + \lambda_n = \sum_{i=1}^n \lambda_i$$

Середній час безвідмовної роботи:

$$T_0 = \frac{1}{\lambda_{\Sigma}} = \frac{1}{\sum_{i=1}^n \lambda_i}$$

Якщо всі модулі мають те саме значення λ , то:

$$P_{\Sigma} = P^n = e^{-n\lambda t} \quad \text{і} \quad T_0 = \frac{1}{n\lambda}$$

Якщо кількість елементів такої системи зростає, та її надійність падає, тобто $n \rightarrow \infty$, то $P_{\Sigma} \rightarrow 0$.

2. Паралельне з'єднання.

Даний граф показує систему, яка буде працездатною коли хоча б один з її елементів буде працювати .

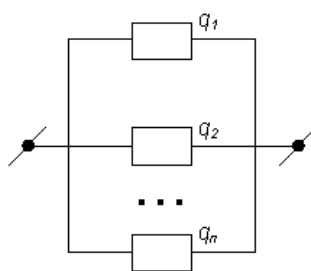


Рисунок 2.1.2 - Паралельне з'єднання

Логічна структурна функція такої системи має вигляд:

$$F(X) = x_1 \vee x_2 \vee x_3 \vee \dots \vee x_n = \bigvee_{i=1}^n \bar{x}_i$$

Надійність такої системи дорівнює:

$$P(F(X)) = 1 - \prod_{i=1}^n (1 - p_i)$$

Зауважимо, що надійність такої системи вже не дотримується експоненційного закону. Справді, якщо надійності елементів у вигляді $P_1 = e^{-\lambda_1 t}$, ..., $P_n = e^{-\lambda_n t}$, то вираз для надійності всієї системи набуде вигляду:

$$P_{\Sigma} = 1 - (1 - e^{-\lambda_1 t}) \cdot (1 - e^{-\lambda_2 t}) \cdot \dots \cdot (1 - e^{-\lambda_n t})$$

При збільшенні кількості елементів надійність системи зростає, тобто якщо $n \rightarrow \infty$, то $P_{\Sigma} \rightarrow 1$.

3. Структури з послідовними та паралельними сполуками.

а) Структура, що складається з m ланцюжків по n елементів, причому ланцюжки з'єднані паралельно:

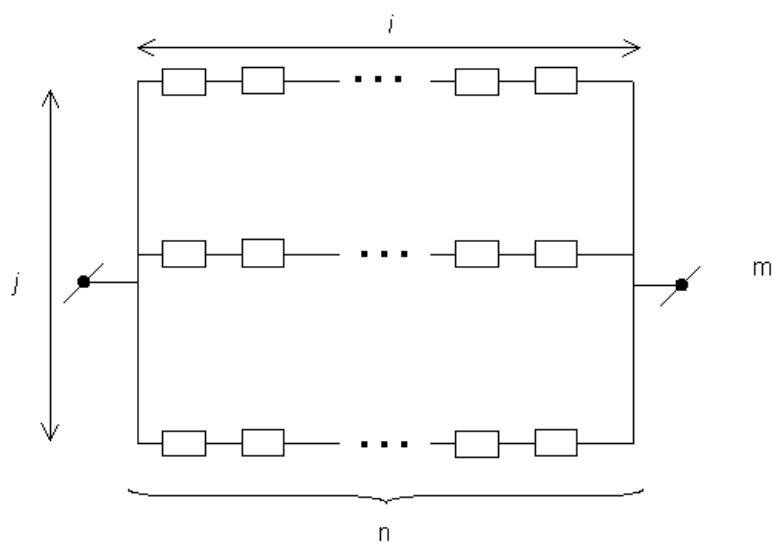


Рисунок 2.1.3 - Структура, що складається з m ланцюжків по n елементів, причому ланцюжки з'єднані паралельно

Структурна функція такої системи має вигляд:

$$F = \bigvee_{j=1}^m \left(\bigwedge_{i=1}^n x_i \right)_j = \overline{\bigwedge_{j=1}^m \left(\bigwedge_{i=1}^n \overline{x_i} \right)_j}$$

Якщо позначити p_{ij} ймовірність безвідмовної роботи x_{ij} -го елемента, то надійність такої системи дорівнюватиме:

$$P(F) = 1 - \prod_{j=1}^m \left(1 - \prod_{i=1}^n p_i \right)_j$$

Поведінка такої системи:

- якщо $m \rightarrow \infty$, то $P(F) \rightarrow 1$, тобто. збільшення числа паралельних ланцюжків збільшує надійність;
- якщо $n \rightarrow \infty$, то $P(F) \rightarrow 0$;
- якщо $m \rightarrow \infty$ и $n \rightarrow \infty$, то $P(F) \rightarrow 0$.

б) Структура, що складається з n груп, з'єднаних послідовно. У кожній групі m паралельно поєднаних елементів :

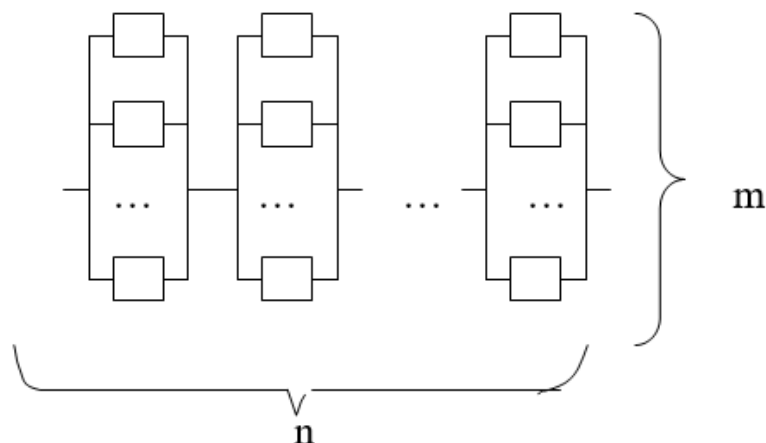


Рисунок 2.1.4 - Структура, що складається з n груп, з'єднаних послідовно. У кожній групі m паралельно поєднаних елементів

Структурна функція такої системи має вигляд:

$$F = \bigwedge_{j=1}^n \left(\bigvee_{i=1}^m x_i \right)_j = \bigwedge_{j=1}^n \left(\bigwedge_{i=1}^m \bar{x}_i \right)_j$$

Якщо позначити p_{ij} ймовірність безвідмовної роботи x_{ij} -го елемента, то надійність такої системи дорівнюватиме:

$$P(F) = \prod_{j=1}^n \left(1 - \prod_{i=1}^m (1 - p_i)_j \right)$$

Для випадку, коли всі блоки однакові з точки зору їхньої ймовірності безвідмовної роботи за заданий час, та їх надійність позначається p , то

$$P(F) = 1 - (1 - p^n)^m$$

Поведінка такої системи:

- якщо $m \rightarrow \infty$, то $P(F) \rightarrow 1$, тобто. і в цьому випадку збільшення числа паралельних елементів збільшує надійність;
- якщо $n \rightarrow \infty$, то $P(F) \rightarrow 0$;
- якщо $m \rightarrow \infty$ и $n \rightarrow \infty$, то $P(F) \rightarrow 1$ – ми отримали теоретичну можливість побудови нескінченно надійних систем.

Розрахунок ймовірності безвідмовної роботи системи, який використовує граfi з паралельно-послідовною структурою, зручно рахувати шляхом поступового спрощення його структури, як показано на прикладі.

Вихідна схема має вигляд:

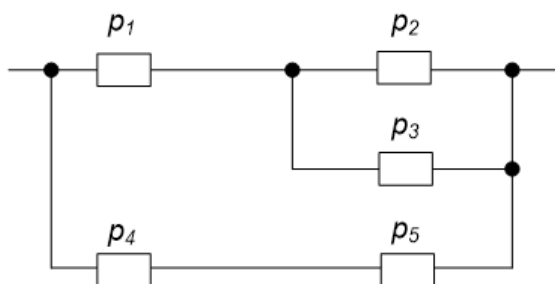


Рисунок 2.1.5 – Схема для приклада

Після заміни паралельного з'єднання другого та третього ребер об'єднаним ребром, еквівалентним з погляду надійності має вигляд:

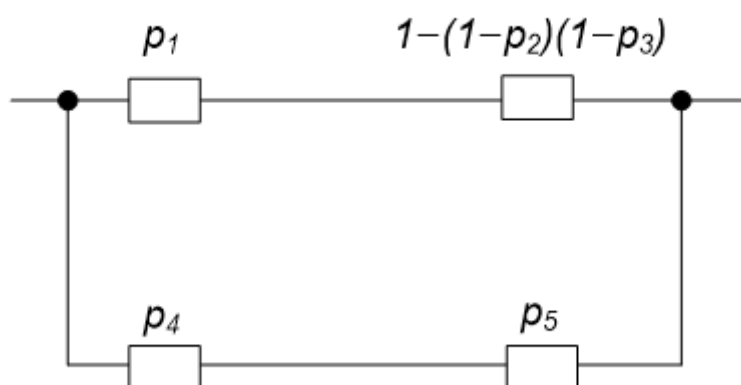


Рисунок 2.1.6 – Схема після першого перетворення

Замінивши дві послідовні сполуки, отримаємо наступну схему:

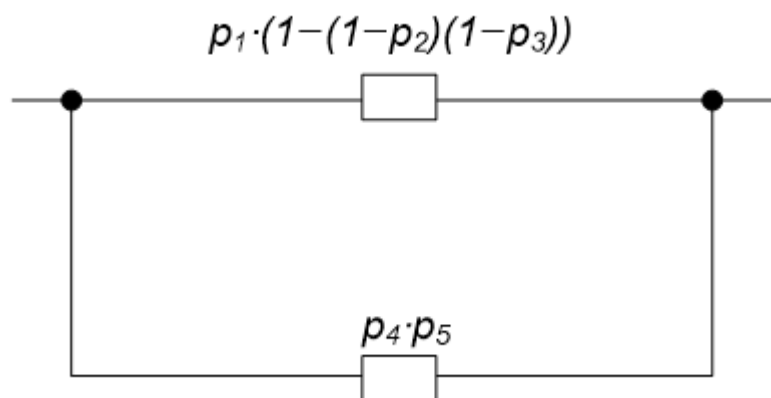


Рисунок 2.1.7 – Схема після другого перетворення

Після останнього еквівалентного з погляду надійності перетворення отримуємо найпростіший двополюсний граф, який зручний для подальшого використання:

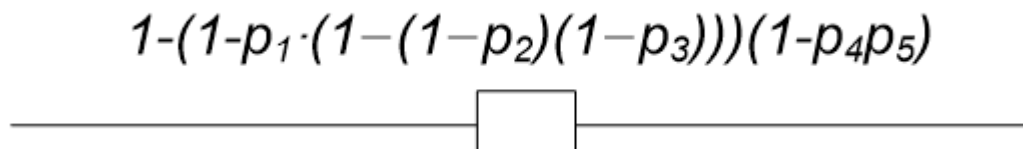


Рисунок 2.1.8 - Схема після останнього перетворення

Також ще існують структури, які неможливо привести, до них можна віднести всі структури, які не можуть бути спрощені за допомогою еквівалентних перетворень. Кількість структур, що не зводяться, зростає не лінійно зі зростанням кількості ребер у графі. Найпростішою з незведених структур, єдиною непривідною для двополюсних графів, що складаються з п'яти ребер, є так звана місткова схема, яка має вигляд:

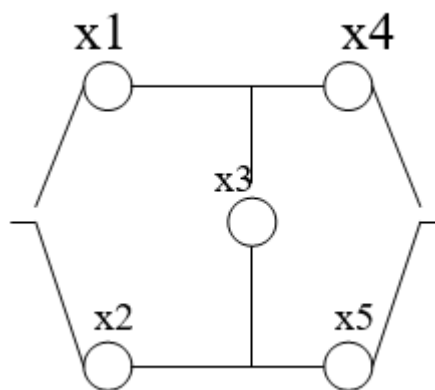


Рисунок 2.1.9 - місткова схема

Недолік таких структур у багатьох випадках є великою перешкодою для проведення точного розрахунку надійності. Складність створення логічних структурних функцій, що описують умову наявності шляху між входом і виходом двополюсного графа. Зазвичай, у цих випадках, використовують один з трьох методів, що дозволяють або суттєво спростити розрахунок, або просто перевести його в розряд конструктивних завдань.

Якщо система має обмежену кількість модулів то можна скласти таблицю істинності логічної структурної функції. Зокрема, для наведеної вище місткової схеми, скорочена таблиця істинності виглядатиме таким чином:

x_1	x_2	x_3	x_4	x_5	$F(X)$
0	0	0	0	0	0
0	0	0	0	1	0
0	0	0	1	0	0
1	•	•	•	•	•
1	0	0	1	0	1
1	•	•	•	•	•
1	0	1	0	1	1
1	•	•	•	•	•
1	1	1	1	1	1

Рисунок 2.1.9 – Таблиця для місткової схеми

Функція $F(X)$, записана за вище наданою таблицею, міститиме всі конституенти одиниці, які дозволять швидко та ефективно записати вираз для визначення надійності системи, уникнувши при цьому навіть перетворень за теоремою де Моргана .

Для місткової схеми при рівній ймовірності безвідмовної роботи всіх її елементів, вираз для розрахунку надійності матиме такий вигляд:

$$P(F) = 2p^5 - 5p^4 + 2p^3 + 2p^2$$

Друга методика розкладає складну логічну структурну функцію на низку більш простих функцій.

