

Хмельницький національний університет
Факультет інформаційних технологій
Кафедра кібербезпеки

КВАЛІФІКАЦІЙНА РОБОТА МАГІСТРА

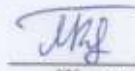
Метод організації тестувань цифрових схем на основі еволюційних алгоритмів
Назва теми

Галузь знань 12 – Інформаційні технології

Спеціальність 123 – Комп'ютерна інженерія

КРМКІ.016012.20.01.06 ПЗ

Виконав: студент 2 курсу, група КІ1м20-1


Підпис

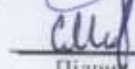
Моторний П.В.

Керівник доц., к. т. н, доцент кафедри Кб



Чешун В.М.

Нормоконтролер старший викладач кафедри Кб


Підпис

Мостовий С.В.

До захисту допускаю:

Зав. кафедри Кб, к.т.н., доц


Підпис

Ключ Ю.П.

3 12 2021 р.

ХМЕЛЬНИЦЬКИЙ НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ

Факультет ІНФОРМАЦІЙНИХ ТЕХНОЛОГІЙ

Кафедра КІБЕРБЕЗПЕКИ

Освітній рівень МАГІСТР

Галузь знань 12 ІНФОРМАЦІЙНІ ТЕХНОЛОГІЇ

Спеціальність 123 КОМП'ЮТЕРНА ІНЖЕНЕРІЯ

Освітня програма ПРОГРАМУВАННЯ ТА ЗАХИСТ КОМП'ЮТЕРНИХ СИСТЕМ І МЕРЕЖ

ЗАТВЕРДЖУЮ

Зав. кафедри Ю.П. Кльоц

“ 1 ” 09 2021 р.

**ЗАВДАННЯ
НА КВАЛІФІКАЦІЙНУ РОБОТУ**

Моторний Павло Володимирович

Прізвище, ім'я, по батькові студента

1. Тема проекту (роботи) Метод організації тестувань цифрових схем на основі еволюційних алгоритмів

Керівник роботи Чешун Віктор Миколайович

Прізвище, ім'я, по батькові, науковий ступінь, вчене звання
кандидат технічних наук, доцент

Затверджена наказом № 118 ректора університету додаток №23 від 01.09.2021

2. Строк подання студентом проекту (роботи) на кафедру 20.11.2021

3. Вихідні дані до проекту (роботи) Удосконалена організації тестувань цифрових схем на основі еволюційних алгоритмів.


4. Зміст пояснювальної записки (перелік питань, які потрібно розробити) _____

Вступ. Способи організації тестувань цифрових схем на основі еволюційних алгоритмів. Математичну модель опису методу. Методи реалізації алгоритму. Оцінка результатів основаних на даних. Висновки.

5. Перелік графічного матеріалу (із зазначенням обов'язкових креслень) Тема, мета магістерської, наукова новизна, практична значення, публікації. Організації тестувань цифрових схем на основі еволюційного алгоритму. Розробка математичної моделі методу.

Реалізація алгоритму на мові програмування #. Узагальнена схема роботи еволюційного алгоритму. Висновки.

6. Консультанти розділів кваліфікаційної роботи

Розділ	Прізвище, ініціали і посада консультанта	Підпис, дата	
		завдання видав	завдання прийняв
Нормоконтроль	Мостовий С.В. Старший викладач кафедри Кб		

7. Дата видачі завдання «_» _____ 2021р.

КАЛЕНДАРНИЙ ПЛАН

№з/п	Назва етапів (розділів) кваліфікаційної роботи	Термін виконання етапів роботи	Примітка
1	Вибір напрямку дослідження і узгодження тематики КРМ з керівником	1.09.2021	
2	Ознайомлення з предметною областю; формулювання мети і задач дослідження; визначення об'єкта і предмета дослідження	2.09.2021	
3	Робота над розділом 1 – аналіз відомих моделей, методів за темою; постановка задачі	6.09.2021	
4	Робота над розділом 2 – розробка моделей і методів для вирішення поставленої задачі	22.09.2021	
5	Робота над науковою публікацією	2.10.2021	
6	Робота над розділом 3 – розробка алгоритмів і технологій, їх аналіз	15.10.2021	
7	Робота над розділом 4 – апробація запропонованих рішень	29.10.2021	
8	Узгодження отриманих; оформлення пояснювальної записки згідно вимог	5.11.2021	
9	Попередній захист роботи	16.11.2021	
10	Захист роботи на засіданні ЕК	6.12.2021	

Студент



Мотомкий П.В.

Підпис

Ініціали, прізвище

Керівник проекту (роботи)



Челішкі В.М.

Ініціали, прізвище

АНОТАЦІЯ

Тема кваліфікаційної роботи: Метод організації тестувань цифрових схем на основі еволюційних алгоритмів

Автор роботи: Моторний Павло Володимирович

Керівник роботи: к.т.н., доц. Чешун Віктор Миколайович

Загальний обсяг роботи: 96 сторінок, 13 рисунків, 3 таблиці, 3 додатків, 61 посилань.

Ключові слова: еволюційний алгоритм, методи організації тестувань, утворення нового покоління, математична модель.

Метою кваліфікаційної роботи є створення нового методу організації тестувань цифрових схем на основі еволюційного алгоритму.

Дана кваліфікаційна робота присвячена підвищенню ефективності організації тестувань цифрових вузлів скороченням розмірностей використовуваних тестових послідовностей на основі застосування еволюційних алгоритмів на етапі організації тестувань цифрових схем.

Дата 3.12.2021

Підпис студента

ANNOTATION

Theme of qualification work: Method of organizing testing of digital circuits based on evolutionary algorithms

Author of the work: Motornyi Pavlo Volodymyrovych

Mentor: Ph.D. Cheshun Victor Nikolaevich


Total volume of work: 96 pages, 13 figures, 3 tables, 3 appendices, 61 links.

Keywords: evolutionary algorithm, methods of organization of tests, formation of new generation, mathematical model.

The purpose of the qualification work is to create a new method of organizing testing of digital circuits based on an evolutionary algorithm.

This qualification work is devoted to increasing the efficiency of the organization of testing of digital nodes by reducing the dimensions of the used test sequences based on the use of evolutionary algorithms at the stage of organizing the testing of digital circuits.

Date 3.12.2021

Student's signature 

ЗМІСТ

ВСТУП.....	3
1 ДОСЛІДЖЕННЯ СПОСОБІВ ОРГАНІЗАЦІЇ ТЕСТУВАНЬ ЦИФРОВИХ СХЕМ НА ОСНОВІ ЕВОЛЮЦІЙНИХ АЛГОРИТМІВ.....	7
1.1 Особливості діагностичних випробувань цифрових схем.....	7
1.2 Еволюційні алгоритми як інструментарій для організації тестування цифрових схем.....	14
1.3 Постановка задачі.....	35
1.4 Висновки.....	36
2 МАТЕМАТИЧНА МОДЕЛЬ МЕТОДУ ОРГАНІЗАЦІЇ ТЕСТУВАНЬ ЦИФРОВИХ СХЕМ НА ОСНОВІ ЕВОЛЮЦІЙНИХ АЛГОРИТМІВ.....	37
2.1 Визначення принципів математичного апарату в поєднанні технічної діагностики і еволюційних алгоритмів.....	37
2.2 Розробка математичної моделі методу.....	44
2.3 Висновки.....	51
3 ОПИС МЕТОДУ ОРГАНІЗАЦІЇ ТЕСТУВАНЬ ЦИФРОВИХ СХЕМ НА ОСНОВІ ЕВОЛЮЦІЙНИХ АЛГОРИТМІВ.....	52
3.1 Опис основних положень методу.....	52
3.2 Опис алгоритмів реалізації методу.....	60
3.3 Висновки.....	71
4 ПРАКТИЧНИЙ МЕТОД ТЕСТУВАНЬ ЦИФРОВИХ СХЕМ НА ОСНОВІ ЕВОЛЮЦІЙНИХ АЛГОРИТМІВ	72
4.1 Приклад практичного застосування методу тестувань цифрових схем.....	72
4.2 Висновки.....	86
ВИСНОВКИ.....	88
ПЕРЕЛІК ДЖЕРЕЛ ПОСИЛАННЯ.....	90
ДОДАТКИ.....	97
ДОДАТОК А - КОПІЇ ПУБЛІКАЦІЙ.....	97
ДОДАТОК Б - ПРЕЗЕНТАЦІЯ РОБОТИ.....	100
ДОДАТОК В - ЛІСТИНГ ПРОГРАМИ РЕАЛІЗАЦІЇ МЕТОДУ ОРГАНІЗАЦІЇ ТЕСТУВАНЬ ЦИФРОВИХ СХЕМ НА ОСНОВІ ЕВОЛЮЦІЙНИХ АЛГОРИТМІВ.....	109

ВСТУП

Актуальність дослідження. Швидкий розвиток і постійне вдосконалення технологій виготовлення цифрових інтегральних компонентів надають останнім зростаючі можливості і постійно розширюють сферу їхнього застосування, що призводить до появи нових і нових різновидів цифрових пристроїв і систем. Функціональна різноманітність та висока продуктивність новітніх цифрових пристроїв і систем в поєднанні зі зниженням вартості дозволяють використовувати їх при вирішенні питань різної складності від елементарного побутового до комплексного керування технологічними процесами та обладнанням, збої в роботі яких можуть бути небезпечними для життя [1].

Не останню роль у широкому розповсюдженні цифрових пристроїв та систем відіграють одні із високих показники їх надійності, що собою забезпечуються проведенням ряду системних заходів у данній сфері. До числа засобів надійності повинні включати верифікацію, моніторинг та діагностику на етапах життєвого циклу елементної бази та електронних продуктів [2-4].

Ускладнення завдань технічної діагностики призвело до створення різноманітних спеціалізованих засобів з використанням елементів інтелектуальної обробки діагностичної інформації, зокрема можна виділити засоби нейромережевої діагностики, систем на основі розмитої логіки, експертних діагностичних системи тощо [5-7]. Представлені засоби мають різну функціональність і призначення та використовуються зокрема для вирішення різного роду проблем, що виникають при технічній діагностиці, але основною метою їх використання завжди була і є підвищення ефективності тестів для діагностики і можливості уникнути ролі людського фактору як основної передумови виникнення багатьох можливих помилок.

Використання генетичних алгоритмів являється визначеним як

перспективний напрямок в інтелектуалізації обробки діагностичної інформації. Практика застосування еволюційних алгоритмів у вирішенні завдань технічної діагностики не є новою, в роботах [8-11] єдеться про дослідження в даному напрямку, що проводилися в далеких 90-х роках ХХ сторіччя. Серед основних застосувань еволюційних алгоритмів в проблемах технічної діагностики є процес оптимізації послідовностей і алгоритмів що на основі діагностування, де еволюційні алгоритми представлені як альтернатива традиційним методам оптимізації, серед яких є метод динамічного програмування, метод гілок та меж тощо [1, 11-14]. Існують також певні застосування еволюційних алгоритмів для вилучення зайвих тестів з їх послідовностей [15]. Відомі роботи, де еволюційні алгоритми використовуються для конкретного вибору контрольних точок з об'єкта діагностування (ОД), які використовуються для виконання тестування, а також для вивчення та модифікації ОД, зокрема, за допомогою ROBDD-графів [16,17].

В умовах постійного зростання інтегральної складності цифрових електронних компонентів залишається проблема організації тестувань, для вирішення якої також може ефективно застосовуватись теорія еволюційних алгоритмів [12].

Задача організації тестів для діагностування цифрових схем з елементами високого порогу інтеграції ніколи не втрачала своєї актуальності і навпаки, актуальність поставленої задачі завжди посилювалась з розробкою нових інтегральних компонентів (схем) та з технологіями їх виготовлення.

Еволюційні алгоритми належать до категорії пошукових процедур, що виникли як ще одна спроба копіювання процесів наслідування (природні) та природного відбору (селекції) як основних діючих факторів еволюційного розвитку. Еволюційні алгоритми використовувались в багатьох різних сферах, де могла б бути використана теорія еволюційного розвитку та відбору, що обумовлювалось проведенням попередніх

тестування та досліджень [12, 18].

Таким чином, ефективне використання еволюційних алгоритмів для організації тестування можливе лише тоді коли виявленні асоціативні аналогії у проведених досліджувальних процесах та теорії еволюційного вдосконалення, що ґрунтується на еволюційних алгоритмах, а також формулювання принципів еволюційних алгоритмів у завданнях технічної діагностики та розробки алгоритмів.

Дослідження присвячене питанням розробки методу організації тестувань цифрових схем на основі еволюційних алгоритмів.

Мета магістерської роботи полягає у ефективному підвищенні організації тестувань цифрових схем зменшення розмірностей використовуваних послідовностей за рахунок використання еволюційних алгоритмів на етапі організації тестувань.

Об'єктом дослідження є процес що забезпечує діагностування цифрових електронних схем.

Предметом дослідження є методи організації тестувань цифрових схем із збільшеною діагностичною здатністю.

Щоб реалізувати програму для досліджень необхідно:

а) виявити аналогії у вирішених задачах технічної діагностики та теорії еволюційних алгоритмів для збільшення ефективності організації тестування цифрових схем за рахунок зменшення розмірів використовуваних тестових послідовностей;

б) визначити основні тези методики організації тестування цифрових схем з урахуванням еволюційних алгоритмів;

в) розробити математичну модель щоб мати змогу реалізувати метод організації тестувань цифрових схем на основі еволюційних алгоритмів;

г) розробити для реалізації методу алгоритмічне забезпечення;

д) випробувати ефективність алгоритмічних рішень що були запропоновані.

В основі методів дослідження лежать основні положення технічної

діагностики, теорії тестової діагностики, теорії еволюційних методів, теорії множин, булевої алгебри та теорії можливостей.

Наукове нововведення отриманих результатів:

1. Роз'яснено математичну модель для діагностичного експерименту з урахуванням явної специфіки для вирішення задачі організації тестувань цифрових схем на основі еволюційних алгоритмів;

2. Розроблено метод організації тестувань цифрових схем збільшеної перевіряючої здатності, для забезпечення використання еволюційних алгоритмів щоб надати тестам властивість одночасного виконання кількох базових перевірок об'єкта діагностики..

Практична цінність методу полягає у здатності визначення принципів і розробці алгоритмів організації тестувань цифрових схем на основі еволюційних алгоритмів, що підвищують ефективність процесу діагностики цифрових схем за рахунок якого відбувається зменшення використовуваних тестових послідовностей.

Публікації. За темою магістерської роботи опубліковано 2 тези доповідей на конференції в Києві - військовий інститут університету Шевченко.

1 ДОСЛІДЖЕННЯ СПОСОБІВ ОРГАНІЗАЦІЇ ТЕСТУВАНЬ ЦИФРОВИХ СХЕМ НА ОСНОВІ ЕВОЛЮЦІЙНИХ АЛГОРИТМІВ

1.1 Особливості діагностичних випробувань цифрових схем

Удосконалення технологій виготовлення цифрових схем надає їм все більше можливостей і постійно збільшує сферу діяльності, що призводить до появи все нових і нових різних видів електронних пристроїв і систем. Більшу роль у широкому застосуванні цифрових схем відіграють високі показники їх надійності, які забезпечуються, зокрема, виконанням комплексу системних заходів, серед яких організація діагностичних випробувань для виявлення несправностей(дефектів) та важлива помилка в дизайні. [2,4].

Досягнення у підвищенні інтегральної складності цифрових компонентів та цифрових схем, реалізованих на їх основі, мають негативні наслідки для організації діагностичних тестів. Це породжує інтерес до автоматизації процесів організації тестів. У зв'язку з необхідністю врахування великої кількості факторів і параметрів, а також використання нетрадиційних підходів при реалізації тестів, найбільш актуальною на сьогоднішній день є розробка інтелектуальних методів, алгоритмів та засобів вирішення діагностичних завдань [2,3].

Зрештою, що не менш важливо, це пов'язано з різноманітністю сучасних технічних засобів, пристроїв та систем, що розглядаються як об'єкти діагностики.

Весь набір сучасних технічних засобів з урахуванням специфіки їх експлуатації можна розділити на два класи - неперервні та цифрові об'єкти, тому способи забезпечення виявлення дефектів у технічній діагностиці також діють на два класи:

- 1) методи та прийоми виявлення та забезпечення розрізнення дефектів технічних засобів безперервної дії;
- 2) методи та прийоми виявлення та розпізнавання дефектів об'єктів

цифрової дії..

Усі існуючі нині методи та методи виявлення та забезпечення фрагментації дефектів у безперервних об'єктах ґрунтуються на методах контролю допусків, тобто, якщо результат знаходиться в межах допустимих значень параметра, що вимірюється, визначених технічним завданням, інформація вважається достовірною. і відповідає значенню одиниці, інакше - нулю. Такий підхід дозволяє уявити будь-який пристрій безперервної дії у вигляді математичної моделі логічного типу незалежно від специфіки його функціонування. Вперше така модель описана в [9], а в роботі [10] надано доказ взаємно однозначної відповідності між досліджуваним об'єктом і моделлю..

В роботі [12] для організації діагностичних випробувань доповнено алгоритм пошуку дефектів цифрових схем на основі моделі у вигляді системи диференціальних рівнянь, доповнено процедурою аналізу структури керованої системи з урахуванням характеристики надійності. У статті показано, що витрати які пов'язані з повною відсутністю статистичної інформації про характеристики надійності елементів, за правило можуть бути незначними, якщо методи ймовірнісного оцінювання поєднуються з іншими методами з урахуванням специфіки системи, що діагностується. Алгоритм пошуку дефектів залежить від методу оцінки стану контрольованих параметрів..

У ряді робіт [13,14] методи розпізнавання зображень використовуються для організації цифрових діагностичних тестів та вирішення проблеми розрізнення. У роботі [13] для організації цифрових діагностичних тестів пропонується вибір контрольованих параметрів шляхом порівняння об'єкта діагностики (ОД) з деякою унікальною моделлю, метою якої є однозначний вибір, для технічного стану, класів.

В роботах [15,16] наведено навчальний алгоритм організації цифрових діагностичних тестів та вибору діагностичних параметрів, який дуже складний з погляду технічної реалізації та не гарантує оптимальний

вибір набору параметрів, що забезпечує однозначну відмінність. дефектів.

У роботах [17-20] в організацію діагностичних випробувань цифрових схем представлений алгоритм вибору діагностичних параметрів, заснований на методах математичного програмування. Коефіцієнти диференційних функцій визначаються методом лінійного програмування. При ідентифікації станів об'єкта аналізуються послідовно знаки коефіцієнтів функцій, а потім на підставі вирішального правила робляться висновки про відповідність стану об'єкта до певного класу.

У статтях [21, 22] описуються метод для оцінки стану об'єктів за допомогою порівняння реакції з реакцією еталонних еквівалентних моделей на ті ж самі дії при функціональній діагностиці. Порівнюючи перехідні характеристики об'єкта та еквівалентної моделі, можна визначити певне відхилення прямих показників щодо відхилень перехідних характеристик. Слід підкреслити, що отримання еталонної моделі становить значні труднощі.

В роботі [24] для організації діагностичних тестів як еталонна модель пропонується використовувати кінцевий автомат, який на кожному кроці видає сигнал, що відрізняється від значення вихідного сигналу UD на величину, зазначену в технічному стані контрольований об'єкт. Цей зв'язок між вихідними сигналами об'єкта та машини повинен підтримуватися для всього набору дійсних вхідних сигналів, що подаються об'єкту. Ця модель може використовуватись як для лінійних, так і для нелінійних об'єктів.

В основі організації діагностичних тестів та діагностики динамічних об'єктів лежать методи, що застосовуються в теорії ідентифікації [26]. За результатами моніторингу змінних об'єкта визначається конкретний клас його оператора, визначаються основні параметри, будується алгоритм розрахунку оцінок, встановлюється для отриманої математичної моделі об'єкта рівень адекватності.

Проблема розрізнення технічних станів об'єкта вирішується за допомогою спостереження вхідних та вихідних сигналів ВП та порівняння

їх із сигналами які були отриманими в моделі, що характеризує об'єкт за відсутності у ньому дефектів [27].

У роботі [28] вирішуються питання планування експериментів з використанням ідентифікації та діагностиці безперервних ланцюгів з метою визначенням контрольних точок та тестових вхідних сигналів, що забезпечують найкращий дозвіл параметрів системи.

Слід підкреслити, що використання аналітичних методів, що описують функціонування об'єктів системою диференціальних рівнянь для організації тестування цифрових схем при побудові моделей об'єктів, що діагностуються, застосовується до значно вузького класу систем. В основному це системи для автоматичного керування. В роботі [29] для організації цифрових випробувань та забезпечення фрагментації дефектів використовувани моделі [27,28] пропонується подати у вигляді полюсного графа, що враховує структуру об'єкта. Кожен із графіків описується поперечними та поздовжніми змінними, що відбивають фізичні властивості компонентів в аналітичних формах. Зв'язки між елементами графа описуються топологічними та компонентними рівняннями. Компонентні рівняння зазвичай пов'язують змінні окремих компонентів, а топологічні рівняння відбивають структуру об'єкта. Таким чином, виходить система рівнянь, що зв'язує вхідні та вихідні значення через коефіцієнти, що характеризують параметри компонентів. Цей метод відрізняється досить високою трудомісткістю, тому коефіцієнти, що характеризують параметри, розраховуються шляхом вирішення n^2 рівнянь, де n – це кількість компонентів об'єкта, а для підстановки діагнозу експериментально обчислюють вектори вхідних впливів та виконуються перевірки, кількість яких дорівнює кількості компонентів, які підлягають ідентифікації.

У роботі [30] для організації тестування цифрових схем та усунення дефектів внутрішніх компонентів оцінено вплив кожного із компонентів на вихідні критерії системи. За відсутності кореляції між компонентами

вихідні критерії системи визначаються регресійною моделлю з використанням гіперплощини першого порядку. Коли початкові функції системи відхиляються від необхідних, визначається багато гіпотетично дефектних компонентів. Аналіз кількох вхідних даних для різних регресійних моделей надає змогу отримати набір таких наборів і розрахувати математичні очікування та дисперсію відхилень для кожного компонента з підозрою на відмову. В якості дефектного компонента вибирається той, для якого дисперсія мінімальна. Слід зазначити, що процес створення регресійної моделі характеризується найвищою трудомісткістю, крім того, задача вирішується на основі припущення, що в системі є лише поодинокі дефекти.

У роботах [31,32] для організації тестування цифрових схем запропоновано методи побудови моделей діагностичних систем на основі виявлення причинно-наслідкових зв'язків між параметрами ОД. Модель будується за допомогою графа, кожна вершина якого пов'язана з тим чи іншим параметром діагностичної системи, а дуги графа представлені у вигляді функціональних чи статистичних залежностей між параметрами. Даний клас моделей характеризується імовірнісними категоріями та діагнозом, встановленим з певним ступенем достовірності.

В роботі [32] для організації тестування цифрових схем описана модель функціонування складної системи що представляє собою топологічний простір. Виділення моделі топологічної системи з топологічного простору є проблемою замикання множини внутрішніх властивостей системи. Побудова такої моделі ґрунтується лише на змістовному описі, тобто на наявній інформації про склад і способах дії ОД. Рішення діагностичних задач з точки зору виявлення несправного елемента на такій моделі носить досить імовірнісний характер, через це простір станів формується сукупністю структурних дефектів неоднозначно.

У роботі [33] описана методика визначення технічного стану у

багатопроцесорних обчислювальних систем для організації тестувань цифрових схем. Головна ідея полягає в наступному: на першому етапі аналізу проводиться ідентифікація на справному елементі процесора, на другому етапі за допомогою ідентифікованих справних елементів визначаються всі несправні та інші справні елементи, визначається із вихідної таблиці синдромів. Методика заснована на результатах тестування із сусідніми елементами процесора та побудові графа синдрому, вершини якого відповідають елементам процесора, а дуги – перевіркам. Основне обмеження перевантаження цього методу полягає в тому, що всі елементи моделі повинні мати певний набір елементів, які її тестують.

У [35,36] для організації тестувань розглядаються питання діагностики замкнутих ланцюгів, які є одними з найскладніших завдань технічного тестування. Традиційне функціонування таких об'єктів у перехідних і режимах, що встановилися, описувалося системними диференціальними рівняннями, що призводило до значних труднощій при розробці стратегії тестувань їх стану.

У [36] для організації тестування цифрових схем запропоновано метод побудови ненадлишкового набору елементарних перевірок, достатнього для побудови всіх тупикових тестів у замкнутих схемах. Вирішується задача визначення мінімальної кількості вузлів вимірювання що забезпечує збереження максимально можливої фрагментації дефектів у гілках схемах. Викладені в цій роботі ідеї базуються на припущеннях про єдність джерела та односпрямованості руху сигналів, що обмежує сферу їх використання. Крім того, виникають труднощі у вирішенні проблеми вибору малих діагностичних тестів з великою розмірністю контрольної матриці.

У роботі [37] для організації тестувань безперервних схем було запропоновано методику тестувань стану об'єктів, заснований на побудові математичної моделі у вигляді орієнтованого графа зі знаком. Об'єкт, що

описується за допомогою системи лінійних диференціальних рівнянь, пов'язаний з графом, вершини якого відповідають вимірювальним і невимірювальним змінним, а дуги - мають безпосередній вплив зі сторони змінних стану один на одного. Знаки «+» або «-» приписуються дугам графіка пов'язаних від знаків відповідних до них коефіцієнтів у вихідній системі рівнянь.

Задача з пошуку множинних відмов полягає у знаходженні мінімальної потужності безлічі вершин (кандидатів на відмову), з яких ймовірно існують узгоджені шляхи до кожної вимірної вершини графа, і вирішується у два етапи. На першому етапі формуються всі шляхи, що ведуть до вимірних вершин; на другому етапі ці шляхи аналізуються та виявляються кандидати на невдачу.

До недоліків цього методу можна віднести відсутність процедури визначення найбільш ймовірних кандидатів на невдачу з набору підозрюваних та необхідність тривалої попередньої методики формування шляхів при вимірюванні вершини.

У Роботах [38,39,40] описано процес розробки алгоритмів організації випробувань тестів, що дають змогу сформувати методику вибору найбільш ефективних, з метою забезпечення різноманітності дефектів, діагностичних параметрів.

На основі наведеного вище огляду літературних джерел можна зробити певний висновок, що відомі на сьогодні методи організації тестів, як правило, є об'єктно-орієнтованими, тобто розробляються кожного разу нові для кожного класу у системі. Для більшості методів проблема локалізації дефектів довільної кратності та вибору найкращого набору контрольованих параметрів тестувального об'єкта з точки зору різниці дефектів не вирішена в загальному вигляді.

Крім того, для реалізації відомих методів організації тестів виникають труднощі, пов'язані з великим обсягом обчислень. Розрахунки ускладнені через відсутність уніфікованої (єдиної) системи та інструментів

оцінки ефективності організації тестувань, обґрунтування чіткого вибору критеріїв, що є суттєвими для організації тестувальних тестів ОД, та забезпечення пріоритетів при виборі рішень. щодо використання певної опції.

Одним із найскладніших завдань при організації тестування цифрових схем є організація та оптимізація тестів, які будуть використовуватись для виявлення та розпізнавання (локалізації) проблем. Процес вимагає обробки величезних обсягів даних з урахуванням багатьох розглянутих критеріїв та параметрів, що призводить до надзвичайно складних та трудомістких розрахунків.

Складність проблеми ефективних тестів організації тестування цифрових схем та її параметризація з алгоритмізацією підвищують актуальність розробки нових методів організації тестування на основі еволюційних алгоритмів.

1.2 Еволюційні алгоритми як інструментарій для організації тестувань цифрових схем

Зростаюча складність завдань технічної діагностики зумовила створення різноманітних цільових засобів з елементами інтелектуальної обробки діагностичних даних, серед яких засоби нейромережевої діагностики, експертні діагностичні системи, системи на основі нечіткої логіки [5-7]. Певні інструменти мають різну функціональність та своє призначення та використовуються для вирішення різноманітних проблем, що виникають у технічній діагностиці, але метою їх застосування майже завжди є збільшення ефективності діагностичних тестів та зменшення людського фактора як основної передумови для виникнення більшості помилок. При цьому не втратили актуальності завдання організації тестів для тестувань цифрових схем з використанням елементів високого ступеня

інтеграції. Навпаки, актуальність цієї проблеми завжди посилювалася з появою нових цілісних компонентів і технологій їх виготовлення.