Зокрема, якщо застосувати розкладання Шеннона для наведеної вище місткової схеми (проведемо розкладання по x_3), то вигляд у логічної структурної функції буде такий:

$$F(X) = x_3 F(X|_{x_3=1}) \vee \bar{x}_3 F(X|_{x_3=0})$$

Структура двополюсного графа такої схеми поділиться на два варіанти, що відповідають розкладу Шеннона по ребру x_3 . У першому варіанті який відповідає умові $x_3 = 1$, відбувається стягування ребра x_3 , воно існує завжди:

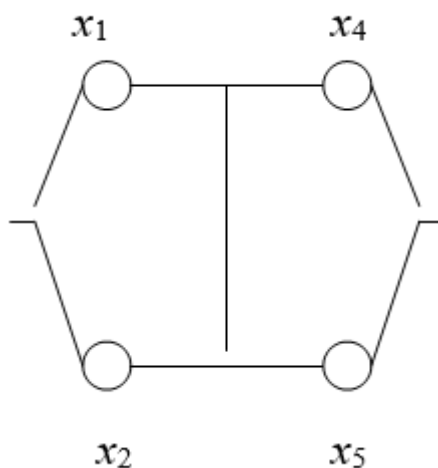


Рисунок 2.1.10 – Схема, після відбувається стягування ребра x_3

Для цього варіанта логічна структурна функція набуде наступною:

$$F(X|_{x_3=1}) = (x_1 \vee x_2)(x_4 \vee x_5)$$

Другий варіант (рис. 2.1.11), що відповідає умові $x_3 = 0$, передбачає «видалення» ребра x_3 :

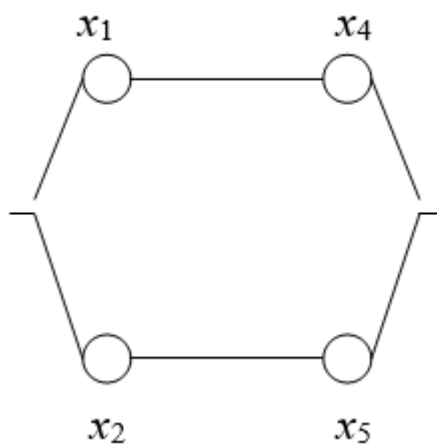


Рисунок 2.1.11 – Схема, після «видалення» ребра x_3

Для цього варіанта логічна структурна функція буде виглядати так:

$$F(X|_{x_3=0}) = x_1x_4 \vee x_2x_5$$

Об'єднавши обидва варіанти у формулі Шеннона і перейшовши від логічної структурної функції схеми до виразу для розрахунку надійності, отримаємо:

$$P(F(X)) = p_3 (1 - q_1q_2)(1 - q_4q_5) + q_3(1 - (1 - p_1p_4)(1 - p_2p_5))$$

де під q_i розуміється ймовірність відмови i -го модуля системи.

У графах з більш складною структурою, що не зводяться, розкладання може бути зроблено не один раз, проте тоді зростає складність функції, що може призвести до того, що розрахунок надійності таким шляхом втратить сенс через дуже високу обчислювальну складність.

Третя методика має прямі аналоги у природі і може бути охарактеризована як еквівалентне перетворення зірки на трикутник і трикутника на зірку.

Якщо застосувати цю методику до місткової схеми, можна помітити, що її ребра, позначені x_1 , x_2 і x_3 , утворюють трикутник, еквівалентна заміна якого призведе до отримання досить простої з точки зору розрахунку надійності схемою з послідовними та паралельними сполуками (рис. 2.1.12), на якій невідомі допоміжні ребра позначені як a , b та c .

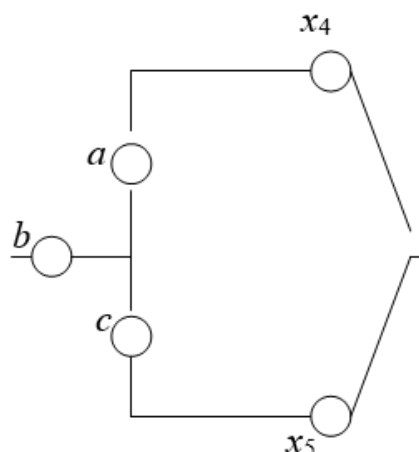


Рисунок 2.1.12 - Схемою з послідовними та паралельними сполуками

Щоб можна було перейти від трикутника до зірки та правильно визначити параметри нових ребер a , b та c треба скласти рівняння працездатності для пар вузлів,

позначених на рисунках 2.1.13 та 2.1.14 цифрами 2 та 3.

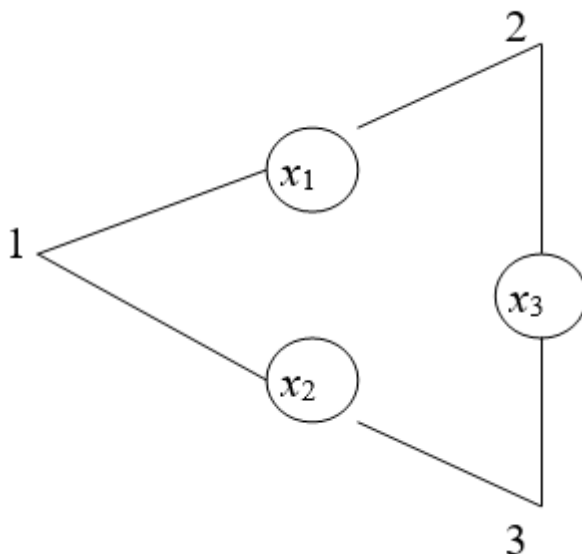


Рисунок 2.1.13 – Схема ребр для переходу від трикутника до зірки

Логічні рівняння, що утворюють систему, будуть мати такий вигляд:

- для вузлів $\langle 1;2 \rangle$: $x_1 \vee x_2x_3 = ab$
- для вузлів $\langle 1;3 \rangle$: $x_2 \vee x_1x_3 = bc$
- для вузлів $\langle 2; 3 \rangle$: $x_3 \vee x_1x_2 = ac$

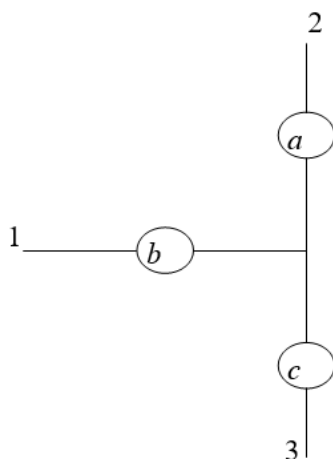


Рисунок 2.1.14 – Схема заміни ребр

Побудуємо системи рівнянь, яка дозволяє виразити невідомі значення ймовірностей елементів, отриманих при перетворенні, через ймовірності модулів системи, ці рівняння будуються на основі наведеної вище системи рівнянь

працездатності для вузлів графа і виглядають так:

$$\left. \begin{aligned} p(x_1) + p(x_2)p(x_3) - p(x_1)p(x_2)p(x_3) &= p_a p_b \\ p(x_2) + p(x_1)p(x_3) - p(x_1)p(x_2)p(x_3) &= p_c p_b \\ p(x_3) + p(x_2)p(x_1) - p(x_1)p(x_2)p(x_3) &= p_a p_c \end{aligned} \right\}$$

Після розв'язання системи рівнянь отримані значення невідомих p використовуються у приведенні графа з послідовними і паралельними сполуками ребер, після чого розрахунок надійності для випадку місткової структури більше не становить труднощів.

Для випадків, коли через складність і розмірність графа не можливо його привести, а ще й надзвичайно складно та неефективно, стане корисним метод, заснований на аналізі мінімальних шляхів та мінімальних перерізів графа, що дозволяє оцінити надійність без отримання її точного значення.

Використовуючи цей метод можливе отримання двох оцінок – верхньої та нижньої, що може виявитися корисним під час аналізу надійності на етапі технічного завдання і етапі проектування.

Метод використовує поняття мінімального шляху та мінімального перерізу з теорії графів.

Для двополюсних графів під мінімальним шляхом розуміють підмножину послідовно суміжних один одному ребер графа, де видалення хоча б одного будь-якого ребра з цієї підмножини гарантовано призводить до втрати зв'язності та виходом двополюсного графа.

Мінімальним перерізом для двополюсного графа називають таку підмножину його ребер, видалення яких із графа призводить до гарантованої втрати зв'язності між вхідним і вихідним полюсами графа.

Імовірність безвідмовної роботи системи оцінюється використовуючи наступну нерівність:

$$P_{\text{нижн.меж}} \leq P_{\text{системи}} \leq P_{\text{верхн.меж.}}$$

Тут $P_{\text{нижн.меж.}}$ визначається як ймовірність безвідмовної роботи допоміжної

системи, що відповідає графу, складеному з послідовно з'єднаних підграфів, побудованих за всіма мінімальними перерізами, тобто. кожен підграфів, складається з паралельно з'єднаних ребер відповідного мінімального перерізу.

Надійність такої системи, можна записати у такому вигляді:

$$P_{\text{нижн.гран}}(F(X)) = \prod_{1 \leq j \leq N} (1 - \prod_{i \in A_j} (1 - p_i)),$$

де p_i - ймовірність безвідмовної роботи i -го модуля, що представляється ребром, та входить до складу j -го мінімального перерізу.

Аналогічно, $P_{\text{верхн.}}$ визначається як можливість безвідмовної роботи допоміжної системи, відповідної графу, складеному з паралельно з'єднаних підграфів, побудованих за всіма мінімальними шляхами двополюсного графа.

Надійність цієї допоміжної системи, можна представити у такому вигляді:

$$P_{\text{верхн.гран.}}(F(X)) = 1 - \prod_{1 \leq j \leq M} (1 - \prod_{i \in B_j} p_i)$$

де p_i - ймовірність безвідмовної роботи i -го модуля, що представляється ребром, що входить до складу j -го мінімального шляху, позначеного як B_j при загальній кількості мінімальних перерізів M .

2.2 Графи під час розрахунку надійності

Велика кількість територіально розподілених систем виконані з нахилом, щоб вони мали мережеву структуру. До них можна віднести комунікаційні системи, сучасні обчислювальні мережі, залізничні мережі, авіаційні лінії, мережі зв'язку і передачі даних, електроенергетика, різні трубопроводи. Всі елементи, з яких вони складаються можна поділити на дві групи: лінії зв'язку та місця відправлення.

В залежності від природи систем комунікаційні канали можуть приймати цілком матеріальну структуру – залізниці, нафтопроводи, системи електропередач і т.п., або

можуть бути уомглядними – морські шляхи, авіаліній, напрямки радіозв'язку. В обох випадках в системи є багато спільного. Через це, для математичного опису таких, на перший погляд різних за виглядом систем, доцільно використовувати моделі, які базуються на теорії графів. За допомогою математичного апарату з цієї теорії можна аналізувати надійності даних систем. Теорія графів є доволі природної для опису багатьох технічних систем (в основному різних комунікаційних систем). Першим етап складання моделі є уявлення системи у вигляді графа. Сформулювавши завдання в термінах графів можна використати апарат теорії графів для її.

Граф – це сукупність вершин (вузлів) та ребер, зв'язаних між собою. Можна сказати, що граф це абстрактна система, яка складається з двох множин - вершин і ребер. Різні вузли графа можуть бути пов'язані між собою ребрами. Вузол може мати ребра які в нього входять та які з нього виходять. Ребра можуть бути простими і кратними - кратні ребра інцидентні одній і тій же парі вершин, спрямованими і ненаправленими.

Якщо в графі, який має кінцеве число вершин (ще називають кінцевим графом), можна з будь-якої вершини потрапити в будь-яку іншу вершину, то говорять про зв'язність графа.

2.3 Графи на основі GL-моделей для підрахунку відмовостійкості ВБС

Введення відмовостійкості це один з методів підвищення надійності ОС. Питання про застосування відмовостійких ОС виникає тоді, коли інші методи підвищення надійності не можуть дати потрібний рівень надійності або коли вони економічно дорожчі. Складні системи реального часу, відмова яких може призвести до катастрофічних проблем, економічних втрат, а також системи, які працюють при певних зовнішніх умовах, повинні мати властивості відмовостійкості.

З кожним роком складні відмовостійкі багатопроцесорні системи з набувають все більшого поширення. Це можна пов'язати з розширенням області їх застосування, в основному в системах управління складними об'єктами. Для всіх обчислювальних систем характерним є велика кількість елементів та регулярність структури їх поєднання. Це трохи полегшує розрахунок потрібних параметрів. Для систем

управління все навпаки – доволі невелика кількість елементів (до тисячі) і нерегулярність структури. Нас цікавить те, що різні частини систем, що керують різними елементами об'єкта (підсистеми), повинні по-різному бути захищені від відмов.

Завдання забезпечення надійності для таких систем доволі однакові. Ми будемо використовувати абревіатуру ВБС (відмовостійкі багатопроцесорні системи) для таких систем, хоча будемо мати на увазі системи управління.

Базова ВБС - система, стійка до відмов певної кратності. Це означає, що число відмов не повинно перевищувати певної фіксованої величини. Але, як правило, сучасні ВБС не підкоряються цьому правилу: для сучасних систем можуть бути деякі правило, коли вони входять з ладу при виникненні k відмов і бути стійкою до s іншим відмов, навіть якщо $s > k$.

З вище сказаного факту можна підкреслити два моменти: дане визначення відноситься тільки до деякої частини системи, тобто - про базові підсистеми ВБС; коли ВБС має велику кількість різних модулів необхідно моделювати роботу системи розбуваючи ВБС на певну кількість підсистем. Якраз у другому випадку ми поговоримо про моделювання ВБС в потоці відмов. Це треба робити для розрахунку надійності ВБС.

Однією з особливостей відмовостійких багатопроцесорних систем керування є те, що доволі часто їх підсистеми не є базових. Тому побудову моделей ВБС в потоці відмов й аналіз підсистем та систем будемо проводити в два етапи: для базових та, для більш складних систем.

Під час проектування ВБС треба вирішувати багато різних задач, в число яких входить передбачення надлишкових засобів для тестування, опис як зберігати дані, які резерву використати і як відновлювати обчислювальний процесу. Щоб правильно і оптимально вибрати ресурси для системи, в тому числі які забезпечують її реконфігурацію, необхідна максимально точна оцінка надійності ВБС на ранніх етапах.

Математичні моделі ВБС треба будувати ще на перших етапах розробки задля

забезпечення високого рівня надійності системи. Через вартість таких систем, будування таких моделей доволі важливе завдання. Проведення експериментів, використовуючи створенні моделі дає можливість оцінити ефективність ВБС на етапі розробки. Важливим фактором є те, щоб вони були адекватні до структури та методам надійності ОС.

При розрахунку надійності складної ВБС недоцільно використовувати методи, які мають за основу моделі двополюсного графа. В цих моделях критерієм працездатності виступає наявність шляху між двома полюсами. Через те, що побудова двополюсного графа для аналізованої системи а також перебір шляхів для цього графа призводить до певних обчислювальних труднощів, використання таких графів для таких задач переводить їх в клас неконструктивних.

Для мережевих структур можна використовувати багатополісних граф. Для цієї моделі критерієм працездатності є зв'язність графа - система справна тільки коли існує зв'язок кожного вузла графа кожним. Завдання аналізу такої моделі полягає у визначенні ймовірності, що хоча б одне дерево графа системи є працездатним. Однак, при збільшенні складності мережі, задача перебору дерев призводить до обчислювальних труднощів, як і для двополюсних графів.

Методи моделювання, що дозволяють вирішити проблему перебору великого числа станів системи, за рахунок статистичних оцінок з деякою вірогідністю, в певній мірі втрачають користь через те, що, як правило, дуже складний опис критеріїв для структурних функцій, що відповідають працездатності системи.