Використання еволюційних алгоритмів визнано перспективним напрямком інтелектуалізації обробки діагностичної інформації. В умовах постійного зростання інтегральної складності цифрових електронних компонентів залишається проблема організації тестувань, для вирішення якої також може ефективно застосовуватись теорія еволюційних алгоритмів [12].

Еволюційні алгоритми відносяться до категорії процедурного пошуку, які виникли на основі спроби копіювання природних процесів наслідування та природного відбору (селекції) як основних чинників еволюційного розвитку. Еволюційні алгоритми застосовувались і застосовуються в багатьох сферах, де можливе використання теорії еволюційного відбору та розвитку [12,18]. Еволюційна теорія запевняє, що кожен вид розвивається і змінюється, щоб пристосуватися до навколишнього середовища як найкраще.

Еволюційний підхід дає змогу аналізувати та адаптувати існуючу популяцію до нових умов середовища, що також скорочує час роботи алгоритму та забезпечує реалізацію принципу машинної адаптації та навчання. [50].

Загалом, еволюційний підхід використовується в цілих класах так відомих як еволюційних алгоритмів.

Еволюційні алгоритми – один із напрямів в штучному інтелекті, який використовує та моделює біологічну еволюцію [51].

Розрізняють різні види еволюційних алгоритмів, серед них [51]:

- еволюційне програмування;
- генетичні алгоритми;
- системи класифікаторів;
- еволюційні стратегії;
- генетичне програмування тощо.

Всі різні види еволюційних алгоритмів моделюють основні концепції в теорії біологічної еволюції такі як процеси відбору, відтворення чи мутації. Широко використовуються еволюційні алгоритми. Однією з найпоширеніших сфер застосування є комбінаторна оптимізація. Еволюційні алгоритми були успішно застосовані до класичних NP-повних задач, таких як проблема комівояжера, проблема упаковки рюкзака, розщеплення чисел і розфарбовування графіка.

Основні принципами реалізації еволюційні алгоритми можна поділити на основні три групи:

- еволюційне програмування - зосереджено більше на адаптації індивідів, ніж на еволюції генетичної інформації. Еволюційне програмування, як правило, передбачає безстатеве розмноження та мутацію, тобто внесення незначних змін до поточних рішень і методів розмноження на основі прямої конкуренції;

- еволюційні стратегії - їх особливість полягає у використанні самоадаптивних механізмів для управління процесом мутації. Ці механізми зосереджені не тільки на еволюції шуканих рішень, та на еволюції параметрів мутації;

- генетичні алгоритми – їх головною особливістю є використання оператора схрещування як основного механізму пошуку. Основний принцип теорії генетичних алгоритмів базується на припущенні, що частини оптимального рішення можна знайти незалежно і повторно об'єднати для отримання найкращого рішення.

Ідея генетичних алгоритмів була висловлена Дж. Холланд наприкінці шістдесятих років XX століття. Він був зацікавлений властивостями процесів природної еволюції (включаючи те що, що еволюціонують хромосоми, а чи не самі живі істоти). Холланд був переконаний у можливості складання та реалізації алгоритму у вигляді комп'ютерної програми, яка б вирішувала складні проблеми, як це робить природа, - шляхом еволюції. Через це він почав працювати над алгоритмами, що

працюють із послідовностей двійкових цифр (одиниць та нулів), званими хромосомами. Цей ранг алгоритмів використовувався для моделювання еволюційних процесів у поколіннях таких хромосом. У них реалізовані механізми відбору та відтворення, аналогічні тим, що використовуються у природній еволюції.

Термін «генетичний алгоритм» був вперше введений у 1975 р. Д. Голдбергом, який почав з розробки теорії еволюційного моделювання, запозичивши прийоми з теоретичних основ популяційної генетики. На сьогоднішній час генетичні алгоритми є унікальним обчислювальним засобом для розв'язування складних математичних задач з великою кількістю правильних рішень.

Генетичні алгоритми набули найбільшої популярності та широкого застосування при вирішенні великої кількості різних завдань. Існує кілька варіантів визначення генетичних алгоритмів, орієнтованих на різні особливості та сфери їх застосування:

– в роботі [52] генетичний алгоритм описується як алгоритм еволюційного пошуку, який використовується для вирішення питань оптимізації та моделювання шляхом послідовного вибору, комбінування та зміни необхідних параметрів з використанням механізмів, що імітують біологічну еволюцію;

– в роботі [53] дається основне визначення, де генетичний алгоритм – це алгоритм, який використовується для пошуку наближених розв'язків важкорозв'язних проблем шляхом застосування головних принципів еволюційної біології до сфери інформатики;

– в роботі [54] генетичний алгоритм розпізнається як метод, який відбиває природну еволюцію методів вирішення питань і, перш за все, задач оптимізації. Генетичні алгоритми зазвичай визначаються як методики пошуку, що ґрунтуються на механізмах природного відбору та успадкування, що використовуються еволюційним принципом - виживання найбільш пристосованих;

– в роботі [55] генетичні алгоритми уточнюються одним із сучасних методів штучного інтелекту, заснований на принципі природного відбору, запропонованому Ч. Дарвіном, і є одним із різновидів еволюційних обчислень.

В процесі роботи еволюційного алгоритму підтримується кілька допустимих рішень проблеми, яке називається генерацією. Робота алгоритму полягає у генерації наступного покоління. Для цього використовується відомі операції «схрещування» та «мутації». Як правило, операція схрещування полягає у побудові двох рішень одного або двох нових, а операція мутації - в невеликій зміні рішення.

Суть генетичного алгоритму базується у використанні біологічних процесів, що ґрунтуються у природньому відборі, для пошуку глобального екстремуму мультиекстремальної функції. Біологічні види розвиваються впродовж багатьох поколінь, дотримуючись законів природного відбору та принципу «виживає найбільш адаптована особина», сформульованому Чарльзом Дарвіном, і використовуючи спадкові механізми передачі кращих якостей однієї істоти з покоління до покоління. Як і біологічні системи, генетичні алгоритми можуть покращити вирішення реальних проблем, якщо вони правильно закодовані.

Еволюційні алгоритми можуть бути застосовані до широкого кола проблем, які можуть бути зведені до глобального чи локального пошуку й оптимізації рішення, або до моделювання змін у динамічному середовищі. Тому це досить стабільний і потужний інструмент, оскільки, незважаючи на те, що складність завдань постійно збільшується, а допустимий час на їх вирішення зменшується, системи на основі генетичних алгоритмів успішно справляються із завданнями.

Еволюційні алгоритми в сучасних умовах використовуються для пошуку рішень різноманітних питань [51, 56]:

- складання порядку рішення завдань.
- біоінформатика (згортання білків);

- штучне життя;
- апроксимація функцій;
- ігрові стратегії;
- створення розкладів;
- задачі компоновки;
- налаштування й навчання штучної нейронної мережі;
- різноманітні задачі на графах;
- оптимізація запитів в базах даних;
- знаходження коренів системи нелінійних рівнянь;
- комбінаторні задачі;
- навчання інтелектуальних моделей;
- задачі оптимізації параметрів;
- оптимізація динамічних систем;
- пошук глобального екстремуму функції;
- оптимізація функцій.

Генетичні алгоритми використовують у своїх моделях еволюційні принципи природного відбору, мінливості та спадковості [56].

При описі еволюційних алгоритмів в основному використовують терміни, що є запозичені з генетики. Наприклад, ми говоримо про популяцію особин, і як основні поняття використовуються, хромосома, ген генотип, алель, фенотип. Також використовуються визначення з технічної лексики, які відповідають цим термінам, зокрема структура, двійкова послідовність (двійкове слово), ланцюг.

Розглянемо основні поняття та визначення еволюційних алгоритмів [54].

Популяція – це кінцевий набір особин. Включені в популяцію особини в генетичних алгоритмах представлені хромосомами з закодованими наборами параметрів завдання, тобто рішеннями, які називаються точками в

просторі пошуку. У деяких видатних роботах особини називають організмами.

Хромосоми (також відомі як кодові послідовності або рядки) — це послідовності генів що впорядковані.

Ген (який також відомий як знак чи детектор) – це елементарний елемент генотипу - хромосоми.

Генотип чи структура - це набір хромосом даної людини. Отже, особи популяції можуть бути генотипами або окремими хромосомами (у досить поширеному випадку, коли генотип складається тільки з однієї хромосоми).

Фенотип – це колекція значень, які відносяться до даного генотипу, тобто декодована структура чи безліч параметрів задачі.

Алель – це значення певного гена, яке також називають значенням властивості або варіанта властивості.

Локус або позиція вказує на місце розміщення конкретного гена в хромосомі. Безліч позицій генів – це локи.

Важливим поняттям еволюційних алгоритмах є функція пристосованості (fitness function), яку також називають оціночною функцією. Це міра придатності особини у популяції. Ця функція відіграє важливу роль, тому що вона дозволяє вам оцінити ступінь придатності конкретних особин у популяції і вибрати серед них найбільш підходящих (тобто тих, які мають найбільші функції придатності) відповідно до еволюційного принципу виживання: «Найбільш адаптований» (найсильніших).

Функція адаптивності також отримала свою назву від генетики. Вона дуже впливає на роботу генетичних алгоритмів і повина також мати точне і правильне визначення. У проблемах оптимізації функція пристосованості зазвичай оптимізується (точніше, максимізується) і має назву цільовою функцією.

Популяція наступного покоління базується відповідно до цільової функції. Чим краще фізична особа, тим більша ймовірність, що вона братиме участь у кросинговері - розмноженні.

Життєвий цикл популяції складається з кількох випадкових схрещувань та мутацій, в результаті яких до певної популяції додається певна кількість нових особин [57]. Відбір у еволюційному алгоритмі — це процес створення нової популяції на основі старої, після чого стара популяція повністю гине. Після відбору в нову популяцію знову використовуються операції кросинговеру та мутації, після чого знову відбувається відбір і так далі.

Відбір в генетичному алгоритмі тісно пов'язаний з принципами відбору в природі наступним чином:

- враховується ступінь адаптації особистості (значення цільової функції цієї особи);

- реалізується виживання найбільш пристосованих особин.

Отже, принцип відбору визначає, як має будуватися популяція наступного покоління. Як правило, ймовірність участі певної особи в схрещуванні приймається пропорційно її придатності. Часто використовується так звана стратегія елітизма, при якій декілька найкращих особин переходять у наступне покоління без жодних нових змін, без участі в схрещуванні та відборі [57].

В більшості випадків кожне наступне покоління буде кращим за попереднє. Коли придатність особин перестає значно збільшуватися, процес припиняється і найкращий із знайдених індивідуумів приймається як рішення задачі оптимізації.

Розглянемо та дамо аналіз трьом найпоширенішим моделям еволюційних алгоритмів [56]:

- класична (канонічна) модель;

- дженітор;

- гібридна модель.

Операції схрещування, відбору та мутації забезпечують створенню нової популяції, яка в наступному циклі еволюційного алгоритму може бути застосована як вихідна для подальшого вдосконалення популяції (рис. 1)

безпосередньо чи після деяких процедур відсіву певних особин, які визнаються безперспективними за вхідними параметрами оцінки адаптивності. Застосування

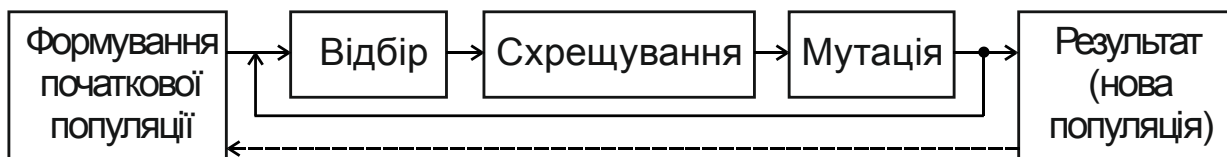


Рисунок 1.1 – Схема роботи узагальненого генетичного алгоритму

Оператор задає правила схрещування, що також називається кросингвером.

Кросингвер є основним генетичним оператором, який генерує бітові рядки — «нащадки» в результаті схрещування батьківських особин. Ця операція включає два бітові слова коду батьківської хромосоми (бітові слова). Ідея кросингвера ґрунтується в утворенні нового розрядного слова хромосоми (нового рішення) в результаті змішування за певними правилами інформації з кодів бітових слів у батьківських хромосом.

Для діагностичних завдань, наприклад, з точки зору завдання призначення оптимального набору контрольних точок, дію кросингверів можна визначити як утворення нового набору контрольних точок елементів двох існуючих наборів шляхом об'єднання частин цих наборів. У задачах організації тестувань цифрових схем з використанням еволюційних алгоритмів для організації тестів дію кросингверів можна визначити як утворення нового набору двійкових слів векторів тестових ефектів на основі наявного набору аналогічних векторів, призначених для виконання фундаментальних перевірок ОД. Особливістю нового набору двійкових слів векторів тестових ефектів має стати набуття здатності виявляти всі несправності з меншою кількістю тестових спроб.

Існує кілька типів кросингверів. Вони відрізняються схемами відбору для двійкових слів коду з батьківських хромосом, що будуть утворювати бітовий рядок - нащадка, і кількістю точок кросингвера.

Підбір двійкових слів коду у батьківських хромосом для оператора кросингвера з головною метою утворення потомства може відбуватися за різними схемами природнього відбору [54-57]:

- панміксія - схема вибору, в якій будь-які два бітові рядки з набору двійкових слів у кодї батьківських хромосом мають рівні можливості здатності до утворення батьківської пари;

- асортативний відбір - схема вибору, в якому для батьківської пари бітові лінії вибираються лише на основі інформації про ступені їх придатності. Чим вищий ступінь придатності особини, тим вища ймовірність, що вона братиме участь у формуванні потомства.

Наступні два методи утворення батьківської пари ґрунтуються на принципі ближньої та далекої "спорідненості" (під "спорідненістю" тут ми маємо на увазі відстань Хеммінга, яка для двох бінарних слів коду хромосоми відповідає кількості значень відповідних бітів у бітовому рядку, які не збігаються):

- інбридинг - схема вибору, при формуванні батьківської пари перший із батьків обирається випадковим чином, але при виборі іншого батька перевага вбільшості надається еволюційним подібним особам, що є «близкими родичами» першого батька;

- аутбридинг - схема вибору, при якій формуванні батьківської пари віддається перевага генетично різних індивідів, що є несумісними бінарними словами.

Наступним типом класифікації оператора кросингверу є кількість існуючих точок розриву у двійковому слові хромосоми що дозволяє обмінювати інформацію у бінарних слів коду хромосом — «батьки». За чисельністю точок розриву кросингвер поділяють на одноточковий (рис. 1.2), двоточковий (рис. 1.3) та рівномірний (рис. 1.4). [51].

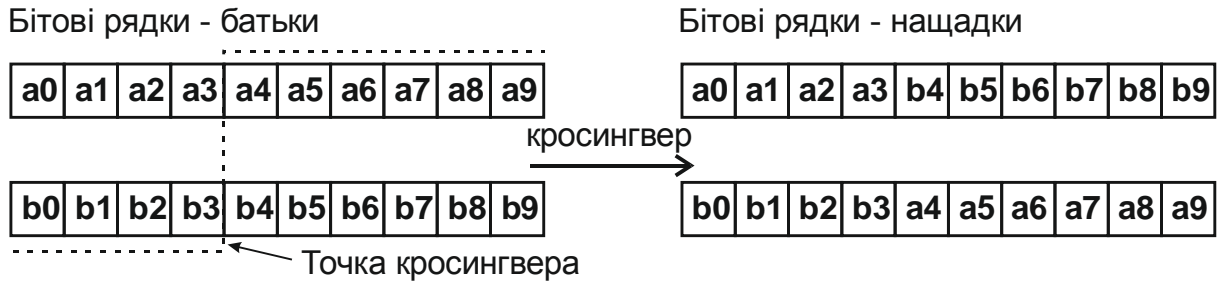


Рисунок 1.2 – Схема одноточкового кросингвера

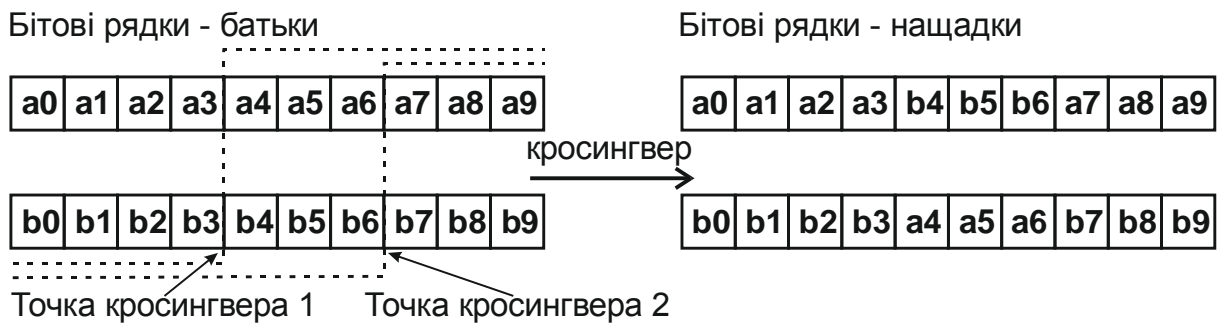


Рисунок 1.3 – Схема двоточкового кросингвера

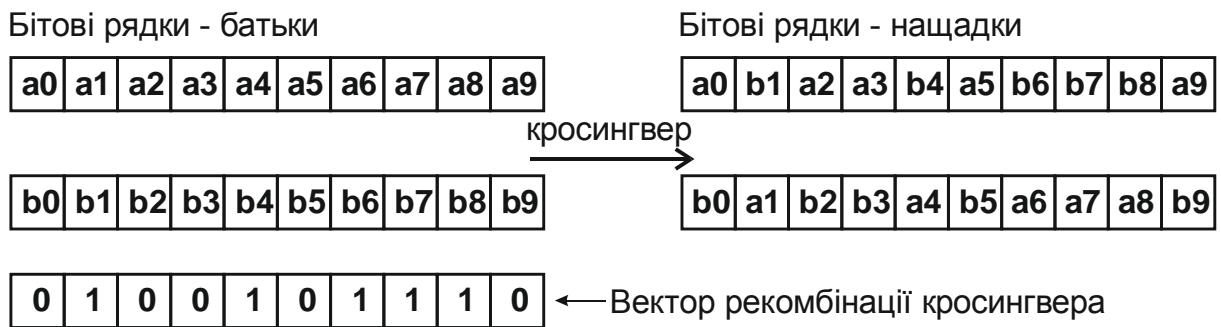


Рисунок 1.4 – Схема рівномірного кросингвера

У різних типів кросингвера є загальна позитивна властивість: вони контролюють загальний баланс між наступним використанням вже раніше знайдених добрих фрагментів простору пошуку та вивчення нових фрагментів. Це досягається за допомоги неруйнування існуючих загальних блоків у двійкових словах коду батьківської хромосоми та одночасного вивчення нових фрагментів за допомогою обміну частин двійкових слів коду у батьківській хромосомі.

Оператор мутації пояснюється як заміна вже існуючого бінарного значення двійкового слова у батьківському коді хромосоми на протилежне своєму значенню (рис. 1.5).

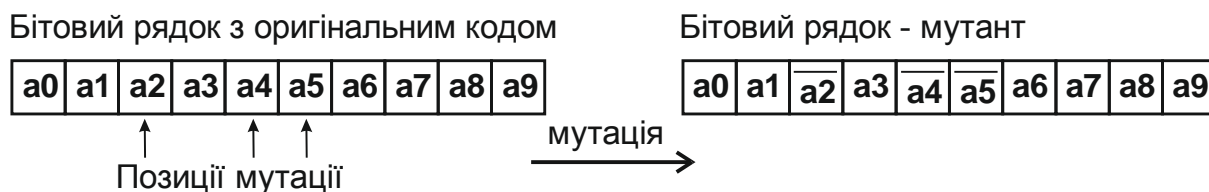


Рисунок 1.5 – Схема оператора мутації

Зрозуміло, що залежно від того, скільки бінарних частин слова хромосома інвертовано, величина відстані залежить від стану простору, що відділяє нащадка від батька.

Оператор зміни дозволяє алгоритмам досліджувати різні просторові області пошуку і в кінцевому підсумку знизити ризик перетворення в просторові крайності.

Для визначення оптимального стану діагностичних параметрів роботу цього оператора можна визначити наступним чином.

Зміна дизайну діагностичних цифрових тестів при виборі контрольних точок змінює зазначені контрольні точки шляхом додавання та видалення з них будь-яких елементів. При організації тесту продуктивності - це змінює окремі види слів, надаючи їм нові можливості для виявлення та вкладення любих зусиль у раніше непередбачені недоліки.

Порівнюючи активність розглянутого оператора, можна сказати, що кодовані рядки перемикання оператора внесли значні зміни, порівнюючи пошук з усім можливим рішенням, і оператор зміни, який поступово «налаштовує» оптимальне рішення.[50]

Останнім кроком в межах одного покоління є виконання процесу «природного відбору», тобто зменшення поточного числа популяції до початкового або переповненого розміру (критерії, що визначаються та

розглядаються з огляду на режим резервування, визначаються характеристиками завдання). Оператора природнього вибору є найважливішим фактором, що має вплив на ефективність генетичного алгоритму. Сліпе дотримання принципу: «виживає найбільш пристосований» надає можливість призвести до вузької області дослідження та вирішення завдання в контексті надзвичайної мети. З іншого боку, погано організований оператор може уповільнити процес прийняття рішень, підвищити якість популяції і, таким чином, сповільнити пошук ефективного чи кращого рішення.

Щоб об'єднати оптимальне рішення, необхідно зв'язати двійковий код нового покоління з батьківської хромосоми та новий (нащадковий) двійковий код між попереднім, зміненим та батьківським кодом. Все більше переваги надається сучасному поколінню.

Відбір особин на нову популяцію, і наступну ітерацію еволюційного алгоритму можна пояснити одним із наступних способів:

1. Випадковий вибір з однаковою ймовірністю вибору доступних пар слів у кодї хромосоми.

2. Пропорційний метод відбору [54], в якому термін бінарного хромосома переходить до наступного покоління, тим вірогідніше, чим більше адаптований код..

3. Турнірний метод відбору [57] - це свого роду пропорційний відбір, оскільки двійкова хромосома вибирається випадковим чином з усього t двійкового слова, а найкращий рядок записується в двійкові слова наступного покоління хромосоми (між. вибраним бітовим рядком є зумовлено проведення турніра). Ця операція циклічно повторюється стільки разів, скільки потрібно для отримання двійкового коду батьківської хромосоми. Розмір пар у хромосомному кодї, обраному для турніру, часто дорівнює двом. У цьому випадку мова йде про подвійний турнір. Параметр t називається кількістю турнірів.

4. Елітний метод вибору або вибір урізанням [55]. Ця стратегія використовує відсортовану спадну популяцію та першу пару слів із N хромосом, щоб скопіювати батьківський код наступного покоління двійкових слів. За задумом елітизму, багато найвитонченіших індивідуумів перейдуть до наступного покоління без змін, без участі у відборі та кросингвері. [57].

Довільне зчитування легко реалізувати, але це не гарантує бажаного результату. Вибраний метод відбору зумовлює до більш швидкого підбору алгоритмів до надійного рішення. Пропорційний метод призводить до більшого тестування області результатів.

Методи відповідності наведеним вище методам показали, що серед пристроїв був обраний найкращий метод, при цьому двійковий код у батьківської хромосоми для наступного покоління двійкових слів було підібрано з вибраної хромосоми та частини – пропорційного вибору.

На основі огляду розглянемо систематичний опис еволюційного алгоритму (класичного), який ще називають елементарним або простим еволюційним алгоритмом, при реалізації його алгоритму. [54].

Блок-схему простого еволюційного алгоритму було зображена на рис. 1.6.

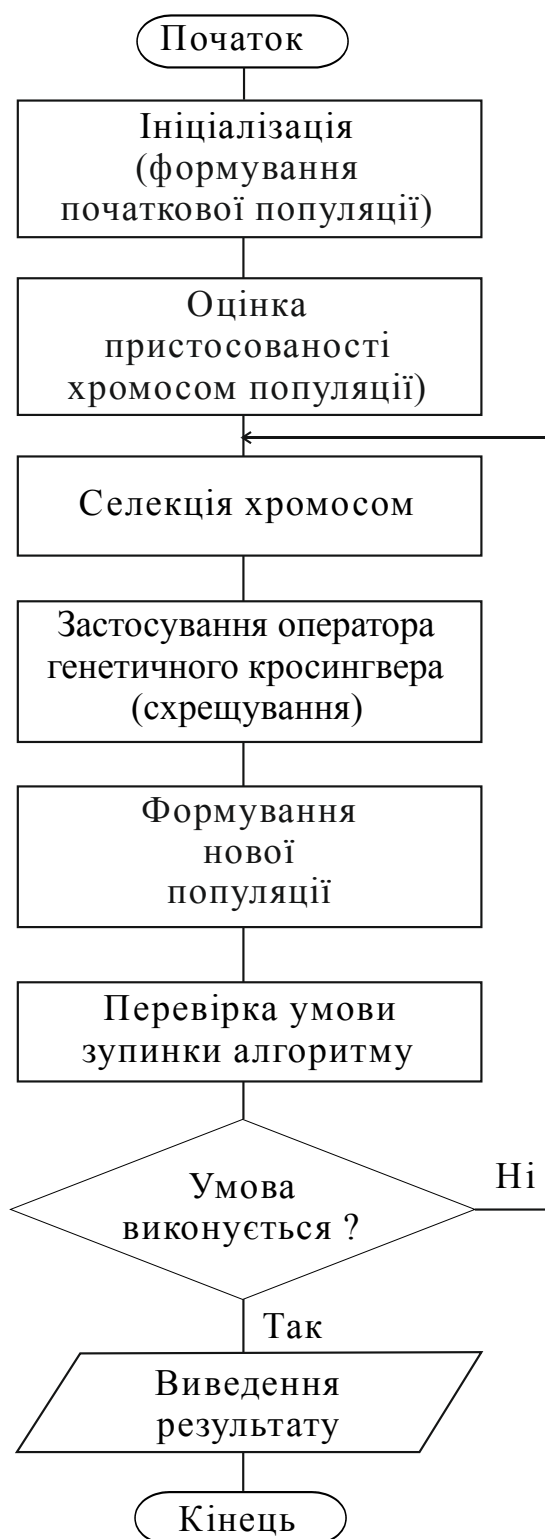


Рисунок 1.6 – Блок-схема простого еволюційного алгоритму

Звідси, простий еволюційний алгоритм базується з послідовностей певних кроків:

- ініціалізація (створення базової популяції хромосом);
- оцінка адаптивності хромосом в природній популяції;

- селекція хромосом;
- використання еволюційних операторів;
- створення нової популяції;
- перевірка умови для зупинки алгоритму;
- вибір серед хромосом «найкращої» (виведення результатів).

Ініціалізація або створення базової популяції хромосом ґрунтується у випадковому виборі певної кількості хромосом (особин), що представлені як коди двійкових слів заданої довжини.

Оцінка придатності людських хромосом на початку є відповідним завданням для обчислення окремо для кожної хромосоми. Чим більша корисність цієї функції, тим вище пристосованість хромосоми. Відповідний клас вибирається виходячи з характеру задач. Функція пристосованості завжди використовується для прийняття невід'ємних значень, тому загалом розв'язання задач є найкращою проблемою, яку нам потрібно збільшити.

Відбір хромосом складається з відбору (за значеннями, розрахованими для другого рівня функції адаптації) тих хромосом, які будуть ділитися потомством наступного покоління особин, тобто наступного покоління. Цей вибір заснований на принципі природного вибору, при якому хромосоми з найбільшими значеннями адаптації мають найбільші шанси брати участь у створенні нових особин.

Є різні способи селекції. Найпопулярнішим методом є так звана рулетка, яка за аналогією отримала свою назву від знаменитих азартних ігор. Кожна хромосома порівнюється з сектором колеса, значення якого розміщується пропорційно функції хромосоми. Чим більше функція адаптована, тим є більший розмір сектора у хромосоми.

Вибір хромосом можна представити як випадкову подію для повороту колеса рулетки зі спалахом на ділянці хромосоми. Чим більший сектор (залежно від адаптації), тим більша ймовірність перемоги даної

хромосоми. Таким чином, ймовірність вибору хромосоми є пропорційна величині її пристосованості у функції.

Результатом вибору хромосоми є базова батьківська популяція, яку також називають батьківським пулом (matingpool) з номером N , рівним заздалегідь визначеному числу нинішньої популяції.

Утворення нової популяції нащадків, створеної батьком на попередньому рівні (з обраної хромосоми), має бути завершено за допомогою еволюційних операторів.