Вище сказані приклади доводять актуальність побудови нових моделей поведінки ВБС, які дали б змогу зробити аналіз системи більш доступним і наглядним, в результаті чого спростився б розрахунок та була б отримана досить точна надійність системи. Для ефективного аналізу системи з багатьма станами, велику роль грає компактне подання критерію, який буде використовуватися для оцінка поведінки в певних умовах.

Головна ідея даного підходу полягає в моделі, що може поєднувати властивості графів і булевих функцій. Така модель має назву графо-логічна модель. Вона

передбачає використання зв'язності деякого графа і певних булевих функцій у ребрах цього графа, які в сукупності відображають поведінку системи в потоці відмов.

Різницею між відомими моделями, які використовують графи у своїй структурі, запропонована модель може мати в разі спрощену структуру ніж модель самої системи. Це стало можливим за допомогою «перенесення» складності в булеві функції, які використовуються у ребрах графа.

Основні принципи побудови графо - логічної моделі. Можна відзначити, що головна ідея цього підходу полягає в тому, що при створенні моделі є можливість міняти структуру графа і ставити її у відповідності до стану елементів системи. Це досягається приписуванням ребрам графа булевих функцій.

Критерієм відмови системи можна назвати порушення здатності її виконувати свої основні функції або ж коли вихідні дані перестають відповідати певним вимогам. Під час створення моделі системи критерій відмови визначається завдяки станів елементів системи. У таких умовах зазвичай кожен елемент системи робиться максимально простим, тобто може мати тільки два стани – працює і не працює.

Сукупність станів елементів системи визначає стан самої системи, тобто критерії відмови системи можна розділити на дві підмножини – коли система працює і коли не працює, беручи до уваги працездатність елементів на даний проміжок часу.

Підхід, який використовує графо-логічні моделі (GL - моделі) ВБС, робить аналіз системи більш простим та доступним. Відображенню поведінки системи відповідає зв'язність певного графа (GL - моделі), ребра якого певним образом відповідають булевим функціям.

Розглянемо будівництво GL-моделі, аналізована система має n модулів і стає переставати працювати, коли вийшли з ладу не менше, ніж $k + 1$ модулів ($k < n$). Така ВБС має назву k -ВБС. Базова ВБС – це система, стійка до певної кратності відмов. Дана система складається з 2^n різних станів. Кожну таку множину станів системи систему можна розбити на дві множини – працездатність та відмова.

Не всі сучасні системи можна розбити на такі множини, доволі часто можна

зустріти випадки, коли ВБС стає непрацездатною при виникненні k відмов і працювати при s інших відмов при $s > k$. Через це треба сказати про два важливі моменти:

1. Ми будемо говорити тільки про базові ВБС;
2. Коли система має велику кількість модулів необхідно моделювати її реакцію на ту чи іншу безліч відмов її елементів.

Другий випадок буде моделювати поведінку системи в потоці відмов. Це треба робити для розрахунку надійності системи. Особливістю вімовостійких багатопроцесорних систем є те, що доволі часто, їх підсистеми не є базовими, тому дослідження ВБС не можна зупиняти тільки на базових системах.

Будь яка ВБС це композиція різних k -ВБС. Будування GL-моделі виконується по наступному алгоритму. У відповідність до k -ВБС ставиться певний неорієнтований граф G . Ребра цього графа певним чином відповідають побудованим булевими функціями $f_1(x_1, \dots, x_n), \dots, f_l(x_1, \dots, x_n)$. Кожна змінна x_i , де $(i=1..n)$ позначає стан i -го елемента в системі, якщо елемент працює то $x_i = 1$, якщо ні то відповідно $x_i = 0$. Ребро, з відповідною йому функцією $f_i(x_1, \dots, x_n)$ буде в графі G , якщо $f_i(x_1, \dots, x_n) = 1$ і випадє з нього, якщо $f_i(x_1, \dots, x_n) = 0$.

Функції $f_1, \dots, f_l$ повинні володіти певною властивістю. Для всіх можливих наборів, що містять поєднання з не менше ніж $k + 1$ нульових розрядів, знайдеться таке r ($r < L$) функцій з множини $(f_1, \dots, f_l)$, які будуть дорівнювати нулю на цих наборах і позначають r ребр, при яких видалення цих ребр призведе до втрати зв'язності графа.

Доволі очевидним є факт, що найбільш складними, при побудові GL – моделі, є метод вибору графа і функцій $f_1, \dots, f_l$. Для k - ВБС, зазвичай використовують кільцеві графи, які стійкі до однієї відмови у своїй структурі. Цей граф і буде базовим.

2.4 Базові $K(2, n)$ системи та GL-моделі для них

Для ефективного аналізу систем, що мають велику кількість станів, дуже важливо мати компактне подання критерію, який використовується для оцінки системи від час відмов. Головна ідея нового підходу це побудова моделі, яка поєднує

властивості графів і булевих функцій. Таку модель будемо називати графо-логічною.

Ця модель, для відображення стану системи, використовує відповідності між зв'язністю графа і обраних булевих функцій. Особливістю даної моделі є те, що побудовані графи можуть мати доволі просту структуру, яка набагато простіше ніж змодельована. Простота побудованого графа виходить з того, що складність системи переноситься в булеві функції, які потім позначаються як ребра графа. Головна полягає в тому, що можна змінювати структуру графа до векторів стану системи.

Дослідимо спочатку простий випадок - система має n -модулів і перестає працювати при відмові не менше $k+1$ модулів, за умови $k < n$. Така ВБС має назву k -ВБС.

Будемо досліджувати систему, яку містить n модулів і стає перестає працювати, коли відмовляють більше двох її елементів. Таку систему прийняти називати $K(2,n)$ – системою або ж 2-ВБС. У побудованого неорієнтованого циклічного n -реберного графа (рис. 2.4.1) кожному i -му ребру відповідає булева функція

$$f_i(x_1, \dots, x_n) = \prod_{j=1, \left\lfloor \frac{n-1}{2} \right\rfloor} (x_i \vee x_{(i+j) \bmod(n)}), \quad (2.4.1)$$

Для кожного $i=(1, n)$.

Можна сказати, що кожне i -е ребро існує, якщо $f_i=1$, і відсутнє, якщо $f_i=0$

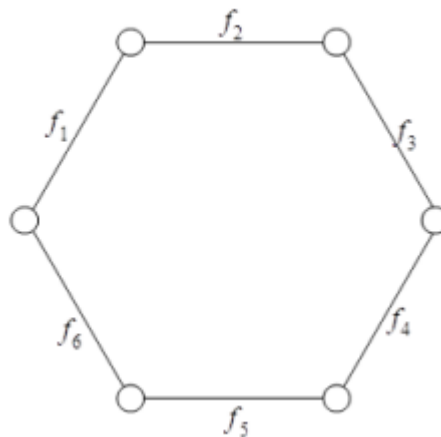


Рисунок 2.4.1 – G1-модель

Доведемо наступне твердження – граф втратить свою зв'язність, коли не менше трьох змінних з множини $\{x_1, \dots, x_n\}$ приймуть нульове значення.

Доведення:

Спочатку покажемо, що коли менше трьох змінних дорівнюють нулю, тільки одна функція може прийняти нульове значення.

Очевидно, що при коли тільки одна змінна дорівнює то $f_i \neq 0$ для будь-якого $i = (1, n)$.

Нехай декілька змінних $x_p, x_q \in \{x_1, \dots, x_n\}$ і тільки вони рівні нулю. Треба відмітити очевидне, в цьому випадку тільки f_p і f_q , із усіх f_i , $i = (1, n)$, можуть бути нульовими. Очевидно також, що $f_p = 0$, якщо виконується

$$q \in \left\{ (p+1) \bmod(n), (p+2) \bmod(n), \dots, \left(p + \left\lceil \frac{n-1}{2} \right\rceil \right) \bmod(n) \right\}, \quad (1)$$

а $f_q = 0$, якщо виконується

$$p \in \left\{ (q+1) \bmod(n), (q+2) \bmod(n), \dots, \left(q + \left\lceil \frac{n-1}{2} \right\rceil \right) \bmod(n) \right\} \quad (2)$$

Одночасне виконання (1) і (2) при $p \neq q$ неможливо, що легко доводиться. Отже, висловлене припущення справедливо.

Покажемо, що для 3 і більше змінних з $\{x_1, \dots, x_n\}$ рівних нулю, як мінімум дві функції f_k и f_l , де $(k = 1, n, l = 1, n, k \neq l)$ приймають нульові значення.

Візьмемо $x_p, x_q, x_r \in \{x_1, \dots, x_n\}$ і $p < q < r$. При виконанні (1) $f_p = 0$. До того ж виповняється одна з двох вимов, або:

$$r \in \left\{ (q+1) \bmod(n), (q+2) \bmod(n), \dots, \left(q + \left\lceil \frac{n-1}{2} \right\rceil \right) \bmod(n) \right\}, \quad (3)$$

$$p \in \left\{ (r+1) \bmod(n), (r+2) \bmod(n), \dots, \left(r + \left\lceil \frac{n-1}{2} \right\rceil \right) \bmod(n) \right\}. \quad (4)$$

і отже $f_q=0$, або (3) не вірно, і тоді, коли $r + \left\lceil \frac{(n-1)}{2} \right\rceil > n$, можна помітити, що

Отже, $f_r=0$.

Якщо умова (2) не виконується, з чого випливає очевидне твердження, що

$$r \notin \left\{ (p+1) \bmod(n), \dots, \left(p + \left\lceil \frac{n-1}{2} \right\rceil \right) \bmod(n) \right\}$$

то $f_p \neq 0$. Але тоді очевидно, що (3) та (4) виконуються. Отже, ми довели, що $f_q = 0$ і $f_r = 0$.

Доведене твердження дозволяє легко будувати графо-логічні моделі для будь якого n при $m=2$. Для випадку $n = 6$, модель вона має вигляд, який представлений на рисунку 2.1, де функції

$$f_1 = x_1 \vee x_2 \cdot x_3$$

$$f_2 = x_2 \vee x_3 \cdot x_4$$

$$f_3 = x_3 \vee x_4 \cdot x_5$$

$$f_4 = x_4 \vee x_5 \cdot x_6$$

$$f_5 = x_5 \vee x_6 \cdot x_1$$

$$f_6 = x_6 \vee x_1 \cdot x_2$$

Як бачимо, функція f_{i+1} робиться з функції f_i завдяки додавання 1 по $\bmod n$ к індексам всіх змінних. Число ребер рівно n .

Проблема цього підходу в тому, що він не дає гарного результату, коли $m>2$, тобто це неузагальнений метод. Як приклад наведемо деякі результати перших функцій для різних значень n при $m=3$.

$$n = 6 \quad f_1 = x_1 \vee x_2 \vee x_5 \cdot x_6$$

$$n = 7 \quad f_1 = x_1 \vee x_2 \cdot x_3 \vee x_5 \cdot x_7$$

$$n = 9 \quad f_1 = x_1 \vee x_2 \cdot x_3 \vee x_6 \cdot x_7 \cdot x_8 \cdot x_9$$

$$n = 10 \quad f_1 = x_1 \vee x_4 \cdot x_7 \cdot x_{10} \vee x_2 \cdot x_5 \cdot x_8$$

$$n = 12 \quad f_1 = x_1 \vee x_2 \cdot x_3 \cdot x_4 \vee (x_2 \vee x_7) \cdot x_{10} \cdot x_{11} \cdot x_{12} \cdot x_8 \cdot x_9$$

$$n = 14 \quad f_1 = x_1 \vee x_2 \cdot x_3 \cdot x_4 \cdot x_5 \cdot (x_6 \vee x_7) \vee (x_8 \vee \overline{x_4} \vee \overline{x_5}) \cdot \\ \cdot (x_9 \vee \overline{x_5}) \cdot x_{14} \cdot x_{10} \cdot x_{11} \cdot x_{12} \cdot x_{13}$$

Функції f_{i+1} формуються з f_i так само, як було описано вище.

Наведемо узагальнене рішення для 3-ВБС. Одразу покажемо вираз для $n = 120$, з якого можна буде побачити спосіб формування функцій в незалежності від значення n :

$$f_1 = x_1 \vee x_2 \cdot x_3 \cdot x_4 \cdot \dots \cdot x_{40} \cdot (x_{41} \vee x_{42}) \vee (\overline{x_{40}} \vee x_{79} \cdot (\overline{x_{39}} \vee x_{78} (\overline{x_{38}} \vee x_{77} (\overline{x_{37}} \vee x_{76} (\dots \\ \dots x_{44} (\overline{x_4} \vee x_{43})))))) \cdot x_{80} \cdot x_{81} \cdot x_{82} \cdot \dots \cdot x_{120}$$

Кількість функцій як і раніше дорівнює n , спосіб формування кожної наступної формується з попередньої так само просто, але цей рішення дуже не практична, як мінімум тому, що функції не виходять неповторними. Відкритим питанням залишається будівництво моделей для інших значень m .

2.5 GL-моделі для небазових ВБС

Ми вже зазначили про необхідність знаходження загального метода будівництва моделей для довільних чисел і неможливість узагальнення отриманих результатів.

Розглянемо k -ВБС (т.е. ВБС, яка буде працювати, якщо відмов буде $\leq k$), що складається з n елементів. Кожному із станів системи можна поставити у відповідність певну неупорядковану m - вибірку з n елементів. Надалі, під m -вибіркою, будемо мати на увазі, неупорядкована вибірка.

Нехай $n_u = \{a_1, a_2, \dots, a_n\}$ - певна підмножина всіх елементів k -ВБС. Для k -ВБС визначимо $K(m_v, n_u)$, як множину, що містить у собі m_v вибірок з n_u елементів.

Візьмемо $K(p, n_h) = \{a_1, a_2, \dots, a_p\}$, a_i - певна P - вибірка з n_h елементів, а $K(j, n_l) = \{b_1, b_2, \dots, b_j\}$, b_i - певна J -вибірка з n_l елементів. Добуток $K(p, n_h)$ та $K(j, n_l)$ визначимо наступним чином:

$K(p, n_h) \times K(j, n_l) = \{(a_1 \times b_1), (a_1 \times b_2), \dots, (a_1 \times b_o), \dots, (a_f \times b_1), (a_f \times b_2), \dots, (a_f \times b_o)\}$,
де $(a_i \times b_j)$ - конкатенація елементів вибірок a_i і b_j .