Простий генетичний алгоритм використовує дві основні генетики: оператор кросинговеру та оператор мутації. Але слід зазначити, що зміна оператора відіграє цілком вигідну роль у порівнянні з пасажирським оператором. Це означає, що класичний генетичний алгоритм виконується майже завжди, але зміна відбувається дуже рідко. Імовірність переходу P_c дуже висока (зазвичай $0,5 < P_c < 1$), ймовірність зміни P_m дуже мала (зазвичай $0 < P_m < 0,1$). Це впливає із аналогії зі світом живих організмів, де зміни є найнезвичайнішими.

У генетичному алгоритмі зміна хромосом може бути виконана в батьківській популяції до передачі або формування потомства потомства.

На першій перехідній стадії вибираються пари хромосом із батьківської популяції (батьківського пулу). Це є тимчасовою популяцією, що складається тільки з хромосом, відібраних для відбору та призначених для подальших трансформацій за допомогою операторів схрещування та мутації для формування нової популяції нащадків. Пари хромосом походять із батьківської популяції.

Це робиться скрізь відповідно до ймовірності проходження P_c для кожної обраної таким чином пари визначається положення (локалізація) гена на хромосомі, що визначає так звану точку перетину. Якщо кожна батьківська хромосома складається з L генів, то зрозуміло, що натуральний номер точки переходу L_k менше L . Точка фіксації точки переходу – від 1 до

L-1. Під час схрещування пари батьківських хромосом створюється пара нащадків:

- нащадок, хромосоми якого знаходяться в позиціях від 1 до L_k та складається з генезу першого батька та позицій генезису L_k+1 до L від другого батька;

- нащадок, хромосоми якого знаходяться в позиціях від 1 до L_k та складається з генезу другого батька та позицій генезису L_k+1 до L від першого з батьків.

Імовірність мутації оператора P_m змінює значення гена на протилежній хромосомі (тобто від 0 до 1 чи навпаки). Наприклад, якщо ген у позиції 8 мутований на хромосомі 100110101010, його значення змінюється з 0 до 1, що веде до формуванню хромосоми 100110111010. Як зазначалося вище, ймовірність мутації дуже мала та залежить від того, чи є конкретний ген змінений чи потрібно вносити зміни.

У такий спосіб здійснювалося формування нової популяції. Хромосоми використовуються як генетичні працівники для застосування тимчасових хромосом-попередників популяції до нової популяції. Це робить так звана популяція сьогоднішньої ітерації кволюційного алгоритму. У класичному еволюційному алгоритмі вся попередня кількість хромосом замінюється новою лінією, що містить те саме число хромосом.

На кожній безперервній ітерації обчислюються значення функції підгонки для всіх хромосом поточної популяції, після чого перевіряється алгоритм умови зупинки.

Умови фехтування визначаються специфічним застосуванням генетичного алгоритму. У задачах оптимізації, якщо ви знаєте максимальне (або мінімальне) значення відповідної функції, алгоритм може зупинитися після того, як воно досягне очікуваного оптимального значення, можливо, з точністю даних.

Стійкий алгоритм також може виникнути, якщо його виконання не збільшує вже досягнуте значення. Алгоритм може бути заблокований

через певний час або через певну кількість ітерацій. Якщо виникає умова термінації, перейдіть до останнього кроку у виборі «кращої» хромосоми. В іншому випадку наступний перехід до наступного кроку здійснюється еволюційним алгоритмом – знову проводиться відбір.

Якщо виникає умова опору, необхідно показати результат роботи, тобто представити шукане рішення задачі. Хромосомний варіант «найкращий». Найкращим рішенням є максимальна функція адаптації хромосоми.

Практика використання еволюційного алгоритму показує, що він має широкий спектр застосування. Ви можете відповідно налаштувати параметри алгоритму, щоб контролювати процес пошуку рішення на основі вимог.

Певні типи проблем можуть виникати при розв'язанні задач на основі генетичного алгоритму. Існує потреба в розробці нових працівників, які беруть участь у розробці рішень. Усі модифікації застосовуваного еволюційного алгоритму засновані на схемі еволюційного алгоритму [56], структура, показана на рис. 1.1.

Типовий етап реалізації включає: утворення першого покоління, утворення потомства з використанням перехрестя кросингвера і варіацій, вибір особин у групі для створення наступного покоління.

Еволюційний алгоритм, згідно з [57], відрізняється від інших методик дослідження:

- операції, виконуються переважно не з параметрами функції безпосередньо, а з набором кодів цих параметрів;
- найкращий пошук виконується не за одним кращим рішенням, а з використанням кількох варіантів у даному рішенні;
- для оцінки якості рішення використовуються значення цільової функції, а не різні прирости;
- щоб знайти рішення проблем аналізу та оптимізації, ми повинні використовувати не детермінований, а імовірнісний аналіз.

Відзначимо наступні переваги еволюційного алгоритму [57]:

- широкі сфери застосування;
- можливість кодування скоординованих задач, виділення певним чином початкової популяції, поєднання еволюційного алгоритму з іншими методами оптимізації та продовження еволюційного процесу є суттєвими можливостями;
- обмеження практично на кожну цільову функцію;
- наочність схеми та принципів еволюційних алгоритмів.

Недоліки еволюційного алгоритму, які слідує [57]:

- відсутня гарантія знаходження глобального оптимуму, особливо за відсутності точної оцінки задач генетичних алгоритмів для вирішення прикладних задач;
- ефективність роботи низьких алгоритмів;
- перехресні параметри еволюційного алгоритму не самоадаптуються, що вимагає коригування «вручну» (результат чого еволюційний алгоритм може бути не успішним через налаштування людської помилки) або створення програмного забезпечення для оптимального обладнання.

Можна помітити, що еволюційні алгоритми більш ефективні для швидкого пошуку достатньо хорошого рішення в ситуаціях, коли конкретні методи відповіді на конкретні проблеми не можуть бути застосовані через помірно велику кількість параметрів проблеми.

Аналіз останніх робіт у сфері діагностики та еволюції моделей виявив неадекватність і надію роботи в розробці перевірених еволюційних алгоритмів для ефективного регулювання діагностичних тестів на цифрових схемах з використанням еволюційного алгоритму. У той же час дослідження нових підходів до вирішення проблем організації тестування є важливим для постійного розширення безлічі цифрових схем і подальших структурних експериментів для їх діагностики.

Існуючі методи синтезу тестувань для синтезу цифрових діагностичних тестів за допомогою еволюційного алгоритму використовуються при аналізі вибірок матриці та графа: матриці суміжності та дотичної, матричні шляхи, стиснення матриці [53] тощо. Ці методи автоматизовані і піддаються перевірці на об'єктах, у яких кількість можливих дефектів незначна через обмеження ємності пам'яті зі збільшенням вимірювання матричних процесів.

Для вирішення задач синтезу та оптимізації тестувань використовуються різні методи перевірки та модифікації таких ячок: режим розгалужень і кінців, режим найшвидшого спуску, режим динамічного програмування тощо. Реалізація цих методів вимагає більш значних обчислювальних ресурсів і не гарантує знаходження глобальної оптимальної цільової функції – форматовані тести часто не забезпечують мінімальний час для відновлення об'єкта після його збою.

Для складних цифрових систем, що складаються із сотень та тисяч елементів, застосування цих методів стає неможливим на початку – виникають проблеми із збереженням пам'яті та видачею проміжних результатів алгоритмів.

Для таких об'єктів визначення точок подання та видалення діагностичної інформації, а також характер їх використання часто є інтуїтивно зрозумілими і не гарантують покращення контролю над ними.

Еволюційні алгоритми – це новий напрямок для дослідження алгоритмів і розробки задач з багатьма можливими станами [56], суть алгоритму пошуку глобальної екстремальної багатofункціональності. Вони значно скорочують пошук складних завдань і легко пристосовуються до мінливих ситуацій.

1.3 Постановка задачі

Аналізуючи тенденції розвитку цифрових схем як об'єктів тестувань і технічних засобів та методик свідчить про актуальність створення методу організації тестувань цифрових схем на основі еволюційних алгоритмів.

Метою магістерської роботи потрібно обрати забезпечення ефективної організації тестувань цифрових схем зменшенням розмірностей використовуваних тестових послідовностей з використанням еволюційних алгоритмів на етапі тестів.

Об'єктом дослідів є процес тестування цифрових схем.

Предметом дослідів є методи організації тестувань із збільшеною перевіряючою здатністю.

Також проведені дослідження дали змогу виявити та підтвердити наявність асоціативних зв'язків у вирішуваних питань організації тестувань і теорії еволюційних алгоритмів, що необхідні для забезпечення можливості ефективності організації тестувань цифрових схем зменшенням розмірностей за рахунок тестової послідовності на основі еволюційних алгоритмів.

Для розвитку отриманих раніше результатів дослідів і підтверджень висунутих гіпотез потрібно:

- а) створити математичну модель що реалізовує методику організації тестувань цифрових схем на основі еволюційних алгоритмів;
- б) визначити базові положення методу організації тестувань цифрових схем на основі еволюційних алгоритмів;
- в) розробити метод з алгоритмічним забезпеченням;
- г) перевірити ефективність наданих алгоритмічних рішень.

1.4 Висновки

Огляд способів тестування цифрових схем показав, що вдосконалення у збільшенні складності цифрових чисел і повноти цифрових схем, створених в основі цифрового тестування, мають негативні наслідки. Оскільки інтегровані компоненти мають невелику кількість контактів, включаючи ускладнення внутрішньої інфраструктури таких компонентів, існує потреба в розробці додатків і придушенні їхньої перевіреної комбінації. Це важливо для проектування роботи - забезпечити ефективне упорядкування діагностики цифрових ланцюгів за допомогою використання еволюційної теорії алгоритмів для скорочення вимірювань за допомогою доведення послідовності.

Вивчення еволюції алгоритмів як інструменту замовлення тестів цифрових схем показує перспективність використання еволюційних алгоритмів для зменшення розмірності доказів, які є суттєвими для замовлення цифрових тестів, що є магістерською дисертацією. .

Відповідно до магістерської роботи розроблено аналіз асоціативної аналогії в діагностичних методиках та запропоновано комплексне дослідження теорії еволюційних алгоритмів..

2 МАТЕМАТИЧНА МОДЕЛЬ МЕТОДУ ОРГАНІЗАЦІЇ ТЕСТУВАНЬ ЦИФРОВИХ СХЕМ НА ОСНОВІ ЕВОЛЮЦІЙНИХ АЛГОРИТМІВ

2.1 Визначення принципів математичного апарату в поєднанні технічної діагностики і генетичних алгоритмів

Досягнення у збільшенні складності інтегрованих цифрових компонентів не мають позитивних наслідків з точки зору технології діагностики, оскільки складові частини мають лише обмежену кількість контактів, а при складних конструкціях внутрішніх частин виникає необхідність розробки та застосування більш складної комбінації внутрішніх компонентів [2]. Зараз еволюція полягає в тестуванні для перевірки якості сучасних електронних цифрових елементів з високим ступенем інтеграції з електронними продуктами, які за фінансовими та часовими витратами значно перевищують вартість їх продукції, а також інтерес до автоматизації тестування- специфічна діагностика. У зв'язку з необхідністю врахування великої кількості факторів і параметрів, а також не забуваючи про використання нетрадиційних підходів у проведенні цифрового тестування, сьогодні все більш актуальною стає розробка інтелектуальних алгоритмів та інструментів для тестування [2-4]. Використання інтелектуальних алгоритмів для вирішення дуже складних, важкоформалізованих і багатопараметричних діагностичних задач неможливе без використання надійних математичних моделей.

Так, після проведеного в минулому розділі огляду досліджень особливостей способів організації для існуючих тестувань цифрових схем і можливостей еволюційних алгоритмів як інструментів для ефективної організації зазначених тестувань, для розробки методу організації тестувань цифрових схем на основі еволюційних алгоритмів необхідно вибрати надійний математичний прилад, що дасть основу для математичної моделі методу.

Для перевірки можливої умови визначення асоціативної конгруентності в досліджуваних процесах на основі еволюційної теорії генетичного алгоритму та конкретних принципів розробки та алгоритмів використання еволюційних алгоритмів у діагностичних методиках використовується ефективний еволюційний алгоритм.

Діагностичне тестування цифрових пристроїв і систем підтримується послідовністю експериментів, які надходять від двійкових фіксованих розрядних транспортних засобів. Існують різні види класифікації свідків послідовності за методами діагностики та вирішення, але в цілому їх можна розділити на дві категорії:

- тести для змоги виявлення помилок (контрольні);
- тести для змоги локалізації помилок (діагностичні).

Тести виявлення помилок або тести для перевірки ОД можуть бути використані для визначення наявності чи відсутності помилок у його структурі, так що ОД називають одним із принципів, а в даному випадку корисним-несправним. Наступним діагностичним тестом є перевірка гіпотез, коли ОД знаходиться в несправному стані, і чим більше гіпотез можна висунути для спростування тесту без збільшення його складності та інших показників цінності, тим краще. Це найкращий спосіб реалізувати тести виявлення несправностей - тестова послідовність, сформована з найменшої кількості свідків і дозволяє перевірити здатність знаходити ОД в кожному із очікуваних станів несправності.

Випробування локалізації несправностей зазвичай застосовуються після верифікаційних випробувань і призначені для визначення місця точності даних ОД в структурі пошуку, щоб визначити визначену глибину тесту. Необхідність використання локалізації доказів помилок після тестування визначається їх побудовою, основна мета – спростувати максимальну кількість гіпотез, доступних в ОД для кожного простого тесту. Іншими словами, для випадку підтвердження в момент найпростішої задачі придушення, яка є в задачі ОД, відповідно до тесту, який очікувався

від несправних умов, виберіть підмножину умов, для яких існує невизначеність щодо їх можливості ОД для будь-якого з них. Розробка тестів на локалізацію несправностей скоріше спростовує гіпотези щодо можливості знаходження ОД у всіх станах, ніж одну зайву підмножину станів помилки, що призводить до розпізнавання статусу технічної помилки ОД з характерним станом основного. несправності. частини ОД. Оскільки застосування несправних тестів може створити різні потокові набори можливих несправних станів ОД, для кожного набору потрібно організувати власний тест локалізації помилок. Що стосується іспитів, то найкращими іспитами з локалізації помилок є лише ті, які дозволяють досягти мети за мінімальну ціну, тобто мінімальну кількість свідків для дачі показань у фіксованій послідовності.

Для покращення ефективності процедур організації тестів в їх роботі часто використовуються статистичні дані як ймовірностей знаходження ОД в кожному з доступних станів.

Щоб уникнути ускладнення майбутнього аналізу для початку приймемо за істину одну із гіпотезу про взаємне незалежне використання векторів.

Як правило, випробувальні вектори вперше створювалися для реалізації елементарних перевірок ОД в існуючих експериментах з моделями нерухомого об'єкта та його фундаментальних елементів, використовуючи при цьому широко використовувані бібліотеки свідків того ж типу, що й вузли в ОД. При цьому для скорочення тривалості процесу тестування, що є основною сферою використання еволюційних алгоритмів, була використана задача виконання узгоджених експериментів над певними компонентами та множинних простих компонентних тестів, представлених різними тестами.

Згідно з [7, 12], для використання еволюційних алгоритмів характерним є ряд властивостей серед яких:

- застосування не всіх значень параметрів завдання, а конкретно їх

закодованої види;

- параметри кодуються для використання двійковим кодом (для класичних еволюційних алгоритмів);
- однакова кількість обчислювальних ланцюгів;
- проведення пошуку не із заданої стартової точки (початкового елемента), а із заданої певної їх стартової послідовності (базової популяції);
- використання цільової методики при проведенні перетворень;
- використання ймовірнісних, а не детермінованих характеристик що потрібно для вибору елементів еволюційних перетворень.

Порівняння еволюційних властивостей алгоритмів і перевірка функцій організації дозволяє виявити подібність між ними. Вхідні параметри із завдання організації тестування, а саме значення відповідних рухових векторів і ячок, відображаються в двійкових кодах з різними значеннями символів без їх відомих характеристик. Весь бітовий код складається з відповідних реакцій пасажира та пасажира (або можливих як таких) і визначається кількістю контрольних точок, які використовуються в тестовій діагностиці ОД. Щоб побудувати якісні докази, нам потрібно проаналізувати різні варіанти одночасної реалізації кількох перевірених основ (та вибір кандидатів на оптимальну ефективність перекладу на основі ймовірності доведення характеристик ОД в кожному тесті. Метою розподілу ролей свідків є використовувати кількість свідків у послідовності свідків.

Таким чином, наявність супутніх аналогій в організації тестування та теорії розробки прикладних алгоритмів визначає перспективність використання еволюційних алгоритмів для автоматизації координації тестування.

В реалізації еволюційних алгоритмів відрізняють чотири основних операції [1,7]:

- утворення початкової популяції;

- вибір батьківських особин для процесу схрещування;
- схрещування за допомогою відібраних батьківських особин та утворення особини потомка;
- процес мутації особини потомка.

Діяльність із відбору, переходу та трансформації забезпечує навчання нової популяції, яка може використовувати еволюційних алгоритм у наступному циклі як відправну точку для подальшого вдосконалення одразу або після специфічних процедур скринінгу, які вважаються заданими критеріями для оцінки придатності.

Кількість особин описується двійковими кодами, які позначені хромосомами, кожна з яких розглядається як окремий генезис, що несе певну спадкову інформацію і може успадковуватися.

Діяльність вибору полягає у визначенні двох окремих батьків, включаючи хромосоми, які забезпечують можливість та надію на перехід. Критерієм транзитної корисності є відношення функцій оцінки ймовірності та параметрів.

Метою операції спарювання є одержання особини наступного покоління (точніше, опис їх генотипу у вигляді хромосоми), яку обирають найкращі (найбільші та наймогутніші) батьки популяції. Генеалогічна спадщина визначається особливостями світового досвіду. Переходи поділяються на єдину точку (одна або кілька робочих точок) і одноточну [7]. Точка кросинговеру визначається кількістю і положеннями частин батьківської хромосоми, а також правилами рекомбінації для частин, які відповідають коду гена наступного покоління хромосом. Оператори однорідного кросинговеру передбачають використання бінарного вектора, який визначає рекомбінацію частин батьківського гена хромосоми в хромосомі наступного покоління. Принципи єдиного оператора допускають довільні комбінації довільних точок кросинговеру (це робить його більш універсальним), і навіть у класичному двійковому представленні вектора хромосоми та вектора рекомбінації можливість генерувати наступне

покоління хромосом за допомогою звичайного генерування коду. Булева алгебра виробляє.

Діяльність мутації тяжіє до нового покоління, яке після смерті особини змінює індивідуальну стать на цінності, не властиві окремим батькам взагалі, чи рекомбінації зокрема. Як правило, зміна полягає в заміні значень кожної хромосоми на перевернутий двійковий код.

Повертаючись до технології діагностики, ми помічаємо, що основою регулювання тестування є опис характеру досвіду та стану контрольних точок у ОД при проведенні деталей та його експериментів, які відображаються у значеннях екіпажів і свідків. рухи обох пасажирів. Це як набір хромосом, що визначає властивості кожного яєчка, можна розглядати діагностичним вектором, який є бінарним сигналом від вектора яєчка і кодом відповідного вектора руху.

Таким чином, з набору яєчок і відповідних векторів руху можна сформуванати набір початкових хромосом для виконання еволюційних алгоритмів.

Особливістю сучасної цифрової діагностичної обробки ОД є те, що різні компоненти контрольних точок можуть бути використані з усієї кількості доступних даних для виконання елементарних перевірок або можуть бути визначені значення засобів у визначених контрольних точках (тобто, вони можуть приймати довільні значення). Іншими словами, для кожного експерименту значення значень в контрольних точках ОД можна розділити на фактичні (з 0 або 1 рівнем фіксованого знакового рівня) і нерелевантні (показники сигналу x і можуть бути замінені на 0 або 1 якщо необхідно). Наявність контрольних точок з невідповідними річними значеннями, які можна довільно змінювати, дозволяє одночасно виконувати декілька свідків за умови, що фактичні значення звичайних пасажирських свідків відповідних екіпажів не суперечать їх відповідним реакціям один з одним.

У концепціях теорії генетичних алгоритмів фактичні значення

діагностичних ознак, знайдених у бітах хромосоми, представляють дійсні класифікації, які успадковуються в наступних поколіннях після перехідних операцій. Оскільки типи сильних і слабких позицій в описі хромосом різних випробувань можуть бути різноманітними, визначення хромосом зламу для використання перехресних точок неможливе в багатьох діагностичних завданнях, які було б вигідно використовувати уніформу хрест-навхрест. При реалізації генетичного алгоритму векторної рекомбінації з однорідним кросингвером краще порівнювати хромосоми кожного пасажера такого ж розміру з видами вектора. У пасажирській рекомбінації значення кожної цифри можна визначити за правилом: рівне, якщо відповідний палець у коді хромосоми відповідає сильному гену, який неможливо втратити при схрещуванні, і нульовий — інакше.

Запропонована діагностична модель даних для векторів хромосом та векторів рекомбінації дозволяє сформулювати основні правила визначення схрещування. Кожну хромосому можна схрещувати за двох наборів умов:

- якщо ділянки сильних генів не збігаються з вектором рекомбінації;
- якщо у разі деяких сильних позицій генів у рекомбінаційних векторах значення генезу генів збігаються з положеннями зазначених хромосом.

Перевірка допустимості схрещування певних особин з хромосомними наборами може бути реалізована порозрядними логічними операціями над векторами рекомбінації та хромосомами.

Для того, щоб покоління наступних поколінь стало вектором рекомбінації, важливо подумати про сильну міграцію окремих генів.

Деталі, які розглядаються на певному етапі виконання алгоритму, обіцяного для створення нащадка, виділяються в набір непридатного відтворення. Код ідеального перехресного виконання генетичних алгоритмів - це переклад усіх людей останнього покоління в набір непридатного відтворення. Виконання генетичних алгоритмів може бути припинено перед кожним набором для належного відтворення, якщо вони

розробляють заздалегідь визначені критерії припинення. Формування родоводів має враховувати ідентичність нащадків або подібні можливості, і для кожного нового покоління було б вигідно орати дублікат, що також можна виконати за допомогою генетичного алгоритму [15].

Для генерованих векторів хромосом, які не беруть участі в найбутніх схрещуваннях, може бути використана модифікована операція, яка полягає в інвертуванні значень для інших слабких типів і оцінці ефективності виконання (шляхом моделювання чи експерименту). Можливі дійсні зміни та типи, але для обслуговування перевірених технічних ресурсів потрібен нагляд.

2.2 Розробка математичної моделі методу

Відповідно до пов'язаних аналогій, знайдених у дослідженнях при проведенні експериментів цифрових схем і еволюційної теорії еволюційного алгоритму, а також деяких загальних принципів тестувальних засобів і еволюції технічних алгоритмів. математики для створення математичної моделі. пристрої на основі еволюційних алгоритмів.

Для опису даної предметної області задач організації тестувань введемо опису наступних множин вихідних параметрів:

- $L: \{l_1, l_2, \dots, l_i, \dots, l_k\}$ - множина можливих непрацюючих технічних головних станів ОД;

- $P: \{P(l_1), P(l_2), \dots, P(l_i), \dots, P(l_k)\}$ - множина заданих значень статистичних даних де імовірності знаходження ОД в кожному з певних технічних станів $l_i \in L$;

- $T: \{t_1, t_2, \dots, t_i, \dots, t_m\}$ - множина тестувальних векторів, розроблених для перевірки станів ОД $l_i \in L$;

- $R: \{r_1, r_2, \dots, r_i, \dots, r_m\}$ - множина векторів що отримуюні з

відповідних реакцій при оголошені на ОД блоків тестувальних векторів $t_i \in T$.

Як вже зазначали, традиційно тестувальні вектори $t_i \in T$ першими впровадили базові діагностичні моделі на основі ОД нерухомого об'єкта та його існуючих компонентів. У той же час типові доступні вузли діагностичної бібліотеки широко використовуються як частина ОД. У таких випадках задача проведення одночасної перевірки цих компонентів і кількох елементарних перевірок з доказами $t_i \in T$ і $t_j \in T$ (при $i \neq j$) в часі скорочують тривалість процесу тестування, яка є цільовою областю генетичні алгоритми.

Вхідні параметри з проблемами організації тесту, а саме значення вектора $t_i \in T$ і векторного руху відповідного $r_i \in R$, представлені в двійковому коді значень без повідомлення про інші ознаки. Весь бітовий код тест-векторів $t_i \in T$ і рух відповідних векторів $r_i \in R$ є постійним (або може виконуватися як такий) і визначається кількістю контрольних точок ОД, які використовуються в діагностичних ящиках. Щоб побудувати якісні докази, нам потрібно проаналізувати різні варіанти одночасної реалізації кількох перевірених основ (для всіх елементів $t_i \in T$ і $t_j \in T$, $i \neq j$), а також відбір кандидатів для підвищення ефективності доказів, які надаються на ймовірних фундаментальних характеристиках. ймовірність знаходження ОД в кожному з технічних станів ОД шляхом поєднання характеристик елементів тестової здатності з $t_i \in T$. Метою організації пробації є зменшення кількості свідків, які дають показання проти наступних свідків.

Основою для організації тестів є описи характеру застосування та стану контрольних точок ОД при виконанні кожного з тестів, які відображаються у значеннях векторів $t_i \in T$ випробувань та векторі відповідних реакцій $r_i \in R$. Це означає, що ми можемо досліджувати набір хромосом, який розпізнає його властивості i -го експерименту, щоб розглянути постачання двійкового векторного коду $t_i \in T$ і $r_i \in R$:

$$c_i = \text{text}(t_i) + \text{text}(r_i), \quad (2.1)$$

де c_i – хромосома що описує тестову перевірку із застосуванням тестового вектора $t_i \in T$, обумовлена додаванням новго значень векторів $t_i \in T$ і $r_i \in R$ що орисується як символічних рядків (функція *text*).

Отже, на основі сукупності тест-векторів T та векторів відповідних операцій R із використанням (2.1) може бути утворена множина хромосом базової популяції C для реалізації еволюційних алгоритмів:

$$C: \{ c_1, c_2, \dots, c_i, \dots, c_m \}.$$

Особливістю сучасної цифрової діагностичної обробки ОД є те, що різні показники можна вибрати із загальної кількості доступних засобів або значень у фіксованих контрольних точках. Іншими словами, значення для знаків $t_i \in T$ і $r_i \in R$ в ОД можна розділити на фактичні точки потужності (чітко визначений рівень сигналу 0 або 1) і невідповідні (позначений x'' і при необхідності їх можна замінити на 0 або 1).

Наявність контрольних точок з індивідуальними значеннями значень, які можуть змінюватися за бажанням, дозволяє одночасно виконувати перевірки свідків $t_i \in T$ і $t_j \in T$ ($i \neq j$), за умови фактичних значень їх екіпажів $t_i \in T$ і $t_j \in T$, наприклад. обидва вектори $r_i \in R$ і $r_j \in R$, що не вступають у протиріччя.

У концепціях теорії генетичних алгоритмів фактичні значення діагностичних стандартів, знайдені в бітах хромосомного коду $c_i \in C$, мають сильні компоненти генів, які повинні бути успадковані в наступних поколіннях після проходження активності. Оскільки типи сильних і слабких позицій в описі хромосом різних випробувань можуть бути різноманітними, визначення хромосом зламу для використання

перехресних точок неможливе в багатьох діагностичних завданнях, які було б вигідно використовувати уніформу хрест-навхрест. . Наприклад, щоб виконати генетичні алгоритми вектора рекомбінації з рівномірним хрест-накрест, ми порівнюємо кожен вектор хромосоми з векторною маскою $c_i \in C$ подібного розміру h_i , тобто для формування набору векторів рекомбінації:

$$H: \{ h_1, h_2, \dots, h_i, \dots, h_m \}.$$

При рекомбінації вектора $h_i \in H$ значення кожної цифри буде дорівнювати одиниці, якщо відповідна цифра в коді хромосоми $c_i \in C$ відповідає гену сильного проходження, який неможливо втратити, а в іншому випадку дорівнює нулю..

Запропонований набір діагностичної інформації у вигляді хромосомних векторів $c_i \in C$ та векторів рекомбінації $h_i \in H$ дозволяє сформулювати основні правила визначення потенціалу переносу..

Особини з набором хромосом $c_i \in C$ і $c_j \in C$ можуть схрещуватись за двох відповідних умов:

- якщо позиції дійсних класів векторів рекомбінації $h_i \in H$ і $h_j \in H$ не збігаються;
- якщо збіг деяких положень сильного генезу у векторах рекомбінації $h_i \in H$ і $h_j \in H$ збігаються, а значення пологів у цих положеннях хромосоми.