Далі множину всіх елементів системи так розіб'ємо на d непересічних підмножин, щоб в 1-ій підмножині було n_1 елементів, у 2-ій– n_2 елементів і т. д.. За відомою комбінаторною тотожністю

$$\sum_{m_1, \dots, m_q} C_{n_1}^{m_1} \cdot C_{n_2}^{m_2} \cdot C_{n_q}^{m_q} = C_n^m \quad (2.5.1)$$

Напишемо наступну нерівність

$$K(m, n) = \bigcup_{m_1, m_2, \dots, m_q} K(m_1, n_1) \times K(m_2, n_2) \times \dots \times K(m_q, n_q) \quad (2.5.2)$$

Використовуючи розбиття множини, яке представлене рівнянням (2.5.2), будемо будувати GL-моделі k-ВБС взявши за основу граф кільцевого типу для будь яких значень k . Використовуючи властивість кільцевого графа, який перестає бути зв'язним коли втрачає не менше двох ребер, можна легко встановити критерії відмови k-ВБС в термінах булевих функцій, які позначаються в ребрах цього графа. Таким критерієм буде рівність нулю значень більше двох функцій.

Система m-ВБС має n елементів, і вся ця множина, розбита на k підмножин з елементами $n_1, n_2, \dots, n_k$ відповідно. Нехай існують такі функції $f_1(x_1, \dots, x_n), \dots, f_L(x_1, \dots, x_n)$, що кожна з них дорівнює нулю, якщо як мінімум m змінних, що є станами m_i елементів n_i – підмножин ($i \in 1, k$). Отже – як мінімум дві функції з множини $f_1, \dots, f_L$ будуть приймати нульові значення, якщо, будь-які $m+1$ з n змінних будуть дорівнюють нулю.

Покажемо, що коли менше ніж $m+1$ змінна дорівнюють нулю тільки одна функція з $f_1, \dots, f_L$ може дорівнювати нулю. Буде достатньо дослідити випадок, коли тільки m змінних будуть нульовими. Так як, будь-якій m -вибірці з n змінних буде відповідати тільки одному набору $m_1, m_2, \dots, m_q$, припущення стає очевидним.

Нехай наступні змінні приймають нульове значення:

$$x_{\alpha_1} = 0, x_{\alpha_2} = 0, \dots, x_{\alpha_m} = 0, x_{\alpha_{m+1}} = 0,$$

де $\alpha_1, \alpha_2, \dots, \alpha_m, \alpha_{m+1}$ будь яка вибірка індексів з n можливих.

З елементів цього набору можна скласти не більше ніж

$$C_{m+1}^m = \frac{(m+1)!}{m!(m+1-m)!} = m+1 \quad \text{різних вибірок. Розглянемо деякі з них}$$

них, а саме: $(\alpha_1, \alpha_2, \dots, \alpha_{m-1}, \alpha_m)$, $(\alpha_1, \alpha_2, \dots, \alpha_{m-1}, \alpha_{m+1})$ і $(\alpha_2, \dots, \alpha_{m-1}, \alpha_m, \alpha_{m+1})$.

Для простоти будемо розуміти що кожному елементу вибірки α_i відповідає елемент x_{α_i}

Нехай є певна вибірка $(\alpha_1, \alpha_2, \dots, \alpha_m)$, яка відповідає єдиному набору значень $m_1, m_2, \dots, m_q$ з певним номером j . Доведемо, що хоча б одна вибірка не може відповідати набору з тим же номером j , який є для вибірки $\alpha_1, \alpha_2, \dots, \alpha_m$. З структури вибірки $\alpha_1, \alpha_2, \dots, \alpha_{m-1}, \alpha_{m+1}$ можна побачити, що α_{m+1} не може потрапити в підмножину n_i , що має елемент α_m , тоді може відбуватися зміна набору $m_1, m_2, \dots, m_q$. Завжди знаходиться α_{m+1} разом з одним з елементів $\alpha_1, \alpha_2, \dots, \alpha_{m-1}$, або ні – відбудеться збільшення на одиницю значення деякого розряду набору $m_1, m_2, \dots, m_q$. Якщо ж вийде так, що α_{m+1} знаходиться в одній підмножині разом з елементом α_m , тоді вибірці $(\alpha_1, \alpha_2, \dots, \alpha_{m-1}, \alpha_{m+1})$ буде відповідати той самий набір з номером j , що і вибірці $(\alpha_1, \alpha_2, \dots, \alpha_m)$.

Таким чином, усі 3 вибірки повинні відповідати не менш, ніж двом різних наборам значень чисел $m_1, m_2, \dots, m_q$, а отже - серед функцій $f_1, \dots, f_L$ знайдуться, хоча б дві функції, які візьмуть нульові значення. Що і потрібно було показати.

Можна сказати, що будь-яка функція f_i з розглянутої множини $f_1(x_1, \dots, x_n), \dots, f_L(x_1, \dots, x_n)$, яка задовольняє умовам теореми, може бути представлена в вигляді:

$$(2.5.3) \quad f_i = f_{m_1} \vee f_{m_2} \vee \dots \vee f_{m_q},$$

де

$$f_{m_j} = y_{a_1} \wedge y_{a_2} \wedge \dots \wedge y_{a_{N_j}}; \quad j = \overline{1, q}; \quad N_j = \overline{C_{n_j}^{m_j}};$$

$$y_{a_t} = x_{\alpha_1} \vee x_{\alpha_2} \vee \dots \vee x_{\alpha_{m_j}}; \quad t = \overline{1, \overline{C_{n_j}^{m_j}}}.$$

Можна зазначити, що функції f_{m_j} - приймають нульові значення, коли дорівнюють 0 всі змінні хоча б з однієї m_j - вибірки з n_j - підмножини.

Нехай $(x_{\alpha_1}, x_{\alpha_2}, \dots, x_{\alpha_m})$ - деякий набір m змінних з n – множини, і кожна змінна в цьому наборі приймає нульові значення. Тоді очевидно, що в цьому наборі є група змінних, що являє з себе якусь m_1 -вибірку з n_1 – підмножини, та група змінних, що являє якусь m_2 -вибірку з n_2 – підмножини, і так далі до групи змінних, що являють з себе якусь m_q - вибірку з n_q – підмножини. Групи змінних, які є в цьому наборі, відповідають деякому єдиному набору чисел $(m_1, m_2, \dots, m_q)$. Нехай, наприклад, m_1 -вибірка представляє набір змінних $(x_{\alpha_1}, x_{\alpha_2}, \dots, x_{\alpha_{m_1}})$. Тоді справедливо наступне:

$$y_{a_1} = x_{\alpha_1} \vee x_{\alpha_2} \vee \dots \vee x_{\alpha_{m_1}} = 0,$$

$$f_{m_1} = y_{a_1} \wedge y_{a_2} \wedge \dots \wedge y_{a_{N_1}} = 0, \quad (N_1 = \overline{C_{n_1}^{m_1}}).$$

Аналогічно для змінних, які складають m_2 - вибірку і т. д. отримаємо, що $f_{m_2}=0, \dots, f_{m_q}=0$. Отже, у цьому випадку функція f_i з виразу (2.5.3) приймає нульове значення.

Тепер розглянемо інші набори змінних, групи яких відповідають деякому набору чисел $\mathbf{m}' = (m_1', m_2', \dots, m_q')$, який відрізняється від розглянутого набору $\mathbf{m} = (m_1, m_2, \dots, m_q)$. Якщо набори $\mathbf{m}$ і $\mathbf{m}'$ не дорівнюють один одному, то для того, щоб виконувались рівності

$$\sum_{i=1}^q m_i = m,$$

$$\sum_{i=1}^q m_i' = m,$$

необхідно, щоб дві наступні нерівності теж виконувалися: $m_p' < m_p$ і $m' > m$, де p і r ($p \neq r$) – деякі цілі числа, які задовольняють нерівності $1 \leq p \leq q$ і $1 \leq r \leq q$. Отже, група змінних, складова будь-якою m' – вибірку з n – множини має містити хоча б одну одиницю. При цьому маємо:

$$y_{a_1} = 1, y_{a_2} = 1, \dots, y_{a_{N_j}} = 1; f_{m_j} = 1; f_i = 1.$$

Якщо ж число змінних у наборі менше числа m , то в одній з підмножин будуть змінні, що мають одиничні значення. Відповідно отримаємо $f_i = 1$.

Отже, функція f_i буде приймати нульове значення тільки, якщо не менш ніж m змінних приймають нульові значення. Також, ці змінні повинні представляти $m_1, m_2, \dots, m_q$ – вибірки з $n_1, n_2, \dots, n$ – підмножини відповідно, для єдиного набору чисел. Висновок: функція f_i задовольняє умовам, описаним вище.

Зробимо список правил, які будуть відповідати вище сказаним умовам, та будуть зручними для побудови GL -моделі:

1. Провести розбиття множини z на q непересічних підмножин з числом елементів, відповідно $n_1, n_2, \dots, n_q$;

$$\left(\sum_{i=1}^q n_i = n \right).$$

2. Перерахувати всі набори $m_1, m_2, \dots, m_q$ за умови, що

$$\sum_{i=1}^q m_i = m, m_i \leq \underline{n_i}.$$

3. Для кожного набору написати вираз виду:

$$K(m_1, n_1) \times K(m_2, n_2) \times \dots \times K(m_q, n_q) = \\ = \{(a_1 \times b_1 \times \dots \times c_1), (a_1 \times b_1 \times \dots \times c_2), \dots, (\underline{a_{n_1}} \times \underline{b_{n_2}} \times \dots \times \underline{c_{n_q}})\},$$

4. Здійснити перехід від виразу у п.3 до булевих виразів, використовуючи вираз:

$$f_i = (\bigwedge_i y_{a_i}) \vee (\bigwedge_j y_{b_j}) \vee \dots \vee (\bigwedge_k y_{c_k})$$

$$\text{де } y_{a_i} = \bigvee_{\sigma=1}^{m_1} x_{\alpha_\sigma}, y_{b_j} = \bigvee_{\theta=1}^{m_2} x_{\beta_\theta}, \dots, y_{c_k} = \bigvee_{\omega=1}^{m_q} x_{\gamma_\omega}.$$

Розглянемо таке розбиття вхідної множини, при якому функції $f_1, \dots, f_L$ спрощуються. Такий варіант можна досягнути, якщо у виразі (2.5.1) для всіх підмножин буде наступне: $m_j = 1$ або $m_j = n_j$. Для яких буде вірне наступне твердження – вираз для f_{mj} становлять елементарні кон'юнкції $f_{mj} = x_{a1} \wedge x_{a1} \wedge \dots \wedge x_{anj}$. Відповідно, при виборі даного варіанта розбиття множини отримуємо функції f_i у виразах (2.4) з більш простою формою запису. Тому розбиття n – множини треба здійснювати у декілька етапів. На першому етапі отримуємо певні підмножини, які далі також розбиваємо на підмножини виду $K(1, n_j)$ або $K(n_j, n_j)$.

2.6 Приклади

Наведемо декілька прикладів:

В нас є 3-ВБС і складається з 9 елементів. Для простоти позначимо номерами 1, 2, 3, 4, 5, 6, 7, 8, 9. Побудуємо GL - модель для цієї системи, використовуючи граф кільцевого типу.

Треба розбити системи на 3 підмножини:

$n_1 = \{1, 2, 3\}$, $n_2 = \{4, 5, 6\}$ і $n_3 = \{7, 8, 9\}$. У відповідності з виразом (2.5.2), отримаємо для множини $K(3, 9)$:

$$K(m, n) = \bigcup K(m_1, n_1) \times K(m_2, n_2) \times K(m_3, n_3), m_1, m_2, m_3$$

Для того, щоб визначити множин $K(m_i, n_i)$ треба скласти таблицю усіх можливих наборів чисел m_1, m_2, m_3 для умов даного прикладу

Таблиця 2.6.1

№ набору	Значення чисел		
	m_1	m_2	m_3
1	0	0	3
2	0	1	2
3	0	2	1
4	0	3	0
5	1	0	2
6	1	1	1
7	1	2	0
8	2	0	1
9	2	1	0
10	3	0	0

Напишем вираз для $K(3, 9)$:

$$K(3, 9) = (K(0, n_1) \times K(0, n_2) \times K(0, n_3)) \cup (K(0, n_1) \times K(1, n_2) \times K(2, n_3)) \cup (K(0, n_1) \times K(2, n_2) \times K(1, n_3)) \cup \dots \cup (K(2, n_1) \times K(1, n_2) \times K(0, n_3))$$

В цей вираз входять наступні множини:

- 1) $K(0, n_i)$, які можна залишити, бо вони нічого не дають;

- 2) $K(3, n_3) = \{(789)\}$;
 3) $K(2, n_3) = \{(78), (89), (79)\}$;
 4) $K(1, n_3) = \{(7), (8), (9)\}$;
 5) $K(1, n_2) = \{(4), (5), (6)\}$;
 6) $K(2, n_2) = \{(45), (56), (46)\}$;
 7) $K(3, n_2) = \{(456)\}$;
 8) $K(1, n_1) = \{(1), (2), (3)\}$;
 9) $K(2, n_1) = \{(12), (23), (13)\}$;
 10) $K(3, n_1) = \{(123)\}$;

Так як не всі підмножини, після розбиття, являють з себе множини типу $K(n_j, n_j)$ або $K(1, n_j)$ то для певних наборів нам треба знов провести розбиття. Результати можна побачити в таблиці 2.6.2.

Таблиця 2.6.2

№ набору	$n_1 = \{1, 2, 3\}$		$n_2 = \{4, 5, 6\}$		$n_3 = \{7, 8, 9\}$	
	n_{11}	n_{12}	n_{11}	n_{22}	n_{31}	n_{32}
	m_1		m_2		m_3	
	m_{11}	m_{11}	m_{21}	m_{22}	m_{31}	m_{32}
1	0		0		3	
2	0		1		1	1
3	0		1		2	0
4	1		0		1	1
5	1		0		2	0
6	0		3		0	
7	1		1	1	0	

8	1		2	0	0
9	0		1	1	1
10	0		2	0	1
11	3		0		0
12	1	1	1		0
13	2	0	1		0
14	1	1	0		1
15	2	0	0		1
16	1		1		1

Використаємо правило написання булевих функцій побудуємо GL-модель:

$$f_1 = x_7 \vee x_8 \vee x_9$$

$$f_2 = x_4 x_5 x_6 \vee x_7 x_8 \vee x_9$$

$$f_3 = x_4 x_5 x_6 \vee x_7 \vee x_8$$

$$f_4 = x_1 x_2 x_3 \vee x_7 x_8 \vee x_9$$

$$f_5 = x_1 x_2 x_3 \vee x_7 \vee x_8$$

$$f_6 = x_4 \vee x_5 \vee x_6$$

$$f_7 = x_1 x_2 x_3 \vee x_4 x_5 \vee x_6$$

$$f_8 = x_1 x_2 x_3 \vee x_4 \vee x_5$$

$$f_9 = x_7 x_8 x_9 \vee x_4 x_5 \vee x_6$$

$$f_{10} = x_7 x_8 x_9 \vee x_4 \vee x_5$$

$$f_{11} = x_1 \vee x_2 \vee x_3$$

$$f_{12} = x_4 x_5 x_6 \vee x_1 x_2 \vee x_3$$

$$f_{13} = x_4 x_5 x_6 \vee x_1 \vee x_2$$

$$f_{14} = x_7 x_8 x_9 \vee x_1 x_2 \vee x_3$$

$$f_{15} = x_7 x_8 x_9 \vee x_1 \vee x_2$$

$$f_{16} = x_1 x_2 x_3 \vee x_4 x_5 x_6 \vee x_7 x_8 x_9$$

Як можна побачити даний метод дозволяє побудувати GL -модель для k -ВБС при будь яких k .