Перевірка допустимості кожного переходу з хромосоною $c_i \in C$ і $c_j \in C$ може бути виконана шляхом логічної побітової рекомбінації векторів і хромосом. Відсутність протипоказань до проходження сигналу не впливає на функцію функції y_{ij} :

$$y_{ij} = (c_i \oplus c_j) \wedge h_i \wedge h_j = 0. \quad (2.2)$$

Коли гени $c_i \in C$ в хромосомі замінюються недійсними нулями, то за контрольної умови (2) вектор c_{ij}^1 окремої хромосоми бітів C^1 нового (першого) покоління визначається як роз'єднання батьківських векторів хромосоми $c_i \in C$ і $c_j \in C$:

$$c_{ij}^1 = c_i \vee c_j. \quad (2.3)$$

Для того, щоб покоління наступних поколінь стало вектором рекомбінації, важливо подумати про сильну міграцію окремих генів. Нове значення вектора рекомбінації також було визначено для вектора пролонгованої рекомбінації батьків $h_i \in H$ і $h_j \in H$, що представлено формулою для $cc_{ij}^1 \in C^1$:

$$h_{ij}^1 = h_i \vee h_j. \quad (2.4)$$

Ми отримали тестовий вектор другого покоління (2.3) для перевірки здатності коду хромосоми розв'язувати стани ОД $l_i \in L$, які дотримувалися індивідуального батьківського коду. Імовірність виявлення несправності з кусанням ОД за допомогою тесту пасажира нового покоління визначається найвищою ймовірністю виявлення несправностей від усіх окремих батьків, чії гени успадковані детермінованим пасажиром.

Для максимальної ефективності використання еволюційних алгоритмів у порядку тестування виявлення несправностей особливо вигідно пройти операцію, яка здатна виявляти технічні стани ОД $l_i \in L$ з найбільшими відповідними значеннями ймовірності $P(l_i)$. Організуючи тести на локалізацію несправностей, працездатному населенню було б добре звернути увагу на залишкові підмножини можливих станів, технічні можливості виявлення результуючого нащадка ОД, якомога ближче до половини значення. повна ймовірність ОД в цих станах.

Особи, які визнані в основі (2.2), виділяються на певному етапі виконання еволюційних алгоритмів, менш придатних для створення потомства. Код ідеального перехресного виконання генетичних алгоритмів - це переклад усіх людей останнього покоління в набір непридатного відтворення. Виконання генетичного алгоритму може бути припинено за наявності відтворюваних особин, за умови, що критерії дострокового припинення були визначені, але простота використовуваних послуг (2.1-2.4) і важливість отримання мінімальних доказів все ще відносно невеликі. Кількість можливих ліній, що використовують генетичний алгоритм при дослідженні задач, упорядкуванні доказів, було б добре провести всі можливі схрещування. При формуванні родоводів необхідно враховувати ідентичність потомства або подібне, і для кожного нового покоління бажано проводити процедуру подвійної елімінації, яку також можна виконати за генетичним алгоритмом[15].

Для генерованих хромосомних векторів, які не беруть участі в подальших схрещуваннях, може бути використана модифікована операція, яка полягає в інвертуванні значень для інших слабких типів і оцінці ефективності виконання (шляхом моделювання або експерименту). Можливі дійсні зміни та типи, але для виявлення перевірених технічних ресурсів потрібен контроль за збереженням.

Таким чином, запропонована математична модель тестування цифрових схем базується на наборі алгоритмів на основі еволюційної математичної теорії, операцій із символічними значеннями, булевої алгебри та теорії фундаментальних еволюційних алгоритмів.

Неявно можна представити математичну модель методу:

$$M = \langle L, P, T, R, C, H \rangle. \quad (2.5)$$

Основні елементи представлені в моделі і дозволяють системно вносити дані в програму за допомогою методу програмної реалізації:

- $L: \{ l_1, l_2, \dots, l_i, \dots, l_k \}$ - множина можливих непрацюючих технічних головних станів ОД;

- $P: \{ P(l_1), P(l_2), \dots, P(l_i), \dots, P(l_k) \}$ - множина заданих значень статистичних даних де імовірності знаходження ОД в кожному з певних технічних станів $l_i \in L$;

- $T: \{ t_1, t_2, \dots, t_i, \dots, t_m \}$ - множина тестувальних векторів, розроблених для перевірки станів ОД $l_i \in L$;

- $R: \{ r_1, r_2, \dots, r_i, \dots, r_m \}$ - множина векторів що отримують з відповідних реакцій при оголошенні на ОД блоків тестувальних векторів $t_i \in T$.

- $C: \{ c_1, c_2, \dots, c_i, \dots, c_m \}$ - множина хромосом базової популяції для реалізації еволюційного алгоритму;

- $H: \{ h_1, h_2, \dots, h_i, \dots, h_m \}$ - сукупність векторів для рекомбінації хромосом $c_i \in C$.

Основні постачальники, які забезпечують представлення даних у вигляді елементів еволюційної теорії алгоритмів і надають доступ до інформації про перетворення при виконанні операцій на основі методів зі значеннями текстів і функцій булевої алгебри.:

$c_i = \text{text}(t_i) + \text{text}(r_i)$ - оператор утворення коду хромосоми $c_i \in C$ з тестувальних векторів $t_i \in T$ і $r_i \in R$;

$y_{ij} = (c_i \oplus c_j) \wedge h_i \wedge h_j$ - функція іцінки реальних можливостей під час селекції особин, що представлені хромосомами $c_i \in C$ і $c_j \in C$:

$$y_{ij} \begin{cases} = 0 & - \text{схрещування хромосом } c_i \in C \text{ і } c_j \in C \text{ можливе;} \\ \neq 0 & - \text{схрещування хромосом } c_i \in C \text{ і } c_j \in C \text{ неможливе.} \end{cases}$$

$c_{ij}^1 = c_i \vee c_j$ - функція кросингвера (оператор що забезпечує селекцію хромосом $c_i \in C$ і $c_j \in C$);

$h_{ij}^1 = h_i \vee h_j$ - оператор що дозволяє векторам рекомбінації

схрещуватись $h_i \in H$ і $h_j \in H$.

2.3 Висновки

За результати досліджень в даному розділі було отримано наступні результати:

1) Було обґрунтовано корисність застосування моделі $M = \langle L, P, T, R, C, H \rangle$ для визначення задач організації тестувань цифрових схем на основі еволюційних алгоритмів;

2) запропоновано оператор для формування коду хромосом $c_i \in C$ з тестувальних векторів $t_i \in T$ і $r_i \in R$;

3) запропоновано методику оцінки можливостей селекції особин, за допомогою хромосом $c_i \in C$ і $c_j \in C$;

4) обчислено оператор кросингвера (оператор селекції хромосом $c_i \in C$ і $c_j \in C$)

5) обчислено оператор селекції векторів рекомбінації $h_i \in H$ і $h_j \in H$.

Також запропонована математична модель та набір певних операторів з функціями, що є основою для створення методу організації тестувань цифрових схем на основі еволюційних алгоритмів.

3 ОПИС МЕТОДУ ОРГАНІЗАЦІЇ ТЕСТУВАНЬ ЦИФРОВИХ СХЕМ НА ОСНОВІ ЕВОЛЮЦІЙНИХ АЛГОРИТМІВ

3.1 Опис основних положень методу

Щоб дослідити фундаментальні принципи проектування цифрових схем на основі еволюційних алгоритмів, спочатку проілюструємо його основне призначення та умови застосування.

В умовах постійного зростання інтегральної комбінації компонентів цифрової електроніки перевірена функція перевіреної конструкції залишається актуальною метою застосування еволюційної теорії алгоритмів.

Роль організації тестів для іспитів з базовими пристроями з високим рівнем інтеграції ніколи не втрачала обертів, і навпаки, сумісність цієї роботи завжди посилювалася з появою нових компонентів та інтегрованих технологій.

Більш точно тестування цифрових схем у більшості випадків унеможливорює повну організацію іспитів для ідентифікації всієї ОД, щоб тимчасово вирішити проблему організації випробувань. У зв'язку з необхідністю врахування великої кількості продуктів і параметрів, сучасною еволюцією є тестування якості для перевірки надвисоких рівнів інтеграції сучасних електронних компонентів та електронних виробів, що за своїми фінансовими та тимчасовими витратами значно перевищує вартість. продовжуючи себе покращувати.

Існує кілька підходів, які широко використовуються в діагностичних методах для спрощення тестування та планування цифрових схем:

- бібліотеки тестів використовуються для кожного компонента, запропонованого їх виробниками;
- до проведення експертизи використовувалися тестові бібліотеки для типів вузлів і блоків, що входять до циклу цифрового тестування;
- розробляються тести для визначення можливої технології ОД на

основі наявної інформації про її структури, схеми та функції.

Крім того, отримані на цьому етапі результати не можна вважати надійним повноцінним тестуванням ефективного проектування цифрових схем на основі розробки еволюційного алгоритму.

Коли окремі тестові бібліотеки, вузли або блоки діагностуються в цифрових схемах, бібліотеки доказів не враховують структурні характеристики, схеми та функції ОД у складній цифровій схемі. З цієї причини була використана процедура адаптації, щоб дозволити застосування структурних, схемних і функціональних характеристик ОД, але вона залишається схваленою з самого початку, коли її модифікують для виявлення несправностей окремих частин, частин або агрегатів, як це передбачено при використанні метод. екзаменаційна бібліотека.

При аналізі тестів ОД для виявлення можливих несправностей ОД на основі наявних даних про їх структури, схеми та відомі функції, таких як стандартний тестовий вектор $t_i \in T$, спочатку розробляється реалізація базової перевірки ОД на основі існуючих діагностичних моделей об'єкта та його складові. Контрольний список ОД використовується як тест, який відноситься до окремої несправності (знову ж таки, несправності одного компонента, вузла або пристрою).

Таким чином, як би не застосовувався підхід до впорядкування свідків, першим кроком є виявлення несправностей окремих частин ОД без урахування можливості одночасної перевірки інших частин.

У таких випадках дослідження виконання одночасної перевірки цих компонентів і багатокomпонентних перевірок з доказами $t_i \in T$ і $t_j \in T$ (при $i \neq j$) залишається відкритим у часі, щоб скоротити тривалість процесу випробування..

Як визначено в огляді існуючих методів і методів організації діагностичних цифрових тестів, упорядкування надійних доказів на основі готовності окремих елементів заблокованих свідків T належить до категорій високоскладних N-P-завершених задач, яких не існує. ефективні

алгоритмічні рішення. Це зумовлює важливість даної роботи та широкого вивчення діагностичних варіантів у пошуку різних типів рішень. Для перевірки організації використовуються різні методи організації тестування та оптимізації, що було зазначено в огляді, але не було знайдено однозначного рішення щодо організації тестування, яке б проводило подальші дослідження і, головне, з використанням алгоритму генетичного тестування для організації. ефективно організувати діагностичні тести на цифрових схемах.

Таким чином, цільова область застосування еволюційних алгоритмів, розроблених у методі упорядкування тестів цифрових схем, базується на масиві доступних фіксованих векторних елементарних перевірок T , який спрямований на поєднання кількох елементів тестів $t_i \in T$ і $t_j \in T$ (при $i \neq j$). на момент виконання координат одночасної перевірки деталей ОД і скорочення тривалості процесу випробування.

- аутбридинг - схема природнього відбору, в якій при формуванні батьківської пари ген розміщується перед бінарними конфліктними словами з різними символами.

Відбір особин перетворився на нову популяцію, і наступну ітерацію генетичного алгоритму можна пояснити одним із наступних способів:

– випадковий вибір з однаковою ймовірністю вибору доступних пар слів у коді хромосоми;

– пропорційний метод відбору, в якому термін бінарного хромосома переходить до наступного покоління, тим вірогідніше, чим більше адаптований код;

– турнірний метод відбору - це свого роду пропорційний відбір, оскільки двійкова хромосома вибирається випадковим чином з усього t двійкового слова, а найкращий рядок записується в двійкові слова наступного покоління хромосоми (між. вибраним бітовим рядком є зумовлено проведення турніра);

– елітний метод вибору або вибір урізанням. Ця стратегія використовує відсортовану спадну популяцію та першу пару слів із N хромосом, щоб скопіювати батьківський код наступного покоління двійкових слів.

Методи відповідності наведеним вище методам показали, що серед пристроїв був обраний найкращий метод, при цьому двійковий код у батьківської хромосоми для наступного покоління двійкових слів було підібрано з вибраної хромосоми та частини – пропорційного вибору.

Метод пропорційного вибору здійснюється за принципом природного вибору, враховуючи, що хромосоми з найбільшими значеннями адаптації мають найбільші шанси брати участь у створенні нових особин. Згідно з концепцією, деякі люди переходять до наступного покоління без змін, без участі в кросовері та читанні.

Розглянемо принципи виконання методу упорядкування доказів за допомогою еволюційного алгоритму, заснованого на декомпозиції методу, з використанням методу пропорційного відбору та дизайну елісій.

При виконанні методу тестування з використанням еволюційних алгоритмів метод пропорційного відбору дозволяє особам вибирати між переходами та проектуванням-виключенням-передачами до наступного покоління індивідів, чий генетичний алгоритм не в змозі зв'язати їхні переходи на цьому етапі. втрата неприпустима.

Імовірність участі в спарюванні за пропорційним методом приймається як пропорційна її конгруентності, що вимагає визначення функції хромосоми для оцінки функції пристосованості хромосоми для виконання нормативного методу реалізації еволюційних алгоритмів.

Функціональна пристосованість, інакше відома як функція оцінки, є остаточним поняттям у генетичному алгоритмі. Способом адаптації є індивідуальний серед людей. Це важливе завдання, яке дозволяє оцінити ступінь адаптації серед особин у популяції та вибрати найбільш підходящі

(тобто ті, які мають найбільший вплив на функцію адаптації) за принципом виживання еволюції. з «найсильніших» (найсильніших).

Функція адаптації також отримала свою назву безпосередньо від генетики. Генетичний алгоритм має потужний вплив на роботу і повинен мати точне і правильне визначення. У задачах оптимізації відповідна функція зазвичай оптимізується (точніше, максимізується) і називається цільовою функцією.

Популяція наступного покоління формується відповідно до цільової функції. Чим адаптивніше кожен, тим більша ймовірність участі в кросовері, тобто розмноженні.

Для досягнення максимальної продуктивності за рахунок використання генетичних алгоритмів при замовленні тестів на виявлення несправностей, особливо вигідно пройти операцію, яка дасть можливість особам виявляти технічні умови ОД $l_i \in L$ з найвищими значеннями ймовірності, що відповідають $P(l_i)$. Якщо векторний екзаменатор $t_j \in T$, з якого утворено двійкове слово хромосоми $c_j \in C$, повинен виявити кілька дефектів, що виражаються в різних технічних станах ОД $l_i \in L$, сумісність деякої функції з кодом хромосоми $c_j \in C$ може бути використовуючи формулу:

$$f_j = \sum_{i=1}^{|L|} q_i P(l_i), \quad (3.1)$$

де f_j – значення пристосованості особини у функції з кодом хромосоми $c_j \in C$; $|L|$ – величина множини можливих помилок технічних станів у ОД L ; $P(l_i)$ – ймовірність пошуку ОД в одному із несправних технічних станів $l_i \in L$; q_i – ознака тестування вектора $t_j \in T$ знаходити різні ознаки ОД в одному із несправних технічному стані $l_i \in L$ ($q_i = 1$ умова у тестувальних векторах $t_j \in T$ даної вимоги, $q_i = 0$ умова за відсутності у тестувальному

векторі $t_j \in T$ відповідної вимоги).

Оцінка придатності людських хромосом на початку є відповідним завданням для обчислення окремо для кожної хромосоми. Друга функція пристосованості (3.1) завжди приймає невід'ємні значення. Чим більша корисність цієї функції, тим вище пристосованість хромосоми. Відповідний клас вибирається виходячи з характеру задач.

Пропорційний відбір на основі функції конгруентності (3.1) при реалізації тестів упорядкування з використанням еволюційних алгоритмів також відповідає фундаментальним вимогам теорії генетичних алгоритмів:

- використання цільової функції з виконанням мутацій;
- використання ймовірних ознак, а не детермінованих, щоб вибрати еволюційні перетворення елементів.

Характеристика процесу діагностики цифрових ланцюгів полягає в тому, що не всі затвержені пасажери $t_j \in T$ можуть бути зібрані разом для одночасної реалізації, а це означає, що не всі хромосоми можуть бути схрещені $c_j \in C$. Кожна хромосома $c_i \in C$ та $c_j \in C$ призначається лише за цієї умови, тому жодна передача не може пошкодити дійсну хромосому (весь сильний генез кожної хромосоми буде успадковано від потомства). Запропонований набір діагностичної інформації у вигляді вектора $c_i \in C$ та векторної рекомбінації $h_i \in H$ дозволив сформулювати основні правила визначення передатної здатності. Кожна хромосома містить список $c_i \in C$ та $c_j \in C$ за двох умов:

- якщо $h_i \in H$ і $h_j \in H$ не збігаються з позицією сильних генів рекомбінаційних векторів;
- якщо збігаються певні позиції сильних генів у векторах рекомбінації $h_i \in H$ і $h_j \in H$ збігаються і значення конкретних генів у визначених ділянках хромосом $c_i \in C$ і $c_j \in C$.

Вирішальну роль відіграє оцінка можливості схрещування кожної пари хромосом особин $c_i \in C$ і $c_j \in C$ на основі математичної схеми шляхом

застосування перевірених еволюційних алгоритмів (2.2).

Втрата в методі виконання використання еволюційних алгоритмів має кілька характеристик, специфічних для діагностичних властивостей:

– Кожна особа в поточній ітерації еволюційного алгоритму може брати участь у схрещуваннях, які синхронізовані з усіма іншими особинами (відповідно до функції (2.2)) інших індивідів;

– результатом переходу двох окремих батьків є одне потомство, яке успадковує всі сильні батьківські роди і замінює їх у наступному поколінні (люди, які стали батьками на певній ітерації генетичного алгоритму, вважаються відчайдушними і не брали участі в наступні ітерації);

– особи, які не брали участі в селекції у своїх популяціях, переходять до найближчої популяції без відбору (крім випадків відхилення на етапі усунення зайвих особин).

Особини, розпізнані на підставі (2.2) для конкретного етапу реалізації генетичного алгоритму створення потомства, поділяються на безліч непридатних репродукцій (переведених до останнього покоління завдяки дизайну Elisism).

Таким чином, нове покоління включає нащадків через пасаж особин з попереднього покоління, і носіїв вихідної хромосоми, які не брали участі в переході і не змогли передати дійсне потомство в потомство.

Для хромосом, створених вектором, можна використовувати модифіковану активність, яка полягає в інвертуванні значень генезису слабких залишків та оцінці ефективності роботи (шляхом моделювання або експерименту). Можливі дійсні зміни та типи, але для виявлення перевірених технічних ресурсів потрібен контроль за збереженням. Не можна відкрито рекомендувати використання хірургічної зміни, тому що зміни, як і в природі, є безцінним процесом і можуть мати як позитивні, так і негативні наслідки. Загалом, модифікована операція використовується рідко і не проводиться.

При формуванні родоводів необхідно враховувати ідентичність

нащадків або подібні можливості, і для кожного нового покоління бажано проводити процедуру елімінації дублікатів, яку також можна виконати за допомогою генетичних алгоритмів або інших зайвих методів виявлення. він схвалює. Це останній крок у генерації єдиного покоління – виконання процесу «природного вибору», тобто зменшення поточного числа до початкового або переповненого розміру (визначення критеріїв визначення та збереження надмірності визначається з урахуванням врахувати проблеми). Вибір операції є найважливішим фактором, що впливає на ефективність генетичного алгоритму та здійснення систематичних тестів з використанням еволюційних алгоритмів.

Визначте умови завершення роботи в еволюційному алгоритмі.

Код ідеального перехресного виконання еволюційних алгоритмів - це переклад усіх людей останнього покоління в набір непридатного відтворення. Виконання еволюційного алгоритму може бути припинено за наявності відтворюваних особин, за умови, що критерії мають заздалегідь визначений проміжок часу (обмеження на кількість ітерацій або час роботи алгоритму), але простота інших функцій (2.1-2.4) і важливість отримання мінімальних тестів, а також порівняння невеликої кількості можливих поколінь з еволюційними алгоритмами в задачах при пошуку користувачів це допомагає забезпечити всі можливі переходи в цифрових доказах.

Відтворення діагностичної інформації за результатами генетичного алгоритму - достатньо розділити кожен вектор хромосоми слова $c_i \in C_{GA}$ (C_{GA} - на копії хромосоми останньої еволюційної лінії, утвореної генетичним алгоритмом), щоб розділити вектор і вектор хромосоми. відповідний рух і формування від значень вектора яєчок T і вектора відповідного руху R .

3.2 Опис алгоритмів реалізації методу

Опис алгоритм реалізації методу організації тестів із застосуванням еволюційних алгоритмів юазується, відповідно, на принципах реалізації еволюційних алгоритмів. З іншого боку, еволюція алгоритмів реалізації методу буде розвиватися відповідно до фундаментальних принципів тестового методу, поряд з розробкою методик еволюції алгоритму, визначених у попередньому розділі.

У роботі над кращим еволюційним алгоритмом індивідууми навчаються виконувати кращі завдання, які, виконуючи еволюційний алгоритм, задають ці копії хромосоми.

Життєвий цикл популяції складається з кількох хрестів (за допомогою хрест-навхрест) і можливих змін (за бажанням) (за бажанням), завдяки чому до створеної популяції додається певна кількість нових людей. Після вибору операції перехресного переходу та зміни знову застосовуються до нових людей, потім відбір відбувається знову і так далі. Відбір у генетичному алгоритмі — це процес формування нової популяції зі старої після смерті старої популяції. Процес відбору-пропуску повторюється до досягнення заданої умови алгоритмів закриття.

Простий еволюційний алгоритм базується на виконанні набору певних кроків:

- ініціалізація (утворення базової популяції хромосом);
- оцінка хромосом пристосованості в популяції;
- схрещування хромосом;
- використання еволюційних операторів;
- утворення нової популяції;
- перевіряється умова зупинки данного алгоритму;
- вибір «найкращої», адаптивної, хромосоми.

Як ми зазначали в основних принципах упорядкування методів з використанням еволюційних алгоритмів, реалізація еволюційних

алгоритмів за допомогою цього методу характеризується кількома спеціалізованими тутсувальними методиками.

Для реалізації методу запропоновано алгоритм, що складається з наступної послідовності операцій.

Примітка 3.1. В данному алгоритмі використовуються булеві значення характеристик виконання умов: 1 – умова що виконується контрольовано (дійсна), 0 – умова контрольована, але не виконується (хибна). При необхідності ці ознаки можна звести до значень логічного типу (true або false).

Алгоритм 3.1.

3.1.1 Ввести вхідні дані про можливі несправності в стані ОД $l_i \in L$.

3.1.2 Визначити K як певну послідовність можливих несправностей у станах ОД:

$$K = / L /.$$

3.1.3 Ввести вхідні дані ймовірностей $P(l_i)$ для знаходження ОД в кожному з можливих помилкових станів $l_i \in L$ (підготувати множину значень даних ймовірностей $P : \{ P(l_1), P(l_2), \dots, P(l_i), \dots, P(l_K) \}$).

3.1.4 Ввести векторів даних $t_i \in T$ і відповідних вектор реакцій $r_i \in R$ (сформувані значень тестових векторів $T : \{ t_1, t_2, \dots, t_i, \dots, t_m \}$ та відповідних векторів реакцій $R : \{ r_1, r_2, \dots, r_i, \dots, r_m \}$).

3.1.5 Обчислити параметр M як кількість тестових спроб, які можуть бути виконані відповідно до наданих діагностичних даних

$$M = / T / = / R /.$$

3.1.6 Взяти значення внутрішніх лічильників i що дорівнює одиниці:

$i=1$.

3.1.7 Відповідно до формули (2.1) значення упорядкування бінарного коду хромосоми $c_i \in C$.

3.1.8 Виконати операцію інкременту для значення внутрішнього лічильника i за формулою: $i = i + 1$.

3.1.9 Якщо $i \leq M$ (якщо не всі хромосоми ще сформовані), тоді слід перейти до п. 3.1.7.

3.1.10 Візьмемо ознаку можливості дострокової повної зупину роботи алгоритму що є дійсною: $Q=1$.

3.1.11 виконати алгоритм визначення дострокової повної зупину умови алгоритму.

3.1.12 Якщо задано дострокову зупинку умову алгоритму перейти до п. 3.1.14.

3.1.13 Отримати ознаку можливості дострокової повної зупину роботи алгоритму що не є дійсною: $Q=0$.

3.1.14 Утворити одномірний масив F розміром $|C|$ для збереження обчислених значень відповідних функцій хромосоми.

3.1.15 За формулою (3.1) обчислити двійковий код для кожної хромосоми значення $c_i \in C$ відповідної функції $f_i \in F$.

3.1.16 Розташуйте значення пар $c_i \in C$ відповідно до відповідних їм значень функції придатності $f_i \in F$ в порядку спадання (від більшого до меншого).

3.1.17 Створити одномірний масив G розміром $|C|$ для збереження ознак g_i що брали участь у селекції хромосом $c_i \in C$ з відповідним значенням $g_i \in G$.

3.1.18 Візьмемо ознаку виконання схрещувань що є рівною нулю: $W=0$.

3.1.19 Візьмемо внутрішнє значення лічильника i що є рівним одиниці $i=1$.

3.1.20 Прийняти внутрішнє значення лічильника j більшим за

значення лічильника i на одиницю: $j = i + 1$.

3.1.21 Обчислити за формулою (2.2) можливість сигналу u_{ij} схрещуючи кожен хромосому з $c_i \in C$ і $c_j \in C$.

3.1.22 Якщо $u_{ij} = 0$ (схрещування вважається неможливим), слід перейти до п. 3.1.27.

3.1.23 Обчислити оператор кросингвера за формулою (2.3) значення коду хромосоми c_{ij}^1 для генерації C^1 та зафіксувати значення c_{ij}^1 у наборі C^1 .

3.1.24 Обчислити оператора схрещування векторів рекомбінації за формулою (2.4) значення хромосоми h_{ij}^1 для хромосоми c_{ij}^1 даної особи покоління C^1 та зафіксувати значення h_{ij}^1 у наборі H^1 .

3.1.25 Візьмемо код виконання для одного проходу: $W = 1$.

3.1.26 Візьмемо ознаки $g_i \in G$ і $g_j \in G$ для участі в схрещуванні хромосом $c_i \in C$ і $c_j \in C$ для одного: $g_i = 1, g_j = 1$.

3.1.27 Виконати операцію інкременту для значення внутрішнього лічильника j за формулою: $j = j + 1$.

3.1.28 Якщо $j \leq M$ (якщо не всі можливі хромосоми-партнери перебрані для хромосоми $c_i \in C$), слід перейти до п. 3.1.21.

3.1.29 Виконати операцію інкременту для значення внутрішнього лічильника i за формулою: $i = i + 1$.

3.1.30 Якщо $i < M$ (якщо даний лічильник i ще не силається на останню хромосому $c_i \in C$, тобто, ще не всі можливі схрещування реалізовані в поточному поколінні), слід перейти до п. 3.1.20.

3.1.31 Якщо $W = 0$ (ознака свідчить про відсутність в останній ітерації - схрещувань), слід перейти до п. 3.1.43.

3.1.32 Візьмемо внутрішнє значення лічильника i що є рівним одиниці: $i = 1$.

3.1.33 Перевірити здатність ознаки g_i на участь у схрещуванні хромосом $c_i \in C$.

3.1.34 Якщо $g_i=1$ (спільна хромосома $c_i \in C$ схрещується і має принаймні одне потомство, успадковане від її генів), слід перейти до п. 3.1.35.

3.1.35 Скопіювати двійковий код слова хромосомного набору $c_i \in C$ в множину C^1 і відповідний бінарний вектор рекомбінації $h \in H$ в наборі H^1 .

3.1.36 Виконати операцію інкременту значення внутрішнього лічильника i за формолою: $i = i + 1$.

3.1.37 Якщо $i \leq M$ (якщо ще не всі хромосоми перебрано), слід перейти до п. 3.1.33.

3.1.38 Замініть батьківський набір C на множину C^1 а, отже, набір H на набір H^1 .

3.1.39 Отримати ознаку можливості поаної дострокової зупинки роботи алгоритму що відповідає: Q .