Приклад 2

GL -моделі для $K(4, 10)$

В цьому випадку використаємо найбільш простий метод, який включає в себе розбиття на моделі виду $K(1, 2)$.

Розділимо $\{x_1, x_2, \dots, x_{10}\}$ на 5 підмножин $\{x_1, x_2\}$, $\{x_3, x_4\}$, $\{x_5, x_6\}$, $\{x_7, x_8\}$ і $\{x_9, x_{10}\}$.

Для кожної з яких справедливі такі формули

$$K(1, 2) = x_i x_{i+1} \quad (2.6.1)$$

$$K(2, 2) = x_i \vee x_{i+1}. \quad (2.6.2)$$

Побудуємо таблиці розподілу відмови для даної системи

Таблиця 2.6.3

$\{x_1, x_2\}$	$\{x_3, x_4\}$	$\{x_5, x_6\}$	$\{x_7, x_8\}$	$\{x_9, x_{10}\}$	
$K(1,2)$	$K(1,2)$	$K(1,2)$	$K(1,2)$		$f_1 = x_1 x_2 \vee x_3 x_4 \vee x_5 x_6 \vee x_7 x_8$
$K(1,2)$	$K(1,2)$	$K(1,2)$		$K(1,2)$	$f_2 = x_1 x_2 \vee x_3 x_4 \vee x_5 x_6 \vee x_9 x_{10}$
$K(1,2)$	$K(1,2)$		$K(1,2)$	$K(1,2)$	$f_3 = x_1 x_2 \vee x_3 x_4 \vee x_7 x_8 \vee x_9 x_{10}$
$K(1,2)$		$K(1,2)$	$K(1,2)$	$K(1,2)$	$f_4 = x_1 x_2 \vee x_9 x_{10} \vee x_5 x_6 \vee x_7 x_8$
	$K(1,2)$	$K(1,2)$	$K(1,2)$	$K(1,2)$	$f_5 = x_9 x_{10} \vee x_3 x_4 \vee x_5 x_6 \vee x_7 x_8$
$K(2,2)$	$K(1,2)$	$K(1,2)$			$f_6 = x_1 \vee x_2 \vee x_3 x_4 \vee x_5 x_6$
$K(2,2)$	$K(1,2)$		$K(1,2)$		$f_7 = x_1 \vee x_2 \vee x_3 x_4 \vee x_7 x_8$
$K(2,2)$	$K(1,2)$			$K(1,2)$	$f_8 = x_1 \vee x_2 \vee x_3 x_4 \vee x_9 x_{10}$
$K(2,2)$		$K(1,2)$	$K(1,2)$		$f_9 = x_1 \vee x_2 \vee x_5 x_6 \vee x_7 x_8$
$K(2,2)$		$K(1,2)$		$K(1,2)$	$f_{10} = x_1 \vee x_2 \vee x_5 x_6 \vee x_9 x_{10}$
$K(2,2)$			$K(1,2)$	$K(1,2)$	$f_{11} = x_1 \vee x_2 \vee x_7 x_8 \vee x_9 x_{10}$
$K(1,2)$	$K(2,2)$	$K(1,2)$			$f_{12} = x_3 \vee x_4 \vee x_1 x_2 \vee x_5 x_6$

$K(1,2)$	$K(2,2)$		$K(1,2)$		$f_{13} = x_3 \vee x_4 \vee x_1 x_2 \vee x_7 x_8$
$K(1,2)$	$K(2,2)$			$K(1,2)$	$f_{14} = x_3 \vee x_4 \vee x_1 x_2 \vee x_9 x_{10}$
	$K(2,2)$	$K(1,2)$	$K(1,2)$		$f_{15} = x_3 \vee x_4 \vee x_7 x_8 \vee x_5 x_6$
	$K(2,2)$	$K(1,2)$		$K(1,2)$	$f_{16} = x_3 \vee x_4 \vee x_9 x_{10} \vee x_5 x_6$
	$K(2,2)$		$K(1,2)$	$K(1,2)$	$f_{17} = x_3 \vee x_4 \vee x_7 x_8 \vee x_9 x_{10}$
$K(1,2)$	$K(1,2)$	$K(2,2)$			$f_{18} = x_5 \vee x_6 \vee x_1 x_2 \vee x_3 x_4$
$K(1,2)$		$K(2,2)$	$K(1,2)$		$f_{19} = x_5 \vee x_6 \vee x_1 x_2 \vee x_7 x_8$
$K(1,2)$		$K(2,2)$		$K(1,2)$	$f_{20} = x_5 \vee x_6 \vee x_1 x_2 \vee x_9 x_{10}$
	$K(1,2)$	$K(2,2)$	$K(1,2)$		$f_{21} = x_5 \vee x_6 \vee x_3 x_4 \vee x_7 x_8$
	$K(1,2)$	$K(2,2)$		$K(1,2)$	$f_{22} = x_5 \vee x_6 \vee x_3 x_4 \vee x_9 x_{10}$
		$K(2,2)$	$K(1,2)$	$K(1,2)$	$f_{23} = x_5 \vee x_6 \vee x_7 x_8 \vee x_9 x_{10}$
$K(1,2)$	$K(1,2)$		$K(2,2)$		$f_{24} = x_7 \vee x_8 \vee x_1 x_2 \vee x_3 x_4$
$K(1,2)$		$K(1,2)$	$K(2,2)$		$f_{25} = x_7 \vee x_8 \vee x_1 x_2 \vee x_5 x_6$
$K(1,2)$			$K(2,2)$	$K(1,2)$	$f_{26} = x_7 \vee x_8 \vee x_1 x_2 \vee x_9 x_{10}$
	$K(1,2)$	$K(1,2)$	$K(2,2)$		$f_{27} = x_7 \vee x_8 \vee x_5 x_6 \vee x_3 x_4$
	$K(1,2)$		$K(2,2)$	$K(1,2)$	$f_{28} = x_7 \vee x_8 \vee x_9 x_{10} \vee x_3 x_4$
		$K(1,2)$	$K(2,2)$	$K(1,2)$	$f_{29} = x_7 \vee x_8 \vee x_5 x_6 \vee x_9 x_{10}$
$K(1,2)$	$K(1,2)$			$K(2,2)$	$f_{30} = x_9 \vee x_{10} \vee x_1 x_2 \vee x_3 x_4$
$K(1,2)$		$K(1,2)$		$K(2,2)$	$f_{31} = x_9 \vee x_{10} \vee x_1 x_2 \vee x_5 x_6$
$K(1,2)$			$K(1,2)$	$K(2,2)$	$f_{32} = x_9 \vee x_{10} \vee x_1 x_2 \vee x_7 x_8$
	$K(1,2)$	$K(1,2)$		$K(2,2)$	$f_{33} = x_9 \vee x_{10} \vee x_5 x_6 \vee x_3 x_4$
	$K(1,2)$		$K(1,2)$	$K(2,2)$	$f_{34} = x_9 \vee x_{10} \vee x_7 x_8 \vee x_3 x_4$
		$K(1,2)$	$K(1,2)$	$K(2,2)$	$f_{35} = x_9 \vee x_{10} \vee x_5 x_6 \vee x_7 x_8$
$K(2,2)$	$K(2,2)$				$f_{36} = x_1 \vee x_2 \vee x_3 \vee x_4$
$K(2,2)$		$K(2,2)$			$f_{37} = x_1 \vee x_2 \vee x_5 \vee x_6$
$K(2,2)$			$K(2,2)$		$f_{38} = x_1 \vee x_2 \vee x_7 \vee x_8$
$K(2,2)$				$K(2,2)$	$f_{39} = x_1 \vee x_2 \vee x_9 \vee x_{10}$

	$K(2,2)$	$K(2,2)$			$f_{40} = x_3 \vee x_4 \vee x_5 \vee x_6$
	$K(2,2)$		$K(2,2)$		$f_{41} = x_3 \vee x_4 \vee x_7 \vee x_8$
	$K(2,2)$			$K(2,2)$	$f_{42} = x_3 \vee x_4 \vee x_9 \vee x_{10}$
		$K(2,2)$	$K(2,2)$		$f_{43} = x_5 \vee x_6 \vee x_7 \vee x_8$
		$K(2,2)$		$K(2,2)$	$f_{44} = x_5 \vee x_6 \vee x_9 \vee x_{10}$
			$K(2,2)$	$K(2,2)$	$f_{45} = x_7 \vee x_8 \vee x_9 \vee x_{10}$

Отримали 45-реберний граф

Другий спосіб

Можна розбити першу множину на 2 підмножини- $\{x_1, x_2, x_3, x_4, x_5\}$ і $\{x_6, x_7, x_8, x_9, x_{10}\}$. Побудуємо таблицю для цього випадка. 2.4

Таблиця 2.6.4

$\{x_1, x_2, x_3, x_4, x_5\}$	$\{x_6, x_7, x_8, x_9, x_{10}\}$
$K(4, 5)$	
$K(3, 5)$	$K(1,5)$
$K(2, 5)$	$K(2, 5)$
$K(1, 5)$	$K(3, 5)$
	$K(4, 5)$

Тепер треба розбити їх на ще менші підмножини: $\{x_1, x_2\}$, $\{x_3, x_4\}$, $\{x_5\}$ для першої множини і $\{x_6, x_7\}$, $\{x_8, x_9\}$, $\{x_{10}\}$ для другої. Далі побудуємо $K(1, 2)$ та $K(2, 2)$ для цих менших підмножин, для всіх варіантів з таблиці 2.6.4. Для $K(4, 5)$ таблиці 2.6.5 та 2.6.6, для $K(3, 5)$ – 2.6.7 та 2.6.8, для $K(2, 1)$ – 2.6.9 та 2.6.10

Таблиця 2.6.5

$\{x_1, x_2\}$	$\{x_3, x_4\}$	$\{x_5\}$
$K(2, 2)$	$K(2, 2)$	
$K(2, 2)$	$K(1, 2)$	$K(1, 1)$

$K(1, 2)$	$K(2, 2)$	$K(1, 1)$
-----------	-----------	-----------

Таблица 2.6.6

$\{x_6, x_7\}$	$\{x_8, x_9\}$	$\{x_{10}\}$
$K(2, 2)$	$K(2, 2)$	
$K(2, 2)$	$K(1, 2)$	$K(1, 1)$
$K(1, 2)$	$K(2, 2)$	$K(1, 1)$

Таблица 2.6.7

$\{x_1, x_2\}$	$\{x_3, x_4\}$	$\{x_5\}$
$K(1, 2)$	$K(1, 2)$	$K(1, 1)$
$K(2, 2)$	$K(1, 2)$	
$K(1, 2)$	$K(2, 2)$	
$K(2, 2)$		$K(1, 1)$
	$K(2, 2)$	$K(1, 1)$

Таблица 2.6.8

$\{x_6, x_7\}$	$\{x_8, x_9\}$	$\{x_{10}\}$
$K(1, 2)$	$K(1, 2)$	$K(1, 1)$
$K(2, 2)$	$K(1, 2)$	
$K(1, 2)$	$K(2, 2)$	
$K(2, 2)$		$K(1, 1)$
	$K(2, 2)$	$K(1, 1)$

Таблица 2.6.9

$\{x_1, x_2\}$	$\{x_3, x_4\}$	$\{x_5\}$
$K(1, 2)$	$K(1, 2)$	
$K(1, 2)$		$K(1, 1)$
	$K(1, 2)$	$K(1, 1)$

$K(2, 2)$		
	$K(2, 2)$	

Таблиця 2.6.10

$\{x_6, x_7\}$	$\{x_8, x_9\}$	$\{x_{10}\}$
$K(1, 2)$	$K(1, 2)$	
$K(1, 2)$		$K(1, 1)$
	$K(1, 2)$	$K(1, 1)$
$K(2, 2)$		
	$K(2, 2)$	

Тепер побудуємо функції ребр графа:

$$f_1 = x_1 \vee x_2 \vee x_3 \vee x_4$$

$$f_2 = x_1 \vee x_2 \vee x_3 x_4 \vee x_5$$

$$f_3 = x_3 \vee x_4 \vee x_1 x_2 \vee x_5$$

$$f_4 = x_6 \vee x_7 \vee x_8 \vee x_9$$

$$f_5 = x_6 \vee x_7 \vee x_8 x_9 \vee x_{10}$$

$$f_6 = x_8 \vee x_9 \vee x_6 x_7 \vee x_{10}$$

$$f_7 = x_1 x_2 \vee x_3 x_4 \vee x_5 \vee x_6 x_7 x_8 x_9 x_{10}$$

$$f_8 = x_1 \vee x_2 \vee x_3 x_4 \vee x_6 x_7 x_8 x_9 x_{10}$$

$$f_9 = x_1 x_2 \vee x_3 \vee x_4 \vee x_6 x_7 x_8 x_9 x_{10}$$

$$f_{10} = x_1 x_2 \vee x_5 \vee x_6 x_7 x_8 x_9 x_{10}$$

$$f_{11} = x_3 x_4 \vee x_5 \vee x_6 x_7 x_8 x_9 x_{10}$$

$$f_{12} = x_6 x_7 \vee x_8 x_9 \vee x_{10} \vee x_1 x_2 x_3 x_4 x_5$$

$$f_{13} = x_6 x_7 \vee x_8 \vee x_9 \vee x_1 x_2 x_3 x_4 x_5$$

$$f_{14} = x_6 x_7 \vee x_8 \vee x_9 \vee x_1 x_2 x_3 x_4 x_5$$

$$f_{15} = x_6 x_7 \vee x_{10} \vee x_1 x_2 x_3 x_4 x_5$$

$$f_{16} = x_8 x_9 \vee x_{10} \vee x_1 x_2 x_3 x_4 x_5$$

$$f_{17} = x_1x_2 \vee x_3x_4 \vee x_6x_7 \vee x_8x_9$$

$$f_{18} = x_1x_2 \vee x_3x_4 \vee x_6x_7 \vee x_{10}$$

$$f_{19} = x_1x_2 \vee x_3x_4 \vee x_{10} \vee x_8x_9$$

$$f_{20} = x_1x_2 \vee x_3x_4 \vee x_6 \vee x_7$$

$$f_{21} = x_1x_2 \vee x_3x_4 \vee x_8 \vee x_9$$

$$f_{22} = x_1x_2 \vee x_5 \vee x_6x_7 \vee x_8x_9$$

$$f_{23} = x_1x_2 \vee x_5 \vee x_6x_7 \vee x_{10}$$

$$f_{24} = x_1x_2 \vee x_5 \vee x_{10} \vee x_8x_9$$

$$f_{25} = x_1x_2 \vee x_5 \vee x_6 \vee x_7$$

$$f_{26} = x_1x_2 \vee x_5 \vee x_8 \vee x_9$$

$$f_{27} = x_3x_4 \vee x_5 \vee x_6x_7 \vee x_8x_9$$

$$f_{28} = x_3x_4 \vee x_5 \vee x_6x_7 \vee x_{10}$$

$$f_{29} = x_3x_4 \vee x_5 \vee x_{10} \vee x_8x_9$$

$$f_{30} = x_3x_4 \vee x_5 \vee x_6 \vee x_7$$

$$f_{31} = x_3x_4 \vee x_5 \vee x_8 \vee x_9$$

$$f_{32} = x_1 \vee x_2 \vee x_6x_7 \vee x_{10}$$

$$f_{33} = x_1 \vee x_2 \vee x_6x_7 \vee x_8x_9$$

$$f_{34} = x_1 \vee x_2 \vee x_{10} \vee x_8x_9$$

$$f_{35} = x_1 \vee x_2 \vee x_6 \vee x_7$$

$$f_{36} = x_1 \vee x_2 \vee x_8 \vee x_9$$

$$f_{37} = x_3 \vee x_4 \vee x_6x_7 \vee x_{10}$$

$$f_{38} = x_3 \vee x_4 \vee x_6x_7 \vee x_8x_9$$

$$f_{39} = x_3 \vee x_4 \vee x_{10} \vee x_8x_9$$

$$f_{40} = x_3 \vee x_4 \vee x_6 \vee x_7$$

$$f_{41} = x_3 \vee x_4 \vee x_8 \vee x_9$$

Висновки до розділу 2

У цьому розділі були представленні методи використання графів для підрахунку надійностей складних систем. Були представленні так звані двополюсні графи і їх різновиди.

Представлений алгоритм будування GL-моделей дозволяють будувати графи для систем будь якої складності

З ІСНУЮЧІ МЕТОДИ ПІДРАХУНКУ НАДІЙНОСТІ ДЛЯ БАЗОВИХ ВБС

3.1 Структурна функція k -out-of- n :G и k -out-of- n :F системи

Багато зарубіжних робіт, останнім часом, досліджують k -out-of- n (або « k з n ») системи – а саме їх властивостей, різні алгоритми оптимізації визначення стану системи, пропонують формули для розрахунку надійності. K -out-of n система може працювати, якщо k модулів з n функціонують, по нашому визначенню це - базова система. З цього можна підчеркнути інтерес до теоретичної і практичної важливості проблеми. До того ж, доволі часто, можна побачити певні обмеження які облегшують рішення задач. Ось наприклад декілька зарубіжних методик підрахунку надійності, розглянемо їх та відзначимо переваги і недоліки.

Можна відзначити, що взагалі є два розбиття k -out-of- n систем: k -out-of- n : G і k -out-of- n : F системи.

k -out-of- n : G система – це система, яка працює (може виконувати свої функції) тільки коли працюють хоча б k її елементів. k -out-of- n : F система – це система, що має n компонентів і перестає працювати, коли відмовляють хоча б k елементів. З цього можна визначити, що k -out-of- n : G система це $(n - k + 1)$ -out-of- n : F система. Доволі часто терміном « K -out-of- n система» позначають будь яку з них або обидві системи. Є окремі випадки k -out-of- n систем – послідовні (система еквівалентна 1 -out-of- n : F системі і n -out-of- n : G системі) та паралельна (система еквівалентна n -out-of- n : F системі і 1 -out-of- n : G системі).

Надійність k -out-of- n : G системи, у якої компоненти можуть мати різну надійність, можна розрахувати використовуючи метод мінімального набору шляхів і розрізів. Є декілька доволі ефективних алгоритмів для підрахунку надійності таких систем - Барлоу і Хейдтманном (Barlow and Heidtmann) та - Рушді (Rushdi). Складність обох цих алгоритмів однакова - порядку $O(k(n-k+1))$.

Логічна структурна функція - це математична модель системи, на якій базуються всі ці методи. Працездатність елементів системи характеризується

логічний вектором $x = (x_1, x_2, \dots, x_n)$ функції $F(X)$.

Ознакою працездатності і відмови i -го елемента буде значення $x_i = 1$ і $x_i = 0$ відповідно, а сама функція $F(X)$ побудована так, щоб вона дорівнювала одиниці тільки коли система працездатна. Беремо до уваги, що $x \geq y$, якщо $x_i \geq y_i$ для усіх $i = 1, n$.

Симетричними елементами системи називають такі компоненти, перестановка яких в системі не впливає на її стан. Тобто, компоненти i та j можна назвати симетричними, якщо для векторів стану системи виконується наступна умова:

$$\varphi(x_1, \dots, x_{i-1}, x_i, x_{i+1}, \dots, x_{j-1}, x_j, x_{j+1}, \dots, x_n) = \varphi(x_1, \dots, x_{i-1}, x_j, x_{i+1}, \dots, x_{j-1}, x_i, x_{j+1}, \dots, x_n).$$

Симетричною називають структурна функцію, якщо всі її компоненти симетричні між собою. Симетричні системи можуть включати в себе всі вище сказані структура, а саме послідовні, паралельні і k-out-of-n.

3.2 Мінімальний набір шляхів і розділів

Це найбільш простий метод оцінки значення ймовірності безвідмовної роботи (дозволяє оцінити зверху і знизу) тому він дозволяє оцінювати досить складні системи.

Щоб описати метод використовується логічна функція системи. Визначимо поняття «мінімальний шлях» і «мінімальний розріз». Якщо $F(X) = 1$ і $F(Y) = 0$ при будь-яких $y < x$, тоді мінімальним шляхом ($x = \alpha$) —називають мінімальний шлях, який складається з локально мінімальної сукупності підсистем, яка необхідна для забезпечення роботи системи в незалежності від інших підсистем. Зазвичай у системі, є декілька мінімальних шляхів. Ознакою мінімального шляху є те, що відмова хоча б однієї підсистеми призводить до відмови системи.

Якщо $F(X) = 0$ і $F(Y) = 1$ при будь-якому $y > x$, то мінімальним перерізом ($x = \beta$) є мінімальна сукупність підсистем, які ,при одночасній відмові, призведе до відмови всієї системи. В цього метода є своя характерна особливість - відновлення однієї підсистеми в мініальному розрізі зробить системи працездатною.

Використовуючи цей метод можна отримати тільки оцінку ймовірності знизу і зверху p_n та p_v , яку можна показати наступною нерівністю:

$$p_n \leq p_c \leq p_v.$$

p_n це ймовірність безвідмовної роботи допоміжної системи, яка складається з послідовно працюючих груп підсистем, які відповідають мінімальним перетинам системи, групи яких складаються з паралельно включених підсистем відповідного мінімального розрізу.

p_v це ймовірність безвідмовної роботи допоміжної системи, яка складена з паралельно працюючих груп підсистем, які відповідають мінімальним шляхам системи, група яких складаються з послідовно включених підсистем відповідного мінімального шляху

Отже цей метод (мінімальних шляхів і мінімальних розрізів) може звести аналіз систем до аналізу простої систем яка будується завдяки послідовно-паралельної і паралельно-послідовною структури, через це він може бути використаний під час аналізу складних систем. Головним недоліком методу є те, що рішення не буде точним, воно буде у вигляді оцінок знизу і зверху.

Як вже було зазначено, надійність системи дорівнює ймовірності, що працює як мінімум один з мінімальних наборів шляхів. Ненадійність, це коли хоча б один мінімальний розріз відмовив. Мінімальний набір шляхів буде працювати, коли всі його компоненти працюють.

Можна зазначити, що мінімальний набір шляхів має рівно k різних елементів, а мінімальний набір розрізів - рівно $(n - k + 1)$ різних елементів. Тобто ми знаємо всі мінімальні набори розрізів і шляхів. Основною задачею є визначення ймовірності, що хоча б в одному мінімальному шляху всі елементи працюють, чи навпаки – що у мінімальному розрізі всі елементи відмовили.

Якщо для k -out-of- n : G систем відомі всі мінімальні набори шляхів і розрізів можна використовувати метод ВВ(включення - виключення) для оцінки надійності. У цього метода є головний недолік - через велику кількість скорочуваних змінних, що негативно впливає на точність.

3.3 Барлоу і Хейдтманна

Було представлено два алгоритми оцінки надійності систем з незалежними елементами від Барлоу і Хейдтманна. Перший використовує певну функцію генерування і її форму:

$$g_n(z) = \prod_{i=0}^n (q_i + p_i z) = \sum_{i=0}^n P_e(i, n) z^i,$$

де z – фіктивна змінна, а $P_e(i, n)$ – ймовірність працювання i -го компонента. З форми $g_n(z)$ можна побачити, що $P_e(i, n)$ також представляє коефіцієнт біля z^i для функції генерації. Всі значення для $P_e(i, n)$ обчислюються використовуючи рекурсію. Можна сказати, що алгоритм базується на формулі (таке пояснення запропонував Рушді)

$$P(k, n) = \sum_{i=k}^n P_e(i, n),$$

яка прямо випливає з визначення k-out-of-n: G системи. Алгоритм вираховує значення $P_e(i, n)$ з цього рекурсивного співвідношення

$$P_e(i, j) = q_j P_e(i, j-1) + p_j P_e(i-1, j-1), \quad 0 \leq i \leq n, \quad 0 \leq j \leq n,$$

з граничними умовами

$$P_e(-1, j) = P_e(j+1, j) = 0 \quad \text{для } j = 0, 1, 2, \dots$$

$$P_e(0, 0) = 1.$$

Треба скласти функцію генерування щоб перейти до рекурсивного відношення:

$$g_{j-1}(z) = \prod_{i=1}^{j-1} (q_i + p_i z) = \sum_{i=0}^{j-1} P_e(i, j-1) z^i.$$

Через те, що $g_j(z) = (q_j + p_j z)g_{j-1}(z)$, то порівняння коефіцієнтів при z^i в обох частинах виразу

$$\sum_{i=0}^j P_e(i, j) z^i = (q_j + p_j z) \sum_{i=0}^{j-1} P_e(i, j-1) z^i = \sum_{i=0}^j [q_j P_e(i, j-1) + p_j P_e(i-1, j-1)] z^i$$

Було досліджено загальну кількість значень $P_e(i, j)$, які мають бути пораховані використовуючи граничні умови, було доведено що це значення дорівнює

$$(n - k + 1)(k + 1) - 1 + \frac{1}{2} (n - k)^2.$$

Можна порахувати, що загальна кількість арифметичних операцій, які треба зробити, щоб знайти значення надійності системи дорівнює

$$3[(n - k + 1)(k + 1) - \frac{1}{2} (n - k)^2] + n - k = (n - k)(1,5n + 1,5k + 4) + 3k.$$

Отже, можна зробити висновок, що складність алгоритму порядку $O(n^2)$ коли k прагне до 1 і порядку $O(n)$ коли k пригне до n .

3.4 Метод Рушді

Цей метод базується на аналізі структурної функції k -out-of- n : G системи. З визначення структурної функції $\phi(x)$ вона є симетричною. Через те, що вона може приймати тільки два значення (0 і 1), так як стан компонентів подається у двійковому значенні, то цей метод називають методом симетричної перемикальної функції.

x_i позначає стан i -го елемента, а $S(k, n)$ позначає структурну функцію системи. Обидві ці змінні є двійковими, якщо середнє арифметичне цих значень дорівнює одиниці, то це означає робочий стан, а нульове - відмову. Можна визначити такі функції надійності і ненадійності системи:

$$P(k, n) = \Pr(S(k, n) = 1), \quad Q(k, n) = \Pr(\bar{S}(k, n) = 1).$$

Зробимо розкладання по n -му компоненту, щоб знайти вираз, який визначає стан системи:

$$S(k, n) = x_n(S(k-1, n-1)) + \bar{x}_n(S(k, n-1)),$$

$$\bar{S}(k, n) = x_n(\bar{S}(k-1, n-1)) + \bar{x}_n(\bar{S}(k, n-1)).$$

Використовуючи ці два рівняння, можна сказати, що стан k -out-of- n : G системи

можна виразити як функцію із станами двох підсистем з такими ж $n-1$ компонентами. Цім двом системами треба різна кількість компонентів, однією необхідно, k працюючих елементів, а інший - $k - 1$ елемент. Візьмемо систему, яка має j компонентів, і якій необхідно, щоб хоча б i компонент були справні. Напишемо рівняння для цієї системи:

$$S(i, j) = x_j(S(i-1, j-1)) + \bar{x}_j(S(i, j-1)),$$

$$\bar{S}(i, j) = x_j(\bar{S}(j-1, i-1)) + \bar{x}_j(\bar{S}(i, j-1)),$$

де i може бути будь яким цілим значенням від 1 до k , а j - від 0 до n . Напишемо граничні умови для цих рівнянь:

$$S(0, j) = \bar{S}(j+1, j) = 1,$$

$$S(j+1, j) = \bar{S}(0, j) = 0.$$

Вважаючи, що компоненти незалежні, рівняння можна перетворити для наступних виразів надійності:

$$P(i, j) = p_j P(i-1, j-1) + q_j P(i, j-1),$$

$$Q(i, j) = p_j Q(i-1, j-1) + q_j Q(i, j-1).$$

Граничні умови для них будуть наступні:

$$P(0, j) = Q(j+1, j) = 1,$$

$$P(j+1, j) = Q(0, j) = 0.$$

Висновки до розділу 3

У цьому розділі були представленні альтернативні методи підрахунку надійності ВБС, такі як Метод Рушді, метод Барлоу і Хайдменна та метод мінімальних наборів шляхів і перерізів. Показане їх застосування та недоліки

4 МОДИФІКОВАНИЙ АЛГОРИТМ БУДУВАННЯ GL-МОДЕЛЕЙ ДЛЯ СКЛАДНИХ СИСТЕМ, ДЕ ПЕВНІ ПРОЦЕСОРИ МОЖУТЬ БРАТИ НА СЕБЕ РОБОТУ ІНШИХ.