3.1.40 Якщо $Q = 0$ (раннє попередження не потрібне), слід перейти до п. 3.1.14.

3.1.41 Виконати параметри корекції режиму та умови перевірки першого завершення алгоритму.

3.1.42 Якщо умова алгоритму дострокового відключення не виконана, перейти до п. 3.1.14.

3.1.43 Вивести поточний результат роботи еволюційного алгоритму.

3.1.44 Кінець алгоритму.

Розглянемо послідовність основних кроків алгоритму упорядкування виконання тестів з використанням алгоритмів на основі еволюційного методу 3.1.

Зокрема, алгоритм надає діагностичні дані, необхідні для реалізації методу замовлення тесту з використанням еволюційних алгоритмів, та основу для визначення параметрів K та M у використовуваному алгоритмі (пп. 3.1.1-3.1.5). алгоритму 3.1).

Ініціалізація еволюційного алгоритму – це формування початкової кількості хромосом для популяції.

Виходячи з основоположних засад методу організацій тестувань цифрових схем на основі еволюційних алгоритмах, вимоги до підготовки для реалізації генетичного алгоритму першого покоління ознак хромосоми C представлені у формулі (2.1) з урахуванням станів ОД $l_i \in L$, та рух відповідних векторів R , шляхом застосування ОД до блоків тестів елементарних векторів. Кожен двійковий екземпляр хромосомного коду відповідно до запропонованої математичної моделі та формули (2.1) визначається з двійкового вектора $t_i \in T$ і $r_i \in R$.

Початковий еволюційний алгоритм (початкове створення хромосоми за формулою (2.1)) реалізовано в п. 3.1.6–3.1.9 алгоритму 3.1.

Готуючи вихідну необхідну інформацію для реалізації генетичних алгоритмів (формування батьківської популяції у вигляді хромосомних копій особин першого покоління C), її робочі параметри розташовані в алгоритмі 3.1, а саме умова алгоритму ранньої роботи (пп. 3.1.10–3.1.13 алгоритму 3.1) розміщується. Характеристиками класичних генетичних алгоритмів можуть бути: обмеження кількості ітерацій, обмеження часу алгоритму тощо. Неможливо визначити умову першого алгоритму відключення, для чого необхідно, щоб алгоритм не міг отримати сигнал про можливість першого відключення ($Q = 0$ - п. 3.1.13 алгоритму 3.1).

Для початку відбору та селекцію на еволюційних алгоритм проводиться оцінка придатності числа хромосом (якщо це доцільно для ефективності реалізації ОД). Виходячи з фундаментальних принципів упорядкування тестів з використанням еволюційних алгоритмів, найбільш підходящими є особи, які мають можливість перевірити гіпотезу ОД в певному несправному стані (набір несправних станів, якщо вони можуть виявити кілька станів). Оцінка пристосованості хромосом у популяції здійснюється шляхом обчислення значень функції пристосованості за формулою (3.1).

В алгоритмі 3.1 масив F є одновимірним розмірності $/ C /$, до якого вводяться отримані значення формули (3.1) (п. 3.1.14–3.1.15 алгоритму

3.1).

Після обчислення значень функції конгруентності алгоритм 3.1 обробляє ітерації схрещування хромосом відповідно до вказівок генетичного алгоритму.

На першій перехідній стадії вибираються пари хромосом із батьківської популяції (батьківського пулу). Це тимчасова популяція, що складається з хромосом, відібраних для відбору та призначених для подальших трансформацій за допомогою операторів трансформації та мутації для формування нової популяції нащадків. Хромосоми з батьківської популяції з'єднуються, відмирають і дають нове потомство.

Кожен перехід, відповідно до принципів, визначених під час виконання методу тестування з використанням еволюційних алгоритмів, призводить лише до однієї проблеми, яка є спільною для обох дійсних (технічних і діагностичних) батьківських типів.

Найкращого потомка очікують батьки з найважливішими значеннями конгруентності функцій, тому їх бажано здати якомога швидше. Для зручності вибору в алгоритмі 3.1 значення бінарних хромосом $c_i \in C$ розташовуються відповідно до відповідних їм значень функції підгонки $f_i \in F$ в порядку спадання (від великого до мінусового – п. 3.1.16 алгоритму. 3.1). Отже, щоб вибрати найбільш підходящі хромосоми, достатньо вибрати першу хромосому з набору.

Відповідно, деякі окремі схрещування можуть взагалі не бути присутніми на кожній ітерації, але, згідно з методом упорядкування доказів за допомогою еволюційних алгоритмів, ці особини не можуть бути втрачені як надзвичайно цінні носії для реалізації хромосомних тестувальних експериментів.

Для виявлення осіб, які взагалі не беруть участь у переході в даній ітерації, а лише в алгоритмі 3.1, одновимірна G розмірності $/C/$, що забезпечує ознаки g_i та участь у схрещуванні хромосом (алгоритм 3.1 п. 3.1.17). На початку алгоритму всі стандарти $g_i \in G$ позначені без значень,

оскільки поточна ітерація ще не почалася. Під час роботи алгоритму 3.1 особи, які брали участь у переході на генетичний алгоритм, змінюють значення рядка $g_i \in G$ з нуля на одиницю (п. 3.1.26 алгоритму 3.1), що це дозволяє в подальшому; вони легко виробляють потомство від тих, хто від батьківського ярма і народження свого потомства не згоден на створення.

Кінцеву продуктивність генетичного алгоритму при реалізації запропонованого способу організації діагностичних експериментів цифрових ланцюгів необхідно виконати для отримання популяції, яка не включає подальші схрещування. Щоб визначити, які ітераційні схрещування закінчені, алгоритм III.1 використовує виконуючий код W , який дорівнює нулю на початку ітерації (алгоритм 3.1 у параграфі 3.1.18), і якщо це вдається, він змінює одне значення переходу (п. 3.1.25, див. Алгоритм 3.1).

Згідно з методом тестування з використанням алгоритмів еволюційного впорядкування, коли алгоритм 3.1 реалізується в наборі двійкових кодів C хромосоми, вибирається перша хромосома $c_i \in C$, яка відповідає кожній людині відповідно. по черзі між особинами (сигнал формується можливістю передавання кожної пари хромосом за допомогою $c_i \in C$ і $c_j \in C$ п. 3.1.21 алгоритму 3.1).

Якщо цей перехід прийняти, значення оператора транзиту (2.3) означає код хромосоми C^1 наступного покоління, а значення оператора пасажера рекомбінації (2.4) — вектор рекомбінації h_{ij}^1 для хромосом 3.1. 24 та алгоритм хромосоми C^1 3.1.24. Зазначені раніше стандарти виконання схрещень $W=1$, $g_i \in G$ і $g_j \in G$ (пункти 3.1.23–3.1.24 алгоритму 3.1) були зафіксовані в одиниці.

Перехід однієї особини $c_i \in C$ з іншими окремими особинами $c_l \in C$ ($i \neq l$) контролюється значенням j . Повний цикл кожного з хрестів в алгоритмі виконується на 3,1 бала від 3,20 до 3,28. У цьому циклі за умови (2.2) кожен з цих селекцій $c_i \in C$ обмежується тими, які менші за значення відповідності функції, і встановлюються в бінарних кодах C хромосоми

після неї. не вводьте вибрані деталі перед перетином.)

Повний цикл селекції особин (ітерація) контролюється значенням графіка G (алгоритм 3.1.1 у п. 3.1.19-3.1.30). $g_i \in G$ і $g_j \in G$ беруть участь у переході хромосом $c_i \in C$ і $c_j \in C$

Після кожної ітерації генетичного алгоритму розробляється значення знака хреста на цій ітерації (алгоритм 3.1 п. 3.1.32). Якщо на цій ітерації немає хрестів ($W = 0$), алгоритм зупиняється. У цьому випадку алгоритм припиняється, оскільки його виконання не збільшує вже досягнутого значення.

Якщо селекція відбувається хоча б один раз на цій ітерації ($W = 1$), то перевірте значення ознак участі в схрещуванні $g_i \in G$ для всіх хромосом (п. 3.1.33 алгоритму 3.1) для перевірки кожної хромосоми. остаточна ітерація може брати участь у переході (п. 3.1 17 алгоритму 3.1.34), а інші, відповідно до встановленої методики тестування з використанням еволюційних алгоритмів, можуть переходити до найближчого числа C^1 без додаткового виділення (п. 3.1.17. Алгоритм 3.1.35), і один із них копіюється в вектори рекомбінації в наборі H^1 .

В даному етапі батьківська популяція C втрачає імпульс і, як передбачає еволюційний алгоритм, вся попередня популяція хромосом замінюється новою популяцією (новоутворене потомство C^1). Алгоритм сховища популяцій виконано в Розділі 3.1.38 3.1. Популяція C^1 стає так званою популяцією, яка виконує наступну ітерацію генетичного алгоритму (або коли алгоритм перестає працювати).

Як було зазначено вище, алгоритм може бути закритий через певний час виконання або після певної кількості ітерацій. Перевірка стану алгоритму дострокового відключення реалізована в розділі 3.1.41. Алгоритм 3.1 запускається як спеціальна процедура для виправлення параметрів і перевірки умови (таким чином ми можемо проаналізувати час роботи алгоритму або кількість ітерацій; змінюються значення ітераційних розрахунків та виконуються інші вимоги до реалізації методу тестування з

використанням алгоритмів розробки).

Якщо умова припинення не настає, то наступний перехід до наступної ітерації генетичного алгоритму (перехід з п. 3.1.42 до алгоритму 3.1 до п. 3.1.14). В іншому випадку переходимо до кінцевої стадії – похідної від «найкращого» встановленої хромосоми – для поточного набору алгоритмів C (п. 3.1.43 алгоритму 3.1).

Слід зазначити, що при утворенні першої умови алгоритму подання, складаючи кількість необхідних ітерацій для формування нових популяцій, алгоритм 3.1 буде чітко працювати. При формуванні першої умови алгоритму, обмежуючи час виконання, роботу алгоритму можна виділити до кінця інтервалу ітерації. Можливо, можна внести зміни до Алгоритму 3.1 для більш жорстких часових обмежень, але є ризик зупинити алгоритми в середині ітерації та втратити результати неповного обчислення (повернення до останньої повністю завершеної ітерації).

Узагальнена блок-схема алгоритму організації тестувань цифрових схем на основі еволюційних алгоритмів для реалізації запропонованого методу організації тестувань цифрових схем представлена на рис. 3.1.

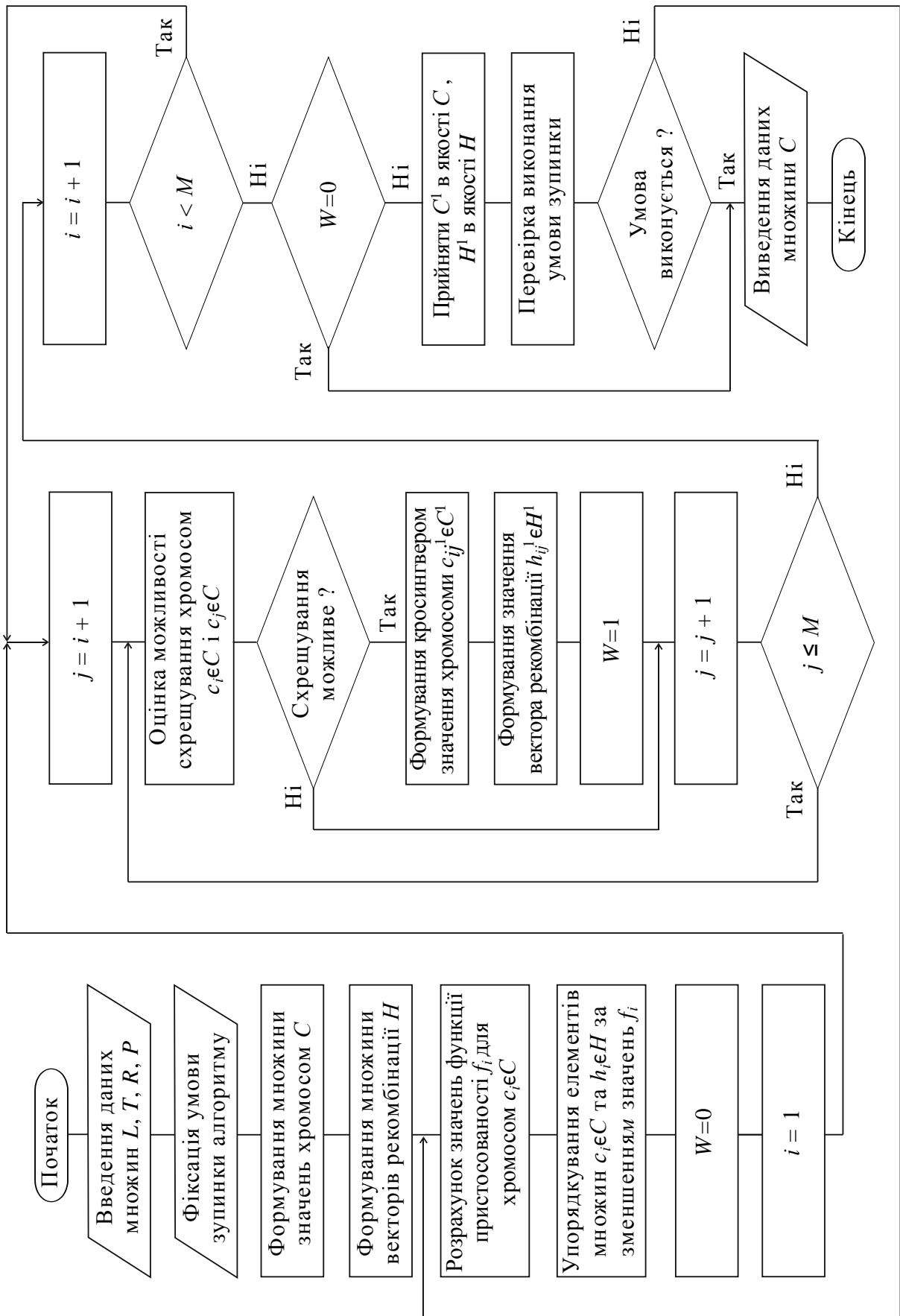


Рисунок 3.1 – Узагальнена блок-схема реалізації алгоритму методу

3.3 Висновки

У третьому розділі наведено опис основних рекомендацій і принципів реалізації алгоритмів методу організації тестів із застосуванням еволюційних алгоритмів.

На початку були сформульовані основні положення методики:

- використаний вид еволюційних алгоритмів – гібридний (має найменшу кількість обмежень та дає можливість подальшого використанні різноманітних існуючих методів для оптимізації отримуваних результатів);

- математичний апарат – так званна математична модель M і формули (2.1)-(2.4), (3.1);

- спосіб утворення кодів хромосом (з множини C) – утворення значень $c_i \in C$ зі значень $t_i \in T$ і $r_i \in R$ ($i = 1, 2 \dots |T|$) за допомогою формули (2.1);

- спосіб застосування методу схрещування - пропорційний ймовірності адаптації функції (3.1) і для контролю придатності функцій. (2.2);

- кросинговер - рівномірний тип оператора (формула (2.3));

- спосіб утворення нового покоління – перехресної популяції нащадків на основі результатів відбору пропорцій та елітарно відібраних особин з контрольної функції сумісності (2.2), які не допускають селекції;

- основний спосіб запобігання необхідності еволюційного алгоритму - втрата здатності керувати переходом від одного покоління до іншого;

- додаткові (або можливі) способи зупинки алгоритму – обмеження часу виконання або кількості ітерацій (утвореного покоління).

Відповідно до основних рекомендацій щодо організації тестувань цифрових схем на основі еволюційних алгоритмах детально описано реалізацію алгоритму та запропоновано узагальнену блок-схему для нього.

4 ПРАКТИЧНИЙ МЕТОД ТЕСТУВАНЬ ЦИФРОВИХ СХЕМ НА ОСНОВІ ЕВОЛЮЦІЙНИХ АЛГОРИТМІВ

4.1 Приклад практичного застосування методу перевірки цифрових схем

Для затвердження розробленого методу організації тестів із використанням еволюційних методів розглянемо приклад його застосування на основі експериментальних даних.

Для проведення експерименту будемо використовувати набір множини T і R з чотирнадцяти елементів, кожен з яких спрямований на виявлення певної несправності об'єкта в якості вихідних даних.

Тобто, для експерименту вибирається набір з чотирнадцяти несправних станів об'єкта діагностування $L: \{l_1, l_2, l_3, l_4, l_5, l_6, l_7, l_8, l_9, l_{10}, l_{11}, l_{12}, l_{13}, l_{14}\}$.

Набір P значень статистичних даних ймовірності знаходження об'єкта діагностування в кожному із можливих технічних станів $l_i \in L$ характеризується наступними значеннями: $p_1=0.05$; $p_2=0.07$; $p_3=0.03$; $p_4=0.01$; $p_5=0.08$; $p_6=0.10$; $p_7=0.04$; $p_8=0.01$; $p_9=0.02$; $p_{10}=0.06$; $p_{11}=0.005$; $p_{12}=0.007$; $p_{13}=0.003$; $p_{14}=0.001$.

Початковий набір тестових векторів розроблено для ідентифікації станів ОД $l_i \in L$ тест-векторів $T: \{t_1, t_2, t_3, t_4, t_5, t_6, t_7, t_8, t_9, t_{10}, t_{11}, t_{12}, t_{13}, t_{14}\}$, представлена елементами: $t_1=01x101xx$; $t_2=0x1001x1$; $t_3=0x1101x0$; $t_4=01x100xx$; $t_5=0x1101xx$; $t_6=0xx101x1$; $t_7=0111xxxx$; $t_8=11xx11xx$; $t_9=111111xx$; $t_{10}=xxx10110$; $t_{11}=110101xx$; $t_{12}=111001x1$; $t_{13}=111101x0$; $t_{14}=11x1001x$.

Початковий набір векторів відповіді $R: \{r_1, r_2, r_3, r_4, r_5, r_6, r_7, r_8, r_9, r_{10}, r_{11}, r_{12}, r_{13}, r_{14}\}$, отримуваний шляхом представлення на ОД блоків тест-векторів $t_i \in T$, розроблених для ідентифікації стану ОД $l_i \in L$, представлений елементами: $r_1=01xx$; $r_2=0101$; $r_3=1x11$; $r_4=010x$; $r_5=0x1x$; $r_6=0xx1$; $r_7=0xx0$; $r_8=1xx1$; $r_9=1xxx$; $r_{10}=xxx1$;

$r_{11}=01x0$; $r_{12}=1101$; $r_{13}=1011$; $r_{14}=000x$.

На основі наведених даних множини тестових векторів $T:\{t_1, t_2, t_3, t_4, t_5, t_6, t_7, t_8, t_9, t_{10}, t_{11}, t_{12}, t_{13}, t_{14}\}$ та набору векторів відповідних реакцій $R:\{r_1, r_2, r_3, r_4, r_5, r_6, r_7, r_8, r_9, r_{10}, r_{11}, r_{12}, r_{13}, r_{14}\}$ за формулою (2.1) сформуємо транзитивне представлення множини хромосом початкової популяції

$C^*:\{c^*_1, c^*_2, c^*_3, c^*_4, c^*_5, c^*_6, c^*_7, c^*_8, c^*_9, c^*_{10}, c^*_{11}, c^*_{12}, c^*_{13}, c^*_{14}\}$ для реалізації генетичного алгоритму:

$c^*_1=01x101xx01xx$;	$c^*_2=0x1001x10101$;	$c^*_3=0x1101x01x11$;
$c^*_4=01x100xx010x$;	$c^*_5=0x1101xx0x1x$;	$c^*_6=0xx101x10xx1$;
$c^*_7=0111xxxx0xx0$;	$c^*_8=11xx11xx1xx1$;	$c^*_9=111111xx1xxx$;
$c^*_{10}=xxx10110xxx1$;	$c^*_{11}=110101xx01x0$;	$c^*_{12}=111001x11101$;
$c^*_{13}=111101x01011$;	$c^*_{14}=11x10011000x$.	

В якості векторів рекомбінації для реалізації генетичних методів з рівномірним кросинговером ми співставимо кожному вектору хромосоми $c^*_i \in C^*$ вектор-маску h_i подібної розрядності, тобто, формуємо множину векторів рекомбінації $H:\{h_1, h_2, h_3, h_4, h_5, h_6, h_7, h_8, h_9, h_{10}, h_{11}, h_{12}, h_{13}, h_{14}\}$. У векторі рекомбінації $h_i \in H$ значення кожного розряду будемо визначати як одиниці, якщо відповідний цифра в коді хромосоми $c^*_i \in C^*$ відповідає сильному гену, який не може бути втрачений при кросинговері, і дорівнює нулю – в іншому випадку (якщо відповідна цифра хромосоми $c^*_i \in C^*$ у відповідному розряді містить x).

Таким чином, ми отримуємо набір векторів рекомбінації $H:\{h_1, h_2, h_3, h_4, h_5, h_6, h_7, h_8, h_9, h_{10}, h_{11}, h_{12}, h_{13}, h_{14}\}$ з такими значеннями елементів $h_i \in H$: $h_1=110111001100$; $h_2=101111011111$; $h_3=101111011011$; $h_4=110111001110$; $h_5=101111001010$; $h_6=100111011001$; $h_7=111100001001$; $h_8=110011001001$; $h_9=111111001000$; $h_{10}=000111110001$; $h_{11}=111111001101$; $h_{12}=111111011111$; $h_{13}=111101011111$; $h_{14}=110111111110$.

З урахуванням отриманих значень векторів рекомбінації, які зберігають опис позицій сильних і слабких генів хромосом $c^*_i \in C^*$, можна усунути в кодах хромосом $c^*_i \in C^*$ ознаку невизначеності значень окремих

розрядів і перейти до класичного двійкового представлення хромосом (при переході міняємо символ x на 0). Таким чином, виконаємо завершальну операцію в формуванні множини хромосом початкової популяції для реалізації генетичного алгоритму $C:\{c_1, c_2, c_3, c_4, c_5, c_6, c_7, c_8, c_9, c_{10}\}$:
 $c_1=010101000100$; $c_2=001001010101$; $c_3=001101001011$; $c_4=010100000100$;
 $c_5=001101000010$; $c_6=000101010001$; $c_7=011100000000$; $c_8=110011001001$;
 $c_9=111111001000$; $c_{10}=000101100001$; $c_{11}=110101000100$; $c_{12}=111001011101$;
 $c_{13}=111101001011$; $c_{14}=110100110000$.

В якості функцій адаптивності f_i , відповідно до принципів реалізації методу організації тестування з використанням еволюційних алгоритмів, ми використовуємо дані статистичних імовірностей $p_i \in P$ знаходження ОД у кожному з технічних станів $l_i \in L$.

Відповідно із алгоритмом реалізації методу виконуємо впорядкування хромосом вихідної популяції $C:\{c_1, c_2, c_3, c_4, c_5, c_6, c_7, c_8, c_9, c_{10}\}$ у порядку зменшення пристосованості (табл. 4.1).

За формулою (2.2) обчислимо функцію сумісності хромосоми $c_6 \in C$, яка має найбільше значення функції адаптації, з іншими хромосомами $c_i \in C$:

$$y_{6-5} = (000101010001 \oplus 001101000010) \wedge 100111011001 \wedge 101111001010 = \\ = 000000000000;$$

$$y_{6-2} = (000101010001 \oplus 001001010101) \wedge 100111011001 \wedge 101111011111 = \\ = 000100000000;$$

$$y_{6-10} = (000101010001 \oplus 000101100001) \wedge 100111011001 \wedge 000111110001 = \\ = 000000010000;$$

$$y_{6-1} = (000101010001 \oplus 010101000100) \wedge 100111011001 \wedge 110111001100 = \\ = 000000000000;$$

$$y_{6-7} = (000101010001 \oplus 011100000000) \wedge 100111011001 \wedge 111100001001 = \\ = 000000000001;$$

$$y_{6-3} = (000101010001 \oplus 001101001011) \wedge 100111011001 \wedge 101111011011 = \\ = 000000011000;$$

$$y_{6-9} = (000101010001 \oplus 111111001000) \wedge 100111011001 \wedge 111111001000 = \\ = 100010001000;$$

$$y_{6-4} = (000101010001 \oplus 010100000100) \wedge 100111011001 \wedge 110111001110 = \\ = 000001000000;$$

$$y_{6-8} = (000101010001 \oplus 110011001001) \wedge 100111011001 \wedge 110011001001 = \\ = 100010001000.$$

Таблиця 4.1 – Упорядковані дані початкової популяції

i	f_i	$c_i \in C$	$h_i \in H$
6	0.10	000101010001	100111011001
5	0.08	001101000010	101111001010
2	0.07	001001010101	101111011111
10	0.06	000101100001	000111110001
1	0.05	010101000100	110111001100
7	0.04	011100000000	111100001001
3	0.03	001101001011	101111011011
9	0.02	111111001000	111111001000
4	0.01	010100000100	110111001110
8	0.01	110011001001	110011001001
12	0.007	111001011101	111111011111
11	0.005	110101000100	111111001101
13	0.003	111101001011	111101011111
14	0.001	110100110000	110111111110

Умова $y_{ij} = 0$ для вибору в якості першої батьківської особини хромосом $c_6 \in C$ є справедливою тільки для двох пар хромосом: $c_6 \in C$ і $c_5 \in C$; $c_6 \in C$ і $c_1 \in C$.

Виконаємо схрещування цих хромосом із застосуванням оператора схрещування (2.3):

$$c_{6-5} = 000101010001 \vee 001101000010 = 001101010011;$$

$$c_{6-1} = 000101010001 \vee 010101000100 = 010101010101.$$

Виконуємо аналогічний перетин векторів рекомбінації за допомогою оператора (2.4):

$$h_{6-5} = 100111011001 \vee 101111001010 = 101111011011;$$

$$h_{6-1} = 100111011001 \vee 010101000100 = 110111011101.$$

Наступним кроком, відповідно до запропонованого алгоритму реалізації методу організації тестування із застосуванням еволюційних алгоритмів, є виключення з наступного розгляду батьківської особини, представленої хромосомою $c_6 \in C$.

Після виключення з наступного розгляду батьківської особини, представленої хромосомою $c_6 \in C$, найбільше значення функції пристосованості характерне хромосомі $c_5 \in C$.

За формулою (2.2) обчислимо значення функції сумісності хромосоми $c_5 \in C$ з іншими хромосомами $c_i \in C$, що дасть три нульових значення $y_{ij} = 0$: $y_{5-1} = 0$; $y_{5-7} = 0$; $y_{5-10} = 0$.

Виконаємо почергове схрещування хромосоми $c_5 \in C$ із сумісними хромосомами $c_1 \in C$, $c_7 \in C$ і $c_{10} \in C$ за допомогою оператора кросингвера (2.3):

$$c_{5-10} = 001101000010 \vee 000101100001 = 011100000000;$$

$$c_{5-1} = 001101000010 \vee 010101000100 = 010101000100;$$

$$c_{5-7} = 001101000010 \vee 011100000000 = 011100000000.$$

Виконуємо подібне схрещування векторів рекомбінації за застосуванням оператора (2.4):

$$h_{5-10} = 101111001010 \vee 000111110001 = 111100001001;$$

$$h_{5-1} = 101111001010 \vee 110111001100 = 110111001100;$$

$$h_{5-7} = 101111001010 \vee 111100001001 = 111100001001.$$

Наступним кроком за запропонованим алгоритмом реалізації методу організації тестування із використанням еволюційних алгоритмів, є виключення із подальшого розгляду батьківської особини, представленої хромосоною $c_5 \in C$.

Після виключення з наступного розгляду батьківської особини, представленої хромосоною $c_5 \in C$, найбільшим значенням функції пристосованості є характерне хромосомі $c_2 \in C$.

За формулою (2.2) виконаємо обчислення значень функції сумісності хромосоми $c_2 \in C$ з іншими $c_i \in C$, що не дає жодного результату, якої дорівнював би нулю – схрещування хромосоми $c_2 \in C$ з іншими хромосомами $c_i \in C$ не є можливим. Виключаємо з наступного розгляду особину, представлену хромосоною $c_2 \in C$.

Після виключення її подальшого розгляду батьківської особини, що представлена хромосоною $c_2 \in C$, її найбільше значення функції пристосованості характерне даній хромосомі: $c_{10} \in C$.

За формулою (2.2) виконаємо обчислення значення функції сумісності хромосоми $c_{10} \in C$ із іншими хромосомами $c_i \in C$, що дає два значення рівних нулю $y_{ij} = 0$: $y_{10-1} = 0$; $y_{10-3} = 0$.

Виконуємо почергове схрещування хромосом $c_{10} \in C$ з сумісними їй хромосомами $c_1 \in C$ та $c_3 \in C$ із застосуванням оператора кросингвера (2.3):

$$c_{10-1} = 000101100001 \vee 010101000100 = 010101100101;$$

$$c_{10-3} = 000101100001 \vee 001101001011 = 001101101011.$$

Виконуємо аналогічні схрещування векторів рекомбінації із застосуванням оператора (2.4):

$$h_{10-1} = 000111110001 \vee 110111001100 = 110111111101;$$

$$h_{10-3} = 000111110001 \vee 101111011011 = 101111111011.$$

Наступним кроком, за запропонованим алгоритмом реалізації методу організації тестування із застосуванням еволюційних алгоритмів, є виключення з подальшого розгляду батьківської особини, що представлена хромосоною

$c_{10} \in C$.

Після вилучення з подальшого розгляду батьківської особини, представлені хромосомами $c_{10} \in C$, максимальне значення функції пристосованості характерні хромосомі $c_1 \in C$.

За формулою (2.2) виконаємо обчислення значень функцій сумісностей хромосоми $c_1 \in C$ з іншими хромосомами $c_i \in C$, що дає лише одне значення рівне нулю $y_{1-7} = 0$.

Виконуємо схрещування хромосом $c_1 \in C$ із сумісною їй хромосомою $c_7 \in C$ з застосуванням кросингвера (2.3):

$$c_{1-7} = 010101000100 \vee 011100000000 = 011101000100.$$

Виконуємо аналогічне схрещування векторів рекомбінації з використанням оператора (2.4):

$$h_{1-7} = 110111001100 \vee 111100001001 = 111111001101.$$

Наступним кроком, згідно з запропонованим алгоритмом є реалізації методу організації тестування з застосуванням еволюційних алгоритмів, є виключення із подальшого розгляду батьківської особини, що представлена хромосомою $c_1 \in C$.

Після вилучення з подальшого розгляду батьківської особини, що представлена хромосомою $c_1 \in C$, та має найбільше значення функції пристосованості, що характерне хромосомі $c_7 \in C$.

За формулою (2.2) виконаємо обчислення значень функції сумісності хромосоми $c_7 \in C$ з іншими хромосомами $c_i \in C$, що дає лише одне нульове значення $y_{7-4} = 0$.

Виконуємо схрещування хромосоми $c_7 \in C$ із сумісною їй хромосомою $c_4 \in C$ з застосуванням оператора кросингвера (2.3):

$$c_{7-4} = 011100000000 \vee 010100000100 = 010100000100.$$

Виконаємо аналогічне схрещування векторів рекомбінації з застосуванням оператора (2.4):

$$h_{7-4} = 111100001001 \vee 110111001110 = 110111001110.$$

Наступним кроком, згідно із запропонованим алгоритмом реалізації методу організації тестів із застосуванням еволюційних алгоритмів, є виключення з подальшого розгляду батьківської особини, представленої хромосомою $c_7 \in C$.

Після виключення з подальшого розгляду батьківської особини, представленої хромосомою $c_7 \in C$, найбільше значення функції пристосованості характерне хромосомі $c_3 \in C$.

За формулою (2.2) виконуємо обчислення значення функції сумісності хромосоми $c_3 \in C$ із іншими хромосомами $c_i \in C$, які не дають жодного результату, що рівний нулю – схрещування хромосоми $c_3 \in C$ із іншими хромосомами $c_i \in C$ не є можливим. Виключаємо із подальшого розгляду особу, представлену хромосомою $c_3 \in C$.

Після виключення з подальшого розгляду батьківської особини, представленої хромосомою $c_3 \in C$, найбільше значення функції пристосованості характерне хромосомі $c_9 \in C$.

За формулою (2.2) виконаємо обчислення значень функцій сумісності хромосоми $c_9 \in C$ із іншими хромосомами $c_i \in C$, це дасть лише одне нульове значення $y_{9-8} = 0$.

Виконаємо схрещування хромосоми $c_9 \in C$ з сумісною їй хромосомою $c_8 \in C$ із застосуванням оператора кросингвера (2.3):

$$c_{9-8} = 111111001000 \vee 110011001001 = 111111001001.$$

Виконуємо аналогічне схрещування векторів рекомбінації із застосуванням оператора (2.4):

$$h_{9-8} = 111111001000 \vee 110011001001 = 111111001001.$$

Наступним кроком, що запропонований алгоритмом реалізації методу організації тестування з застосуванням еволюційних алгоритмів, є виключення із подальшого розгляду батьківської особини, представленої хромосомою $c_9 \in C$.

Після вилучення із подальшого розгляду батьківської особини, що

представлена хромосомою $c_9 \in C$, найбільше значення функції пристосованості характерне хромосомі $c_4 \in C$.

За формулою (2.2) виконаєм обчислення значення функції сумісності хромосоми $c_4 \in C$ із іншими хромосомами $c_i \in C$, що не дають жодного результату, який рівний нулю – схрещування хромосоми $c_4 \in C$ з іншими хромосомами $c_i \in C$ не є можливим. Виключаємо із подальшого розгляду особину, що представлену хромосомою $c_4 \in C$.

Аналогічним способом, обчислюєм значення функцій сумісності хромосом $c_4 \in C$, $c_8 \in C$, $c_{11} \in C$, $c_{12} \in C$ і $c_{13} \in C$ із іншими хромосомами $c_i \in C$, що не дає жодного результату який рівний нулю – схрещування хромосоми $c_4 \in C$, $c_8 \in C$, $c_{11} \in C$, $c_{12} \in C$ і $c_{13} \in C$ із іншими хромосомами $c_i \in C$ які не є можливим. Виключаєм із подальшого розгляду особини, представлені хромосомами $c_4 \in C$, $c_8 \in C$, $c_{11} \in C$, $c_{12} \in C$ і $c_{13} \in C$.

Після вилучення з подальшого розгляду батьківської особини, представленої хромосомою $c_{13} \in C$ в масиві C залишається одина особина $c_{14} \in C$, у якої партнерів для схрещування немає – процес схрещувань та, відповідно, формування осіб-нащадків нового покоління на першій ітерації алгоритму завершується.

Аналіз характеру виконаних схрещувань показав, що хромосоми $c_2 \in C$, $c_{11} \in C$, $c_{12} \in C$, $c_{13} \in C$ і $c_{14} \in C$ не було застосовано в жодній операції схрещувань – переносимо хромосоми $c_2 \in C$, $c_{11} \in C$, $c_{12} \in C$, $c_{13} \in C$ і $c_{14} \in C$ до нового покоління за принципом елітарного відбору.

Таким чином, отримуємо в результаті першої ітерації генетичного алгоритму згідно з даним принципом реалізації методу організації тестування з застосуванням еволюційних алгоритмів в яких отримано двійкові слова кодів хромосом першого покоління нащадків:

$$C^1: \{c_{6-5}, c_{6-1}, c_{5-10}, c_{5-1}, c_{5-7}, c_{10-1}, c_{10-3}, c_{1-7}, c_{7-4}, c_{9-8}, c_2, c_{12}, c_{11}, c_{13}, c_{14}\}.$$

Згідно з запропонованим алгоритмом реалізації методу організації тестування з застосуванням еволюційних алгоритмів, приймаємо множину C^1 в

якості вихідного масива для реалізації другої ітерації алгоритму.

За формулою (3.1) обчислимо значення функції пристосованості кожної особини $c_i \in C$.

Результати обрахунків в упорядкованому порядку представлено в таблиці 4.2.

Таблиця 4.2 – Упорядковані дані першої популяції нащадків

i	f_i	$c_i \in C$	$h_i \in H$
6-5	0.18	001101010011	101111011011
6-1	0.15	010101010101	110111011101
5-10	0.14	001101100011	101111111011
5-1	0.13	011101000110	111111001110
5-7	0.12	011101000010	111111001011
10-1	0.11	010101100101	110111111101
10-3	0.09	001101101011	101111111011
1-7	0.09	011101000100	111111001101
2	0.07	001001010101	111111111111
7-4	0.05	011100000100	111111001111
9-8	0.03	111111001001	111111001001
12	0.007	111001011101	111111011111
11	0.005	110101000100	111111001101
13	0.003	111101001011	111101011111
14	0.001	110100110000	110111111110

Другим етапом даного алгоритму реалізації методу організації тестування із застосуванням еволюційних алгоритмів виконується аналогічно першій, через це не будемо зупинятись на детальному покроковому описі його виконання, а розглянемо лише основні отримувані результати.

Обчислення значення функції сумісності хромосом $c_i \in C$ за формулою

(2.2) дає три результати рівних нулю: $y_{(6-5)-(6-1)} = 0$, $y_{(5-10)-(5-1)} = 0$, $y_{(5-1)-(5-7)} = 0$.

Виконання схрещування сумісних хромосом з застосуванням оператора кросингвера (2.3) дозволило нам отримати 3 хромосоми особин нового покоління:

$$c_{(6-5)-(6-1)} = 001101010011 \vee 010101010101 = 011101010111;$$

$$c_{(5-10)-(5-1)} = 001101100011 \vee 011101000110 = 011101000110;$$

$$c_{(5-1)-(5-7)} = 011101000110 \vee 011101000010 = 011101000010.$$

Виконаємо аналогічні схрещування інших векторів рекомбінації з застосуванням оператора (2.4):

$$h_{(6-5)-(6-1)} = 101111011011 \vee 110111011101 = 111111011111;$$

$$h_{(5-10)-(5-1)} = 101111111011 \vee 111111001110 = 111111001111;$$

$$h_{(5-1)-(5-7)} = 111111001110 \vee 111111001011 = 111111001011.$$

Аналіз характеристик виконаних схрещувань показав, що є ще множина хромосом $C: \{ c_{10-1}, c_{10-3}, c_{1-7}, c_{7-4}, c_{9-8}, c_2 \}$, які не застосовано в жодній операції схрещувань – виконаєм перенесення хромосоми $c_i \in C$ до нового покоління за принципом елітарного відбору.

Таким чином, це дозволить другій ітерації генетичного алгоритму згідно з принципами реалізації методу організації тестування з застосуванням еволюційних алгоритмів отримали двійкові слова кодів хромосом 2 покоління нащадків (для зручності дублювання у нумерації індексів усунуто та числові значення упорядковані, індекс типу (5-10)-(5-1) приведено до наступного вигляду 1-5-10):

$$C: \{ c_{1-5-6}, c_{1-5-10}, c_{1-5-7}, c_{1-10}, c_{3-10}, c_{1-7}, c_{4-7}, c_{8-9}, c_2 \}.$$

Результати розрахунків у упорядкованому вигляді представлено в даній таблиці 4.3.

Таблиця 4.3 – Упорядковані дані другої популяції нащадків

i	f_i	$c_i \in C$	$h_i \in H$
1-5-6	0.33	011101010111	111111011111

1-5-10	0.27	011101000110	111111001111
1-5-7	0.25	011101000010	111111001011
1-10	0.11	010101100101	110111111101
3-10	0.09	001101101011	101111111011
1-7	0.09	011101000100	111111001101
4-7	0.05	011100000100	111111001111
8-9	0.03	111111001001	111111001001
2	0.07	001001010101	111111111111
12	0.007	111001011101	111111011111
11	0.005	110101000100	111111001101
13	0.003	111101001011	111101011111
14	0.001	110100110000	110111111110

Спроба виконання 3 ітерації алгоритму реалізації методу організації тестування з застосуванням еволюційних алгоритмів за даними які дані в таблиці 4.3 засвідчила про відсутність можливості подальших схрещувань, що дає підстави для зупинки роботи генетичного алгоритму.

Завершальним етапом виконання алгоритму може бути вибір найбільш пристосованих хромосом, що дозволять вирішити цільові завдання (задач технічної діагностики), але, як зазначалося у огляді, особливістю застосованого гібридного варіанту генетичних алгоритмів є те, що можливість використовувати до отримуваних результатних класичних методів оптимізації, характерних для сфери застосування генетичних алгоритмів.

Такий підхід є виправдним у більшій мірі, тому завершальним кроком застосування даною методу організації тестування з застосуванням еволюційних алгоритмів є зворотній перехід від 2-вих слів кодів хромосом та векторів рекомбінації до діагностичних векторів тестованих впливів та відповідних реакцій.

Вихідними даними на даному етапі є множини двійкових слів кодів

хромосом $c_i \in C$ та векторів рекомбінації $h_i \in H$, які описує в таблиці 4.3.

Перетворення буде виконуватися в зворотному порядку відносно порядку отримання початкових множин 2-вих слів кодів хромосом C та векторів рекомбінації H .

На основі цих двійкових слів кодів хромосом $c_i \in C$, з застосуванням векторів рекомбінації $h_i \in H$ ми визначимо позиції розрядів, що можуть приймати у діагностичних векторах довільні значення та позначимо їх символом «х» (формування перехідного представлення множини $C^* = \{c^*_{1-5-6}, c^*_{1-5-10}, c^*_{1-5-7}, c^*_{1-10}, c^*_{3-10}, c^*_{1-7}, c^*_{4-7}, c^*_{8-9}, c^*_2\}$):

$$c^*_{1-5-6} = 011101x10111;$$

$$c^*_{1-5-10} = 011101xx0110;$$

$$c^*_{1-5-7} = 011101xx0x10;$$

$$c^*_{1-10} = 01x1011001x1;$$

$$c^*_{3-10} = 0x1101101x11;$$

$$c^*_{1-7} = 011101xx01x0;$$

$$c^*_{4-7} = 011100xx0100;$$

$$c^*_{8-9} = 111111xx1xx1;$$

$$c^*_2 = 0x1001x10101;$$

$$c^*_{11} = 110101xx01x0;$$

$$c^*_{12} = 111001x11101;$$

$$c^*_{13} = 111101x01011;$$

$$c^*_{14} = 11x10011000x.$$

На основі даного перехідного представлення множини двійкових слів кодів хромосом $C^* = \{c^*_{1-5-6}, c^*_{1-5-10}, c^*_{1-5-7}, c^*_{1-10}, c^*_{3-10}, c^*_{1-7}, c^*_{4-7}, c^*_{8-9}, c^*_2\}$ формуємо множину діагностичних тестових векторів $T = \{t_{1-5-6}, t_{1-5-10}, t_{1-5-7}, t_{1-10}, t_{3-10}, t_{1-7}, t_{4-7}, t_{8-9}, t_2\}$ та множину векторів відповідних реакцій $R = \{r_{1-5-6}, r_{1-5-10}, r_{1-5-7}, r_{1-10}, r_{3-10}, r_{1-7}, r_{4-7}, r_{8-9}, r_2\}$.

Множина діагностичних тестових векторів T буде складатися із елементів:

$$t_{1-5-6} = 011101x1;$$

$$t_{1-5-10}=011101xx;$$

$$t_{1-5-7}=011101xx;$$

$$t_{1-10}=01x10110;$$

$$t_{3-10}=0x110110;$$

$$t_{1-7}=011101xx;$$

$$t_{4-7}=011100xx;$$

$$t_{8-9}=111111xx;$$

$$t_2=0x1001x1;$$

$$t_{11}=110101xx;$$

$$t_{12}=111001x1;$$

$$t_{13}=111101x0;$$

$$t_{14}=11x1001x.$$

А множина векторів відповідних реакцій R буде складатися з:

$$r_{1-5-6}=0111;$$

$$r_{1-5-10}=0110;$$

$$r_{1-5-7}=0x10;$$

$$r_{1-10}=01x1;$$

$$r_{3-10}=1x11;$$

$$r_{1-7}=1x0;$$

$$r_{4-7}=0100;$$

$$r_{8-9}=1xx1;$$

$$r_2=0101;$$

$$r_{11}=01x0;$$

$$r_{12}=1101;$$

$$r_{13}=1011;$$

$$r_{14}=000x.$$

Моделювання нашого діагностичного експерименту на основі отриманих даних дає зрозуміти, що для виявлення усіх несправностей, опис яких надано в обраній нами для постановки експерименту множині із десяти несправних станів ОД $L:\{l_1, l_2, l_3, l_4, l_5, l_6, l_7, l_8, l_9, l_{10}\}$, достатньо

використати замість десяти початково наданих тестових векторів лише 5 організованих на основі методу організації тестування з застосуванням еволюційних алгоритмів:

$$t_{1-5-6}=011101x1;$$

$$t_{3-10}=0x110110;$$

$$t_{4-7}=011100xx;$$

$$t_{8-9}=111111xx;$$

$$t_2=0x1001x1;$$

$$t_{11}=110101xx;$$

$$t_{12}=111001x1;$$

$$t_{13}=111101x0;$$

$$t_{14}=11x1001x.$$

Отримані нами результати підтверджують ефективність положень та алгоритмів реалізації розробленого методу організації тестування цифрових схем на основі розроблених еволюційних алгоритмів. Подальше підвищення ефективності даного процесу тестування електронних схем з використанням організованих на основі еволюційних алгоритмів цифрових тестів можна очікувано досягнути застосуванням методів оптимізації послідовностей діагностичних тестових векторів, що описуються в попередніх роботах [1, 11-14, 21].

4.2 Висновки

Останній розділ присвячений питанням практичного застосування методів організації тестування цифрових схем на основі еволюційних алгоритмів. Демонстрування принципів застосування методу проведено на прикладах, що змодельовані процесом постановки реальних тестових експериментів.

При експериментальній реалізації розробленого методу організації

тестувань цифрових схем на основі еволюційних алгоритмів, нами було отримане зменшення кількості тестів, для яких необхідне проведення перевірки об'єкта тестувань, на 35.7 відсотків.

Отримані нами результати підтверджують ефективність положення і алгоритмів реалізацій розробленого методу організації тестування цифрових схем, що основані на еволюційних алгоритмах. Подальше підвищення ефективності процесів тестування електронних схем з застосуванням еволюційних алгоритмів цифрових тестів можна буде очікувано досягти із застосуванням методів оптимізації послідовності діагностичних тестових векторів, які описуються в роботах [1, 11-14, 21].

ВИСНОВКИ

У роботі за результатами проведених теоретичних та практичних досліджень розроблено метод організації тестування цифрових схем на основі еволюційних алгоритмів.

При розробці методу ставилася мета підвищити ефективність організації цифрових вузлів, що тестуються, за рахунок зменшення розмірності використовуваних тестових послідовностей за рахунок використання еволюційних алгоритмів на етапі організації тестування цифрових схем..

Об'єктом дослідження є процес тестової діагностики цифрових схем, предметом дослідження є методи організації тестування цифрових схем з підвищеною здатністю до тестування.

Для реалізації програми досліджень виконано такі роботи.

У першому розділі проводиться дослідження особливостей організації тестування цифрових схем і методів їх організації, а також можливостей еволюційних алгоритмів як інструменту для організації тестування цифрових схем. Дослідження показало перспективність використання еволюційних алгоритмів зменшення розмірності тестів, необхідні організації тестування цифрових схем, що відповідає мети магістерської роботи. За результатами дослідження було сформульовано завдання.

У другому розділі приведені алгоритми та розроблено математичну модель методу організації тестування цифрових схем на основі еволюційних алгоритмів.

У третьому розділі описано метод організації тестування цифрових схем з урахуванням еволюційних алгоритмів. Спочатку розглянуто опис основних положень методу, основа яких була початком для розроблення алгоритмів реалізації методу та дано їх докладний опис.

У четвертому розділі описується практичне застосування методу організації тестувань цифрових схем на основі еволюційних алгоритмів, розглядається приклад застосування розробленого методу на практиці та його алгоритмів, надано опис програмної реалізації методу.

Реалізація результатів розробленого методу дозволила підтвердити ефективність методу та його реалізаційних алгоритмів.

За темою магістерської роботи опубліковано 2 тези доповідей на конференції в Києві - військовий інститут університету Шевченко.

ПЕРЕЛІК ДЖЕРЕЛ ПОСИЛАННЯ

1. Абдуллаев П. Ш. Применение генетических алгоритмов при диагностировании авиационных ГТД / П. Ш. Абдуллаев, А. Дж. Мирзоев // Авіаційно-космічна техніка і технологія. – Харків : ХАІ, 2016. – № 7(134)– С. 139-146.

2. Жердєв М.К. Концептуальні засади методу діагностування сучасних цифрових типових елементів заміни по форматним частотам перехідного процесу в шині живлення / М.К. Жердєв, В.О. Савран // Зб. наук. праць Військового інституту Київського НУ ім. Тараса Шевченка. – К. : ВІКНУ, 2016. – Вип. 52. – С. 20-32.

3. Ленков Є.С. Узагальнена математична модель процесу технічного обслуговування і ремонту складної техніки / Є.С. Ленков // Вісник Хмельницького національного університету. Технічні науки. – Хмельницький : 51, 2017. – № 2 (247). – С. 186-191.

4. Шевченко В.В. Контроль технічного стану цифрових типових елементів заміни за допомогою електромагнітного методу (способу) діагностування / В. В. Шевченко // Системи обробки інформації. - Харків: ХАІ, 2015. - Вип. 8. - С. 57-59.

5. Сівак В.А. Методи функціональної належності компонентів діагностичних засобів, які використовують технологію нечіткої логіки / В.А. Сівак // Вісник Хмельницького національного університету. Технічні науки. – Хмельницький : 51, 2015. – № 6 (231). – С. 203-208.

6. Чорненький В.І. Алгоритми діагностування цифрових пристроїв апаратними засобами на базі штучних нейронних мереж / В.І. Чорненький, В.М. Чешун // Зб. наук. праць Військового інституту Київського НУ ім. Тараса Шевченка. – К. : ВІКНУ, 2010. – Вип. 27. – С. 154–161.

7. Рутковская Д. Нейронные сети, генетические алгоритмы и нечеткие системы / Д. Рутковская, М. Пилиньский, Л. Рутковский. - М.: Горячая линия-Телеком, 2006. - 452 с.

8. Prinetto P. An automatic test pattern generator for larges equential circuits based on genetic algorithms / P. Prinetto, M. Rebaudengo, M. Sonza Reorda // Proc. Int. Test Conf. – 1994. – P.240–249.

9. Rudnick E.M. Sequential Circuit Test Generation in a Genetic Algorithm Framework / E.M. Rudnick, J.H. Patel, G.S. Greenstein, T.M. Niermann // Proc. Design Automation Conf. – 1994. – P.698–704.

10. Городилов А.Ю. Генетический алгоритм диагностирования цифровых устройств / А.Ю. Городилов // Вестник Пермского национального исследовательского политехнического университета. Электротехника, информационные технологии, системы управления. – Пермь: ПНИПУ, 2013. – № 7. – С. 54-62.

11. Дубровин В.И. Диагностика на основе генетических алгоритмов / В.И. Дубровин, Е. Н. Федорченко // Радіоелектроніка, Інформатика, Управління. – Запоріжжя: ЗНТУ, 2006. – № 2. – С. 115-120.

12. Попов В.А. Оптимизационные задачи на основе генетического поиска / В.А. Попов, А.В. Бердочник // Системи обробки інформації – Харків: ХУПС, 2010. – Вип. 9 (90). – С.217-220.

13. Соколова Э.С. Оптимизация коэффициента глубина поиска дефектов методом генетических алгоритмов / Э.С. Соколова, С.Н. Капранов // Контроль. Диагностика. – М. : ООО "Издательский дом "Спектр", 2004. – №4. – С. 32-39.

14. Иванов Д.Е. Генетические алгоритмы построения входных идентифицирующих последовательностей цифровых устройств. / Д.Е. Иванов – Донецк: ТОВ «Цифровая типография», 2012. – 240 с.

15. Миронов С.В. Генетические алгоритмы для сокращения диагностической информации / С.В. Миронов, Д.В. Сперанский // Автоматика и телемеханика. – М. : Академиздатцентр «Наука» РАН, 2008. – No7. – С.146-156.

16. Дмитриев Д.В. Адаптация генетических алгоритмов к решению задач назначения точек контроля в объектах с большим числом состояний /

Д.В. Дмитриев, Э.С. Соколова, С.Н. Капранов // Нейрокомпьютеры, М: Издательство «Радиотехника» № 11, 2007. – С.59-64.

17. Евдокимов А.А. Применение символьных вычислений к исследованию дискретных моделей некоторых классов генных сетей / А.А. Евдокимов, С.Е. Кочемазов, А.А. Семенов // Вычислительные технологии. – М. : НИВЦ МГУ, 2011. – Т. 16. № 1. - С. 30-47.

18. Juang C.F. A TSK-Type Recurrent Fuzzy Net-work for Dynamic Systems ceasing by Neural Network and Genetic Algorithms / C.F. Juang //IEEE Trans. on Fuzzy Systems. 2002. – Vol. 10, Issue 2. – P. 155-170.

19. Тюрин С.Ф. Разработка контрольных и диагностических тестов для КМОП элементов с избыточным базисом / С.Ф. Тюрин, О.А. Громов // Приволжский научный вестник. – Ижевск : ИЦНП, 2013. –№ 1 (17). – С.13-21.

20. Согомоян Е.С. Построение дискретных устройств с диагностикой в процессе функционирования / Е.С. Согомоян // Автоматика и телемеханика. - №11. - 1971. С.153-160.

21. Тестовый контроль микропроцессорных БИС на производстве /Г.М.Ясинявичене, Б.В.Бургис, Е.А.Мецаев и др.- М.:Радио и связь,1989.- 120 с.

22. Буков В.Н. Три подхода к задаче контроля технического состояния / В.Н. Буков, И.М. Максименко // Автоматика и телемеханика.- 1995.- N3.- С. 165-178.

23. Гроль В.В. Синтез контролепригодных цифровых схем / В.В. Гроль– М.:ИСМО, 1996.- 105с.

24. Преснухин Л.Н. Цифровые вычислительные машины: Учебное пособие для студентов вузов. / Л.Н. Преснухин, П.В. Нестеров – 2-е изд. – М.: Высшая школа, 1981.-511с.

25. Королев Ю.В. Проблемы макро моделирования интегральных хем. / Ю.В. Королев, И.А. Ющенко //Диэлектрики и полупроводники. - 1984. - Вып.26.

26. Митюк В.Г. Маршевый тест ОЗУ для обнаружения связанных неисправностей взаимного влияния / В.Г. Митюк, В.Н. Ярмолик // Управляющие системы и машины. – 1996. – №3. – С. 44-48.

27. Байда Н.П. Метод поиска дефектов в шинных структурах / Н.П. Байда, В.Н. Локазюк, С.И. Перевозников // Автоматизация контроля вычислительных устройств и систем: тез. докл. республиканской научно-технической конференции. – К. - 1988.- С. 18-19.

28. Рувинский В.М. Тестирование мультипроцессорных вычислительных структур / В.М. Рувинский // Электронное моделирование.- 1989.- №6.- С.54-58.

29. Иоффе М.И. Диагностирование логических схем. Алгоритмы моделирования и автоматического синтеза теста. / Иоффе М.И. - М.:Наука, 1989.-158с.

30. Локазюк В.Н. Комбинированное диагностирование устройств с компонентами повышенной степени интеграции. / В.Н. Локазюк, В.А. Карякин // Тез. доп. I-ї Української конференції з автоматичного керування. «Автоматика-94». - Київ.- 1994. - С.493.

31. Лихтциндер Б.Я. Внутрисхемное диагностирование узлов радиоэлектронной аппаратуры. / Б.Я. Лихтциндер - К.:Техника,1988.-168с.

32. Основы технической диагностики // В.В.Карибский, П.П.Пархоменко, Е.С. Согомонян, В.Ф.Халчев. - М.: Энергия, 1976. - 464с.

35. Гуляев В.А. Техническая диагностика управляющих систем. / В.А. Гуляев. - К.:Наукова думка, 1983.- 17 с.

36. Romankevich A. On Digital Circuits Checkability under Pseudo Random Testing / A. Romankevich, V. Groll //XII International Conf. on Fault Tolerant Systems and Diagnostics.- Praha.- 1989.- P. 217-218.

37 Romankevich A. On Testability of Checkable Digital Circuits under Pseudorandom Signals / A. Romankevich, V. Groll //The European Design and Test Conference.- Paris.- 1995.- P. 602.

38. Бедренко С.В. О вероятностном подходе к проверке последовательностных устройств / С.В. Бедренко, А.Ю. Матросова // Автоматика и телемеханика.- 1980.- N1.- С.97-103.

39. Жуков М.В. Метод построения тестов дискретных устройств с помощью настраиваемого генератора случайных входных воздействий / М.В. Жуков // Автоматика и телемеханика.- 1980.- N4.- С. 122-131.