4.1 Алгоритм будування GL-моделі простої системи

Як вже було сказано, в сучасному світі існує безліч різновидів складних систем управління. Кожна така система може мати свої унікальні характеристики і можливості. Неможливо створити один унікальний алгоритм, котрий буде однаково ефективний для систем з різними властивостями.

Розглянемо випадок, коли система складається з декількох підсистем. Доволі часто можна зустріти приклади, коли деякі підсистеми можуть брати на себе завдання інших, якщо вони відмовили. Саме такий випадок і стане об'єктом дослідження.

Опишемо саму просту систему, яка підпадає під вищесказані умови:

- системи складається з двох базових підсистем N та P ;
- кожна підсистема має вигляд $K(1, n)$;
- система P має унікальні процесори, які можуть брати на себе роботу процесорів системи N під час їх відмови, але не навпаки.

Розглянемо всі можливі випадки відмови даної системи. Уточнимо, що система вся система буде працювати тільки коли працюють всі її підсистеми.

1. Відмова 2 чи більше елементів підсистеми P . Так як, по умові, системи мають вигляд $K(1, n)$, то по визначенню базової системи кількість відмов більша допустимих – підсистема не працює;
2. Відмова 2 чи більше елементів у обох підсистемах;
3. Відмова 3 чи більше елементів у системі N , так як роботу одного процесора може взяти на себе система P , то насправді, в цьому випадку відмовили 2 процесора в системі N ;
4. Відмова 2 чи більше елементів в N та одного в P .

Всі інші комбінації не призводять до відмови працездатності всієї системи,

тобто модель втрачає не більше одного ребра – граф залишається зв'язним.

Дослідивши всі умови, можна зробити висновок – дану систему можна представити наступним чином: для В так само як МВР-модель 1-відмовостійкої ВБС, для підсистеми А, як для МВР-моделі 2-відмовостійкої системи.

Приклад

Покажемо різницю між звичайним і модифікованим алгоритмом на прикладі. Нехай в нас є система яка складається з двох підсистем $N = K(1, 6)$ та $P = K(1, 3)$. Елементи системи Р можуть брати на себе роботу інших під час їх відмов.

Не модифікований алгоритм:

Так як це 1-ВБС системи то реберні функції будуть дорівнювати елементам системи, тобто:

$$f_1 = n_1$$

$$f_2 = n_2$$

$$f_3 = n_3$$

$$f_4 = n_4$$

$$f_5 = n_5$$

$$f_6 = n_6$$

$$f_7 = p_1$$

$$f_8 = p_2$$

$$f_9 = p_3$$

Отримали 9-реберний кільцевий граф.

Модифікований алгоритм:

Тепер N система є 2-ВБС системою, розділимо елементи цієї системи на підмножини:

$\{n_1, n_2, \dots, n_6\}$ на 3 підмножин $\{n_1, n_2\}$, $\{n_3, n_4\}$, $\{n_5, n_6\}$

Для кожної з них вірно:

$$K(1, 2) = n_i n_{i+1}$$

$$K(2, 2) = n_i \vee n_{i+1}.$$

Побудуємо таблицю розподіли відмов по підмножинам та напишемо реберні функції:

Таблиця 4.1.1

$\{n_1, n_2\}$	$\{n_3, n_4\}$	$\{n_5, n_6\}$	
		$K(2,2)$	$f_1 = n_5 \vee n_6$
	$K(1,2)$	$K(1,2)$	$f_2 = n_3 n_4 \vee n_5 n_6$
	$K(2,2)$		$f_3 = n_3 \vee n_4$
$K(1,2)$	$K(1,2)$		$f_4 = n_1 n_2 \vee n_3 n_4$
$K(2,2)$			$f_5 = n_1 \vee n_2$

В результаті функції графа мають такий вигляд:

$$f_1 = n_5 \vee n_6$$

$$f_2 = n_3 n_4 \vee n_5 n_6$$

$$f_3 = n_3 \vee n_4$$

$$f_4 = n_1 n_2 \vee n_3 n_4$$

$$f_5 = n_1 \vee n_2$$

$$f_6 = p_1$$

$$f_7 = p_2$$

$$f_8 = p_3$$

Як бачимо, завдяки модифікації алгоритму, граф зменшився на 1 ребро, що зробило його більш простішим для дослідження.

4.2 Модифікація алгоритму для складних систем

Можливі такі випадки, коли кожна підсистема має свою унікальну, максимально допустиму кількість відмов. Для такого випадку можна використовувати наслідки з твердження:

- якщо система А має відмовостійкість m_a , то під час побудови моделі для всієї системи кожна така підсистема формує реберні функції як для $(m_a + 1)$ -відмовостійкої МВР моделі;

- для системи В- як для 1-відмовостійкої МВР моделі.

Наведемо декілька прикладів для складних систем:

Приклад 1

Є три системи $Z - K(3, 7)$, $X - K(2, 6)$, $Y - K(1, 4)$. Процесори системи Y можуть брати на себе роботу інших процесорів під час їх відмови.

Розділимо елементи систем Z та X на підмножини, отримаємо:

Для $Z - \{z_1, z_2, \dots, z_7\}$ на 4 підмножин $\{z_1, z_2\}$, $\{z_3, z_4\}$, $\{z_5, z_6\}$, $\{z_7\}$

Для $X - \{x_1, x_2, \dots, x_6\}$ на 3 підмножин $\{x_1, x_2\}$, $\{x_3, x_4\}$, $\{x_5, x_6\}$

Звичайний алгоритм:

Побудуємо таблицю розподіли відмов по підмножинам та напишемо реберні функції для систем Z та X :

Таблиця 4.2.1

$\{z_1, z_2\}$	$\{z_3, z_4\}$	$\{z_5, z_6\}$	$\{z_7\}$	
	$K(1,2)$	$K(1,2)$	$K(1,1)$	$f_1 = z_3 z_4 \vee z_5 z_6 \vee z_7$
$K(1,2)$	$K(1,2)$		$K(1,1)$	$f_2 = z_1 z_2 \vee z_3 z_4 \vee z_7$
$K(1,2)$		$K(1,2)$	$K(1,1)$	$f_3 = z_1 z_2 \vee z_3 z_4 \vee z_7$
$K(1,2)$	$K(1,2)$	$K(1,2)$		$f_4 = z_1 z_2 \vee z_3 z_4 \vee z_5 z_6$
$K(2,2)$	$K(1,2)$			$f_5 = z_1 \vee z_2 \vee z_3 z_4$
$K(2,2)$		$K(1,2)$		$f_6 = z_1 \vee z_2 \vee z_5 z_6$
$K(2,2)$			$K(1,1)$	$f_7 = z_1 \vee z_2 \vee z_7$
$K(1,2)$	$K(2,2)$			$f_8 = z_3 \vee z_4 \vee z_1 z_2$
	$K(2,2)$	$K(1,2)$		$f_9 = z_3 \vee z_4 \vee z_5 z_6$
	$K(2,2)$		$K(1,1)$	$f_{10} = z_3 \vee z_4 \vee z_7$
$K(1,2)$		$K(2,2)$		$f_{11} = z_5 \vee z_6 \vee z_1 z_2$

	$K(1,2)$	$K(2,2)$		$f_{12} = z_5 \vee z_6 \vee z_3 z_4$
		$K(2,2)$	$K(1,1)$	$f_{13} = z_5 \vee z_6 \vee z_7$

Таблиця 4.2.2

$\{x_1, x_2\}$	$\{x_3, x_4\}$	$\{x_5, x_6\}$	
	$K(1,2)$	$K(1,2)$	$f_{14} = x_3 x_4 \vee x_5 x_6$
$K(1,2)$	$K(1,2)$		$f_{15} = x_1 x_2 \vee x_3 x_4$
$K(1,2)$		$K(1,2)$	$f_{16} = x_1 x_2 \vee x_5 x_6$
$K(2,2)$			$f_{17} = x_1 \vee x_2$
	$K(2,2)$		$f_{18} = x_3 \vee x_4$
		$K(2,2)$	$f_{19} = x_5 \vee x_6$

Отримаємо такі функції в ребрах графа:

$$f_1 = z_3 z_4 \vee z_5 z_6 \vee z_7$$

$$f_2 = z_1 z_2 \vee z_3 z_4 \vee z_7$$

$$f_3 = z_1 z_2 \vee z_3 z_4 \vee z_7$$

$$f_4 = z_1 z_2 \vee z_3 z_4 \vee z_5 z_6$$

$$f_5 = z_1 \vee z_2 \vee z_3 z_4$$

$$f_6 = z_1 \vee z_2 \vee z_5 z_6$$

$$f_7 = z_1 \vee z_2 \vee z_7$$

$$f_8 = z_3 \vee z_4 \vee z_1 z_2$$

$$f_9 = z_3 \vee z_4 \vee z_5 z_6$$

$$f_{10} = z_3 \vee z_4 \vee z_7$$

$$f_{11} = z_5 \vee z_6 \vee z_1 z_2$$

$$f_{12} = z_5 \vee z_6 \vee z_3 z_4$$

$$f_{13} = z_5 \vee z_6 \vee z_7$$

$$f_{14} = x_3 x_4 \vee x_5 x_6$$

$$f_{15} = x_1 x_2 \vee x_3 x_4$$

$$f_{16} = x_1 x_2 \vee x_5 x_6$$

$$f_{17} = x_1 \vee x_2$$

$$f_{18} = x_3 \vee x_4$$

$$f_{19} = x_5 \vee x_6$$

$$f_{20} = y_1$$

$$f_{21} = y_2$$

$$f_{22} = y_3$$

$$f_{23} = y_4$$

Отже, в нас вийшов 23-реберний кільцевий граф.

Можна також розбити система на інші підмножини, наприклад:

Для Z - $\{z_1, z_2, \dots, z_7\}$ на 2 підмножин $\{z_1, z_2, z_3, z_4\}$, $\{z_5, z_6, z_7\}$

Для X - $\{x_1, x_2, \dots, x_6\}$ на 2 підмножин $\{x_1, x_2, x_3\}$, $\{x_4, x_5, x_6\}$

Побудуємо таблицю розподіли відмов по підмножинам та напишемо реберні функції для систем Z та X :

Для системи Z :

Таблиця 4.2.3

$\{z_1, z_2, z_3, z_4\}$	$\{z_5, z_6, z_7\}$
$K(3,4)$	
$K(2,4)$	$K(1,3)$
$K(1,4)$	$K(2,3)$
	$K(3,3)$

Побудуємо таблиці для розбиття підсистем на системи виду $K(1, 2)$

Таблиця 4.2.4

$\{z_1, z_2\}$	$\{z_3, z_4\}$
$K(2,2)$	$K(1,2)$
$K(1,2)$	$K(2,2)$

Таблиця 4.2.5

$\{z_1, z_2\}$	$\{z_3, z_4\}$
$K(2,2)$	
$K(1,2)$	$K(1,2)$
	$K(2,2)$

Таблиця 4.2.5

$\{z_5, z_6\}$	$\{z_7\}$
$K(2,2)$	
$K(1,2)$	$K(1,1)$

Для системи X:

Таблиця 4.2.6

$\{x_1, x_2, x_3\}$	$\{x_4, x_5, x_6\}$
$K(2,3)$	
$K(1,3)$	$K(1,3)$
	$K(2,3)$

Побудуємо таблиці для розбиття підсистем на системи виду $K(1, 2)$

Таблиця 4.2.7

$\{x_1, x_2\}$	$\{x_3\}$
$K(2,2)$	
$K(1,2)$	$K(1,1)$

Таблиця 4.2.8

$\{x_1, x_2\}$	$\{x_3\}$
$K(1,2)$	
	$K(1,1)$

Таблиця 4.2.7

$\{x_4, x_5\}$	$\{x_6\}$
$K(2,2)$	
$K(1,2)$	$K(1,1)$

Таблиця 4.2.8

$\{x_4, x_5\}$	$\{x_6\}$
$K(1,2)$	
	$K(1,1)$

Побудуємо ребра графа:

$$f_1 = z_1 \vee z_2 \vee z_3 z_4$$

$$f_2 = z_3 \vee z_4 \vee z_1 z_2$$

$$f_3 = z_1 \vee z_2 \vee z_5 z_6 z_7$$

$$f_4 = z_1 z_2 \vee z_3 z_4 \vee z_5 z_6 z_7$$

$$f_5 = z_3 \vee z_4 \vee z_5 z_6 z_7$$

$$f_6 = z_5 \vee z_6 \vee z_1 z_2 z_3 z_4$$

$$f_7 = z_5 z_6 \vee z_7 \vee z_1 z_2 z_3 z_4$$

$$f_8 = z_5 \vee z_6 \vee z_7$$

$$f_9 = x_1 \vee x_2$$

$$f_{10} = x_1 x_2 \vee x_3$$

$$f_{11} = x_1 x_2 \vee x_4 x_5$$

$$f_{12} = x_1 x_2 \vee x_6$$

$$f_{13} = x_3 \vee x_4 x_5$$

$$f_{14} = x_3 \vee x_6$$

$$f_{15} = x_4 \vee x_5$$

$$f_{16} = x_4 x_5 \vee x_6$$

$$f_{17} = y_1$$

$$f_{18} = y_2$$

$$f_{19} = y_3$$

$$f_{20} = y_4$$

Отримали 20-реберний граф

Модифікований алгоритм

Варіант розбиття одразу на множини з двома елементами:

Побудуємо таблицю розподіли відмов по підмножинам та напишемо реберні функції для систем Z та X :

Таблиця 4.2.9

$\{z_1, z_2\}$	$\{z_3, z_4\}$	$\{z_5, z_6\}$	$\{z_7\}$	
$K(2,2)$	$K(2,2)$			$f_1 = z_1 \vee z_2 \vee z_3 \vee z_4$
$K(2,2)$		$K(2,2)$		$f_2 = z_1 \vee z_2 \vee z_5 \vee z_6$

	$K(2,2)$	$K(2,2)$		$f_3 = z_3 \vee z_4 \vee z_5 \vee z_6$
$K(2,2)$	$K(1,2)$	$K(1,2)$		$f_4 = z_1 \vee z_2 \vee z_3 z_4 \vee z_5 z_6$
$K(2,2)$	$K(1,2)$		$K(1,1)$	$f_5 = z_1 \vee z_2 \vee z_3 z_4 \vee z_7$
$K(2,2)$		$K(1,2)$	$K(1,1)$	$f_6 = z_1 \vee z_2 \vee z_5 z_6 \vee z_7$
$K(1,2)$	$K(2,2)$		$K(1,1)$	$f_7 = z_1 z_2 \vee z_3 \vee z_4 \vee z_7$
	$K(2,2)$	$K(1,2)$	$K(1,1)$	$f_8 = z_3 \vee z_4 \vee z_5 z_6 \vee z_7$
	$K(1,2)$	$K(2,2)$	$K(1,1)$	$f_9 = z_3 z_4 \vee z_5 \vee z_6 \vee z_7$
$K(1,2)$		$K(2,2)$	$K(1,1)$	$f_{10} = z_1 z_2 \vee z_3 \vee z_4 \vee z_7$
$K(1,2)$	$K(1,2)$	$K(1,2)$	$K(1,1)$	$f_{11} = z_1 z_2 \vee z_3 z_4 \vee z_5 z_6 \vee z_7$