40. Дяченко О.Н. Способы выделения диагностической информации из сигнатур / О.Н. Дяченко, А.Н. Тарасенко // Электронное моделирование.- 1990.- №5.- С.64-70.

41. Ярмолик В.Н. Эффективность сигнатурного анализа в самотестирующихся СБИС / В.Н. Ярмолик, Е.П. Калоша // Электронное моделирование.- 1992.-№3.- С.51-56.

42. Гессель М. Построение кодоразделительных самопаритетных комбинационных схем для самотестирования и функционального диагностирования / М. Гессель, Е.С. Согомоян // Автоматика и телемеханика.- 1996.- N11.- С. 155-165.

43. Данилов В.В. Модели диагностирования и методы тестирования цифровых схем / В.В. Данилов // Электронная техника. - Сер 8.-1983. - Вып. 6.- С.22-27.

44. Байда Н.П. Микропроцессорные системы поэлементного диагностирования РЭА. / Н.П. Байда, И.В. Кузьмин, В.Г. Шпилевой - М.:Радио и связь, 1987.- 256с.

45. Локазюк В.Н. Тестирование БИС программируемых периферийных адаптеров в составе микропроцессорного устройства / В.Н. Локазюк // Обмен программно-техническим опытом. - 1987.- N 11.- С.55-58.

46. Калявин В.П. Технические средства диагностирования / В.П. Калявин, А.В. Мозгалевский - Л.:Судостроение,1984.- 210 с.

47. Жирабок А.Н. Диагностирование логико-динамических систем / А.Н. Жирабок // Электронное моделирование. - 1996.- №6.- С.48-57.

48. Кондратьев В.В. Контрольно-диагностический автоматизированный комплекс / В.В. Кондратьев, В.П. Сидоренко, О.Д. Руккас, М.С. Берштейн // Приборы и системы управления.- 1981. - N5.- С.4-6.

49. Кудряшов В.И. Сигнатурные анализаторы как средство автоматизации наладки и эксплуатации дискретной техники / В.И. Кудряшов, Г.Х. Новик // Приборы и системы управления. - 1984.- N3.- С.5-6.

50. Глибовець М.М. Гібридний генетичний алгоритм вирішення задачі оптимізації структури інтегральної схеми / М.М. Глибовець, С.С. Гороховський, О.В. Краткова. // Інженерія програмного забезпечення. - 2011.- N1.- С70-76.

51. Рутковская Д. Нейронные сети, генетические алгоритмы и нечеткие системы / Д. Рутковская, М. Пилиньский, Л. Рутковский– М.: Горячая линия – Телеком, 2006. – 452 с.

52. Генетичний алгоритм [Електронний ресурс] / Портал «Вільна енциклопедія Вікіпедія». – Режим доступу: https://uk.wikipedia.org/wiki/Генетичний_алгоритм (дата звернення 30.09.2018). – Назва з екрана.

53. Категорія Генетичні алгоритми [Електронний ресурс] / Портал «Вільна енциклопедія Вікіпедія». – Режим доступу: https://uk.wikipedia.org/wiki/Категорія:Генетичні_алгоритми (дата звернення 28.11.2018). – Назва з екрана.

54. Генетичні алгоритми. Ключові поняття і методи [Електронний ресурс] / Портал «Портал знань». – Режим доступу: http://www.znannya.org/?view=ga_general (дата звернення 30.09.2018). – Назва з екрана.

55. Основні поняття генетичних алгоритмів [Електронний ресурс] / Портал «StudFiles». – Режим доступу: <https://studfiles.net/preview/5465767/> (дата звернення 30.09.2018). – Назва з екрана.

56 Калініна. І.В. Використання генетичних алгоритмів в задачах

оптимізації /Л.В. Калініна, О.І. Лісовиченко// Міжвідомчий науково-технічний збірник «Адаптивні системи автоматичного управління».- 2015.- №1.- С48-61.

57. Кононюк А.Ю. Нейронні мережі і генетичні алгоритми / А.Ю. Кононюк – К.:«Корнійчук». 2008. – 446 с.

58. Бадіміна Л.А. Генетичні алгоритми в задачах технічної діагностики дискретних систем / О.І. Стецюк, В.М. Чешун // Тези доповідей Всеукраїнської науково-практичної конференції молодих вчених, ад'юнктів, слухачів, курсантів і студентів "Молодіжна військова наука у Київському національному університеті імені Тараса Шевченка" /за заг. редакцією І.В.Толока. – К.:ВІКНУ, 2018. – С. 112.

59. Бадіміна Л.А. Генетичні алгоритми в синтезі діагностичних тестів для цифрових пристроїв / Л.А. Бадіміна, В.М. Чешун, В.І. Чорненький // Міжнародний науково-технічний журнал «Вимірювальна та обчислювальна техніка в технологічних процесах». – Хмельницький: ХНУ, 2018. – Вип. №2 (62) – С.92-98.

60. Бадіміна Л.А. Синтез дискретних діагностичних тестів із застосуванням генетичних алгоритмів / Л.А. Бадіміна, В.М. Чешун // «Інтелектуальний потенціал – 2018» - збірник наукових праць молодих науковців і студентів з нагоди 30-річчя підготовки ІТ- фахівців в ХНУ/ Колектив авторів – Хмельницький: ПВНЗ УЕП, 2018. – Ч.3: Кібербезпека та актуальні проблеми комп'ютерних систем і мереж – С.5-9.

61. Бадіміна Л.А. Генетичні алгоритм синтезу дискретних тестів / Л.А. Бадіміна, В.М. Чешун // Студентський науковий вісник ПВНЗ "УЕП": матеріали студентських конференцій, грудень 2018р. – ПВНЗ УЕП, 2018. – С.35-37.

Додаток А
(обов'язковий)
Копії публікацій

Таким чином, можливо стверджувати, що актуальність правового врегулювання права використання об'єктів подвійного призначення, незалежно від форми власності, цілком очевидна. Захист суверенітету та державної цілісності України, не належить виключно, до повноважень складових сил оборони.

*к.т.н., доц. Чешун В.М. (ХмНУ)
Моторний П.В. (ХмНУ)*

Тестування цифрових схем на основі еволюційних алгоритмів

Збільшення функціональних можливостей і складності цифрових електронних схем загострює актуальність питання автоматизованого синтезу якісних тестів для їх діагностування, що призводить до потреби застосування новітніх методів вирішення цієї задачі. Одним із таких методів є постановка діагностичних експериментів на основі інтелектуальних підходів, зокрема, на основі еволюційних алгоритмів.

Аналіз принципів дії еволюційних алгоритмів [1] дає чотири основних операції, що дозволяють отримати результат, а саме: формування початкової популяції, відбір батьківських особин, схрещування і мутація. Основною операцією для отримання більш якісного рішення є схрещування, що виконується над двійковими кодами хромосом із застосуванням кросингвера.

Спосіб тестування цифрових схем на основі еволюційних алгоритмів базується на наступних основних принципах (або положеннях):

- застосований вид еволюційних алгоритмів – гібридний;
- спосіб генерації двійкових слів кодів хромосом – оператором схрещування зі значень діагностичних векторів елементарних тестів;
- спосіб відбору претендентів для схрещування – пропорційний, на основі ймовірнісної функції пристосованості і контрольного оператора оцінки сумісності;
- тип оператора кросингвера – рівномірний;
- спосіб формування нового покоління – нащадки схрещувань особин за результатами пропорційного відбору і елітарно відібрані особини, що не допускають схрещувань за контрольним оператором оцінки сумісності;
- основний спосіб зупинки роботи еволюційного алгоритму – втрата можливості реалізації схрещувань особин в отриманому поколінні;
- додаткові (можливі) способи зупинки роботи алгоритму – обмеження на час виконання або на кількість ітерацій (генерованих поколінь).

Алгоритм реалізації діагностичного експерименту містить етапи формування початкової популяції, схрещування, обробки отримуваних результатів і перевірки умови зупинки роботи генетичного алгоритму.

Список використаних джерел:

1. Шадура О. В. Модифікація генетичних алгоритмів на основі методу нецентрованих головних компонент та стандартні тести / О. В. Шадура // World science. – 2019. – № 4(44), Vol.1. – С. 4-10.
2. Бадіміна Л.А. Генетичні алгоритми в синтезі діагностичних тестів для цифрових пристроїв / Л.А. Бадіміна, В.М. Чешун, В.І. Чорненький // Вимірвальна та обчислювальна техніка в технологічних процесах. – 2018. – № 2. – С. 92-99.

АЛГОРИТМ ТЕСТУВАННЯ ЦИФРОВИХ ЕЛЕКТРОНИХ СХЕМ НА
ОСНОВЕ ЕВОЛЮЦІЙНИХ ПРОЦЕДУР

Моторний Паско Володимирова

студентка групи КПІ-20-1, ХНУ

Національний університет «Харківський національний університет імені
Г.С.Сквирського»

к.м.н., доцент, декан кафедри кібернетики

Збільшення функціональних можливостей і складності цифрових електронних схем змушує актуалізувати питання автоматизованого синтезу живих тестів для їх діагностування, що призводить до потреби застосування нових методів вирішення цієї задачі.

Основні і нові методи є постановка діагностичних експериментів на основі інтелектуальних підходів, зокрема, на основі еволюційних алгоритмів.

Автори пропонують дві еволюційні алгоритми [1, 2] для чотирьох основних етапів, що дозволяють отримати результати, а саме:

- формування початкової популяції,
- вибір батьківських особин,
- схрещування,
- мутація.

Основною перевагою для отримання більш якісного рішення є схрещування, що виконується над двійковими кодами хромосом і застосування кроссовера.

Спосіб тестування цифрових схем на основі еволюційних алгоритмів базується на застосуванні основних принципів [3]:

- застосування вид еволюційних алгоритмів – гібридний;
- спосіб генерації двійкових сіт кодів хромосом – оператори схрещування та вищого діагностичного вектора елементарних тестів;
- спосіб вибору пресидентів для схрещування – проєкційний, на основі функцій пристоєвості і контрольного оператора ознаки суцільності;
- тип оператора кроссовера – рівномірний;

3

алгоритму:

1.12 Якщо ознаку дострокового зупинку алгоритму задано (обмеження на кількість ітерацій, час роботи алгоритму тощо), перейти до п. 1.14.

1.13 Прийняти ознаку можливості дострокового зупинку роботи алгоритму не дійсною: $Q=0$.

1.14 Сформулювати одинірний масив F розмірності $|C|$ для зберігання розрахункових значень функцій пристосованості $f \in F$ хромосом $c \in C$.

1.15 Для кожного двійкового коду хромосоми $c \in C$ обчислити значення функції пристосованості $f \in F$.

1.16 Упорядкувати значення двійкових кодів хромосом $c \in C$ за зростаючим (а значенням функції пристосованості $f \in F$ в порядку зменшення (від більшого до меншого).

1.17 Сформулювати одинірний масив G розмірності $|C|$ для зберігання ознак g участі у схрещуваннях хромосом $c \in C$ з нульовими значеннями елементів $g \in G$.

1.18 Прийняти ознаку реалізації схрещування рівною нулю: $W=0$.

1.19 Прийняти значення внутрішнього лічильника i рівним одиниці (вказівник на першу хромосому $c_1 \in C$): $i=1$.

1.20 Прийняти значення внутрішнього лічильника j більшим на одиницю за значення лічильника i (вказівник на наступну після $c_1 \in C$ хромосому $c_j \in C$): $j=i+1$.

1.21 Розрахувати ознаку z_j можливості схрещування особин з хромосомами $c_i \in C$ і $c_j \in C$.

1.22 Якщо $z_j=0$ (схрещування визначено неможливим), перейти до п. 1.27.

1.23 Розрахувати за формулою оператора кроссовера значення коду хромосоми c_j' особи наступного покоління C' і зафіксувати значення c_j' в масиві C' .

1.24 Розрахувати за формулою оператора схрещування вектора рекомбінації значення вектора рекомбінації h_j' для коду хромосоми c_j' особи

3

- спосіб формування нового покоління – випадок схрещування, особин за результатами проєкційного вибору і випадок вибору особин, що не допускають схрещування і контролюють оператором ознаки суцільності.

- основний спосіб зупинки роботи еволюційного алгоритму – вправа можливості реалізації схрещування особин з отриманою популяцією.

- додатковий (можливо) спосіб зупинки роботи алгоритму – обмеження на кількість ітерацій або на кількість ітерацій (генераций поколінь).

Для реалізації діагностичного експерименту на основі еволюційних підходів пропонується алгоритм 1, що базується на запропонованій [4] математичній моделі.

Алгоритм 1.

1.1 Ввести дані про можливі несправні стани ОД $L \in L$.

1.2 Визначити K як кількість можливих несправних станів ОД:

$$K=|L|.$$

1.3 Ввести дані статистичних ймовірностей $P(i)$ походження ОД в кожному з можливих несправних станів $i \in L$ (сформулювати відповідні значення статистичних даних ймовірностей $P(P(i_1)), P(i_2), \dots, P(i_3), \dots, P(i_K)$).

1.4 Ввести дані тест-вектора $c \in T$ і вектора відповідних реакцій $r_i \in R$ (сформулювати відповідні значення тест-вектора $T=(t_1, t_2, \dots, t_n, \dots, t_m)$ і вектора відповідних реакцій $R=(r_1, r_2, \dots, r_n, \dots, r_m)$).

1.5 Визначити параметр M як кількість тестових випробувань, що можуть бути реалізовані на основі заданих діагностичних даних.

$$M=|T| \cdot |R|.$$

1.6 Прийняти значення внутрішнього лічильника i рівним одиниці: $i=1$.

1.7 Синтезувати значення двійкового коду хромосоми $c \in C$.

1.8 Виконати інкремент значення внутрішнього лічильника $i: i=i+1$.

1.9 Якщо $i \leq M$ (якщо сформовано ще не всі хромосоми), перейти до п. 1.7.

1.10 Прийняти ознаку можливості дострокового зупинку роботи алгоритму дійсною: $Q=1$.

1.11 Реалізувати процедуру вимірювання ознаки дострокового зупинку

3

покоління C' і зафіксувати значення h_j' в масиві H' .

1.25 Прийняти ознаку реалізації схрещування рівною одиниці: $W=1$.

1.26 Прийняти ознаки $g \in G$ і $g' \in G$ участі у схрещуваннях хромосом $c \in C$ і $c' \in C'$ рівними одиниці: $g=1, g'=1$.

1.27 Виконати інкремент значення внутрішнього лічильника $j: j=j+1$.

1.28 Якщо $j \leq M$ (якщо перебрано ще не всі можливі хромосоми-партнери для хромосом $c \in C$), перейти до п. 1.21.

1.29 Виконати інкремент значення внутрішнього лічильника $i: i=i+1$.

1.30 Якщо $i < M$ (якщо лічильник i ще не вказує на останню хромосому $c \in C$, тобто, не всі можливі схрещування в поточному поколінні реалізовані), перейти до п. 1.20.

1.31 Якщо $W=0$ (ознака реалізації схрещування свідчить про відсутність схрещування в останній ітерації), перейти до п. 1.43.

1.32 Прийняти значення внутрішнього лічильника i рівним одиниці: $i=1$.

1.33 Перевірити ознаку g участі у схрещуваннях хромосом $c \in C$.

1.34 Якщо $g=1$ (хромосома $c \in C$ прийняла участь у схрещуваннях і має показуєтьє одного випадка, що унеаслідують її ген), перейти до п. 1.35.

1.35 Скопіювати слово двійкового коду хромосоми $c \in C$ в множини C' і слово відповідного її двійкового коду вектора рекомбінації $h \in H$ в множини H' .

1.36 Виконати інкремент значення внутрішнього лічильника $i: i=i+1$.

1.37 Якщо $i \leq M$ (якщо перебрано ще не всі хромосоми), перейти до п. 1.33.

1.38 Виконати зміну батьківської множини C на множини C' і, відповідно, множини H на множини H' .

1.39 Перевірити ознаку ознаку можливості дострокового зупинку роботи алгоритму: Q .

1.40 Якщо $Q=0$ (дострокового зупинку не перебрано), перейти до п. 1.14.

1.41 Реалізувати процедуру корекції параметрів та перевірки виконання ознаки дострокового зупинку роботи алгоритму.

4

1.42 Якщо умова дострокової зупинки роботи алгоритму не виконується, перейти до п. 1.14.

1.43 Вивести поточну множину S як результат роботи алгоритму.

1.44 Кінець алгоритму.

Запропонований алгоритм містить етапи формування початкової популяції, схрещування, обробки отримуваних результатів і перевірки умови зупинки роботи генетичного алгоритму.

Для підтвердження ефективності розробленого алгоритму було проведено його апробацію на реальних діагностичних даних. В результаті застосування запропонованого алгоритму було отримано скорочений набір тестів, деякі з яких набули здатність виявляти декілька несправностей, що дозволило скоротити тестову послідовність майже на 36 відсотків.

Список використаних джерел

1. Шадура О. В. Модифікація генетичних алгоритмів на основі методу нецентрованих головних компонент та стандартні тести / О. В. Шадура // World science. – 2019. – № 4(44), Vol.1. – С. 4-10.

2. An Improved Genetic Algorithm for the Optimal Distribution of Fresh Products under Uncertain Demand / Hao Zhang, Yan Cui, Hepu Deng, Shuxian Cui, Huijia Mu. // Mathematics, 9(18), 2021. – 18 p.

3. Чешун В.М. Тестування цифрових схем на основі еволюційних алгоритмів / В.М. Чешун, П.В. Моторний // Тези доповідей XVII Міжнародної науково-практичної конференції "Військова освіта і наука: сьогодення та майбутнє", 26 листоп. 2021 р. – Київ : ВІКНУ, 2021. – Т. 1. – С. 63.

4. Бадіміна Л.А. Генетичні алгоритми в синтезі діагностичних тестів для цифрових пристроїв / Л.А. Бадіміна, В.М. Чешун, В.І. Чорненький // Вимірвальна та обчислювальна техніка в технологічних процесах. – 2018. – № 2. – С. 92-99.

Хмельницький національний університет
Факультет програмування та комп'ютерних і телекомунікаційних систем
Кафедра кібербезпеки та комп'ютерних систем і мереж

Моторний Павло Володимирович

Метод організації тестувань цифрових схем на основі еволюційних алгоритмів

спеціальність 123 – Комп'ютерна інженерія

Науковий керівник: к.т.н., доцент **Чешун Віктор Миколайович**

Додаток Б
(обов'язковий)
Презентація роботи

ЗАГАЛЬНА ХАРАКТЕРИСТИКА МАГІСТЕРСЬКОЇ РОБОТИ

Мета магістерської роботи полягає у підвищенні ефективності організації тестувань цифрових вузлів скороченням розмірності використовуваних тестових послідовностей на основі застосування еволюційних алгоритмів на етапі організації тестувань цифрових схем.

Об'єктом дослідження є процес тестового діагностування цифрових схем.

Предметом дослідження є методи організації тестувань з підвищеною перевіряючою здатністю.

Задачі досліджень у роботі формуються наступним чином:

- а) виявити аналогії у вирішуваних задачах технічної діагностики і теорії еволюційних алгоритмів для підвищення ефективності організації тестувань цифрових вузлів скороченням розмірностей використовуваних тестових послідовностей;
- б) визначити основні положення методу організації тестувань цифрових схем на основі еволюційних алгоритмів;
- в) розробити математичну модель для реалізації методу організації тестувань цифрових схем на основі еволюційних алгоритмів;
- г) розробити алгоритмічне забезпечення для реалізації методу.

Методи досліджень базуються на основних положеннях технічної діагностики, теорії тестового діагностування, теорії еволюційних алгоритмів, теорії множин, теорії ймовірностей, булевій алгебрі.

Наукова новизна отриманих результатів:

1. Уточнено математичну модель діагностичного експерименту з урахуванням специфіки вирішення задачі організації тестувань на основі теорії еволюційних алгоритмів;
2. Розроблено метод організації дискретних діагностичних тестів підвищеної перевіряючої здатності, що забезпечується застосуванням еволюційних алгоритмів для надання тестам властивості одночасної реалізації декількох елементарних перевірок об'єкта діагностування.

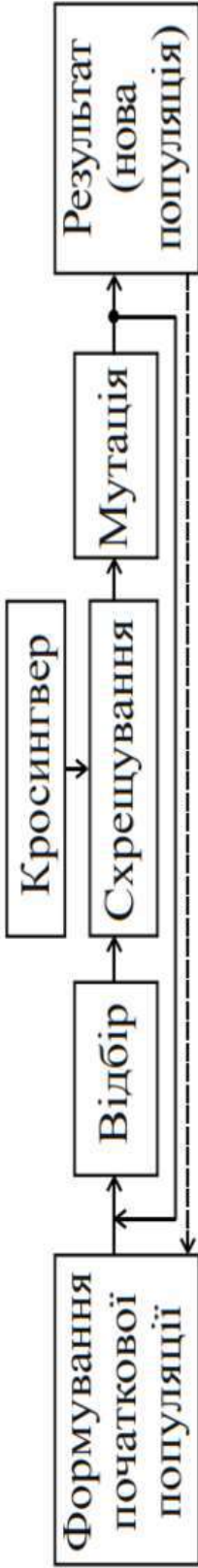
Практична цінність розробленого методу полягає у визначенні принципів і розробці алгоритмів синтезу дискретних тестів із застосуванням генетичних алгоритмів, які надають змогу збільшити ефективність організації процесу діагностування цифрових пристроїв за рахунок зменшення розмірностей використовуваних тестових послідовностей.

Апробація роботи. Наукові результати і основні положення магістерської роботи доповідались і обговорювались на конференції в Києві - військовий інститут університету Шевченка.

Публікації. За темою магістерської роботи опубліковано 2 тези доповідей на конференції в Києві - військовий інститут університету Шевченка.

ДОСЛІДЖЕННЯ ВЛАСТИВОСТЕЙ ЕВОЛЮЦІЙНИХ АЛГОРИТМІВ

Узагальнена схема дії еволюційного алгоритму (ГА)



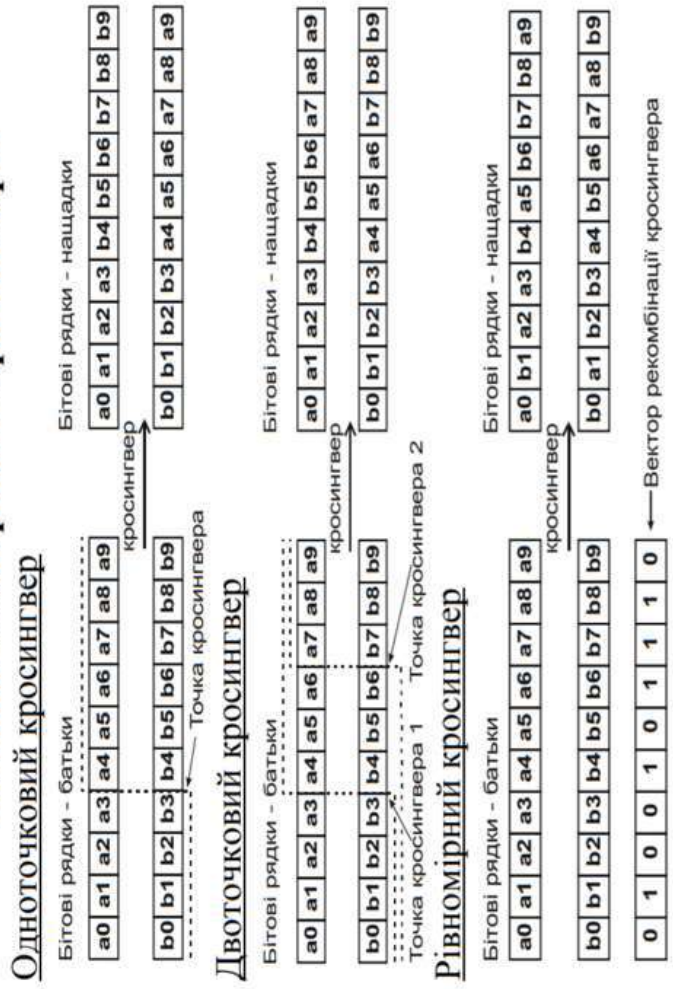
Ключові - робота з двійково-кодованими представленнями параметрів задач;

властивості ГА: - застосування імовірнісних оціночних функцій як критерію пристосованості.

Основні види ГА та їх властивості

Вид ГА	Простий	Дженітор	Гібридний
Параметр	Простий	Дженітор	Гібридний
Формат чисел	Двійкові натуральні	Двійкові натуральні	Двійкові дійсні
Відбір батьків	Випадковий	Випадковий	За пристосованістю
Кросингвер	Одно-точковий	Немає обмежень	Немає обмежень
Формування нової популяції	Нашадки заміщують батьків	Нашадки заміщують найслабших	Вибір за специфікою задач
Оптимізація результатів	Не передбачена	Не передбачена	Класичні методи

Основні схеми роботи кросингверів



МАТЕМАТИЧНА МОДЕЛЬ МЕТОДУ

Загальне представлення

$$M = \langle L, P, T, R, C, H \rangle.$$

де

- $L: \{l_1, l_2, \dots, l_b, \dots, l_k\}$ - множина можливих несправних технічних станів ОД;
- $P: \{P(l_1), P(l_2), \dots, P(l_i), \dots, P(l_k)\}$ - множина значень статистичних даних імовірності знаходження ОД в кожному з можливих технічних станів $l_i \in L$;
- $T: \{t_1, t_2, \dots, t_b, \dots, t_m\}$ - множина тест-векторів, розроблених для ідентифікації станів ОД $l_i \in L$;
- $R: \{r_1, r_2, \dots, r_b, \dots, r_m\}$ - множина отримуваних векторів відповідних реакцій при поданні на ОД блоку тест-векторів $t_i \in T$;
- $C: \{c_1, c_2, \dots, c_b, \dots, c_m\}$ - множина хромосом початкової популяції для реалізації генетичного алгоритму;
- $H: \{h_1, h_2, \dots, h_b, \dots, h_m\}$ - множина векторів рекомбінації хромосом $c_i \in C$.

Основні оператори

Оператор формування коду хромосоми $c_i \in C$ з діагностичних векторів $t_i \in T$ і $r_i \in R$:

$$c_i = \text{text}(t_i) + \text{text}(r_i) \quad (1)$$

Оператор оцінювання можливості схрещування особин, представлених хромосомами $c_i \in C$ і $c_j \in C$:

$$y_{ij} = (c_i \oplus c_j) \wedge h_i \wedge h_j \quad (2)$$

- $= 0$ – схрещування хромосом $c_i \in C$ і $c_j \in C$ можливе;
- $\neq 0$ – схрещування хромосом $c_i \in C$ і $c_j \in C$ неможливе.

Оператор кросингвера (оператор схрещування хромосом $c_i \in C$ і $c_j \in C$):

$$c_{ij}^1 = c_i \vee c_j \quad (3)$$

Оператор кросингвера векторів рекомбінації $h_i \in H$ і $h_j \in H$:

$$h_{ij}^1 = h_i \vee h_j \quad (4)$$

Функція пристосованості особини з кодом хромосоми $c_i \in C$:

$$f_j = \sum_{i=1}^{|L|} q_i P(l_i) \quad (5)$$

де $|L|$ – розмірність L ; $P(l_i)$ – імовірність знаходження ОД в стані $l_i \in L$; q_i – ознака здатності тест-вектора $t_j \in T$ виявляти ознаки технічного стану $l_i \in L$ ($q_i = 1$ за наявності у тест-вектора $t_j \in T$ відповідної здатності, $q_i = 0$ за відсутності у тест-вектора $t_j \in T$ відповідної здатності)

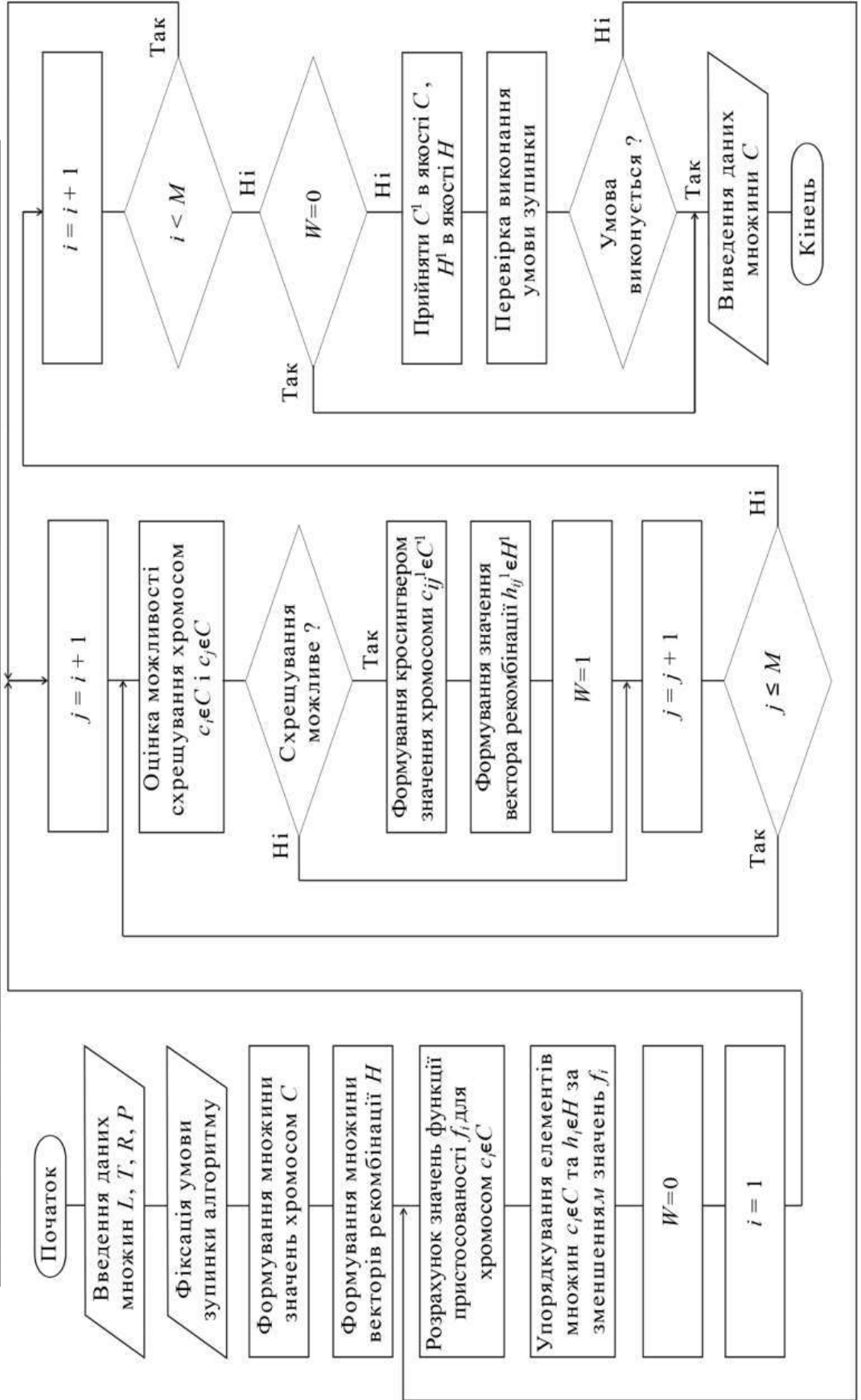
Другий науковий результат

МЕТОД ОРГАНІЗАЦІЇ ТЕСТІВ – ОСНОВНІ ПОЛОЖЕННЯ

- застосований вид еволюційних алгоритмів – гібридний (має найменше обмежень і передбачає можливість подальшого застосування різноманітних існуючих методів оптимізації отримуваних результатів);
- математичний апарат – математична модель M , оператори (1)-(5);
- спосіб генерації двійкових слів кодів хромосом (множини C) – формування значень $c_i \in C$ зі значень діагностичних векторів $t_i \in T$ і $r_i \in R$ ($i=1,2 \dots |T|$) оператором (1);
- спосіб відбору претендентів для схрещування – пропорційний, на основі ймовірнісної функції пристосованості (5) і контрольного оператора оцінки сумісності (2);
- тип оператора кросінгвера – рівномірний (3);
- спосіб формування нового покоління – нащадки схрещувань особин за результатами пропорційного відбору і елітарно відібрані особини, що не допускають схрещувань за контрольним оператором оцінки сумісності (2);
- основний спосіб зупинки роботи генетичного алгоритму – втрата можливості реалізації схрещувань особин в отриманому поколінні;
- додаткові (можливі) способи зупинки роботи алгоритму – обмеження на час виконання або на кількість ітерацій (генерованих поколінь).

Другий науковий результат

МЕТОД ОРГАНІЗАЦІЇ ТЕСТІВ – АЛГОРИТМІЧНА РЕАЛІЗАЦІЯ



ЕКСПЕРИМЕНТАЛЬНА АПРОБАЦІЯ МЕТОДУ

Початкові дані

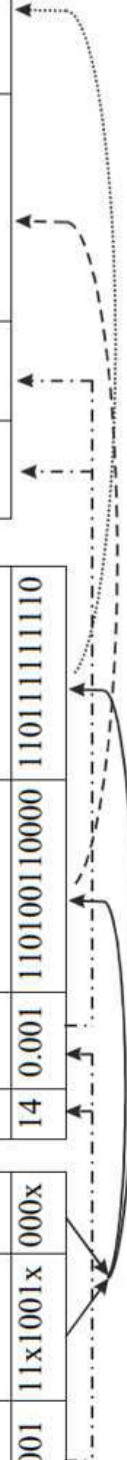
i	$p_i \in P$	$t_i \in T$	$r_i \in R$
1	0.05	01x101xx	01xx
2	0.07	0x1001x1	0101
3	0.03	0x1101x0	1x11
4	0.01	01x100xx	010x
5	0.08	0x1101xx	0x1x
6	0.10	0xx101x1	0xx1
7	0.04	0111xxxx	0xx0
8	0.01	11xx11xx	1xx1
9	0.02	111111xx	1xxx
10	0.06	xxx10110	xxx1
11	0.005	110101xx	01x0
12	0.007	111001x1	1101
13	0.003	111101x0	1011
14	0.001	11x1001x	000x

Ініціалізація

i	f_i	$c_i \in C$	$h_i \in H$
6	0.10	000101010001	100111011001
5	0.08	001101000010	101111001010
2	0.07	001001010101	101111011111
10	0.06	000101100001	000111110001
1	0.05	010101000100	110111001100
7	0.04	011100000000	111100001001
3	0.03	001101001011	101111011011
9	0.02	111111001000	111111001000
4	0.01	010100000100	110111001110
8	0.01	110011001001	110011001001
12	0.007	111001011101	111111011111
11	0.005	110101000100	111111001101
13	0.003	111101001011	111101011111
14	0.001	110100110000	110111111110

Ітерація 2 (кінцева)

i	f_i	$c_i \in C$	$h_i \in H$
1-5-6	0.33	011101010111	111111011111
1-5-10	0.27	011101000110	111111001111
1-5-7	0.25	011101000010	111111001011
1-10	0.11	010101100101	110111111101
3-10	0.09	001101101011	101111111101
1-7	0.09	011101000100	111111001101
4-7	0.05	011100000100	111111001111
8-9	0.03	111111001001	111111001001
2	0.07	001001010101	111111111111
12	0.007	111001011101	111111011111
11	0.005	110101000100	111111001101
13	0.003	111101001011	111101011111
14	0.001	110100110000	110111111110



Скорочене тестове покриття

i	1-5-6	3-10	4-7	8-9	2	11	12	13	14
t_i	011101x1	0x110110	011100xx	111111xx	0x1001x1	110101xx	111001x1	111101x0	11x1001x
r_i	0111	1x11	0100	1xx1	0101	01x0	1101	1011	000x

ВИСНОВКИ

В роботі за результатами виконаних теоретичних та практичних досліджень виконано розробку методу організації тестувань цифрових схем на основі еволюційних алгоритмів. При розробці методу переслідувалась мета, що полягає у підвищенні ефективності організації діагностування цифрових вузлів скороченням розмірностей використовуваних тестових послідовностей за рахунок застосування еволюційних алгоритмів на етапі організації тестувань цифрових схем.

Для реалізації програми досліджень виконано наступні роботи:

- проведене дослідження особливостей організації тестових діагностичних випробувань цифрових схем і способів організації тестів для них, а також можливостей еволюційних алгоритмів як інструментарію для організації тестів. Дослідження показало перспективність застосування еволюційних алгоритмів для скорочення розмірності тестів, необхідних для організації діагностичних випробувань цифрових схем дискретних схем, що відповідає меті магістерської роботи.
- визначено принципи поєднання математичного апарату технічної діагностики і еволюційних алгоритмів та розроблено математичну модель методу. В моделі запропоновано власні оператори, розроблені з урахуванням специфічних особливостей задачі застосування еволюційних алгоритмів організації тестів: оператор формування коду хромосоми з типових діагностичних векторів; оператор оцінювання можливості схрещування особин; оператор кросингвера векторів рекомбінації; формулу функції пристосованості.
- сформульовано основні положення методу організації тестувань на основі еволюційних алгоритмів, на основі яких розроблено алгоритм реалізації методу та надано його детальний опис.
- проведено апробацію запропонованого методу та алгоритмів його реалізації на основі реальних діагностичних даних, що засвідчило можливість досягти скорочення розмірностей використовуваних тестових послідовностей за рахунок застосування еволюційних алгоритмів на етапі організації тестів до 35,7%.

Додаток В
(обов'язковий)

Лістинг програми реалізації методу організації тестувань цифрових схем на основі еволюційних алгоритмів

Програма розроблена в середовищі Microsoft Visual Studio 2019, мовою С# і призначена для виконання в операційних системах сімейств Windows.

Вхідні вектори в даній програмі мають бути представлені як масив string, результати виводяться в файл Rezult.txt.

```
using System;
using System.Collections.Generic;
using System.IO;
using System.Text;

namespace Diagnostics
{
class MainMainVector<T>
{
    private T[] _array;

    public MainMainVector()
    {
        _array = new T[0];
    }

    public MainMainVector(int length)
    {
        _array = new T[length];
    }

    public T GetMainVector(int index)
    {
        return _array[index];
    }

    public T Back
    {
```

```
        get
        {
            return _array[_array.Length - 1];
        }
        set
        {
            _array[_array.Length - 1] = value;
        }
    }

    public T Front
    {
        get
        {
            return _array[0];
        }
        set
        {
            _array[0] = value;
        }
    }

    public void Crear()
    {
        _array = new T[0];
    }

    public void Insert(int pos, T item)
    {
        Array.Resize(ref _array, _array.Length + 1);
        for (int i = _array.Length - 1; i > pos; i--)
            _array[i] = _array[i - 1];
        _array[pos] = item;
    }

    public void PushBack(T item)
    {
        Array.Resize(ref _array, _array.Length + 1);
        _array[_array.Length - 1] = item;
    }

    public T GetSum()
    {
```

```

    T Sred = default;
    for (int i = 0; i < _array.Length; i++)
    {
        Sred = Sred + _array[i];
    }

    return Sred;
}

string RemoveDoubleSpaces(string str)
{
    if (string.IsNullOrEmpty(str))
    {
        return str;
    }

    // remove spaces
    RegexOptions options = RegexOptions.None;
    Regex regex = new Regex(" ]{2,}", options);
    str = regex.Replace(str, " ");
    return str;
}

string RemoveDoubleMinuses(string str)
{
    if (string.IsNullOrEmpty(str))
    {
        return str;
    }

    // remove spaces
    RegexOptions options = RegexOptions.None;
    Regex regex = new Regex("[-]{2,}", options);
    str = regex.Replace(str, "-");
    return str;
}

string SeoFilterString(string str)
{
    StringBuilder sb = new StringBuilder();
    var tmp = str ?? string.Empty;

    for (int i = 0; i < tmp.Length; i++)

```

```
{
    char ch = tmp[i];
    sb.Append(char.IsLetterOrDigit(ch) ? ch : '-');
}

return sb.ToString();
}

public void PushFront(T item)
{
    Insert(0, item);
}

public void Delete(int index)
{
    for (int i = index; i < _array.Length - 2; i++)
    {
        _array[i] = _array[i + 1];
    }
    Array.Resize(ref _array, _array.Length - 1);
}

public int Count
{
    get
    {
        return _array.Length;
    }
}

public T[] ToArray()
{
    return _array;
}
}

class Program
{
    static void Main(string[] args)
    {
        StreamWriter file = new StreamWriter("Data.txt");
        StreamReader fileIN = new StreamReader("Input.txt");
        StreamWriter fileOUT = new StreamWriter("Rezult.txt");
    }
}
```

```

/// Формування початкового масиву діагностичних даних (векторів)
{

/// Утворення початкового масиву кодів хромосом
Console.WriteLine("Зачекайте, програма працює...");
MainVector<BaseCode> InputCodes = new MainVector<BaseCode>();
for (int i = 0; i < 300; i++)
{
    BaseCode code = new BaseCode();
    code.Data = fileIN.ReadLine();
    InputCodes.PushBack(code);
}
for (int i = 0; i < InputCodes.Count; i++)
{
    InputCodes.GetMainVector(i).Chek = false;
    InputCodes.GetMainVector(i).History.Add((short)(i+1));
}

file.WriteLine("Початкові дані:");
WriteToFile(InputCodes);

///Функція проведення однієї ітерації схрещувань
void Stage(MainVector<BaseCode> Last, MainVector<BaseCode> Next)
{
    for (int i = 0; i < Last.Count - 1; i++)
    {
        for (int j = i + 1; j < Last.Count; j++)
        {
            if (Possible(Last.GetMainVector(i).Data, Last.GetMainVector(j).Data))
            {
                BaseCode code = new BaseCode();
                code.Union(Last.GetMainVector(i), Last.GetMainVector(j));
                Next.PushBack(code);
            }
        }
    }
    Array.ForEach(Last.ToArray(), item =>
    {
        if (item.Chek == false) Next.PushBack(item);
    });
    Last.Crear();
}

```

```

Array.ForEach(Next.ToArray(), item =>
{
    item.History.Sort();
});
int indx = 0;
while (indx < Next.Count - 1)
{
    int j = indx + 1;
    while (j < Next.Count)
    {
        if ((Next.GetMainVector(indx).Data == Next.GetMainVector(j).Data) &&
Equal(Next.GetMainVector(indx).History,          Next.GetMainVector(j).History))
Next.Delete(j);
        else j++;
    }
    indx++;
}
}

string Encode(string url)
{
    return HttpUtility.UrlEncode(url);
}

string StripHTML(string input)
{
    input = (input ?? "").Replace("<br>", "\r\n").Replace("<br />",
"\r\n").Replace("<br/>", "\r\n");
    input = Regex.Replace(input, "<style(.|\s)*?</style>", String.Empty);
    input = Regex.Replace(input, "<(.\s)*?>", String.Empty);
    input = Regex.Replace(input, "<.*?>", string.Empty);
    return input;
}

string To3DigitString(long str)
{
    StringBuilder sb = new StringBuilder(str.ToString());
    while (sb.Length < 3)
    {
        sb.Insert(0, "0");
    }
    return sb.ToString();
}
}

```

```

    ///Перевірка двох кодів на можливість схрещування
bool Possible(string s1, string s2)
{
    bool chek = false;
    for (int i = 0; i < s1.Length; i++)
    {
        if ((s1[i] == '0' && s2[i] == '1') || (s1[i] == '1' && s2[i] == '0')) {
chek = true; break; }
    }
    return !chek;
}

///Функція пошуку дублікатів
bool Equal(List<short> l1, List<short> l2)
{
    if (l1.Count != l2.Count) return false;
    else
    {
        MainVector<short> v1 = new MainVector<short>();
        int i = 0;
        foreach (short item in l1)
        {
            v1.PushBack(item);
            i++;
        }
        i = 0;
        foreach (short item in l2)
        {
            if (v1.GetMainVector(i) != item) return false;
            i++;
        }
        return true;
    }
}

bool SolutionCompare(string first, string second)
{
    int compareOrdinal = String.Compare(first, second,
StringComparison.Ordinal);
    if (compareOrdinal == 0)
    {
        return true;
    }
}

```

```
}

if (first == null || second == null)
{
    return false;
}

int m = first.Length;
int n = second.Length;

if (Math.Abs(m - n) > 1)
{
    return true;
}

int edits = 0;
char characterOne = default;
char characterTwo = default;

int i = 0, j = 0;

while (i < m && j < n)
{
    if (first[i] != second[j])
    {
        if (m > n)
        {
            i++;
        }
        else if (m < n)
        {
            j++;
        }
        else
        {
            i++; j++;
        }

        edits++;

        if (m == n)
        {
            characterOne = first[i - 1];
        }
    }
}
```

```
        characterTwo = second[j - 1];
    }
    else
    {
        characterOne = m > n ? first[i - 1] : second[j - 1];
    }
}

else
{
    i++; j++;
}
}

if (i < m)
{
    edits++;

    if (m == n)
    {
        characterOne = first[i - 1];
        characterTwo = second[j - 1];
    }
    else
    {
        characterOne = m > n ? first[i - 1] : second[j - 1];
    }
}

if (j < n)
{
    edits++;

    if (m == n)
    {
        characterOne = first[i - 1];
        characterTwo = second[j - 1];
    }
    else
    {
        characterOne = m > n ? first[i - 1] : second[j - 1];
    }
}
}
```

```

        return edits > 1;
    }

    /// Перше схрещування
    MainVector<BaseCode> FirstStage = new MainVector<BaseCode>();
    Stage(InputCodes, FirstStage);
    file.WriteLine("-----");
-----");
    file.WriteLine();
    file.WriteLine("Перше схрещування:");
    WriteToFile(FirstStage);

    /// Друге схрещування
    MainVector<BaseCode> SecondStage = new MainVector<BaseCode>();
    Stage(FirstStage, SecondStage);
    file.WriteLine("-----");
-----");
    file.WriteLine();
    file.WriteLine("Друге схрещування:");
    WriteToFile(SecondStage);

    /// Додаткові схрещування
    MainVector<BaseCode> ThirdStage = new MainVector<BaseCode>();
    Stage(SecondStage, ThirdStage);
    MainVector<BaseCode> FourthStage = new MainVector<BaseCode>();
    Stage(ThirdStage, FourthStage);
    MainVector<BaseCode> FifthStage = new MainVector<BaseCode>();
    Stage(FourthStage, FifthStage);
    MainVector<BaseCode> Stage6 = new MainVector<BaseCode>();
    Stage(FifthStage, Stage6);
    MainVector<BaseCode> Stage7 = new MainVector<BaseCode>();
    Stage(Stage6, Stage7);
    file.WriteLine("-----");
    file.WriteLine();
    file.WriteLine("Результат усіх схрещувань:");
    WriteToFile(Stage7);
    WriteToFileOUT(Stage7);
    Console.WriteLine("Виконання завершено...");
    file.Close();
    fileIN.Close();
    fileOUT.Close();
    Console.ReadKey(true);

```

```

/// Виведення результату у файли
void WriteToFile(MainVector<BaseCode> vector)
{
    for (int i = 0; i < vector.Count; i++)
    {
        string code1 = "", code2 = "", history = "";
        foreach (char c in vector.GetMainVector(i).Data)
        {
            if (c == 'x') code1 += '0'; else code1 += c;
            if (c == 'x') code2 += '0'; else code2 += '1';
        }
        foreach (short h in vector.GetMainVector(i).History)
        {
            history += h.ToString(); history += ' ';
        }
        while (history.Length < 20) history += ' ';
        file.WriteLine(history + ' ' + code1 + "      " + code2);
    }
}

void WriteToFileOUT(MainVector<BaseCode> vector)
{
    for (int i = 0; i < vector.Count; i++)
    {
        string code1 = "", history = "";
        int j = 0;
        foreach (char c in vector.GetMainVector(i).Data)
        {
            j++;
            code1 += c;
            if (j == 32) code1 += " ";
        }
        foreach (short h in vector.GetMainVector(i).History)
        {
            history += h.ToString(); history += ' ';
        }
        while (history.Length < 20) history += ' ';
        fileOUT.WriteLine(history + ' ' + code1);
    }
}
}

```

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
ХМЕЛЬНИЦЬКИЙ НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ

РЕЦЕНЗІЯ НА ДИПЛОМНУ РОБОТУ
освітньо-кваліфікаційного рівня «магістр»

Магістр Моторний Павло Володимирович

Тема Метод організації тестувань цифрових схем на основі еволюційних алгоритмів

Спеціальність 123 – Комп'ютерна інженерія

Обсяг дипломної роботи освітньо-кваліфікаційного рівня «магістр»:

кількість листів креслень 9; кількість сторінок записки 119

1. Короткий зміст ДР та прийнятих рішень У дипломній роботі розроблено метод організації тестувань цифрових схем на основі еволюційних алгоритмів

2. Висновок про відповідність ДР дипломному завданню Дипломна робота у повній мірі відповідає поставленому завданню як в теоретичній, так і в практичній частині дипломної роботи

3. Характеристика виконання кожного розділу роботи, ступінь використання останніх досягнень науки і техніки і передових методів роботи: У вступі подана загальна характеристика поставленої задачі, чітко визначено об'єкт, предмет та методи дослідження, сформульована актуальність. Визначені задачі, які необхідно вирішити для досягнення поставленої мети, практична цінність отриманих результатів, їхня новизна та наведені відомості про публікації. У першому розділі проведено огляд використовуваних в комп'ютерних системах методів організації тестувань на основі еволюційних алгоритмів, виконане обґрунтування актуальності теми дослідження і виконана постановка задачі. В другому розділі виконана розробка математичної моделі для реалізації методу, яка базується на основних положеннях теорії ймовірності та математичної статистики, теорії інформації та кодування, криптографії та прикладної криптології. В третьому розділі визначено основні положення методу та розроблено алгоритми його реалізації. Четвертий розділ присвячено апробації методу та алгоритмів його реалізації моделюванням.

4. Позитивні сторони проекту Дипломна робота має комплексну наукову і практичну цінність. Наукова цінність полягає у розробці методу організації тестувань цифрових схем на основі еволюційних алгоритмів. Практична цінність присвячена підвищенню ефективності організації тестувань цифрових вузлів скороченням розмірностей використовуваних тестових послідовностей на основі застосування еволюційних алгоритмів на етапі організації тестувань цифрових схем

5. Негативні сторони проекту В роботі відсутній результатів практичної апробації методу

6. Оцінка графічного оформлення та пояснювальної записки роботи Графічне оформлення виконане відповідно до теми дипломної роботи з дотриманням стандартів. В загальному графічне оформлення виконане якісно, пояснювальна записка відповідає нормам щодо її оформлення.

7. Відгук про роботу в цілому В загальному дипломна робота заслуговує позитивної оцінки. Весь матеріал дипломної роботи структурований, чіткий та послідовний. Усі розділи роботи послідовні та логічні, що дозволяє чітко розуміти викладений матеріал в рамках тематики дипломної роботи. Графічний матеріал дозволяє наочно побачити доцільність та ефективність рішень, які були прийняті за основу для досягнення поставленої мети.

8. Інші зауваження Окремі описи в пояснювальній записці подано занадто деталізовано, що ускладнює сприйняття матеріалу наукової роботи фахівцями в обраній предметній галузі. Не на всі джерела з переліку є посилання в тексті пояснювальної записки

9. Оцінка дипломної роботи Враховуючи всі позитивні та негативні сторони представленої дипломної роботи, можна зробити висновок, що вона заслуговує оцінку «добре».

РЕЦЕНЗЕНТ (прізвище, ім'я, по батькові, посада, місце роботи)

Гурман Іван Васильович

кандидат технічних наук, доцент,

доцент кафедри інженерії програмного забезпечення

« 3 » 2021.

(підпис)

Anti-Plagiarism v-15.257

Максимальное совпадение с одним документом 3.0%

Словари проверки: en_US, ru_RU, ua_UA. **Ошибок в документах: 7%**

ID: 97780 Название: Метод організації тестування цифрових схем на основі еволюційних алгоритмів Добавлено в БД: 2021-12-02 Авторы: Моторний П.В. Руководители: Чешун В.М. Консультанты: Оponentы:	Документ		Суммарное совпадение по Базе Данных	
	Символы	Лексемы	Символы	Лексемы
	136216	951	7084 (5%)	91 (10%)

Источник плагиата

ID	Описание	Наличие плагиата в документе	
		Символы	Лексемы



Ім'я користувача:
Кафедра кібербезпеки

ID перевірки:
1009466402

Дата перевірки:
02.12.2021 10:00:29 EET

Тип перевірки:
Doc vs Internet + Library

Дата звіту:
02.12.2021 10:05:38 EET

ID користувача:
100008300

Назва документа: **Моторний - ПЗ**

Кількість сторінок: 99 Кількість слів: 20910 Кількість символів: 155860 Розмір файлу: 8.75 MB ID файлу: 1009480865

8.7% Схожість

Найбільша схожість: 3.39% з Інтернет-джерелом (<http://elar.khnu.km.ua/bitstream/123456789/8247/1/VOTTP-2018-N2%...>)

7.52% Джерела з Інтернету 297 Сторінка 101

2.42% Джерела з Бібліотеки 106 Сторінка 103

0% Цитат

Вилучення цитат вимкнене

Вилучення списку бібліографічних посилань вимкнене

0% Вилучень

Немає вилучених джерел

Модифікації

Виявлено модифікації тексту. Детальна інформація доступна в онлайн-звіті.

Замінені символи 292

РІШЕННЯ ЕКСПЕРНОЇ КОМІСІЇ

КАФЕДРИ КІБЕРБЕЗПЕКИ ТА КОМП'ЮТЕРНИХ СИСТЕМ І МЕРЕЖ

ПРО ДОПУСК КВАЛІФІКАЦІЙНОЇ РОБОТИ ДО ЗАХИСТУ

Підтверджуємо ознайомлення з результатом звіту подібності щодо роботи, генерованого системою виявлення текстових збігів/ідентичності/схожості:

Назва: Метод організації тестувань цифрових схем на основі еволюційних алгоритмів

Автор: Моторний Павло Володимирович

Спеціальність: 123 – Комп'ютерна інженерія

Освітня програма: Програмування та захист комп'ютерних систем і мереж

Науковий керівник: Чешун Віктор Миколайович, к.т.н., доцент

Після аналізу звіту подібності зроблено такий висновок:

№	Висновок	Позначка про відповідність
1	Запозичення, виявлені в роботі, є законними і не є плагіатом (далі – зазначаються підстави віднесення запозичень до правомірних). Робота приймається до захисту.	відповідає
2	Виявлені запозичення не є плагіатом, розміщені в розділах, які не описують безпосередньо авторське дослідження, але кількість цитат перевищує обсяг, виправданий поставленою метою роботи (далі – зазначаються детальні та аргументовані підстави віднесення запозичень до правомірних). Робота приймається до захисту, але має бути відкоригована. Відкоригований варіант має бути поданий на кафедру за 2 дні до захисту, разом із заявою щодо самостійності виконання письмової роботи та ідентичності друкованої та електронної версії роботи.	
3	Виявлені запозичення не є плагіатом, але частково розміщені в розділах, які описують безпосередньо авторське дослідження, а кількість цитат перевищує обсяг, виправданий поставленою метою роботи. В зв'язку з цим мета роботи та поставлені завдання не були досягнені. Робота може бути допущена до захисту (наступного року) після того як буде відкоригована та допрацьована і успішно пройде повторну перевірку на академічний плагіат.	
4	Робота містить навмисні текстові спотворення, передбачувані спроби укріття запозичень або інші прояви академічного плагіату. Робота містить фабрикацію або фальсифікацію даних. Робота не допускається до захисту.	
5	Інше:	

Підтвердження:

Запозичення, виявлені в роботі, є законними і не є плагіатом, оскільки:

- 1) запозичення розміщені в розділах аналізу існуючих аналогів та прототипів, які не описують безпосередньо авторське дослідження і не стосуються результатів роботи;
- 2) усі запозичення фрагментарні, або мають належним чином оформленні посилання;
- 3) окремі виявлені збіги є загальноживаними фразами або виразами, про що свідчить посилання системи на збіг з 10-20 джерелами на один фрагмент речення;
 - 4) в якості запозичень зафіксовано опис джерел інформації, наданий у розділі «ПЕРЕЛІК ДЖЕРЕЛ ПОСИЛАННЯ», а також стандартні формулювання з типових форм супроводу кваліфікаційної роботи (титульна сторінка, завдання тощо);
- 5) всі зафіксовані системою ознаки модифікації тексту відносяться до комбінування латинських символів зі україномовними скороченнями індексів в формулах, що не є модифікацією тексту.

Сумарний обсяг всіх запозичень, визначений системою виявлення збігів/ідентичності/схожості, складає 8.7% і адресується до 297 першоджерел, що, з урахуванням наведених обґрунтувань, відповідає характеру наукового дослідження і свідчить на користь кваліфікаційної роботи.

Керівник роботи

Завідувач кафедри КБКСМ, гарант ОП

Дата: 03.12.2021



В.М. Чешун

Ю.П. Кльоц