Таблиця 4.2.10

$\{x_1, x_2\}$	$\{x_3, x_4\}$	$\{x_5, x_6\}$	
$K(1,2)$	$K(1,2)$	$K(1,2)$	$f_{12} = x_1 x_2 \vee x_3 x_4 \vee x_5 x_6$
$K(2,2)$	$K(1,2)$		$f_{13} = x_1 \vee x_2 \vee x_3 x_4$
$K(2,2)$		$K(1,2)$	$f_{14} = x_1 \vee x_2 \vee x_5 x_6$
$K(1,2)$	$K(2,2)$		$f_{15} = x_1 x_2 \vee x_3 \vee x_4$
	$K(2,2)$	$K(1,2)$	$f_{16} = x_3 \vee x_4 \vee x_5 x_6$
	$K(1,2)$	$K(2,2)$	$f_{17} = x_3 x_4 \vee x_5 \vee x_6$
$K(1,2)$		$K(2,2)$	$f_{18} = x_1 x_2 \vee x_5 \vee x_6$

Отримаємо такі функції в ребрах графа:

$$f_1 = z_1 \vee z_2 \vee z_3 \vee z_4$$

$$f_2 = z_1 \vee z_2 \vee z_5 \vee z_6$$

$$f_3 = z_3 \vee z_4 \vee z_5 \vee z_6$$

$$f_4 = z_1 \vee z_2 \vee z_3 z_4 \vee z_5 z_6$$

$$f_5 = z_1 \vee z_2 \vee z_3 z_4 \vee z_7$$

$$f_6 = z_1 \vee z_2 \vee z_5 z_6 \vee z_7$$

$$f_7 = z_1 z_2 \vee z_3 \vee z_4 \vee z_7$$

$$f_8 = z_3 \vee z_4 \vee z_5 z_6 \vee z_7$$

$$f_9 = z_3 z_4 \vee z_5 \vee z_6 \vee z_7$$

$$f_{10} = z_1 z_2 \vee z_3 \vee z_4 \vee z_7$$

$$f_{11} = z_1 z_2 \vee z_3 z_4 \vee z_5 z_6 \vee z_7$$

$$f_{12} = x_1 x_2 \vee x_3 x_4 \vee x_5 x_6$$

$$f_{13} = x_1 \vee x_2 \vee x_3 x_4$$

$$f_{14} = x_1 \vee x_2 \vee x_5 x_6$$

$$f_{15} = x_1 x_2 \vee x_3 \vee x_4$$

$$f_{16} = x_3 \vee x_4 \vee x_5 x_6$$

$$f_{17} = x_3 x_4 \vee x_5 \vee x_6$$

$$f_{18} = x_1 x_2 \vee x_5 \vee x_6$$

$$f_{19} = y_1$$

$$f_{20} = y_2$$

$$f_{21} = y_3$$

$$f_{22} = y_4$$

Отже, в нас вийшов 22-реберний кільцевий граф.

Варіант розбиття на множини з більше ніж двома елементами:

Для системи Z:

Таблиця 4.2.11

$\{z_1, z_2, z_3, z_4\}$	$\{z_5, z_6, z_7\}$
$K(4,4)$	
$K(3,4)$	$K(1,3)$
$K(2,4)$	$K(2,3)$
$K(1,3)$	$K(3,3)$

Побудуємо таблиці для розбиття підсистем на системи виду $K(1, 2)$

Таблиця 4.2.12

$\{z_1, z_2\}$	$\{z_3, z_4\}$
$K(2,2)$	$K(1,2)$
$K(1,2)$	$K(2,2)$

Таблиця 4.2.13

$\{z_1, z_2\}$	$\{z_3, z_4\}$
$K(2,2)$	
$K(1,2)$	$K(1,2)$
	$K(2,2)$

Таблиця 4.2.14

$\{z_5, z_6\}$	$\{z_7\}$
$K(2,2)$	
$K(1,2)$	$K(1,1)$

Для системи X:

Таблиця 4.2.15

$\{x_1, x_2, x_3\}$	$\{x_4, x_5, x_6\}$
$K(3,3)$	
$K(2,3)$	$K(1,3)$
$K(1,3)$	$K(2,3)$

	$K(3,3)$
--	----------

Побудуємо таблиці для розбиття підсистем на системи виду $K(1, 2)$

Таблиця 4.2.16

$\{x_1, x_2\}$	$\{x_3\}$
$K(2,2)$	
$K(1,2)$	$K(1,1)$

Таблиця 4.2.17

$\{x_1, x_2\}$	$\{x_3\}$
$K(1,2)$	
	$K(1,1)$

Таблиця 4.2.18

$\{x_4, x_5\}$	$\{x_6\}$
$K(2,2)$	
$K(1,2)$	$K(1,1)$

Таблиця 4.2.19

$\{x_4, x_5\}$	$\{x_6\}$
$K(1,2)$	
	$K(1,1)$

$$f_1 = z_1 \vee z_2 \vee z_3 \vee z_4$$

$$f_2 = z_1 \vee z_2 \vee z_3 z_4 \vee z_5 z_6 z_7$$

$$f_3 = z_1 z_2 \vee z_3 \vee z_4 \vee z_5 z_6 z_7$$

$$f_4 = z_1 \vee z_2 \vee z_5 \vee z_6$$

$$f_5 = z_1 \vee z_2 \vee z_5 z_6 \vee z_7$$

$$f_6 = z_1 z_2 \vee z_3 z_4 \vee z_5 \vee z_6$$

$$f_7 = z_1 z_2 \vee z_3 z_4 \vee z_5 z_6 \vee z_7$$

$$f_8 = z_3 \vee z_4 \vee z_5 \vee z_6$$

$$f_9 = z_3 \vee z_4 \vee z_5 z_6 \vee z_7$$

$$f_{10} = x_1 \vee x_2 \vee x_3$$

$$f_{11} = x_1 \vee x_2 \vee x_4 x_5 x_6$$

$$f_{12} = x_1 x_2 \vee x_3 \vee x_4 x_5 x_6$$

$$f_{13} = x_4 \vee x_5 \vee x_1 x_2 x_3$$

$$f_{14} = x_4 x_5 \vee x_6 \vee x_1 x_2 x_3$$

$$f_{15} = x_4 \vee x_5 \vee x_6$$

$$f_{16} = y_1$$

$$f_{17} = y_2$$

$$f_{18} = y_3$$

$$f_{19} = y_4$$

Отримали 19-реберний кільцевий граф.

Висновки до розділу 4

У цьому розділі наведено модифікований алгоритм будування GL-моделей для складних систем, де певні процесори можуть брати на себе роботу інших, під час їх відмов, але не навпаки. Наведенні приклади будування GL-моделей використовуючи звичайний та модифікований алгоритми

Висновки

У магістерській дисертації на основі аналізу загального методу формування GL-моделей представлена його модифікація для систем, де певні процесори можуть брати на себе роботу інших, під час їх відмов, але не навпаки, шляхом заміни алгоритму формування моделі для основної підсистеми. Розроблена модифікація дозволяє знизити кількість ребр графа GL-моделі як мінімум на одне ребро, що робить дослідження такого графа простішим.

У першому розділі описано властивості і застосування відмовостійких багатопроцесорних систем у сучасному світі. Показані головні характеристики надійності системи. Приведені терміни теорії надійності.

В другому розділі магістерської роботи показано значення графів в підрахунку надійності системи. Показані двополюсні графи та їх застосування. Описано алгоритм будування GL-моделей для базових систем різної складності. Наведені приклади використання цього алгоритму.

Третій розділ присвячений відомим методам підрахунку надійності складних систем. Показані методи Рушді, Барлоу і Хайдменна та метод мінімальних наборів шляхів і перерізів.

В четвертому розділі описана модифікація існуючого алгоритму, для систем, де певні процесори можуть брати на себе роботу інших під час їх відмови, але не навпаки. Наведені порівняльні приклади будування GL-моделей використовуючи старий і модифікований методи.

Отже, було зроблено модифікацію алгоритму будування GL-моделей для складних систем, яка дозволяє будувати графи з меншою кількістю ребер.

СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. Каравай М.Д., Согомонян Е.С., Шагаев И.В. Відмовостійкі обчислювальні системи // Підсумки науки і техніки. Сер. Технічна кібернетика.- 1989.- Т. 28.- С. 77-118.
2. Романкевич А.М., Карачун Л.Ф., Романкевич В.А. Графо-логічні моделі для аналізу складних відмовостійких обчислювальних систем // Електронне моделювання.— 2001.—т.23, №1.— с. 102-111.
3. Романкевич В.А. Про одну модель поведінки відмовостійкої багатопроцесорної системи // Радіoeлектроніка та інформатика.-1999.- №1.- С.75-76.
4. Романкевич О.М., Карачун Л.Ф., Романкевич В.О. До питання побудова моделі поведінки багатомодульних систем // Наукові вісті НТУУ "КПІ". - 1998. - №1. - с. 38-40.
5. А.М. Романкевич, В.В. Іванов, В.А Романкевич. Аналіз відмов багатомодульних систем зі складним розподілом відмов на основі циклічних GL-моделей // Електронне моделювання.-№5, т.26, 2004,с.67-81.
6. Романкевич О.М, Романкевич В.О., Кононова А.А., Рабах Ал Шбул Про деякі особливості GL-моделей $K(2, n)$ // Вісник НТУУ "КПІ" .- Інформатика, управління та ОТ.- 2004. №41.-С.85-92.
7. Романкевич О.М., Романкевич В.О., Богуславський О.В., Ал Шбул Рабах. Аналіз відмовостійких багатопроцесорних систем на основі графо-логічних моделей нециклічного типу // Вісник технологічного університету Поділля. - 2002. - Т.2, №3. - с. 30-33.
8. Романкевич В.А., Потапова Є.Р., Ал Шбул Рабах Об'єднання моделей підсистем в рамках GL-моделей // Інформаційно-керуючі системи на залізничному транспорті. - 2002.- №4,5.- С. 22.
9. Додонов О.Г., Кузнецова М.Г., Горбачек Е.С. Введення в теорію живучості

обчислювальних систем.- К .: Наукова думка, 1990.- 182с

- 10.Березюк Н.Т., Гапунін А.Я., Підлісний Н.І. Живучість мікропроцесорних систем управління.- К .: Техніка, 1989.- 144 с.
- 11.Lee Y. H., Shin K. G. Design and evaluation of a fault-tolerant multiprocessor, using hardware recovery block // IEEE Trans. Comput.- 1984.- Vol.C-33, №2.- P.113-124.
- 12.Anderson T., Lee P.A. Fault Tolerance Terminology Proposals // IEEE.- 1982.-№2.- P.29-32.
13. Прикладна теорія цифрових автоматів / Самофалов К.Г., 14.Романкевич А.М., Валуйський В.М., Канівський Ю.С., Піневича М.М.
К .: Вища школа, 1987, 375с.
- 15.Salam Salloum, Melvin A. Breuer. Fast Optimal Diagnosis Procedures for k-out-of-n: G Systems // IEEE Trans. on Reliability.- 1997.-Vol.46, №2.-P.283-290.
- 16.Chen R.W., Hwang F.K., Li Wen-Ching. Consecutive 2-out-of-n: F systems with node and link failures // IEEE Trans. on Reliability.- 1993.-Vol.42, №3.- P.497-502.
- 17.Іуду К.А. Надійність контроль і діагностика обчислювальних машин і систем // М .: Вища школа - 1989, с.4 - 39.
- 18.Бельдхарі Н., Болдак А.А., Чебаненко Т.М. Забезпечення надійності комп'ютерних систем. // ВЕК + Київ - 1998. - с.8 - 35
- 19.Пацюра І.В., Корнійчук В.І., Довбиш Л.В. Надійність електронних систем // СВІТ Київ - 1997, с.2 - 20.
20. Румшійскій Л.З. Елементи теорії ймовірностей // М .: Наука - 1976,21.240с.
- 22.Крайник А.В., Курдіков Б.А., Лебедєв О.М., Недосекин Д.Д., 23.Подобєд М.В., Полянська Т.І., Чернявський Е.А. Імовірнісні методи в обчислювальній техніці // М .: Вища школа - 1986, 312с.
- 24.Е.С. Вентцель Теорія ймовірностей // М .: Вища Школа - 2002 575с. 25.Кульбак С. Теорія інформації та статистика // М .: Наука - 1967 408с.
- 26.Way Kuo, Ming J. Zuo Optimal Reliability Modeling. Principles and Applications // John Wiley & Sons, Inc. - 2002 p.15 - 62, 85 - 140, 148 - 164, 231 - 281.

27. А. Кофман. Введения в прикладну комбінаторіку.- М.: Наука, 1975.- 479с.
28. Романкевич В.А., Морозов К.В., Фесенюк А.П. Об одном методе модификации рёберных функций GL-моделей // Радіоелектронні і комп'ютерні системи.-№6, 2014.- С.95-99
29. Романкевич В.А., Малышева М.О., Примаков И.К. Об одном способе формирования рёберных функций GL-модели // Системный анализ и информационные технологии: материалы 18-й Международной научно-практической конференции SAIT 2016.- К.: УНК «ИПСА» НТУУ «КПИ», 2016.- С.406
30. Морозов К.В., Романкевич А.М., Романкевич В.А. О характере влияния модификации рёберных функций GL-модели на её поведение в потоке отказов // Радіоелектронні і комп'ютерні системи.-№6, 2016.- С.108-112
31. Морозов К.В., Романкевич В.А., Сапсай Т.Г. О модификации графо- логической модели для систем и их компонентов с множественными состояниями // Системный анализ и информационные технологии: материалы 19-й Международной научно-практической конференции SAIT 2017.- К.: УНК «ИПСА» КПИ им. Игоря Сикорского, 2017.- С.94
32. Романкевич В.О. Методи і засоби оцінки технічних характеристик гарантоздатності відмовостійких багатопроцесорних систем управління складними об'єктами: Дис. на здобуття наукового ступеня доктора технічних наук: 05.13.05; - Захищена 29.01.2018; Затв. 20.03.2018.- К., 2017.- 388с.

